Robust Generalized Weighted Probabilistic Principal Component Regression with Application in Data-driven Optimization

by

Alireza Memarian

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

The operations of the plant may deviate from the initial design due to the uncertainties and changes in the several conditions as a result of market demand, operation conditions, and safety regulations over time. To maintain productivity, safety, and efficiency, operators should ensure the plant to be operating around its optimal point. However, due to the changes in the operating conditions of the plant, the current optimal point may deviate from the one obtained during the initial design. Alongside finding the optimal point, it is essential to find the optimal path that steers the plant from the current operating conditions to the optimal operating point. Hence, automated self-optimization of the plants is gaining popularity in academia and industry. One of the approaches that is in practice in plant optimization is optimizing the plant with the aid of the model. Thus, developing a model that can mimic the plant with the utmost accuracy is important. However, due to the possible differences between the developed model and the plant (model-plant mismatch), the obtained optimal point from the model may not be accurate. The main objective of this thesis is to develop a general framework for optimization of a plant that can handle the model-plant mismatch. A model-based optimization strategy is utilized to achieve this objective.

To develop a model that is robust to outliers, and can handle delays, missing data in input and output, and also is simple to use in plant optimization, two extensions of a generalized weighted probabilistic principal component regression method are proposed in this thesis. In addition, the proposed model is able to deal with high-dimensional plant datasets, multi-modal and/or nonlinear nature of the plants.

The high dimensionality, multi-modal nature of plants, missing data in input and output variables, and outliers are addressed simultaneously in Chapter 2, the mixture robust semi-supervised probabilistic principal component regression model with missing input data. The main challenge with the model developed in Chapter 2 is to determine the optimal number of mixture components to be used while modeling. In Chapter 3 entitled weighted semi-supervised probabilistic principal component regression with missing input and delayed output variables, challenges like the delay between each input and output variable and missing data are addressed. These extensions are developed under the expectation maximization (EM) framework owing to the fact that they can efficiently deal with hidden variables like missing data, delays, and outliers. To account for the missing input and output data in these models, the data imputation method and semi-supervised framework are utilized, respectively. To deal with the presence of outliers, a combination of two Gaussian distributions is used as a prior for the noise, and a model-free distribution is considered for the delay variables. Finally, a strategy to update the range of delay in the variables is proposed to help speeding up the convergence of the algorithm.

A combination of these two proposed algorithms is capable of making the most use of all available information and address uncertainties that may occur in plants. Therefore, by incorporating the proposed extensions of the PPCR model together, a generalized weighted PPCR model is developed to describe the plant, which is able to deal with different types of uncertainties while performing the plant optimization. To account for the model-plant mismatch between the generalized weighted PPCR model and the plant in addition to steering the solution closer to the plant's optimal point, a robust Gaussian process regression model is utilized. To increase the accuracy of the generalized weighted PPCR model, a nonlinearity index is proposed that defines the range of the data to be used while developing a model. The proposed algorithm builds a local model around the current operating point and tries to find its optimal point by solving the optimization problem, and then steer the plant to the obtained optimal solution. By repeating these two steps, i.e. 1) building a local model and 2) steering the plant to the obtained optimal point, the algorithm tries to gradually move the plant from its initial operating point to the optimal point. Finally, the applicability and performance of all the proposed methods are tested and demonstrated through several numerical, simulation, experimental, and industrial examples.

To my family, whom I missed so much and supported me in this challenging path

Acknowledgments

First and foremost, I should appreciate my supervisor Professor Biao Huang who gave me this opportunity to continue my studies in his research group and follow my interests. He helped me patiently and believed in me on the way to the destination through his constructive suggestions. I really appreciate him for his supportive attitude and inspiration during my studies. It was a great pleasure for me to investigate my graduate studies in M.Sc. under his supervision.

Next, I should appreciate my friend Dr. Santhosh Kumar Varanasi, a postdoctoral fellow in our research group, for his indefinite help during these two years. He was beside me from the beginning of my research, and he was always there to help me and discuss various topics regarding the research. Not only had he several discussions with me about the research topics, but also he taught me how to talk and write in an academic way.

It was my honor to be a member of the Computer Process Control (CPC) group, where we broadened our knowledge not only in our research but also in the industry, and not hesitating to join discussions and express our thoughts. Here, I like to express my gratitude to all my colleagues in this group, specifically those who joined with me in September 2019, Kiran Raviprakash, Yousef Salehi, Vamsi Krishna Puli, Chaitanya Manchikatla. Moreover, I should appreciate my friends who helped me a lot during these two years by supporting me emotionally or making discussions regarding my research like Anudari Khosbayar, Arun Senthil Sundaramoorthy, Ranjith Chiplunkar, Mengqi Fang, Hongtian Chen, Yousef Alipouri, Alireza Kheradmand, Anahita Sadeghian, and others for their help and support.

I would like to acknowledge the Department of Chemical and Materials Engineering and the University of Alberta for giving me this opportunity to continue my studies in Masters by providing a good and pleasant environment. In addition, I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada for their financial support that made it possible for me to study in this prestigious university and research group.

Last but not least, I would like to thank my family for all their support emotionally and financially. They encouraged me to pursue my dreams and studies and believe in my abilities. I would like to thank my brother Amirreza, who helped me in every step of this adventure and supported me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Processes are designed to operate optimally and safely; however, due to process disturbances and changes in plant conditions, process operations frequently deviate from the initial design. Thus, developing an online framework to optimize the plant along with finding the optimal path for steering the plant from its current operating point to the optimal point is desirable. Many methods have been developed for optimizing the plants and finding the optimal condition of the processes [3, 4, 5]. Real-time optimization (RTO) is one of the solutions. Extensive studies have been conducted about the RTO that relies on a first principle model [6, 7]. The first principle model-based RTO requires an in-depth understanding of the process through the governing equations, which may not always be available. On the other hand, data-driven RTO utilizes the available data to model the plant and is gaining popularity.

One of the main challenges with the data-driven models lies in its ability to simultaneously deal with different uncertainties like outliers, missing data, and delay alongside dealing with the nonlinearity and/or multi-modal nature of the plant [8, 9, 10]. In addition, in applications like RTO, the data-driven model should also be able to deal with high-dimensional datasets. Due to the differences between the model and the plant, the optimal solution obtained by solving the RTO will be different from the true optimal point of the plant. Therefore, it is essential to consider the differences termed as a model-plant mismatch, while solving the optimization problem, which is one of the main objectives of the thesis. Further, the proposed framework should be able to provide an optimal solution. Another objective of this thesis is to address the aforementioned challenges by developing an online data-driven modeling framework.

1.2 General literature review

Self-optimization aims to optimize the process and make process operations efficient and profitable [11]. Optimizing a process can be performed based on the development of a model, which is generally obtained through two different approaches, i) first principle model-based optimization and ii) data-driven model-based optimization [12, 13]. In the first principle model-based optimization, the plant is modeled with the help of deriving the governing equations from the fundamental laws, which needs an in-depth understanding of the plant [14]. On the other hand, in data-driven model-based optimization, a model is built based on the historical data. Therefore, neural networks, being simple and efficient, are utilized as a data-driven model in [15]. Moreover, in [15], authors applied recurrent neural networks to solve the optimization problem. However, their developed model is not able to consider data uncertainty like missing data, outliers, and time delay. Recently, authors in [16] investigate the application of derivative-free optimization in the data-driven optimization framework. They compared the performance of different model-based methods on several chemical engineering benchmarks. Authors in [17] proposed a time-varying extremum-seeking control (ESC) approach for discrete-time systems that tries to find the local optimum in convex functions through a model-free manner. One of the main drawbacks of method is utilizing gradient information that requires a process knowledge. Hence, in [18], authors proposed using GPR and Bayesian optimization to accelerate self-optimization of ESC by utilizing an expected improvement (EI) acquisition function.

In all the aforementioned self-optimization methods, obtaining a model that can handle various uncertainties like missing data in both input and output variables, outliers, and time-delay has not been considered. Further, the acquisition functions are only considered in the objective function for improving the optimal solution, and exploring optimization constraints has not been addressed. In this thesis, a data-driven self-optimization to handle various uncertainties in the presence of model-plant mismatch is proposed that can reduce the possibility of getting into local optimal points. The detailed literature review of data-driven modeling and data uncertainty will be provided in Chapters 2 and 3. The detailed literature review of self-optimization algorithm will be provided in Chapter 4.

1.3 Thesis outline and contributions

The rest of this thesis is organized as follows:

In Chapter 2, the mixture robust semi-supervised probabilistic principal component regression with missing input data is developed to deal with the high dimensionality in the plant datasets. The proposed method is able to tackle the multi-modal nature of the process, scaled outliers with varying properties across different input and output variables, and simultaneously handle the missing data in each input and output variable. The method is developed based on the expectation maximization (EM) algorithm owing to its ability to efficiently deal with hidden variables like missing data in input variables, latent variables, outliers, models. The EM algorithm provides a maximum likelihood estimation of the parameters by iteratively updating the estimated values of the hidden variables and the parameters of the model. The proposed model enables each input and output variables to have scaled outliers with different properties that are more common in industrial processes. Finally, a numerical and an experimental case study are provided that validate the performance of the proposed method.

Although the proposed method in Chapter 2 is able to deal with outliers with different properties, determining the number of the mixture components can be challenging. In addition, the problem of delays between input and output variables has not been addressed in Chapter 2. Therefore, in Chapter 3, a weighted semi-supervised probabilistic principal component regression model in the presence of missing input and delayed output data is proposed. The proposed model provides an online model based on the current query point by assigning weights to the historical data and uses the most relevant data to develop the model. It also deals with time delay in each output variable that is the most common uncertainty in the real processes and directly affects the quality of the models. Moreover, the issue of missing data in input variables is handled through the data imputation method. This model is developed under the framework of just-in-time learning, and the estimation of parameters is performed through the use of the expectation maximization (EM) algorithm. In addition, the EM algorithm allows model parameters to be estimated through the maximum a posteriori (MAP) principle. An update strategy is developed for the range of the delay terms to speed up the convergence of the EM algorithm and providing more accurate estimates for the delays. To verify the applicability of the proposed method and its performance, a numerical example and experimental example of the hybrid tank pilot plant model are provided.

In Chapter 4, a data-driven self-optimization of the plant operations in the presence of model-plant mismatch is proposed. It is an online data-driven framework for plant optimization. A generalized weighted probabilistic principal component regression model that is the combination of the proposed models in Chapter 2 and Chapter 3 is used as a data-driven model while solving the optimization problem. This generalized model is able to model nonlinear and/or multi-modal processes and can deal with the different uncertainties that may occur in datasets like missing data in input and output variables, outliers, and time delay. Due to the possible difference between the generalized weighted PPCR model and the plant (model-plant mismatch), a penalty term in the robust Gaussian process regression model was introduced in the optimization to account for the mismatch and correct the final solution of the optimization problem. To overcome the most common challenge in the optimization problem, i.e., stuck in local optimal points, exploration through the acquisition functions that are commonly used in reinforcement learning and Bayesian optimization is utilized. Finally, the accuracy of the online data-driven optimization is tested by simulation and industrial examples.

In Chapter 5, a conclusion of the thesis and some possible future work for future research is provided.

1.4 Publications

The following contributions are published or submitted for publications/presentations:

1. A. Memarian, S. K. Varanasi, B. Huang. ”Mixture robust semi-supervised probabilistic principal component regression with missing input data”. *Chemometrics and Intelligent Laboratory Systems*, vol. 214,p. 104315, 2021

2. A. Memarian, S. K. Varanasi, B. Huang. ”Soft sensor development in the presence of missing input and delayed output data through weighted semi-supervised probabilistic principal component regression”. Submitted to *IEEE Transactions on Industrial Electronics”*, 2021 (Chapter 3 - Short Version)

3. A. Memarian, S. K. Varanasi, B. Huang. ”Data-Driven Self-Optimization for plant Operations”. Presented in *Canadian Chemical Engineering Conference 2021, October 24-27, Montreal, Quebec, Canada*, 2021 (Chapter 4 - Extended abstract)

4. A.Memarian, S. K. Varanasi, B. Huang. "Data-driven self-optimization of processes in the presence of the model-plant mismatch". Submitted to 13^{th} *IFAC Symposium on Dynamics and Control of Process Systems, (DYCOPS), June 14-27 2022, Busan, Republic of Korea*, 2022 (Chapter 4 - Short Version)

Chapter 2

Mixture robust semi-supervised probabilistic principal component regression with missing input data[1]

2.1 Introduction

Improving efficiency, profitability, and safety are the main objectives of industries [19, 20]. Monitoring and optimal control of processes are essential to achieving these objectives, for which the availability of online measurements is necessary. Online measurements are not always available for several reasons, such as unavailability of measuring devices or measurements obtained only through offline laboratory analysis, which can lead to delays or missing in data samples. A soft sensor is essential for solving these challenges and providing frequent on-line predictions of quality variables.

The predictive models for soft sensors can be derived from either the first principles or data-driven methods. Models derived from first principles use in-depth knowledge of the process, which is not always available. Further, these models may be computationally expensive for online predictions and may not be feasible as a soft sensor model. On the other hand, data-driven models are developed directly from the data and hence, complete understanding of the process is not essential. With the collection of a large amount of data in process industries, data-driven soft sensors are gaining popularity. Data-driven soft sensors can be modeled using different ap-

[1] A. Memarian, S. K. Varanasi, B. Huang. "Mixture robust semi-supervised probabilistic principal component regression with missing input data". *Chemometrics and Intelligent Laboratory Systems*, vol. 214,p. 104315, 2021

proaches such as artificial neural networks (ANNs) [21, 22], support vector machines (SVMs) [23], principal component analysis (PCA) and its regression model extension (PCR) [24].

PCA is a linear modeling method that is mostly used for dimensionality reduction [25] by mapping data into its principal components, which are also called latent variables. PCR is used to build a relationship between input and output variables through regression after extracting the latent variables. Due to its deterministic nature, PCA has some disadvantages, especially when dealing with missing data and outliers, which are common in process industries. These issues are addressed by probabilistic PCA (PPCA) [26, 27], which can only perform well on linear and unimodal data. Most of the industrial processes, however, operate in multiple operating modes where the mapping between system states and measurements is nonlinear, on the whole. Hence, an extension of PPCA to model multiple modes, termed as a mixture PPCA, has been developed [28, 9]. In the mixture PPCA model, a combination of multiple linear models is considered to handle the issue of operating in multiple modes [28, 29].

The problem of missing data while modeling can be handled using two approaches: 1) Neglecting and removing data corresponding to that sampling instant, which can lead to the loss of information, 2) Imputation methods, wherein the missed data is replaced with an estimate. Depending on the method of estimation, there exist several imputation approaches such as mean substitution, regression imputation, and the last observation carried forward (LOCF) [30, 31]. In the framework of a mixture PPCR model, the problem of missing data in output is addressed in [9], wherein the authors developed a mixture semi-supervised PPCR (MSSPPCR) model, which is further extended for the missing data in both inputs and outputs in [8]. In such a framework, the entire dataset is divided into the labeled and unlabeled parts; thereby, a semi-supervised learning strategy is employed. However, the MSSPPCR model developed in [9, 8] cannot handle outliers in the data, which is another critical factor that affect the accuracy of the soft sensor.

Outliers in a dataset are those measurements that deviate from the rest of the data [32], which usually occur due to hardware failure, operator's incorrect recording and transmission issues [33]. Several methods exist in literature wherein, different choices of noise distributions are considered to make the regression robust to outliers [34]. For instance, the authors in [10, 33, 35] considered a mixture of Gaussian distributions for noise measurements. The authors in [36, 37, 38, 39, 40] used student's t-distribution and in [24, 41, 42], Laplace distribution is considered for dealing with outliers. In the framework of MSSPPCR model, the problem of outliers in data is considered in [43], wherein a student's t-distribution is considered for all the variables. Although this framework effectively handles the outliers, the main drawback comes with the assumption that all input and output variables have the same properties regarding outliers. In most of the process industries, outliers might occur in each of the input and output variables with different properties. Therefore, it can lead to information loss when an assumption that all variables being affected by outliers having the same properties while modeling. Hence, it is essential to develop a mixture PPCR model which can simultaneously handle different properties of outliers for each input and output variables, and can also handle their missing data.

This chapter proposes a mixture robust semi-supervised PPCR (MRSSPPCR) model with missing input data in the presence of outliers with different outlier properties among different variables. The proposed approach can handle the multi-modal nature of the data and efficiently handle the missing data in input and output variables along with the outliers. The significance of this chapter comes with the fact of providing flexibility to each input or output variable to have its own outlier properties while simultaneously dealing with missing data problem. Since some of the variables are not observed directly, the approaches like maximum likelihood estimation and maximum-a-posteriori are not tractable [44, 45]. Therefore, Expectation-Maximization (EM) algorithm is utilized owing to the fact that it can approach a maximum likelihood estimation and the estimated values of the missing data can be iteratively updated while updating the parameters of the model [10]. However, the main challenge of this method is its possible convergence to a local optimum. Therefore, Monte Carlo simulations i.e., initialization of algorithm with different values is followed for a better convergence [33].

The remainder of the chapter is organized as follows. In Section 2.2, preliminaries about the PPCR and its extension named MSSPPCR are provided. In Section 2.3, a detailed description of the proposed method is presented. The accuracy of the proposed method is demonstrated through a simulated and an experimental case study in Section 2.4. Finally, the conclusions are drawn in Section 2.5.

2.2 Preliminaries

2.2.1 PPCR model

In this section, the details of PPCR model are presented by considering $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{r\times n}$ to be the input and output datasets, respectively, where n represents the total number of samples, and m and r denote the number of input and output variables, respectively. The PPCR model is derived based on the following generative model.

$$\boldsymbol{x}_i = \boldsymbol{P}\boldsymbol{t}_i + \boldsymbol{e}_i \tag{2.1}$$

$$\boldsymbol{y}_i = \boldsymbol{C}\boldsymbol{t}_i + \boldsymbol{f}_i \tag{2.2}$$

where, $\boldsymbol{x}_i \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{y}_i \in \mathbb{R}^{r\times 1}$ denote the input and output data at i^{th} sampling instant of the datasets $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. $\boldsymbol{P} \in \mathbb{R}^{m\times q}$ and $\boldsymbol{C} \in \mathbb{R}^{r\times q}$ are weighting matrices. $\boldsymbol{t}_i \in \mathbb{R}^{q\times 1}$ is a vector of latent variables, and $\boldsymbol{e}_i \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{f}_i \in \mathbb{R}^{r\times 1}$ are measurement noises of input and output, respectively.

In PPCR, the latent variables and noise measurements of inputs and outputs are assumed to be independent and identically distributed (i.i.d) with Gaussian distribution, i.e., $\boldsymbol{t}_i \sim \mathcal{N}(0, \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{e}_i \sim \mathcal{N}(0, \sigma_x^2\boldsymbol{I})$ and $\boldsymbol{f}_i \sim \mathcal{N}(0, \sigma_y^2\boldsymbol{I})$ where σ_x^2 and σ_y^2 represent the corresponding noise variances. In such a model, the objective is to estimate the parameters, i.e., $\{\boldsymbol{P}, \boldsymbol{C}, \sigma_x^2, \sigma_y^2\}$ by maximizing the likelihood function:

$$L(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{\theta}) = \log p(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{P}, \boldsymbol{C}, \sigma_x^2, \sigma_y^2) = \sum_{i=1}^{n} \log p(\boldsymbol{x}_i, \boldsymbol{y}_i \mid \boldsymbol{P}, \boldsymbol{C}, \sigma_x^2, \sigma_y^2) \tag{2.3}$$

where

$$p(\boldsymbol{x}_i, \boldsymbol{y}_i \mid \boldsymbol{P}, \boldsymbol{C}, \sigma_x^2, \sigma_y^2) = \int p(\boldsymbol{x}_i \mid \boldsymbol{t}_i, \boldsymbol{P}, \sigma_x^2) p(\boldsymbol{y}_i \mid \boldsymbol{t}_i, \boldsymbol{C}, \sigma_y^2) p(\boldsymbol{t}_i) d\boldsymbol{t}_i \tag{2.4}$$

An illustration of the PPCR model is shown in Fig. 2.1. The parameters of the PPCR model can be estimated by using the EM algorithm and a detailed description of the algorithm is given in [28].

2.2.2 MSSPPCR model

In a MSSPPCR model, a total of K individual sub-models are incorporated. In each sub-model (denoted by a variable k), a semi-supervised PPCR with n_1 labeled and

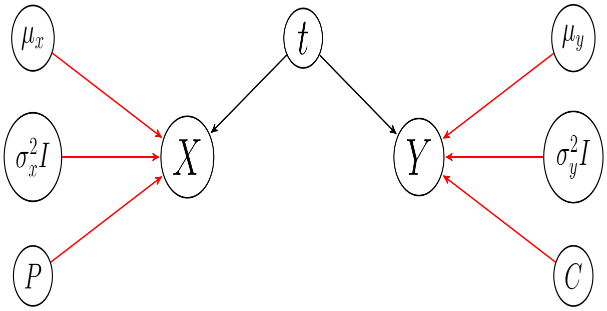


Figure 2.1: Illustration of the PPCR model

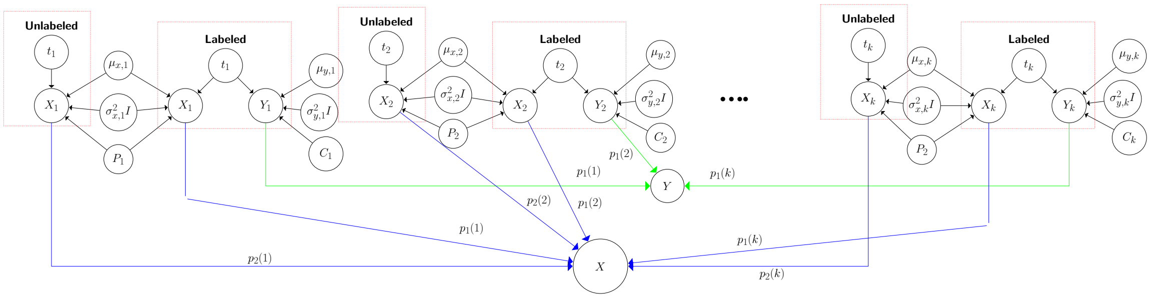


Figure 2.2: Illustration of the MSSPPCR model

n_2 unlabeled data is used. It is to be noted that all the sub-models are independent, and their parameters are different, as shown in Fig. 2.2.

In such a scenario, the model of MSSPPCR can be represented as

$$\boldsymbol{x}_{i,k} = \boldsymbol{P}_k \boldsymbol{t}_{i,k} + \boldsymbol{e}_{i,k} + \boldsymbol{\mu}_{x,k}, \quad k = 1, 2, ..., K \tag{2.5}$$

$$\boldsymbol{y}_{j,k} = \boldsymbol{C}_k \boldsymbol{t}_{j,k} + \boldsymbol{f}_{j,k} + \boldsymbol{\mu}_{y,k}, \quad k = 1, 2, ..., K \tag{2.6}$$

$$\boldsymbol{x}_i = \begin{cases} \sum_{k=1}^{K} \boldsymbol{p}_1(k)\boldsymbol{x}_{i,k} & 1 \leq \text{i} \leq n_1 \\ \sum_{k=1}^{K} \boldsymbol{p}_2(k)\boldsymbol{x}_{i,k} & n_1 + 1 \leq \text{i} \leq n_1 + n_2 \end{cases} \tag{2.7}$$

$$\boldsymbol{y}_j = \sum_{k=1}^{K} \boldsymbol{p}_1(k)\boldsymbol{y}_{j,k} \quad 1 \leq \text{j} \leq n_1 \tag{2.8}$$

where, $\boldsymbol{\mu}_{x,k}$ and $\boldsymbol{\mu}_{y,k}$ are the mean of the input and output measurements in the k^{th} sub-model. In Eq. (2.7), $\boldsymbol{p}_1(k)$ and $\boldsymbol{p}_2(k)$ denote the mixing proportions of the k^{th} sub-model for the labeled and unlabeled data, respectively. Further, these mixing proportions should have the following constraints.

$$\sum_{k=1}^{K} \boldsymbol{p}_1(k) = \sum_{k=1}^{K} \boldsymbol{p}_2(k) = 1 \tag{2.9}$$

In each sub-model, the nature of the parameters and the properties of latent variables and noise measurements are identical with the PPCR model. The main objective is to find the optimal values of the parameters ($\{\boldsymbol{P}_k, \boldsymbol{C}_k, \sigma^2_{x,k}, \sigma^2_{y,k}, \boldsymbol{\mu}_{x,k}, \boldsymbol{\mu}_{y,k}\}$) by maximizing the following likelihood function.

$$L(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{\theta}) = \log p(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{\theta}) = \log p(\boldsymbol{X}_1, \boldsymbol{Y} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{X}_2 \mid \boldsymbol{\theta}) \tag{2.10}$$

$\boldsymbol{X}_1 \in \mathbb{R}^{m \times n_1}$ represents the labeled input dataset, whose corresponding output, i.e., $\boldsymbol{Y} \in \mathbb{R}^{r \times n_1}$ is available; however, $\boldsymbol{X}_2 \in \mathbb{R}^{m \times n_2}$ whose corresponding output is not observed, is the unlabeled dataset in the current formulation. A detailed description of this model and the algorithm for updating the parameters using the EM algorithm is given in [9].

2.3 Development of the MRSSPPCR with missing input data

In this section, a general MRSSPPCR model is developed by considering a Gaussian scaled mixture noise for the input and output data that can identify outliers for each input or output variable independently. Moreover, a semi-supervised learning is used to cope with missing data in output variables and imputation of available input data is used to deal with missing input data. A formulation of mixture model is utilized to handle the multi-modal nature of the process. The parameters in the developed model are estimated through an EM algorithm, and the steps involved in updating the parameters are detailed in the rest of this section.

2.3.1 Model Formulation

For developing the proposed MRSSPPCR model, the following assumptions are made. As a result of changes in set points and/or process drifts, different operating modes appear in a process. The number of modes can sometimes be known from the operator's knowledge. However, in most scenarios, it is an unknown parameter and is required to be defined. One such algorithm for identification of modes is provided in [46]. In the current work, an assumption of number of modes to be given/known is made i.e., the model is assumed to have K operating modes, where K is known/given. The dimension of latent variable in each mode is denoted as q. In such a model, the total number of samples is n, wherein n_1 samples are labeled, and the remaining $n - n_1 (= n_2)$ samples are unlabeled. Finally, the input variables are assumed to have missing values completely at random (MCAR) [47]. The input dataset at time instant i can be partitioned into two sub-vectors as $\boldsymbol{x}_i^T = [\boldsymbol{x}_{i,o}^T, \boldsymbol{x}_{i,m}^T]$. It can be noted that due to the assumption of data being missing completely at random, the dimensions of $\boldsymbol{x}_{i,o}$ and $\boldsymbol{x}_{i,m}$ may vary at each time instant.

The generative model for MRSSPPCR will be the same as the one given in Eqs. (2.5)-(2.8). To make the model robust to outliers, the noise is assumed to follow a mixture of Gaussian distributions with two components in each mode. One component of this distribution has a mean and variance that correspond to the normal data and the second component has the same mean but with a larger variance to account for the outliers in the data. Further, the variance of outliers is inflated with respect to the variance of the normal noise by an inflation factor, $(\boldsymbol{\rho}^{-1})$, which is reflected in a diagonal matrix with all values being constrained within $\rho \in (0, 1]$. It can be noted that the inflation factor is considered in the form of a matrix, instead of a single scalar value as considered in [48], indicating different variables can have different outlier properties. Therefore, $\boldsymbol{\rho}_{x,k_{jj}}$ denotes the outlier level of the j^{th} variable i.e., the j^{th} diagonal element of the matrix $\boldsymbol{\rho}_{x,k}$, where $j = 1, 2, ..., m$. This modification is considered owing to the fact that the former provides an advantage of dealing with outliers of different variances in different variables. Thus, the distribution of input and output noise in each mode in Eqs. (2.5) and (2.6) will be as follows

$$\boldsymbol{e}_{i,k} \sim (1 - \delta_{x,k})\mathcal{N}(0, \sigma_{x,k}^2 \boldsymbol{I}) + \delta_{x,k}\mathcal{N}(0, \boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2 \boldsymbol{I}) \tag{2.11}$$

$$\boldsymbol{f}_{i,k} \sim (1 - \delta_{y,k})\mathcal{N}(0, \sigma_{y,k}^2 \boldsymbol{I}) + \delta_{y,k}\mathcal{N}(0, \boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2 \boldsymbol{I}) \tag{2.12}$$

To differentiate between the outlier data from a normal data, two binary indicators, $q_{x_{i,k}}$ and $q_{y_{i,k}}$, are introduced for input and output variables in each mode, respectively. The property of this binary indicator is such that when $q_{x_{i,k}} = 1$, the input data noise $(\boldsymbol{e}_{i,k})$ corresponds to the distribution of normal data, i.e., $\mathcal{N}(0, \sigma_{x,k}^2 \boldsymbol{I})$ and when $q_{x_{i,k}} = min(\rho_{x,k_{jj}})$, the input data noise corresponds to the distribution of outlier data, i.e., $\mathcal{N}(0, \boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2 \boldsymbol{I})$. A similar definition holds for $q_{y_{i,k}}$. To denote such binary indicator when the inflation factor is a scalar, the authors in [48] utilized a Bernoulli distribution. Inspired by the idea of [48], a Bernoulli distribution for the case when inflation factor is a matrix is provided as follows.

$$p(q_{x_{i,k}} \mid \boldsymbol{\rho}_{x,k}, \delta_{x,k}) = \delta_{x,k}^{\left[1-\left(\prod_{j=1}^{m} \frac{q_{x_{i,k}}-\rho_{x,k_{jj}}}{1-q_{x_{i,k}}\rho_{x,k_{jj}}}\right)\right]} \times (1-\delta_{x,k})^{\left[\prod_{j=1}^{m} \frac{q_{x_{i,k}}-\rho_{x,k_{jj}}}{1-q_{x_{i,k}}\rho_{x,k_{jj}}}\right]} \tag{2.13}$$

$$p(q_{y_{i,k}} \mid \boldsymbol{\rho}_{y,k}, \delta_{y,k}) = \delta_{y,k}^{\left[1-\left(\prod_{j=1}^{r} \frac{q_{y_{i,k}}-\rho_{y,k_{jj}}}{1-q_{y_{i,k}}\rho_{y,k_{jj}}}\right)\right]} \times (1-\delta_{y,k})^{\left[\prod_{j=1}^{r} \frac{q_{y_{i,k}}-\rho_{y,k_{jj}}}{1-q_{y_{i,k}}\rho_{y,k_{jj}}}\right]} \tag{2.14}$$

where, m and r denote the number of input and output variables, respectively. In Eqs. (2.13) and (2.14), $\delta_{x,k}$ and $\delta_{y,k}$ denote the probability by which the input $(\boldsymbol{x}_i)$ and output observation $(\boldsymbol{y}_i)$ follow second component of the distribution, i.e., the probability of an observation is an outlier. The hidden variables in the MRSSPPCR model are $\{X,\ \{\boldsymbol{t}_{i,k}\}_{i=1}^{n},\ \{q_{x_{i,k}}\}_{i=1}^{n},\ \{q_{y_{i,k}}\}_{i=1}^{n_1}$, and $K\}$ and the parameters are as follows.

$$\boldsymbol{\theta} = \{\boldsymbol{P}_k, \boldsymbol{C}_k, \sigma_{x,k}^2, \sigma_{y,k}^2, \boldsymbol{\mu}_{x,k}, \boldsymbol{\mu}_{y,k}, \delta_{y,k}, \delta_{x,k}, \boldsymbol{\rho}_{x,k}, \boldsymbol{\rho}_{y,k}, \boldsymbol{p}_1(k), \boldsymbol{p}_2(k)\}$$

Since the EM algorithm can deal with missing data alongside the hidden variables when used for parameter estimation, it is utilized in the current work. The first step is to build a $Q - function$, i.e., the expectation of the complete data log-likelihood. Since the complete data consists of observed and hidden variables, the resultant $Q - function$ has two components and is given as

$$Q = \\ E_{\boldsymbol{X},\boldsymbol{T}_k,Q_{x,k},Q_{y,k},K|\boldsymbol{X}_o,\boldsymbol{Y},\boldsymbol{\theta}^{old}}\left[\log p(\boldsymbol{X}_1, \boldsymbol{Y}, \boldsymbol{T}_k, Q_{x,k}, Q_{y,k}, k \mid \boldsymbol{\theta}) + \log p(\boldsymbol{X}_2, \boldsymbol{T}_k, Q_{x,k}, Q_{y,k}, k \mid \boldsymbol{\theta})\right] \tag{2.15}$$

Since the noise in input and output variables, latent variables, and input and output sample indicators are assumed to be independent and identically distributed (i.i.d), the terms in the $Q - function$ can be written as

$$\begin{aligned} Q = E_{\boldsymbol{X},\boldsymbol{T}_k,Q_{x,k},Q_{y,k},K|\boldsymbol{X}_o,\boldsymbol{Y},\boldsymbol{\theta}^{old}} & \Bigg(\Bigg[\sum_{i=1}^{n_1} \log p(\boldsymbol{x}_i \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}) + \sum_{i=1}^{n_1} \log p(\boldsymbol{y}_i \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}) \\ & + \sum_{i=1}^{n_1} \log p(\boldsymbol{t}_{i,k} \mid k, \boldsymbol{\theta}) + \sum_{i=1}^{n_1} \log p(q_{x_{i,k}} \mid k, \boldsymbol{\theta}) + \sum_{i=1}^{n_1} \log p(q_{y_{i,k}} \mid k, \boldsymbol{\theta}) + \sum_{i=1}^{n_1} \log \boldsymbol{p}_1(k)\Bigg] \\ & + \Bigg[\sum_{i=n_1+1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}) + \sum_{i=n_1+1}^{n} \log p(\boldsymbol{t}_{i,k} \mid k, \boldsymbol{\theta}) + \sum_{i=n_1+1}^{n} \log p(q_{x_{i,k}} \mid k, \boldsymbol{\theta}) \\ & + \sum_{i=n_1+1}^{n} \log p(q_{y_{i,k}} \mid k, \boldsymbol{\theta}) + \sum_{i=n_1+1}^{n} \log \boldsymbol{p}_2(k)\Bigg]\Bigg) \\ & \triangleq \underbrace{Q_1 + Q_2 + Q_3 + Q_4 + Q_5 + Q_1^{'} + Q_2^{'} + Q_3^{'} + Q_4^{'}}_{E(L_1)} + \underbrace{Q_6}_{E(L_2)} + \overbrace{Q_5^{'}}^{E_(L_3)} \end{aligned} \tag{2.16}$$

where $Q_i,\ i = 1, 2, \cdots, 6$, represent the terms belonging to the labeled part, and $Q_i^{'},\ i = 1, 2, \cdots, 5$, represent the terms belonging to unlabeled part. Since the last two terms in Eq. (2.16), Q_6 and $Q_5^{'}$, are independent of the hidden variables, they can be derived separately by incorporating the constraints given in Eq. (2.9). To derive the posterior probabilities of $\boldsymbol{p}_1(k)$ and $\boldsymbol{p}_2(k)$, it is essential to determine the posterior probabilities of $p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})$, $p(\boldsymbol{x}_{i,o}, \boldsymbol{y}_i \mid k, \boldsymbol{\theta}^{old})$ for labeled dataset and $p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})$, $p(\boldsymbol{x}_{i,o} \mid k, \boldsymbol{\theta}^{old})$ for unlabeled dataset. These terms are estimated using the Bayes rule as follows:

For the labeled part,

$$p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i,o}, \boldsymbol{y}_i \mid k, \boldsymbol{\theta}^{old}) \times p(k \mid \boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i,o}, \boldsymbol{y}_i \mid \boldsymbol{\theta}^{old})} \tag{2.17}$$

$$p(\boldsymbol{x}_{i,o}, \boldsymbol{y}_i \mid k, \boldsymbol{\theta}^{old}) \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{x,k,o}\\ \boldsymbol{\mu}_{y,k}\end{bmatrix}, \begin{bmatrix}\boldsymbol{P}_{k,o}\boldsymbol{P}_{k,o}^T + \sigma_{x,k,o}^2\boldsymbol{I}_o & \boldsymbol{P}_{k,o}\boldsymbol{C}_k^T\\ \boldsymbol{C}_k\boldsymbol{P}_{k,o}^T & \boldsymbol{C}_k\boldsymbol{C}_k^T + \sigma_{y,k}^2\boldsymbol{I}\end{bmatrix}\right) \tag{2.18}$$

and for the unlabeled part,

$$p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i,o} \mid k, \boldsymbol{\theta}^{old}) \times p(k \mid \boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i,o} \mid \boldsymbol{\theta}^{old})} \tag{2.19}$$

$$p(\boldsymbol{x}_{i,o} \mid k, \boldsymbol{\theta}^{old}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x,k,o}, \boldsymbol{P}_{k,o}\boldsymbol{P}_{k,o}^T + \sigma_{x,k,o}^2\boldsymbol{I_o}\right) \tag{2.20}$$

Now the posterior probabilities for mixing proportions are as follows.

$$\begin{aligned}\boldsymbol{p}_1(k) &= \frac{1}{n_1} \times \sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})\\ \boldsymbol{p}_2(k) &= \frac{1}{n - n_1} \times \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})\end{aligned} \tag{2.21}$$

A detailed derivation of Eq. (2.21) is given in appendix A. To calculate $E(L_1)$ in

Eq. (2.16), the terms given in Eqs. (2.22)-(2.26) are essential.

$$p(\boldsymbol{t}_{i,k} \mid k, \boldsymbol{\theta}) \sim \mathcal{N}(0, \boldsymbol{I}) \tag{2.22}$$

$$p(q_{x_{i,k}} \mid k, \boldsymbol{\theta}) \sim \mathcal{B}(1, 1-\delta_{x,k}) \tag{2.23}$$

$$p(q_{y_{i,k}} \mid k, \boldsymbol{\theta}) \sim \mathcal{B}(1, 1-\delta_{y,k}) \tag{2.24}$$

$$p(x_i \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}) \sim \begin{cases} \mathcal{N}(\boldsymbol{P}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{x,k}, \sigma^2_{x,k}\boldsymbol{I}) & q_{x_{i,k}} = 1 \quad q_{y_{i,k}} = 1 \\ \mathcal{N}(\boldsymbol{P}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{x,k}, \boldsymbol{\rho}^{-1}_{x,k}\sigma^2_{x,k}\boldsymbol{I}) & q_{x_{i,k}} = \rho \quad q_{y_{i,k}} = 1 \\ \mathcal{N}(\boldsymbol{P}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{x,k}, \sigma^2_{x,k}\boldsymbol{I}) & q_{x_{i,k}} = 1 \quad q_{y_{i,k}} = \rho \\ \mathcal{N}(\boldsymbol{P}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{x,k}, \boldsymbol{\rho}^{-1}_{x,k}\sigma^2_{x,k}\boldsymbol{I}) & q_{x_{i,k}} = \rho \quad q_{y_{i,k}} = \rho \end{cases} \tag{2.25}$$

$$p(\boldsymbol{y_i} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}) \sim \begin{cases} \mathcal{N}(\boldsymbol{C}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{y,k}, \sigma^2_{y,k}\boldsymbol{I}) & q_{x_{i,k}} = 1 \quad q_{y_{i,k}} = 1 \\ \mathcal{N}(\boldsymbol{C}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{y,k}, \boldsymbol{\rho}^{-1}_{y,k}\sigma^2_{y,k}\boldsymbol{I}) & q_{x_{i,k}} = 1 \quad q_{y_{i,k}} = \rho \\ \mathcal{N}(\boldsymbol{C}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{y,k}, \sigma^2_{y,k}\boldsymbol{I}) & q_{x_{i,k}} = \rho \quad q_{y_{i,k}} = 1 \\ \mathcal{N}(\boldsymbol{C}_k\boldsymbol{t}_{i,k} + \boldsymbol{\mu}_{y,k}, \boldsymbol{\rho}^{-1}_{y,k}\sigma^2_{y,k}\boldsymbol{I}) & q_{x_{i,k}} = \rho \quad q_{y_{i,k}} = \rho \end{cases} \tag{2.26}$$

Since 36 terms arise while expanding the expression, the detailed expression for $E(L_1)$ is provided in appendix B. The parameters are estimated from the derived $Q-function$ (Eq. (2.16)) by taking derivatives in the maximization step and are given as follows

$$\begin{aligned} \boldsymbol{P}_k &: \frac{\partial(Q_1 + Q_1^{'})}{\partial \boldsymbol{P}_k} = 0 \\ \boldsymbol{C}_k &: \frac{\partial(Q_2)}{\partial \boldsymbol{C}_k} = 0 \\ \sigma^2_{x,k} &: \frac{\partial(Q_1 + Q_1^{'})}{\partial \sigma^2_{x,k}} = 0 \\ \sigma^2_{y,k} &: \frac{\partial(Q_2)}{\partial \sigma^2_{y,k}} = 0 \\ \boldsymbol{\mu}_{x,k} &: \frac{\partial(Q_1 + Q_1^{'})}{\partial \boldsymbol{\mu}_{x,k}} = 0 \\ \boldsymbol{\mu}_{y,k} &: \frac{\partial(Q_2)}{\partial \boldsymbol{\mu}_{y,k}} = 0 \\ \delta_{x,k} &: \frac{\partial(Q_4 + Q_3^{'})}{\partial \delta_{x,k}} = 0 \\ \delta_{y,k} &: \frac{\partial(Q_5 + Q_4^{'})}{\partial \delta_{y,k}} = 0 \\ \boldsymbol{\rho}_{x,k} &: \frac{\partial(Q_1 + Q_1^{'})}{\partial \boldsymbol{\rho}_{x,k}} = 0 \\ \boldsymbol{\rho}_{y,k} &: \frac{\partial(Q_2)}{\partial \boldsymbol{\rho}_{y,k}} = 0 \end{aligned} \tag{2.27}$$

Derivatives in Eq. (2.27) are estimated by using expressions given in appendix B and the update of the parameters are as follows (Eqs. (2.28)-(2.37)).

The weighting matrices $\boldsymbol{P}_k$ and $\boldsymbol{C}_k$ can be updated as given in Eqs. (2.28) and (2.29), respectively.

$$\begin{aligned}
P_{k_j} = \Bigg[& \sum_{i=1}^{n_1}(p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})[(2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})P_{1,1}E^T_{1,1})+ \\
& (2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})\rho_{x,k_{jj}}P_{\rho,1}E^T_{\rho,1})+ \\
& (2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})P_{1,\rho}E^T_{1,\rho})+ \\
& (2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})\rho_{x,k_{jj}}P_{\rho,\rho}E^T_{\rho,\rho})])+ \\
& \sum_{i=n_1+1}^{n}(p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})[2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \mu_{x,k_j})_j - P^{'}_{1,1}E^{'T}_{1,1}+ \\
& 2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \mu_{x,k_j})_j - \rho_{x,k_{jj}}P^{'}_{\rho,1}E^{'T}_{\rho,1}+ \\
& 2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \mu_{x,k_j})_j - P^{'}_{1,\rho}E^{'T}_{1,\rho}+ \\
& 2(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \mu_{x,k_j})_j - \rho_{x,k_{jj}}P^{'}_{\rho,\rho}E^{'T}_{\rho,\rho}])\Bigg] \times \\
& \Bigg[\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,1}(E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,1}E^T_{1,1} + (E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,1}E^T_{1,1})^T + 2E_{1,1}E^T_{1,1}) \\
& + P_{1,\rho}(E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,\rho}E^T_{1,\rho} + (E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,\rho}E^T_{1,\rho})^T + 2E_{1,\rho}E^T_{1,\rho})+ \\
& \rho_{x,k_{jj}}P_{\rho,1}(E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,1}E^T_{\rho,1} + (E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,1}E^T_{\rho,1})^T + 2E_{\rho,1}E^T_{\rho,1})+ \\
& \rho_{x,k_{jj}}P_{\rho,\rho}(E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,\rho}E^T_{\rho,\rho} + (E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,\rho}E^T_{\rho,\rho})^T + 2E_{\rho,\rho}E^T_{\rho,\rho}))+ \\
& \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})(P^{'}_{1,1}(E^{'}_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,1}E^{'T}_{1,1} + (E^{'}_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,1}E^{'T}_{1,1})^T + 2E^{'}_{1,1}E^{'T}_{1,1}) \\
& + P^{'}_{1,\rho}(E^{'}_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,\rho}E^{'T}_{1,\rho} + (E^{'}_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,\rho}E^{'T}_{1,\rho})^T + 2E^{'}_{1,\rho}E^{'T}_{1,\rho})+ \\
& \rho_{x,k_{jj}}P^{'}_{\rho,1}(E^{'}_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,1}E^{'T}_{\rho,1} + (E^{'}_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,1}E^{'T}_{\rho,1})^T + 2E^{'}_{\rho,1}E^{'T}_{\rho,1})+ \\
& \rho_{x,k_{jj}}P^{'}_{\rho,\rho}(E^{'}_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,\rho}E^{'T}_{\rho,\rho} + (E^{'}_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,\rho}E^{'T}_{\rho,\rho})^T + 2E^{'}_{\rho,\rho}E^{'T}_{\rho,\rho}))\Bigg]^{-1}
\end{aligned} \tag{2.28}$$

where $j = 1, 2, ..., m$, P_{k_j} is the j^{th} row of $\boldsymbol{P}_k$ and $\rho_{x,k_{jj}}$ is the j^{th} row and j^{th} column element of diagonal matrix $\boldsymbol{\rho}_{x,k}$. μ_{x,k_j} and $E(...)_j$ are j^{th} element of $\boldsymbol{\mu}_{x,k}$ and $E(...)$, respectively.

$$
\begin{aligned}
C_{k_j} = & \Bigg[\sum_{i=1}^{n_1} 2(y_{i_j} - \mu_{y,k_j})[p(k \mid \boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,1}E_{1,1}^T + P_{\rho,1}E_{\rho,1}^T + \rho_{y,k_{jj}}P_{1,\rho}E_{1,\rho}^T + \rho_{y,k_{jj}}P_{\rho,\rho}E_{\rho,\rho}^T)]\Bigg] \times \\
& \Bigg[\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,1}(E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}E_{1,1}^T + (E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}E_{1,1}^T)^T + (2E_{1,1}E_{1,1}^T)) \\
& + P_{\rho,1}(E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}E_{\rho,1}^T + (E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}E_{\rho,1}^T)^T + (2E_{\rho,1}E_{\rho,1}^T))+ \\
& \rho_{y,k_{jj}}P_{1,\rho}(E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}E_{1,\rho}^T + (E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}E_{1,\rho}^T)^T + (2E_{1,\rho}E_{1,\rho}^T))+ \\
& \rho_{y,k_{jj}}P_{\rho,\rho}(E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}E_{\rho,\rho}^T + (E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}E_{\rho,\rho}^T)^T + (2E_{\rho,\rho}E_{\rho,\rho}^T)))\Big]^{-1}
\end{aligned}
\tag{2.29}
$$

where $j = 1, 2, ..., r$, C_{k_j} is the j^{th} row of $\boldsymbol{C}_k$ and $\rho_{y,k_{jj}}$ is the j^{th} row and j^{th} column element of diagonal matrix $\boldsymbol{\rho}_{y,k}$. μ_{y,k_j} and y_{i_j} are j^{th} element of $\boldsymbol{\mu}_{y,k}$ and $\boldsymbol{y}_i$, respectively.

The update equations for the covariances of input and output variables are given in Eqs. (2.30) and (2.31), respectively.

$$
\begin{aligned}
\sigma_{x,k}^2 = & \Bigg[\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})[P_{1,1}A_{1,1} + P_{1,\rho}A_{1,\rho} + P_{\rho,1}A_{\rho,1}^{\star} + P_{\rho,\rho}A_{\rho,\rho}^{\star}]+ \\
& \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})[P_{1,1}^{'}A_{1,1}^{'} + P_{1,\rho}^{'}A_{1,\rho}^{'} + P_{\rho,1}^{'}A_{\rho,1}^{'\star} + P_{\rho,\rho}^{'}A_{\rho,\rho}^{'\star}]\Bigg] \times (mn)^{-1}
\end{aligned}
\tag{2.30}
$$

$$
\sigma_{y,k}^2 = \frac{\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})[P_{1,1}B_{1,1} + P_{\rho,1}B_{\rho,1} + \boldsymbol{\rho}_{y,k}P_{1,\rho}B_{1,\rho}^{\star} + \boldsymbol{\rho}_{y,k}P_{\rho,\rho}B_{\rho,\rho}^{\star}]}{r.n_1}
\tag{2.31}
$$

The definitions of $A_{1,\triangle}$, $A_{\rho,\triangle}^{\star}$, $A_{1,\triangle}^{'}$ and $A_{\rho,\triangle}^{'\star}$ are given in appendix C.1, and the terms $B_{\star,1}$ and $B_{\star,\rho}^{\star}$ are presented in appendix C.2.

Similarly, the update equations for the mean values of input and output variables are given in Eqs. (2.32) and (2.33), respectively.

$$
\begin{aligned}
\mu_{x,k_j} = & \Bigg(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})([E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j P_{1,1}+ \\
& \rho_{x,k_{jj}}E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j P_{\rho,1} + E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j P_{1,\rho}+ \\
& \rho_{x,k_{jj}}E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{\rho,\rho}] - P_{k_j}(P_{1,1}E_{1,1} + P_{1,\rho}E_{1,\rho} + \rho_{x,k_{jj}}P_{\rho,1}E_{\rho,1} + \rho_{x,k_{jj}}P_{\rho,\rho}E_{\rho,\rho}))+ \\
& \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})([E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j P_{1,1}^{'}+ \\
& \rho_{x,k_{jj}}E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j P_{\rho,1}^{'} + E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j P_{1,\rho}^{'}+ \\
& \rho_{x,k_{jj}}E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})P_{\rho,\rho}^{'}] - P_{k_j}(P_{1,1}^{'}E_{1,1}^{'} + P_{1,\rho}^{'}E_{1,\rho}^{'} + \rho_{x,k_{jj}}P_{\rho,1}^{'}E_{\rho,1}^{'} + \rho_{x,k_{jj}}P_{\rho,\rho}^{'}E_{\rho,\rho}^{'})\Big) \times \\
& \Bigg(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})[P_{1,1} + P_{1,\rho} + \rho_{x,k_{jj}}P_{\rho,1} + \rho_{x,k_{jj}}P_{\rho,\rho}] + \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})[P_{1,1}^{'} + P_{1,\rho}^{'} + \rho_{x,k_{jj}}P_{\rho,1}^{'} + \rho_{x,k_{jj}}P_{\rho,\rho}^{'}]\Bigg)^{-1}
\end{aligned}
\tag{2.32}
$$

where $j = 1, ..., m$.

$$\begin{aligned}\mu_{y,k_j} =& \sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(y_{i,j}(P_{1,1} + P_{\rho,1} + \rho_{y,k_{jj}} P_{1,\rho} + \rho_{y,k_{jj}} P_{\rho,\rho}) \\ & - C_{k,j}(P_{1,1}E_{1,1} + P_{\rho,1}E_{\rho,1} + \rho_{y,k_{jj}} P_{1,\rho} E_{1,\rho} + \rho_{y,k_{jj}} P_{\rho,\rho} E_{\rho,\rho})) \\ & \times \left(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,1} + P_{\rho,1} + \rho_{y,k_{jj}} P_{1,\rho} + \rho_{y,k_{jj}} P_{\rho,\rho}) \right)^{-1}\end{aligned} \tag{2.33}$$

where $j = 1, 2, ..., r$

The update equations for the probability of each measurement in input and output data being an outlier are given in Eqs. (2.34) and (2.35), respectively.

$$\delta_{x,k} = \frac{\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})[P_{\rho,1} + P_{\rho,\rho}] + \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})[P^{'}_{\rho,1} + P^{'}_{\rho,\rho}]}{n} \tag{2.34}$$

$$\delta_{y,k} = \frac{\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,\rho} + P_{\rho,\rho})}{n_1} \tag{2.35}$$

To analyze the distinction of noise variances between the regular and the outlier data, the update equations for inflation matrices are essential and given in Eqs. (2.36) and (2.37) for input and output data, respectively.

$$\begin{aligned}\rho_{x,k_j} =& \left(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{\rho,1} + P_{\rho,\rho}) + \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})(P^{'}_{\rho,1} + P^{'}_{\rho,\rho}) \right) \times \\ & \left(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{\rho,1} C_{\rho,1_j} + P_{\rho,\rho} C_{\rho,\rho_j}) \sigma_{x,k}^{-2} + \sum_{i=n_1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})(P_{\rho,1} C^{'}_{\rho,1_j} + P_{\rho,\rho} C^{'}_{\rho,\rho_j}) \sigma_{x,k}^{-2} \right)^{-1}\end{aligned} \tag{2.36}$$

$$\boldsymbol{\rho}_{y,k} = \left(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,\rho} + P_{\rho,\rho}) \right) \left(\sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})(P_{1,\rho} + P_{\rho,\rho}) \times D_{\star,\rho_j} \sigma_{y,k}^{-2} \right)^{-1} \tag{2.37}$$

where definitions of $C_{\rho,\triangle_j}$ and $C^{'}_{\rho,\triangle_j}$ are given in appendix C.3 (Eqs. (C.7) and (C.8)), and $D_{\star,\rho_j}$ is defined in appendix C.4 (Eq. (C.9)).

Since EM is an iterative procedure in which the estimated parameters are used to update the posterior probabilities, the above estimated parameters are used in updating the posterior probabilities of $p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y_i}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})$, $p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})$, $p(q_{x_{i,k}}, q_{y_{i,k}} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, k, \boldsymbol{\theta}^{old})$, $p(q_{x_{i,k}}, q_{y_{i,k}} \mid \boldsymbol{x}_{i,o}, k, \boldsymbol{\theta}^{old})$, $p(\boldsymbol{x}_{i,k} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})$ and $p(\boldsymbol{x}_{i,k} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})$, that are required in the $Q - function$ as defined in Eqs. (B.2)-(B.10). Expressions for the aforementioned posterior probabilities are derived using the Bayes rule and are given as follows

$$p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y_i}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i,o},\boldsymbol{y}_i|\boldsymbol{t}_{i,k},q_{x_{i,k}},q_{y_{i,k}},k,\boldsymbol{\theta}^{old})p(\boldsymbol{t}_{i,k}|q_{x_{i,k}},q_{y_{i,k}},k,\boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i,o},\boldsymbol{y}_i|q_{x_{i,k}},q_{y_{i,k}},k,\boldsymbol{\theta}^{old})} \tag{2.38}$$

$$p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i,o} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i,o} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})} \tag{2.39}$$

By using Eqs. (2.25) and (2.26) in Eqs. (2.38) and (2.39) respectively, the terms $p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y_i}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})$ and $p(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old})$ follow a Gaussian distribution with means (denoted as $E_{(\cdot,\cdot)}$) and variances (denoted as $E_{(\cdot,\cdot)}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T)$) given below:

For the labeled dataset:

$$\begin{aligned}
E_{1,1} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}) + \boldsymbol{C}_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}))\\
E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1} + (E_{1,1}E_{1,1}^T)\\
E_{\rho,1} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}) + \boldsymbol{C}_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}))\\
E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1} + (E_{\rho,1}E_{\rho,1}^T)\\
E_{1,\rho} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}) + \boldsymbol{C}_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}))\\
E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1} + (E_{1,\rho}E_{1,\rho}^T)\\
E_{\rho,\rho} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1} \times (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}) + \boldsymbol{C}_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}))\\
E_{\rho\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + C_k^T(\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I})^{-1}\boldsymbol{C}_k + I)^{-1} + (E_{\rho,\rho}E_{\rho,\rho}^T)
\end{aligned} \tag{2.40}$$

and for the unlabeled dataset:

$$\begin{aligned}
E_{1,1}^{'} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}))\\
E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1} + (E_{1,1}E_{1,1}^T)\\
E_{\rho,1}^{'} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}))\\
E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1} + (E_{\rho,1}E_{\rho,1}^T)\\
E_{1,\rho}^{'} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}))\\
E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1} + (E_{1,\rho}E_{1,\rho}^T)\\
E_{\rho,\rho}^{'} &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1}(\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}(\boldsymbol{x}_{i,o} - \boldsymbol{\mu}_{\boldsymbol{x,k,o}}))\\
E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) &= (\boldsymbol{P}_{\boldsymbol{k,o}}^T(\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I})^{-1}\boldsymbol{P}_{\boldsymbol{k,o}} + I)^{-1} + (E_{\rho,\rho}E_{\rho,\rho}^T)
\end{aligned} \tag{2.41}$$

The posterior update for input and output indicators can be estimated using Eqs. (2.42) and (2.43) with the aid of Eqs. (2.22)-(2.26).

$$p(q_{x_{i,k}}, q_{y_{i,k}} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, k, \boldsymbol{\theta}^{old}) = p(\boldsymbol{x}_{i,o}, \boldsymbol{y}_i \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old}) \times p(q_{x_{i,k}} \mid k, \boldsymbol{\theta}^{old}) \times p(q_{y_{i,k}} \mid k, \boldsymbol{\theta}^{old}) \tag{2.42}$$

$$p(q_{x_{i,k}}, q_{y_{i,k}} \mid \boldsymbol{x}_{i,o}, k, \boldsymbol{\theta}^{old}) = p(\boldsymbol{x}_{i,o} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{\theta}^{old}) \times p(q_{x_{i,k}} \mid k, \boldsymbol{\theta}^{old}) \times p(q_{y_{i,k}} \mid k, \boldsymbol{\theta}^{old}) \tag{2.43}$$

The update of the posterior probability related to hidden variable $\boldsymbol{x}_{i,k}$ for the labeled dataset can be derived as follows:

$$\begin{aligned} p(\boldsymbol{x}_{i,k} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) &= \frac{p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{x}_{i,k}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) p(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})}{p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})} \\ &= \frac{p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{x}_{i,m}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) p(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})}{p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})} \end{aligned} \tag{2.44}$$

Since $x_{i,m}$ is not available, the following assumption is made in order to make the problem tractable:

$$p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, x_{i,m}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \;\approx\; p(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \tag{2.45}$$

By substituting Eq. (2.45) in Eq. (2.44), an approximation of the following form is obtained:

$$p(\boldsymbol{x}_{i,k} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \approx p(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \tag{2.46}$$

Similarly, for the unlabeled dataset the posterior probability can be updated as

$$p(\boldsymbol{x}_{i,k} \mid \boldsymbol{t}_{i,k}, q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \;\approx\; p(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \tag{2.47}$$

Therefore, from Eqs. (2.46) and (2.47), it can be concluded that $\boldsymbol{x}_{i,k}$ follows a Gaussian distribution; hence the mean and covariance are required. The mean for the labeled dataset can be derived as:

$$E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \begin{bmatrix} \boldsymbol{x}_{i,o} \\ E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{2.48}$$

and for the unlabeled dataset:

$$E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = \begin{bmatrix} \boldsymbol{x}_{i,o} \\ E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{2.49}$$

The covariance of $\boldsymbol{x}_{i,k}$, for labeled dataset, can be derived as:

$$cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \begin{bmatrix} 0 & 0 \\ 0 & cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{2.50}$$

and for the unlabeled dataset:

$$cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = \begin{bmatrix} 0 & 0 \\ 0 & cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{2.51}$$

The expectation terms in Eqs. (2.48) and (2.49) have to be updated by using the input model equation, and for the labeled dataset it is

$$E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = P_{k,m} E(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) + \mu_{x,k,m} \tag{2.52}$$

and for the unlabeled dataset, it is

$$E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = P_{k,m} E(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) + \mu_{x,k,m} \tag{2.53}$$

where $E(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})$ and $E(\boldsymbol{t}_{i,k} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})$ are derived in Eqs. (2.40) and (2.41), respectively. To compute the covariances in Eqs. (2.50) and (2.51), it is necessary to calculate $cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})$ and $cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}}, q_{y_{i,k}}, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})$. The derivations related to these terms are provided in appendix C.5.

The parameters in Eqs. (2.28)-(2.37) are estimated again by using the updated posterior probabilities given in Eqs. (2.38)-(2.47) and are repeated until the parameters converge.

2.3.2 Online Predictions

For the soft sensor application, it is necessary to predict the variables online. To perform the online predictions given a new data point ($\boldsymbol{x}^{new}$), a weighted predicted value of the output over all K modes is used and is given as

$$\hat{\boldsymbol{y}}^{new} = \sum_{k=1}^{K} p(k \mid \boldsymbol{x}^{new}, \boldsymbol{\theta}) \times \hat{\boldsymbol{y}}_k^{new} \tag{2.54}$$

where $p(k \mid \boldsymbol{x}^{new}, \boldsymbol{\theta})$ is the posterior probability and $\hat{y}_k^{new}$ is the predicted output in each mode and is provided by using the generative model in Eq. (2.6). These terms are given as

$$p(k \mid \boldsymbol{x}^{new}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}^{new} \mid k, \boldsymbol{\theta}) \times p(k \mid \boldsymbol{\theta})}{p(\boldsymbol{x}^{new} \mid \boldsymbol{\theta})} \tag{2.55}$$

$$\hat{\boldsymbol{y}}_k^{new} = \boldsymbol{C}_k \hat{\boldsymbol{t}}_k^{new} + \boldsymbol{\mu}_{y,k} \tag{2.56}$$

in which $\hat{\boldsymbol{t}}_k^{new}$ is the expectation of the latent variable and is computed for each mode as

$$\begin{aligned}
\hat{\boldsymbol{t}}_k^{new} =& E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \\
=& \sum_{q_{x_k}} \sum_{q_{y_k}} E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, q_{x_k}^{new}, q_{y_k}^{new}, k, \boldsymbol{\theta}) \times p(q_{x_k}^{new} \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \times P(q_{y_k}^{new} \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \\
=& E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, q_{x_k}^{new} = 1, q_{y_k}^{new} = 1, k, \boldsymbol{\theta}) \times p(q_{x_k}^{new} = 1 \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \times p(q_{y_k}^{new} = 1 \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \\
&+ E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, q_{x_k}^{new} = \rho, q_{y_k}^{new} = 1, k, \boldsymbol{\theta}) \times p(q_{x_k}^{new} = \rho \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \times p(q_{y_k}^{new} = 1 \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \\
&+ E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, q_{x_k}^{new} = 1, q_{y_k}^{new} = \rho, k, \boldsymbol{\theta}) \times p(q_{x_k}^{new} = 1 \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \times p(q_{y_k}^{new} = \rho \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \\
&+ E(\boldsymbol{t}_k^{new} \mid \boldsymbol{x}^{new}, q_{x_k}^{new} = \rho, q_{y_k}^{new} = \rho, k, \boldsymbol{\theta}) \times p(q_{x_k}^{new} = \rho \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta}) \times p(q_{y_k}^{new} = \rho \mid \boldsymbol{x}^{new}, k, \boldsymbol{\theta})
\end{aligned} \tag{2.57}$$

To evaluate the prediction performance, R-squared (R^2) test and root mean square error ($RMSE$) are commonly used and are also adopted here. Given a dataset y and its prediction $\hat{y}$, the $RMSE$ and R^2 will be calculated according to

$$\begin{aligned}
RMSE &\triangleq \sqrt{\frac{\|y - \hat{y}\|_2^2}{n}} \\
R^2 &= 1 - \frac{\|y - \hat{y}\|_2}{\|y - \bar{y}\|_2}
\end{aligned}$$

where n is the total number of samples used and $\bar{y}$ is the mean of the dataset y.

2.4 Case Studies

In this section, the performance of the proposed method is demonstrated by considering two case studies. A numerical example is considered as the first case study and an experiment on a hybrid tank pilot plant system is considered as the second one to demonstrate the practical applicability of the proposed method.

2.4.1 Numerical Example

The following model is used to generate the dataset.

$$\boldsymbol{x}_{i,k} = \boldsymbol{P}_k \boldsymbol{t}_{i,k} + \boldsymbol{e}_{i,k} + \boldsymbol{\mu}_{x,k} \tag{2.58}$$

$$\boldsymbol{y}_{i,k} = \boldsymbol{C}_k \boldsymbol{t}_{i,k} + \boldsymbol{f}_{i,k} + \boldsymbol{\mu}_{y,k} \tag{2.59}$$

A three-operating-mode problem with five input variables and one output variable in each mode is considered. Therefore, the values of k will be either 1, 2, or 3, and the weighting matrices $\boldsymbol{P}_k \in \mathbb{R}^{5\times3}$ and $\boldsymbol{C}_k \in \mathbb{R}^{1\times3}$ are selected randomly. The latent variable in each operating mode ($\boldsymbol{t}_{i,k}$) follows a Gaussian distribution of $\mathcal{N}(0, \boldsymbol{I})$, and the values of $\boldsymbol{\mu}_{x,k}$ and $\boldsymbol{\mu}_{y,k}$ are set to zero. The input ($\boldsymbol{e}_{i,k}$) and output ($\boldsymbol{f}_{i,k}$) measurement noises in each mode also follow Gaussian distribution with zero mean and variance 0.05 and 0.35, respectively, i.e., in Eqs. (2.11) and (2.12), $\sigma^2_{x,k} = 0.05$ and $\sigma^2_{y,k} = 0.35$.

A total of 900 data samples (300 samples each mode) for each variable are generated as shown in Fig. 2.3. To study the effect of missing output data, $4/5^{\text{th}}$ of output data are removed randomly (approximately 240 output samples per mode). 30% of input samples are also considered to be missing at random locations. Further, to demonstrate the robustness in identification with outliers in data, 5% and 25% of the entire dataset are replaced respectively with data that follow Gaussian distribution of zero mean and variances $(0.2, 0.15, 0.05, 0.1, 0.05)$ as diagonal entries and 0.7 for output data, respectively i.e., the third and the fifth input variables are not contaminated with outliers since the replaced data have the same variance as the normal data.

A comparative study among the proposed method, MSSPPCR [8] and traditional MRSSPPCR [43] is made by showing the training and validation results under various scenarios shown in Tables 2.1 and 2.2. Further, the prediction performance of the proposed method is shown in Fig. 2.4.

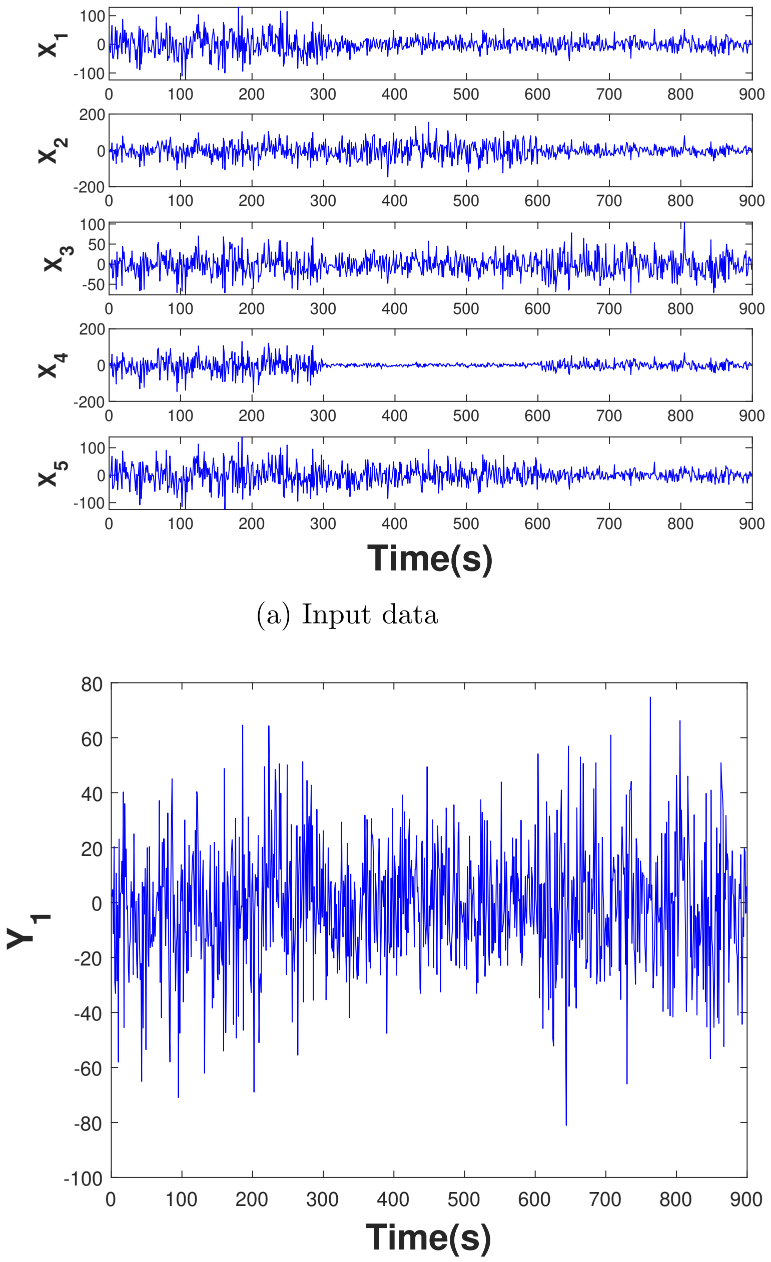
Table 2.1: RMSE and R^2 values under various missing inputs and outliers with sampling ratio of 1/5 for training set

Missing Percentage	No missing input						30%					
Outliers	Proposed Model		MSSPPCR[8]		MRSSPPCR[43]		Proposed Model		MSSPPCR[8]		MRSSPPCR[43]	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
No outlier	5.7301	0.9346	5.7422	0.9348	5.7331	0.9339	5.7828	0.9267	5.7647	0.9282	6.0152	0.8926
5%	5.7749	0.9289	5.9962	0.8974	5.7784	0.9302	5.7627	0.9009	6.1332	0.8493	6.0571	0.8853
25%	5.8413	0.9210	6.4891	0.7698	5.9031	0.9078	6.0410	0.8967	6.8419	0.7166	6.2875	0.8433

Table 2.2: RMSE and R^2 values under various missing inputs and outliers with sampling ratio of 1/5 for online validation set

Missing Percentage	No missing input						30%					
Outliers	Proposed Model		MSSPPCR[8]		MRSSPPCR[43]		Proposed Model		MSSPPCR[8]		MRSSPPCR[43]	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
No outlier	5.8321	0.9194	5.8320	0.9196	5.8318	0.9196	5.8331	0.9172	5.8327	0.9173	5.9982	0.8417
5%	5.8913	0.9176	5.9276	0.8349	5.8908	0.9179	6.0243	0.8995	6.1471	0.8035	6.0986	0.8262
25%	6.1482	0.8969	6.9074	0.6982	6.2033	0.8710	6.2775	0.8846	7.0524	0.6874	6.7868	0.7936

From the results reported in Fig. 2.4 and from Table 2.2, it can be noted that the prediction of the proposed method is more accurate and also robust to different levels

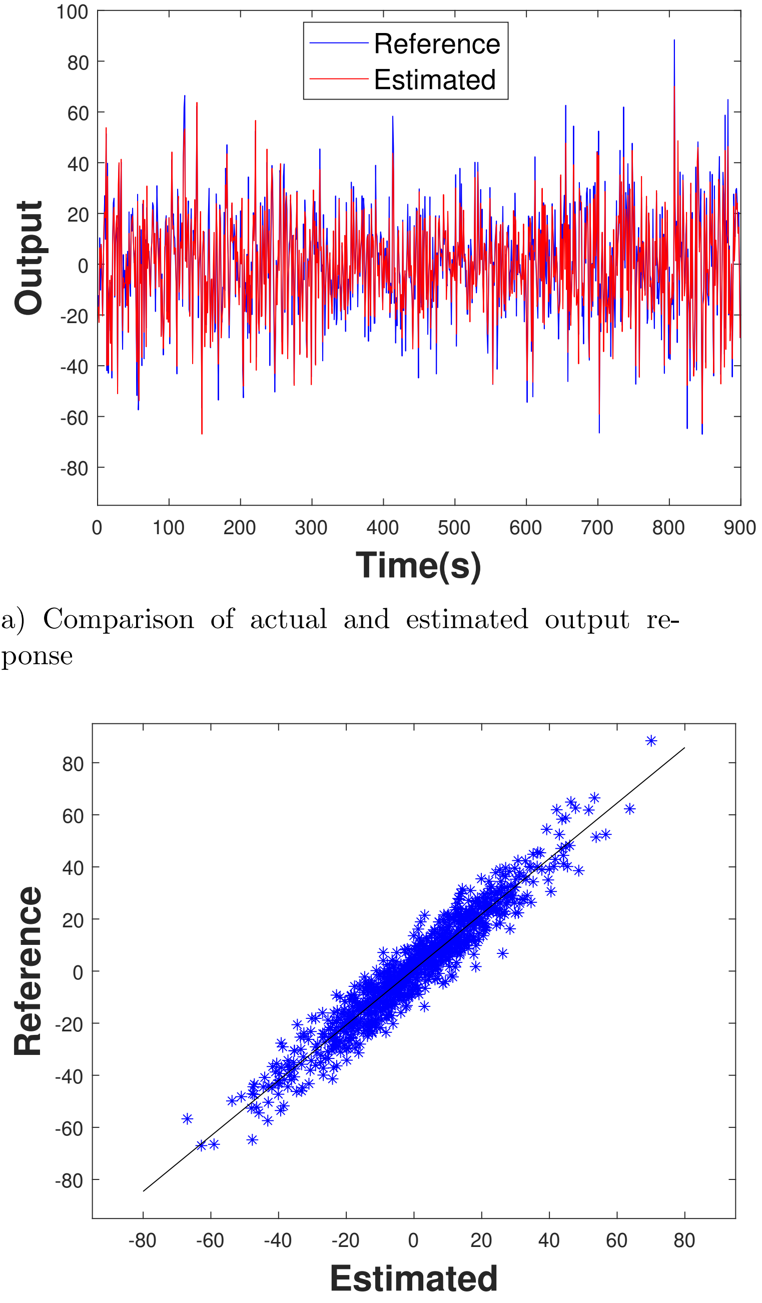


(a) Input data

(b) Output data

Figure 2.3: Generated data for numerical example

of outliers and missing data compared with the MSSPPCR and traditional MRSSPPCR methods. Further, due to the introduction of flattening ratio in the formulation, the variances of outliers in the data can also be inferred. For example, in the case with 25% of outliers and no missing data in the input, the flattening ratio of input and output data is estimated as $\boldsymbol{\rho}_{x,k} = \mathrm{diag}(0.2114, 0.3458, 0.9425, 0.6471, 0.9123)$



(a) Comparison of actual and estimated output response

(b) Parity plot

Figure 2.4: Online prediction performance of the proposed method on the numerical data with 25% outliers and no missing input.

and $\boldsymbol{\rho}_{y,k} = 0.4512$, respectively. From these results, it can be inferred that a value close to 1 for the third and the fifth input variables implies that these input variables have no outliers which is consistent with the real situation. The computation time of the proposed method with 4 CPU, Intel(R) Core(TM) i7 Processor, 2.60 GHz for the case when a 30% of data is missing and outlier percentage is 25%, is approximately 39 seconds. For the case of MRSSPPCR [43] and MSSPPCR [46] the computational time is approximately 31 and 29 seconds, respectively. Though the proposed method is slightly more time consuming ($\approx$ 8 to 9 seconds more when compared to the existing methods in all the cases), the proposed method is better over the other methods in terms of its robustness and more accuracy.

2.4.2 Experimental case study: Hybrid tank pilot plant system

A hybrid tank pilot plant system, as shown in Fig. 2.5, is utilized for demonstrating the practical applicability of the proposed method.

a. Model description

A hybrid tank system consists of three cylindrical tanks connected by six valves, namely V_1-V_4, V_6, and V_8, and a container at the bottom of these three tanks to collect the tanks' outflow. Three level-sensors, namely LT_1-LT_3, are used to measure the level in each of these tanks. The tanks are connected to the container through three valves, namely V_5, V_7, and V_9. Two similar pumps are also connected to this container to send the water into Tanks 1 and 3, and the flowrate of these streams is considered as manipulated variables. The level in Tank 2 is regarded as the variable that is of importance.

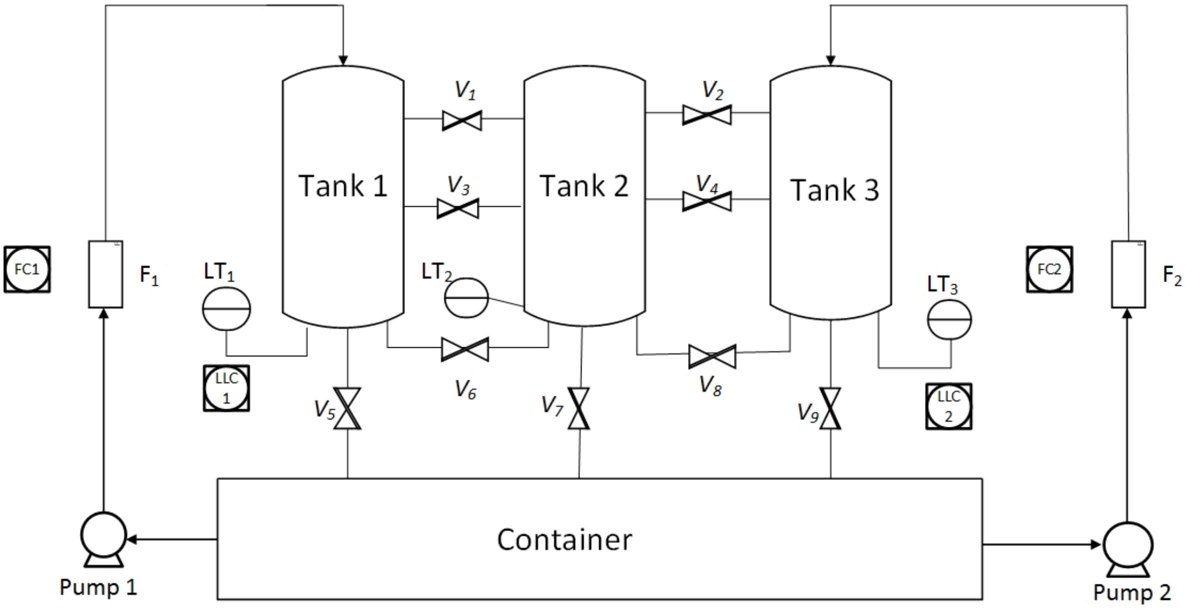


Figure 2.5: Schematic of the Hybrid Tank Pilot Plant

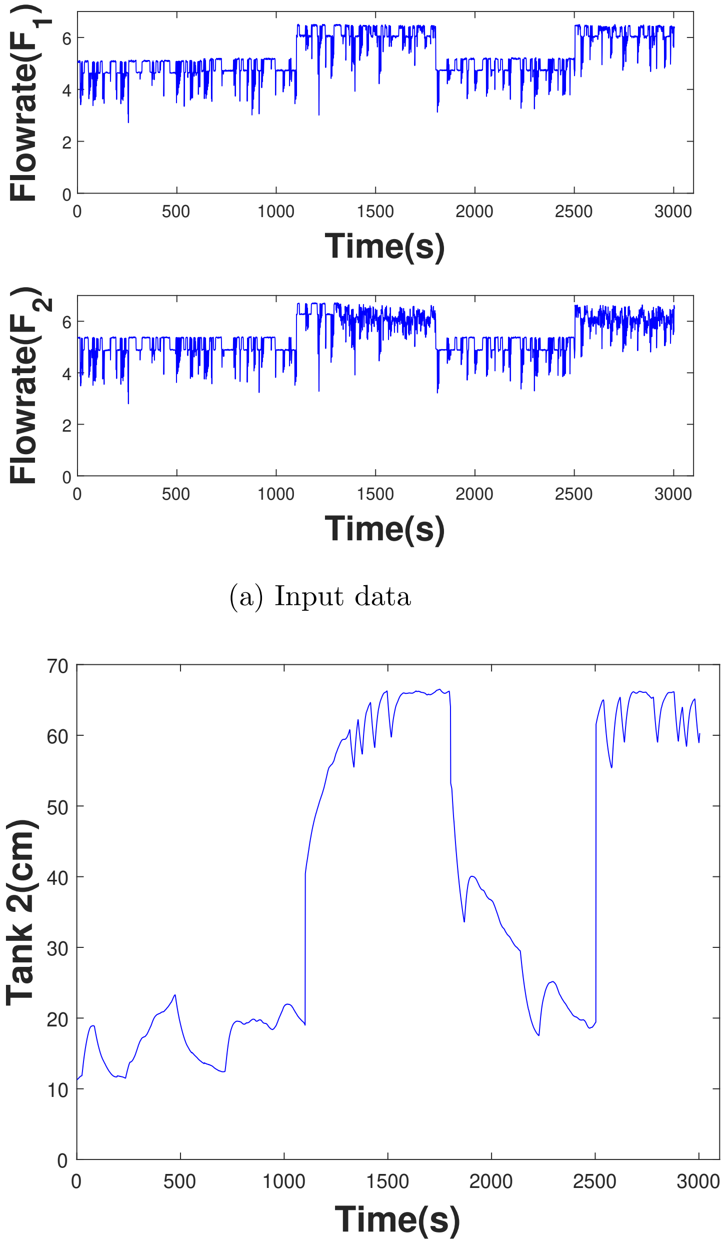
The process operates in different operating conditions depending on the considered range of inlet flowrate to Tanks 1 and 3. In the current case study, two different operating points are considered by manipulating the inlet flowrates of Tanks 1 and 3 around 4.75 and 5.15 for the first operating point, termed as a low-level operating point. For the second operating point, termed as a high-level operating point, the flowrates are manipulated to be around 6.15 and 6.00, respectively. Since the valves V_1 and V_2 are kept open throughout the experiment, the significant difference in operating conditions arises when the level in the tanks exceeds the position of these valves. Different process modes can be generated by changing the valves V_3 and V_4 from open to close. In such a case, an overflow might arise when the system is operated at a high-level operating point, which is considered to be the abnormal process mode. Therefore, two different modes (abnormal and normal process modes) will be present in the system.

The input and output data for the considered operating conditions are shown in Figures 2.6(a) and 2.6(b), respectively [49]. The data consist of almost 3000 samples, of which 1800 samples that correspond to the first period of the signal shown in Fig. 2.6 are used for training the model, and 1200 samples of the remaining data are used for validation. To demonstrate the efficacy of the proposed method for modeling with outliers in data, noise generated from $\mathcal{N}(0, 25\sigma^2 I)$ and $\mathcal{N}(0, 30\sigma^2 I)$, where σ is the variance, is added to the first input variable data and output variable data at random locations, respectively.

b. Identification and validation results

The proposed method is implemented on different outlier levels and with different amounts of missing data in the inputs. It is also assumed that $4/5^{\text{th}}$ of output data is missed completely at random. A comparison in terms of the RMSE values and the R^2 is made between the proposed method and the MSSPPCR method presented in [8] and MRSSPPCR proposed in [43]. The results with training and validation datasets are shown in Tables 2.3 and 2.4. To further demonstrate the performance of the proposed method, a scenario where the input and output data have 30% and 80% missing values, respectively with a 15% of outliers in the data is considered. A comparison of the prediction performance of the proposed method with the MSSPPCR and MRSSPPCR is shown in Fig. 2.7.

From the comparison presented in Table 2.4 and Fig. 2.7, it can be concluded that the proposed method is able to identify the model more accurately compared



(a) Input data

(b) Output data

Figure 2.6: Collected data for experimental example

to the MSSPPCR method presented in [8] and MRSSPPCR presented in [43]. This improvement can be attributed to the fact that the proposed method can deal with different level of outliers in different variables and simultaneously considers missing data in both the input and output variables. Regarding the computational cost, a trend similar to the numerical case study is also observed in the experimental case study.

Table 2.3: RMSE and R^2 values under various missing inputs and outliers with sampling ratio of 1/5 for training set

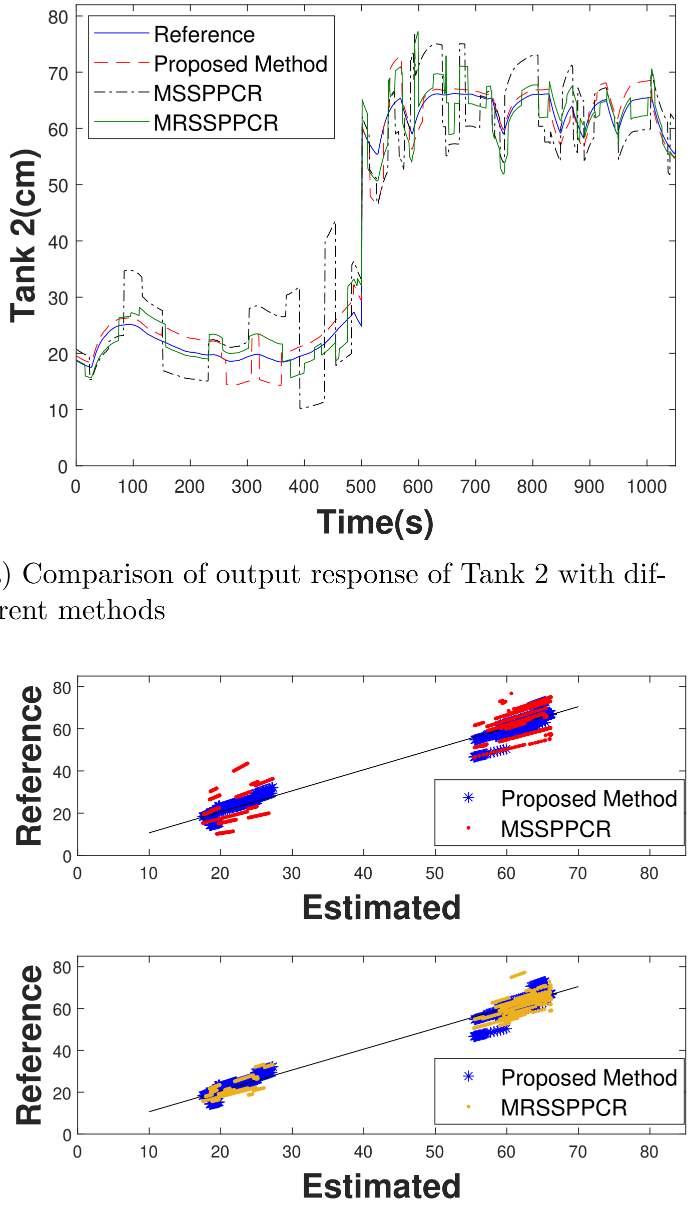
Missing Percentage	No missing input						30%					
Outliers	Proposed Model		MSSPPCR[8]		MRSSPPCR[43]		Proposed Model		MSSPPCR[8]		MRSSPPCR[43]	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
No outlier	9.3162	0.9015	9.3162	0.9014	9.3108	0.9018	9.3249	0.8945	9.3391	0.8951	9.3384	0.8574
5%	9.3310	0.8913	9.5921	0.8445	9.3257	0.8936	9.3847	0.8748	9.7143	0.8186	9.6058	0.8263
15%	9.6823	0.8642	9.8771	0.7427	9.7855	0.8462	9.7956	0.8281	10.0107	0.6797	10.3546	0.7596

Table 2.4: RMSE and R^2 values under various missing inputs and outliers with sampling ratio of 1/5 for online validation set

Missing Percentage	No missing input						30%					
Outliers	Proposed Model		MSSPPCR[8]		MRSSPPCR[43]		Proposed Model		MSSPPCR[8]		MRSSPPCR[43]	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
No outlier	9.8718	0.8249	9.8823	0.8236	9.8524	0.8261	10.0228	0.8058	10.1012	0.8108	10.4852	0.7538
5%	10.1738	0.8172	10.4317	0.7491	10.1442	0.8203	10.3149	0.7863	10.5394	0.7107	10.5973	0.6985
15%	10.4890	0.7694	10.8146	0.6624	10.5758	0.7513	10.7489	0.7437	10.3218	0.5759	10.0913	0.6684

2.5 Conclusion

In this chapter, a new mixture robust semi-supervised PPCR (MRSSPPCR) model with missing input data is developed. The proposed approach can handle the multimodal nature of the process. It can efficiently handle the missing input and output data and have the flexibility of handling different nature of outliers among different variables. Since outliers can appear in different variables with different statistical properties, the proposed approach will provide more practical results over the existing methods since no existing methods have considered this practical issue. The estimated values of the missing data can be iteratively updated while updating the parameters of the model. The robustness and performance of the proposed method are demonstrated on different scenarios through a numerical and an experimental study.



(a) Comparison of output response of Tank 2 with different methods

(b) Parity plots

Figure 2.7: Prediction performance on industrial data with 15% outliers and 30% missing input

Chapter 3

Weighted semi-supervised probabilistic principal component regression with missing input and delayed output data[1]

3.1 Introduction

The mixture robust semi-supervised probabilistic principal component regression (MRSSPPCR) model developed in Chapter 2, extends the PPCR model to handle uncertainties like missing data in input and output variables, outliers with different properties, and dealing with mixture-modal/nonlinear nature of the plants. However, it is not able to identify the presence of time-delays between each input variable and output variables. In addition, determination of the number of mixture components in industries can be challenging. The MRSSPPCR model uses the information of all the available data in dataset to develop the model which increases the computational effort. Hence, in this chapter, a weighted semi-supervised probabilistic principal component regression (WSSPPCR) model is proposed to address the aforementioned challenges.

Inspired by the ideas of locally weighted learning [50, 51] and just-in-time learning [52, 53] techniques, a weighted PPCR (WPPCR) model is proposed in [54]. The advantage of the WPPCR model lies in its adaptability by providing an efficient

[1] A. Memarian, S. K. Varanasi, B. Huang. "Soft sensor development in the presence of missing input and delayed output data through weighted semi-supervised probabilistic principal component regression". Submitted to *IEEE Transactions on Industrial Electronics"*, 2021 (Chapter 3 - Short Version)

model based on selecting a subset of the training data that is relevant to the current operating conditions of the process without the need for determining the number of mixture components. To achieve this, weights are assigned to the training samples based on the Euclidean distance between the current operating point and the training data. The authors in [55] proposed a locally semi-supervised weighted PPCR model to account for the missing data in output variables.

It has to be noted that alongside the outliers and missing data, the efficiency of the soft sensor also depends on the time-delay between the input and output variables. Depending on the time required for lab analysis, the input and the output variables may not be sampled at the same time, and the output often falls behind the input [56]. Therefore, it is important to account for the time-delay while building a soft sensor. The determination of time-delay can be either from the understanding of the process mechanism [57] or data-driven methods. The former typically uses the process information and first principle models to determine the time-delay, while the latter, i.e. data-driven methods, rely only on the process data. To identify time-delay directly from the data, methods like Pearson correlation coefficient (CC) [58], fuzzy curve analysis (FCA) [59], the mutual information (MI) [60, 61] are utilized. In the design of a soft sensor, the authors in [62] proposed an algorithm called a weighted relevance vector machine model based on dynamic time-delay estimation (DTDE-WRVM) that is capable of estimating the dynamic time-delay. However, it uses two separate steps in the determination of time-delays and the modeling, respectively. Clearly, the accuracy of one step will greatly impact the other. Hence, it is desirable to develop a soft sensor that can handle the estimation of time-delay and missing data simultaneously.

In view of aforementioned points, this chapter proposes a weighted semi-supervised probabilistic principal component regression with missing input data and delayed output data. The proposed approach can model non-linear and/or multi-modal processes. The significance of the chapter follows with its ability to provide flexibility to each input variable to have their own time-delay while simultaneously considering missing data in both input and output.

The rest of the chapter is organized as follows. In Section 3.2, a detailed description of the proposed algorithm is presented. The accuracy of the proposed algorithm is demonstrated through a simulated and an experimental case study in Section 3.3. The conclusions are drawn in Section 3.4.

3.2 Weighted semi-supervised probabilistic principal component regression with missing input and delayed output data

In this section, the proposed algorithm of the weighted semi-supervised probabilistic principal component regression model (WSSPPCR) with missing data in both input and output along with time delays is presented. The proposed approach utilizes the techniques of semi-supervised learning to handle missing data, and also provides flexibility for each input variable to have its own distinguished delay. Further, due to the use of a just-in-time learning strategy, the proposed algorithm is able to model nonlinear and multi-modal processes. The details are provided in the rest of the section.

3.2.1 Model Description

In this section, the details on the WSSPPCR model are presented by considering $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{r\times n}$ to be the datasets of the input and output variables, respectively. The total number of samples is denoted as n, and the number of input and output variables is denoted as m and r, respectively. In such a setting, n_1 samples are labeled (observed data), represented by the set $\boldsymbol{O}$. The remaining $n - n_1$ samples are unlabeled (missing) and are represented by the set $\boldsymbol{M}$. It is also assumed that input variables have missing values completely at random (MCAR) [47]. Let the variable λ denote the time-delay and the matrix $\boldsymbol{X}_\lambda$ represent the modified input dataset wherein, the delay for each input variable is accounted for, the generative model of the WSSPPCR can be represented as

$$\boldsymbol{x}_{i_\lambda} = \boldsymbol{P}\boldsymbol{t}_i + \boldsymbol{e}_i + \boldsymbol{\mu}_x, \quad i = 1, 2, \cdots, n \tag{3.1}$$

$$\boldsymbol{y}_j = \boldsymbol{C}\boldsymbol{t}_j + \boldsymbol{f}_j + \boldsymbol{\mu}_y, \quad j = 1, 2, \cdots, n_1 \tag{3.2}$$

where, $\boldsymbol{x}_{i_\lambda} \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{y}_i \in \mathbb{R}^{r\times 1}$ denote the input and output data at i^{th} sampling instant of the datasets $\boldsymbol{X}_\lambda$ and $\boldsymbol{Y}$, respectively i.e.,

$$\boldsymbol{x}_{i_\lambda} = \begin{bmatrix} x_{1_{i-\lambda_1}} \\ x_{2_{i-\lambda_2}} \\ \vdots \\ x_{m_{i-\lambda_m}} \end{bmatrix} \tag{3.3}$$

where $\lambda_1, \lambda_2, \cdots, \lambda_m$ are the time-delays corresponding to the first, second,..., m^{th} input variables, respectively. $\boldsymbol{P} \in \mathbb{R}^{m\times q}$ and $\boldsymbol{C} \in \mathbb{R}^{r\times q}$ are the weighting matrices and $\boldsymbol{t}_i \in \mathbb{R}^{q\times 1}$ is a vector of latent variables. The variables $\boldsymbol{e}_i \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{f}_i \in \mathbb{R}^{r\times 1}$ denote the measurement noise in input and output, respectively, which are assumed to follow a Gaussian distribution as given in Eqs. (3.4)-(3.6). The mean values of the input and output variables are denoted with $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$, respectively.

$$\boldsymbol{t}_i \sim \mathcal{N}(0, \boldsymbol{I}) \tag{3.4}$$

$$\boldsymbol{e}_i \sim \mathcal{N}(0, \sigma_x^2 \boldsymbol{I}) \tag{3.5}$$

$$\boldsymbol{f}_i \sim \mathcal{N}(0, \sigma_y^2 \boldsymbol{I}) \tag{3.6}$$

It is further assumed that the delay of each input variable ($\lambda_z \forall z = 1, \cdots, m$) has an upper and a lower bound. These limits can be assigned based on the process knowledge, or a trivial value of zero is considered as a lower bound, and a large value is considered for the upper bound. Thus, the range of each input delay is defined as:

$$d_1^z \leq \lambda_z \leq d_2^z \qquad\qquad z = 1, ..., m \tag{3.7}$$

It has to be noted that the range of the delay considered can be updated and optimized through the updating equation provided in Eq. (3.34) in Section 3.2.3 and this type of updating helps in faster convergence and better results. In such a setting, the objective is to estimate the parameters given in $\boldsymbol{\theta} = \{\boldsymbol{P}, \boldsymbol{C}, \sigma_x^2, \sigma_y^2, \boldsymbol{\mu}_x, \boldsymbol{\mu}_y\}$ along with the hidden or latent variables i.e., $\{\boldsymbol{X}_{\boldsymbol{\lambda}}, \{\boldsymbol{t}_{i,k}\}_{i=1}^n, \boldsymbol{\Gamma}\}$ where, $\boldsymbol{\Gamma} = \{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ is the vector of all input delays.

3.2.2 Weight Assignment

Since most of the industrial processes are non-linear and/or multi-modal, the development of a single-PPCR model using the information of complete data set is not practical. Thus, to improve the accuracy in modeling, weights are assigned to all the data points in such a way that relevant data points are selected for modeling. Therefore, similar to the work in [54], a weighted log-likelihood function is used in the development of the model. The weights, w_i used in the $Q - function$ (given in Eq. (3.10)) are calculated using Eq. (3.8).

$$w_i = \exp(\frac{-d_i^2}{\phi}) \tag{3.8}$$

where, ϕ is a tuning parameter that is defined based on a trial and error and it controls the distribution of the weights as shown in Fig. 3.1 i.e., it controls the range of the data that are used while building a model. d_i is the Euclidean distance and is chosen as [63]

$$d_i = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_q)^T \boldsymbol{\psi}^{old} (\boldsymbol{x}_i - \boldsymbol{x}_q)} \tag{3.9}$$

In Eq. (3.9), $\boldsymbol{\psi} \coloneqq \text{diag}\left[\boldsymbol{C}(\boldsymbol{P}^T\boldsymbol{P} + \sigma_x^2\boldsymbol{I})^{-1}\boldsymbol{P}^T\right]$ represents the parameter that intensifies the importance of each input variable to the output variable and $\boldsymbol{x}_q$ is the query point around which the model needs to be developed.

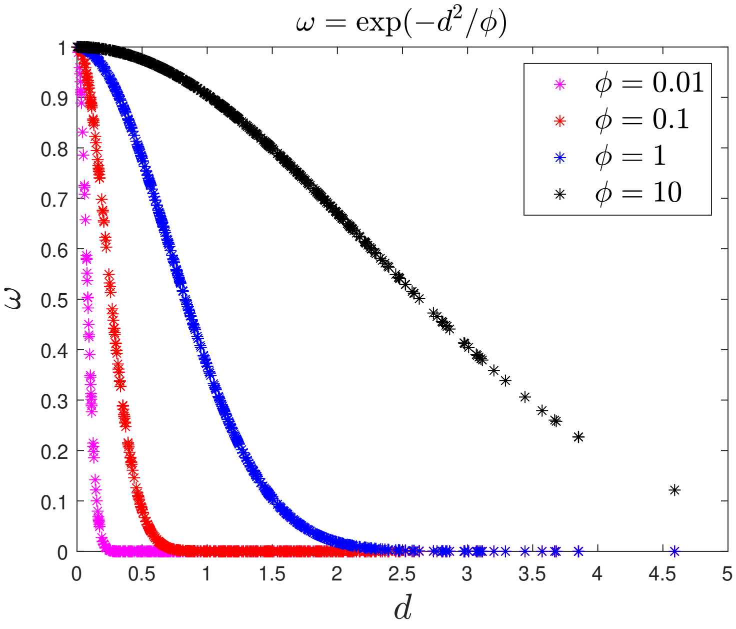


Figure 3.1: Weight assignment based on the tuning parameter ϕ.

3.2.3 Parameter estimation through EM algorithm

In the case of missing data and/or in the presence of latent variables, the likelihood function is not tractable [44, 45]. Therefore, the approaches like maximum likelihood estimation and/or maximum-a-posteriori are difficult to use for the estimation of parameters and in such scenarios, Expectation-Maximization (EM) algorithm is an efficient choice. EM algorithm consists of two steps, the expectation (E) step, and the maximization (M) step.

In E-step, the expectation of the weighted log-likelihood, i.e. $Q - function$, is calculated. Due to the presence of observed and missed values, the weighted log-likelihood comprises of two parts as provided in Eq. (3.10)

$$Q = E_{\boldsymbol{X}_\lambda,\boldsymbol{T},\boldsymbol{\Gamma}|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\theta}^{old}} \left[\sum_{i \in \boldsymbol{O}} w_i \log p(\boldsymbol{x}_{i_\lambda}, \boldsymbol{y}_i, \boldsymbol{t}_i, \boldsymbol{\Gamma} \mid \boldsymbol{\theta}) + \sum_{i \in \boldsymbol{M}} w_i \log p(\boldsymbol{x}_{i_\lambda}, \boldsymbol{t}_i, \boldsymbol{\Gamma} \mid \boldsymbol{\theta}) \right] \tag{3.10}$$

Since noise in input and output variables, delays, and latent variables are assumed to be independent and identically distributed (i.i.d), the terms in $Q - function$ can be expanded as:

$$\begin{aligned}
Q = &\sum_{i \in \boldsymbol{O}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\Gamma}=J,\boldsymbol{\theta}^{old}}(w_i \log p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{\theta})) \\
&+ \sum_{i \in \boldsymbol{O}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\Gamma}=\boldsymbol{J},\boldsymbol{\theta}^{old}}(w_i \log p(\boldsymbol{y}_i \mid \boldsymbol{t}_i, \lambda_1 = j_1, ..., \lambda_m = j_m, \boldsymbol{\theta})) \\
&+ \sum_{i \in \boldsymbol{O}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\Gamma}=\boldsymbol{J},\boldsymbol{\theta}^{old}}(w_i \log p(\boldsymbol{t}_i \mid \boldsymbol{\theta})) \\
&+ \sum_{i \in \boldsymbol{O}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times w_i \times [\log p(\lambda_1 = j_1) + \cdots + \log p(\lambda_m = j_m)] \\
&+ \sum_{i \in \boldsymbol{M}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{\theta}^{old}) E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{\Gamma}=\boldsymbol{J},\boldsymbol{\theta}^{old}}(w_i \log p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{\theta})) \\
&+ \sum_{i \in \boldsymbol{M}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_\lambda}, \boldsymbol{\theta}^{old}) E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{\Gamma}=\boldsymbol{J},\boldsymbol{\theta}^{old}}(w_i \log p(\boldsymbol{t}_i \mid \boldsymbol{\theta})) \\
&+ \sum_{i \in \boldsymbol{M}} \sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \times w_i \times [\log p(\lambda_1 = j_1) + \cdots + \log p(\lambda_m = j_m)] \\
&= Q_1 + Q_2 + Q_3 + Q_4 + Q_1^{'} + Q_2^{'} + Q_3^{'}
\end{aligned} \tag{3.11}$$

where $Q_i \forall i \in 1, 2, 3, 4$ denotes the terms corresponding to the observed data or labeled measurements and $Q_i^{'} \forall i \in 1, 2, 3$ denotes the terms corresponding to missing or unlabeled data. To expand the terms in $Q - function$, the distributions given in Eqs. (3.12)-(3.14) are used.

$$p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{P}\boldsymbol{t}_i + \boldsymbol{\mu}_x, \sigma_x^2 \boldsymbol{I}) \tag{3.12}$$

$$p(\boldsymbol{y}_i \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{C}\boldsymbol{t}_i + \boldsymbol{\mu}_y, \sigma_y^2 \boldsymbol{I}) \tag{3.13}$$

$$p(\boldsymbol{t}_i \mid \boldsymbol{\theta}) \sim \mathcal{N}(0, \boldsymbol{I}) \tag{3.14}$$

Now, to calculate the terms of $E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\Gamma}=J,\boldsymbol{\theta}^{old}}(.)$ and $E_{\boldsymbol{X}_\lambda,\boldsymbol{T}|\boldsymbol{X}_{o_\lambda},\boldsymbol{\Gamma}=J,\boldsymbol{\theta}^{old}}(.)$ in Eq. (3.11), the posterior probabilities (derived using Bayes rule) given in Eqs. (3.15)-(3.30) are needed.

For labeled data:

$$p(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i_{o_\lambda}}|t_i,\lambda_1,\cdots,\lambda_m,\boldsymbol{\theta}^{old})p(\boldsymbol{y}_i|t_i,\lambda_1,\cdots,\lambda_m,\boldsymbol{\theta}^{old})p(\boldsymbol{t}_i|\lambda_1,\cdots,\lambda_m,\boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i|\lambda_1,\cdots,\lambda_m,\boldsymbol{\theta}^{old})} \tag{3.15}$$

From Eqs. (3.12)-(3.14), it can be observed that Eq. (3.15) follows a Gaussian distribution. Thus, the mean, and the variance are defined by following a similar approach as detailed in [26] and is given in Eq. (3.16).

$$\begin{aligned} E(\boldsymbol{t}_i\boldsymbol{t}_i^T \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) &= \xi + E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) \times E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old})^T \\ E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) &= \xi \times (\sigma_{x,o}^{-2}\boldsymbol{P}_o^T(\boldsymbol{x}_{i_{o_\lambda}} - \boldsymbol{\mu}_{x,o}) + \sigma_y^{-2}\boldsymbol{C}^T(\boldsymbol{y}_i - \boldsymbol{\mu}_y)) \end{aligned} \tag{3.16}$$

where $\xi = (\sigma_{x,o}^{-2}\boldsymbol{P}_o^T\boldsymbol{P}_o + \sigma_y^{-2}\boldsymbol{C}^T\boldsymbol{C} + \boldsymbol{I})^{-1}$.

Similarly for the unlabeled data,

$$p(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}^{old}) = \frac{p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}^{old})p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}^{old})}{p(\boldsymbol{x}_{i_{o_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{\theta}^{old})} \tag{3.17}$$

and the mean and variance for Eq. (3.17) are provided by Eq. (3.18).

$$\begin{aligned} E(\boldsymbol{t}_i\boldsymbol{t}_i^T \mid \boldsymbol{x}_{i_{o_\lambda}}, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) &= \xi' + E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) \times E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old})^T \\ E(\boldsymbol{t}_i \mid \boldsymbol{x}_{i_{o_\lambda}}, \lambda_1, ..., \lambda_m, \boldsymbol{\theta}^{old}) &= \xi' \times (\sigma_{x,o}^{-2}\boldsymbol{P}_o^T(\boldsymbol{x}_{i_{o_\lambda}} - \boldsymbol{\mu}_{x,o})) \end{aligned} \tag{3.18}$$

where $\xi' = (\sigma_{x,o}^{-2}\boldsymbol{P}_o^T\boldsymbol{P}_o + \boldsymbol{I})^{-1}$.

Alongside these expressions, the posterior probability of the hidden variable x_{i_λ} for both labeled and unlabeled data is essential and is given by Eqs. (3.19) and (3.22) respectively.

$$\begin{aligned} p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) =& \frac{p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_\lambda}, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})p(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})}{p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})} \\ =& \frac{p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})p(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})}{p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})} \end{aligned} \tag{3.19}$$

Since $x_{i_{m_\lambda}}$ is not measurable, the following approximation is considered to make the problem tractable.

$$p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \approx p(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \tag{3.20}$$

By substituting Eq. (3.20) in Eq. (3.19), the following approximation can be obtained:

$$p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \approx p(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \tag{3.21}$$

Similarly, the approximation of the posterior probability of x_{i_λ} for unlabeled data is given as follows:

$$p(\boldsymbol{x}_{i_\lambda} \mid \boldsymbol{t}_i, \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \approx p(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \tag{3.22}$$

The sufficient statistics of Eqs. (3.21) and (3.22) i.e, the mean values and covariances, can be obtained by combining the distributions of both observed and missing input variables as suggested in [64]. Thus, for the labeled data the mean value can be obtained as:

$$E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \begin{bmatrix} \boldsymbol{x}_{i_{o_\lambda}} \\ E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{3.23}$$

For the unlabeled data, the mean value is

$$E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) = \begin{bmatrix} \boldsymbol{x}_{i_{o_\lambda}} \\ E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{3.24}$$

The expectation terms in the right hand side of the Eqs. (3.23) and (3.24) can be calculated through the input model equation stated in Eq. (3.2) and for the labeled data, the mean value is as follows:

$$E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \boldsymbol{P}_m E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) + \boldsymbol{\mu}_{x,m} \tag{3.25}$$

For the unlabeled data, the expectation term is calculated as:

$$E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) = \boldsymbol{P}_m E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) + \boldsymbol{\mu}_{x,m} \tag{3.26}$$

The covariance of x_{i_λ} for labeled data can be obtained as:

$$\mathrm{cov}(\boldsymbol{x}_{i_\lambda}, \boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{cov}(\boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{3.27}$$

and for the unlabeled data,

$$\mathrm{cov}(\boldsymbol{x}_{i_\lambda}, \boldsymbol{x}_{i_\lambda} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{cov}(\boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \end{bmatrix} \tag{3.28}$$

where,

$$\begin{aligned} &\mathrm{cov}(\boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = \boldsymbol{P}_m[E(\boldsymbol{t}_i\boldsymbol{t}_i^T \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \\ &- E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})^T]\boldsymbol{P}_m^T + \sigma_{x,m}^2 \boldsymbol{I} + \\ &E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})^T \end{aligned} \tag{3.29}$$

and

$$\begin{aligned}&\operatorname{cov}(\boldsymbol{x}_{i_{m_\lambda}}, \boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) = \boldsymbol{P}_m[E(\boldsymbol{t}_i\boldsymbol{t}_i^T \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})\\ &- E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})E(\boldsymbol{t}_i \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})^T]\boldsymbol{P}_m^T + \sigma_{x,m}^2\boldsymbol{I} + \\ &E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})E(\boldsymbol{x}_{i_{m_\lambda}} \mid \lambda_1, \cdots, \lambda_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})^T\end{aligned} \tag{3.30}$$

Finally, the posterior probabilities of variables' delay for labeled and unlabeled data are given in Eqs. (3.31) and (3.32), respectively.

$$\begin{aligned}G_o = p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) &= \frac{p(\boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}{\sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}\\ &= \frac{\prod_{i\in\boldsymbol{O}} p(\boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}{\sum_{j_1=d_1^1}^{d_2^1} \ldots \sum_{j_m=d_1^m}^{d_2^m} \prod_{i\in\boldsymbol{O}} p(\boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}\end{aligned} \tag{3.31}$$

where, $p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old}) = p(\lambda_1 = j_1 \mid \boldsymbol{\theta}^{old}) \times \cdots \times p(\lambda_m = j_m \mid \boldsymbol{\theta}^{old})$, and

$$p(\boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{x,o}\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{P}_o\boldsymbol{P}_o^T + \sigma_{x,o}^2\boldsymbol{I}_o & \boldsymbol{P}_o\boldsymbol{C}^T\\ \boldsymbol{C}\boldsymbol{P}_o^T & \boldsymbol{C}\boldsymbol{C}^T + \sigma_y^2\boldsymbol{I}\end{bmatrix}\right)$$

Similarly,

$$\begin{aligned}G_m = p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) &= \frac{p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}{\sum_{j_1=d_1^1}^{d_2^1} \cdots \sum_{j_m=d_1^m}^{d_2^m} p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}\\ &= \frac{\prod_{i\in\boldsymbol{M}} p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}{\sum_{j_1=d_1^1}^{d_2^1} \ldots \sum_{j_m=d_1^m}^{d_2^m} \prod_{i\in\boldsymbol{M}} p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \times p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old})}\end{aligned} \tag{3.32}$$

where $p(\boldsymbol{\Gamma} = \boldsymbol{J} \mid \boldsymbol{\theta}^{old}) = p(\lambda_1 = j_1 \mid \boldsymbol{\theta}^{old}) \times \cdots \times p(\lambda_m = j_m \mid \boldsymbol{\theta}^{old})$ and

$$p(\boldsymbol{x}_{i_{o_\lambda}} \mid \boldsymbol{\Gamma} = \boldsymbol{J}, \boldsymbol{\theta}^{old}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x,o}, \boldsymbol{P}_o\boldsymbol{P}_o^T + \sigma_{x,o}^2\boldsymbol{I}_o\right)$$

The final step is to provide the prior probability of $p(\lambda_z = j_z \mid \boldsymbol{\theta}^{old})$. In the absence of prior knowledge, the simplest and efficient choice is to assume that the initial probabilities follow a uniform distribution and then update the probabilities through Eq. (3.33).

$$p(\lambda_z = L) = \frac{\sum G_o^{z,L} + \sum G_m^{z,L}}{\sum G_o + \sum G_m} \tag{3.33}$$

where $1 \le z \le m$ and $d_1^z \le L \le d_2^z$, and $G_o^{z,L}$ and $G_m^{z,L}$ denote the z^{th} delay variable has value L where G_o and G_m are defined in Eqs. (3.31) and (3.32), respectively. Unlike the existing methods, the probability value defined in Eq. (3.33) is updated at every iteration for faster convergence and better accuracy in the proposed algorithm.

In most cases, it is not feasible to obtain efficient bounds on the initial limits of delays and one typically considers a wide range of delay. Thus, a strategy of updating this range of delays is proposed in this chapter. In the proposed approach, the starting and the end values of the range for each input delay are modified based on their difference from the most probable delay in the current iteration. In other words, the search starts from the beginning and the end of the range simultaneously, and each candidate delay value will be removed until the criterion given in Eq. (3.34), is met.

$$p(\lambda_z = d_1^z) \text{ or } p(\lambda_z = d_2^z) \geq \Upsilon \times \max p(\lambda_z) \tag{3.34}$$

In Eq. (3.34), $\Upsilon \in [0, 1)$ is the tuning parameter that controls the range of removal of delays, i.e. when $\Upsilon = 0$, then the delays in the initial range will remain the same throughout, and when $\Upsilon = 1$, the most probable delay will only be present. It has to be noted that this idea of updating the range is only applicable to the case of a fixed delay in each input. In the case when the criterion in Eq. (3.34) is met, the search is stopped, and the range of delay will be updated from $d_1 = d_1^{new}$ to $d_2 = d_2^{new}$ as shown in Fig. 3.2. This step of updating the delay range will help in faster convergence.

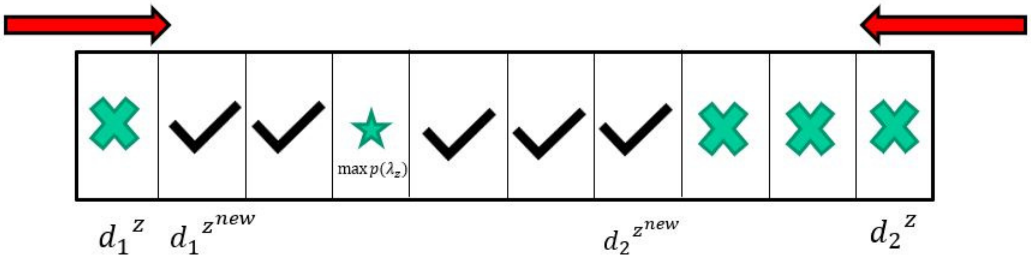


Figure 3.2: Search algorithm representation

After calculating all the terms in $Q - function$ (Eq. (3.11)), the parameters are estimated by taking derivatives in the maximization step and are given as follows:

$$\boldsymbol{P} : \frac{\partial(Q_1 + Q_1^{'})}{\partial \boldsymbol{P}} = 0 \qquad \boldsymbol{\mu}_x : \frac{\partial(Q_1 + Q_1^{'})}{\partial \boldsymbol{\mu}_x} = 0 \qquad \sigma_x^2 : \frac{\partial(Q_1 + Q_1^{'})}{\partial \sigma_x^2} = 0$$

$$\boldsymbol{C} : \frac{\partial(Q_2)}{\partial \boldsymbol{C}} = 0 \qquad \boldsymbol{\mu}_y : \frac{\partial(Q_2)}{\partial \boldsymbol{\mu}_y} = 0 \qquad \sigma_y^2 : \frac{\partial(Q_2)}{\partial \sigma_y^2} = 0$$

and the updating equations for the parameters are given in Eqs. (3.35)- (3.40)

The weighting matrices $\boldsymbol{P}$ and $\boldsymbol{C}$ are updated based on Eqs. (3.35) and (3.36), respectively.

$$\begin{aligned}\boldsymbol{P}^{new} =& [\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})\\ &\times(2(E(\boldsymbol{x}_{i_\lambda}\mid\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)E(\boldsymbol{t}_i\mid\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T)\\ &+\sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})\\ &\times(2(E(x_{i_\lambda}\mid\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)E(\boldsymbol{t}_i\mid\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})^T)]\\ &\times[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(B)\\ &+\sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(B^{'})]^{-1}\end{aligned}\tag{3.35}$$

where

$$A=E(\boldsymbol{t}_i\boldsymbol{t}_i^T\mid\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})-E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})^T$$

$$A^{'}=E(\boldsymbol{t}_i\boldsymbol{t}_i^T\mid\boldsymbol{x}_{i_{o_\lambda}},\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})-E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})^T$$

$$B=A+A^T+2E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})^T$$

$$B^{'}=A^{'}+A^{'T}+2E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})E(\boldsymbol{t}_i\mid\boldsymbol{x}_{i_{o_\lambda}},\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{\theta}^{old})^T$$

$$\begin{aligned}\boldsymbol{C}^{new} =& \left[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})\right.\\ &\left.\times(2(\boldsymbol{y}_i-\boldsymbol{\mu}_y)E(\boldsymbol{t}_i\mid\lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T)\right]\\ &\times\left[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m\mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(B)\right]^{-1}\end{aligned}\tag{3.36}$$

Similarly, the update equations for input and output covariances are provided in

Eqs. (3.37) and (3.38), respectively.

$$
\begin{aligned}
\sigma_x^{2new} = &\Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(S_x)\\
&\sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(S_x^{'})\Bigg]\\
&\times\Bigg[m\times(\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})\\
&+\sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi}))\Bigg]^{-1}
\end{aligned}
\tag{3.37}
$$

where

$$
\begin{aligned}
S_x = &(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)^T(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)\\
&-E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T\boldsymbol{P}^T(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)\\
&-(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)^T\boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})+tr(\boldsymbol{P}^T\boldsymbol{P}(A))\\
&+E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T\boldsymbol{P}^T\boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})
\end{aligned}
$$

$$
\begin{aligned}
S_x^{'} = &(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)^T(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)\\
&-E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})^T\boldsymbol{P}^T(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)\\
&-(E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})-\boldsymbol{\mu}_x)^T\boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})+tr(\boldsymbol{P}^T\boldsymbol{P}(A^{'}))\\
&+E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})^T\boldsymbol{P}^T\boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{\theta}^{old})
\end{aligned}
$$

and:

$$
\begin{aligned}
\sigma_y^{2new} = &\Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi})(S_y)\Bigg]\\
&\times\Bigg[r\times(\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1=j_1 \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\cdots p(\lambda_m=j_m \mid \boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})\times\exp(\frac{-d_i^2}{\phi}))\Bigg]^{-1}
\end{aligned}
\tag{3.38}
$$

where

$$
\begin{aligned}
S_y = &(\boldsymbol{y}_i-\boldsymbol{\mu}_y)^T(\boldsymbol{y}_i-\boldsymbol{\mu}_y)-E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T\boldsymbol{C}^T(\boldsymbol{y}_i-\boldsymbol{\mu}_y)\\
&-(\boldsymbol{y}_i-\boldsymbol{\mu}_y)^T\boldsymbol{C}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})+tr(\boldsymbol{C}^T\boldsymbol{C}(A))\\
&+E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})^T\boldsymbol{C}^T\boldsymbol{C}E(\boldsymbol{t}_i \mid \lambda_1=j_1,\cdots,\lambda_m=j_m,\boldsymbol{x}_{i_{o_\lambda}},\boldsymbol{y}_i,\boldsymbol{\theta}^{old})
\end{aligned}
$$

Finally, the updating equations for the input and output mean are derived in Eqs (3.39) and (3.40), respectively.

$$\begin{aligned}
\boldsymbol{\mu}_x^{new} = & \Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi}) \\
& \times (E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) - \boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})) \\
& + \sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi}) \\
& \times (E(\boldsymbol{x}_{i_\lambda} \mid \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) - \boldsymbol{P}E(\boldsymbol{t}_i \mid \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}))\Bigg] \\
& \times \Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi}) \\
& + \sum_{i\in\boldsymbol{M}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi})\Bigg]^{-1}
\end{aligned} \tag{3.39}$$

$$\begin{aligned}
\boldsymbol{\mu}_y^{new} = & \Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi}) \\
& \times (\boldsymbol{y}_i - \boldsymbol{C}E(\boldsymbol{t}_i \mid \lambda_1 = j_1, \cdots, \lambda_m = j_m, \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{\theta}^{old}))\Bigg] \\
& \times \Bigg[\sum_{i\in\boldsymbol{O}}\sum_{j_1=d_1^1}^{d_2^1}\cdots\sum_{j_m=d_1^m}^{d_2^m} p(\lambda_1 = j_1 \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})\cdots p(\lambda_m = j_m \mid \boldsymbol{x}_{i_{o_\lambda}}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times \exp(\frac{-d_i^2}{\phi})\Bigg]^{-1}
\end{aligned} \tag{3.40}$$

The parameters in Eqs. (3.35)-(3.40) are estimated in an iterative manner by using the updated posterior probabilities given in Eqs. (3.12)-(3.32) till the estimates converge.

3.2.4 Online Predictions

For the application of soft sensor, it is necessary to predict the variables online. The term given in Eq. (3.41) is used to perform the online predictions given a new data point, $\boldsymbol{x}_q$.

$$\hat{\boldsymbol{y}}^{new} = \boldsymbol{C}\hat{\boldsymbol{t}}^{new} + \boldsymbol{\mu}_y \tag{3.41}$$

Now, to estimate $\hat{\boldsymbol{t}}^{new}$ in Eq. (3.41), the posterior probability of the latent variable given the input information (new data point) is required. This posterior probability is given as

$$p\left(\boldsymbol{t}^{new} \mid \boldsymbol{x}_q\right) = \frac{p\left(\boldsymbol{t}^{new}\right)p\left(\boldsymbol{x}_q \mid \boldsymbol{t}^{new}\right)}{p\left(\boldsymbol{x}_q\right)} \tag{3.42}$$

From Eq. (3.42), it can be observed that the latent variable follows a Gaussian distribution and hence $\hat{\boldsymbol{t}}^{new}$ is estimated by the mean value of the distribution as,

$$\hat{\boldsymbol{t}}^{new} \ = E(\boldsymbol{t}^{new} \mid \boldsymbol{x}_q) = (\sigma_x^{-2}\boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{I})(\sigma_x^{-2}\boldsymbol{P}^T(\boldsymbol{x}_q - \boldsymbol{\mu}_x)) \tag{3.43}$$

The predictions of the quality variables i.e., $\hat{\boldsymbol{y}}^{new}$ is obtained by substituting Eq. (3.43) in Eq. (3.41).

3.3 Case Studies

In this section, the performance of the proposed algorithm is demonstrated by considering a numerical example, first. An experiment on a hybrid tank pilot plant system is performed to demonstrate the practical applicability of the proposed algorithm.

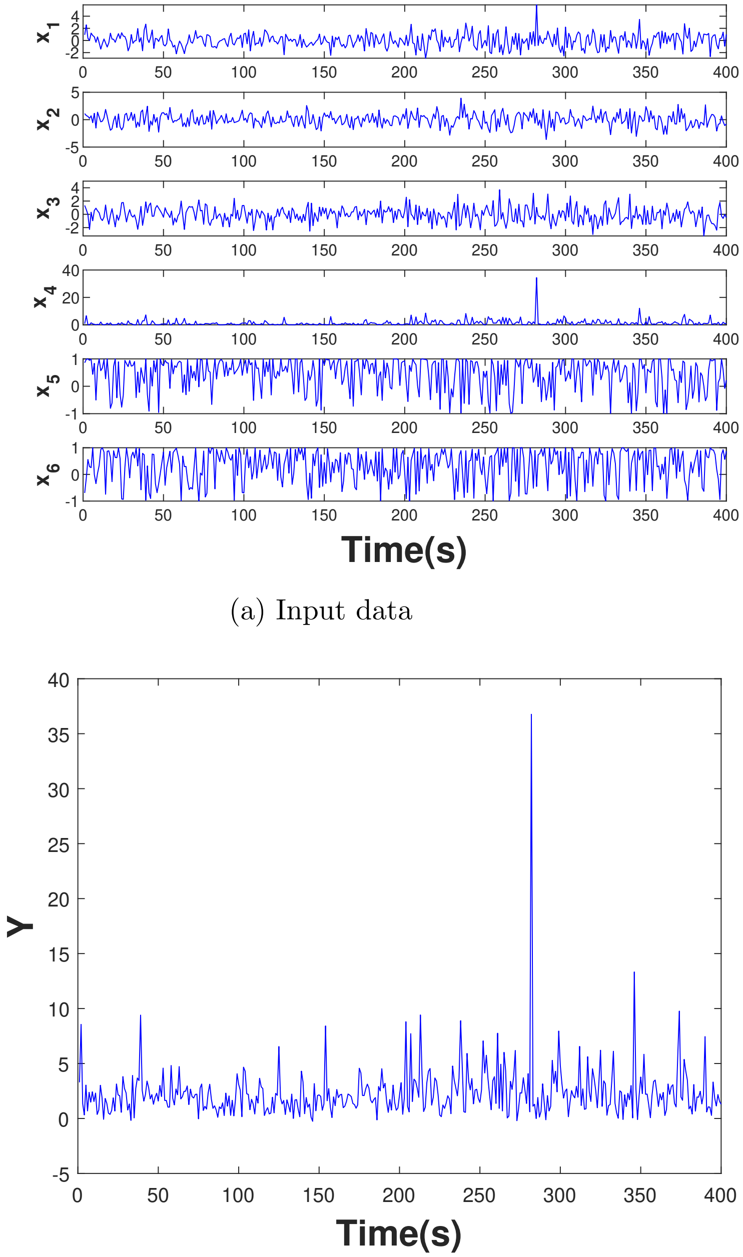
3.3.1 Numerical Example

A two-operating-mode problem with 6 input variables and one output variable as given in Eq. (3.44) is considered.

$$\begin{aligned}
&\text{mode 1}: \ x_i \sim \mathcal{N}(0, \sigma_1) && i = 1, 2, 3 \\
&\text{mode 2}: \ x_i \sim \mathcal{N}(0, \sigma_2) + 1.2 && i = 1, 2, 3 \\
&x_4 = x_1^2 \qquad x_5 = sin(x_2 + 1) && x_6 = cos(x_3 + 1) \\
&\qquad y = x_1^2 + \exp(x_2/3) + sin(x_3)
\end{aligned} \tag{3.44}$$

where σ_1 is 0.1 and σ_2 is 0.3 [54]. After generating the data using Eq. (3.44), the input variables 1 to 6 are shifted artificially by $\{4, 2, 1, 5, 3, 2\}$ samples respectively to account for the delay factor. A total of 400 samples (200 samples for each mode) are generated, of which, 300 samples are used for training, and 100 samples are used for validation. The trend of the input variables and the output variable is presented in Fig. 3.3.

While developing a weighted PPCR model, the number of relevant samples is chosen as 20, and the dimension of the latent variable is determined as 3 by doing cross validation analysis [10]. To demonstrate the superiority of the proposed algorithm, a comparative study is performed with the cross correlation (CC) analysis [65], which is one of the most prominent methods in identifying time-delays and also with a regular PPCR method [66]. Though CC analysis is one of the prominent methods, it is well known that the performance of this method degrades in the presence of missing and/or high noises [67]. The results of the comparative study in the presence of



(a) Input data

(b) Output data

Figure 3.3: Generated data for numerical example.

different amount of missing data are provided in Table 3.1 which are the mean values after a hundred iterations with different initialization, and the prediction results for the case of 15% missing in input and 30% missing in output are shown through a scatter plot in Fig. 3.4. The modes of the estimated values of the time-delays with the proposed algorithm are $\{4, 2, 1, 5, 3, 2\}$.

Table 3.1: RMSE and R^2 values under various missing inputs and outputs data

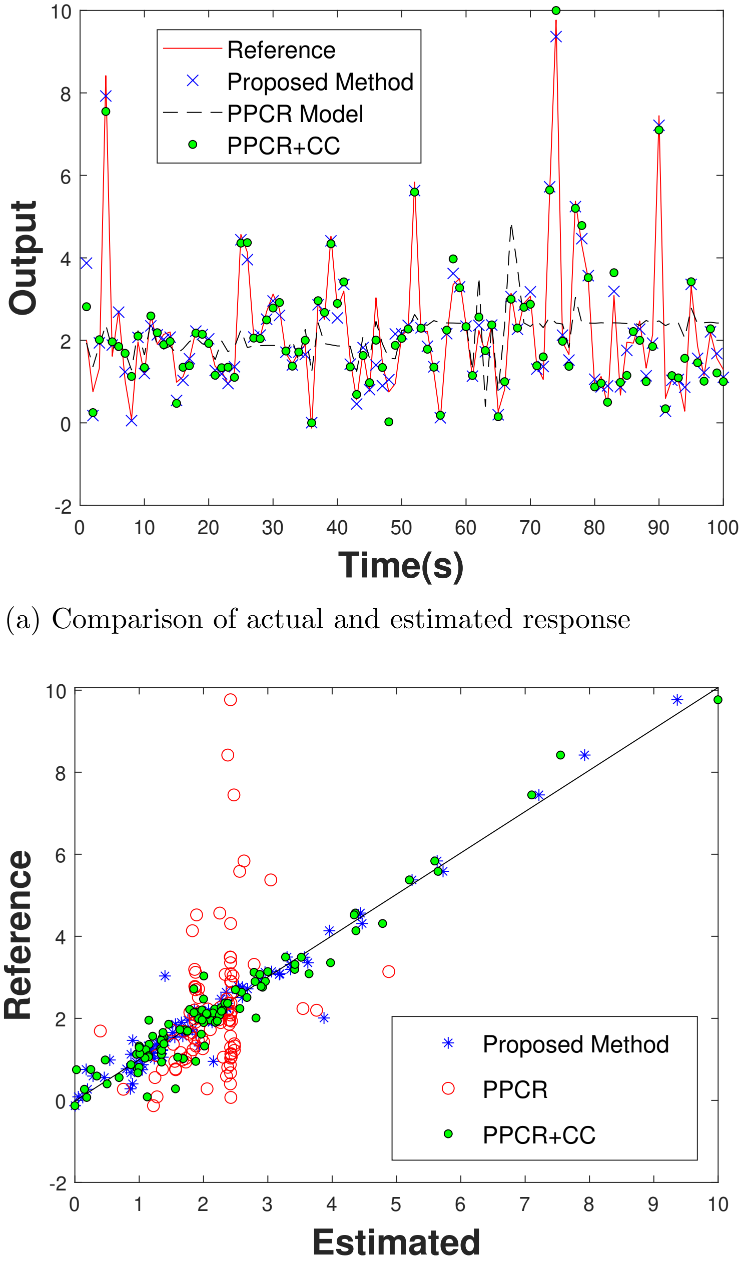
	No missing input						15%					
	Proposed Model		PPCR		PPCR+CC		Proposed Model		PPCR		PPCR+CC	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
No missing output	0.2831	0.9415	0.9924	0.4231	0.3072	0.9342	0.2543	0.9406	1.003	0.403	0.4642	0.9222
5%	0.2842	0.9399	1.0970	0.3942	0.6788	0.8990	0.2973	0.9379	1.1438	0.3869	0.6822	0.8978
30%	0.3095	0.9351	1.283	0.3641	0.7236	0.8804	0.3442	0.9245	1.5270	0.3257	0.8729	0.8664

From the results presented in Table 3.1 and Fig. 3.4, it can be concluded that the proposed algorithm is able to provide better performance and more accurate estimations compared to the traditional methods due to its ability in handling noise, missing data and identifying time-delays.

3.3.2 Experimental study: Hybrid tank pilot plant system

A hybrid tank pilot plant system, as shown in Fig. 3.5, is considered for demonstrating the practical applicability of the proposed algorithm. A hybrid tank system consists of three cylindrical tanks connected by six valves, namely V_1-V_4, V_6, and V_8, and a container at the bottom of these three tanks to collect the tanks' outflow. Three level-sensors, namely LT_1, LT_2 and LT_3, are used to measure the level in each of these tanks. The tanks are connected to the container through three valves, namely V_5, V_7, and V_9. Two similar pumps are also connected to this container to send the water into Tanks 1 and 3, and the flowrate of these streams is considered as manipulated variables. The level in Tank 2 is regarded as the variable that is of interest. A more detailed explanation of this hybrid tank pilot plant system can be found in [68, 49].

To account for the multi-modal nature of processes, two different operating regions are considered by manipulating the inlet flowrates of Tanks 1 and 3 around 4.75 and 5.15 for the first operating region that is called normal operating mode and around 6.15 and 6 for the second operating point that is called the abnormal operating mode, respectively, where the phenomena of overflow might arise. The input and output data for the mentioned operating condition are shown in Figures 3.6(a) and 3.6(b), respectively. The data consist of nearly 1100 samples, of which 600 samples are used for training the model, and 500 samples are used for validation. To demonstrate the efficacy of the proposed algorithm for modeling with time-delay in data, an artificial time-delay of 6 is introduced to the left flowrate, and the right flowrate has the time-delay of 2. A comparative study is made among the proposed algorithm, PPCR, and PPCR with the help of CC to identify the delays before modeling where 10% of input variables and 45% of the output variable are missed. The mean values



(a) Comparison of actual and estimated response

(b) Parity plot

Figure 3.4: Online prediction performance of the proposed algorithm on the numerical data with delayed and 15% missing input data and 30% missing in output.

after 100 different initializations are provided in Table 3.2, and the estimated values are visualized in Fig. 3.7.

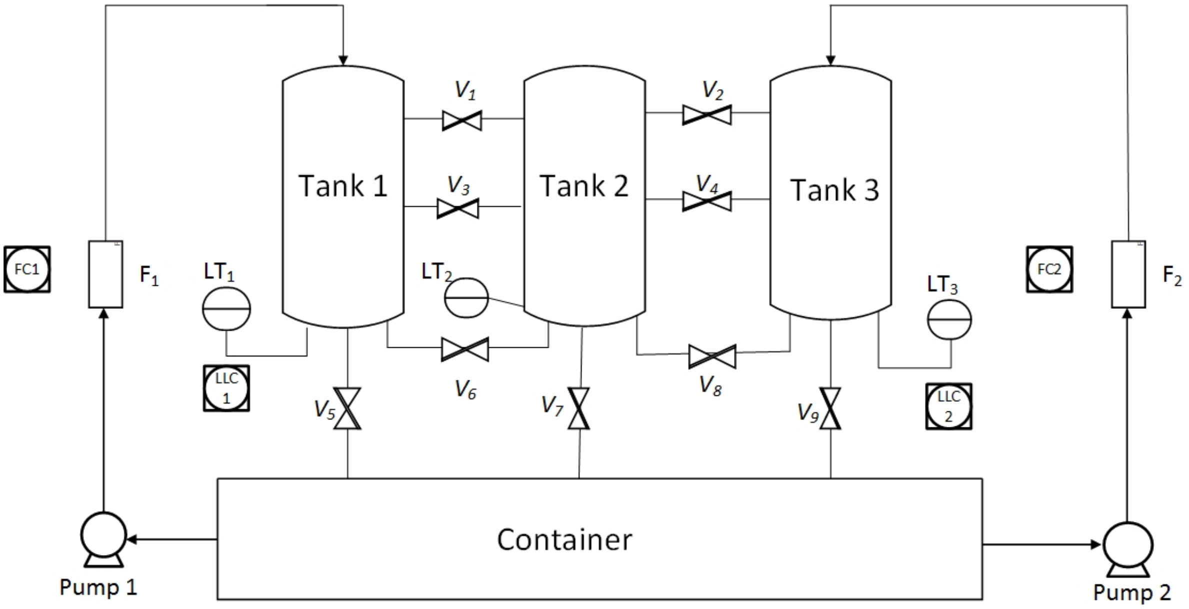


Figure 3.5: Schematic of the Hybrid Tank Pilot Plant

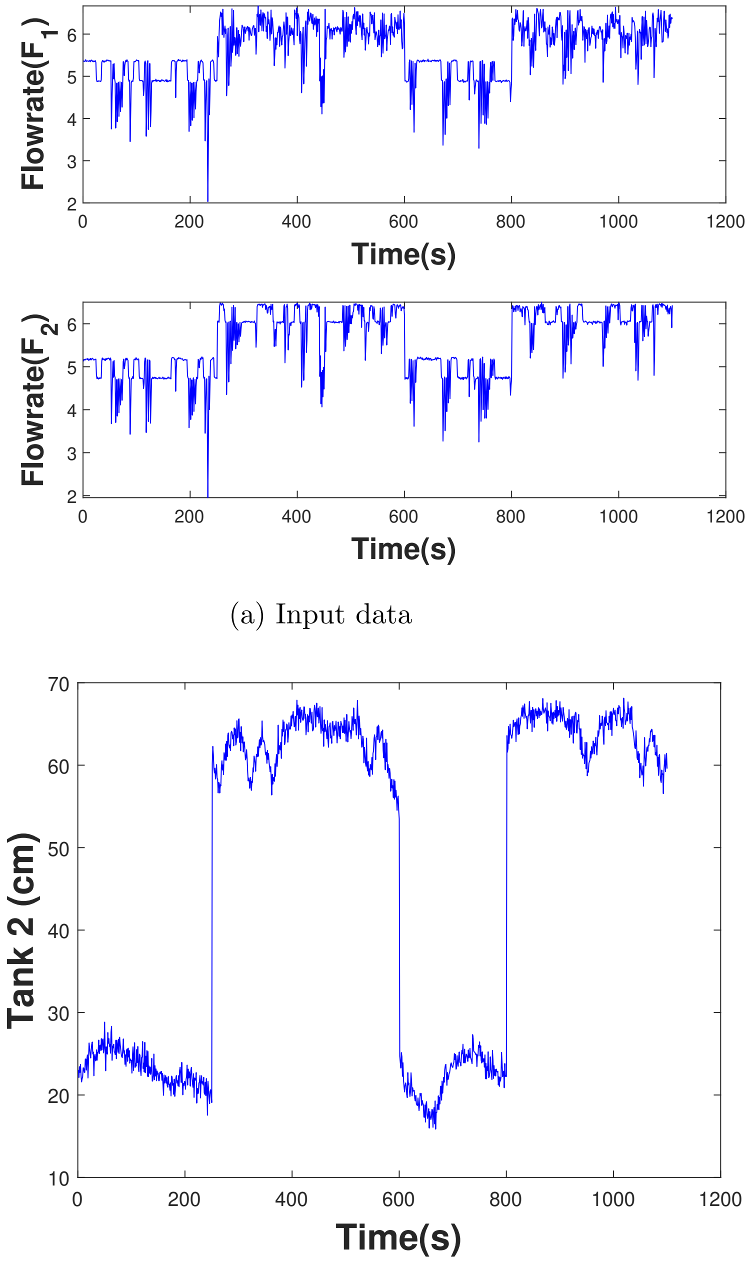
Table 3.2: $RMSE$ and R^2 values of three different algorithms when 10% of input variables and 45% of output variable are missed

	proposed algorithm	PPCR	PPCR+CC
R^2	0.8427	0.4476	0.7125
$RMSE$	1.1271	2.8515	1.5246

From the comparison provided in Table 3.2 and Fig. 3.7, it can be concluded that the proposed algorithm is able to deal with the presence of constant but different time-delay for each input variable while building the PPCR model. Moreover, the proposed algorithm is able to provide more accurate model compared to the other algorithms in the presence of missing values in input and output variables. In addition, the mode of the estimated time-delays are $\{5, 2\}$ for the left and the right flowrate, respectively that is very close to the original values, $\{6, 2\}$.

3.4 Conclusion

In this chapter, a weighted semi-supervised PPCR (WSSPPCR) model with missing input and delayed output data for modeling non-linear and/or multi-modal processes is developed. The proposed algorithm can deal with missing input and output data with the help of data imputation and semi-supervised learning, respectively. It can also efficiently cope with the existence of time-delays between each input and out-



(a) Input data

(b) Output data

Figure 3.6: Collected data for experimental example.

put variable. Further, to optimize the time-delay range for each input variable and speed up the modeling process with less computational effort, a searching approach is proposed. The performance of the proposed algorithm is demonstrated through a numerical example and an experimental study. From the comparative studies between the proposed algorithm, regular PPCR, and PPCR with CC analysis for the delay, it

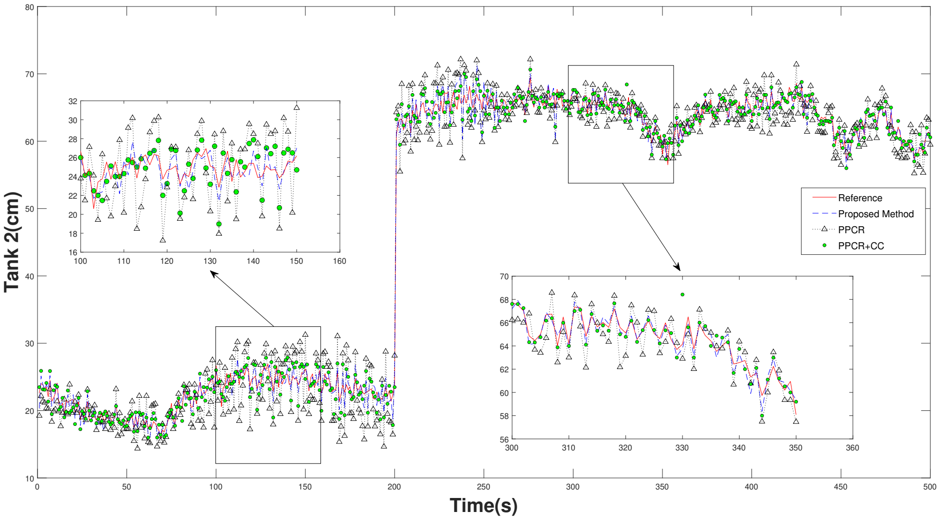


Figure 3.7: Comparison of output response with 45% missing in output and 10% missing in input

can be observed that the proposed algorithm has a better performance compared to all the competing methods.

Chapter 4

Data-driven self-optimization of process in the presence of the model-plant mismatch[1,2]

4.1 Introduction

Increasing productivity, safety, and efficiency have always been the main objectives of industrial plants. The objective of plant optimization is to reduce or abolish resource wastage and bottlenecks while accomplishing the objective of the plant and meeting all plant constraints, including operational, economic, and safety. Due to the reduction in the availability of the raw materials [69], the increase in the market demand for the products because of escalation in the number of the world's population [70], and the environmental concerns like global warming as a result of the emission of the greenhouse gases (GHG) [71], plant optimization has gained more popularity, recently. One of the approaches to do plant optimization is optimizing through the model.

Plant optimization can be performed based on the development of a model, which is generally obtained through two different approaches, i) first principle model-based optimization and ii) data-driven model-based optimization [12, 13]. In the first prin-

[1] A. Memarian, S. K. Varanasi, B. Huang. "Data-Driven Self-Optimization for plant Operations". Presented in *Canadian Chemical Engineering Conference 2021, October 24-27, Montreal, Quebec, Canada*, 2021 (Chapter 4 - Extended abstract)

[2] A.Memarian, S. K. Varanasi, B. Huang. "Data-driven self-optimization of processes in the presence of the model-plant mismatch". Submitted to 13th *IFAC Symposium on Dynamics and Control of Process Systems, (DYCOPS), June 14-27 2022, Busan, Republic of Korea*, 2022 (Chapter 4 - Short Version)

ciple model-based optimization, the plant is modeled with the help of deriving the governing equations from the fundamental laws, which needs an in-depth understanding of the plant [14]. On the other hand, in data-driven model-based optimization, a model is built based on the historical data. The closer the developed model is to the plant, the more accurate optimal point can be obtained by solving the optimization problem [72]. However, due to the differences between the model and plant (model-plant mismatch) and the disturbances that may occur during the data collection, the optimal point obtained by solving the optimization problem can be different from the true plant's optimal point [73].

To account for the model-plant mismatch in process optimization, the scheme of modifier adaptation is proposed in [74, 75, 76, 77]. In this scheme, the error between the developed model and the plant is incorporated in the objective function while performing the optimization by using the information and measurements collected from the plant. The authors in [74] provided a theorem that demonstrates the equivalence of KKT conditions of the plant and the model with the inclusion of the modifier adapters. They suggested the use of gradients of the objective function and constraints calculated from plant measurements as a functional form of modifier adapter. Although the calculation of gradients from noisy plant measurements can be challenging, it is demonstrated to be a reasonably reliable and effective approach [78]. To overcome the challenges with the calculation of gradients, several methods such as nested modifier adapters [79], recursive modifier adapters [80], and derivative-free modifier adapters [81] are proposed. Recently, the authors in [73] proposed using Gaussian process regression (GPR) as a modifier adapter. In this work, the historical data and real measurements obtained from the plant are used to train the GP; thereby a nonlinear model on the modeling error that accounts for the model-plant mismatch is obtained. The authors in [77] proposed a trust-region framework and the Gaussian process modifier adapters to control the optimization region and to avoid the possibility of violation of constraints. However, the convergence to a local optimal is still a challenging problem in all the aforementioned methods. One of the approaches to overcome this challenge is considering uncertainty in solving the optimization problem [82].

With the fast pace of development in the field of reinforcement learning [83, 84, 85], the concept of self-reflective objective is gaining popularity. In this concept, the accuracy and reliability of the optimization problem are improved by consideration of uncertainty. Although many studies have focused on increasing the accuracy of the scheme of modifier adaptation, the potential of reinforcement learning has not been studied extensively in the modifier adaptation and optimization problems in general. One of the concepts that can help to increase the accuracy of the optimization is acquisition functions that are used in Bayesian optimization and provide the balance between exploration(trying something new) and exploitation(keep doing what has been done) [82]. In all the aforementioned studies, the modifier adaptation scheme is used along with the first principle models, which essentially requires an in-depth understanding of the process and hence, is not always feasible. In addition to finding the optimal point, an efficient way to steer the process to the optimal point is of paramount importance. Trust-region-based Real-time optimization (RTO) is one of the solutions finding an efficient way to the optimal point [86]. However, the application of the data-driven RTO has not been well studied [87, 88].

In view of the aforementioned points, a novel self-optimization algorithm is developed in this work that can find both the plant optimal point and the efficient way to shift from the current operating condition to the determined optimal point. The proposed algorithm considers a generalized weighted PPCR model due to its ability to deal with missing data in both input and output variables, outliers, and time-delays [68, 89]. Since weighted PPCR is a linear model, and the plant is nonlinear in general, a non-linearity index is used to help the local data-driven model to determine its accuracy based on the plant operating point. The proposed non-linearity index measures the mismatch between the locally weighted PPCR model and the nonlinear GPR model. Then, this non-linearity index is used in determining the trust range of the generalized weighted PPCR model; thereby, an increase in the accuracy of the model is obtained. In addition, the GPR is used as a modifier adapter to account for the model-plant mismatch. Finally, the concept of acquisition functions is used in optimization problem to study the significance of exploration.

The remainder of this chapter provides a brief overview on modifier adaptation, Gaussian process regression, and acquisition function in section 4.2. The proposed method of the data-driven self-optimization in the presence of the model-plant mismatch and the study of acquisition functions for exploration is detailed in section 4.3. The efficiency of the algorithm is illustrated through a simulation case study on a deethanizer column and an industrial application to show its applicability and feasibility in section 4.4, and the conclusions are drawn in section 4.5.

4.2 Preliminaries

This section provides details regarding the modifier adaptation scheme, Gaussian process regression model, and acquisition functions.

4.2.1 Model-plant mismatch

To find the optimal point of a steady-state plant, the following optimization problem needs to be solved.

$$\begin{aligned} &\min_u G_0^p(u) := g_0(u, y^p(u)) \\ &\text{s.t. } G_i^p(u) := g_i(u, y^p(u)) \leq 0, \qquad i = 1, \cdots, n_g \end{aligned} \tag{4.1}$$

where G_0^p is the objective function of the plant where the superscript "p" denotes the plant, and $G_i^p, \ \ \forall i = 1, \cdots, n_g$ are the constraints that should be satisfied while solving the optimization problem, with n_g denoting the total number of constraints in the optimization problem.

Finding a mapping between input and output variables to describe the plant with a greater accuracy can be challenging and is not always possible due to the complexity of the process and lack of in-depth understanding of the operations [77]. Therefore, the mapping between input and output variables is modeled with the help of data-driven models in this chapter, and the model-based optimization problem is defined in Eq. (4.2).

$$\begin{aligned} &\min_u G_0(u) := g_0(u, y(u, \theta)) \\ &\text{s.t. } G_i(u) := g_i(u, y(u, \theta)) \leq 0, \qquad i = 1, \cdots, n_g \end{aligned} \tag{4.2}$$

The term $y(u, \theta)$ in Eq. (4.2) represents the mapping between input and output variables, and θ is the parameters of the model.

The model-plant mismatch is frequently observed while developing a model. Hence, the optimal point of Eq. (4.1) is usually different from the one found from the modified optimization problem in Eq. (4.2), as shown in Fig. 4.1. Hence, the authors in [74] proposed using the modifier adaptation scheme for both objective and constraints to account for the model-plant mismatch.

4.2.2 Gaussian process regression

Gaussian process regression (GPR) is a non-parametric modeling approach that is first proposed in [90]. The assumption in GP regression is that any function can be

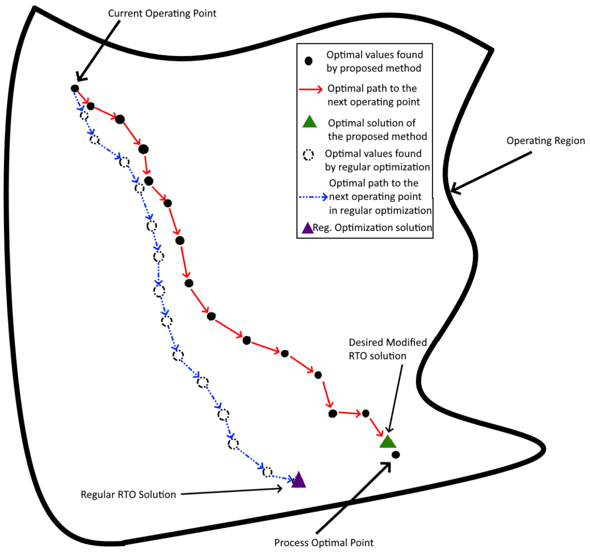

Current Operating Point
Optimal values found by proposed method
Optimal path to the next operating point
Optimal solution of the proposed method
Optimal values found by regular optimization
Optimal path to the next operating point in regular optimization
Reg. Optimization solution
Operating Region
Desired Modified RTO solution
Regular RTO Solution
Process Optimal Point

Figure 4.1: model-plant mismatch effect on optimization

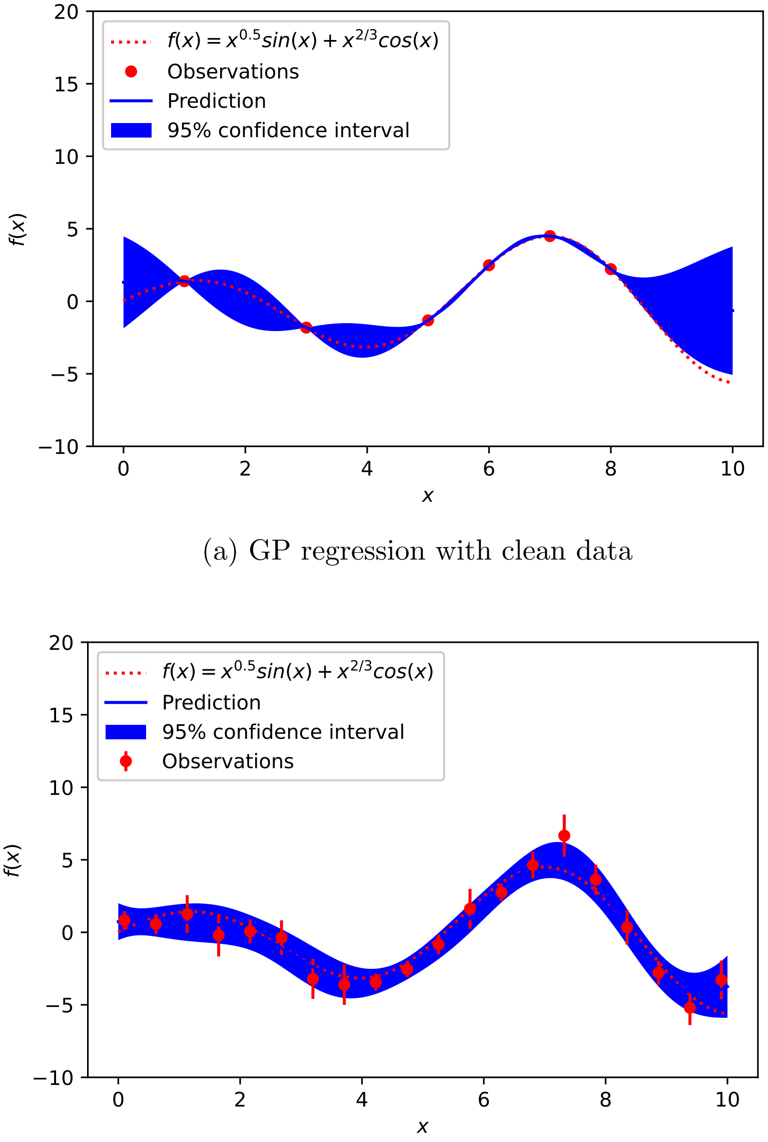
modeled using a combination of multivariate Gaussian distributions in the presence of noisy measurements through a varying number of parameters. This model requires the mean function and the covariance function, where its representation is provided in Eq. (4.3) and Fig.4.2.

$$f(u, y(u, \theta)) \sim \mathcal{GP}(m(u), K(u, u^{'})) \tag{4.3}$$

In Eq. (4.3), $m(u)$ is the mean function and a constant mean value function is used in the current chapter. The $K(u, u^{'})$ is the covariance function calculated based on different positive definite kernel matrices. There are different choices, like Squared Exponential Kernel, Rational Quadratic Kernel, Periodic Kernel, Locally Periodic Kernel, Linear Kernel [91]. In Eq. (4.4), the derivations of the squared exponential kernel and rotational quadratic kernel, which are among the most popular kernel matrices, are provided [91]. In this work, the squared exponential kernel is considered because it can be utilized for most functions and has fewer parameters.

$$\begin{aligned} K_{SE}(u, u^{'}) &:= \sigma^2 \exp(-\frac{1}{2}(u - u^{'})^T \Lambda (u - u^{'})) \\ K_{RQ}(u, u^{'}) &:= \sigma^2 \left(1 + \frac{1}{2\alpha}(u - u^{'})^T \Lambda (u - u^{'})\right)^{-\alpha} \end{aligned} \tag{4.4}$$

One of the challenges with the industrial datasets is the presence of the outliers that may affect the accuracy of the model in Eq. (4.3). The authors in [92] proposed a robust GP regression that can efficiently handle outliers. This method is utilized in the proposed optimization.



(a) GP regression with clean data

(b) GP regression with noisy data

Figure 4.2: Mean prediction with the prediction interval by 2 standard deviation from the mean.

4.2.3 Acquisition functions

Acquisition functions are mathematical equations that account for the exploration of the parameter space in an algorithm in addition to the exploitation. They use the predicted mean and predicted variance generated by the Gaussian process regression model. Exploitation consists of searching the limited parameter space and hoping for improving the in-hand solution. However, exploration pushes the search area to a larger space to find better solutions towards the unexplored regions.

There are different types of acquisition functions proposed in the literature. Among the proposed acquisition functions, expected improvement [93], lower or upper confidence bound [94], probability of improvement [95], entropy search [96] are the most commonly used. Among the mentioned acquisition functions, lower confidence bound (LCB) is the most common function used in the literature due to its simplicity [97]. LCB tries to consider both the exploitation (GP's mean) and the exploration (GP's variance) at the same time to improve the solution of optimization by considering the uncertainty to minimize the regret and loss while using the Bayesian optimization. In Eq. (4.5), the LCB acquisition function is presented.

$$\mathcal{A}_{LCB}[\mu, \sigma](u) := \mu_f(u) - \beta\sigma(u) \tag{4.5}$$

where μ is the estimated mean, and σ is the estimated uncertainty term. $\beta \in [0, \infty)$ is the exploration tuning parameter. Most acquisition functions have an exploration parameter that defines how much exploration is desired and needs to be tuned to obtain the best solution. By setting $\beta = 0$, it is dictated that no exploration will be added to the optimization problem.

4.3 Data-driven self-optimization of processes in the presence of the model-plant mismatch

In this section, the data-driven self-optimization of processes in the presence of the model-plant mismatch is presented. The proposed approach utilizes a generalized weighted PPCR model that can handle the missing data in both input and output variables, time-delay, and outliers in data. Moreover, due to its weighted local model property, it can efficiently handle the nonlinearity and/or multi-modal nature of plants [68, 89]. A robust Gaussian process regression model is used to account for the model-plant mismatch between the weighted PPCR model and the plant. To balance the exploitation and exploration terms in the optimization problem, the lower confidence bound is used as an acquisition function both in the objective and constraint functions in this chapter. The details are provided in the rest of this section.

4.3.1 Generalized weighted PPCR model formulation

The most important step while solving an optimization problem is to build a suitable model that can describe the plant with sufficient accuracy. Hence, data-driven modeling is one of the approaches that can help. In the proposed self-optimization algorithm, a generalized weighted PPCR model is used as a data-driven model to mimic the plant that is described in [68, 89]. The generalized weighted PPCR model is one of the simplest models yet effective in dealing with different possible uncertainties in the plant's datasets.

The generative equation for the generalized weighted PPCR model is presented in Eq. (4.6).

$$\begin{aligned} \boldsymbol{x}_{i_\lambda} &= \boldsymbol{P}\boldsymbol{t}_i + \boldsymbol{e}_i + \boldsymbol{\mu}_x, \quad i = 1, 2, \cdots, n \\ \boldsymbol{y}_j &= \boldsymbol{C}\boldsymbol{t}_j + \boldsymbol{f}_j + \boldsymbol{\mu}_y, \quad j = 1, 2, \cdots, n_1 \end{aligned} \tag{4.6}$$

where, $\boldsymbol{x}_{i_\lambda} \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{y}_i \in \mathbb{R}^{r\times 1}$ denote the input and output data, respectively, i.e.

$$\boldsymbol{x}_{i_\lambda} = \begin{bmatrix} x_{1_{i-\lambda_1}} \\ x_{2_{i-\lambda_2}} \\ \vdots \\ x_{m_{i-\lambda_m}} \end{bmatrix} \tag{4.7}$$

where $\lambda_1, \lambda_2, \cdots, \lambda_m$ are the time-delays for the m input variables, respectively. $\boldsymbol{P} \in \mathbb{R}^{m\times q}$ and $\boldsymbol{C} \in \mathbb{R}^{r\times q}$ are the weighting matrices, and $\boldsymbol{t}_i \in \mathbb{R}^{q\times 1}$ is a vector of latent variables. The variables $\boldsymbol{e}_i \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{f}_i \in \mathbb{R}^{r\times 1}$ denote the measurement noise in input and output, respectively, which are assumed to follow a mixture of two Gaussian distributions given in Eqs. (4.9) and (4.10) to account for outliers and regular noises. The mean values of the input and output variables are denoted with $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$, respectively.

$$\boldsymbol{t}_i \sim \mathcal{N}(0, \boldsymbol{I}) \tag{4.8}$$

$$\boldsymbol{e}_i \sim (1-\delta_x)\mathcal{N}(0, \sigma_x^2\boldsymbol{I}) + \delta_x\mathcal{N}(0, \boldsymbol{\rho}_x^{-1}\sigma_x^2\boldsymbol{I}) \tag{4.9}$$

$$\boldsymbol{f}_i \sim (1-\delta_y)\mathcal{N}(0, \sigma_y^2\boldsymbol{I}) + \delta_y\mathcal{N}(0, \boldsymbol{\rho}_y^{-1}\sigma_y^2\boldsymbol{I}) \tag{4.10}$$

Due to the nonlinear and/or multi-modal nature of the plants, developing a single PPCR model to capture the entire plant is not suitable. Thus, to improve the accuracy in modeling, exponential weights are calculated based on Euclidean distance assigned to pick the most relevant data points for building the model. The weights are calculated based on Eq. (4.11).

$$w_i = \exp(\frac{-d_i^2}{\phi}) \tag{4.11}$$

where ϕ is a tuning parameter that defines how the weights are spread across the neighborhood of the testing data to develop the weighted PPCR model, and d_i is the Euclidean distance. The detailed introduction of the weighted PPCR model can be found in [89]. The model is developed under the framework of expectation maximization (EM) algorithm. The parameters are estimated from the $Q - function$ presented in Eq. (4.12).

$$Q = \\ E_{\boldsymbol{X_\lambda},\boldsymbol{T},Q_x,Q_y,\Gamma|\boldsymbol{X}_{o_\lambda},\boldsymbol{Y},\boldsymbol{\theta}^{old}}\left[\sum_{i\in\boldsymbol{O}} w_i \log p(\boldsymbol{x}_{i_\lambda},\boldsymbol{y}_i,\boldsymbol{t}_i,Q_x,Q_y,\Gamma \mid \boldsymbol{\theta}) + \sum_{i\in\boldsymbol{M}} w_i \log p(\boldsymbol{x}_{i_\lambda},\boldsymbol{t}_i,Q_x,Q_y,\Gamma \mid \boldsymbol{\theta})\right] \tag{4.12}$$

The $Q - function$ presented in Eq. (4.12) is solved by incorporating the equations derived in Chapter 2 and Chapter 3. In the rest of this chapter, the generalized weighted PPCR model is denoted as G^{PPCR}.

4.3.2 Data-driven self-optimization algorithm formulation

Since the plant conditions change over time, the historical data that is used to build the model may not be able to accurately describe the current condition of the plant. Therefore, a model-plant mismatch exists between the weighted PPCR model built from the historical data and the current condition of the plant. To account for this model-plant mismatch, the authors in [73] proposed to use the Gaussian process regression (GPR). The objective of this GPR model is to build a model by considering the differences between the values of the objective function that are calculated from the plant (real-time measurements) and the estimation from the weighted PPCR model. A similar approach is also followed for the constraints, and the resultant set of equations are shown in Eq. (4.13).

$$\delta G_i = G_i^P - G_i^{PPCR} \sim \mathcal{GP}(\boldsymbol{\mu}_{\delta G_i}, \boldsymbol{\sigma}^2_{\delta G_i}), \quad i = 0, \cdots, n_g \tag{4.13}$$

where n_g is the total number of constraints.

Hence, the optimization problem in Eq. (**??**) can be modified and is given in Eq. (4.14)

$$\begin{aligned} \boldsymbol{u}^{k+1^*} \in \arg\min_{u\in\mathcal{U}} \ & [G_0^{PPCR} + \boldsymbol{\mu}^k_{\delta G_0}](\boldsymbol{u}) \\ \text{s.t. } & [G_i^{PPCR} + \boldsymbol{\mu}^k_{\delta G_i}](\boldsymbol{u}) \leq 0, \quad i = 1, \cdots, n_g \end{aligned} \tag{4.14}$$

where $\boldsymbol{\mu}^k_{\delta G_i}$ is the estimated mean of the GP regression that accounts for the term of model-plant mismatch in iteration k. The mean values used in Eq. (4.14) are those values that are estimated from Eq. (4.13) which approximates the model-plant mismatch and correct the objective function and constraints in the optimization problem.

As discussed in section 4.3.1, the amount of data points relevant to the current operating condition is determined by tuning the parameter ϕ i.e., by decreasing ϕ, fewer data points will contribute to the model construction. If the current operating point is in a highly nonlinear region, building the weighted PPCR model might not have a sufficient accuracy. Thus, by decreasing the parameter ϕ, less number of data points will receive significant weights for the corresponding data points to be effectively utilized; hence, a generalized weighted PPCR model in a smaller region will be built. On the other hand, when the weighted PPCR model approximates the nonlinear plant very well, we can increase area and have more data with sufficient weight while developing a model. Hence, a non-linearity index is proposed to define the range of data to be effectively used, and based on the index, the parameter ϕ can be tuned. The non-linearity index calculates the performance ratio between the nonlinear model (GP regression model) built from the historical data and the linear weighted PPCR model, as can be seen in Eq. (4.15).

$$\rho^{k+1} := \frac{G_0^{GP}(\boldsymbol{u}^k) - G_0^{GP}(\boldsymbol{u}^{k+1^*})}{[G_0^{PPCR} + \boldsymbol{\mu}_{\delta G_0}^k](\boldsymbol{u}^k) - [G_0^{PPCR} + \boldsymbol{\mu}_{\delta G_0}^k](\boldsymbol{u}^{k+1^*})} \tag{4.15}$$

After calculating the non-linearity index from Eq. (4.15), similar to the concept of trust-region optimization [77], three different thresholds are determined to tune ϕ. These three thresholds are $0 < \eta_1 \leq \eta_2 < \eta_3 \leq 1$. The shrinking and expansion actions to change ϕ are $0 < t_1 < 1 < t_2$ where t_1 and t_2 are shrinking and expansion values, respectively. It has to be noted that these parameters should be tuned before starting the algorithm [77].

The size of the weighted PPCR model is updated based on the following steps:

1. If $G_i^P(u^{k+1^*}) > 0$ for some $i = 1, \cdots, n_g$ or $\rho^{k+1} < \eta_2$ then $\phi := t_1 \times \phi$
2. Else if $\rho^{k+1} > \eta_3$ then $\phi := \min\{t_2 \times \phi, \phi^{max}\}$
3. Else $\phi := \phi$

where ϕ^{max} is the maximum allowable value that ϕ can take. Based on the value of ρ, the decision will be made on whether to repeat the optimization, or the obtained optimal point can be used as the operating point for the next iteration. The decision criterion is as following:

1. If $G_i^P(u^{k+1^*}) > 0$ for some $i = 1, \cdots, n_g$ or $\rho^{k+1} < \eta_1$ then $u^{k+1} := u^k$

2. Else $u^{k+1} := u^{k+1^*}$

Based on the aforementioned steps, the number of effective points, which need to be accounted for the optimization, will be changed and adjusted based on the performance of the previous iteration. The steps of the proposed algorithm is provided in Algorithm 1.

Algorithm 1: Data-driven self-optimization algorithm

```
**Input:** historical data (input and output); initial (query) point, $x_q$;
maximum value for $\phi^{max}$ and an initial value for $\phi$; non-linearity threshold
parameters $0 < \eta_1 \leq \eta_2 < \eta_3 \leq 1$; expansion and shrinking parameters $t_1$
and $t_2$; objective and $n_g$ constraint functions of the optimization problem
**Repeat:** **for** $k = 0, 1, \cdots$
1: Build the generalized weighted PPCR model for the given $x_q$ and the
   historical data
2: Train GP regression modifiers based on the weighted PPCR estimates and
   the real-time measurements of the plant
3: Solve the modified optimization problem provided in Eq. (4.14) and obtain
   $u^{k+1}$
4: Calculate the non-linearity index $\rho^{k+1}$
5: Update the value of $\phi$ based on the value of $\rho^{k+1}$
6: Based on the developed criterion decide to accept the new operating point
   or to repeat the optimization problem in step 3.
7: $x_q \leftarrow u^{k+1}$ or $x_q \leftarrow u^k$ based on the previous step's result
```

One of the drawbacks of algorithm 1 is that the solution obtained from the optimization problem can get into the local optimum. To circumvent this problem and letting the optimization explore more locations, the acquisition function from the theory of reinforcement learning and Bayesian optimization is used. The authors in [82] proposed using the acquisition functions in objective function. However, in our proposed method, acquisition functions are used both in objective and constraint functions. Therefore, the LCB acquisition function is used [97] and the modified optimization problem is given in Eq. (4.16):

$$\begin{aligned} \boldsymbol{u}^{k+1^*} \in \arg\min_{u \in \mathcal{U}} \; & [G_0^{PPCR} + \boldsymbol{\mu}_{\delta G_0}^k - \beta \boldsymbol{\sigma}_{\delta G_0}^2](\boldsymbol{u}) \\ \text{s.t. } & [G_i^{PPCR} + \boldsymbol{\mu}_{\delta G_i}^k - \beta \boldsymbol{\sigma}_{\delta G_i}^2](\boldsymbol{u}) \leq 0, \quad i = 1, \cdots, n_g \end{aligned} \tag{4.16}$$

In Eq. (4.16), the variances estimated from the GPR in Eq. (4.13) are used to take the optimization search to a newer area and may therefore escape the local optimum points. The negative sign before β is consistent with the optimization problem as the goal is to minimize the objective function. Introducing the LCB acquisition function in the constraints helps to relax these functions while solving the optimization problem. However, if it is needed to tighten the constraints, the UCB acquisition function can be used. With the introduction of acquisition functions, the optimization problem provided in Eq. (4.16) is solved in the step 3 of Algorithm 1, and the rest of the steps remain the same.

4.4 Case Studies

In this section, the performance of the proposed algorithm is demonstrated by a simulation of a deethanizer column through the Aspen HYSYS V.9 [1]. An industrial example on the zinc roasting unit is also used to demonstrate the practical applicability of the proposed method.

4.4.1 Simulation Example: Deethanizer column

The deethanizer column is a continuous operating distillation column used for extracting ethane as a distillate from a mixed feed that contains light hydrocarbons. Deethanizer column is one of the most important units in refineries and is usually located ahead of other units in the plant.

In Fig. 4.3, a principle of the deethanizer column is demonstrated. The objective of the deethanizer column in the refinery plants is to separate C_3+ components from the upstream feed.

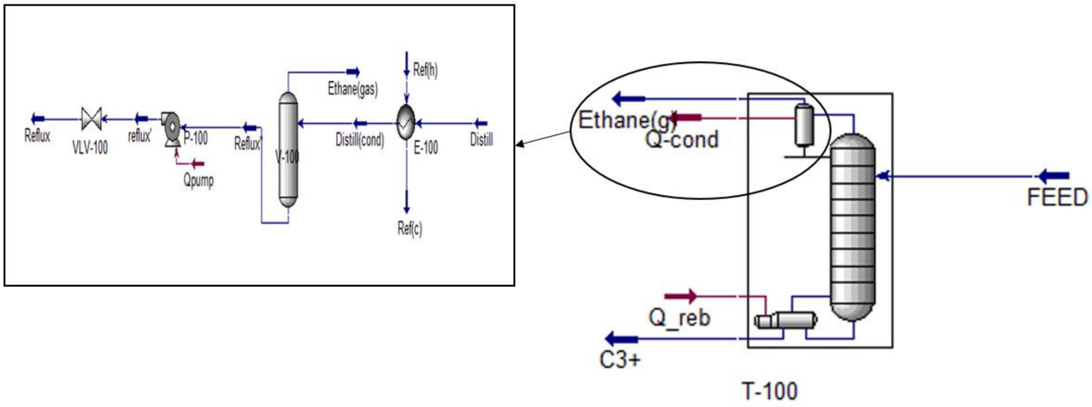


Figure 4.3: The schematic of the deethanizer plant [1]

The main objective of optimization is to minimize the operational cost of the unit, which depends on the energy consumption in the reboilers, the condensers and the pumps. To minimize the energy consumption, the temperature and the feed rate of the input stream needs to be regulated. Hence, the objective function is defined as:

$$\begin{aligned} &\min_{T_{feed},F_{feed}} \quad Q_{reb} + Q_{cond} + Q_{pump} \\ &\quad S.T.: \\ &\quad f(Q_{reb}, Q_{cond}, Q_{pump}, F_{bottom}, X_{Ethane,bottom}, T_{feed}, F_{feed}) = 0 \\ &\quad T_{feed} \in [15, 30] \\ &\quad F_{feed} \in [8000, 11000] \\ &\quad F_{bottom} < 2 \times 10^5 [kg/h] \\ &\quad X_{ethane,bottom} < 0.05 \end{aligned} \tag{4.17}$$

where F_{feed} and F_{bottom} are the flow rate of the feed and bottom product, respectively. T_{feed} is the feed's temperature, and $X_{ethane,bottom}$ is the molar fraction of the ethane in the bottom product. Q_{reb}, Q_{cond}, and Q_{pump} are the terms corresponding to the energy consumption of the reboiler, condenser, and pump, respectively. $f(.) = 0$ is the PPCR model that relates input and output variables to each other. The first two constraints i.e., T_{feed}, F_{feed} which are defined in Eq. (4.17) are the operational constraints, and the F_{bottom} and $X_{ethane,bottom}$ are the planning constraints. In such a setting, 15% of the input and 35% of the output variables are assumed to be missed, and 10% of the data is replaced with outliers.

By solving the optimization of Eq. (4.17) with the *optimization module* of Aspen HYSYS, the minimum energy consumption is $1.082 \times 10^8 [W]$ and the decision variables are found to be $T_{feed} = 16.3$ and $F_{feed} = 10485$. The operating region and the actual solution to the optimization of Eq. (4.17) are presented in Fig. 4.4

To demonstrate the efficacy of the proposed method in steering the plant to its optimal point, two different initializations (current operating points, COPs) are considered. In Fig. 4.5(a), the locations of these COPs are shown in Fig. 4.5(b), where the path and the final solution obtained by the proposed method for each COP are provided.

Based on the results demonstrated in Fig. 4.5, the proposed algorithm is able to find the optimal path and solution, and steer the plant to the desired point. It is well known that the amount of signal to noise ratio (SNR) can affect the performance of

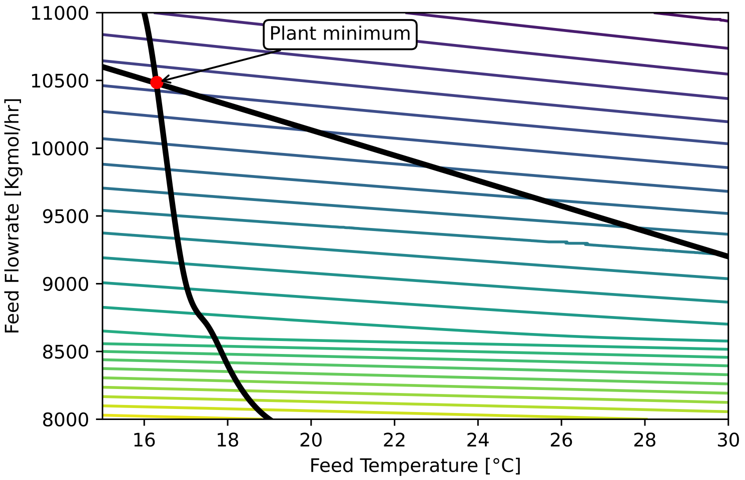


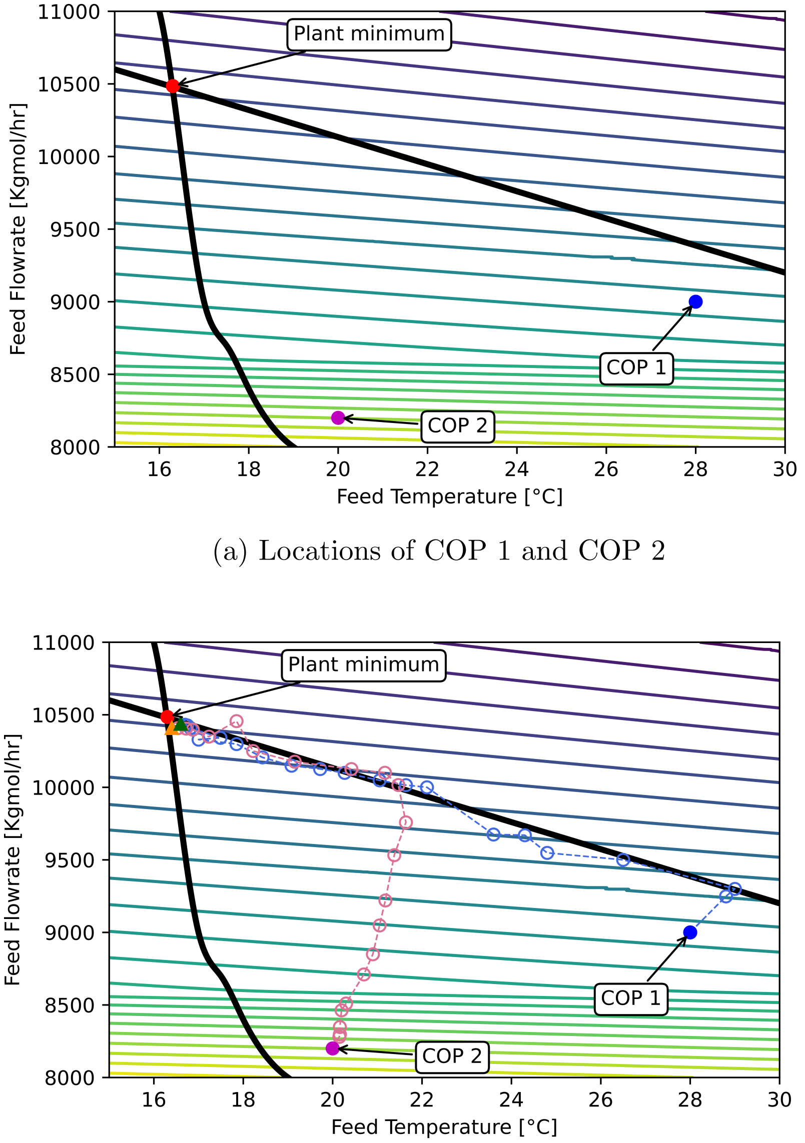
Figure 4.4: Operating region and optimal point of the deethanizer problem

the algorithm. To study the effect of measurement noise on the proposed algorithm, eight different noise levels are considered for this study whose optimal points are shown with △. Two different initial operating points are considered for this study which are similar in all the noise levels. As it can be seen from Figure 4.6(a), solutions corresponding to noisy data (low SNR) are getting trapped in local optimum points instead of getting close to the true optimum. To obtain a possible better solution through the discovery of the new path by searching through a wider optimization region, the exploration described in Eq. (4.16) is applied, and the results are demonstrated in Figure 4.6(b).

From the results of Fig. 4.6, it can be concluded that with the inclusion of exploration in the optimization as explained in Eq. (4.16), helps in better convergence to the actual optimal point, and avoid being trapped in the local optimum points.

4.4.2 Industrial case study: Zinc roasting unit

One of the minerals that have vast application in different industries is zinc. One of the applications of zinc is galvanizing other materials like iron and aluminum to prevent rusting. The galvanized steel is used as the main material for car bodies, street lamps, safety barriers beside the roads, and suspension bridges. Moreover, zinc can be used to produce die-castings which are essential for electrical and automobile industries. Zinc is also one of the components used in alloy productions. In addition, zinc oxide can be used in rubber, pharmaceuticals, paints, textiles, and soap manufacturers [98]. One of the processes to produce zinc or zinc oxide is the roasting unit that is shown



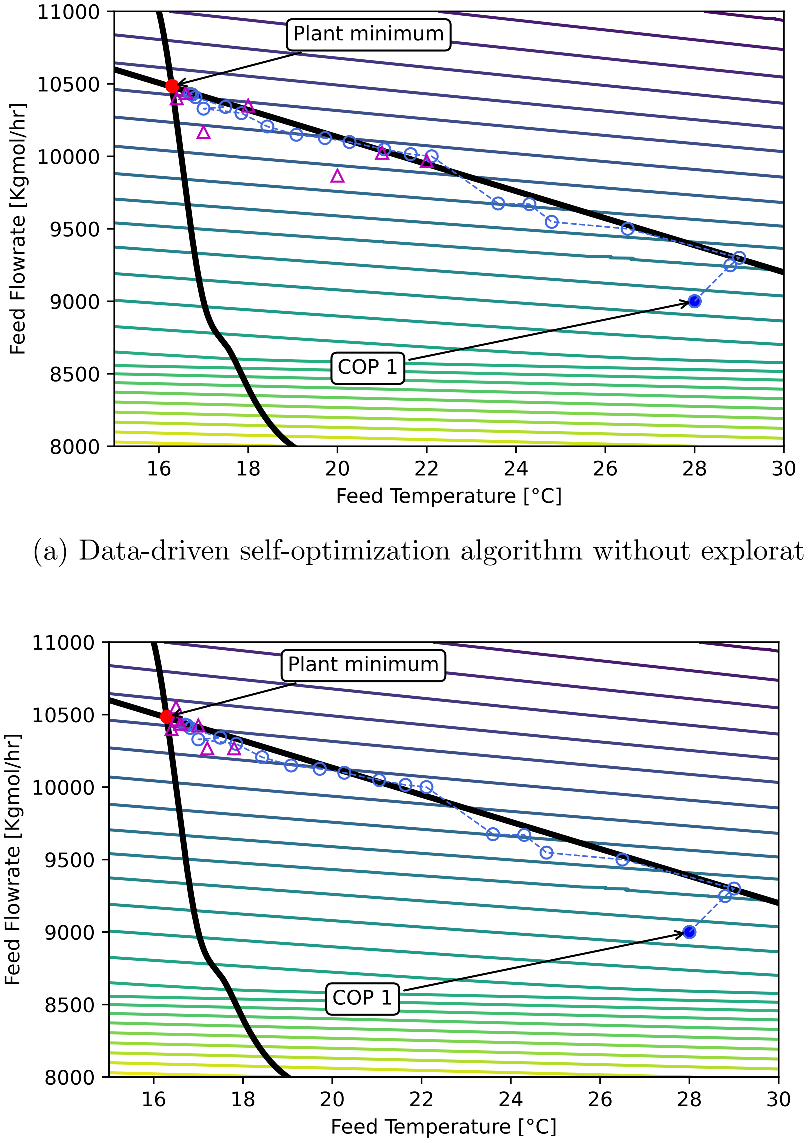
(a) Locations of COP 1 and COP 2

(b) Optimal path and solution from the proposed algorithm

Figure 4.5: Initial points and the solutions obtained by the proposed data-driven self-optimization algorithm

in Fig. 4.7. In this unit, zinc sulfide feed turn into zinc oxide at high temperatures which contains impurity. The reactions that take place in the fluidized-bed roaster are provided in Eq. (4.18):

$$\begin{aligned} 2\,\mathrm{ZnS} + 3\,\mathrm{O_2} &\xrightarrow{\triangle} 2\,\mathrm{ZnO} + 2\,\mathrm{SO_2} \\ 2\,\mathrm{SO_2} + \mathrm{O_2} &\xrightarrow{\triangle} 2\,\mathrm{SO_3} \end{aligned} \tag{4.18}$$



(a) Data-driven self-optimization algorithm without exploration

(b) Data-driven self-optimization algorithm with with exploration

Figure 4.6: Study the effect of exploration in the optimization problem. Solutions to the optimization problem with 8 different noise levels (shown with △) without exploration (Figure 4.6(a)) and with exploration (Figure 4.6(b))

The fluidized-bed roaster is operating below the atmospheric pressure and at temperatures around 1000°C [99]. Providing greater capacity, better sulfur removal capability, and lower cost for maintenance is the advantage of the fluidized-bed roaster [100]. After the fluidized-bed roaster, products are sent to the leaching plant to leach zinc oxide out of zinc.

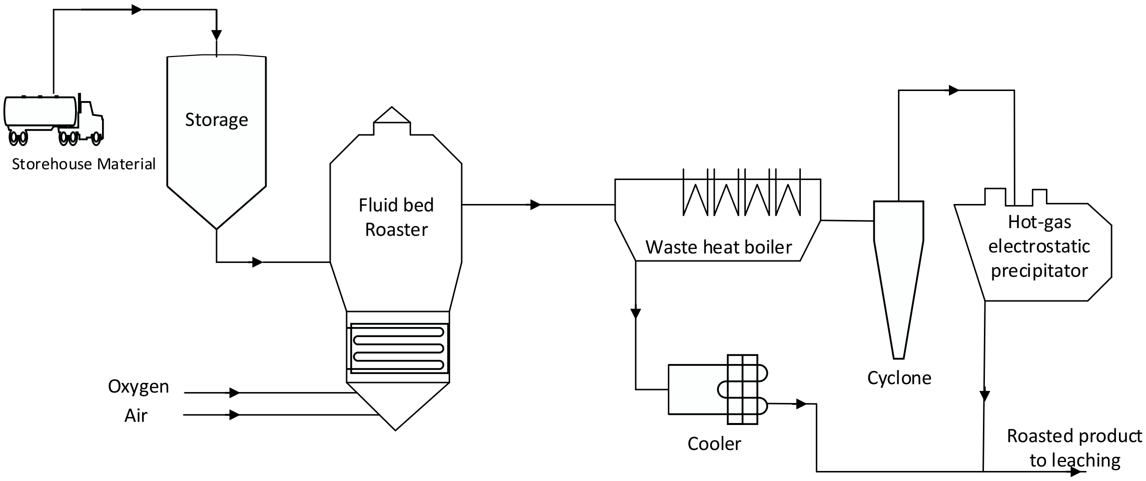


Figure 4.7: The schematic of the zinc roasting unit [2]

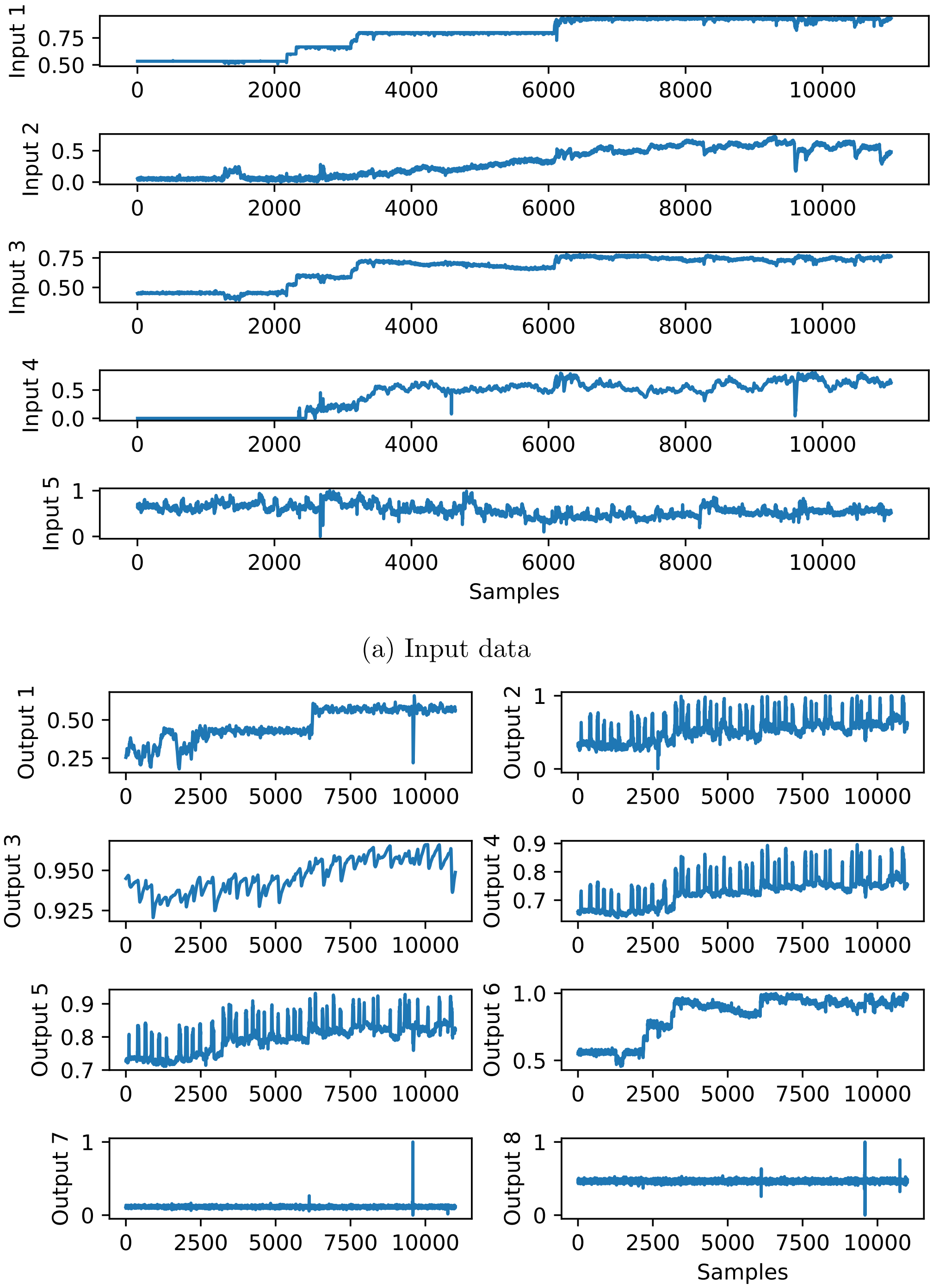
The objective of the optimization is to maximize the feed rate coming from the warehouse, along with minimizing the required amount of oxygen for roasting operation in the fluidized bed while keeping the other variables within the limits. Such a process consists of 5 inputs (MVs) and 8 outputs (CVs) as described in Table 4.1. A total of 1100 data points are collected from the plant and the normalized values of all the variables are shown in Fig. 4.8, and the optimization problem is defined as:

$$\begin{aligned}
&\min_{allMV's} \quad - FeedRate + 1.2 \times \lambda \\
&S.T. : \\
&L_i \leq var_i \leq H_i \ ; \forall i \\
&y_j = f(\boldsymbol{u}) \ \ \forall j \quad , \ u \in MV
\end{aligned} \tag{4.19}$$

where L_i is the lower and H_i is the upper bound, and λ is the required amount of oxygen for roasting operation in the fluidized-bed roaster. var_i represents input and output variables.

The self-optimization algorithm is tested on two different initializations, and the obtained results are shown in Fig 4.9. From the results shown in Fig. 4.9, it can be concluded that the proposed method is attempting to maximize the feed rate and minimizing the oxygen demand while simultaneously satisfying all other constraints.

Though the results provided in Fig. 4.9 are satisfactory, there is room for further improvement as the feed rate is not at its upper bound. Thus the acquisition function is applied to investigate the possibilities of further improvement in the result of optimization. The results obtained are shown in Fig. 4.10. As it can be observed

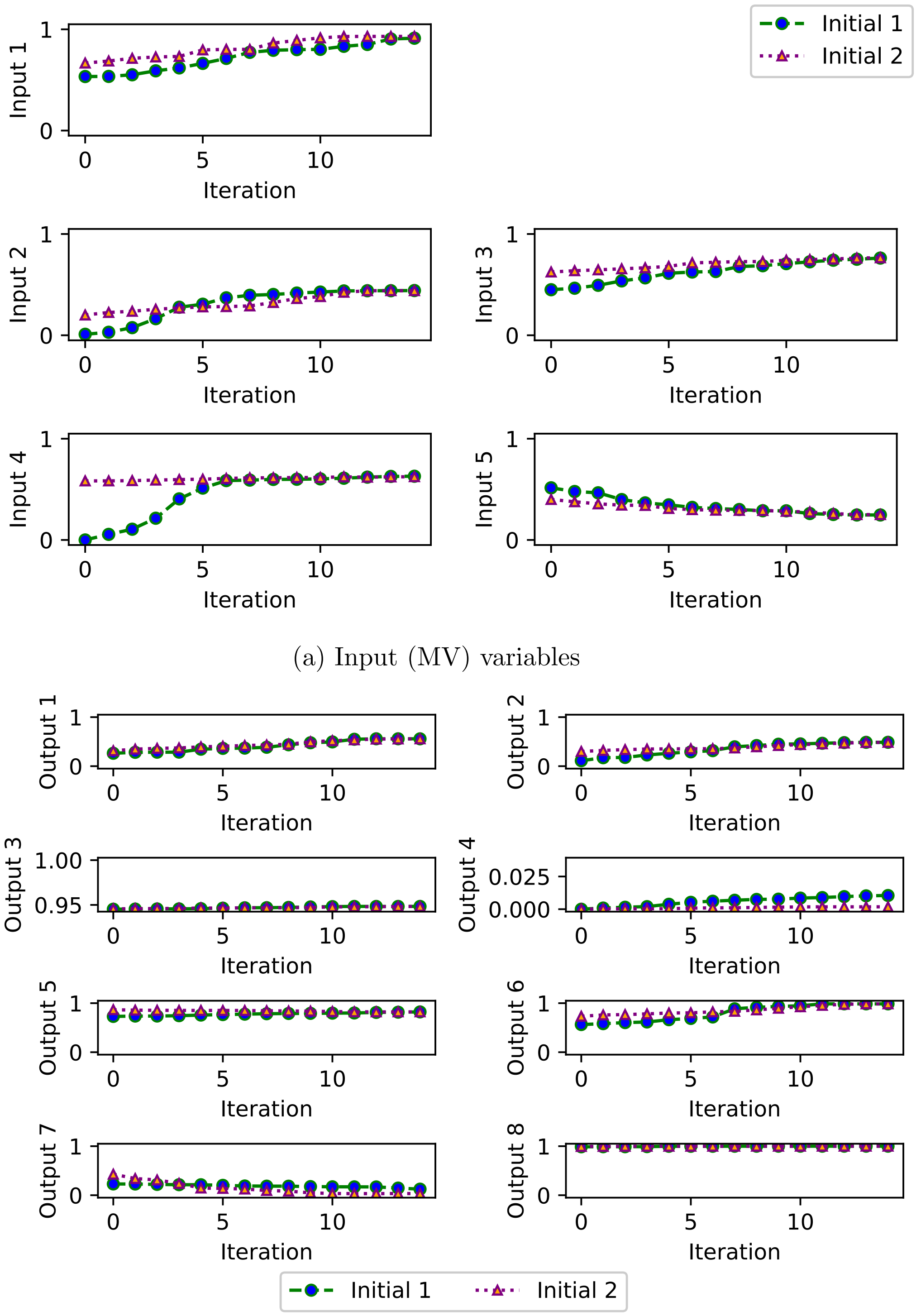


(a) Input data

(b) Output data

Figure 4.8: Historical data of the zinc roasting unit.

from Fig 4.10, the acquisition function helps to improve the optimization and steer the plant to a higher feed rate, and the algorithm reduces the possibility of getting trapped in the local optimum.



(a) Input (MV) variables

(b) Output (CV) variables

Figure 4.9: Optimizing the zinc roasting unit and finding the optimal path from two different initializations

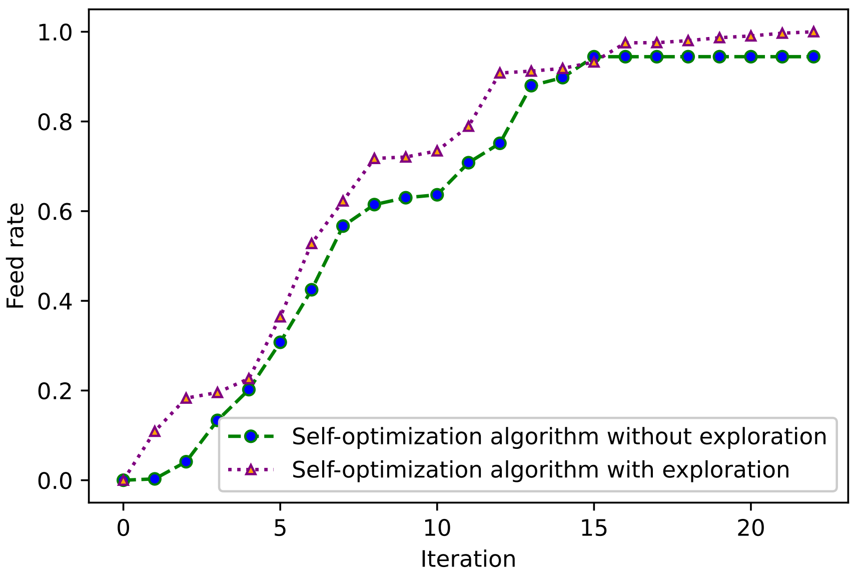


Figure 4.10: Zinc roasting unit optimization with the help of acquisition function

4.5 Conclusion

In this work, a data-driven self-optimization of the process in the presence of model-plant mismatch is proposed to find the plant optimum along with the optimal path to reach the obtained point. The objective of the proposed algorithm is to automate the procedure of finding optimal operating points of a process. It models the plant with a generalized weighted PPCR model and the Gaussian process regression model is utilized to compensate the model-plant mismatch. A non-linearity index is proposed to adjust the weighted PPCR model to ensure its accuracy at a sufficient level. Finally, to make a balance between exploitation and exploration, acquisition function is used in the optimization. The performance of the proposed algorithm is demonstrated on the simulated deethanizer column and an industrial zinc roasting unit. Based on the results obtained from the case studies, it can be concluded that the proposed algorithm is able to move the plant towards the plant's optimal point.

Table 4.1: Process variables description

Variable	**Process Variable Description**	**Variable**	**Process Variable Description**
Input 1	Feed Rate	Output 1	Fluidized-bed Temperature
Input 2	Air Flow Rate	Output 2	Precipitator Pressure
Input 3	Oxygen Flow Rate	Output 3	Cyclone Temperature
Input 4	Fluidized-bed Flow Rate	Output 4	Cooler Output
Input 5	Inlet Pressure	Output 5	Coolant Flow Rate
		Output 6	Oxygen Percentage
		Output 7	Required Amount of Oxygen
		Output 8	Air:Feed Ratio

Chapter 5

Conclusions

In this chapter, summaries of the thesis are provided in section 5.1, and some possible future research is discussed in section 5.2.

5.1 Summary

The main objective of this thesis is to introduce an online framework for plant optimization and finding the optimal path for steering the plant to the optimal operating point. Most of the available works rely on the first-principle models to describe the plant and solve the optimization problem, which needs an in-depth understanding of the process. Hence, a data-driven self-optimization algorithm in the presence of the model-plant mismatch, outliers, delays, and missing data in both input and output variables is proposed in this thesis. This algorithm is developed by utilizing a generalized weighted probabilistic principal component regression (PPCR) model. It attempts to model the plant by using the plant datasets that contain different types of uncertainties like outliers, missing data in input and output variables, and delays between the variables.

Chapter 1 provides the motivation and challenges in process optimization and modeling of the industrial processes. An overview of contributions of the thesis presented.

In Chapter 2, modeling of the plant through the incorporation of a mixture robust semi-supervised probabilistic principal component regression (MRSSPPCR) model is proposed. This approach can address the high dimensionality of the process alongside the multi-modal nature of the processes. The main advantage of the proposed approach lies in its ability to deal with outliers in each input and output variable with different properties. Further, due to the sensor failure or delays in measuring process variables, some measurements may be missing at times. The problem of missing data in the input variables is addressed by the data imputation methods when the data is missing completely at random (MCAR). For output variables, the issue of missing data is addressed with the help of the semi-supervised learning. The proposed model is developed through the expectation maximization (EM) algorithm to estimate model parameters in the presence of hidden variables like missing data in input variables, hidden operating modes, and outliers statistics. The significance of this chapter is to develop a reliable model that can deal with different uncertainties in the data. To demonstrate the prediction performance of the proposed model, a numerical example and an experimental example on the hybrid tank pilot plant system are provided where an improvement compared to the previously available models is observed in both cases.

In Chapter 3, a weighted semi-supervised probabilistic principal component regression with missing input and delayed output data is proposed to address the nonlinear nature of the processes, taking care of data high dimensionality. The proposed model is able to deal with the time-delay between each input variable and output variable. In addition, the model is robust to the missing data in both input and output variables while an assumption of MCAR is considered. By utilizing the just-in-time learning and locally weighted modeling, the proposed model is able to provide an online local model based on the query points. Euclidean distance-based weights are assigned to the process datasets such that only the most relevant data information is utilized while developing a model. Similar to the previous chapter, the expectation maximization (EM) algorithm is utilized for the development of a model. The EM algorithm enables the model to identify time-delays, impute missing data in input variables, and estimate the hidden variables of the PPCR model. In order to improve the convergence of the PPCR model, an updating strategy for the delay ranges is proposed, which can reduce the range of the considered delay for some/all of the variables at each iteration. In contrary to the previous works that consider a fixed-distribution for time-delay variables while developing a model, in the proposed model, a free-distribution model is considered that gives more flexibility for each time-delay to act independently. Finally, the model accuracy is demonstrated through a numerical example and experimental study on the hybrid tank pilot plant system, and the results are compared with the other methods, demonstrating the superiority of the proposed method.

In Chapter 4, a data-driven self-optimization in the presence of the plant-model mismatch, is proposed which is an online data-driven approach for process optimization. A combination of the modeling methods in Chapter 2 and Chapter 3 namely the generalized weighted probabilistic principal component regression (PPCR) model, is used to model the process. To account for the plant-model mismatch, a robust Gaussian process regression model is used. A nonlinearity index is proposed to determine the extent of nonlinearity in the process and adjust the generalized weighted PPCR model accordingly. To increase the possibility of finding the optimal solution, the acquisition function from reinforcement learning is used to make the exploration in optimization process and obtain a trade off between exploitation and exploration. The applicability of the proposed algorithm is demonstrated through an example on the deethanizer column simulated through the Aspen HYSYS software. Finally, an industrial zinc roasting unit is considered to demonstrate the practical applicability of the proposed method.

5.2 Future Work

In Chapter 2, a MRSSPPCR model is proposed that deals with outliers with different properties along with missing data in input and output variables. In the proposed model, scaled outliers are studied, and a mixture of two Gaussian models is used to differentiate the regular noise from outliers. Scaled outliers have a different variance from the rest of the data, whereas location outliers represent a common problem such as a jammed instrument which have different mean values. Considering the location outliers with the scaled outliers can be a future work. Moreover, instead of the mixture of two Gaussian distributions, a Laplace distribution or student's t distribution can be used. For missing data in input and output variables, the assumption of missing completely at random (MCAR) is considered. The other types of missing data like missing at random (MAR) or missing not at random (MNAR) can be explored. In addition, the variational Bayesian (VB) algorithm can be used instead of the expectation maximization (EM) algorithm where the VB provides a measure for the amount of uncertainty in the parameters.

In Chapter 3, a weighted semi-supervised probabilistic principal component regression (PPCR) is proposed. In the proposed model, the presence of the constant time-delay is studied in the process datasets. In future work, the time-varying time-delay can be investigated in the framework of the PPCR model. Moreover, the Kullback–Leibler (KL) divergence can be replaced instead of Euclidean distance-based weight assignment as Euclidean distance measures the distance point by point, it is not robust to uncertainties. On the other hand, KL divergence is known to be a better metric for measuring the distance that is used to find the similarity between two different distributions and can possibly be robust to uncertainties.

In Chapter 4, an online data-driven optimization framework is proposed that considers a steady-state model as its main model. The drawback of the static RTO implementation is the steady-state wait time that delays the model adaptation [101]. In future work, this linear static model can be replaced with a dynamic model to consider transitions between operating modes in addition to steady-state. For the exploration of the proposed optimization algorithm, the lower confidence bound (LCB) acquisition function is used in both optimization objective function and constraints. However, the study of the other types of the acquisition functions like the probability of improvement, entropy search, and expected improvement is worthy to study, and comparing these functions together can be an interesting direction. Further, extension to the Gaussian process regression model to deal with the missing data and delay can be considered in future works.

Bibliography

[1] Amel BELHOCINE, Riad BENDIB, and Youcef ZENNIR. Simulation and analysis of a petrochemical process (deethanizer column-mle field) using hysys aspen simulator. *Algerian Journal of Signals and Systems*, 5(2):86–91, 2020.

[2] BS Boyanov, MP Sandalski, and KI Ivanov. Zinc sulfide concentrates and optimization of their roasting in fluidezed bed reactor. *World Academy of Science, Engineering and Technology*, 73:326–332, 2011.

[3] George Tsakalidis and Kostas Vergidis. Towards a comprehensive business process optimization framework. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 1, pages 129–134. IEEE, 2017.

[4] Ying Chen, Zhihong Yuan, and Bingzhen Chen. Process optimization with consideration of uncertainties—an overview. *Chinese journal of chemical engineering*, 26(8):1700–1706, 2018.

[5] Amir M Aboutaleb, Mohammad J Mahtabi, Mark A Tschopp, and Linkan Bian. Multi-objective accelerated process optimization of mechanical properties in laser-based additive manufacturing: Case study on selective laser melting (slm) ti-6al-4v. *Journal of Manufacturing Processes*, 38:432–444, 2019.

[6] Tomoyuki Taguchia and Yoshiyuki Yamashitab. A hybrid approach for process optimization of distillation reflux condition using first principle models and least squares regression. In *Computer Aided Chemical Engineering*, volume 44, pages 229–234. Elsevier, 2018.

[7] Stefano Di Cairano and Ilya V Kolmanovsky. Real-time optimization and model predictive control for aerospace and automotive applications. In *2018 annual American control conference (ACC)*, pages 2392–2409. IEEE, 2018.

[8] Shabnam Sedghi, Anahita Sadeghian, and Biao Huang. Mixture semisupervised probabilistic principal component regression model with missing inputs. *Computers & Chemical Engineering*, 103:176–187, 2017.

[9] Zhiqiang Ge, Biao Huang, and Zhihuan Song. Mixture semisupervised principal component regression model and soft sensor application. *AIChE Journal*, 60(2):533–545, 2014.

[10] A Sadeghian, O Wu, and B Huang. Robust probabilistic principal component analysis based process modeling: Dealing with simultaneous contamination of

both input and output data. *Journal of Process Control*, 67:94–111, 2018.

[11] Joachim Bocker, Bernd Schulz, Tobias Knoke, and Norbert Frohleke. Self-optimization as a framework for advanced control systems. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 4671–4675. IEEE, 2006.

[12] Johannes Wiebe, Inês Cecílio, and Ruth Misener. Data-driven optimization of processes with degrading equipment. *Industrial & Engineering Chemistry Research*, 57(50):17177–17191, 2018.

[13] Shi Chen, Lei Dai, Jianjun Liu, Yanfeng Gao, Xinling Liu, Zhang Chen, Jiadong Zhou, Chuanxiang Cao, Penggang Han, Hongjie Luo, et al. The visible transmittance and solar modulation ability of vo 2 flexible foils simultaneously improved by ti doping: an optimization and first principle study. *Physical Chemistry Chemical Physics*, 15(40):17537–17543, 2013.

[14] Ajaya Kumar Pani and Hare Krishna Mohanta. A survey of data treatment techniques for soft sensor design. *Chemical Product and Process Modeling*, 6(1), 2011.

[15] Wen Yu and America Morales. Data driven fast real-time optimization with application to crude oil blending. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6. IEEE, 2019.

[16] Thomas Savage, Panagiotis Petsagkourakis, Dongda Zhang, Nilay Shah, Ehecatl Antonio del Rio-Chanona, et al. Data-driven optimization for process systems engineering applications. *Chemical Engineering Science*, page 117135, 2021.

[17] Martin Guay. A time-varying extremum-seeking control approach for discrete-time systems. *Journal of Process Control*, 24(3):98–112, 2014.

[18] Ankush Chakrabarty, Claus Danielson, Scott A Bortoff, and Christopher R Laughman. Accelerating self-optimization control of refrigerant cycles with bayesian optimization and adaptive moment estimation. *Applied Thermal Engineering*, 197:117335, 2021.

[19] M Poggio, MA Renouf, and BL Schroeder. Balancing profitability and environmental considerations in best practice cane growing. In *Proceedings of the International Society of Sugar Cane Technologists*, volume 29, pages 1840–1849, 2016.

[20] Hasan Y Alhammadi and Jose A Romagnoli. Incorporating environmental, profitability, heat integration and controllability considerations. *The integration of process design and control*, 17:264, 2004.

[21] Young-Don Ko and Helen Shang. A neural network-based soft sensor for particle size distribution using image analysis. *Powder Technology*, 212(2):359–366, 2011.

[22] Xiao Wang and Han Liu. Soft sensor based on stacked auto-encoder deep neural network for air preheater rotor deformation prediction. *Advanced Engineering Informatics*, 36:112–119, 2018.

[23] Wang Jianlin, YU Tao, and JIN Cuiyun. On-line estimation of biomass in fermentation process using support vector machine. *Chinese Journal of Chemical Engineering*, 14(3):383–388, 2006.

[24] Pengbo Zhu, Xianqiang Yang, and Hang Zhang. Mixture robust l1 probabilistic principal component regression and soft sensor application. *The Canadian Journal of Chemical Engineering*, 2020.

[25] Mark Joswiak, You Peng, Ivan Castillo, and Leo H Chiang. Dimensionality reduction for visualizing industrial chemical process data. *Control Engineering Practice*, 93:104189, 2019.

[26] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[27] Zhiqiang Ge. Process data analytics via probabilistic latent variable models: A tutorial review. *Industrial & Engineering Chemistry Research*, 57(38):12646–12661, 2018.

[28] Zhiqiang Ge, Furong Gao, and Zhihuan Song. Mixture probabilistic pcr model for soft sensing of multimode processes. *Chemometrics and intelligent laboratory systems*, 105(1):91–105, 2011.

[29] Reza Sharifi and Reza Langari. Nonlinear sensor fault diagnosis using mixture of probabilistic PCA models. *Mechanical Systems and Signal Processing*, 85:638–650, 2017.

[30] Shima Khatibisepehr and Biao Huang. Dealing with irregular data in soft sensors: Bayesian method and comparative study. *Industrial & Engineering Chemistry Research*, 47(22):8713–8723, 2008.

[31] Matteo Magnani. Techniques for dealing with missing data in knowledge discovery tasks, Department of Computer Science. *University of Bologna, Italy*, pages 1–10, 2004.

[32] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[33] Anahita Sadeghian and Biao Huang. Robust probabilistic principal component analysis for process modeling subject to scaled mixture gaussian noise. *Computers & Chemical Engineering*, 90:62–78, 2016.

[34] Hariprasad Kodamana, Biao Huang, Rishik Ranjan, Yujia Zhao, Ruomu Tan, and Nima Sammaknejad. Approaches to robust process identification: A review and tutorial of probabilistic methods. *Journal of Process Control*, 66:68–83, 2018.

[35] Atefeh Daemi, Yousef Alipouri, and Biao Huang. Identification of robust gaussian process regression with noisy input using em algorithm. *Chemometrics and Intelligent Laboratory Systems*, 191:1–11, 2019.

[36] Jinlin Zhu, Zhiqiang Ge, and Zhihuan Song. Robust modeling of mixture probabilistic principal component analysis and process monitoring application. *AIChE journal*, 60(6):2143–2157, 2014.

[37] Junhua Zheng, Jinlin Zhu, Guangjie Chen, Zhihuan Song, and Zhiqiang Ge. Dynamic bayesian network for robust latent variable modeling and fault classification. *Engineering Applications of Artificial Intelligence*, 89:103475, 2020.

[38] Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing*, 71(7-9):1274–1282, 2008.

[39] Wonkeun Youn, Yulong Huang, and Hyun Myung. Outlier-robust student's-t-based IMM-VB localization for manned aircraft using tdoa measurements. *IEEE/ASME Transactions on Mechatronics*, 2020.

[40] Jingbo Wang, Weiming Shao, and Zhihuan Song. Semi-supervised variational bayesian student'st mixture regression and robust inferential sensor application. *Control Engineering Practice*, 92:104155, 2019.

[41] Weixing Song, Weixin Yao, and Yanru Xing. Robust mixture regression model fitting by laplace distribution. *Computational Statistics & Data Analysis*, 71:128–137, 2014.

[42] Hien D Nguyen, Geoffrey J McLachlan, Jeremy FP Ullmann, and Andrew L Janke. Laplace mixture autoregressive models. *Statistics & Probability Letters*, 110:18–24, 2016.

[43] Jinlin Zhu, Zhiqiang Ge, and Zhihuan Song. Robust semi-supervised mixture probabilistic principal component regression model development and application to soft sensors. *Journal of Process Control*, 32:25–37, 2015.

[44] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[45] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[46] Shabnam Sedghi and Biao Huang. Simultaneous estimation of sub-model number and parameters for mixture probability principal component regression. In *2017 11th Asian Control Conference (ASCC)*, pages 1069–1074. IEEE, 2017.

[47] Beata Walczak and Désiré L Massart. Dealing with missing data: Part II. *Chemometrics and Intelligent Laboratory Systems*, 58(1):29–42, 2001.

[48] Shima Khatibisepehr and Biao Huang. A bayesian approach to robust process identification with arx models. *AIChE Journal*, 59(3):845–859, 2013.

[49] Mengqi Fang, Hariprasad Kodamana, and Biao Huang. Real-time mode diagnosis for processes with multiple operating conditions using switching conditional random fields. *IEEE Transactions on Industrial Electronics*, 67(6):5060–5070, 2019.

[50] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. *Lazy learning*, pages 11–73, 1997.

[51] Peter Englert. Locally weighted learning. In *Seminar Class on Autonomous Learning Systems*. Citeseer, 2012.

[52] Zhiqiang Ge and Zhihuan Song. A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemometrics and Intelligent Laboratory Systems*, 104(2):306–317, 2010.

[53] Xiaofeng Yuan, Zhiqiang Ge, Biao Huang, and Zhihuan Song. A probabilistic just-in-time learning framework for soft sensor development with missing data. *IEEE Transactions on Control Systems Technology*, 25(3):1124–1132, 2016.

[54] Xiaofeng Yuan, Zhiqiang Ge, Zhihuan Song, Yalin Wang, Chunhua Yang, and Hongwei Zhang. Soft sensor modeling of nonlinear industrial processes based on weighted probabilistic projection regression. *IEEE Transactions on Instrumentation and Measurement*, 66(4):837–845, 2017.

[55] Xiaofeng Yuan, Zhiqiang Ge, Biao Huang, Zhihuan Song, and Yalin Wang. Semisupervised jitl framework for nonlinear industrial soft sensing based on locally semisupervised weighted pcr. *IEEE Transactions on Industrial Informatics*, 13(2):532–541, 2016.

[56] Le Yao and Zhiqiang Ge. Refining data-driven soft sensor modeling framework with variable time reconstruction. *Journal of Process Control*, 87:91–107, 2020.

[57] Jiabao Zhu, Jim Zurcher, Ming Rao, and Max QH Meng. An on-line wastewater quality predication system based on a time-delay neural network. *Engineering Applications of Artificial Intelligence*, 11(6):747–758, 1998.

[58] Le Yao and Zhiqiang Ge. Cooperative deep dynamic feature extraction and variable time-delay estimation for industrial quality prediction. *IEEE Transactions on Industrial Informatics*, 2020.

[59] Weili Xiong, Yanjun Li, Yujia Zhao, and Biao Huang. Adaptive soft sensor based on time difference gaussian process regression with local time-delay reconstruction. *Chemical Engineering Research and Design*, 117:670–680, 2017.

[60] NJI Mars and GW Van Arragon. Time delay estimation in non-linear systems using average amount of mutual information analysis. *Signal processing*, 4(2-3):139–153, 1982.

[61] Robin AA Ince, Bruno L Giordano, Christoph Kayser, Guillaume A Rousselet, Joachim Gross, and Philippe G Schyns. A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38(3):1541–1573, 2017.

[62] Wei Wang, Chunhua Yang, Jie Han, Wenting Li, and Yonggang Li. A soft sensor modeling method with dynamic time-delay estimation and its application in wastewater treatment plant. *Biochemical Engineering Journal*, page 108048, 2021.

[63] Bingyun Yan, Fei Yu, and Biao Huang. Generalization and comparative studies of similarity measures for just-in-time modeling. *IFAC-PapersOnLine*, 52(1):760–765, 2019.

[64] Tao Chen and Yue Sun. Probabilistic contribution analysis for statistical process monitoring: A missing variable approach. *Control Engineering Practice*, 17(4):469–477, 2009.

[65] YM Kang, WC Zhu, GY Chen, and XQ Yin. Cross correlation analysis and time delay estimation of acoustic emission signals of rock based on wavelet transform. *Rock and Soil Mechanics*, 32(7):2079–2084, 2011.

[66] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–473, 2006.

[67] Roberto Vio and Willem Wamsteker. Limits of the cross-correlation function in the analysis of short time series. *Publications of the Astronomical Society of the Pacific*, 113(779):86, 2001.

[68] Alireza Memarian, Santhosh Kumar Varanasi, and Biao Huang. Mixture robust semi-supervised probabilistic principal component regression with missing input data. *Chemometrics and Intelligent Laboratory Systems*, 214:104315, 2021.

[69] Andreas Manhart, Regine Vogt, Michael Priester, Günter Dehoust, Andreas Auberger, Markus Blepp, Peter Dolega, Claudia Kämper, Jürgen Giegrich, Gerhard Schmidt, et al. The environmental criticality of primary raw materials–a new methodology to assess global environmental hazard potentials of minerals and metals from mining. *Mineral Economics*, 32(1):91–107, 2019.

[70] Seema Mehta, Tanjul Saxena, and Neetu Purohit. The new consumer behaviour paradigm amid covid-19: Permanent or transient? *Journal of Health Management*, 22(2):291–301, 2020.

[71] Xinying Zhang, Huanhuan Xie, Xiaoyan Liu, Dewen Kong, Shenyu Zhang, and Chuanhua Wang. A novel green substrate made by sludge digestate and its biochar: Plant growth and greenhouse emission. *Science of The Total Environment*, page 149194, 2021.

[72] Claudio F Lima, Fernando G Lobo, Martin Pelikan, and David E Goldberg. Model accuracy in the bayesian optimization algorithm. *Soft Computing*, 15(7):1351–1371, 2011.

[73] Tafarel de Avila Ferreira, Harsh A Shukla, Timm Faulwasser, Colin N Jones, and Dominique Bonvin. Real-time optimization of uncertain process systems via modifier adaptation and gaussian processes. In *2018 European Control Conference (ECC)*, pages 465–470. IEEE, 2018.

[74] Alejandro Marchetti, B Chachuat, and Dominique Bonvin. Modifier-adaptation methodology for real-time optimization. *Industrial & engineering chemistry research*, 48(13):6022–6033, 2009.

[75] Alejandro G Marchetti, Grégory François, Timm Faulwasser, and Dominique Bonvin. Modifier adaptation for real-time optimization—methods and applications. *Processes*, 4(4):55, 2016.

[76] Erika Oliveira-Silva, Cesar de Prada, and Daniel Navia. Economic mpc with modifier adaptation using transient measurements. In *Computer Aided Chemical Engineering*, volume 50, pages 1253–1258. Elsevier, 2021.

[77] Ehecatl Antonio del Rio Chanona, JE Alves Graciano, Eric Bradford, and

Benoit Chachuat. Modifier-adaptation schemes employing gaussian processes and trust regions for real-time optimization. *IFAC-PapersOnLine*, 52(1):52–57, 2019.

[78] Dong Hwi Jeong, Chang Jun Lee, and Jong Min Lee. Experimental gradient estimation of multivariable systems with correlation by various regression methods and its application to modifier adaptation. *Journal of Process Control*, 70:65–79, 2018.

[79] D Navia, L Briceño, G Gutiérrez, and C De Prada. Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem. *Industrial & engineering chemistry research*, 54(48):12054–12071, 2015.

[80] Alejandro Marchetti, Benoit Chachuat, and Dominique Bonvin. A dual modifier-adaptation approach for real-time optimization. *Journal of Process Control*, 20(9):1027–1037, 2010.

[81] Weihua Gao, Simon Wenzel, and Sebastian Engell. A reliable modifier-adaptation strategy for real-time optimization. *Computers & chemical engineering*, 91:318–328, 2016.

[82] Ehecatl Antonio del Rio Chanona, Panagiotis Petsagkourakis, Eric Bradford, JE Alves Graciano, and Benoît Chachuat. Real-time optimization meets bayesian optimization and derivative-free optimization: A tale of modifier adaptation. *Computers & Chemical Engineering*, 147:107249, 2021.

[83] Kate Nussenbaum and Catherine A Hartley. Reinforcement learning across development: What insights can we draw from a decade of research? *Developmental cognitive neuroscience*, 40:100733, 2019.

[84] Jincong He, Meng Tang, Chaoshun Hu, Shusei Tanaka, Kainan Wang, Xian-Huan Wen, and Yusuf Nasir. Deep reinforcement learning for generalizable field development optimization. *SPE Journal*, pages 1–20, 2021.

[85] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[86] Tyng-Luh Liu and Hwann-Tzong Chen. Real-time tracking using trust-region methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):397–402, 2004.

[87] Kody M Powell, Derek Machalek, and Titus Quah. Real-time optimization using reinforcement learning. *Computers & Chemical Engineering*, 143:107077, 2020.

[88] Titus Quah, Derek Machalek, and Kody M Powell. Comparing reinforcement learning methods for real-time optimization of a chemical process. *Processes*, 8(11):1497, 2020.

[89] Alireza Memarian, Santhosh Kumar Varanasi, and Biao Huang. Soft sensor development in the presence of missing input and delayed output data through weighted semi-supervised probabilistic principal component regression (under

review). *IEEE Transactions on Industrial Electronics.*

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

[91] David Duvenaud. *Automatic model construction with Gaussian processes.* PhD thesis, University of Cambridge, 2014.

[92] Atefeh Daemi, Hariprasad Kodamana, and Biao Huang. Gaussian process modelling with gaussian mixture likelihood. *Journal of Process Control*, 81:209–220, 2019.

[93] Zhaoyi Xu, Yanjie Guo, and Joseph H Saleh. Efficient hybrid bayesian optimization algorithm with adaptive expected improvement acquisition function. *Engineering Optimization*, pages 1–19, 2020.

[94] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[95] Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

[96] Syrine Belakaria and Aryan Deshwal. Max-value entropy search for multi-objective bayesian optimization. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[97] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[98] John Emsley. *Nature's building blocks: an AZ guide to the elements.* Oxford University Press, 2011.

[99] PRIMARY COPPER SMELTING. Background report ap-42 section 12.3 primary copper smelting.

[100] US Environmental. Protection agency. 1990. *Quality assurance project plan for characterization sampling and treatment tests conducted for the Contaminated Soil and Debris (CSD) Program: Washington, DC, USEPA Office of Solid Waste*, 1990.

[101] Dinesh Krishnamoorthy, Bjarne Foss, and Sigurd Skogestad. Steady-state real-time optimization using transient measurements. *Computers & Chemical Engineering*, 115:34–45, 2018.

Appendices

Appendix A

Mixing Proportions

$$\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \log p(k \mid \boldsymbol{\theta}) + \lambda \left(\sum_{k=1}^{K} p(k \mid \boldsymbol{\theta}) - 1 \right) = E(L_2) = Q_6 \quad \text{(A.1)}$$

$$\frac{\partial E(L_2)}{\partial p(k \mid \boldsymbol{\theta})} = 0 \Longrightarrow \sum_{i=1}^{n_1} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) + \lambda \sum_{k=1}^{K} p(k \mid \boldsymbol{\theta}) = 0 \Longrightarrow \lambda = -n_1 \quad \text{(A.2)}$$

$$\sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \log p(k \mid \boldsymbol{\theta}) + \lambda \left(\sum_{k=1}^{K} p(k \mid \boldsymbol{\theta}) - 1 \right) = E(L_3) = Q_5^{'} \quad \text{(A.3)}$$

$$\frac{\partial E(L_3)}{\partial p(k \mid \boldsymbol{\theta})} = 0 \Longrightarrow \sum_{i=n_1+1}^{n} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) + \lambda \sum_{k=1}^{K} p(k \mid \boldsymbol{\theta}) = 0 \Longrightarrow \lambda = -(n-n_1) \quad \text{(A.4)}$$

Appendix B

Q-function Terms

The variables given below are used to simplify the notations while defining terms in $Q - function$.

$$
\begin{aligned}
P_{\star,\triangle} &= p(q_{x_{i,k}} = \star \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, k, \boldsymbol{\theta}^{old}) \times p(q_{y_{i,k}} = \triangle \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, k, \boldsymbol{\theta}^{old}) \\
P^{'}_{\star,\triangle} &= p(q_{x_{i,k}} = \star \mid \boldsymbol{x}_{i,o}, k, \boldsymbol{\theta}^{old}) \times p(q_{y_{i,k}} = \triangle \mid \boldsymbol{x}_{i,o}, k, \boldsymbol{\theta}^{old}) \\
E_{\star,\triangle} &= E(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, q_{x_{i,k}} = \star, q_{y_{i,k}} = \triangle, k, \boldsymbol{\theta}^{old}) \\
E^{'}_{\star,\triangle} &= E(\boldsymbol{t}_{i,k} \mid \boldsymbol{x}_{i,o}, q_{x_{i,k}} = \star, q_{y_{i,k}} = \triangle, k, \boldsymbol{\theta}^{old}) \\
E_{\star,\triangle}(\boldsymbol{t}_{i,k}, \boldsymbol{t}^T_{i,k}) &= E(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k} \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, q_{x_{i,k}} = \star, q_{y_{i,k}} = \triangle, k, \boldsymbol{\theta}^{old}) \\
E^{'}_{\star,\triangle}(\boldsymbol{t}_{i,k}, \boldsymbol{t}^T_{i,k}) &= E(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k} \mid \boldsymbol{x}_{i,o}, q_{x_{i,k}} = \star, q_{y_{i,k}} = \triangle, k, \boldsymbol{\theta}^{old})
\end{aligned}
\tag{B.1}
$$

where $\star$ and $\triangle$ can be either 1 or ρ. The expressions for $Q - function$ are as follows

$$
\begin{aligned}
&Q_1 = \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) P_{1,1}[\frac{-1}{2}\log(2^m\pi^m \times |\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{1,1}^T\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) \\
&- (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE_{1,1} + tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}E_{1,1}^T)) + E_{1,1}^T\boldsymbol{P}_k^T\boldsymbol{P}_kE_{1,1})]+ \\
&\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) P_{\rho,1}[\frac{-1}{2}\log(2^m\pi^m \times |\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\times \\
&\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{\rho,1}^T\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - \\
&(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,1} + \boldsymbol{\rho}_{x,k}tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}E_{1\rho,1}^T))+ \\
&E_{\rho,1}^T\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,1})] \\
&+\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) P_{1,\rho}[\frac{-1}{2}\log((2\pi)^m \times |\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{1,\rho}^T\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) \\
&- (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE_{1,\rho} + tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}E_{1,\rho}^T)) + E_{1,\rho}^T\boldsymbol{P}_k^T\boldsymbol{P}_kE_{1,\rho})]+ \\
&\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) P_{\rho,\rho}[\frac{-1}{2}\log((2\pi)^m \times |\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\times \\
&\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{\rho,\rho}^T\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - \\
&(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,\rho} + \boldsymbol{\rho}_{x,k}tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}E_{\rho,\rho}^T))+ \\
&E_{\rho,\rho}^T\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,\rho})]
\end{aligned} \tag{B.2}
$$

$$
\begin{aligned}
Q_4 =&\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{1,1} \times [\log(1-\delta_{x,k})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{\rho,1} \times [\log(\delta_{x,k})] \\
&+\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{1,\rho} \times [\log(1-\delta_{x,k})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{\rho,\rho} \times [\log(\delta_{x,k})]
\end{aligned} \tag{B.3}
$$

$$
\begin{aligned}
Q_3^{'} =& \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{1,1}^{'} \times [\log(1-\delta_{x,k})] + \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{\rho,1}^{'} \times [\log(\delta_{x,k})] \\
&+ \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{1,\rho}^{'} \times [\log(1-\delta_{x,k})] + \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{\rho,\rho}^{'} \times [\log(\delta_{x,k})]
\end{aligned} \tag{B.4}
$$

$$
\begin{aligned}
Q_5 =&\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{1,1} \times [\log(1-\delta_{y,k})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{\rho,1} \times [\log(1-\delta_{y,k})] \\
&+\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{1,\rho} \times [\log(\delta_{y,k})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) \times P_{\rho,\rho} \times [\log(\delta_{y,k})]
\end{aligned} \tag{B.5}
$$

$$
\begin{aligned}
Q_4^{'} =& \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{1,1}^{'} \times [\log(1-\delta_{y,k})] + \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{\rho,1}^{'} \times [\log(1-\delta_{y,k})] \\
&+ \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{1,\rho}^{'} \times [\log(\delta_{y,k})] + \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) \times P_{\rho,\rho}^{'} \times [\log(\delta_{y,k})]
\end{aligned} \tag{B.6}
$$

$$\begin{aligned}
&Q_1^{'} = \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})P_{1,1}^{'}[\frac{-1}{2}\log((2\pi)^m \times |\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{1,1}^{'T}\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) \\
&- (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE_{1,1}^{'} + tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{1,1}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}^{'}E_{1,1}^{'T})) + E_{1,1}^{'T}\boldsymbol{P}_k^T\boldsymbol{P}_kE_{1,1}^{'})]+ \\
&\sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})P_{\rho,1}^{'}[\frac{-1}{2}\log((2\pi)^m \times |\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times \boldsymbol{\rho}_{x,k} \times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{\rho,1}^{'T}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})- \\
&(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = 1, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,1}^{'} + \boldsymbol{\rho}_{x,k}tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{\rho,1}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}^{'}E_{\rho,1}^{'T})) + E_{\rho,1}^{'T}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,1}^{'})]+ \\
&\sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})P_{1,\rho}^{'}[\frac{-1}{2}\log((2\pi)^m \times |\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{1,\rho}^{'T}\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) \\
&- (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE_{1,\rho}^{'} + tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{1,\rho}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}^{'}E_{1,\rho}^{'T})) + E_{1,\rho}^{'T}\boldsymbol{P}_k^T\boldsymbol{P}_kE_{1,\rho}^{'})]+ \\
&\sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})P_{\rho,\rho}^{'}[\frac{-1}{2}\log((2\pi)^m \times |\boldsymbol{\rho}_{x,k}^{-1}\sigma_{x,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{x,k}^{-2}((E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T \\
&\times \boldsymbol{\rho}_{x,k} \times (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - E_{\rho,\rho}^{'T}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})- \\
&(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \rho, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,\rho}^{'} + \boldsymbol{\rho}_{x,k}tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{\rho,\rho}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}^{'}E_{\rho,\rho}^{'T}))+ \\
&E_{\rho,\rho}^{'T}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,\rho}^{'})]
\end{aligned} \tag{B.7}$$

$$\begin{aligned}
Q_2 = &\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{1,1} \times [\frac{-1}{2}\log((2\pi)^r|\sigma_{y,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{y,k}^{-2}((\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - E_{1,1}^T\boldsymbol{C}_k^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})- \\
&(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{C}_kE_{1,1} + tr(\boldsymbol{C}_k^T\boldsymbol{C}_k(E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}E_{1,1}T)) + E_{1,1}^T\boldsymbol{C}_k^T\boldsymbol{C}_kE_{1,1})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{\rho,1} \\
&\times [\frac{-1}{2}\log((2\pi)^r|\sigma_{y,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{y,k}^{-2}((\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - E_{\rho,1}^T\boldsymbol{C}_k^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{C}_kE_{\rho,1}+ \\
&tr(\boldsymbol{C}_k^T\boldsymbol{C}_k(E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}E_{\rho,1}T)) + E_{\rho,1}^T\boldsymbol{C}_k^T\boldsymbol{C}_kE_{\rho,1})] + \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{1,\rho} \times [\frac{-1}{2}\log((2\pi)^r|\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I}|) \\
&- \frac{1}{2}\sigma_{y,k}^{-2}((\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{\rho}_{y,k}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - E_{1,\rho}^T\boldsymbol{C}_k^T\boldsymbol{\rho}_{y,k}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{\rho}_{y,k}\boldsymbol{C}_kE_{1,\rho}+ \\
&\boldsymbol{\rho}_{y,k}tr(\boldsymbol{C}_k^T\boldsymbol{C}_k(E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}E_{1,\rho}T)) + E_{1,\rho}^T\boldsymbol{C}_k^T\boldsymbol{\rho}_{y,k}\boldsymbol{C}_kE_{1,\rho})]+ \\
&\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{\rho,\rho} \times [\frac{-1}{2}\log((2\pi)^r|\boldsymbol{\rho}_{y,k}^{-1}\sigma_{y,k}^2\boldsymbol{I}|) - \frac{1}{2}\sigma_{y,k}^{-2}((\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{\rho}_{y,k}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})- \\
&E_{\rho,\rho}^T\boldsymbol{C}_k^T\boldsymbol{\rho}_{y,k}(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{\rho}_{y,k}\boldsymbol{C}_kE_{\rho,\rho} + \boldsymbol{\rho}_{y,k}tr(\boldsymbol{C}_k^T\boldsymbol{C}_k(E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}E_{\rho,\rho}T)) + E_{\rho,\rho}^T\boldsymbol{C}_k^T\boldsymbol{\rho}_{y,k}\boldsymbol{C}_kE_{\rho,\rho})]
\end{aligned} \tag{B.8}$$

$$\begin{aligned}
Q_3 = &\sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{1,1} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{1,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}E_{1,1}^T) + E_{1,1}^TE_{1,1})] \\
&+ \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{\rho,1} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{\rho,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}E_{\rho,1}^T) + E_{\rho,1}^TE_{\rho,1})] \\
&+ \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{1,\rho} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{1,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}E_{1,\rho}^T) + E_{1,\rho}^TE_{1,\rho})] \\
&+ \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})P_{\rho,\rho} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{\rho,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}E_{\rho,\rho}^T) + E_{\rho,\rho}^TE_{\rho,\rho})]
\end{aligned} \tag{B.9}$$

$$
\begin{aligned}
Q_2^{'} = &\sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) P_{1,1}^{'} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{1,1}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,1}^{'}E_{1,1}^{'T}) + E_{1,1}^{'T}E_{1,1}^{'})] \\
&+ \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) P_{\rho,1}^{'} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{\rho,1}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,1}^{'}E_{\rho,1}^{'T}) + E_{\rho,1}^{'T}E_{\rho,1}^{'})] \\
&+ \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) P_{1,\rho}^{'} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{1,\rho}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{1,\rho}^{'}E_{1,\rho}^{'T}) + E_{1,\rho}^{'T}E_{1,\rho}^{'})] \\
&+ \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k \mid \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) P_{\rho,\rho}^{'} \times [\frac{-1}{2}\log((2\pi)^q|I|) - \frac{1}{2}(tr(E_{\rho,\rho}^{'}(\boldsymbol{t}_{i,k}\boldsymbol{t}_{i,k}^T) - E_{\rho,\rho}^{'}E_{\rho,\rho}^{'T}) + E_{\rho,\rho}^{'T}E_{\rho,\rho}^{'})]
\end{aligned}
\tag{B.10}
$$

Appendix C

Definitions and covariance computation

C.1 $\sigma^2_{x,k}$'s Terms Definitions

$$\begin{aligned}
&A_{1,\triangle} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - \\
&E^T_{1,\triangle}\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE_{1,\triangle} \\
&+ tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E_{1,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,\triangle}E^T_{1,\triangle})) + E^T_{1,\triangle}\boldsymbol{P}_k^T\boldsymbol{P}_kE_{1,\triangle} + tr(cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}))
\end{aligned} \tag{C.1}$$

$$\begin{aligned}
&A^{'}_{1,\triangle} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) \\
&- E^{'T}_{1,\triangle}\boldsymbol{P}_k^T(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{P}_kE^{'}_{1,\triangle} \\
&+ tr(\boldsymbol{P}_k^T\boldsymbol{P}_k(E^{'}_{1,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,\triangle}E^{'T}_{1,\triangle})) + E^{'T}_{1,\triangle}\boldsymbol{P}_k^T\boldsymbol{P}_kE^{'}_{1,\triangle} + tr(cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}))
\end{aligned} \tag{C.2}$$

$$\begin{aligned}
&A^{'\star}_{\rho,\triangle} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - \\
&E^{'T}_{\rho,\triangle}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE^{'}_{\rho,\triangle} \\
&+ tr(\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_k(E^{'}_{\rho!}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,\triangle}E^{'T}_{\rho,\triangle})) + E^{'T}_{\rho,\triangle}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE^{'}_{\rho,\triangle} + tr(cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}))
\end{aligned} \tag{C.3}$$

$$\begin{aligned}
&A^{\star}_{\rho,\triangle} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - \\
&E^T_{\rho,\triangle}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}(E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k}) - (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} =, \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) - \boldsymbol{\mu}_{x,k})^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_k \times E_{\rho,\triangle} \\
&+ tr(\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_k(E_{\rho,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,\triangle}E^T_{\rho,\triangle})) + E^T_{\rho,,\triangle}\boldsymbol{P}_k^T\boldsymbol{\rho}_{x,k}\boldsymbol{P}_kE_{\rho,\triangle} + tr(cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}))
\end{aligned} \tag{C.4}$$

C.2 $\sigma^2_{y,k}$'s Terms Definitions

$$\begin{aligned}
B_{\star,1} = (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - E^T_{\star,1}\boldsymbol{C}_k^T(\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T\boldsymbol{C}_kE_{\star,1} \\
+ tr(\boldsymbol{C}_k^T\boldsymbol{C}_k(E_{\star,1}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\star,1}E^T_{\star,1})) + E^T_{\star,1}\boldsymbol{C}_k^T\boldsymbol{C}_kE_{\star,1}
\end{aligned} \tag{C.5}$$

$$\begin{aligned} B_{\star,\rho} = (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T \boldsymbol{\rho}_{y,k} (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - E^T_{\star,\rho} \boldsymbol{C}^T_k \boldsymbol{\rho}_{y,k} (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k}) - (\boldsymbol{y}_i - \boldsymbol{\mu}_{y,k})^T \boldsymbol{\rho}_{y,k} \boldsymbol{C}_k E_{\star,\rho} \\ + tr(\boldsymbol{C}^T_k \boldsymbol{\rho}_{y,k} \boldsymbol{C}_k (E_{\star,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\star,\rho}E^T_{\star,\rho})) + E^T_{\star,\rho} \boldsymbol{C}^T_k \boldsymbol{\rho}_{y,k} \boldsymbol{C}_k E_{\star,\rho} \end{aligned} \tag{C.6}$$

C.3 ρ_{x,k_j}'s Terms Definitions

$$\begin{aligned} C_{\rho,\triangle_j} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})^T (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j}) - \\ E^T_{\rho,\triangle} P^T_{k_j} (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}\rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j}) - E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}}\rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})^T P_{k_j} E_{\rho,\triangle} \\ + tr(P^T_{k_j} P_{k_j} (E_{\rho,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,\triangle}E^T_{\rho,\triangle})) + E^T_{\rho,\triangle} P^T_{k_j} P_{k_j} E_{\rho,\triangle} + (cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}}\rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}))_{jj} \end{aligned} \tag{C.7}$$

$$\begin{aligned} C^{'}_{\rho,\triangle_j} = (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})^T (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j}) - \\ E^{'T}_{\rho,\triangle} P^T_{kj} (E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j}) - E(\boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})_j - \mu_{x,k_j})^T P_{k_j} E^{'}_{\rho,\triangle} \\ + tr(P^T_{k_j} P_{k_j} (E^{'}_{\rho,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,\triangle}E^{'T}_{\rho,\triangle})) + E^{'T}_{\rho,\triangle} P^T_{k_j} P_{k_j} E^{'}_{\rho,\triangle} + (cov(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}))_{jj} \end{aligned} \tag{C.8}$$

where j=1,..., m.

C.4 ρ_{y,k_j}'s Terms Definitions

$$\begin{aligned} D_{\star,\rho_j} = (y_{i,j} - \mu_{y,k_j})^T (y_{i,j} - \mu_{y,k_j}) - E^T_{\star,\rho} C^T_{k,j} (y_{i,j} - \mu_{y,k_j}) - (y_{i,j} - \mu_{y,k_j})^T C_{k,j} E_{\star,\rho} \\ + tr(C^T_{k,j} C_{k,j} (E_{\star,\rho}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\star,\rho}E^T_{\star,\rho})) + E^T_{\star,\rho} C^T_{k,j} C_{k,j} E_{\star,\rho} \end{aligned} \tag{C.9}$$

where j=1,..., r.

C.5 Covariance Calculations

For labeled dataset:

$$\begin{aligned} cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = P_{k,m}[E_{1,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{1,\triangle}E^T_{1,\triangle}]P^T_{k,m} + \sigma^2_{x,k,m} I \\ + E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})^T \end{aligned} \tag{C.10}$$

$$\begin{aligned} cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) = P_{k,m}[E_{\rho,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E_{\rho,\triangle}E^T_{\rho,\triangle}]P^T_{k,m} + \boldsymbol{\rho}^{-1}_{x,k,m}\sigma^2_{x,k,m} I \\ + E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old}) E(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{y}_i, \boldsymbol{\theta}^{old})^T \end{aligned} \tag{C.11}$$

And for unlabeled dataset:

$$\begin{aligned} cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = P_{k,m}[E^{'}_{1,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{1,\triangle}E^{'T}_{1,\triangle}]P^T_{k,m} + \sigma^2_{x,k,m} I \\ + E^{'}(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) E^{'}(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = 1, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})^T \end{aligned} \tag{C.12}$$

$$\begin{aligned} cov(\boldsymbol{x}_{i,k,m}, \boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) = P_{k,m}[E^{'}_{\rho,\triangle}(\boldsymbol{t}_{i,k}\boldsymbol{t}^T_{i,k}) - E^{'}_{\rho,\triangle}E^{'T}_{\rho,\triangle}]P^T_{k,m} + \boldsymbol{\rho}^{-1}_{x,k,m}\sigma^2_{x,k,m} I \\ + E^{'}(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old}) E^{'}(\boldsymbol{x}_{i,k,m} \mid q_{x_{i,k}} = \rho, q_{y_{i,k}} = \triangle, k, \boldsymbol{x}_{i,o}, \boldsymbol{\theta}^{old})^T \end{aligned} \tag{C.13}$$