Robust Gaussian Process Regression and its Application in Data-driven Modeling and Optimization

by

Rishik Ranjan

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

Availability of large amounts of industrial process data is allowing researchers to explore new data-based modelling methods. In this thesis, Gaussian process (GP) regression, a relatively new Bayesian approach to non-parametric data based modelling is investigated in detail. One of the primary concerns regarding the application of such methods is their sensitivity to the presence of outlying observations. Another concern is that their ability to predict beyond the range of observed data is often poor which can limit their applicability. Both of these issues are explored in this work.

The problem of sensitivity to outliers is dealt with by using a robust GP regression model. The common approach in literature for identification of this model is to approximate the marginal likelihood and maximize it using conjugate gradient algorithm. In this work, an EM algorithm based approach is proposed in which an approximate lower bound on the marginal likelihood is iteratively maximized. Models identified using this method are compared against those identified using conjugate gradient method in terms of prediction performance on many synthetic and real benchmark datasets. It is observed that the two approaches are similar in prediction performance. However the advantages of EM approach are numerical stability, ease of implementation and theoretical guarantee of convergence.

The application of proposed robust GP regression in chemical engineering is also explored. An optimization problem for an industrial water treatment and steam generation network is formulated. Process models are constructed using material balance equations and used for data reconciliation and steady state optimization of the cost of steam production. Since the overall network is under manual operation, a dynamic optimization framework is constructed to find a set point change strategy which operators can use for minimizing steam production cost. Dynamic models for process units and tanks are integrated into this framework. Some of these models are identi-

fied using proposed robust GP regression method. Extrapolation ability of identified GP models is improved by applying a suitable GP kernel structure and by using some ad hoc scaling techniques. Based on the application of robust GP regression to an industrial optimization problem, it is shown that non-parametric data-based modelling can be successfully integrated with process optimization objectives.

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Biao Huang for giving me the freedom to explore different research possibilities and at the same time helping me focus on the task at hand whenever I found myself lost in a problem. I am also grateful to him for entrusting me with a challenging yet interesting industrial project. My special thanks also goes Dr. Alireza Fatehi for giving valuable suggestions and encouragement throughout my research journey. I greatly appreciate his help in reviewing my paper and thesis.

This work would not have been possible without the help of several past and present members of the Computer Process Control group. Many of my research ideas were generated from constructive discussions I had with Dr. Swanand Khare, Rahul Raveendran and Shekhar Sharma. I am also thankful to Yaojie Lu, Abhinandhan Raghu, Nima Sammaknejad, Ruomu Tan, Anahita Sadeghian, Elham Naghoosi, Mohammed Rashedi, Yujia Zhao, Ouyang Wu and several others in our research group for their help and support. My journey through graduate studies would have felt long and arduous without the joyous company of my friends: Siddhant Panda, Sahil Bangar, Nitin Arora and Satarupa Dhir.

I would like to thank Dr. Ramesh Kadali, Dr. Fei Qi and Eliyya Shukeir from Suncor for their help and guidance. Financial support provided by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

Last but not least, I would like to express my deepest gratitude to my parents for their unconditional love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Plant wide optimization is an important research subject in process industries. With the increase in complexity of plant processes and networks, it has become difficult for operators to optimize process operations. Real-time optimization strategies rely on the use of complex but accurate first principle plant models which can be difficult to construct. With the availability of large amounts of process data from historical operation, it has become feasible to construct data-driven regression models for processes and use it in optimization strategies.

The use of data-driven models comes with its own unique challenges. Very often the relationship between process variables is not known. Arbitrary assumption of a model structure, may lead to inaccurate representation of the process. Consequently, the use of neural networks, support vector regression and fuzzy models for modeling unknown non-linear systems has increased in recent years. Several works have focused on the application of such models in chemical engineering problems [1, 2, 3, 4]. These models can automatically learn the relationship between inputs and output and approximate non-linear systems. However, it is not easy to analyze such models statistically. Cross-validation is required for parameter estimation and it can be difficult to avoid over-fitting. Gaussian process (GP) regression models offer a viable alternative to these techniques. It allows a fully Bayesian approach for non-linear modeling which can help address most of the drawbacks mentioned above without compromising on prediction performance. This thesis focuses on GP models for performing nonlinear regression.

Process data are often noisy and contaminated with outliers arising out of instrument malfunctioning or process disturbances. Since most data driven models including GP are sensitive to the presence of outliers in data, the problem of robustness is an important issue. The primary objective of this thesis is to find a method for robust identification of GP models. The secondary objective is application of robust GP models in process optimization. The achievement of these two objectives can help in the use of data-driven models in optimization.

1.2 Thesis contribution

The main contributions of this thesis are the development of an identification method for robust GP regression model and the formulation of a plant wide optimization problem involving nonlinear process models identified using the proposed method. Specifically, the contributions of this thesis can be summarized as follows:

1. An EM based algorithm is proposed for identification of robust GP models. Two different distributions namely Student's t-distribution and Laplace distribution are used to model noise characteristics with outliers. A novel lower bound on the Q function is proposed for the t-distribution case.

2. The proposed approach is compared against conjugate gradient based method for robust GP identification. Convergence of the two methods is analyzed based on the effects of step size, step direction and gradient of their respective objective functions.

3. An optimization problem is formulated using industrial data from a water treatment and steam generator network. Data reconciliation and steady state optimization is performed.

4. A manual set point changing strategy for achieving optimal operation of network is obtained using a novel optimization framework.

5. Both linear and nonlinear dynamic process models are used in optimization. Nonlinear process identification is performed using robust GP regression identified based on the proposed method. This application gave several insights into

the properties of GP regression such as its ability to handle noisy time series data and the effect of the choice of kernel function in Gaussian process prior.

1.3 Thesis outline

The rest of the thesis is organized as follows:
Chapter 2 deals with identification of robust GP regression model with EM algorithm.
Chapter 3 introduces a SAGD water treatment network optimization problem. It includes model description for various units in the network. Data reconciliation and steady state optimization results are also presented in this chapter.
Chapter 4 addresses the problem of finding a manual set point change strategy for optimization of SAGD water treatment network. Dynamic process models which are required for this task are identified using robust GP regression.
Chapter 5 concludes the thesis.

Chapter 2

Robust Gaussian process regression using EM algorithm

2.1 Introduction

With the availability of large amounts of data, industries are increasingly looking towards new ways for extracting useful information. There has been significant interest in non-parametric regression methods since they are completely data-driven and do not assume any knowledge about the functional relationship between variables. Support vector regression [5], artificial neural networks [6] and Gaussian process (GP) regression [7] are some such techniques which are widely used in literature. Gaussian processes are unique in that they offer a probabilistic framework for non-parametric regression. They have been shown to be a powerful nonlinear regression technique [7].

GP regression as a non-parametric regression technique has been around for a long time with roots in geostatistics [8], where it is known as Kriging, derived from Krige who first introduced this technique [9]. It was designed for use as a statistical interpolation tool to estimate the probable distribution of gold based on samples from a few underground locations. Today, applications of Gaussian process modeling can be found in machine learning, pattern recognition [10, 11], remote sensing [12] and neural image processing [13] to name a few. It is also widely used as a metamodeling tool in the design and analysis of computer experiments [14]. Recently, it has seen applications in chemical engineering problems as well [15, 16, 17, 18, 19]. One of the earliest applications was in multivariate spectroscopic calibration [19]. Since then, GP

regression has been used in several areas ranging from soft sensor development for industrial processes [17] to estimation of state of health of Lithium-ion batteries [16]. Other authors have used it to develop algorithms for the optimization of constrained computer experiment systems [15, 20]. The use of GP regression model from the point of view of system identification and control has also been explored [21, 22, 18]. Some of these applications use industrial data for training which can be noisy. To account for this, noise in observations is often assumed to follow a normal distribution. However, predictions under this noise assumption are highly susceptible to the presence of "outliers" or extreme observations in data.

Just as in other regression models, in GP regression, outliers are handled by selecting a noise model which accounts for the possibility of extreme observations. Mathematically, this amounts to using a noise model with heavier tails compared to Gaussian distribution. A mixture of normal distributions is often used to build robust models [23]. A more general choice is the Student's t-distribution in which the degree of freedom hyper-parameter ν can be used to adjust the probability of observing extreme values [24]. The Laplace distribution is also a suitable choice since it has heavier tails compared to normal distribution. This distribution is well studied in the context of least absolute deviations estimate in robust regression [25].

Regardless of the choice of noise model, learning hyper-parameters for Gaussian process regression is an important problem. Hyper-parameters for GP regression are usually estimated by optimizing the log marginal likelihood of evidence [26, 27]. For noise following normal distribution, a closed form expression for the log marginal likelihood is available. However, in the case of robust GP regression the tractability of the log marginal likelihood is lost. The common approach in literature is to find an approximation for the log marginal likelihood and maximize it with respect to hyper-parameters using any gradient based method. Kuss [27] has compared the different approximations to the log marginal likelihood for robust GP regression. Two of them are Laplace's method and Expectation Propagation (EP). Laplace's method is a well known approximation technique explained in most introductory machine learning books [28, chap. 27]. Expectation Propagation method is another powerful approximation technique first proposed by [29]. Several works have focused on improving these approximation techniques in the context of robust GP regression

[30, 31].

In the context of sparse GP regression, [32] proposed a different approach to learning hyper-parameters wherein a lower bound on the log marginal likelihood was maximized using EM (Expectation Maximization) algorithm. Later, [33] and [34] provided a similar EM algorithm implementation for binary GP classification and ordinal GP regression respectively.

In this chapter, we present an EM based algorithm for training a Gaussian process regression model which is robust to outliers. We explore the use of both Laplace's approximation method and Expectation Propagation approximation method. To the best of our knowledge, there exists no EM implementation for robust GP regression. Moreover, there exists no comparison of results from the two competing approaches for hyper-parameter estimation, viz., maximization of approximate log marginal likelihood using conjugate gradient method and EM algorithm based lower bound maximization. We demonstrate our approach using t-distribution and Laplace distribution for noise modeling. Steps involved in EM algorithm are explained in detail and the proposed and existing methods are compared on the basis of prediction performance. It is often argued that EM convergence can be extremely slow [35]. Several techniques have been proposed to speed up EM [36, 37]. One such method known as Expectation Conjugate Gradient (ECG) [38] is implemented in this work to address this issue.

The effects of initial guess, gradients, step size and step direction on EM and gradient based algorithms are also explored. Simulation results on various data sets and implementation on an industrial system identification problem show that the proposed method can give reliable estimates for hyper-parameters.

The rest of the chapter is arranged in the following manner. First we introduce the problem in mathematical terms in Section 2.2. Section 2.3 gives an overview of approximations used for robust GP regression. These approximations are used in Section 2.4 to derive EM algorithm steps for two robust noise models namely, t-distribution and Laplace distribution. Section 2.4 also contains a description of the ECG algorithm which is closely related to EM. Regression results from our EM implementation on synthetic and real data-sets are given in Section 2.5. Results from competing approaches are also provided. Section 2.6 contains a discussion on the advantages and disadvantages of EM algorithm. In Section 2.7, a non-linear dynamic

process model is identified using proposed GP regression. Section 2.8 concludes the chapter.

2.2 Problem description

Before explaining the approach, we briefly describe the problem statement. In a typical regression problem,

$$y = f(\mathbf{x}, \theta) + \epsilon \tag{2.1}$$

$\mathbf{x} \in \mathbf{R}^D$ is input, $y \in \mathbf{R}$ is scalar output, $f : \mathbf{R}^D \to \mathbf{R}$ is a function with fixed parametric model structure with parameter θ and ϵ is a random variable representing noise in the output. If it is assumed that noise is independent and identically distributed, then the likelihood of observing outputs $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ at $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ can be written as:

$$p(\mathbf{y}|\mathbf{X}, \theta_e) = \prod_{i=1}^{n} p\left(y_i|f(\mathbf{x}_i), \theta_e\right) \tag{2.2}$$

where θ_e is the hyper-parameter for noise distribution. Typically, this expression is maximized to estimate the model parameters and the noise hyper-parameters. However, in GP regression, no parametric model structure is assumed for the function. Instead a Gaussian prior distribution is specified over the vector $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T$ where f_i is a random variable associated with the value of "latent" function f at input location $\mathbf{x}_i$. In other words, it is assumed that $\mathbf{f}$ is a random variable which follows a multivariate Gaussian distribution. Usually the mean is assumed to be zero and each element of the covariance matrix is an exponentially decreasing function of the distance between points in the input space. This function is governed by a set of hyper-parameters θ_{cov}. Therefore, the Gaussian prior distribution can be mathematically expressed as,

$$p\left(\mathbf{f}|\mathbf{X}, \theta_{cov}\right) = \mathcal{N}(\mathbf{0}, K(\mathbf{X}, \mathbf{X})) \tag{2.3}$$

where, K is the covariance matrix defined by the function,
$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_{se}^2 exp\left(-\frac{1}{2l}(\mathbf{x}_i - \mathbf{x}_j)^2\right)$ and $\theta_{cov} = [\sigma_{se}^2, l]$. Other covariance functions can also be used. More details regarding Gaussian processes can be found in the book

by Rasmussen and Williams [26]. Using the likelihood and prior distributions from Equations 2.2 and 2.3, the marginal likelihood of evidence is given by:

$$p(\mathbf{y}|\mathbf{X},\Theta) = \int p(\mathbf{y}|\mathbf{f},\theta_e)p(\mathbf{f}|\mathbf{X},\theta_{cov})d\mathbf{f} \tag{2.4}$$

where $\Theta = [\theta_{cov}, \theta_e]$. As explained earlier, hyper-parameters for GP regression are estimated by optimizing the log marginal likelihood of evidence. Thus we have,

$$\Theta^* = [\theta^*_{cov}, \theta^*_e] = \arg\max_{\theta_{cov},\theta_e} \log p(\mathbf{y}|\mathbf{X},\theta_{cov},\theta_e) = \log \int p(\mathbf{y}|\mathbf{f},\theta_e)p(\mathbf{f}|\mathbf{X},\theta_{cov})d\mathbf{f} \tag{2.5}$$

This integral is tractable with Gaussian noise assumption but intractable in the case of the heavy-tailed distributions for example, Student's t and Laplace distributions used in this work. Typically, researchers approximate the log marginal likelihood directly and optimize it using the conjugate gradient method. In this work, we use an approximation for the posterior distribution of the latent variable $p(\mathbf{f}|\mathbf{y},\mathbf{X},\Theta)$ to find a lower bound to the log marginal likelihood and maximize it iteratively using EM algorithm.

2.3 Approximation techniques for distributions: An overview

There are several approximation techniques which can be used with robust GP regression. In literature, these methods have been used for approximating the log marginal likelihood $p(\mathbf{f}|\mathbf{y},\mathbf{X},\Theta)$ [27, 30, 31]. This work uses deterministic techniques namely Laplace's method [28, chap. 27] and Expectation Propagation method [29] to approximate the posterior distribution of the latent variable $p(\mathbf{f}|\mathbf{y},\mathbf{X},\Theta)$. These techniques are described in this section.

2.3.1 Laplace's Method

Laplace approximation can be used to find a Gaussian approximation to the posterior of the latent variable $\mathbf{f}$ by doing a second-order Taylor expansion of $\log p(\mathbf{f}|\mathbf{y},\mathbf{X})$ around the mode of the posterior. The covariance of this approximation is given by the curvature of the true posterior at the mode:

$$p(\mathbf{f}|\mathbf{y},\mathbf{X},\Theta) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{h},\mathbf{A}) \tag{2.6}$$

where $\mathbf{h} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta)$ and $\mathbf{A} = -\nabla\nabla \log p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta)\Big|_{\mathbf{f}=\mathbf{h}}$ are the mean and the Hessian of the negative log posterior at $\mathbf{f} = \mathbf{h}$. In this work, a MATLAB function based on a numerically stable Laplace approximation approach [31] using t-distribution as noise model is implemented. This function provides the mean $\mathbf{h}$ and covariance matrix $\mathbf{A}$ of the approximation.

2.3.2 Expectation Propagation method

Expectation Propagation (EP) method also gives a Gaussian approximation for the posterior distribution of the latent variable $\mathbf{f}$. However, it is based on an iterative algorithm that exploits the factorization structure of the target distribution. In the case of robust GP regression, the target distribution is $p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta)$. From Bayes' rule we know that the posterior is proportional to the prior multiplied by the likelihood:

$$p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta) \propto p(\mathbf{f}|\mathbf{X}, \theta_{cov}) \prod_{i=1}^{n} p(y_i|f_i, \theta_e) \tag{2.7}$$

In EP for Gaussian process, the likelihood for each observation $p(y_i|f_i, \theta_e)$ is replaced by a so-called site function $\tilde{t}_i(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma}_i^2)$ in Equation 2.7. The site function is an un-normalized Gaussian $\tilde{t}_i(f_i) = Z_i \mathcal{N}(f_i|\tilde{\mu}_i, \tilde{\sigma}_i^2)$. Thus, we get an approximation $q(\mathbf{f})$:

$$q(\mathbf{f}) \propto p(\mathbf{f}|\mathbf{X}, \theta_{cov}) \prod_{i=1}^{n} \tilde{t}_i(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma}_i^2) \tag{2.8}$$

EP algorithm visits each site iteratively and adjusts the site parameters to match the moments of the approximation (Equation 2.8) with that of posterior (Equation 2.7). Once converged, it gives a Gaussian approximation to the posterior:

$$p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{h}, \mathbf{A}) \tag{2.9}$$

where $\mathbf{h}$ and $\mathbf{A}$ are functions of site parameters $\tilde{\mu}_i$'s , $\tilde{\sigma}_i^2$'s and prior covariance matrix $\mathbf{K}$. The derivations involved differ with the type of likelihood distribution. In this work, the implementation available in GPML toolbox [39] has been utilized. The EP inference function in the toolbox provides the mean $\mathbf{h}$ and covariance matrix $\mathbf{A}$ of the approximation.

2.4 EM algorithm derivation for robust GP regression

The proposed approximate EM algorithm for learning hyper-parameters for robust GP regression is presented in this section. The algorithm has been derived for two different noise models namely t-distribution and Laplace distribution. Before showing the EM step derivations, we give the lower bound on the log marginal likelihood. As shown earlier, the optimization of log marginal likelihood with respect to hyper-parameters can be written as:

$$\begin{aligned}\Theta^* &= \arg\max_{\Theta} \log p(\mathbf{y}|\mathbf{X}, \Theta) \\ &= \arg\max_{\Theta} \log \int p(\mathbf{y}|\mathbf{f}, \theta_e) p(\mathbf{f}|\mathbf{X}, \theta_{cov}) \, \mathrm{d}\mathbf{f}\end{aligned} \tag{2.10}$$

where $\Theta = [\theta_{cov}, \theta_e]$. Multiplying and dividing the integrand by the approximate posterior distribution $q(\mathbf{f})$ for $p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \theta_{cov}, \theta_e)$ and using Jensen's inequality we get the lower bound [33]:

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X}, \Theta) &= \log \int q(\mathbf{f}) \frac{p(\mathbf{y}|\mathbf{f}, \theta_e) p(\mathbf{f}|\mathbf{X}, \theta_{cov})}{q(\mathbf{f})} \, \mathrm{d}\mathbf{f} \\ &\geq \int q(\mathbf{f}) \log \left(\frac{p(\mathbf{y}|\mathbf{f}, \theta_e) p(\mathbf{f}|\mathbf{X}, \theta_{cov})}{q(\mathbf{f})} \right) \mathrm{d}\mathbf{f}\end{aligned} \tag{2.11}$$

This lower bound can be further simplified as follows

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X}, \Theta) &\geq \int q(\mathbf{f}) \log \left(p(\mathbf{y}|\mathbf{f}, \theta_e) p(\mathbf{f}|\mathbf{X}, \theta_{cov}) \right) \mathrm{d}\mathbf{f} - \int q(\mathbf{f}) \log q(\mathbf{f}) \, \mathrm{d}\mathbf{f} \\ &= \mathbf{E}_q \left[\log \left(p(\mathbf{y}, \mathbf{f}|\mathbf{X}, \Theta) \right] + \mathcal{H} \right. \\ &\geq \mathbf{E}_q \left[\log \left(p(\mathbf{y}, \mathbf{f}|\mathbf{X}, \Theta) \right] \right.\end{aligned} \tag{2.12}$$

$\mathbf{E}_q \left[\log \left(p(\mathbf{y}, \mathbf{f}|\mathbf{X}, \Theta) \right) \right]$ denotes the expectation of the joint distribution of $\mathbf{y}$ and $\mathbf{f}$ with respect to $q(\mathbf{f})$ and $\mathcal{H}$ represents the entropy of the approximate posterior distribution $q(\mathbf{f})$ which is always non-negative. Note that the expectation term is similar to the actual lower bound in the EM algorithm [40] which uses the true posterior of the latent function: $\mathbf{E}_{p(\mathbf{f}|\mathbf{y}, \mathbf{X})} \left[\log \left(p(\mathbf{y}, \mathbf{f}|\mathbf{X}, \Theta) \right) \right]$. By replacing the true posterior with $q(\mathbf{f})$, we get an approximate EM algorithm consisting of the following steps:

Repeated until Θ converges:

1. **E-step:**
   (a) Given Θ^t find approximate posterior distribution $q(\mathbf{f}) = \mathcal{N}(\mathbf{h}, \mathbf{A})$ using either Laplace approximation or EP.
   (b) Use $q(\mathbf{f})$ to find the lower bound $\mathbf{E}_{q(\mathbf{f})}\left[\log\left(p(\mathbf{y}, \mathbf{f}|\mathbf{X}, \Theta)\right]\right.$. This expression is referred to as Q function: $\mathcal{Q}(\Theta|\Theta^t)$
2. **M-step:** $\Theta^{t+1} = \arg\max_{\Theta} \mathcal{Q}(\Theta|\Theta^t)$

In the M-step, the Q function can be separately maximized with respect to the noise hyper-parameters θ_e and the GP prior hyper-parameters θ_{cov} since

$$\mathcal{Q}(\Theta|\Theta^t) = \int log\left(p(\mathbf{y}|\mathbf{f}, \theta_e)\right) \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})\, \mathrm{d}\mathbf{f} + \int log\left(p(\mathbf{f}|\mathbf{X}, \theta_{cov})\right) \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})\, \mathrm{d}\mathbf{f} \quad (2.13)$$

$$\mathcal{Q}(\Theta|\Theta^t) = \mathcal{Q}_e + \mathcal{Q}_{cov} \quad (2.14)$$

In the next two subsections we give detailed derivations of the above algorithm for t-distribution and Laplace distribution noise models and find the closed-form expression for their Q functions which are easily optimized using any gradient based optimizer.

2.4.1 EM steps for Student's t-likelihood

The expression for the t-distribution based noise model is as follows:

$$p(y_i|f_i, \nu, \sigma) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma} \left(1 + \frac{(y_i - f_i)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \quad (2.15)$$

Note that there are two noise hyper-parameters $\theta_e = [\nu, \sigma]$. ν is the degree of freedom and σ is the scaling parameter. In our approach Laplace's method was used for approximating the posterior. EP was not used since it fails to converge for non log concave distributions such as Student's t-distribution [27]. A robust EP implementation has been proposed to solve this problem [30]. However, it requires adaptive selection of step sizes in difficult cases.

In this work, we use Laplace approximation to approximate the posterior distribution $p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta^t)$. In order to use EM algorithm we need to evaluate $\mathcal{Q}_{cov}$ and $\mathcal{Q}_e$ in Equation 2.14. The expression for $\mathcal{Q}_{cov}$ can be easily obtained in closed form

[33]. Closed-form expression for $\mathcal{Q}_e$ requires calculating the following expectation previously shown in Equation 2.13:

$$\begin{aligned}\mathcal{Q}_e(\theta_e|\Theta^t) &= \int log\left(p(\mathbf{y}|\mathbf{f},\theta_e)\right)\mathcal{N}(\mathbf{f}|\mathbf{h},\mathbf{A})\,\mathrm{d}\mathbf{f} \\ &= \sum_{i=1}^{n}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log\left(p(y_i|f_i,\nu,\sigma)\right)\right]\end{aligned} \tag{2.16}$$

Here, $\mathcal{N}(\mathbf{f}|\mathbf{h},\mathbf{A})$ is the Laplace approximation for the posterior $p(\mathbf{f}|\mathbf{y},\mathbf{X},\Theta^t)$. Using Equation(2.15) in Equation 2.16 we get:

$$\mathcal{Q}_e = \sum_{i=1}^{n}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\log\left(1+\frac{(y_i-f_i)^2}{\nu\sigma^2}\right)\right] \tag{2.17}$$

$$\begin{aligned}\mathcal{Q}_e &= C(\nu,\sigma) - \frac{\nu+1}{2}\sum_{i=1}^{n}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log\left(1+\frac{(y_i-f_i)^2}{\nu\sigma^2}\right)\right] \\ &\text{where } C(\nu,\sigma) = n\log\left(\Gamma(\frac{\nu+1}{2})/\Gamma(\frac{\nu}{2})/(\sqrt{\pi\nu}\sigma)\right)\end{aligned} \tag{2.18}$$

The expectation terms in Equation(2.18) cannot be derived in closed form. Nevertheless, using Jensen's inequality on the expectation terms, we can write

$$\mathcal{Q}_e \geq \mathcal{Q}_e^l := C(\nu,\sigma) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[1+\frac{(y_i-f_i)^2}{\nu\sigma^2}\right] \tag{2.19}$$

Closed-form expression for $\mathcal{Q}_e^l$ is available. Thus, instead of maximizing $\mathcal{Q}_e$, we propose to maximize its lower bound $\mathcal{Q}_e^l$ to update the value of $\theta_e = [\nu,\sigma]$. The complete expression for the $\mathcal{Q}_e^l$ and its gradient is given in Appendices.

The results from this technique are comparable to the existing parameter estimation methods (see Section 2.5). Also, we observed no convergence problems with this implementation. One reason could be that the point of maximum for $\mathcal{Q}_e(\theta_e|\Theta^t)$ and $\mathcal{Q}_e^l(\theta_e|\Theta^t)$ are very close to each other for most data sets. This would ensure that EM algorithm convergence is not affected. Moreover, the fact that partial maximization in M step has also been shown to converge in practice [40, 41] can be used to justify this approach.

2.4.2 EM steps for Laplace likelihood

In the case of Laplace likelihood, the expression for noise term is as follows:

$$p(y_i|f_i,s) = \frac{1}{2s}exp(-\frac{|y_i-f_i|}{s}) \tag{2.20}$$

Note that there is only one noise hyper-parameter $\theta_e = s$. Laplace approximation cannot be used with this distribution since it has discontinuous derivatives [27]. Therefore, EP method was used to approximate the posterior distribution $p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta^t) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})$. In order to use EM algorithm we need to evaluate $\mathcal{Q}_{cov}$ and $\mathcal{Q}_e$ in Equation 2.14. The expression for $\mathcal{Q}_{cov}$ can be easily obtained in closed form [33]. Closed-form expression for $\mathcal{Q}_e$ requires calculating the following expectation previously shown in Equation 2.13:

$$\begin{aligned} \mathcal{Q}_e(\theta_e|\Theta^t) &= \int log\left(p(\mathbf{y}|\mathbf{f}, \theta_e)\right) \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})\, \mathrm{d}\mathbf{f} \\ &= \sum_{i=1}^{n} \mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log\left(p(y_i|f_i, s)\right)\right] \end{aligned} \tag{2.21}$$

Expanding Equation 2.21 using Equation 2.20, we get:

$$\begin{aligned} \mathcal{Q}_e &= \sum_{i=1}^{n} \mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[-\log(2s) + \log\left(\exp\left(-\frac{|y_i - f_i|}{s}\right)\right)\right] \\ \mathcal{Q}_e &= -n\log(2s) - \frac{1}{s}\sum_{i=1}^{n} \mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}[|y_i - f_i|] \end{aligned} \tag{2.22}$$

It is possible to derive the exact expression for $\mathcal{Q}_e$ in Equation 2.22. Individual expectation terms can be expanded as follows:

$$\begin{aligned} \mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[|y_i - f_i|\right] &= \int |y_i - f_i| \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})\, \mathrm{d}\mathbf{f} \\ &= \int |y_i - f_i| \mathcal{N}(f_i|h_i, A_{ii})\, \mathrm{d}f_i \\ &= (y_i - f_i)\left[2\Phi\left(\frac{y_i - h_i}{A_{ii}}\right) - 1\right] + 2A_{ii}\left[\phi\left(\frac{y_i - h_i}{A_{ii}}\right)\right] \end{aligned} \tag{2.23}$$

Thus, the integral in Equation 2.23 can be expressed in terms of standard normal cdf (Φ) and pdf (ϕ). Using the above expression in Equation 2.22 we can get the final exact expression for $\mathcal{Q}_e$. The equation for $\mathcal{Q}_e$ and its gradient is given in Appendices.

2.4.3 Expectation Conjugate Gradient (ECG) Algorithm

An EM based algorithm known as Expectation Conjugate Gradient [38] was also implemented in this work. The derivative of Q function is same as the derivative of the log marginal likelihood. This property holds true when we use exact EM algorithm to find the Q function. In ECG algorithm the derivative of the Q function

is used to supply gradient to the conjugate gradient optimizer for the log marginal likelihood. This work uses a modified version of ECG algorithm. The gradient of the Q function derived originally for EM algorithm was used in place of the gradient of the approximate log marginal likelihood. The resulting approach can be described as follows:

ECG Algorithm Use a conjugate gradient optimizer to maximize the approximate log marginal likelihood $L(\Theta) \approx \log p(\mathbf{y}|\mathbf{X}, \Theta)$, employing the following steps whenever the value or gradient of $L(\Theta^t)$ at a particular Θ^t is required (eg. during line search)

1. E-step: Find an approximation $q(\mathbf{f})$ for the true posterior distribution $p(\mathbf{f}|\mathbf{y}, \mathbf{X}, \Theta^t)$. Use it to find the approximate log marginal likelihood $L(\Theta^t)$

2. G-step: Find gradient of Q function and use it in place of the gradient of the approximate log marginal likelihood

$$\nabla_L(\Theta^t) \approx \frac{\partial}{\partial \Theta} \mathbf{Q}(\Theta|\Theta^t)\Big|_{\Theta^t} \tag{2.24}$$

■

We implemented this algorithm for the Laplace noise distribution problem using EP for approximate inference. The derivative expression for $\mathbf{Q}(\Theta|\Theta^t)$ was simplified using matrix algebra properties and can be found in Appendices. Based on simulation results given in the next section, we found that ECG converges much faster than EM in most cases.

2.5 Regression results

The equation for making predictions using robust GP regression models can be found in several works [26, 31]. Before discussing regression results we give a brief description of the prediction equations. Given a new test point x_*, GP can be used to predict mean of $f(x_*)$ as follows:

$$\begin{aligned} \mathbf{E}_{p(\mathbf{f}|\mathbf{y},\mathbf{X})}\left[f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*\right] &\approx \mathbf{E}_{q(\mathbf{f})}\left[f^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*\right] \\ &= K(\mathbf{x}^*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}\mathbf{h} \end{aligned} \tag{2.25}$$

Since the posterior distribution $p(\mathbf{f}|\mathbf{y},\mathbf{X})$ is not analytically tractable it is approximated with $q(\mathbf{f})$ using the methods described in Section 2.3. The variance of $f(x_*)$ is also found using the approximation:

$$\begin{aligned}\mathbf{Var}_{p(\mathbf{f}|\mathbf{y},\mathbf{X})}\left[f^*|\mathbf{X},\mathbf{y},\mathbf{x}^*\right] &\approx \mathbf{Var}_{q(\mathbf{f})}\left[f^*|\mathbf{X},\mathbf{y},\mathbf{x}^*\right] \\ &= K(\mathbf{x}^*,\mathbf{x}^*) - K(\mathbf{x}^*,\mathbf{X})\mathbf{A}^{-1}K(\mathbf{X},\mathbf{x}^*) \qquad (2.26)\end{aligned}$$

Thus, given training locations $\mathbf{X}$ and any new test point $\mathbf{x}^*$, a predictive distribution for f^* can be defined using the above equations. Note that GP not only predicts the value of the function, but also provides an uncertainty of prediction that varies with location of the test point location $\mathbf{x}_*$. In Figure 2.1 prediction results from a toy problem using GP regression can be seen. Test points which are far away from training set locations have a larger uncertainty in prediction. Proposed EM

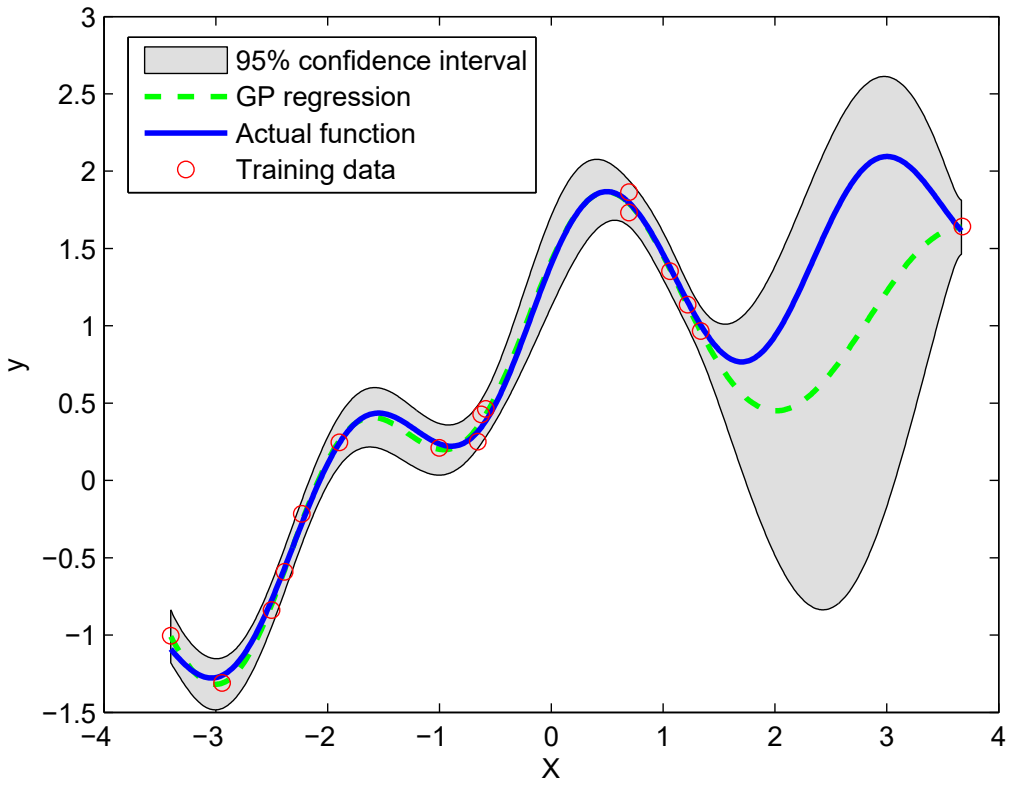


Figure 2.1: Example of prediction results from GP regression: Both mean and variance of prediction can be obtained

approach for finding hyper-parameters was compared with some conjugate-gradient (CG) based approaches using both synthetic and real-world data-sets. Table 2.1 contains the list of applied methods and their abbreviations. GPML toolbox [39] was used to implement snML, lnEP and tLAP. A Polack-Ribière flavor of conjugate gradient method has been implemented in this toolbox. The remaining proposed methods were implemented in MATLAB. Approximate inference using EP method was obtained from the GPML toolbox. For Laplace approximation method, the numerical stable approach as proposed by [31] was implemented. Average root mean square error of prediction was used to compare the methods.

Table 2.1: Robust GP regression methods compared in this work

Method	Noise distribution	Algorithm	Approximation
snML	Gaussian	CG	-
tLAP	Student's t	CG	Laplace
tEMLAP	Student's t	EM (proposed)	Laplace
lnEP	Laplace	CG	EP
lnEMEP	Laplace	EM (proposed)	EP
lnECG	Laplace	ECG (proposed hybrid)	EP

2.5.1 Synthetic data sets

Neal data-set

Neal created a synthetic regression problem with one input variable x where the true function value $f(x)$ was given by [42]:

$$f(x) = 0.3 + 0.4x + 0.5sin(2.7x) + \frac{1.1}{1+x^2} \tag{2.27}$$

10 training data-sets were created each containing 100 data points drawn from a standard normal distribution. Function values at 85% of points were corrupted using Gaussian noise with standard deviation 0.1. Remaining 15% of points served as outliers with function values corrupted using Gaussian noise with standard deviation 1. A test set of 1000 points was generated uniformly in the range $x \in [-3, 3]$. Equation 2.28 gives the expression for the co-variance function used in this problem.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_{se}^2 exp\left(-\frac{1}{2}\frac{(\mathbf{x_i} - \mathbf{x_j})^2}{l}\right) + v_0\delta_{ij} \tag{2.28}$$

The first term reflects the strength of correlation based on the distance between input locations. δ_{ij} is the Kronecker operator. The second term represents white noise which captures the random error effects. This term also helps in ensuring that the matrix inversions involved in the EM steps are stable. Thus the total covariance parameter vector is $\theta_{cov} = [l, \sigma^2_{se}, v_0]$.

Methods introduced in Table 2.1 are applied for parameter estimation of this problem. Box and whisker plots for average root mean square error for each method is given in Figure 2.2. Table 2.2 presents the root mean square error (RMSE) of the prediction error for each method.

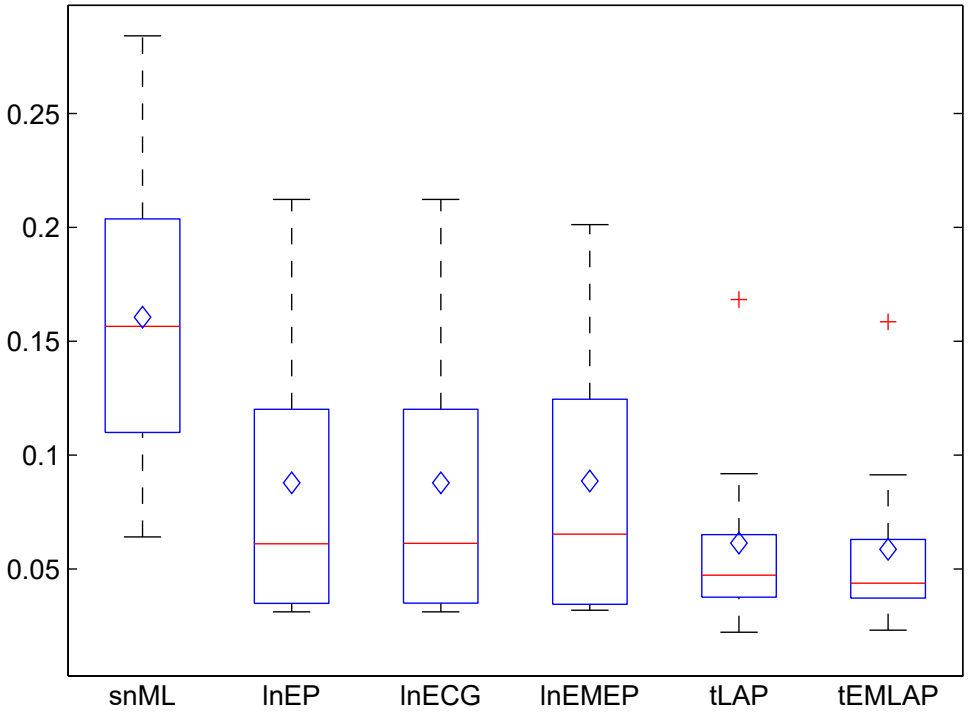


Figure 2.2: Neal data set: 15% outliers with standard deviation of 1

We see that proposed tEMLAP approach did marginally better than CG approach (tLAP) in the case of Student's t noise model. lnEMEP was slightly worse than lnEP. Overall we can see that the Student's t-distribution based models performed the best with this regression problem.

Table 2.2: Summary of RMSE results

Method	Neal	Friedman	Boston	Meat NIR
snML	0.161	0.287	2.606	0.910
tLAP	0.061	**0.194**	3.422	2.348
tEMLAP	**0.059**	0.207	**2.491**	**0.864**
lnEP	0.088	0.204	2.545	1.105
lnEMEP	0.089	0.207	2.665	0.907
lnECG	0.088	0.203	2.945	0.955
PLS	-	-	-	3.095

Friedman data-set

Friedman constructed the following regression problem which accepts a 10-dimensional input vector $\mathbf{x}$ and yet the function value $f(\mathbf{x})$ depends only on the first five input dimensions [43].

$$f(\mathbf{x}) = 10sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{2.29}$$

The purpose of the remaining input dimensions $x_6, ..., x_{10}$ is to complicate the task. 10 sets of training data were created and the identified models were used against a test set containing no outliers [27]. The square exponential covariance function with separate length scale for each feature was used for this data set.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_{se}^2 exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{\left(x_i^d - x_j^d\right)^2}{l_d}\right) + v_0\delta_{ij} \tag{2.30}$$

Here, $\theta_{cov} = [l_1, l_2, ..., l_d, \sigma_{se}^2, v_0]$. If a feature is uninformative, the corresponding length scale should automatically converge to a large value when optimized. To illustrate the automatic relevance determination ability of the chosen kernel, we present the optimized value of the hyper-parameters using the lnEMEP approach for one of the training sets:

$$\log\theta_{cov} = [-0.42, -0.58, -0.42, 0.04, 0.85, 15.98, 16.48, 16.39, 16.14, 16.42, 0.54, -1.86] \tag{2.31}$$

It can be observed that variables x_6 to x_{10} have very large values for optimized length scale hyper-parameters. This shows that they are uninformative.

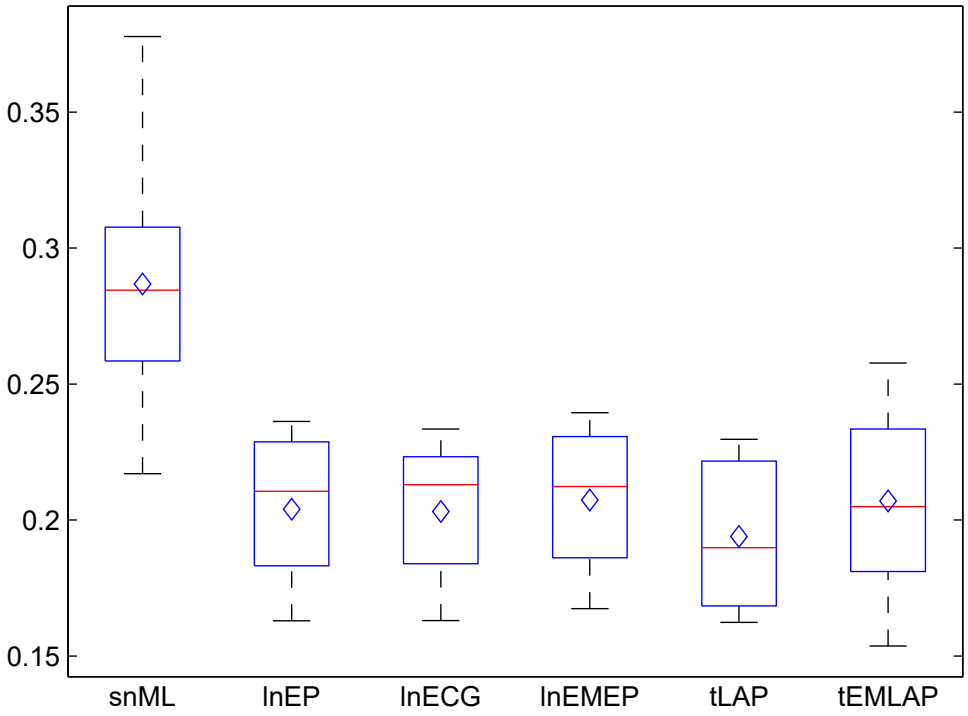


Figure 2.3: Friedman data set: 10% outliers with standard deviation 3

A box and whisker plot for root mean square error for each method using ten different training sets is given in Figure 2.3. From Table 2.2 we can see that once again Student's t-distribution based models performed better than others. lnECG, which uses a gradient of the Q function to maximize the approximate marginal likelihood did better than other EP based approaches.

2.5.2 Real data sets

Boston housing data set

This data set [44] is often used to test non-linear regression methods. 13 input variables are used to predict the median price of houses in Boston. There are 506 observations available in the data set. All variables are normalized to zero mean and unit variance before performing regression. Training and test data set partitions are made as shown in literature [27]. Once again, the squared exponential covariance function

with separate length scale for each feature was used for this task (see Equation 2.30).

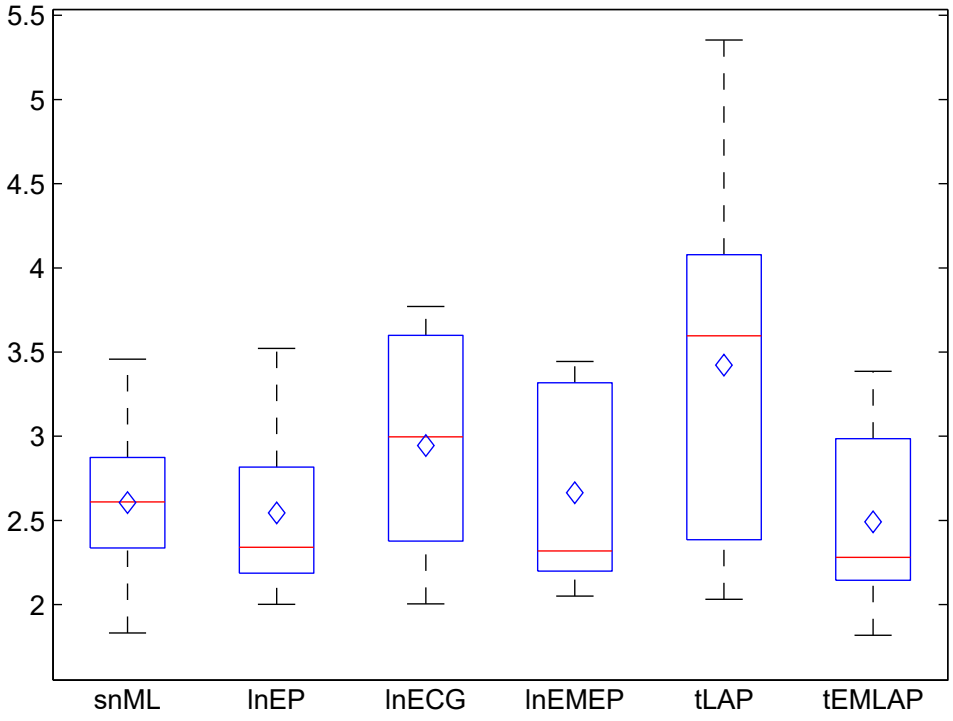


Figure 2.4: Boston housing data set

We can see from Table 2.2 that among the EP based approaches, lnEP performed better than other methods. However, tEMLAP gave the best prediction performance for this data-set. tLAP results were the worst compared to all other methods. It was observed that tLAP approach in GPML toolbox faced some numerical stability issues with this data set. In our implementation for tEMLAP, a more numerically stable approach [31] was implemented.

Nonlinear NIR data set

Chen et al. [19] have successfully applied GP regression on NIR data sets. The performance of proposed robust GP regression was assessed on the "Meat" data set [45] that was used in their work. Fat content in meat is known to exhibit a non-linear relationship with NIR spectra and hence was chosen as the response variable for our

purpose. Data was pre-processed and partitioned according to literature [19]. The following co-variance function was used by the authors and has been replicated in our approach giving $\theta_{cov} = [m_0, m_1, l_1, l_2, ..., l_d, \sigma_{se}^2, v_0]$

$$K(\mathbf{x}_i, \mathbf{x}_j) = m_0 + m_1 \sum_{d=1}^{D} x_i^d x_j^d + \sigma_{se}^2 exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{\left(x_i^d - x_j^d\right)^2}{l_d}\right) + v_0\delta_{ij} \tag{2.32}$$

m_0 is a constant bias in the covariance function. $m_1 \sum_{d=1}^{D} x_i^d x_j^d$ is used to capture linear correlation between input and response variables. 10 random partitions of training and test set pairs were generated from the data set and used for regression. The corresponding results are shown in Figure 2.5. Once again tEMLAP performed

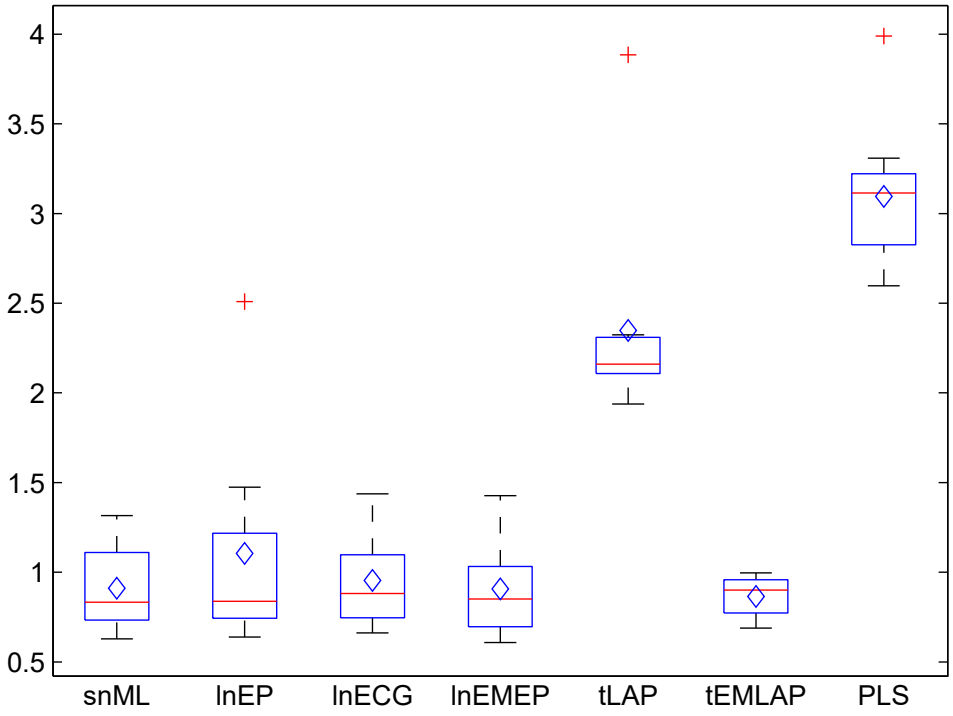


Figure 2.5: Meat NIR data set

the best among all methods, followed by lnEMEP. Other methods except tLAP and PLS, are marginally weaker than the proposed EM approaches. PLS fails to perform well due to its inability to model non-linearities whereas tLAP was affected with numerical stability issues just as in the case of Boston data set.

2.6 EM and conjugate gradient methods: A comparison

Results show that approximate EM technique is similar to direct maximization of approximate marginal likelihood in terms of prediction performance. Also, it was observed that both direct CG and ECG are faster than EM by a factor ranging from 2 to 10 depending on the size of the data set. This was expected since gradient based methods are generally faster than EM [38].

Another observation was that given the same initial guess, hyper-parameters learned using EM and CG approaches occasionally converged to widely different values. To understand this, it is important to observe that both EM and gradient based methods are optimization algorithms which do not guarantee global convergence. Convergence of such optimization schemes is dependent on iteration step size or step direction. It was also observed that even in usual cases, EM and CG solutions are close but not identical. This is because of the slight difference in gradient expression for the two algorithms. The effect of step size, step direction and gradient expression on the convergence of EM and CG methods is discussed in the next two subsections.

2.6.1 Step direction and step size

In most cases, the two algorithms gave similar estimates for the hyper-parameters. Thus, a toy problem was created wherein the two approaches converged to extremely different solutions given the same initial guess. Revisiting the Neal data set in 2.5.1, a new set of training points was created using the function in Equation 2.27. 100 data points were generated uniformly in the range $x \in [0, 30]$. Gaussian noise with standard deviation 0.1 was added to the function value at these locations. To generate outliers, 10 locations in X were chosen randomly and the function values at these locations were corrupted with Gaussian noise with standard deviation 1. Test set of 3000 points was generated uniformly in the range $x \in [0, 30]$. No outliers were included in the test set. The covariance function chosen for this example was:

$$k(x, x') = exp(-\frac{||x - x'||^2}{2l}) \tag{2.33}$$

For the same initial guess $[-1.5, 0]$, the EM based approaches converged to a different (in this case, better) maximum compared to the direct conjugate gradient based approaches as seen in Figures 2.6b and 2.6a. This suggests that results from the two algorithms are not always similar. The different outcomes can be attributed to different step size and step direction. It has been shown that for the Gaussian mixtures problem, "at each iteration of the EM algorithm, the EM step can be obtained by pre-multiplying the gradient by a transformation matrix $P(\Theta^t)$" [46]. This expression holds true for many other problems as well [38].

$$\Theta^{t+1} - \Theta^t = P(\Theta^t)\nabla_L(\Theta^t) \tag{2.34}$$

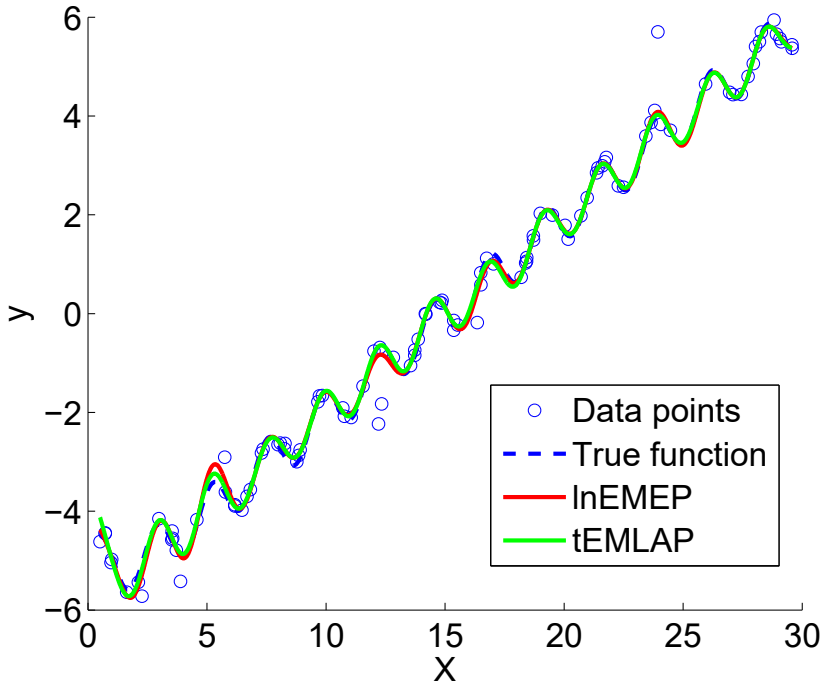
$P(\Theta^t)$ is positive definite under certain conditions [38]. In Newton based methods, instead of $P(\Theta^t)$ the inverse of the Hessian of log likelihood is used. When the Hessian is close to singular, Newton's method can diverge. For this reason, the inverse of the Hessian is often approximated (quasi-Newton methods). Another way to avoid inverse Hessian calculation is to use non-linear conjugate gradient (CG) method against which we have compared the EM implementation. However, all these methods require line-search techniques to find the best step size for iteration. A careful selection of tuning parameters is required to ensure that the optimization scheme does not diverge. Even in our simulations, it was observed that sometimes the conjugate gradient approach failed to converge. On the other hand, EM algorithm did not face any such problems owing to the positive definite nature of the transformation matrix. Since the Q function is well-behaved, maximizing it with respect to hyper-parameters is not as difficult.

2.6.2 Gradient with respect to hyper-parameters

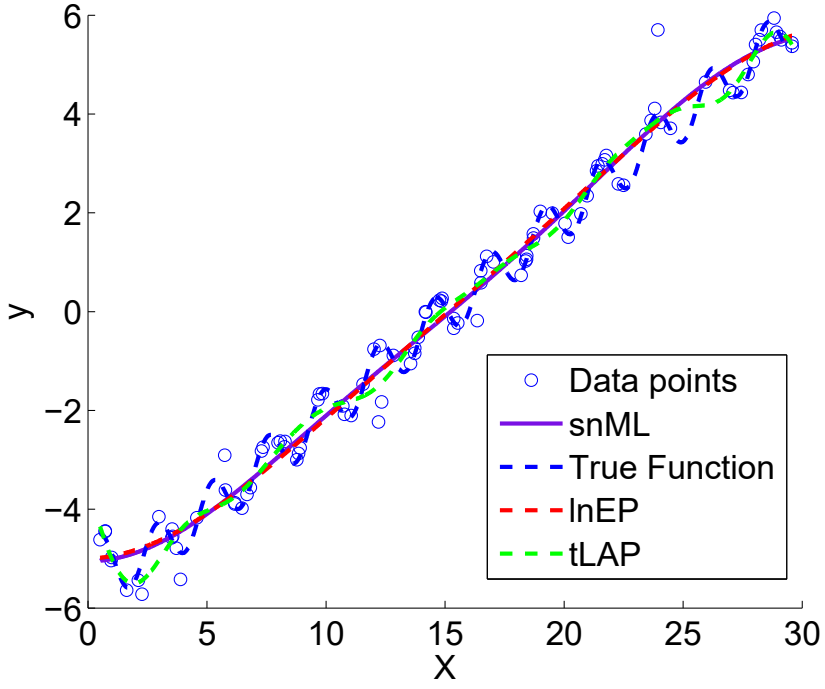
Another difference between the two methods is due to the gradient with respect to the hyper-parameters. Ideally the Q function derivative equals derivative of log likelihood at a given set of hyper-parameters, that is:

$$\nabla L(\Theta)\Big|_{\Theta^t} = \frac{\partial}{\partial \Theta}\mathbf{Q}(\Theta|\Theta^t)\Big|_{\Theta^t} \tag{2.35}$$

where $L(\Theta) = \log p(\mathbf{y}|\mathbf{X}, \Theta)$ is the log marginal likelihood. This fact was used to justify the ECG algorithm [38]. However, for robust GP regression, the Q function



(a) Proposed EM method



(b) Direct maximization of approximate marginal likelihood

Figure 2.6: Different optima reached by direct and EM methods

gradient used in the M step does not match the gradient of the log marginal likelihood exactly. This is because there are approximations on both sides of Equation 2.35. On the left side the log marginal likelihood is approximated and on the right side, in the Q function the expectation is taken over the approximate posterior distribution. Despite this, the two derivatives were observed to be fairly close to each other in practice. [33] showed that for EM algorithm using EP for approximating posterior distribution, the gradient of the approximate Q function with respect to the covariance function hyper-parameters θ_{cov} is equal to gradient of the approximate marginal likelihood using EP. However, the gradient with respect to noise hyper-parameters is not the same for the two functions under EP approximation. To verify this for robust GP regression, derivatives of the Q function of the lnECG implementation were compared with that for the approximate log likelihood function. It was found that the two derivatives are exactly equal except for the noise hyper-parameter derivatives which differ slightly. For example in the simulation used to generate Figure 2.6, where two parameters had to be learned; kernel length scale and noise parameter, at the initial guess [-1.5,0], derivative of log likelihood with respect to hyperparameters was [-82.9920,40.8471] and derivative of Q function with respect to hyperparameters was [-82.9920,39.4111].

The slight difference can also be seen in Figure 2.7 where ECG and CG methods are close to each other but not exactly identical. EM takes a different route to optimization due to the transformation matrix $P(\Theta^t)$ as explained in previous subsection. Since the derivative of the Q function is not exactly equal to the derivative of marginal likelihood for the robust problem, the ECG algorithm stopped before converging completely in our simulations. Despite this, in terms of RMSE, ECG results were comparable with EM and direct methods. This is because, typically in optimization, the number of iterations increase close to the optimal point and so the last few iterations do not significantly affect the result.

2.6.3 Advantages of EM over direct methods

Based on the above discussion one can conclude that the main advantage of EM algorithm is that it does not require any tuning parameter and is numerically stable. Also, it is easier to implement. Another potential advantage of EM is that the Q

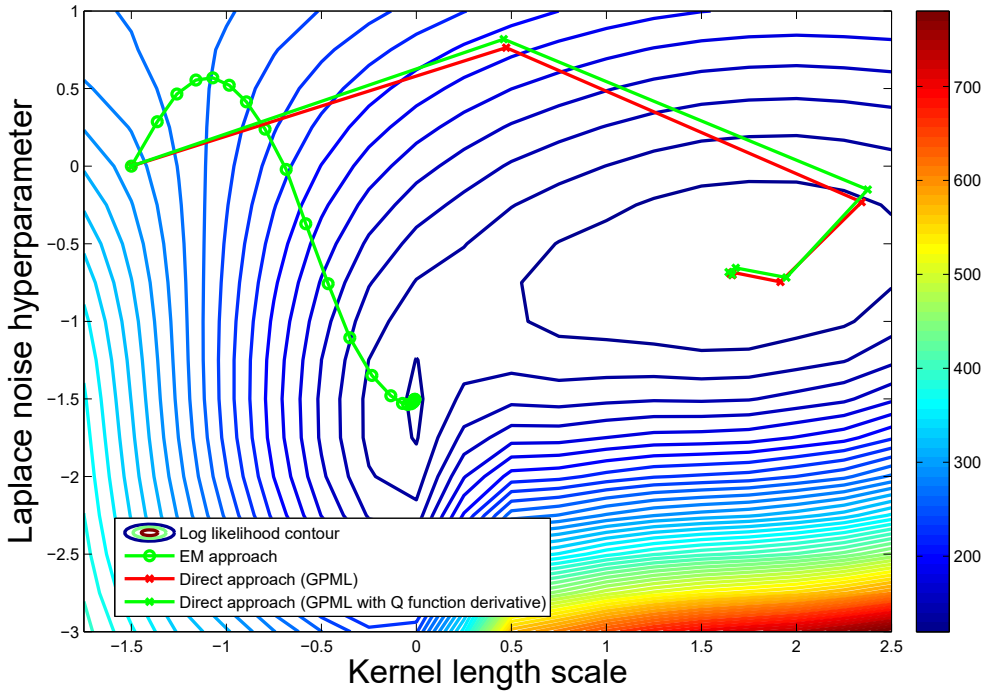


Figure 2.7: Comparing EM, ECG and Direct CG approach: ECG is close to the direct approach but not identical due to difference in gradients. EM takes a different route to optimization

function can supply the gradient when it is difficult to estimate the gradient of the log marginal likelihood [27]. The proposed scheme for robust GP regression could also be extended to handle sparse GP formulations, in the context of which lower bound maximization using EM algorithm was first proposed [32]. The main drawback of EM approach is that it can be slow, in which case one could switch to the ECG method discussed above.

2.7 Industrial application

Steam Assisted Gravity Drainage (SAGD) is used to recover heavy crude oil from Alberta's vast underground oil sands reserves. This technology requires large amounts of high pressure steam which is injected into deep underground oil sands reservoirs. In order to generate steam, a network of water treatment units is used to treat produced water and supply boiler feed quality water to steam generators. Control and optimization of this network is important for oil sands industries. As a part

of this objective, proposed robust GP regression using EM algorithm was used for constructing a data-based dynamic model of water treatment units in SAGD process.

A single water treatment unit consists of a series of tanks in which chemicals are added to remove hardness, silica and dissolved solids in the form of a sludge. Pure water and sludge exit the unit in the form of two separate streams. Flow rate data is available for all streams entering and exiting the unit. The input to the process is impure water flowrate $u(t)$ and the output is combined flowrate of pure water and slurry $y(t)$. Due to mass balance principle, at steady state $y(t)$ equals $u(t)$. A non-linear dynamic model for this process was required as part of a larger plant-wide optimization objective. The following structure was chosen for the model:

$$y(t+1) = f(y(t), u(t), y(t-1), u(t-1)) + \epsilon \tag{2.36}$$

According to this equation, the one-step ahead output $y(t+1)$ is given by an unknown function of the input $u(t), u(t-1)$ and output $y(t), y(t-1)$ corrupted by noise ϵ.

Step response data set was used for training a robust Gaussian process regression model. Due to proprietary reasons, input and output data were normalized in this example. Average values of $u(t)$ and $y(t)$ before step change and after achieving new steady state were used as maximum and minimum values for normalization. Training data points were arranged as follows:

$$\mathbf{y} = \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ \vdots \\ y(n) \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} y(1) & u(1) & y(0) & u(0) \\ y(2) & u(2) & y(1) & u(1) \\ y(3) & u(3) & y(2) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ u(n-1) & u(n-2) & y(n-1) & y(n-2) \end{bmatrix} \tag{2.37}$$

Gaussian process model was chosen since it can be used to model non-linearities without knowledge of the underlying model structure. Moreover, to account for outliers in output, noise was assumed to follow Laplace distribution (Equation 2.20). Squared exponential covariance function with automatic relevance determination (Equation 2.30) was used for designing the Gaussian prior.

Hyper-parameters for the covariance function were learned using the proposed lnEMEP approach. Figure 2.8 shows the training data plots and the one-step ahead prediction from identified robust GP regression model. We can see that the predic-

tions match well with output with an RMSE of 0.1269.

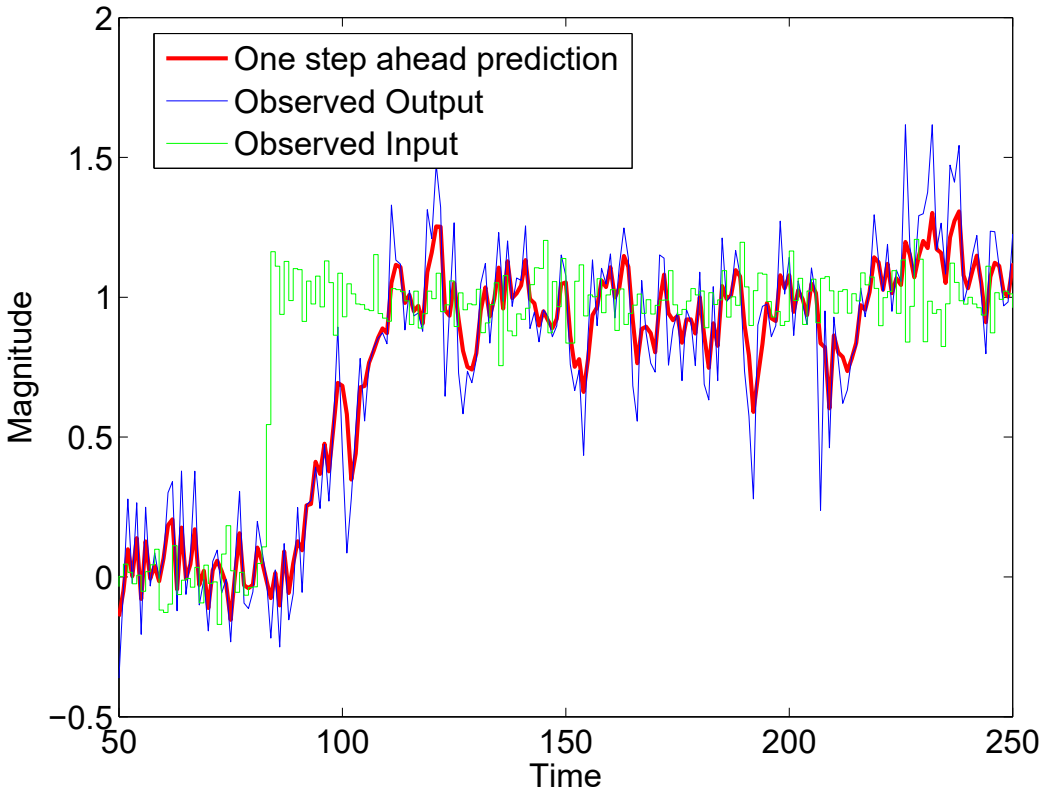


Figure 2.8: One step ahead response from robust GP regression model

2.8 Conclusion

In this chapter, we proposed an approximate EM algorithm for constructing a robust GP regression model. EM steps were derived for two noise models, Student's t-distribution and Laplace distribution. A new lower bound in EM algorithm was proposed for the Student's t-distribution. The proposed approach was validated using both synthetic and industrial data sets. Furthermore, we compared the proposed method against conjugate gradient maximization of approximate marginal likelihood. The two approaches are similar in terms of prediction performance. However, in some cases, EM approach outperforms the direct approach. The effects of step size, step direction and gradient on the two methods were analyzed. The advantage of the EM approach lies in its ease of implementation, stability and theoretical convergence

guarantees. Finally, an industrial application of robust GP regression was explored in which it was used to identify a nonlinear dynamic model for a water treatment unit in SAGD process. The identified process model was used as a constraint as part of a larger optimization framework which is discussed in Chapter 4.

Chapter 3

Steady-state modeling and optimization of water treatment network

This chapter focuses on first-principles steady-state modeling and optimization of water treatment network in SAGD process. Section 3.1 introduces the problem and specifies the objective. Description of process units and available data is given in Section 3.2, while data preprocessing steps are explained in Section 3.3. Section 3.4 contains methodology adopted for data reconciliation and corresponding results. Finally, in Section 3.5 steady state optimization scheme is discussed and results are compared against historical data.

3.1 Problem statement

Steam is an important utility in SAGD process. Water is treated in large treatment units and fed to steam generators. Scheduling and planning of steam production is essential. For this reason, industries maintain an overall process capacity in excess of the actual requirement and use buffer tanks in the network to regulate water flow. In the event of planned or unplanned shutdowns, operators must divert water flow to other units and make efficient use of limited tank volumes to minimize any drop in processing capacity during the transition. Operators would like to have a tool which they can use to assess or improve their decision making during such situations.

3.1.1 Objective

The objective is to find an effective tool which can provide a quick solution and guide the operators in their decision making. Mathematically, this involves a two-fold objective. The first objective is to find what should be the optimal throughput for all units, when some of the units have planned or unplanned shutdowns. In this study, optimality is defined based on minimization of the cost of steam production. In continuation, the second objective is to arrive at this optimal operating point, as soon as possible using buffer tank capacities and honoring all process limitations.

3.2 Process description

In this section we give a brief overview of SAGD process. Water treatment and steam generator units are described in some detail. A description of industrial data is also provided.

3.2.1 SAGD Process

Steam Assisted Gravity Drainage or SAGD technology is used to recover heavy crude oil from Alberta's vast underground oil sands reserves. Figure 3.1 shows the flow chart for the complete process. Large volumes of high pressure steam are injected into deep horizontal pipes which are buried in underground oil sands reservoir. Heat from steam helps reduce the viscosity of bitumen (or heavy oil) in the reservoir. A mixture of bitumen and water is then pumped to the surface from production wells. Bitumen is separated from water, diluted using a diluent and sent downstream for further processing while produced water is treated and reused for steam generation. Some makeup water is also mixed with produced water to meet the required steam demand. Two different produced water treatment technologies namely, Warm Lime Softeners and Evaporators are used in industry. Once treated, water is sent to steam generators. Here too, there are two different types of technologies: OTSGs (Once Through Steam Generators) and Cogens (Co-generators). Not all water is converted to steam in these generators. Some of it is recycled back to the upstream units and is known as "blowdown". Tanks are provided upstream of water treatment plants and steam generators. These act as buffers and are used by operators to absorb changes

in operating conditions.

This work focuses on modeling and optimization of the water treatment and steam generator network encircled in Figure 3.1. The figure shows a simplified process flow diagram. In reality there may be several water treatment and steam generator units operating in parallel with interconnections to allow for water distribution in the network. An example is depicted in Figure 3.2. Note that this figure is not identical to the actual process network which has not been shown due to proprietary reasons. The ensuing subsections contain description for various units in the network.

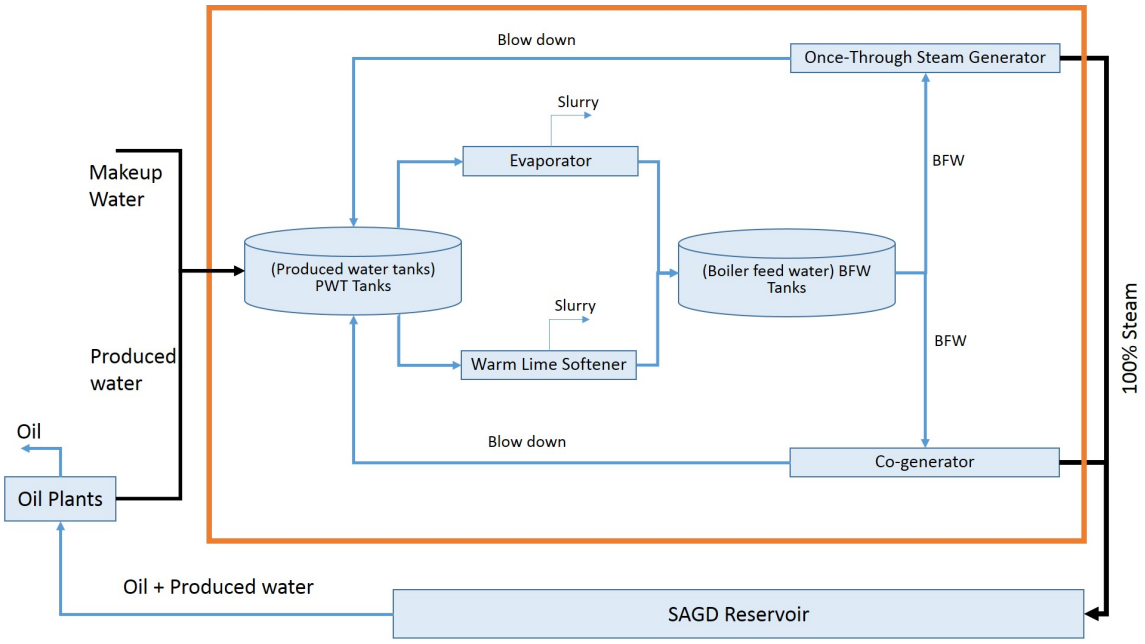


Figure 3.1: General SAGD process overview. Region enclosed within the orange box was modeled and optimized in this work

3.2.2 Water treatment units

De-oiled water (produced water separated from bitumen) contains traces of dissolved impurities which can cause scaling in steam generators. Different methods can be used for reducing hardness and silica in produced water which constitute the bulk of impurities. Addition of certain chemicals can neutralize the impurities and convert them into insoluble sludge which can be removed using settling tanks and filters. This is accomplished in the Warm Lime Softener units. Evaporators offer an alternative approach to water treatment wherein impurities are separated by converting the water

content in the mixture into water vapor. Both these technologies have been used in industry and are described below:

Warm Lime Softeners

Warm Lime Softener units as a general rule consist of the following three sections:

- Warm Lime Softener (WLS) Tank
- After Filters
- Weak Acid Cation (WAC) Tank

In the WLS tank, chemicals are added to remove calcium and magnesium salts and silica. These impurities settle at the bottom of the tank and are removed in the form of sludge. Treated water is sent to After Filters which remove suspended solids using a filter media. To avoid pressure drop in the line, filters must be cleared of accumulated solids using a backwash procedure. Finally, filtered water is sent to WAC Tanks where the majority of hardness is removed using weak acid cation resins. These resins must also be regenerated from time to time.

This process cannot handle water with very high amounts of total dissolved solids (TDS>7000 ppm) and requires experienced operators [47]. However, they are cheaper to operate since they require less energy compared to evaporators.

Evaporator

This process offers an alternative evaporative method of treating produced water. Different types of evaporators are used in industry. The key idea in this method is to allow the evaporation of produced water to separate impurities and obtain boiler feed quality water output. The vertical tube falling film vapor compression evaporator offers an energy efficient system for water evaporation and is used in many SAGD plants. Yet, despite technological improvements, this process is more energy intensive than addition of chemicals as done in WAC or WLS. Nevertheless, the advantage of these units is that they can handle water with higher amounts of total dissolve solids. Details regarding this technology can be found in several articles [47, 48].

3.2.3 Steam generators

After treatment, water is sent to steam generators where it is converted to steam. There are different types of steam generators which are used in industry. Two of the popular designs are heat recovery and once through steam generators. They are described below:

Co-generator

Cogenerator or heat recovery steam generator (HRSG) is an energy recovery steam generator that uses excess heat from hot gas stream to generate electricity. A typical co-generator consists of three sections namely the economizer, evaporator and superheater. Pressurized water is pre-heated in economizer section by passing it through hot tubes. It is then further heated in the feed water drum. Boiling water rises into the steam drum where it is separated from steam. Saturated steam is drawn off from the top of the steam drum and sent to the superheater section where further heat is supplied to increase steam temperature. Through efficient design waste heat from hot gas is utilized for electricity production.

Once Through Steam Generator

In OTSGs, unlike conventional drum boiler based HRSGs, there are no segmented sections for economizer, evaporator and superheater and waste heat is not used for electricity production. Pressurized water is fed into hot tubes at one end and superheated steam is produced at the other end. The advantage of OTSGs is that they can handle water with high levels of total dissolved solids (TDS) and silica. The heat flux in OTSG is also much lower than co-generator which makes it more tolerant to overheating. Startup and shutdown in OTSGs is also much faster.

Ideally, 100% steam quality (ratio of steam to total output) is desirable, i.e. steam uncontaminated with any liquid water. To achieve this, steam must be either 100% saturated or superheated. However, in practice water fed to steam generators may contain trace amounts of dissolved salts which can damage pipes at high steam quality conditions. Therefore, SAGD steam generators are operated at marginally lower steam quality range of 70% to 80%.

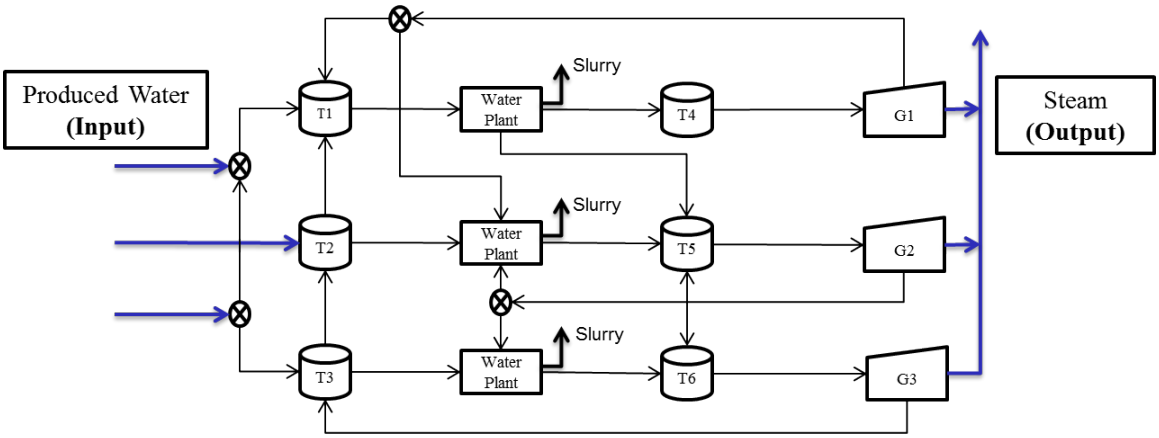


Figure 3.2: Water treatment and steam generator network. Actual network is not shown due to confidentiality reasons

3.2.4 Buffer tanks

Tanks are installed upstream of water treatment and steam generator units (see Figure 3.2). These act as buffers in the event of changes in operating point for a unit or shutdown/startup. Tanks upstream of water treatment units are known as produced water tanks (PWTs) and those upstream of steam generators are called boiler feed water tanks (BFW tanks). Each tank has its own lower and upper limit for tank level which must not be violated to ensure safe operation.

3.2.5 Process Data

Data over a 1 year period was provided by the industry. It consists of process measurements regarding water bearing streams in the network. The contents are summarized below in Table 3.1:

Table 3.1: Summary of data-set

Sr. No.	Data-type	No. of variables	Units	Interval
1	Water Flow rate	40	m^3/hr	1 min
2	Tank level	7	m	1 min
3	Steam quality (fraction)	18	-	1 min
4	Design processing capacities	27	m^3/hr	-
5	Tank upper & lower limits	14	m	-
6	Tank area	7	m^2	-

As shown in the table, time series measurements are available for water flow rates, tank levels and steam quality. Barring a few streams, all measured water flow streams including input produced water flow streams as shown in Figure 3.2 are available. Similarly all tank levels are measured and steam quality ratios are available for all generators. However actual steam flow rate is not measured and can only be estimated based on steam quality ratio and input boiler feed flow rate. Note that the Figure 3.2 is not the actual process flow sheet and thus the number of variables may not match those from the figure.

All flow rate measurements are at standard conditions, i.e. densities are not required to use this data for verifying mass balance. Steam quality is available individually for all steam generators. Design processing capacity refers to the maximum flow rate limit for the total feed to a unit as per process design.

3.3 Data preprocessing

Since our objective in this chapter is data reconciliation and steady state optimization, the available time series data was averaged over 30 minute intervals to remove the effect of process transients. This resulted in a shorter time series of measurements of 17520 data points. Original data set with 1 minute frequency will be used in Chapter 4 for dynamic process identification. The rest of this section covers other preprocessing steps.

3.3.1 Identifying shutdown process units

At a particular time, some units may be under shutdown or under reduced operating limits. Such situations must be identified before data reconciliation or optimization. Shut down condition for a plant at a particular time t was inferred from zero or negligible flow rate measurements for input streams to that plant at time t. Inferring reduced operating limits using just flow rate data was not possible and hence it has not been considered in this work.

3.3.2 Identifying missing flow measurements

Average steady state mass balance error for every unit in the network over all data points was used to verify the process flow sheet provided by industry. Although it is not entirely reasonable to assume steady state conditions for mass balance, it can be argued that any significant missing flow measurement would result in abnormally large average mass balance error (say more than 20%) in affected units. On the basis of this exercise a few significant missing streams were identified. Data for these streams was sought from the industry and the complete process flow sheet was built. The data-set summary in Table 3.1 lists the complete set of information available.

Variable notation

Before proceeding, it is important to summarize the variable notations that will be used in the rest of the chapter.

Table 3.2: Summary of variable notations for this chapter

Variable	Notation
All measurements (flow rates and tank levels) at time t	$\mathbf{x}(t)$
Steam quality fraction at time t	$\eta(t)$
Maximum processing capacities of units	$\mathbf{C}(t)$
Upper limit of all measured variables at time t	$\mathbf{h}(t)$
Lower limit of all measured variables at time t	$\mathbf{l}(t)$
Tank cross-sectional area of tank	A_{tank}
Vector of unmeasured variables at time t	$\mathbf{z}(t)$

Relevant subscripts will be used when referring to flow measurements (for example x_{wp_in}) or tank level measurements (x_{level}) within the vectors $\mathbf{x}(t), \mathbf{l}(t)$ and $\mathbf{h}(t)$. Elements of $\mathbf{C}(t)$ represent the maximum processing design capacity of units. If the flow rate across the unit is observed to be negligible, the corresponding element in $\mathbf{C}(t)$ vector is set to 0. This represents the shutdown situation. Maximum limits for individual flow rate measurements were empirically chosen based on the process design limits of units upstream or downstream of the measured line. Lower limit for all flow rate measurements was fixed to be 0 except for one stream which allowed flow in both directions. An empirical lower limit was set for this stream. Tank area cross sections

are provided in the data-set and are referred by A_{tank}. Data for some streams was not available. These are represented by the vector $\mathbf{z}(t)$. Their values were estimated based on the data reconciliation technique described in the next section.

3.4 Data reconciliation

Process measurements $\mathbf{x}(t)$ are usually noisy. They may contain random errors or gross errors. Data reconciliation refers to the estimation of process variables using process measurements and models [49]. In this work, it was used to get reconciled measurements $\hat{\mathbf{x}}(t)$ and -estimate unmeasured variables $\hat{\mathbf{z}}(t)$.

Simple yet reasonably accurate process models were used for data reconciliation. Steady state mass balance models were built to describe the water treatment units and steam generators. Tanks were modelled using a simple material balance model involving changes in tank level. Energy balance models could not be constructed since only material flow rate information is available. Reconciled measurements were made to honor the process models introduced in the following subsections.

3.4.1 Water treatment unit

Evaporators were modeled using a simple mass balance model.

$$\sum_{i \in wp\_in} \hat{x}_i(t) = \hat{x}_{wp\_slurry}(t) + \sum_{j \in wp\_out, j \notin wp\_slurry} \hat{x}_j(t) \tag{3.1}$$

$\hat{x}_{wp\_slurry}(t)$ is also an output of the water plant but the distinction has been made because it is sent to disposal, whereas other streams contain treated water which are fed to steam generators. For the evaporators, all process measurements are available. In the case of WLS units, slurry flow measurements are not available. For this reason, $\hat{x}_{wp\_slurry}(t)$ is replaced by $\hat{z}_{wp\_slurry}(t)$ in the above equation and approximated using data reconciliation. A similar change was made in other process models (steam generator, tanks etc.) to deal with missing measurements. Besides mass balance, the following inequality was also used:

$$\sum_{i \in wp\_in} \hat{x}_i(t) \leq C_{wp}(t) \tag{3.2}$$

All water plants have a maximum processing capacity C_{wp} which must be honoured by the total input flowrate.

3.4.2 Steam generator

In steam generators, feed water is converted into a saturated mixture of steam and water. Water is separated from steam and recycled. This stream is known as blow-down. Flow measurements are available for feed water and blowdown. Steam quality ratio is also available. These variables are related as follows:

$$\sum_{j \in sg\_bd} \hat{x}_j(t) = (1 - \eta_{sg}(t))\hat{x}_{sg\_in}(t) \tag{3.3}$$

where η_{sg} refers to the steam quality, x_j is the one of the blow down flow rates and x_{sg_in} is the input to the steam generator. While analysing process data, it was found that blow down flow rate for steam generators receiving water from WLS units, was always less than that given by steam quality ratio using above equation. This was perhaps due to bias in blow down flow rate measurement. The steam quality data which is also available was considered to be a more accurate predictor of blow down flow rate. Thus Equation 3.3 was used as the mass balance model for steam generator. Just as in the case of water treatment units, all steam generators have a maximum processing capacity and thus:

$$\hat{x}_{sg\_in}(t) \leq C_{sg}(t) \tag{3.4}$$

Here $\hat{x}_{sg\_in}$ is the feed water entering the steam generator and C_{sg} is the maximum rated process capacity of the steam generator.

3.4.3 Tanks

It was observed that tank dynamics are significant even after data preprocessing wherein measurements were time averaged over 30 minutes. Therefore, a material balance model involving tank level measurements was constructed for tanks and used in data reconciliation.

$$A_{tank}\hat{x}_{level}(t) + \sum_{i \in tank\_in} \hat{x}_i(t) - \sum_{j \in tank\_out} \hat{x}_j(t) = A_{tank}x_{level}(t+1) \tag{3.5}$$

$\hat{x}_i$ is an input stream, $\hat{x}_j$ is an output stream and $\hat{x}_{level}(t)$ is the tank level at time t. $x_{level}(t+1)$ is the tank level measurement at time $t+1$ and was deemed constant. This equation was used to reconcile all measurements at time t.

3.4.4 Data reconciliation framework

Based on above explanations, the reconciled data ($\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{z}}(t)$) are obtained by:

$$\underset{\hat{\mathbf{x}},\hat{\mathbf{z}}}{\operatorname{argmin}} \ (\mathbf{x}-\hat{\mathbf{x}})^T\mathbf{V}^{-1}(\mathbf{x}-\hat{\mathbf{x}}) \tag{3.6}$$

such that:

$$\mathbf{A}_{eq,x}\hat{\mathbf{x}} + \mathbf{A}_{eq,z}\hat{\mathbf{z}} = \mathbf{b_{eq}} \tag{3.7}$$

$$\mathbf{A}_x\hat{\mathbf{x}} + \mathbf{A}_z\hat{\mathbf{z}} \leq \mathbf{C} \tag{3.8}$$

$$\mathbf{l} \leq \hat{\mathbf{x}} \leq \mathbf{h} \tag{3.9}$$

where $\mathbf{x}$ is the vector of raw measurements, $\hat{\mathbf{x}}$ is the reconciled value of process measurements and $\hat{\mathbf{z}}$ contains the estimate for unmeasured variables. $\mathbf{V}$ is a diagonal covariance matrix representing noise in the measurements. Equation 3.7 represents all the process models based on mass balance equations described above (namely Equations 3.1, 3.3 and 3.5). Equation 3.8 represents the limits to processing capacities described in Equations 3.2 and 3.4. Equation 3.9 specifies the limits to the flow streams and tank levels as described previously in Table 3.2.

Use of covariance matrix $\mathbf{V}$ in the objective function ensures that variables with noisier measurements (higher variance) are adjusted more than less noisy ones. In order to estimate the variance of measurements the following approach was taken:

1. The complete available time series data was passed through a non-causal filter ("filtfilt" in MATLAB)

2. Variance of measurement was estimated using the deviation of measured data from filtered data

3. $\mathbf{V}$ was constructed as a diagonal matrix using the variance estimations

3.4.5 Results

The above described data reconciliation framework was used to obtain reconciled measurement values and estimate unmeasured variables. Results show that reconciled values match well with the raw measurements for all streams. This suggests that process models used to reconcile the process measurements are valid. Note that dynamics of steam generators and water treatment units were not considered. However, tank models were built based on material balance incorporating tank level measurements. As mentioned earlier, this was done because tank dynamics were found to be significant.

A scatter plot of reconciled v/s measured tank levels is given in Figure 3.3. Another scatter plot of reconciled v/s measured input produced water flow rates is shown in Figure 3.4. Barring a few data points, both figures show that data is well explained by the chosen process models.

One drawback of the proposed data reconciliation approach is that gross error in a measurement may not be detected since it is distributed over all other variables. It is possible that small adjustments in some of the variables can hide the gross error in another variable. The possibility of such a scenario can be reduced by a good choice of covariance matrix $\mathbf{V}$. Another solution could be to identify and remove the gross error prior to performing data reconciliation.

3.5 Steady state optimization

For a given fixed amount of produced water received from the reservoir (see input in Fig 3.2), operators would like to know the optimal distribution of water flow in the network. There are several ways in which the optimality condition can be defined. In this work, the objective was defined as the minimization of the average cost of production per unit m^3 of dry steam given a fixed amount of produced water entering the network. This is a reasonable objective since steam production is the most energy and cost intensive operation in the whole process. The cost of production per unit m^3 of steam for different steam generators in the network was available from industry.

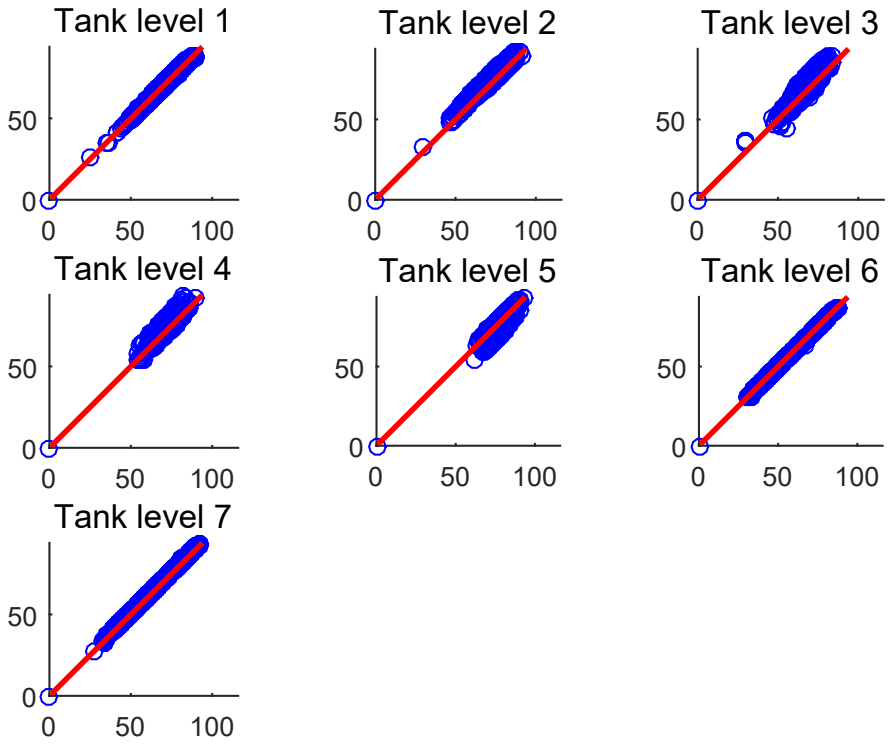


Figure 3.3: Scatter plot for tank levels (in %). X-axis is raw measured values and Y-axis is reconciled values

Thus the objective was defined as the following:

$$F = \frac{\sum_{i=1}^{n} s_i \mu_i \tilde{x}_{i,sg\_in}}{\sum_{i=1}^{n} \mu_i \tilde{x}_{i,sg\_in}} \tag{3.10}$$

Variables marked with the $\sim$ sign are decision variables in the optimization. In the above equation $\tilde{x}_{i,sg\_in}$ is the input feed water flow rate to i^{th} steam generator,μ_i is the steam quality and s_i is the cost of production which was assumed to be fixed value for each generator. Co-generators had a lower cost of production due to their capability of using waste heat for producing electricity. Once-through steam generators on the other hand had a higher cost of production. The minimization of the above mentioned objective function suggests that co-generators must always be preferred over OTSGs. However, it may not always be possible to divert maximum possible flow to co-generators owing to network limitations. Thus, mass balance and throughput limitations are included as constraints to the optimization. Unlike data reconciliation, dynamic tank models were not included in the list of constraints since we need to find

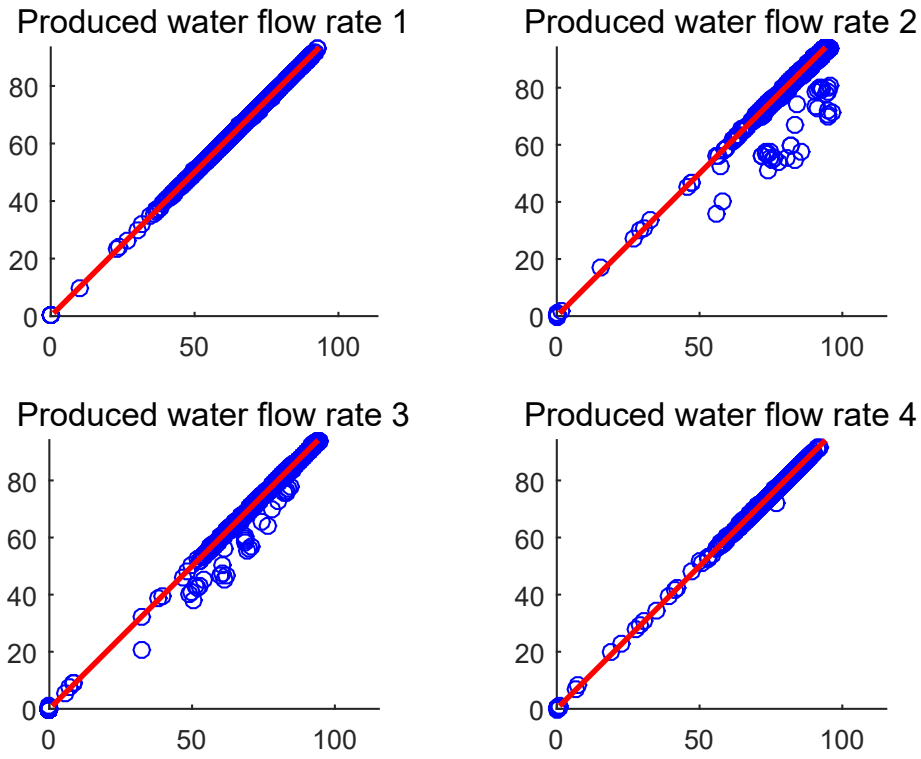


Figure 3.4: Scatter plot for produced water flow rates (normalized). X-axis is raw measured values and Y-axis is reconciled values

steady state optimal operating point. Instead, a simple steady state mass balance model was used. Moreover, it was observed that the amount of slurry water exiting the water treatment units is usually a fixed fraction of the input water flowrate. In other words:

$$\phi_{wp}\left(\sum_{i\in wp\_in}\tilde{x}_i(t)\right) = \tilde{x}_{wp\_slurry}(t) \tag{3.11}$$

Here ϕ_{wp} is the slurry ratio parameter which is learned by performing linear regression using reconciled flow rates $\hat{\mathbf{x}}$ in place of $\tilde{\mathbf{x}}$ in Equation 3.11. Use of reconciled measurements is necessary to ensure meaningful parameter estimation [50]. Data regions were identified where the reconciled flow rates appear to be in steady state. These data points were used to estimate the value of slurry ratio parameter which are listed in Table 3.3.

The value of this parameter was found to lie around 2% to 3% for all water plants. Equations 3.1 and 3.11 are together used to model the steady state model for water

Table 3.3: Slurry ratio for water plants

Water plant	Slurry ratio ϕ_{wp}
1	0.0262
2a	0.0247
2b	0.0282
2c	0.0261
2d	0.0209
3	0.0312

treatment units. The complete optimization problem can be expressed as:

$$\mathbf{x_f^*}, \mathbf{z_f^*} = \underset{\tilde{\mathbf{x}}_\mathbf{f}, \tilde{\mathbf{z}}_\mathbf{f}}{\operatorname{argmin}} \; F = \underset{\tilde{\mathbf{x}}_\mathbf{f}, \tilde{\mathbf{z}}_\mathbf{f}}{\operatorname{argmin}} \; \frac{\sum_{i=1}^{n} s_i \mu_i \tilde{x}_{i,sg\_in}}{\sum_{i=1}^{n} \mu_i \tilde{x}_{i,sg\_in}} \tag{3.12}$$

such that:

$$\mathbf{A}_{f,eq,x}\tilde{\mathbf{x}}_\mathbf{f} + \mathbf{A}_{f,eq,z}\tilde{\mathbf{z}}_\mathbf{f} = \mathbf{b_{f,eq}} \tag{3.13}$$

$$\mathbf{A}_{f,x}\tilde{\mathbf{x}}_\mathbf{f} + \mathbf{A}_{f,z}\tilde{\mathbf{z}}_\mathbf{f} \leq \mathbf{C_f} \tag{3.14}$$

$$\mathbf{l_f} \leq \tilde{\mathbf{x}}_\mathbf{f} \leq \mathbf{h_f} \tag{3.15}$$

$$\tilde{x}_{pw,in} = \hat{x}_{pw,in} \tag{3.16}$$

The objective F represents minimization of average cost of steam production. $\mathbf{x_f^*}$ and $\mathbf{z_f^*}$ are the optimized value of flow rates. Optimal tank levels are not found since the objective is steady state optimization. The subscript f is used to highlight this fact. Equation 3.13 represents the flow rate based mass balance equations previously described in Equations 3.1 and 3.3. It also includes the slurry ratio model shown in Equation 3.11 and steady state tank models expressed as follows:

$$\sum_{i \in tank\_in} \tilde{x}_i(t) - \sum_{j \in tank\_out} \tilde{x}_j(t) = 0 \tag{3.17}$$

Equations 3.14 and 3.15 are the same as those used in data reconciliation. The final constraint in Equation 3.16 signifies that the input produced water flow rates to the network are kept same as the reconciled value.

3.5.1 Optimization procedure summary

We summarize the procedure for optimization. First we select process measurements at time instant t and identify the process capacity limitations active at that time.

Given process measurements at time t, we perform data reconciliation and compute the cost of steam production using Equation 3.10 with reconciled flow rate values. Finally, we perform steady state optimization fixing the input produced water flow rate and compare the achieved optimal cost of production to historical cost of production. This procedure has also been summarized in the form of a flow chart in Figure 3.5.

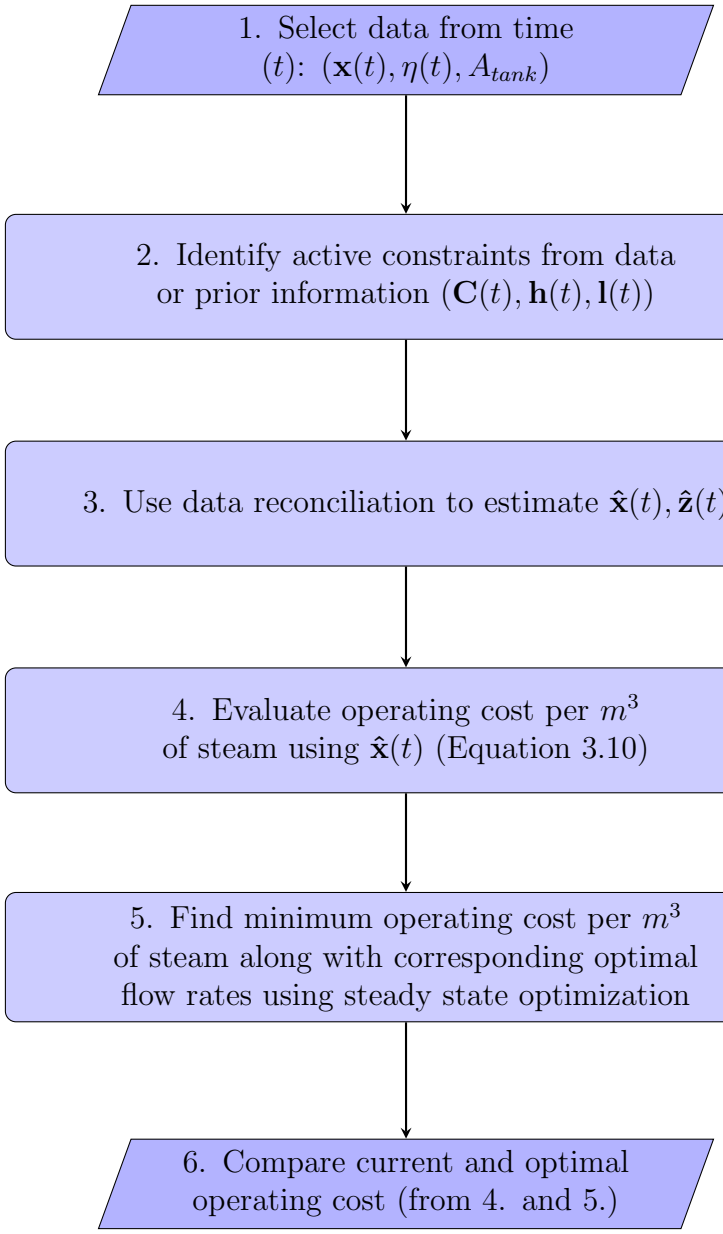


Figure 3.5: Summary of data reconciliation and optimization

3.5.2 Results

Historically achieved cost of production was compared against steady state optimization results for minimum optimal cost of production. Figure 3.6 depicts the comparison for some randomly selected data-points. The cost of production has been normalized for proprietary reasons. Historically achieved operating cost is in blue and minimum optimal cost is in green. In addition, the maximum cost of production is also plotted in red. This was obtained by maximizing the cost of production instead of minimizing in the above described framework. The red and green plots represent the worst and best possible operating strategies respectively. An operator would like to operate closer to the minimum operating limit. It can be observed that for most scenarios historical cost of production was closer to minimum limit compared to maximum limit. However there still appears to be a significant room for improvement.

Operators can use such results to assess their performance and find the set points $\mathbf{x}^*$ for achieving minimum operating cost. Besides steam production cost, other costs of operation such as water treatment costs can also be included in the objective function. Sometimes, a particular set of process units may be preferred for operation despite resulting in higher operating costs. Such priorities can be easily included in the optimization framework to arrive at a more practical solution.

3.5.3 Conclusion

In this chapter, we introduced the water treatment network optimization problem. First principle mass balance models were constructed based on process description and problem requirements. Raw measurements were reconciled using these models. It was observed that process models explained the data reasonably well. This allowed their use in steady state optimization. Results from optimization can be used by operators to assess how close they are operating to the optimal operating point. The next natural objective is to arrive at the steady state optimal operating point as quickly as possible by making use of buffer tank capacities and while adhering to all process limitations. An approach to handle this is discussed in the next chapter.

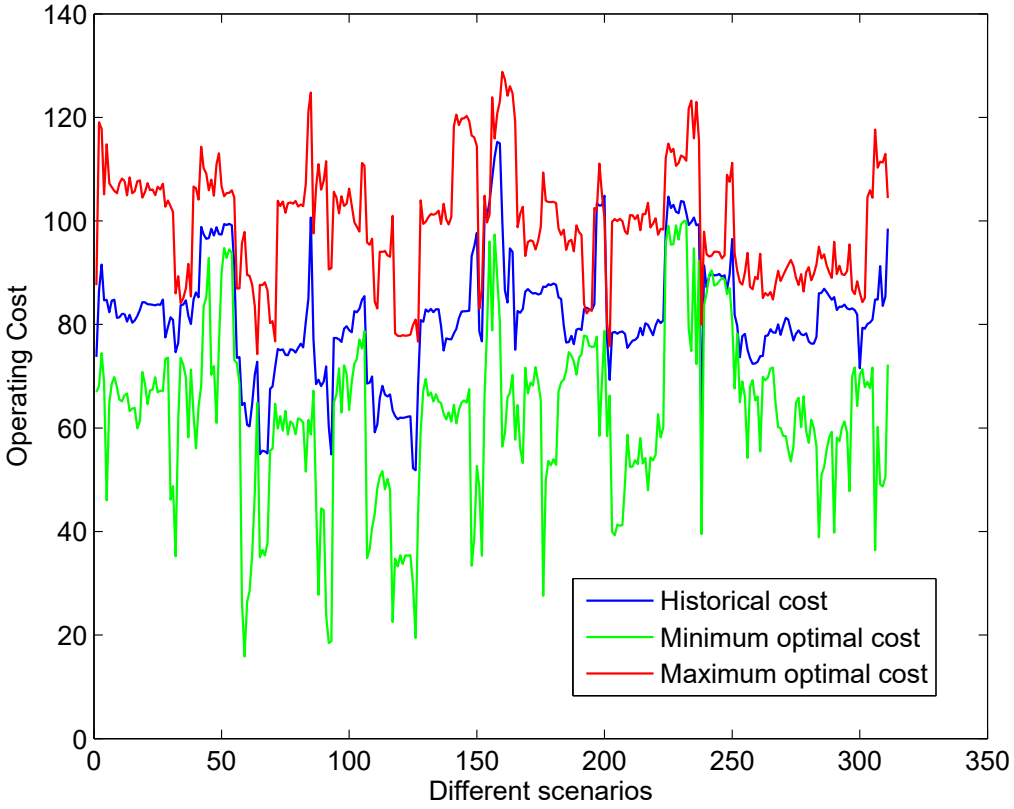


Figure 3.6: A comparison of steady state optimization results against selected historical data-points

Chapter 4

Optimal set point change strategy for water treatment network

Operators are often faced with the problem of negotiating sudden changes in operating conditions. The previous chapter describes a method using which operators can find a steady state optimal solution for the entire network given the active operating constraints. The next logical step is to find a strategy to arrive at this operating point. In this chapter, we discuss a strategy to arrive at the optimal steady state operating point for the water treatment network. Section 4.1 contains problem description. Assumptions made while constructing the problem statement are given in Section 4.2. Section 4.3 gives the mathematical problem statement. Dynamic process models are identified in Section 4.4. Results based on optimization framework are discussed in Section 4.5. Next, some comments are made regarding the use of Gaussian process models in system identification in Section 4.6. Finally, Section 4.7 concludes the chapter.

4.1 Problem description

The water treatment and steam generator network discussed in Chapter 3 consists of several process units which are controlled manually by different operators. There is no automatic control that ensures that the units are working at optimal set point. Skilled operators are able to maneuver plant set points based on their experience and fine judgment. Tanks in the network are used as buffers to ensure smooth change in process operating conditions. However, operators do not possess any tool which can

guide them towards the optimal operating point.

In industry, this problem is usually handled by adopting a real time optimization(RTO) and control strategy. In RTO, a global real time optimizer is used to obtain set points for local model predictive controllers [51] which are tuned to work in a coordinated fashion and drive the process to the optimal point. Several industrial applications of RTO based process control have been shown to be successful. Unfortunately, very often it is not economically feasible to implement RTO especially when most of the units are under manual operation, as in case of the water treatment network that is being studied in this work. In practice, RTO has been found to be a profitable investment only in high margin plants such as FCCUs or hydrocrackers [52]. This is because RTO implementations are hard to build and maintain and involve significant investment. Complex models are used to describe the process as accurately as possible. Parameters for these models are updated from time to time. In the case of a major change in plant operation, the complete RTO implementation must be revised. Owing to these challenges, it makes sense to find a simpler yet practical solution for optimization of the water treatment and steam generator network.

4.2 Assumptions

Certain assumptions were made regarding the network and the strategy using which it is driven to its new optimal point. These are given below:

1. **Manual control of variables**: It was assumed that as far as the available flow rate and tank variables are concerned, the complete network is under open loop control. Operators make decisions for changing the set points for some of the variables, while the rest are governed by dynamic process models or constraints identified using data. The number of manually controlled variables were assumed to be given by the difference between total number of variables and number of equality constraints. The rest were the so-called dependent variables whose behavior was deemed to be governed by process constraints. As seen in Figure 4.1, the input streams for water treatment units and steam generators are assumed to be operator controlled. Some of the output streams

are also considered to be operator controlled. 26 manually operated variables were identified for the actual industrial network used in this work.

2. **Data based models**: Dynamic models are required to predict the behavior of the process under set point change. First principle models for water treatment and steam generators can be difficult to build and use in optimization. This is because of the inherent complexity of the process. Thus, it was concluded that data based models would be a viable modeling option. Use of both linear and nonlinear data based modeling strategies was explored. Linear process models were identified using first order plus dead time model structure. Non linear modeling was performed using EM algorithm based robust GP regression as proposed in Chapter 2. Results from both methods are provided in Section 4.4.

3. **Time horizon for achieving new steady state**: A time horizon of size N was assumed for achieving new steady state condition. This is similar to the finite horizon condition in model predictive control.

4. **Time interval between set point changes**: It was assumed that within the time horizon N, all operators simultaneously change set points on manually controlled variables after every M time steps. Although this type of coordination between operators may not be achievable in practice, such an assumption can provide an ideal solution to the problem and hence can be used as a guideline.

5. **Fixed input water flow**: It was assumed that throughout the time horizon of transition, the input produced water flow to the network as seen in Figure 3.2, is fixed.

4.3 Problem statement

Based on above mentioned assumptions, the set point change solution was obtained using a dynamic optimization framework constructed as follows:

$$\underset{\mathbf{x}(t=1),\ldots,\mathbf{x}(t=N),\mathbf{z}(t=1),\ldots,\mathbf{z}(t=N)}{\operatorname{argmin}} \sum_{t=1}^{N} \left(||\mathbf{x_f}^* - \mathbf{x_f}(t)||^2 + ||\mathbf{z_f}^* - \mathbf{z_f}(t)||^2 \right) \tag{4.1}$$

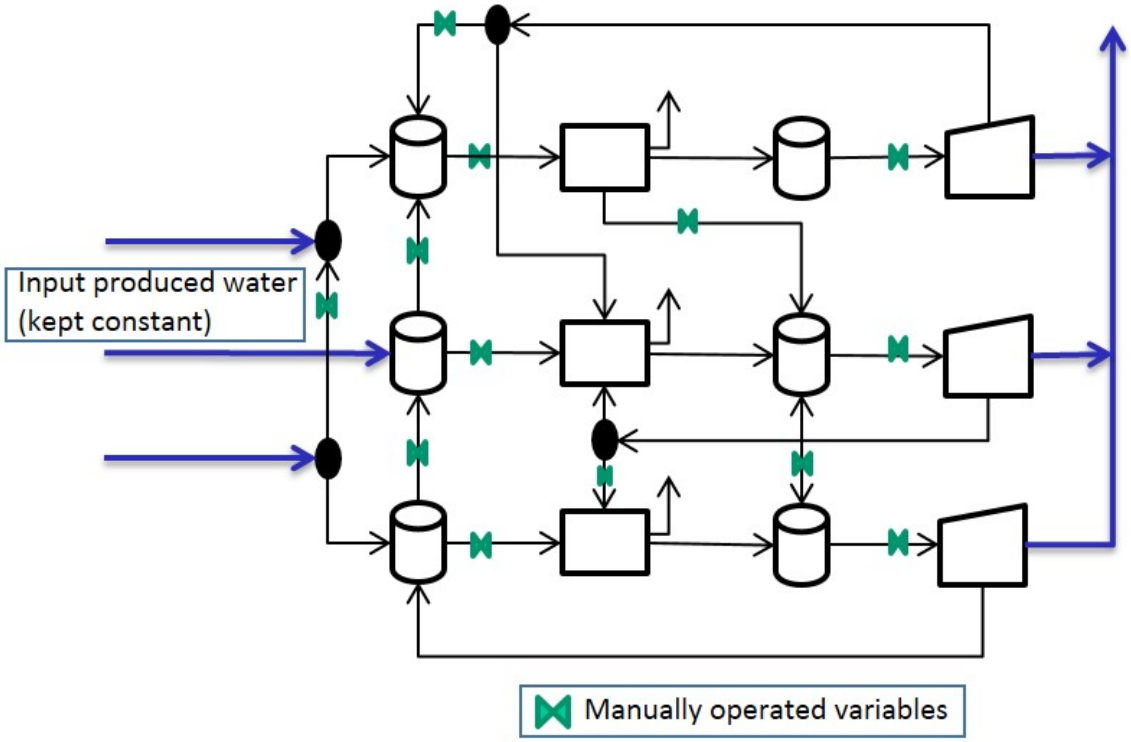


Figure 4.1: Manually operated variables in the network: All inputs to water treatment and steam generator units are manually operated. Some other streams are also under manual operation based on the number of independent variables in the network. Note that blocks in the figure represent the same process units as shown in Figure 3.2

such that,

1) Steam gen/water plants:

$$y(t) = G(z^{-1})u(t)$$

$$\textbf{or} \tag{4.2}$$

$$y(t) = f(y(t-1), u(t-1), y(t-2), u(t-2), \ldots)$$

2) Tank models:

$$A_{tank}x_{level}(t) + \sum_{i \in tank\_in} x_i(t) - \sum_{j \in tank\_out} x_j(t) = A_{tank}x_{level}(t+1) \tag{4.3}$$

3) Inequality constraints:

$$\mathbf{l} \leq \mathbf{x}(t) \leq \mathbf{h} \tag{4.4}$$

4) Operator controlled variables:

$$x_k(t) = x_k(t-1) \tag{4.5}$$

where k denotes the operator controlled variables for all $t \neq 1, M+1, 2M+1 \ldots$

5) Fixed produced water input:

$$x_{pw,in}(t) = x_{pw,in}(0) \tag{4.6}$$

6) Initialization:

$$\mathbf{x}(0) = \hat{\mathbf{x}}, \mathbf{z}(0) = \hat{\mathbf{z}} \tag{4.7}$$

where N is the time horizon over which new optimal steady state flow rates represented by $\mathbf{x_f}^*$ and $\mathbf{z_f}^*$ are to be achieved. Values for $\mathbf{x_f}^*$ and $\mathbf{z_f}^*$ are obtained from steady state optimization in Chapter 3. Note that optimal tank levels are not given by steady state optimization. That's why they do not appear in the objective function even though they appear in Equation 4.3 and 4.4 as constraints. Equation 4.2 is the dynamic model for steam generator or water treatment unit. Both linear and nonlinear models have been explored in this work. $y(t)$ represents the sum of all measurements of streams exiting a unit and $u(t)$ represents sum of all measurements of streams entering the unit.

$$\begin{aligned} u(t) &= \sum_{i \in plant\_in} x_i(t) \\ y(t) &= \sum_{i \in plant\_out} x_i(t) \end{aligned} \tag{4.8}$$

In the case of linear model $G(z^{-1})$ is a discrete transfer function. In the case of nonlinear model, $f(.)$ represents prediction from Gaussian process with regressors $y(t-1), u(t-1), y(t-2)$ and $u(t-2)$. The choice of regressors may vary depending upon the process model. Equation 4.3 represents the discretized first principles based dynamic tank model. Process inequalities are incorporated using Equation 4.4. In Equation 4.5, the freedom to change operator controlled variables at finite time intervals of size M is represented. Finally in Equation 4.6 the input to the system is fixed over the time horizon period and in Equation 4.7 the value of process variables is initialized at time $t = 0$.

In our simulations, M was chosen to be $1/6^{th}$ of the horizon length N. $\mathbf{x}(0)$ was initialized using the data reconciliation results from Chapter 3. As explained before, two different model identification strategies were explored in Equation 4.2.

4.4 Model identification

As mentioned in Chapter 2, process measurements are available at a sampling rate of 1 minute. Step response data was isolated for model identification from the time series of measurements. Data had to be preprocessed using the system ID toolbox in MATLAB before identifying model parameters. This was done as follows:

1. Data regions where input measurements were close to steady state were avoided in the selection of training set. A continuous time series region involving a step input response was extracted from the dataset without excessive fluctuations or noise in process measurements. A similar exercise was performed to extract a test set.

2. Outliers were removed from the training set based on visual inspection and replaced by the mean of adjacent process measurements.

3. Mean of the data was removed from both the training and test sets.

4. Training and test data were down-sampled from 1 minute to 3 minute for all plants before identification.

5. Input and output data for training as well as test set were normalized between 0 and 1.

4.4.1 Linear model identification

Linear model identification was performed as follows:

1. First order plus dead time models were identified using preprocessed step response data. Since a steady state material input must equal output, the gain of the dynamic process model must be 1. In reality, some flow rates are missing such as slurry flow rate for some water plants. Moreover, measurements are noisy. As a result input and output flow rates are never balanced. Therefore after identification the gain for the process was approximated bys 1.

2. Identified unit gain first order plus dead time model was converted to discrete transfer function formulation.

3. WLS/WAC water plant models were found to have a response time between 10-20 minutes. The poles for the identified transfer function were between 0.6 and 0.9 for these water plants. Higher order time series models could be identified; however it was not leading to sufficient improvement in prediction performance on test set. Therefore the following models were used in optimization code for the two WLS/WAC water plants (Plants 1 and 3).

$$\begin{aligned} &\text{water plant 1: } G_1(z^{-1}) = z^{-3}\frac{0.3792 + 0.007235z^{-1}}{1 - 0.6136z^{-1}} \\ &\text{water plant 3: } G_3(z^{-1}) = z^{-1}\frac{0.1511}{1 - 0.8489z^{-1}} \end{aligned} \tag{4.9}$$

4. Identified steam generator models and evaporator (water plants 2a, 2b, 2c and 2d) models had a very small time constant (less than 3 minutes). From an optimization point of view these units can be considered to be fast rate in comparison to WLS/WAC water treatment plants. For simplicity, the following arbitrary chosen linear process model was used for all these units:

$$G(z^{-1}) = \frac{0.92z^{-1}}{1 - 0.08z^{-1}} \tag{4.10}$$

In Figures 4.2 and 4.3 we can see the one-step ahead prediction performance of identified linear models on training set for water plants 1 and 3. Note that the step input response for water plant 1 in Figure 4.2 has a small overshoot which is not captured by the identified process model.

4.4.2 Nonlinear model identification using EM based robust GP regression

Nonlinear model identification using EM based robust GP regression was performed for water plants 1 and 3 using the same training and test sets extracted for linear model identification. Laplace likelihood was selected for noise model and Expectation propagation was used for approximating the posterior distribution. This corresponds to the lnEMEP procedure described in Chapter 2. One and two step delayed input and output were chosen as regressors in the model. Increasing the number of regressors did not improve prediction performance on test set. The evaporators and steam

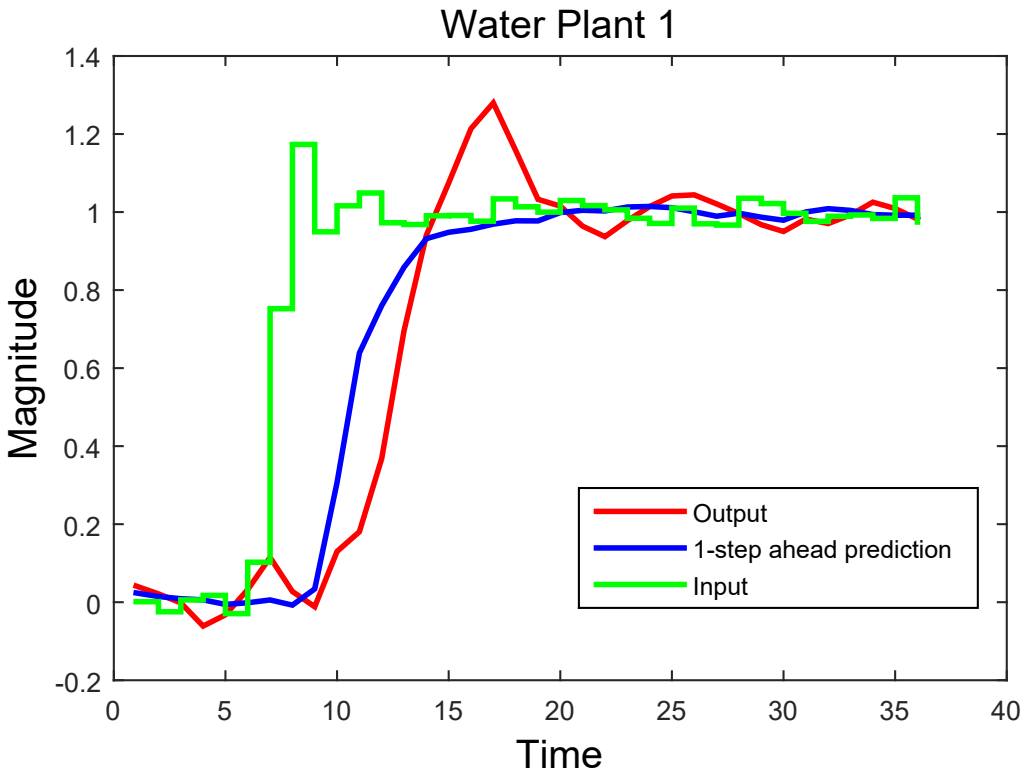


Figure 4.2: One-step ahead prediction using identified linear process model for water plant 1

generators were assumed to follow the fast rate process model described in Equation 4.10. The following nonlinear autoregressive models were considered for water plants 1 and 3:

$$\begin{aligned} y(k) &= f(\mathbf{x}(k)) + \epsilon \\ \text{where } \mathbf{x}(k) &= [y(k-1), u(k-1), y(k-2)]^T \text{ for water plant 1} \\ \text{where } \mathbf{x}(k) &= [y(k-1), u(k-1), y(k-2), u(k-2)]^T \text{ for water plant 3} \\ \text{and } \epsilon &\sim \text{ Laplace distribution}(0, b) \end{aligned} \tag{4.11}$$

f refers to Gaussian process. Noise is assumed to follow Laplace distribution with hyper-parameter b as described in Chapter 2. Similar to Equation2.32, the following choice of kernel gave the best results [19].

$$K(\mathbf{x}_i, \mathbf{x}_j) = m_0 + \sum_{d=1}^{D} m_d x_i^d x_j^d + \sigma_{se}^2 exp\left(-\frac{1}{2}\frac{\left(x_i^d - x_j^d\right)^2}{l}\right) + v_0\delta_{ij} \tag{4.12}$$

The first term in this kernel is a constant bias while the second term captures linear correlations between input variables and response variables. Nonlinearity in the rela-

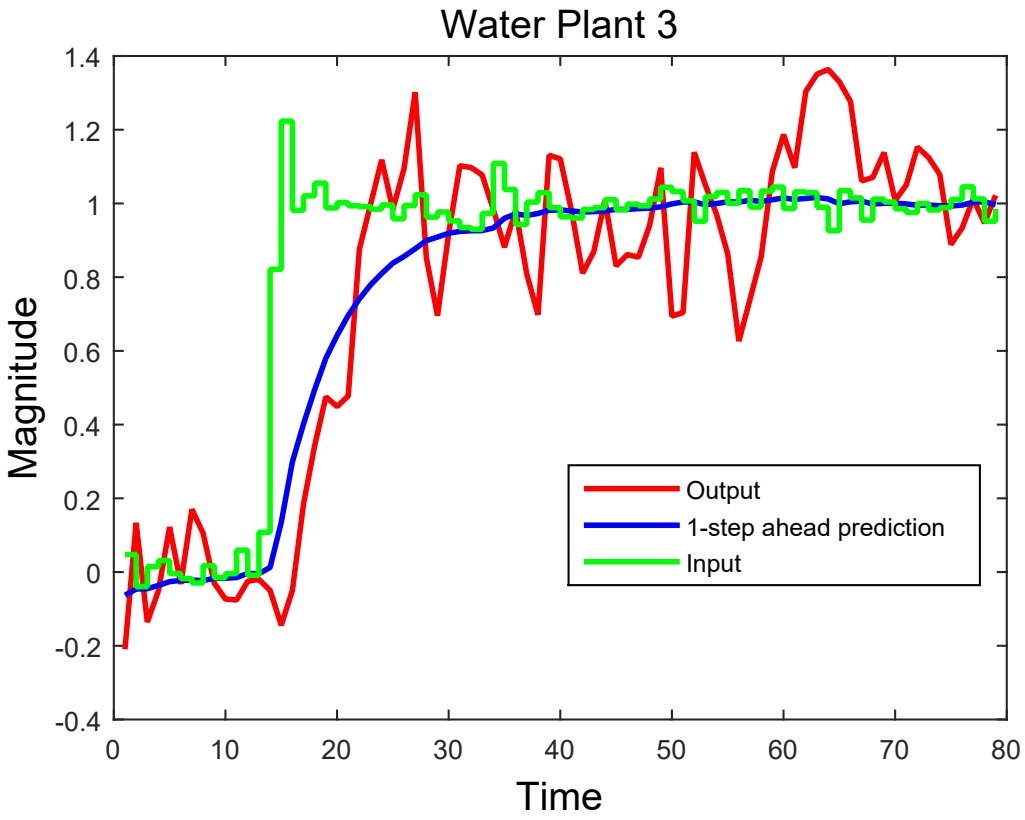


Figure 4.3: One-step ahead prediction using identified linear process model for water plant 3

tionship is modeled by the third term which is a squared exponential function. The last term accounts for random error effect in the input variables. It was observed that the use of linear kernel function (second term) is important for improving the extrapolation ability of the identified Gaussian process model. This will be further discussed in Section 4.6.

The kernel hyper-parameters and noise hyper-parameters are learned using the proposed EM-EP approach. The hyper-parameter values for the identified models are given in Table 4.1.

As mentioned earlier, both the training and test data were normalized between 0 and 1. In the optimization procedure, input was not restricted to unit changes in magnitude. Therefore the output response for a step change in input was modeled by scaling the unit step response according to the size of the step input. The reason for adopting this approach is discussed in subsection 4.6.1. From Figures 4.4 and 4.5, we can see that one-step ahead predictions from trained robust GP models match

Table 4.1: Optimized hyper-parameter values for identified robust GP models: Values correspond to the sequence $\log([m_0, m_1, m_2, \ldots, m_d, l, \sigma_{se}, v_0])$ as per Equation 4.12, where d is the number of regressors

Water Plant	No of regressors	Model hyper-parameters
1	3	[-33.78 -1.85 -4.89 0.55 -4.23 -2.36 -9.42]
3	4	[-42.52 -3.08 -55.91 -40.67 -0.46 12.74 -21.26 -5.11]

well with the output.

Simulation results on validation set are shown in Figures 4.6 and 4.7 for both linear and robust GP models. In Table 4.2, we compare the performance of robust GP models and linear process models in terms of RMSE on validation data set. Both one-step ahead and infinite-step ahead (simulation) results are shown in the table for robust GP models. One-step ahead results from linear models is same as simulation results since it is an output-error (OE) model. As seen from the table, robust GP models are good at one-step ahead prediction but not as good in the case of infinite-step ahead prediction. Nevertheless, it is still better than or comparable to linear process models.

In case of water plant 1, robust GP models give a better performance than linear models. This is because process dynamics of water plant 1 are nonlinear. The nonlinearity of water plant 1 dynamics can also be assessed from the hyper-parameter values for squared exponential kernel in Table 4.1. A large value of σ_{se} increases the contribution of squared exponential kernel to the covariance matrix. On the other hand, a small value of l gives more weight to nearby input locations thereby providing a nonlinear structure to the model. Based on these facts, it can be observed that in the case of water plant 1 the values for σ_{se} and l make the model nonlinear whereas in the case of water plant 3 they do not impose significant nonlinearity. In conclusion we can say that the use of GP kernels allows the identification of both linear and nonlinear regression models and the use of heavy tailed noise distribution helps reduce the effect of outliers.

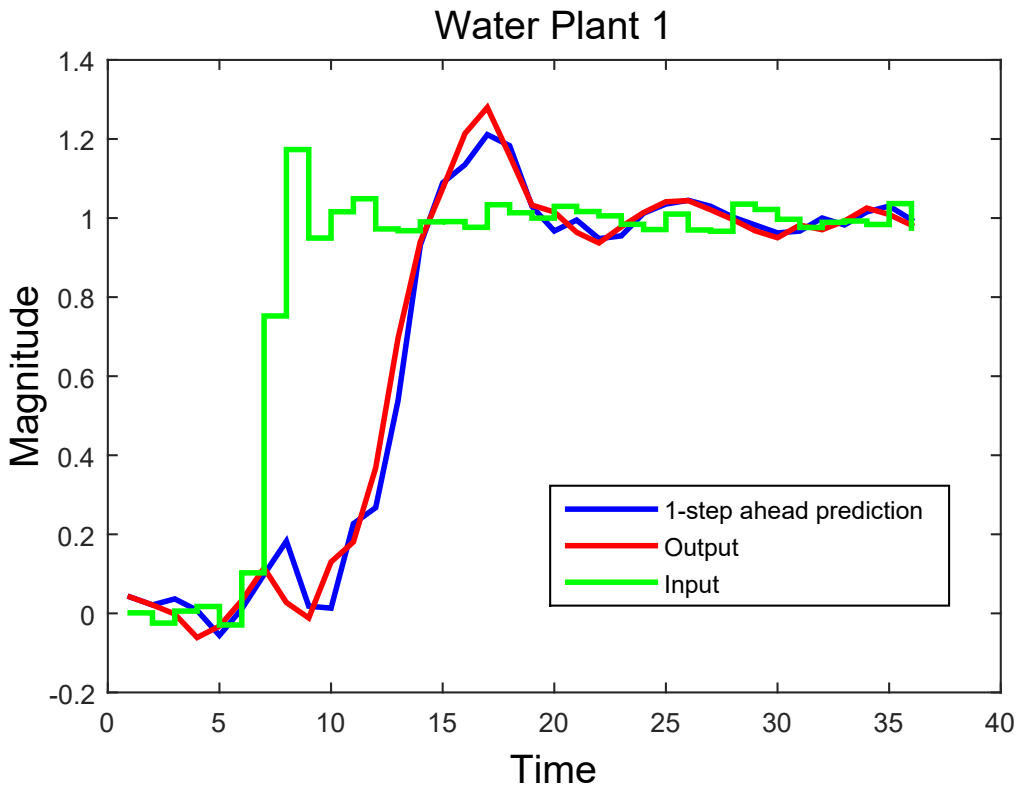


Figure 4.4: One-step ahead prediction using identified robust GP model for water plant 1 on training data

4.5 Results

The complete data reconciliation, optimization and set point change strategy proposed in Chapters 3 and 4 is summarized in form of a flowchart in Figure 4.8. In this section we discuss the results from the optimization framework. The main output is the set point change strategy for 26 manually operated variables. Optimization with linear models was found to be approximately 5 times faster than optimization with nonlinear robust GP models.

Based on the choice of process models for water plants 1 and 3 (linear or nonlinear), two sets of results are shown in subsections 4.5.1 and 4.5.2 respectively. In addition to exploring linear and nonlinear constraints for optimization, a variation to the set point change constraint was studied and the corresponding results are shown in subsection 4.5.3.

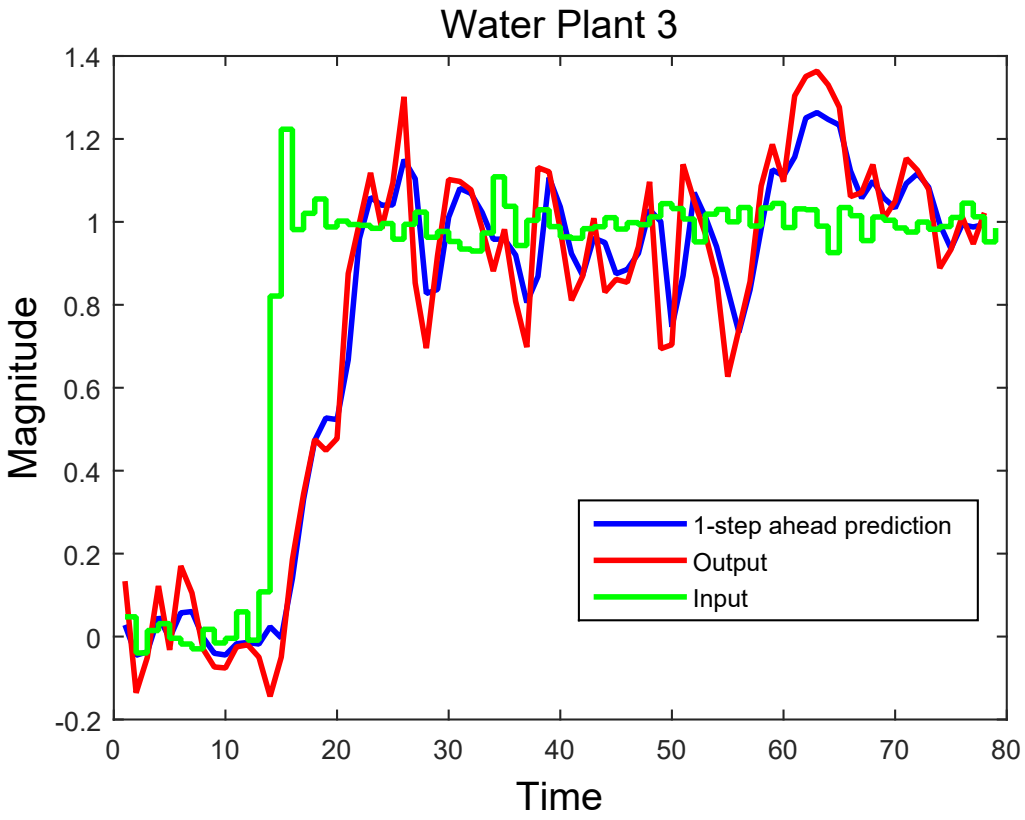


Figure 4.5: One-step ahead prediction using identified robust GP model for water plant 3 on training data

4.5.1 Optimization with linear models

Linear models identified as shown in Equation 4.9 were used in optimization to find the set point changing strategy. A horizon of 60 time steps was chosen for optimization. In Figure 4.9a, the optimized set point change solution for 9 of the 26 manually operated variables can be observed. It can be seen that set point changes are initiated at intervals of 10 time steps. For some set points, the transition to final set point value is performed immediately whereas for others it follows a small set of changes. These small delays in arriving at final set point ensure that constraints on tank levels are not violated. In Figure 4.10a, tank level changes are plotted. Here we can see the impact of set point changes on tank levels. Almost all tanks witness a rise in their level and none of the tanks shows any drop in levels. We can also notice how the set point manipulations ensure that tank levels never breach the constraints. Cost of production plot is shown on the bottom right of Figure 4.10a. It can be seen that cost

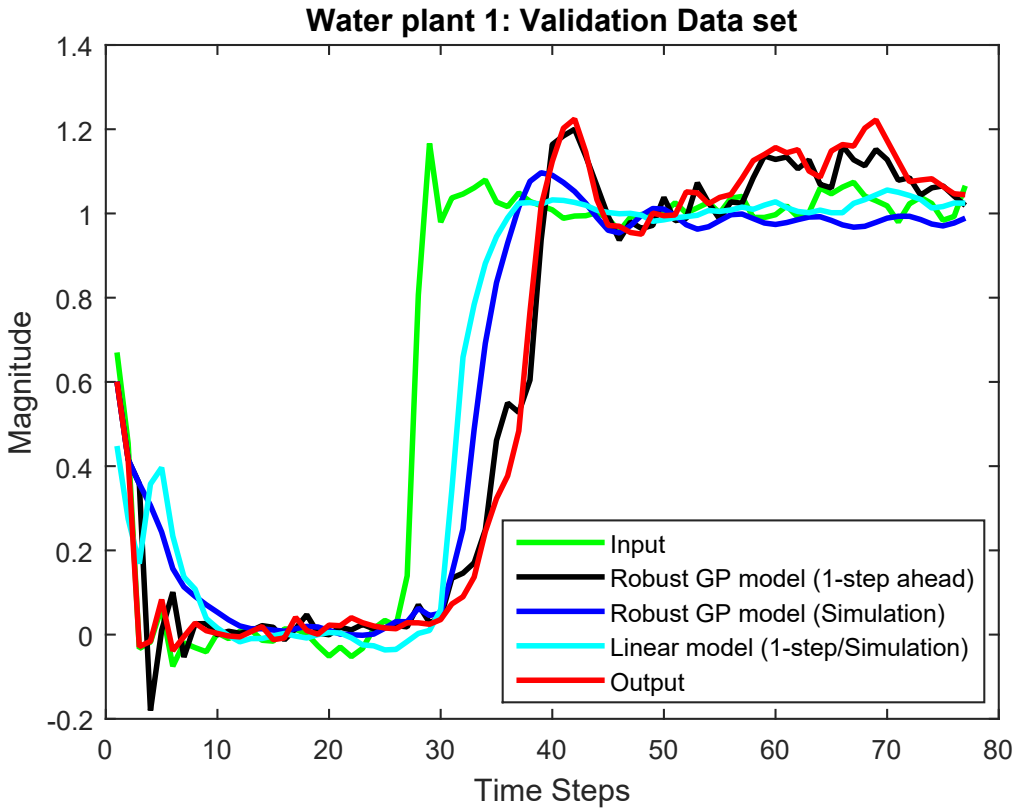


Figure 4.6: Prediction results using identified robust GP model and linear model for water plant 1 on validation set

of production immediately drops to a lower value. This is because the optimal steady state set point was found based on minimizing the cost of steam production and since steam generator dynamics modeled using Equation 4.10 are very fast, any change to boiler feed water input flow rate is immediately reflected in the steam production.

4.5.2 Optimization with nonlinear robust GP models

Gaussian process models identified using the approach described earlier were used in optimization to find the set point changing strategy. Figure 4.9b shows the results from this method for 9 of the manually operated variables. Comparing Figures 4.9a and 4.9b, it can be seen that the set point change results are similar. This is to be expected since the same dataset was used for model identification. Also the same reconciled and steady state optimal values were used in both optimization methods. Small differences in the two methods could be attributed to the different process models.

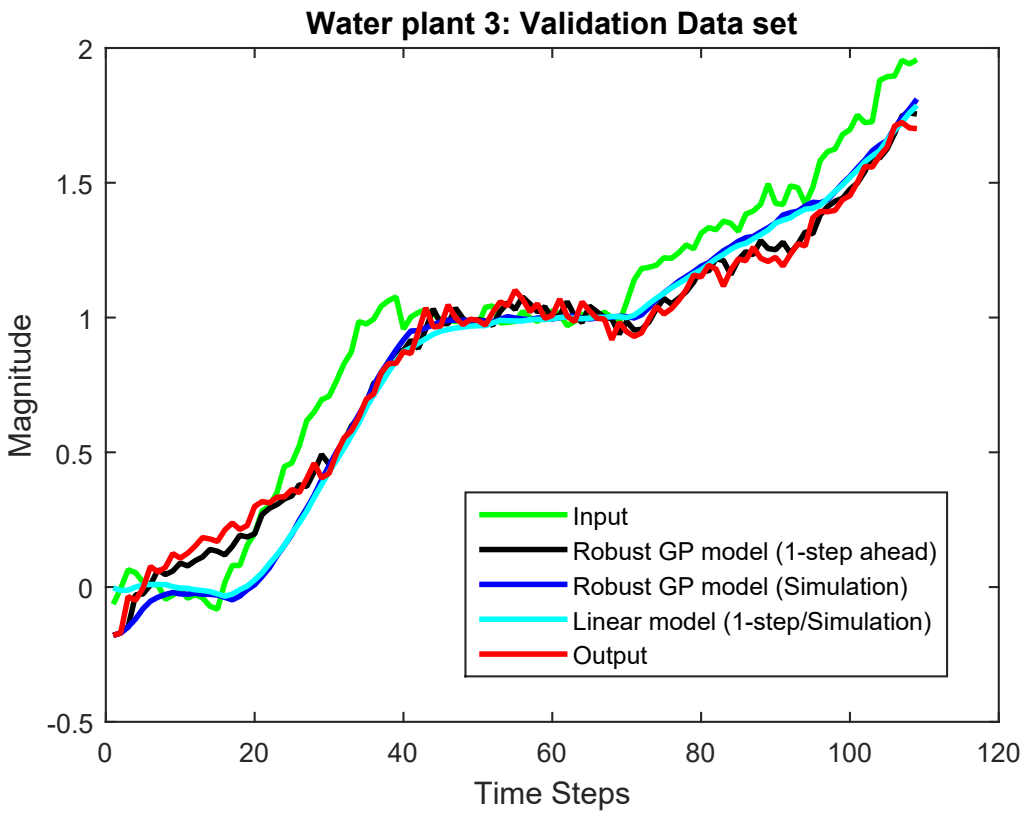


Figure 4.7: Prediction results using identified robust GP model and linear model for water plant 3 on validation set

As far as tank levels are concerned in Figure 4.10b, most tanks arrive close to their steady state. A slight difference can be levels at which tank levels stabilize in Figure 4.10b in comparison to Figure 4.10a. This is because of the use of different process models for water plants in the two cases. On the bottom right of Figure 4.10b, the drop in cost of production can be seen. Once again, the final optimal value of objective function is the same as in the case of linear model based optimization. Depending on which model (linear or robust GP) describes the process dynamics better, one of the two optimization schemes can be adopted. In this case since validation test performance of robust GP models was better than linear models, as seen in Table 4.2, it can be concluded that use of robust GP models for optimization is more suitable.

4.5.3 Optimization with "ramped" up set point changes

It was observed in the process data that manipulated variables are not changed based on abrupt step changes; instead they are changed in smaller steps over a short period

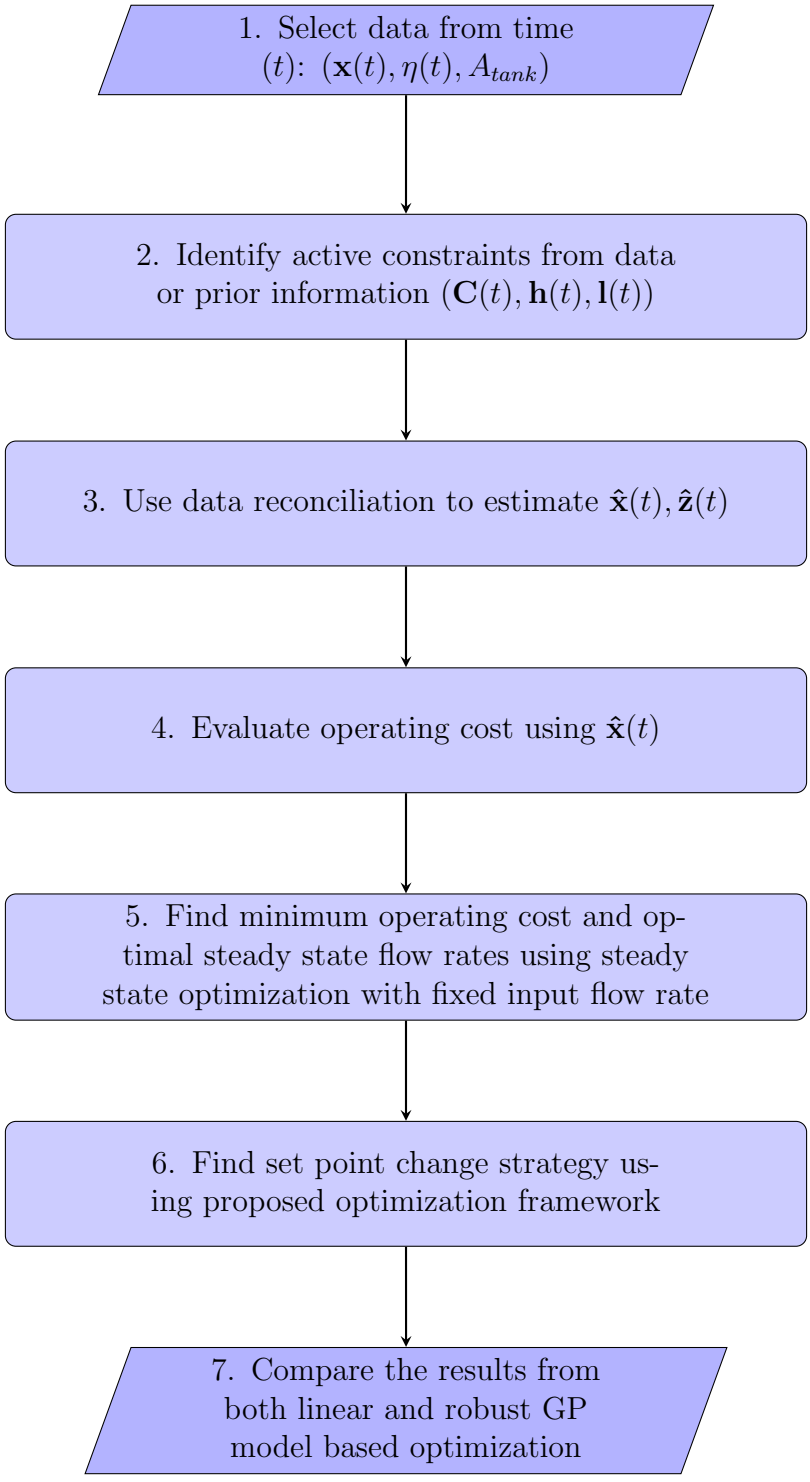
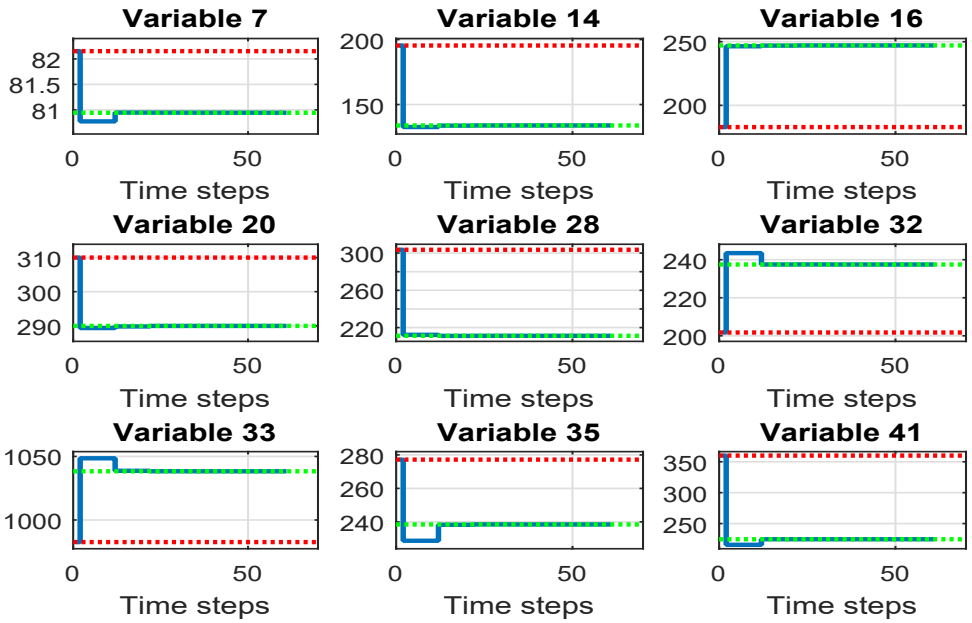
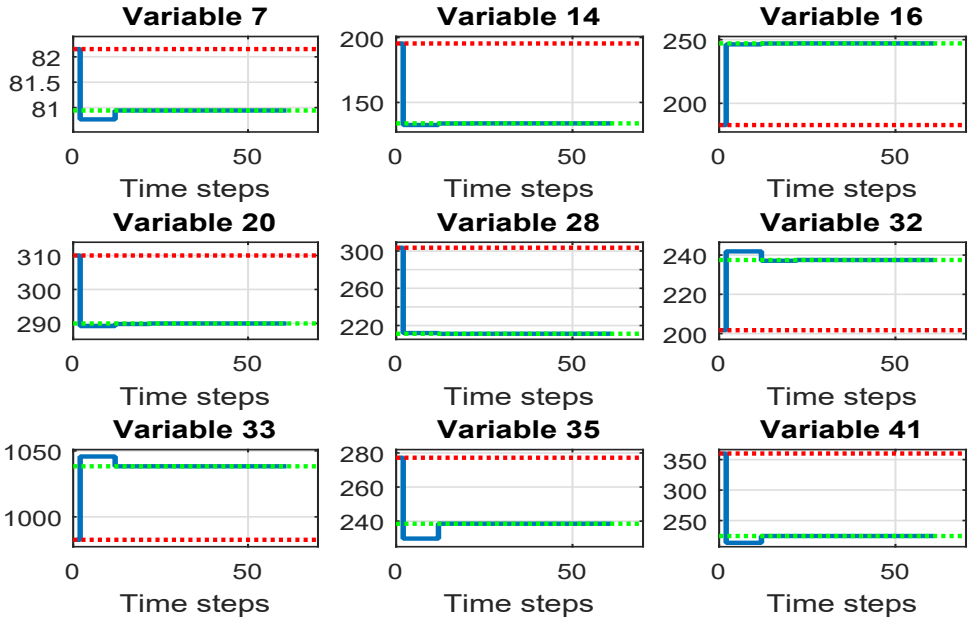


Figure 4.8: Approach for finding optimal set point change strategy

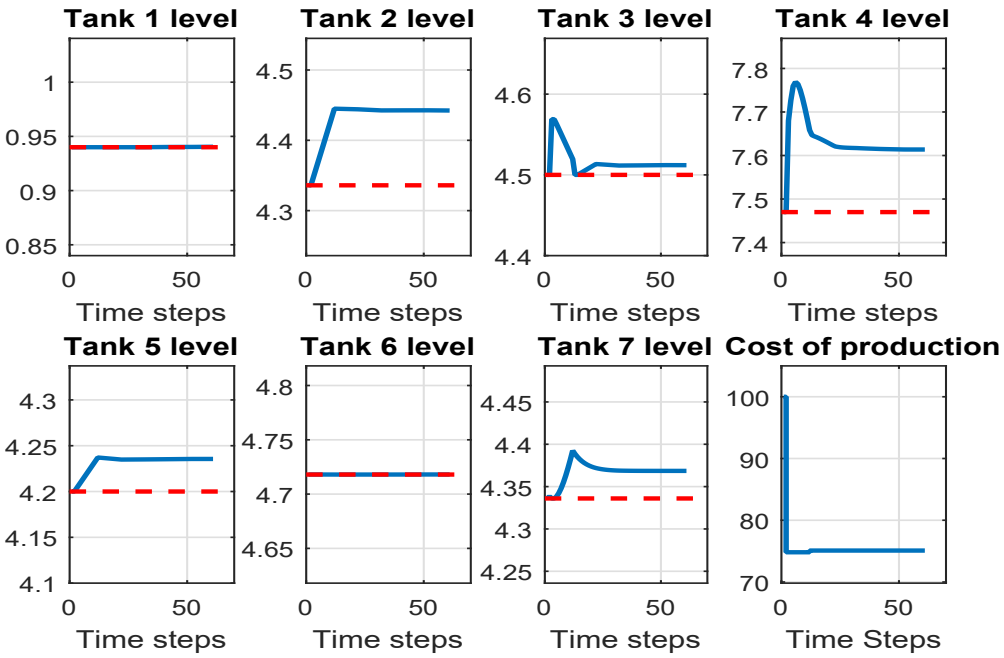


(a) Using linear water plant models

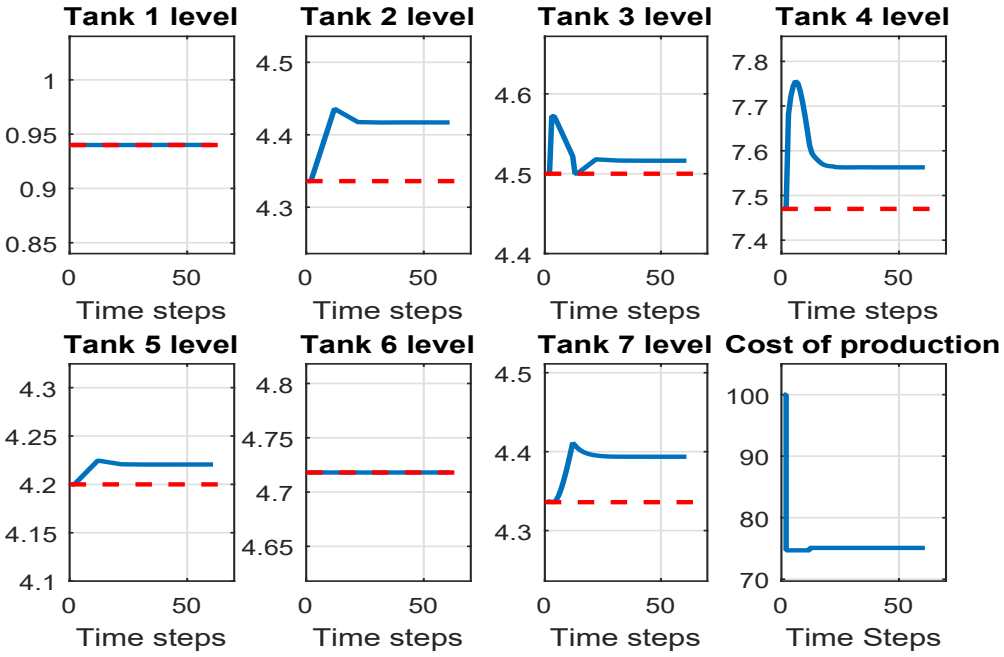


(b) Using robust GP water plant models

Figure 4.9: Results for set point changing strategy for manually operated variables with linear and robust GP water plant models: Red dotted line represents the initial set point and green dotted line represents the optimal set point. Blue bold line signifies the set point change strategy. Clearly, results are not very different for the two modelling methods



(a) Using linear water plant models



(b) Using robust GP water plant models

Figure 4.10: Tank level plots for optimization with linear and robust GP water plant models: Blue solid lines represent the change in tank levels. Red dotted lines in some of the plots depict the lower operating limit for the tank; On the bottom right, blue solid line signifies cost of production. Tank level variations in the case of linear models are a little different than in the case of robust GP models

Table 4.2: RMSE on validation data set: Both simulation and one-step ahead prediction results are shown

Water plant	Robust GP model (Simulation)	Linear process model (Simulation)	Robust GP (1-step prediction)	Linear process model (1-step prediction)
1	0.1697	0.2024	0.0680	0.2024
3	0.1063	0.1018	0.0428	0.1018

of time. The dynamic optimization proposed above provides a step change solution where there is no limit to the step size. In order to simulate a slower ramp like change in set points, a different set point change constraint was used. If the absolute difference between initial and final optimal set point for a manually operated variable was large - defined to be atleast 20% of the difference between maximum and minimum possible value for that variable - then that variable was constrained to follow a slow ramp like transition to the optimal value. For such variables Equation 4.5 is modified as follows

$$\begin{aligned} x_k(t) &= x_k(t-1) \\ \text{where } k &\in \text{operator controlled variable for all} \\ t &\neq 1, 0.5M+1, M+1, 2M+1, 2.5M+1, 3M+1, \ldots \end{aligned} \tag{4.13}$$

The following constraints are added:

$$\begin{aligned} x_k(t) &= \frac{4}{7}x_k(t+M) + \frac{3}{7}x_k(t-1) \\ x_k(t+0.5M) &= \frac{6}{7}x_k(t+M) + \frac{1}{7}x_k(t-1) \\ \text{where } k &\in \text{operator controlled with large change for all} \\ t &= 1, 2M+1, 4M+1, \ldots \end{aligned} \tag{4.14}$$

In other words, pair-wise set point values at time steps $(1, 0.5M+1), (2M+1, 2.5M+1), \ldots$ are constrained to follow a progression between the set point values at time steps $(0, M+1), (2M, 3M+1), \ldots$ respectively. The set point values at times $M+1, 3M+1, \ldots$ are kept free. For example if $M = 10$, then based on the constraints in Equations 4.13 and 4.14, a sample set point change strategy is given in Figure 4.11. To generate this figure, set point value at time $M+1$ was fixed as 1 and at time

$3M + 1$ was fixed as 0.87.

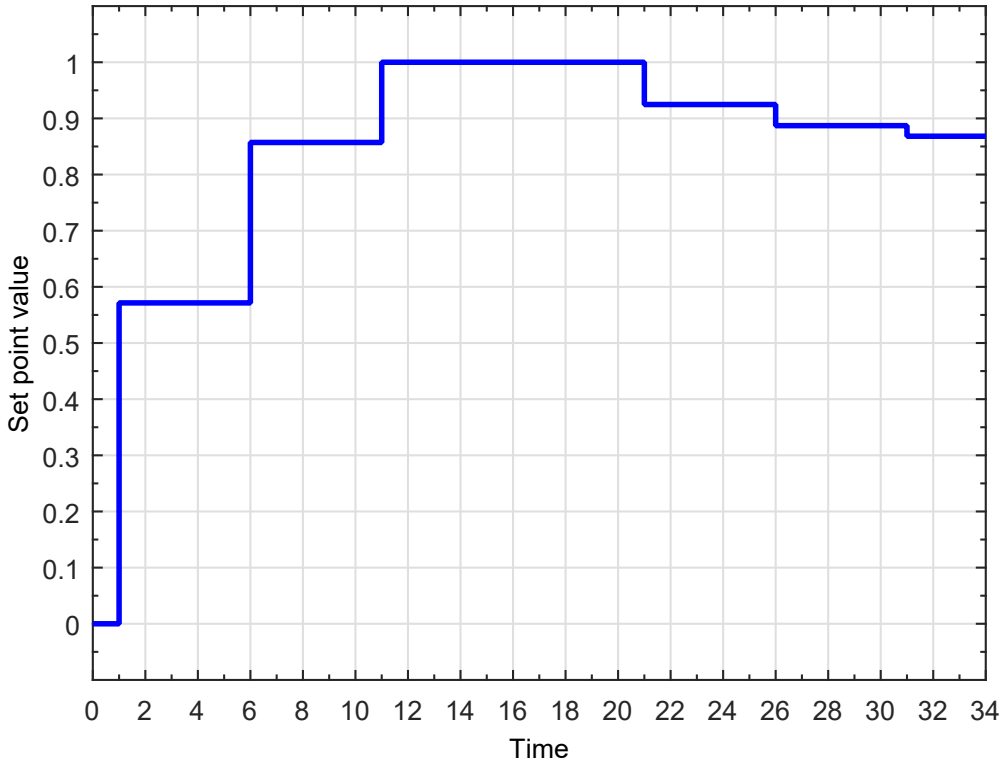


Figure 4.11: Sample set point change strategy with constraints based on Equations 4.13 and 4.14

The results for this problem formulation in case of linear model are given in Figures 4.12a and 4.13a, and in case of robust GP models are given in Figures 4.12b and 4.13b. We can see that in both these cases, the solution is noticeably different from the original problem formulation. Set point changes suggested by optimization are more intricate. The resulting changes in tank levels are also significantly different. This is expected since large set point changes are not allowed in the new formulation.

A closer look at the subfigures within Figure 4.12 shows that the use of robust GP models instead of linear models gave similar results for set point changes. However, predicted tank level variations were noticeably different especially in case of Tanks 3 and 4 as seen in Figures 4.13a and 4.13b. This is because of the use of linear and GP based process models for water plants.

Finally, a summary of results from the different modelling and optimization strategies discussed in this chapter is shown in Table 4.3. The objective function refers to

Equation 4.1 which is the same in all cases. Use of robust GP models gives a slightly higher objective function value in comparison to the use of linear models. This could be because GP models are nonlinear and take slightly longer to achieve steady state as seen in Figure 4.6. As expected, a ramped up set point change strategy also results in a higher objective function value. The effects of ramped up set point change strategy and slower dynamics of identified GP models are also reflected in the increase in tank water volumes in the third column of the table.

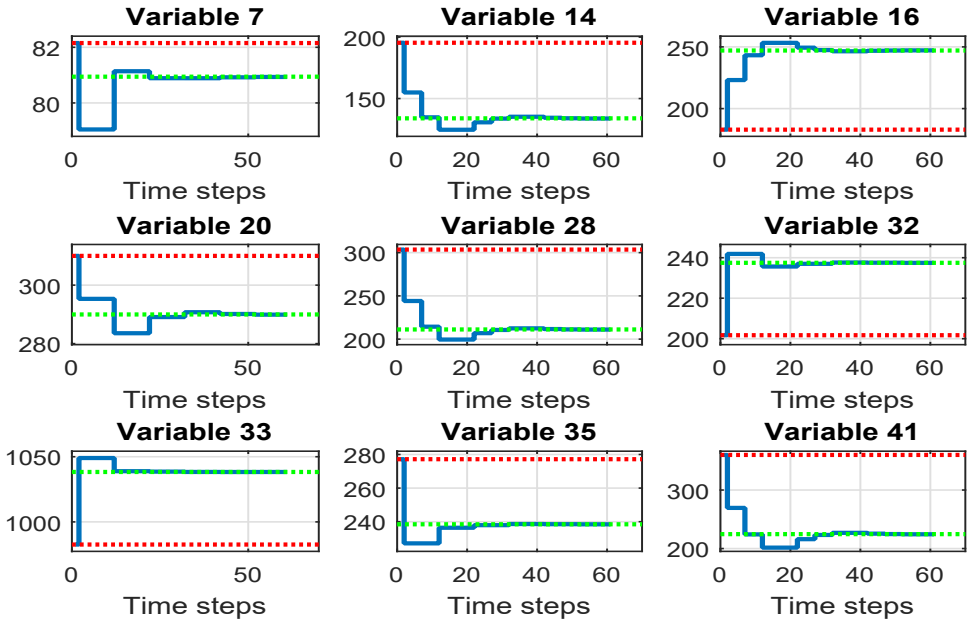
Table 4.3: Value of optimized objective function and % increase in water stored in tanks

Method	Objective function value	Increase in tank water storage (in %)
Step change (Linear model)	$4.7800e5$	0.6007
Step change (Robust GP model)	$4.8302e5$	0.6771
Ramped change (Linear model)	$5.6999e5$	2.2788
Ramped change (Robust GP model)	$5.7818e5$	2.4662

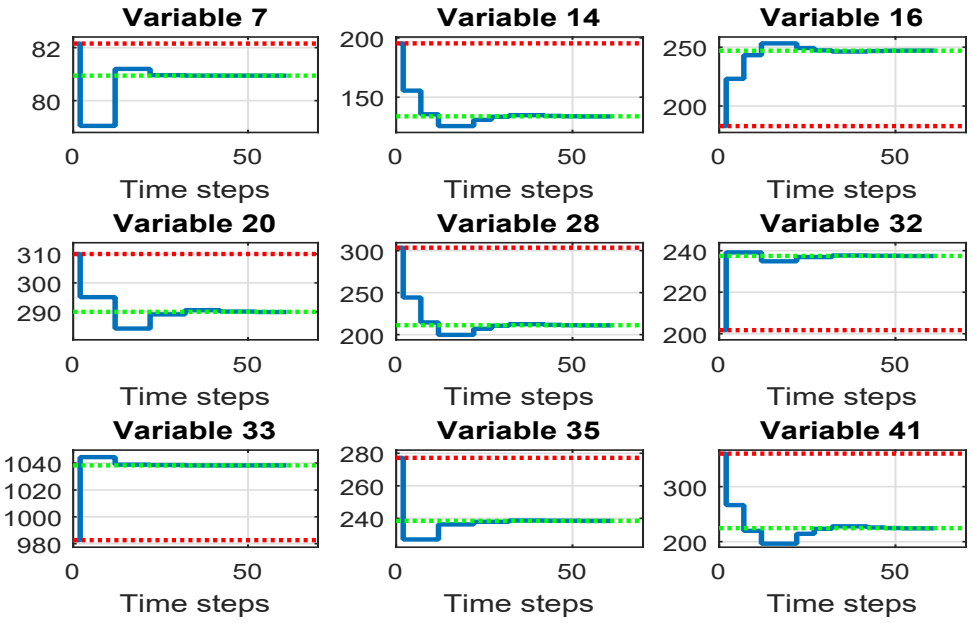
4.6 Comments on use of robust GP regression for process identification

This section contains some discussions regarding the benefits and challenges associated with the use of GP regression for process identification.

In this study, a robust GP regression model identified using proposed EM algorithm was used on an industrial dataset as part of an optimization task. Results suggest that it is possible to apply robust GP models to chemical engineering optimization problems. Due to the similarity of proposed optimization approach with model predictive control (MPC), it can be supposed that GP regression can be successful even with conventional MPC formulations. In fact several authors have applied GP regression successfully to synthetic process identification datasets and used it in MPC [22, 18]. They have also made use of the uncertainty in prediction given by GP models in designing better model predictive control solutions. The proposed robust

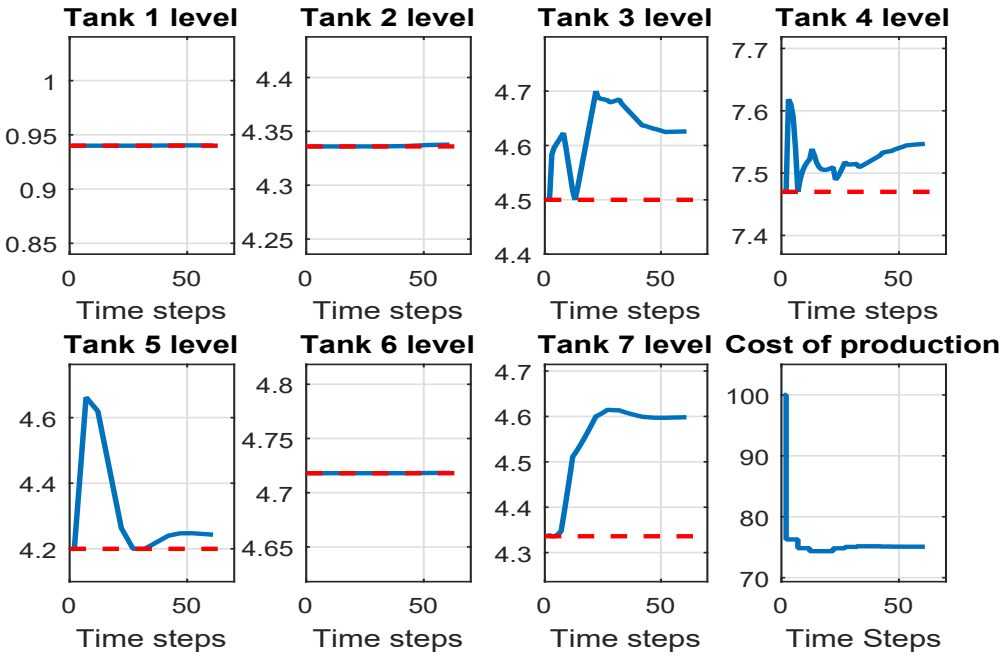


(a) Using linear water plant models

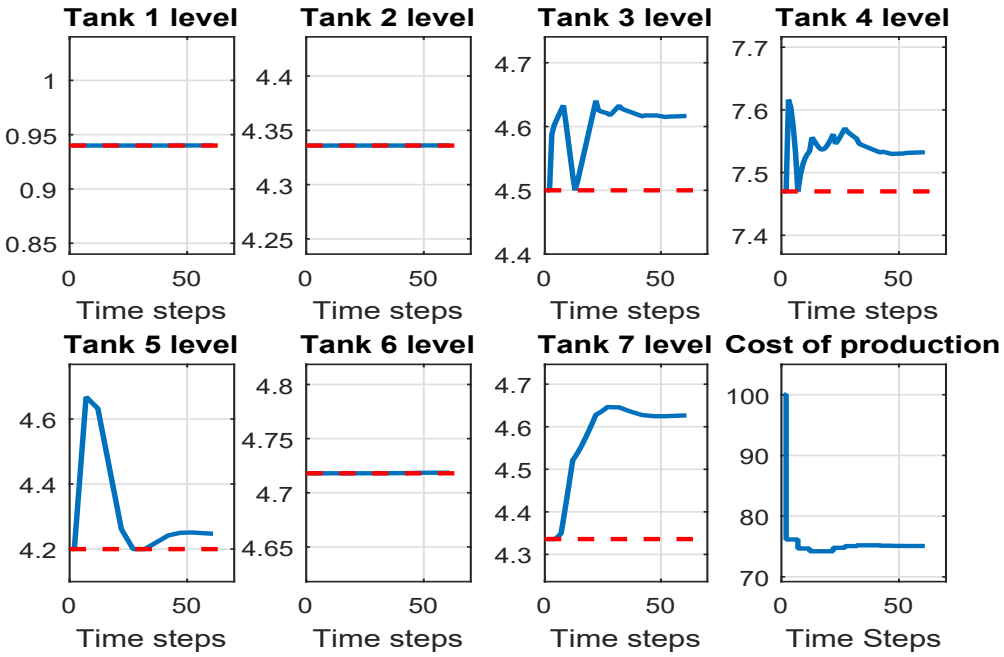


(b) Using robust GP water plant models

Figure 4.12: Results for "ramped" up set point changing strategy for manually operated variables: In this case we can see that some variables have a gradual change in set points.



(a) Using linear water plant models



(b) Using robust GP water plant models

Figure 4.13: Tank level and cost of production plots for "ramped" up set point changing strategy: Notice that tank levels show a different trend compared to original formulation. The cost of production dynamics are also different in this case.

GP regression identified using EM algorithm can also be used in such an implementation. Based on our observations, some factors that must be accounted for before using GP regression models for system identification are mentioned below:

4.6.1 Extrapolation performance of GP process models

In the above model identification procedure, training step test data for both input and output were normalized between 0 and 1. At steady state the output stabilizes at 1. In other words, the gain of the process model must be 1. This is because of mass balance property for the unit. It was observed that when a different magnitude of step input was given to the GP model, the infinite ahead prediction output did not behave according to a process with gain 1. This is shown in Figure 4.15, where both GP models fail to maintain the process gain when input is increased to 2. Both models were identified using the training dataset for water plant 1.

One reason for not maintaining process gain is that industrial training data is noisy. To verify this, a linear model with process gain 1 and high signal to noise ratio was used to generate step response training data in the $[0, 1]$ range. This model is given in Equation 4.15.

$$\begin{aligned} y(t) &= 0.75 * y(t-1) + 0.25 * u(t-1) + \epsilon \\ \text{where } \epsilon &\sim \mathcal{N}(0, 10^{-4}) \end{aligned} \tag{4.15}$$

A robust GP model using the kernel in Equation 4.12 was trained using this dataset and validated against a step response dataset in the $[0, 5]$ range. In Figure 4.14, we can see that low noise in training data allows good extrapolation results. This is because constructing a robust GP model is similar to constructing a nonlinear ARX model wherein both process and noise models are identified. This training data for GP models must either have high signal to noise ratio or contain a signal with sufficient range and excitation. Similar findings have been made by other authors [53].

Since industrial data used in this work is noisy, certain steps were taken to address the error in extrapolation problem. The optimization framework allows a step change of variable magnitude. Thus, for every step change in input, the magnitude of the step change was used to scale the output of the trained GP process model. This ensured that the gain of the process was 1 irrespective of step size.

4.6.2 Choice of kernels

The choice of kernels (covariance function) can affect the performance of identified process models significantly. In the above described implementation a combination of linear, constant and squared exponential covariance functions was used to construct a kernel. From the point of view of extrapolation, it was observed that the use of linear kernel is desirable. Relying only on squared exponential or radial basis function kernel may not lead to successful model identification. This is because, radial basis function only recognizes strong correlation between outputs and nearby inputs. Predictions at test set input locations which are not close to training set locations tend to be poor. In Figure 4.15, we can see the poor prediction from a robust GP model using only radial basis function kernel.

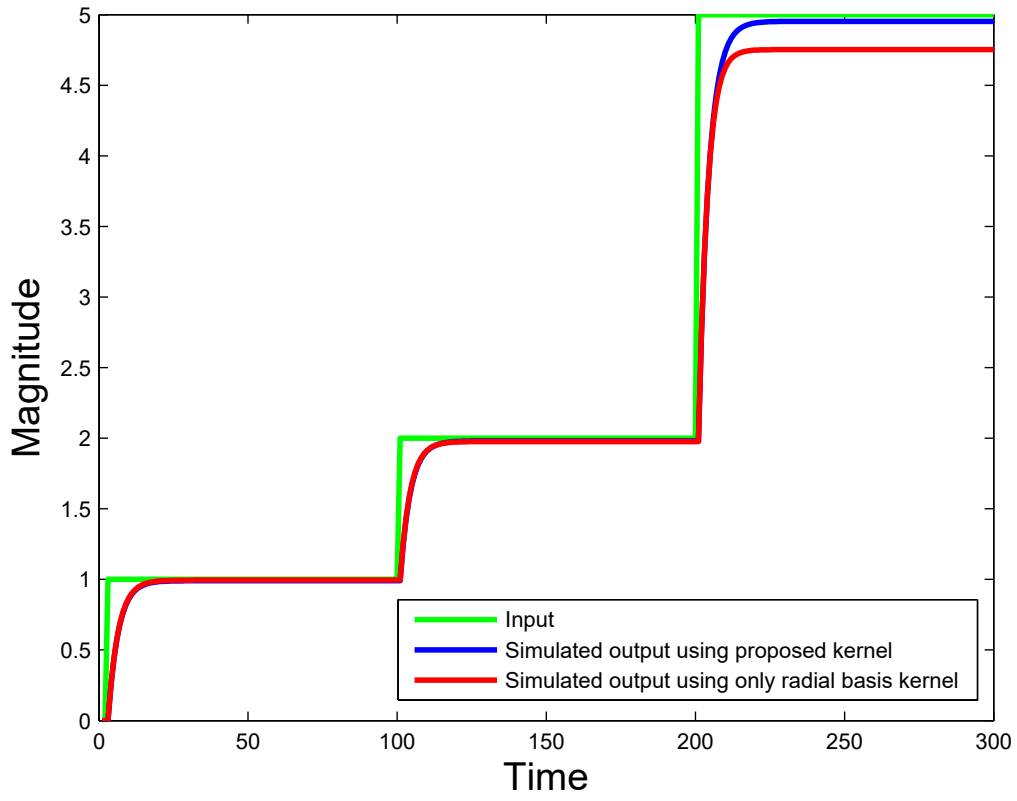


Figure 4.14: Example to show effect of noise in training data on extrapolation performance. Also note that the proposed kernel which also uses a linear covariance function performs better than radial basis kernel

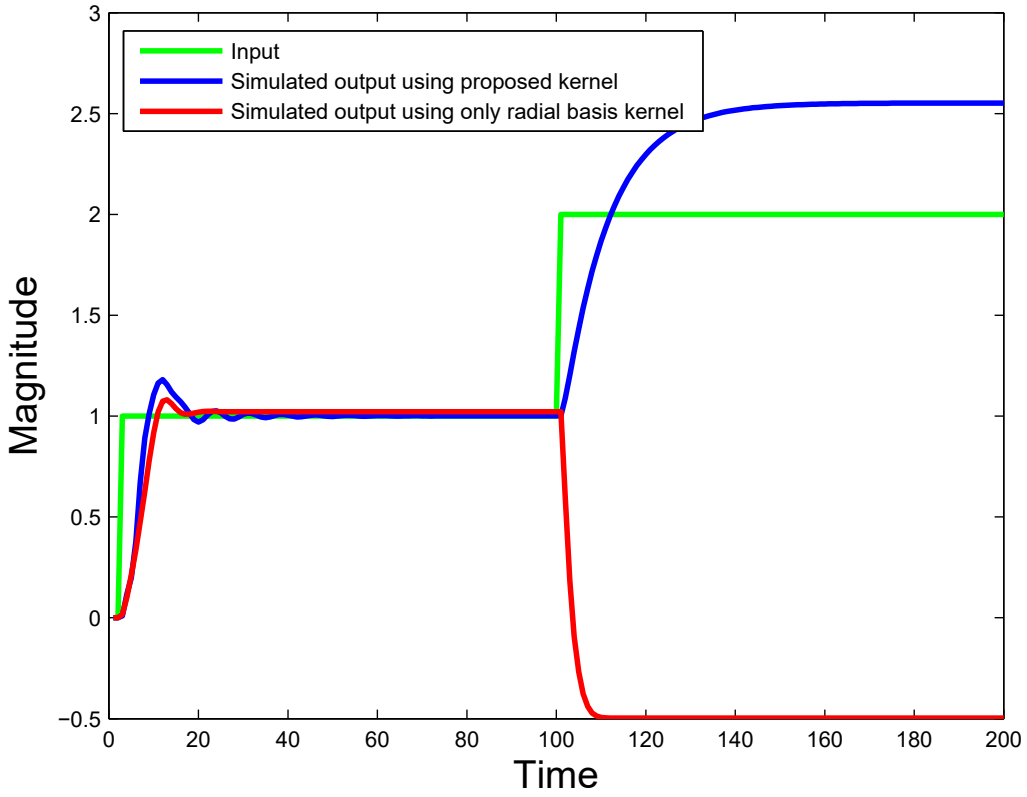


Figure 4.15: Example to show extrapolation from GP models as well as effect of choice of kernels. The blue simulated response is from robust GP model with kernel given in Equation 4.12 whereas the red simulated response is from robust GP model with kernel given in Equation 2.30.

4.6.3 Computation time

Use of GP models in the proposed optimization framework is computationally demanding. This is because it introduces nonlinear constraints. Thus, it must be ensured that GP models are used specifically in the case of processes which involve high degree of nonlinearity. Since GP models are completely data based, large size of training data set can also slow down the optimization. In such cases use of sparse Gaussian process models can yield significant improvement. Several works have focused building sparse GP regression models [32, 54].

4.7 Conclusion

In this chapter, a novel optimization framework was proposed and implemented to find a set point changing solution for the water treatment and steam generator network. Two different process identification methods were tested. Linear discrete transfer function models and robust GP regression models were explored. Some unique aspects related to GP modeling for system identification were discussed. A third "ramp up" based set point changing strategy was also presented. It was found that the proposed method can give reasonable solutions for optimization of a network of units. Results from this method can be improved by realistic constraints on the set point changing strategy. Moreover, total produced water flow rate entering the system may not be fixed during the period of transition. Uncertainties associated with such constraints can be included into the proposed method and solved using robust optimization methods.

Chapter 5

Conclusion

5.1 Summary of thesis

This thesis is concerned with identification of robust GP regression model using EM algorithm. Dynamic nonlinear process models identified using proposed approach are used in optimization and control of an industrial process.

The challenges associated with the use of data-driven models in chemical engineering optimization problems motivated us to explore Gaussian process regression. This is explained in more detail in Chapter 1.

Chapter 2 contains the proposed method for identifying robust GP models. The effect of outliers on regression is curtailed by the use of heavy tailed noise distributions such as t-distribution and Laplace distribution. An EM based approach was used to estimate the hyper-parameters for the Gaussian process prior as well as noise distribution. Another EM based approach known as Expectation Conjugate Gradient (ECG) algorithm was derived and implemented. The proposed methods were then successfully applied on simulation as well as real datasets. A soft sensor regression problem using NIR spectroscopy data was also solved using the proposed methods. Finally, a detailed comparison was made between EM based and conjugate gradient based parameter estimation techniques. Through these discussions, it can be concluded that EM algorithm is often easier to implement and numerically stable in comparison to gradient based methods. Moreover, it comes with certain convergence guarantees which make more attractive.

In Chapter 3 an optimization problem was formulated based on a SAGD water treatment network. Process models were constructed for all the units and used for

data reconciliation. Steady state optimization was performed for a fixed produced water input to the network and the optimal cost of production was compared against historically achieved cost of production.

Next, in Chapter 4 a novel set point change strategy was proposed for achieving optimal cost of production in the network. This approach was formulated as an optimization problem involving dynamic process models. The robust GP regression technique described in Chapter 2 was used for modeling process dynamics of water treatment units. The factors to be considered while identifying a nonlinear process model using GP regresson are discussed. Results suggest that robust GP models can be used in process optimization problems.

5.2 Future work

Although GP models offer a powerful tool for regression and nonlinear system identification, there are several other issues which must be overcome before they can be used in a wider range of practical applications. The areas for improvement regarding application of GP regression are listed below:

1. A well thought out choice of kernel function for designing the GP prior can be instrumental in improving the performance of GP regression problems. For example, if a relationship is known to contain a linear or periodic component, it can be included in the kernel function. Recent works in machine learning literature have focused on automatic pattern discovery and extrapolation in GP models. In one paper this has been achieved by the use of the so called spectral mixture kernels [55]. Such techniques can be useful in chemical engineering applications.

2. In MPC type of applications, such as the one discussed in Chapter 4, use of a reliable simulator model can be advantageous. Table 4.2 shows how one-step ahead predictions from robust GP models are better than simulation. This is because the strategy proposed in this work involves minimization of one-step ahead predictions from robust GP models. The remedy is to try and extend the proposed robust GP identification method to output-error models.

Some areas for improvement regarding the optimization of the water treatment and steam generator network are as follows:

1. The proposed set point change strategy basically relies on simulation of process models using dynamic equations. There is no feedback taken from the actual plant measurements. This could cause problems if the measurements do not follow the simulation trend. Use of a feedback mechanism can help design a model predictive control kind of operation strategy.

2. Cost parameters in the steady state objective function used in this thesis may vary over a historical range of values. Other parameters appearing in constraints can also have an uncertainty associated with their values. The proposed steady state optimization problem can be improved by implementing a robust optimization strategy which takes such uncertainties into account.

Bibliography

[1] David M. Himmelblau. Accounts of experiences in the application of artificial neural networks in chemical engineering. *Industrial & Engineering Chemistry Research*, 47(16):5782–5796, 2008.

[2] Enrique Arce-Medina and Jos I. Paz-Paredes. Artificial neural network modeling techniques applied to the hydrodesulfurization process. *Mathematical and Computer Modelling*, 49(12):207 – 214, 2009.

[3] Hiromasa Kaneko and Kimito Funatsu. Application of online support vector regression for soft sensors. *AIChE Journal*, 60(2):600–612, 2014.

[4] XinJiang Lu, Han-Xiong Li, Ji-An Duan, and Dong Sun. Integrated design and control under uncertainty: A fuzzy modeling approach. *Industrial & Engineering Chemistry Research*, 49(3):1312–1324, 2010.

[5] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[6] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[7] Carl Edward Rasmussen. Evaluation of gaussian processes and other methods for non-linear regression. Technical report, 1996.

[8] Georges Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

[9] D. G. Krige. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society*, 52:119–139, 1951.

[10] Michael Kemmler, Erik Rodner, Esther-Sabrina Wacker, and Joachim Denzler. One-class classification with gaussian processes. *Pattern Recognition*, 46(12):3507 – 3518, 2013.

[11] Yingchao Xiao, Huangang Wang, and Wenli Xu. Hyperparameter selection for gaussian process one-class classification. *Neural Networks and Learning Systems, IEEE Transactions on*, 26(9):2182–2187, 2015.

[12] Yakoub Bazi and Farid Melgani. Gaussian process approach to remote sensing image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(1):186–197, 2010.

[13] Gholamreza Salimi-Khorshidi, Thomas E Nichols, Stephen M Smith, and Mark W Woolrich. Using gaussian-process regression for meta-analytic neuroimaging inference based on sparse observations. *Medical Imaging, IEEE Transactions on*, 30(7):1401–1416, 2011.

[14] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, 1998.

[15] Fani Boukouvala and Marianthi G. Ierapetritou. Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function. *AIChE Journal*, 60(7):2462–2474, 2014.

[16] 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.

[17] Y. Liu, T. Chen, and J. Chen. Auto-switch gaussian process regression-based probabilistic soft sensors for industrial multigrade processes with transitions. *Industrial and Engineering Chemistry Research*, 54(18):5037–5047, 2015.

[18] Lester Lik Teck Chan, Yi Liu, and Junghui Chen. Nonlinear system identification with selective recursive gaussian process models. *Industrial & Engineering Chemistry Research*, 52(51):18276–18286, 2013.

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

[20] Robert B. Gramacy and Herbert K. H. Lee. Optimization under unknown constraints. *arXiv preprint arXiv:1004.4027*, 2010.

[21] 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.

[22] J. Kocijan, R. Murray-Smith, CE. Rasmussen, and B. Likar. Predictive control with gaussian process models. pages 352–356. Max-Planck-Gesellschaft, 2003.

[23] Xing Jin and Biao Huang. Robust identification of piecewise/switching autoregressive exogenous process. *AIChE Journal*, 56(7):1829–1844, 2010.

[24] 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.

[25] Terry E. Dielman. Least absolute value regression: recent contributions. *Journal of Statistical Computation and Simulation*, 75(4):263–286, 2005.

[26] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[27] M. Kuss. *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. PhD thesis, Technische Universität Darmstadt, 2006.

[28] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[29] Thomas Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 362–369, 2001.

[30] Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *J. Mach. Learn. Res.*, 12:3227–3257, 2011.

[31] Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari. Gaussian process regression with student-t likelihood. In *Advances in Neural Information Processing Systems 22*, pages 1910–1918. 2009.

[32] M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003.

[33] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian gaussian process classification with the EM-EP algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1948–1959, 2006.

[34] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *J. Mach. Learn. Res.*, 6:1019–1041, 2005.

[35] Richard A. Redner and Homer F. Walker. Mixture Densities, Maximum Likelihood and the Em Algorithm. *SIAM Review*, 26:195–239, 1984.

[36] Mortaza Jamshidian and Robert I. Jennrich. Acceleration of the em algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59:pp. 569–587, 1997.

[37] Meng and Van Dyk. Fast em-type implementations for mixed effects models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60:559–578, 1998.

[38] Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. Optimization with em and expectation-conjugate-gradient. In *Proceedings of the International Conference on Machine Learning*, volume 20, pages 672–679, 2003.

[39] CE. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

[40] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[41] Radford M. Neal and Geoffrey E. Hinton. In *Learning in Graphical Models*, chapter A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants, pages 355–368. MIT Press, 1999.

[42] R. M. Neal. Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. *ArXiv Physics e-prints*, 1997.

[43] Jerome H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–67, 1991.

[44] David Harrison Jr. and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81 – 102, 1978.

[45] Tecator. Tecator dataset, available at http://lib.stat.cmu.edu/datasets/tecator. 1992.

[46] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1995.

[47] Vicki Lightbown. New sagd technologies show promise in reducing environmental impact of oil sand production. *Journal of Environmental Solutions for Oil, Gas, and Mining*, 1(1):47–58, 2015.

[48] W Heins, D Peterson, et al. Use of evaporation for heavy oil produced water treatment. *Journal of Canadian Petroleum Technology*, 44(01):26–30, 2005.

[49] Cameron M Crowe. Data reconciliationprogress and challenges. *Journal of Process Control*, 6(2):89–98, 1996.

[50] Tyler A. Soderstrom, Thomas F. Edgar, Louis P. Russo, and Robert E. Young. Industrial application of a large-scale dynamic data reconciliation strategy. *Industrial & Engineering Chemistry Research*, 39(6):1683–1693, 2000.

[51] Lorenz T. Biegler and Ignacio E. Grossmann. Retrospective on optimization. *Computers and Chemical Engineering*, 28:1169–1192, 2004.

[52] Mark L. Darby, Michael Nikolau, James Jones, and Doug Nicholson. Rto: An overview and assessment of current practice. *Journal of Process Control*, 21(6):874 – 884, 2011.

[53] Gregor Gregorcic and Gordon Lightbody. Gaussian processes for modelling of dynamic non-linear systems. *Proceedings of the Irish Signals and Systems Conference*, pages 141–147, 2002.

[54] Matthias Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14:69–106, 2004.

[55] Andrew Gordon Wilson. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

$\mathcal{Q}$ function expressions

As seen in Equation 2.14, the $\mathcal{Q}$ function is composed of $\mathcal{Q}_{cov}$ and $\mathcal{Q}_e$. Complete expressions for these terms are as follows:

$\mathcal{Q}_{cov}$ function

The expression for $\mathcal{Q}_{cov}$ is as follows [33]:

$$\begin{aligned}\mathcal{Q}_{cov} &= \mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log p(\mathbf{f}|\mathbf{X},\theta_{cov})\right] \\ &= -\frac{1}{2}\log|2\pi\mathbf{K}| - \frac{1}{2}\mathbf{E}\left[\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\right] \\ &= -\frac{1}{2}\log|2\pi\mathbf{K}| - \frac{1}{2}\mathbf{E}[\mathbf{f}^T]\mathbf{K}^{-1}\mathbf{E}[\mathbf{f}] - \frac{1}{2}tr\left(\mathbf{K}^{-1}cov\left(\mathbf{f}\right)\right) \\ &= -\frac{1}{2}\log|2\pi\mathbf{K}| - \frac{1}{2}\mathbf{h}^T\mathbf{K}^{-1}\mathbf{h} - \frac{1}{2}tr\left(\mathbf{K}^{-1}\mathbf{A}\right)\end{aligned} \tag{1}$$

The above expression involves finding the inverse of $\mathbf{K}$ which can be ill-conditioned. Cholesky decomposition was used to find $\mathbf{K}^{-1}$ since it is a more numerically stable technique. Apart from this, a small amount of nugget was added to the diagonal of $\mathbf{K}$ to improve the conditioning. Fixing a value for this nugget was avoided by using the white noise term in the kernel as described in the regression results section. The magnitude of this term was considered to be a variable and learnt using the proposed EM scheme along with other hyper-parameters.

The values for $\mathbf{h}$ and $\mathbf{A}$ were obtained by the posterior approximation techniques discussed in the chapter, viz., Laplace approximation or EP approximation.

$\mathcal{Q}_e$ function for t-distribution case

$\mathcal{Q}_e^l$ for Student's t-likelihood is given in Equation 2.19. The hyper-paramaters in this case are $\theta_e = [\nu, \sigma]$.

$$\begin{aligned}
\mathcal{Q}(\theta_e|\Theta^t)^l &= n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[1+\frac{(y_i-f_i)^2}{\nu\sigma^2}\right] \\
&= n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\left(1+\frac{(y_i^2+h_i^2-2y_ih_i+A_{ii})}{\nu\sigma^2}\right) \\
&= n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\left(1+\frac{s_i^2}{\nu\sigma^2}\right)
\end{aligned} \tag{2}$$

Where $s_i^2 = (y_i^2 + h_i^2 - 2y_ih_i + A_{ii})$.

$\mathcal{Q}_e$ function for Laplace distribution case

In the case of Laplace likelihood, the noise hyper-parameter is $\theta_e = s$. Here, exact $\mathcal{Q}_e$ expression can be evaluated by substituting Equation 2.22 in Equation 2.23. This gives the following:

$$\begin{aligned}
\mathcal{Q}(\theta_e|\Theta^t) &= -n\log(2s) - \frac{1}{s}\sum_{i=1}^{n}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[|y_i-f_i|\right] \\
&= -n\log(2s) - \frac{1}{2}\sum_{i=1}^{n}\left((y_i-h_i)\left[2\Phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)-1\right] + 2\sqrt{A_{ii}}\left[\phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)\right]\right)
\end{aligned} \tag{3}$$

Where Φ is standard normal cumulative density function and ϕ is standard normal probability density function.

$\mathcal{Q}$ function derivatives

$\mathcal{Q}_{cov}$ derivative

The derivative of $\mathcal{Q}_{cov}$ with respect to covariance function hyper-parameters θ_{cov} can be found as follows:

$$\begin{aligned}\frac{\partial}{\partial\theta_{cov}}\mathcal{Q}(\theta_{cov}|\Theta^t) &= \frac{\partial}{\partial\theta_e}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[\log p(\mathbf{f}|\mathbf{X},\theta_{cov})\right] \\ &= \frac{\partial}{\partial\theta_{cov}}\left(-\frac{1}{2}\log|2\pi\mathbf{K}| - \frac{1}{2}\mathbf{h}^T\mathbf{K}^{-1}\mathbf{h} - \frac{1}{2}tr(\mathbf{K}^{-1}\mathbf{A})\right) \\ &= -\frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\right) + \frac{1}{2}\mathbf{h}^T\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K}^{-1}\mathbf{h} + \frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K}^{-1}\mathbf{A}\right)\end{aligned} \tag{4}$$

$\mathcal{Q}_e$ derivative for Student's t-likelihood

In the case of Student's t-likelihood, $\mathcal{Q}_e^l$ is evaluated instead of $\mathcal{Q}_e$. The expression for $\mathcal{Q}_e^l$ is given in Equation 2.19. Using this expression the derivative was computed as follows

$$\frac{\partial}{\partial\theta_e}\mathcal{Q}(\theta_e|\Theta^t)^l = \frac{\partial}{\partial\theta_e}\left(n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[1+\frac{(y_i-f_i)^2}{\nu\sigma^2}\right]\right) \tag{5}$$

For ν and σ the respective derivative expression are given by

$$\begin{aligned}
&\frac{\partial}{\partial \theta_e}\mathcal{Q}(\theta_e|\Theta^t)^l \\
&= \frac{\partial}{\partial \theta_e}\left(n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\left(1+\frac{(y_i^2+h_i^2-2y_ih_i+A_{ii})}{\nu\sigma^2}\right)\right) \\
&= \frac{\partial}{\partial \theta_e}\left(n\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\log\left(1+\frac{s_i^2}{\nu\sigma^2}\right)\right)
\end{aligned} \tag{6}$$

Where $s_i^2 = (y_i^2 + h_i^2 - 2y_ih_i + A_{ii})$. Therefore derivatives with respect to ν and σ are given by

$$\begin{aligned}
&\frac{\partial}{\partial \nu}\mathcal{Q}(\theta_e|\Theta^t)^l = \\
&\frac{n}{2}\psi(\frac{\nu+1}{2}) - \frac{n}{2}\psi(\frac{\nu}{2}) - \frac{n}{2\nu} - \frac{1}{2}\sum_{i=1}^{n}\left(\log\left(1+\frac{s_i^2}{\nu\sigma^2}\right)\right) - \frac{\nu+1}{2}\sum_{i=1}^{n}\frac{-s_i^2}{\nu^2\sigma^2\left(1+\frac{s_i^2}{\nu\sigma^2}\right)}
\end{aligned} \tag{7}$$

and

$$\frac{\partial}{\partial \sigma}\mathcal{Q}(\theta_e|\Theta^t)^l = -\frac{n}{\sigma} - \frac{\nu+1}{2}\sum_{i=1}^{n}\frac{-2s_i^2}{\nu\sigma^3\left(1+\frac{s_i^2}{\nu\sigma^2}\right)} \tag{8}$$

$\mathcal{Q}_e$ derivative for Laplace likelihood

The complete expression for $\mathcal{Q}_e$ for Laplace likelihood is given in Equation 3. Using it the derivative can be found as follows:

$$\frac{\partial}{\partial s}\mathcal{Q}(\theta_e|\Theta^t) = \frac{\partial}{\partial s}\left(-n\log(2s) - \frac{1}{s}\sum_{i=1}^{n}\mathbf{E}_{\mathbf{f}|\mathbf{h},\mathbf{A}}\left[|y_i - f_i|\right]\right) \tag{9}$$

$$\begin{aligned}
&= \frac{\partial}{\partial s}\left(-n\log(2s) - \frac{1}{s}\sum_{i=1}^{n}\left[(y_i-h_i)\left(2\Phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)-1\right) + 2\sqrt{A_{ii}}\phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)\right]\right) \\
&= -\frac{n}{s} + \frac{1}{s^2}\sum_{i=1}^{n}\left[(y_i-h_i)\left(2\Phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)-1\right) + 2\sqrt{A_{ii}}\phi\left(\frac{y_i-h_i}{\sqrt{A_{ii}}}\right)\right]
\end{aligned} \tag{10}$$

Simplified $\mathcal{Q}_{COV}$ function derivative for ECG

The $\mathcal{Q}_{cov}$ function derivative can be further simplified in the case of ECG. This is because for ECG algorithm the value of derivative is required only at Θ^t, i.e. $\frac{\partial}{\partial\theta_{cov}}\mathcal{Q}_{cov}\Big|_{\Theta^t}$.
Recall that the expression for $\mathcal{Q}_{cov}$ at each iteration requires expectation with respect to posterior distribution $p(\mathbf{f}|\mathbf{X},\mathbf{y},\Theta^t)$ where Θ^t is the value of hyper-parameters at t^{th} iteration. The approximation for the posterior distribution $q(\mathbf{h})$ which is used in ECG for Laplace noise distribution was obtained from the gpml toolbox which admits the form:

$$q(\mathbf{f}) = \mathcal{N}\left(\mathbf{f}|\mathbf{h}=\mathbf{K}\alpha, \mathbf{A}=(\mathbf{K}^{-1}+\mathbf{W})^{-1}\right) \tag{11}$$

where $\mathbf{W}$ is a diagonal matrix with $W_{ii} = s_i^2$. The EP approximation function in gpml toolbox returns α, $\mathbf{W}^{1/2}$ as well as $\mathbf{L} = chol(\mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2}+\mathbf{I})$. "chol" refers to Cholesky decomposition. Using $\alpha, \mathbf{W}^{1/2}, \mathbf{L}$ the expression in Equation 1 can be simplified as shown in Equation 12. Since, $\mathbf{h} = \mathbf{K}\alpha$ and $\mathbf{A} = (\mathbf{K}^{-1}+\mathbf{W})^{-1}$, we get,

$$\begin{aligned}
\frac{\partial}{\partial\theta_{cov}}\mathcal{Q}_{cov}\Big|_{\Theta^t} &= -\frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\right) + \frac{1}{2}\mathbf{h}^T\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K}^{-1}\mathbf{h} + \frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K}^{-1}\mathbf{A}\right) \\
&= -\frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\right) + \frac{1}{2}\alpha^T\frac{\partial\mathbf{K}}{\partial\theta}\alpha + \frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\mathbf{K}^{-1}(\mathbf{K}^{-1}+\mathbf{W})^{-1}\right) \\
&= \frac{1}{2}\alpha^T\frac{\partial\mathbf{K}}{\partial\theta}\alpha - \frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\left(\mathbf{I}-\mathbf{K}^{-1}(\mathbf{K}^{-1}+\mathbf{W})^{-1}\right)\right) \\
&= \frac{1}{2}\alpha^T\frac{\partial\mathbf{K}}{\partial\theta}\alpha - \frac{1}{2}tr\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta}\left(\mathbf{I}-\mathbf{K}^{-1}(\mathbf{K}^{-1}+\mathbf{W}^{1/2}\mathbf{W}^{1/2})^{-1}\right)\right)
\end{aligned} \tag{12}$$

Using the Woodbury identity,

$$
\begin{aligned}
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta}\left(\mathbf{I} - \mathbf{K}^{-1}\left(\mathbf{K} - \mathbf{K}\mathbf{W}^{1/2}\left(\mathbf{I} + \mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2}\right)^{-1}\mathbf{W}^{1/2}\mathbf{K}\right)\right)\right) \\
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{W}^{1/2}\left(\mathbf{I} + \mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2}\right)^{-1}\mathbf{W}^{1/2}\mathbf{K}\right)
\end{aligned}
\tag{13}
$$

Using $\mathbf{L} = chol(\mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2} + \mathbf{I})$ the above expression can be further simplified to give,

$$
\begin{aligned}
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{W}^{1/2}\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{W}^{1/2}\mathbf{K}\right) \\
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{W}^{1/2}\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{W}^{1/2}\right) \\
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\frac{\partial \mathbf{K}}{\partial \theta}(\mathbf{L}^{-1}\mathbf{W}^{1/2})^T(\mathbf{L}^{-1}\mathbf{W}^{1/2})\right) \\
&= \frac{1}{2}\alpha^T \frac{\partial \mathbf{K}}{\partial \theta}\alpha - \frac{1}{2} tr\left(\frac{\partial \mathbf{K}}{\partial \theta}(\mathbf{M})^T\mathbf{M}\right)
\end{aligned}
\tag{14}
$$

where $\mathbf{M} = \mathbf{L}^{-1}\mathbf{W}^{1/2}$. The final expression which involves less computations in comparison to Equation 1, uses only α, $\mathbf{W}^{1/2}$ and $\mathbf{L}$ which are obtained using EP approximation function in gpml toolbox.