Predictive Models for Quality Variables with Regular Sensors and Images

by

Yousef Salehi

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

Doctor of Philosophy

in

Process Control

Department of Chemical and Materials Engineering
University of Alberta

Abstract

In the process industry, certain quality variables cannot be measured regularly due to technical limitations or economic constraints. Consequently, the industry relies on laboratory analysis to measure such quality variables. However, laboratory analysis introduces long time-delays in obtaining measurements due to the need to collect representative samples, transport them, and conduct the analysis. The associated delay can pose challenges in terms of timely decision-making and real-time control. Therefore, fast-rate measurements in the process industry are crucial for real-time quality monitoring and control. By obtaining real-time data, industries can improve efficiency, meet customer expectations, and minimize risks associated with the quality variables.

Data-driven predictive models, also known as soft sensors, have emerged as valuable tools in predicting quality variables in the process industry. The predictive models employ advanced data analysis techniques to predict the values of quality variables using different data sources including regularly measured variables by online sensors and visual information provided by video cameras. By leveraging historical data and identifying patterns, the models can provide fast-rate predictions of quality variables. This capability enables proactive decision-making and facilitates timely interventions to maintain product quality and process control, and contribute to process optimization.

In this thesis, we develop predictive models based on both conventional sensor measurements and image data, each with its own advantages. The first two contributions are related to conventional sensors data and the following two contributions are concerned with image data as a means to develop predictive models. The main contributions of this thesis are listed as follows.

The first contribution involves developing a statistical predictive model for quality variables with simultaneous consideration of time-varying time-delays, time-varying sample collection periods, and varying operating points. A non-parametric distribution is used to describe the distributions of the time-delays, sample collection periods, and switching of different operating modes, eliminating the need for prior knowledge about the distributions. Furthermore, the work enables online updating of the model parameters using a recursive Expectation-Maximization (EM) algorithm.

Then, we extend the linear time invariant predictive model to a linear parameter varying (LPV) predictive model, and enhance robustness to outlying output observations through the use of t-distribution. Additionally, uncertainty of the unknown model parameters is estimated using variational Bayesian (VB) algorithm.

Development of a computer vision model to predict quality variables is the third contribution. A modified Kalman filter is formulated to restore degraded images caused by factors like lighting conditions changes and camera noise. Additionally, to estimate the predictive model parameters, a robust-to-outlier EM algorithm is developed. The proposed model was validated on a tailing flotation process.

In the last contribution, the development of a computer vision model that enables fast-rate prediction of quality variables is considered. To address the impact of environmental conditions like steam and lighting on images, an atuoencoder-based image inpainting algorithm is developed to fill in the missing regions in the images. The restored images, along with slow-rate sampled measurements, are then used in conjunction with the EM algorithm to construct an auto-regressive with eXogenous input (ARX) predictive model.

All the proposed models in this thesis have been validated through experimental studies.

Preface

This thesis is an original work conducted by Yousef Salehi under the supervision of Dr. Biao Huang and is funded in part by Natural Sciences and Engineering Research Council (NSERC) of Canada. Portions of the thesis have been published in peer-reviewed journals.

1. Chapter 3 of this thesis has been published as **Y. Salehi** and B. Huang, "Offline and online parameter learning for switching multirate processes with varying delays and integrated measurements", in *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 7213-7222, July 2021.

2. Chapter 4 of this thesis has been published as **Y. Salehi**, B. Huang. "Robust variational Bayesian-based soft sensor model for LPV processes with delayed and integrated measurements", in *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-10, Art no. 1006310, Aug. 2022.

3. Chapter 5 of this thesis has been accepted for publication as **Yousef Salehi**, Kaiyu Zhou, Biao Huang, Xuehua Zhang, "Image restoration and analysis with application to quality variable prediction in flotation process", in *Journal of Process Control*, Sept. 2023.

The remaining chapters have also been either published in conference proceedings or under review for journal publications. They include:

- A short version of Chapter 5 has also been accepted for publication as **Yousef Salehi**, Amir Mohseni, Kaiyu Zhou, Biao Huang, Xuehua Zhang, "A computer vision system for bitumen content estimation in flotation froth with degraded images", in *IFAC World Congress*, 2023.

- Chapter 6 of this thesis is under review as **Yousef Salehi**, Ranjith Chiplunkar, and Biao Huang, "Robust machine vision model for predicting quality variables with slow-rate measurements", in *Computers and Chemical Engineering*, 2023.

This thesis has therefore been written and organized in paper format.

To my loving family and close friends.

Acknowledgements

Above all, I would like to express my heartfelt appreciation to my supervisor, Dr. Biao Huang. His guidance and unwavering support have been invaluable throughout this journey. I am thankful for all the discussions, the opportunities he has provided, and the freedom he has given me during my Ph.D. studies. I am also grateful for the opportunity to be exposed to real-world process systems engineering problems through industrial projects with leading oil sands companies, which he facilitated.

I want to express my genuine gratitude to Dr. Ranjith Chiplunkar, whose collaborations have significantly propelled my research forward. I also wish to convey a special acknowledgment to Dr. Fadi Ibrahim, Mrs. Terry Runyon , and Dr. Yousef Alipouri for their priceless assistance, as well as to our skilled group members and friends who have eased the path of this journey.

I extend my gratitude to Dr. Xuehua Zhang and Kaiyu Zhou for their invaluable contribution in providing the data related to tailing froth. This data played a crucial role in validating a section of my research, and I am grateful for their contribution. Further, I would like to thank the Institute for Oil Sands Innovation (IOSI) laboratory, the Director of IOSI, Natalia Semagina, and the IOSI laboratory technician, Brittany Mackinnon, for their support.

I would also like to thank our industrial partners and acknowledge the financial support I received from the Natural Sciences and Engineering Research Council (NSERC) of Canada. Furthermore, I am grateful to the University of Alberta and the Department of Chemical and Materials Engineering for their financial support and provision of other necessary resources that have enabled me to conduct my research.

Contents

List of Tables

List of Figures

Abbreviations

AI Artificial intelligence.

ARX Auto-regressive with exogenous input.

BJ Box-Jenkins.

CNN Convolutional neural network.

ELBO Evidence lower bound.

EM Expectation-Maximization.

FFT Fast Fourier transform.

FIR Finite impulse response.

GAN Generative adversarial network.

GEM Generalized Expectation-Maximization.

GLCM Gray level co-occurrence matrix.

GPR Gaussian process regression.

HSV Hue-saturation-Value.

HTT Hybrid three tank.

IOSI Institute for oil sands innovation.

KF Kalman filter.

KL Kullback-Leibler.

LBP Local binary pattern.

LPV Linear parameter varying.

LS Least squares.

LSTM Long short-term memory.

LTI Linear time invariant.

MAP Maximum *a posteriori*.

MCMC Markov chain Monte Carlo.

MLE Maximum likelihood estimation.

NSERC Natural Sciences and Engineering Research Council.

OE Output error.

PCA Principal component analysis.

PDF Probability distribution function.

PLS Partial least squares.

PRBS Pseudorandom binary signal.

PSC Primary separation cell.

ReLU Rectified Linear Unit.

RGB Red-green-blue.

RMSE Root mean square error.

RTS Rauch-Tung-Striebel.

SVR Support vector regression.

UKF Unscented Kalman filter.

VB Variational Bayesian.

Chapter 1

Introduction

1.1 Motivation

Real-time and accurate measurement of key variables are crucial to process control and optimization applications. Certain industrial processes, such as chemical processes, often encounter technical or economical limitations in measuring quality variables, such as chemical product concentration. As a result, the measurement of quality variables only occurs at a slow rate in laboratory (lab) settings. The lack of frequent measurement for quality variables hampers the implementation of effective control and optimization strategies, impeding timely adjustments, deviation identification, and prompt response to changing conditions. As a result, process performance may be suboptimal, quality may be impaired, and resources may be overused.

To address this issue, the development of predictive models becomes essential. By leveraging advanced data analytics techniques, predictive models can be developed based on different data sources, primarily sensor measurements and image data, to provide fast-rate predictions for the slow-rate (quality) variables. The primary objective of predictive models is to bridge the gap between slow-rate measurements and the real-time demands of control and optimization applications. Through accurate predictions of quality variable values at a fast rate, these models facilitate monitoring, control, and optimization of industrial processes.

Our research aims to tackle the difficulties presented by slow-rate measurements of quality variables in industrial processes. The main focus is on the development of predictive models that use two different sources of data, namely sensor data and image data, then employ advanced data analysis techniques. By offering timely measurement of important quality variables, these models enable timely decision-making and efficient resource allocation, and ultimately drive advancements in industrial operations.

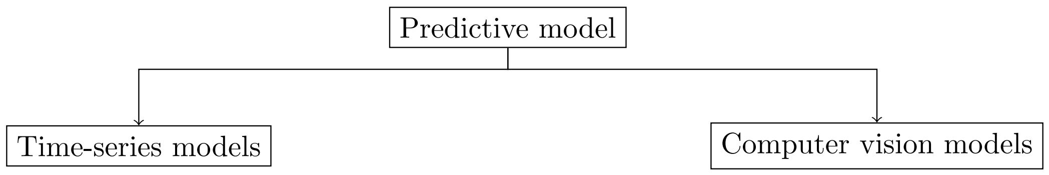


Figure 1.1: Data-driven predictive models based on the input data

1.2 Predictive Model Development

Predictive models, also referred to as soft sensors or inferential sensors, are essential tools used in the process industry to predict the hard-to-measure or non-measurable variables based on historical data, including sensor data and image data. These models leverage statistics, machine learning, and artificial intelligence to make informed predictions. By analyzing the data, they establish connections between input variables and target variable(s), enabling decision-makers in the process industry to gain valuable insights, predict trends, and make proper decisions [1]. According to the source of data used, *i.e.* sensor data and image data, we have divided predictive models into two categories: time-series predictive models and computer vision models, as shown in Figure 1.1.

1.2.1 Time-Series Models

Generally, there are three approaches for predictive model development, namely first-principles, data-driven, and gray-box models. The latter combines both. In order to develop first-principles models, theoretical knowledge about the underlying mechanisms is required [2]. The use of first-principles models is useful, but it is difficult to obtain due to the complexity and lack of thorough understanding of some processes. The use of data-driven models is common when a physical understanding of a process under investigation is lacking or incomplete. Data-driven models, which are constructed based on the historical relationships between the existing sensor measurements and target variable(s) measurements, thereby avoiding the investigation of complex chemical and physical phenomena [3]. These models incorporate statistical and machine learning techniques such as partial least squares (PLS) [4], support vector regression (SVR) [5], Gaussian process regression (GPR) [6]. Also, deep architectures such as stacked autoencoders [7], long short-term memory (LSTM) networks [8] are widely used to estimate variables of interest based on the sensor measurements [9], enabling continuous monitoring and control of critical process parameters. We will discuss some challenges associated with the data-driven approach in 1.3.

1.2.2 Computer Vision Models

With the advancement of artificial intelligence (AI), the gap between humans and machines is narrowing rapidly. There are numerous researchers and enthusiasts exploring different aspects of AI and achieving remarkable innovations. Computer vision stands out as one of these innovations. A key objective of computer vision is to enable machines to perceive the world similar to what humans do. It covers a wide range of tasks, such as object detection, tagging, recognition, classification, and analysis, as well as natural language processing [10].

In the process industry, monitoring cameras can be mounted in a proper position in order to visualize certain process behaviors, as has been demonstrated in numerous studies [11–14]. It enables non-intrusive and cost-effective monitoring and estimation of variables that are otherwise difficult or impossible to measure directly, offering valuable insights for process optimization and control. Computer vision models leverage image processing and machine learning techniques to predict the slow-rate sampled variables [15]. Through the analysis of images, the models can relate the relevant image features or patterns to the target process variables [16]. The advantage of computer vision predictive models lies in their non-intrusive nature, as they do not require physical contact with the subject. Moreover, they can capture spatial and temporal information, enabling real-time or near-real-time monitoring and control, and mimicking human visual perception to some degree. Accurate vision-based predictions, however, require high-quality images.

1.3 Common Problems and Related Works

This section reviews the common problems associated with sensor data and image data, as well as solutions that have been proposed to address the issues. Several of these challenges are common to both time-series and computer vision predictive models, but some of them are unique to the latter.

1.3.1 Sampling Rate

Instantaneous values of variables such as flow rate and temperature are routinely recorded using online sensors. On the other hand, it is technically or economically difficult to acquire the instant values of some quality variables. Samples of quality variables are commonly sent to lab for offline analysis. The lab measurements are generally more accurate compared with the routinely recorded measurements using online instruments. However, due to the sample collection time, transportation, and long processing time in lab, the measurements are available at a slow rate with considerable delays. Cement industry and oil sands processes are two examples among various industries where some quality variables are often measured through the lab analysis. A multirate data problem therefore occurs by the difference in measurement sampling rates.

The modeling of multirate systems has gained a significant attention by researchers

[17–19]. Down-sampling is one of the most frequently used methods in dealing with multirate systems where fast-rate data samples are downsampled in accordance with slow-measured variables, which will synchronize samples between the number of input and output data [20]. Based on this technique, [21] proposed a dynamic PLS model to predict the bottom composition of a pilot-scale distillation column utilizing online measurements of other process variables. In this straightforward method, some information will be lost, resulting in less accurate models. The authors in [22] combined the expectation-maximization (EM) algorithm with the Kalman filter (KF) to conduct state-space model identification from multirate data in the presence of irregularly sampled outputs. The work was concerned with linear time-invariant (LTI) systems. The EM-based identification method was further extended to nonlinear multirate systems with irregularly missing observations, where particle filter was used to approximate the required density functions [17]. In [23], the modeling of multirate processes are discussed in more detail.

1.3.2 Time-Delay

Time-delay often exists in industrial processes due to time-consuming lab analysis, unavoidable transport delays, etc, and can be divided into constant delay and time-varying delay categories. Common methods to estimate the time-delay are through the overparameterization method, non-parametric methods like step response and correlation analysis, gradient-based optimization, grid searching method, etc. In these methods, the estimation of model parameters is carried out independent of estimation of time-delays. The incorrect estimation of the delay will lead to biased parameter estimation [24]. To avoid this issue, the delay can be regarded as a hidden variable with a prior distribution such as uniform distribution over a relatively large range. An EM and generalized EM (GEM) have been used to solve the problem in [25] and [26], respectively. The complexity and uncertainty of offline analysis or communication network often lead to outputs with varying time-delays. The traditional identification algorithms such as least square (LS) are not applicable for handling systems with time-varying time-delays [27]. On the other hand, statistical methods are widely used to cope with incomplete data or missing variable problems. In [28], the EM algorithm was used to estimate the parameters of a dual-rate finite impulse response (FIR) model in which varying time-delays were assumed to follow a uniform distribution. However, [29] and [30] claimed that the variations in time-delays are not completely random, and there is a correlation between subsequent time-delays. Hence, a Markov chain was utilized to model the delays and the correlations between them. The EM and variational Bayesian (VB) algorithms were employed to estimate the parameters of a slow-rate model in [29] and [30], respectively. In [31] a VB algorithm for processes described by Auto-regressive with eXogenous input (ARX) model was studied to estimate the parameters of a multirate system subject to varying time-delays. The main difference between [31] and other aforementioned methods in handling the time-delays is that in [31] it is assumed that the accurate time-delay interval is unknown but the upper bound of the time-delay is known. In this statistical approach, the occurrence probability of different time-delays was determined by introducing a set of significance coefficients. Furthermore, the identification problem in the presence of time-varying time-delays has been addressed in other works [32]. A number of methods have been proposed to estimate time-delays [27, 33].

1.3.3 Integrated Samples

Chemical process samples are usually collected over a certain period of time, and then sent to a lab for analysis. This type of sampling is known as integrated sampling [34, 35]. Although integrated sampling is a common practice in chemical processes, most of the proposed solutions have ignored the integration in multirate modeling problem. Recently, this topic is gaining attention in process control research. [34] solved state estimation problem for states with infrequent, delayed, and integrated measurements. In [34], an augmented state-space model, consisting of fast-rate and integrated states, was defined. By fusing the fast and slow-rate measurements and applying a variable dimension unscented Kalman filter (UKF), the states were estimated. The state estimation and fusion problem in the presence of integrated outputs have been investigated in [36] as well. In [36], a modified KF called integrated measurement KF (IMKF) was utilized to estimate the integration of variables at a slow rate. After extracting fast-rate samples by a smoothing technique, the fusion of slow-rate state estimates and other fast-rate measurements improved the estimation results. Furthermore, the state estimation problem of the processes with integrated measurements in the presence of parametric uncertainties was addressed in [37]. The convergence properties of the proposed filter were derived in the work. The integrated measurement problem was investigated for nonlinear systems under the framework of Bayesian state estimation [38] where particle filter was used to deal with the estimation problem. Moreover, controller design for quality variables with integrated measurements was addressed in [39]. Recently, model identification problem of such processes was considered in [40]. In this work, FIR and ARX model structures were selected to model the fast-rate values of the integrated outputs where the EM algorithm was used to estimate the unknown parameters.

1.3.4 Outliers

The presence of outliers is another problem in data-driven modeling. An outlier is a data point that significantly differs from other observations, commonly occurring in real processes. Outlier can be caused by various factors, such as sensor malfunctions, unusual measurement noise, equipment failures, human errors, and unanticipated temporary disturbances, among others. Simply discarding outlier data points can lead to biased estimation [41]. To handle outliers, robust solutions have been proposed, including both deterministic and probabilistic approaches. The Gaussian distribution, commonly used as a noise model, lacks robustness to outliers. Various distributions, such as the t-distribution, contaminated Gaussian distribution, and

Laplace distribution, have been used to model data contaminated by outliers. The contaminated Gaussian distribution combines Gaussian distributions with different noise variance levels and weights [42]. However, the method is suitable only for modeling data with a limited degree of freedom. The t-distribution due to its longer tails, with a adjustable degree of freedom, is a more general approach to model the effect of abnormal data and has been widely used to cope with outliers in different applications [30, 43].

1.4 Image Processing

1.4.1 Image Representation

A digital picture is made up of tiny dots called pixels, and each pixel has a number that shows how bright it is. These numbers usually go from 0 to 255 for pictures that use 8-bit numbers. The higher the intensity value is, the higher is the brightness of the pixel. Therefore, each image can be defined as a two-dimensional matrix I, where $I(x, y)$ indicates the pixel intensity at position (x, y). In a gray-scale image, every pixel is represented by a single value, whereas in color images like red-green-blue (RGB) images, they are defined by a vector comprising three components. While color images offer additional information, they also bring about increased complexity when subjected to image processing and analysis.

Various forms of color images exist, each with its own way of representing and conveying visual information. The most common type is RGB, where each pixel is composed of three color channels that blend together to create a full spectrum of colors. It is widely used in digital displays, cameras, and computer graphics. In addition to RGB, there is also HSV (hue-saturation-value), which allows you to adjust and control colors more intuitively, which is useful in editing and manipulating images. Figure 1.2 shows a single image in three different representations. Further, infrared and thermal images, often used in scientific and industrial applications, reveal temperature variations through color variations. Furthermore, hyperspectral and multispectral images capture information that is not visible to the naked eye. In diverse industries and creative endeavors, each type of color image serves a specific purpose.

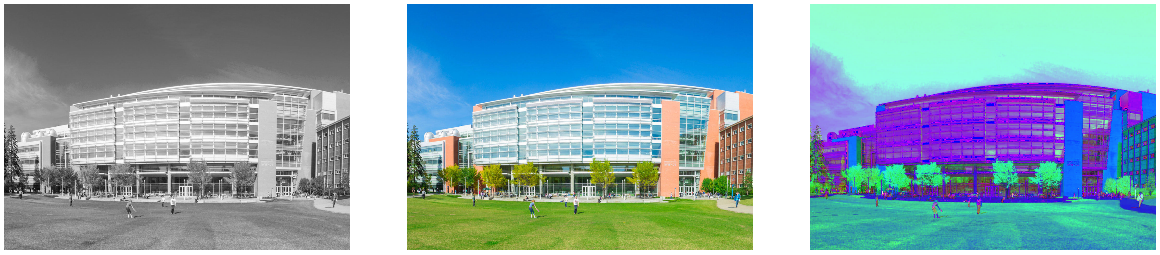
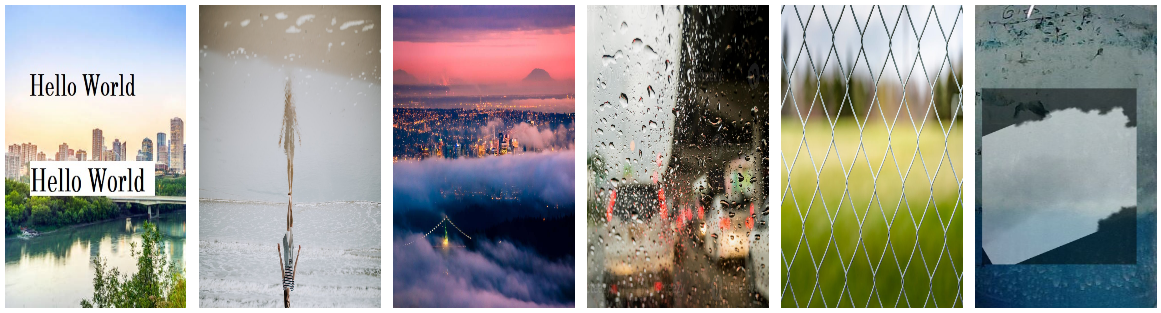

(a) Gray-scale image (b) RGB image (c) HSV image

Figure 1.2: Different representations of an image

1.4.2 Image Degradation

Real-world visual data often suffers from certain degradation. Some common examples of image degradation are shown in Figure 1.3. Image degradation in industrial settings can stem from various factors, including poor lighting conditions, dust and debris on cameras or lenses, occlusions, motion blur, and compression artifacts during data transmission or storage [44]. This issue can significantly impact the accuracy and efficiency of computer vision algorithms, leading to erroneous results and potentially compromising critical processes. Image restoration, also called image reconstruction, techniques, therefore, become critical in these scenarios [45]. By employing advanced algorithms for denoising, deblurring, and enhancing image resolution, computer vision systems can effectively counteract the detrimental effects of image degradation to certain degree. High-quality image restoration ensures that relevant features are preserved, enabling more accurate image analysis. Ultimately, incorporating image restoration into computer vision models in industrial applications can boost productivity, enhance product quality, and strengthen safety measures, making the overall processes more efficient, reliable, and cost-effective. A detailed description of image restoration methods can be found in [46, 47].



(a) (b) (c) (d) (e) (f)

Figure 1.3: Image degraded by: (a) Text (b) Reflection (c) Cloud (d) Rain (e) Unwanted object (f) Artificial steam.

1.4.3 Image Restoration

Image restoration is an essential part in the field of computer vision, which aims at predicting and filling the missing or damaged parts of images to achieve satisfactory visual effects. The restoration process uses mathematical algorithms and statistical models to correct various distortions like noise, blur, occlusion, artifacts and so on. In order to enhance visual quality, recover lost details, and improve overall perceptual clarity, restoration techniques analyze degradation factors and understanding how images form. Medical imaging, satellite imagery, historical documents, and industry all require accurate and unaltered visual information to be analyzed, diagnosed, and interpreted. For image restoration, a variety of approaches have been proposed, including traditional methods and deep learning techniques.

- **Traditional image restoration methods:** The traditional approach to image restoration relies on the principles of maintaining texture coherence and preserving content similarity. Based on mathematical and physical principles, these methods adapt based on the extent of image damage. When dealing with minor image distortions, the restoration process involves reconstructing geometric models and synthesizing textures to restore the image. When addressing significant image degradation, this approach, however, lacks semantic comprehension, leading to challenges such as incomplete semantics and indistinct content.

- **Deep learning-based image restoration:** In recent years, a range of emerging image restoration methodologies have been introduced and applied as a result of the rapid advancement of machine learning techniques, notably deep learning. In addition to excelling at image processing, convolutional neural networks (CNNs) can also capture and articulate intricate image characteristics. Image restoration techniques involving autoencoders and variational autoencoders are derived from CNNs. In addition, generative adversarial networks (GANs) have progressively gained significant attention as a method for learning generative models that align with data distributions. By leveraging deep learning methodologies, semantic insights can be extracted from images and absent semantic content can be predicted, thus overcoming the limitations of traditional techniques. Consequently, image restoration results are more likely to reflect objective reality.

1.5 Thesis Objectives

In this thesis, the primary goal is to construct predictive models tailored for multirate processes, ensuring the reliable and regular prediction of variables with slower rates. Diverse data sources, encompassing conventional sensors and image data, are used to achieve this objective. To enhance the applicability of these models, practical challenges, notably integration intervals, often overlooked in existing literature, time-delays, changing operating conditions and degradation in image data are carefully taken into account. The regular and timely predictions predicted by these models hold significant potential for a range of applications in process industry control and optimization.

1.6 Thesis Outline and Contributions

In Chapter 2, a background to the problems of parameter estimation, state estimation, and image feature extraction is provided. The thesis continues by explaining each of the specific contributions that will be presented in the following chapters. The rest of the thesis is organized as follows.

Chapter 3 focuses on the development of predictive models, both offline and online, for quality variables in multirate systems with switching operating modes. These models specifically account for slow-rate integrated measurements that experience time-varying time-delays as well as variable integration periods. The main contributions of Chapter 3 are as follows:

- Formulating the multirate process identification problem considering time-varying integration periods, time-varying time-delays, switching operating modes, simultaneously.
- Using a nonparametric form of distribution with no requirement of prior knowledge to model the integration periods, delays, and switching between different operating modes.
- Online prediction of the slow-measured variables.

Chapter 4 addresses the development of a robust model for predicting quality variables. This model aims to tackle multiple practical challenges along with those discussed in the previous chapter. The main contributions of this work, building upon the findings of Chapter 3, can be summarized as follows:

- Developing a global linear parameter varying (LPV) model for the processes with the slow-rate integrated output measurements.
- Improving robustness to outlying output observations using t-distribution.

- Considering the time-varying integration intervals, time-varying time-delays, and uncertainties of the unknown parameters, simultaneously, using the VB algorithm.

Chapter 5 presents the third contribution to predict flotation froth cumulative concentration using a computer vision model, taking into account contaminated images and outlier affected observations. A regression model with multiple inputs is considered to estimate the output. The EM algorithm is used to estimate the unknown parameters of the model. The principal contributions of this chapter are as follows.

- Restoration of the contaminated images with bright lighting spots using a modified spatial-based KF.
- Extraction of deep froth image features using CNN and transfer learning, and integrating features over the batch with unique attention weights assigned to successive frames.
- Development of a robust-to-outlier model for cumulative bitumen content prediction of each batch based on image features.

Chapter 6 presents the final contribution of the thesis in which we propose another computer vision model for predicting slow-measured variables at a fast rate. The main contributions of chapter 6 include:

- Development of an autoencoder-based algorithm to inpaint images with relatively large missing portions of images.
- Development of an ARX model using image data and slow-measured output to predict the output at a fast rate.
- Integration of the Rauch-Tung-Striebel (RTS) smoother into the KF framework, coupled with the utilization of the EM algorithm, thereby enabling estimation of unknown model parameters.

Chapter 7 summarizes the conclusions drawn from the developed models and algorithms. The possible future work is also outlined in the chapter.

Chapter 2

Background

2.1 Parameter Estimation

In this section, we delve into a detailed explanation of two highly recognized algorithms that are commonly employed to address the challenge of parameter estimation in scenarios where hidden or latent variables are present, or when dealing with missing data. These algorithms serve as powerful tools to navigate through complex modeling situations and offer effective solutions to estimate the unknown parameters involved.

2.1.1 Introduction to the EM Algorithm

The EM algorithm is a powerful computational technique widely used in statistical modeling and machine learning. It provides an elegant solution to problems involving incomplete or missing data, where the goal is to estimate the parameters of a probabilistic model. The EM algorithm was first introduced in [48], and has since then become a fundamental tool in various fields such as signal processing, natural language processing, and computer vision. At its core, the EM algorithm is an iterative optimization method that seeks to find the maximum likelihood estimates (MLE) or maximum *a posteriori* (MAP) estimates of the parameters in a statistical model. EM algorithm addresses scenarios where the presence of missing data prevents the direct application of traditional optimization techniques.

The key idea behind the EM algorithm is to iteratively alternate between two steps: the expectation step (E-step) and the maximization step (M-step). The E-step provides the expectation of the complete data likelihood based on the current model parameters and observed data. In the M-step, the algorithm updates the model parameters by maximizing the expected complete data likelihood obtained in the E-step. This step is often a straightforward optimization problem, as the complete data likelihood is usually easier to optimize than only the observed data likelihood. By iteratively repeating these two steps, the EM algorithm converges to a set of parameter estimates that maximize the observed data likelihood, in the presence of missing data.

The mathematical formulation of the EM algorithm can be expressed through the following steps. Assume a complete dataset that is comprised of an observed part D_{obs} and a hidden part D_{hid}.

E-step: In this step, expectation of the log-likelihood function of the complete data with respect to the conditional distribution of the hidden variables given the observed data and the parameters estimated at the previous iteration is derived.

$$Q(\Theta|\Theta^h) = \mathbb{E}_{D_{hid}|D_{obs},\Theta^h}\left[\log(p(D_{hid}, D_{obs}|\Theta))\right], \tag{2.1}$$

where Θ^h stands for the estimated parameters at the previous iteration.

M-step: In this step, the parameters are updated by maximizing the Q function with respect to each corresponding parameter. That is to find:

$$\Theta^{h+1} = arg \quad \max_{\Theta} Q(\Theta|\Theta^h). \tag{2.2}$$

Convergence: If the following relative change of parameter estimates is less than a prespecified small tolerance, then the algorithm is completed.

$$\frac{||\Theta^{h+1} - \Theta^h||^2}{||\Theta^h||^2} \leq \varepsilon \tag{2.3}$$

The performance of the EM algorithm can be influenced by the initial values of the model parameters. As the algorithm iterates and optimizes a likelihood function, different solutions may be reached with different initial parameter values. To mitigate the impact of initialization, the EM algorithm is commonly executed multiple times with various initial guesses. Researchers may employ strategies like random initialization, leveraging prior knowledge for informed initial guesses, or conducting a grid search across a range of values to enhance the likelihood of obtaining an optimal solution.

It is important to note that although the EM algorithm guarantees convergence under specific assumptions, it may still converge to a local maximum rather than the global maximum of the likelihood function [48, 49]. Consequently, careful selection of appropriate initial values and exploration of multiple starting points remain vital considerations to achieve reliable results with the EM algorithm.

2.1.2 Introduction to the VB Algorithm

The VB algorithm is a computational method that facilitates approximate Bayesian inference in complex probabilistic models. It offers an efficient and scalable solution for estimating the posterior distribution of latent variables and model parameters. VB is particularly useful when exact inference is intractable due to the complexity of the model or the size of the dataset [50]. The VB algorithm is rooted in Bayesian statistics, which provides a principled framework for modeling uncertainty and making probabilistic inferences. Bayesian inference aims to compute the posterior distribution, which represents the updated belief about the unknown variables given the observed data and prior knowledge. In many cases, obtaining the exact posterior distribution is analytically or computationally challenging. The VB algorithm addresses this challenge by approximating the true posterior with a simpler distribution that belongs to a parameterized family of distributions. This approximation is achieved by minimizing the Kullback-Leibler (KL) divergence between the true posterior and the approximating distribution. The VB algorithm offers several advantages, including its ability to handle large datasets and complex models, and its computational efficiency compared to more computationally demanding methods such as Markov chain Monte Carlo (MCMC). It has been successfully applied to a wide range of problems, including Bayesian inference, Bayesian neural networks, and latent variable modeling [51].

The VB algorithm transforms the inference problem into an optimization problem. In this algorithm, the parameters of the approximating distribution are iteratively updated by maximizing the evidence lower bound (ELBO). Maximizing the ELBO is equivalent to minimizing the KL divergence between the true posterior and the approximating distribution. The optimization process of the VB algorithm typically involves two steps [52]. In the first step, the algorithm computes the posterior distribution of the hidden variables given the current values of the model parameters. This step involves calculating the expected sufficient statistics of the hidden variables under the current approximating distribution. In the next step, the algorithm updates the posterior distribution of model parameters by maximizing the expected complete log-likelihood based on the current posterior distribution of the hidden variables. This step involves optimizing the ELBO with respect to the model parameters poterior distribution. By iteratively repeating these steps, the VB algorithm refines the approximating distribution and improves the lower bound on the log marginal likelihood. The algorithm continues until convergence, where the changes in the parameters and the ELBO become negligible.

Denote Δ as the model structure. In Bayesian estimation, the marginal likelihood of observed data given the model structure, $p(D_{obs}|\Delta)$, can be evaluated through integration over hidden variables and unknown parameters as:

$$p(D_{obs}|\Delta) = \iint p(D_{obs}, D_{hid}, \Theta|\Delta)\ dD_{hid}\ d\Theta. \tag{2.4}$$

Due to the complexity of the above integral and its high dimension, the log marginal likelihood can be expanded by introducing a free distribution $q(D_{obs}, D_{\Theta})$,

$$\log p(D_{obs}|\Delta) = \log \iint q(D_{obs}|\Delta) \frac{p(D_{obs}, D_{hid}, \Theta|\Delta)}{q(D_{obs|\Delta})} \, dD_{hid} \, d\Theta. \tag{2.5}$$

Then by applying Jensen's inequality,

$$\log p(D_{obs}|\Delta) \geq \iint q(D_{obs}|\Delta) \log \frac{p(D_{obs}, D_{hid}, \Theta|\Delta)}{q(D_{obs}|\Delta)} \, dD_{hid} \, d\Theta. \tag{2.6}$$

The difference between $\log p(D_{obs}|\Delta)$ and the lower bound is the KL divergence which is defined as follows.

$$KL(q||p) = \int q(D_{hid}, \Theta) \log \frac{q(D_{hid}, \Theta)}{p(D_{hid}, \Theta|D_{obs}, \Delta)} \, dD_{hid} \, d\Theta \tag{2.7}$$

Thus, (2.6) can be rewritten as

$$\log p(D_{obs}|\Delta) \geq \log \iint q(D_{obs}|\Delta) \frac{p(D_{obs}, D_{hid}, \Theta|\Delta)}{q(D_{obs}|\Delta)} \, dD_{hid} \, d\Theta \equiv F_{\Delta}(q(D_{hid})q(\Theta)) \tag{2.8}$$

Therefore, maximizing $\log(D_{obs}|\Delta)$ is shifted to maximizing $F_{\Delta}(q(D_{hid})q(\Theta))$ over $q(D_{hid})$ and $q(\Theta)$. This will lead to the following updating equations.

- **Updating posterior distribution of hidden variables**

$$q(D_{hid}) = \frac{\exp\left(\langle \log p(D_{obs}, D_{hid}, \Theta|\Delta) \rangle_{q(\Theta)}\right)}{\int \exp\left(\langle \log p(D_{obs}, D_{hid}, \Theta|\Delta) \rangle_{q(\Theta)}\right) \, dD_{hid}} \tag{2.9}$$

- **Updating posterior distribution of model parameters**

$$q(\Theta) = \frac{\exp\left(\langle \log p(D_{obs}, D_{hid}, \Theta|\Delta) \rangle_{q(D_{hid})}\right)}{\int \exp\left(\langle \log p(D_{obs}, D_{hid}, \Theta|\Delta) \rangle_{q(D_{hid})}\right) \, d\Theta} \tag{2.10}$$

where $\langle . \rangle_Y$ denotes the expectation with respect to Y. The aforementioned steps are iteratively updated until a convergence condition is met.

2.2 State Estimation

In this section, we present an introduction to the KF, which serves as a widely utilized tool for addressing linear state estimation problems. The KF is a powerful algorithm that allows for optimal estimation of the hidden or unobservable states in a linear dynamical system.

2.2.1 Kalman Filter

The KF is a powerful tool for estimating the state of a dynamic system based on noisy measurements. It provides an optimal recursive solution to the problem of state estimation in linear dynamic systems, combining information from both measurements and predictions [53]. The KF is based on Bayesian estimation principles and has since become a fundamental technique in various fields, including control systems, signal processing, robotics, and navigation.

To illustrate the KF, we consider a standard linear state-space model, which can be represented as follows:

$$x_k = Ax_{k-1} + Bu_k + w_k \tag{2.11}$$

$$y_k = Cx_k + v_k \tag{2.12}$$

where:

- x_k is the state vector at time k
- A is the state transition matrix
- B is the control input matrix
- u_k is the control input at time k
- w_k is the process noise with covariance Q
- y_k is the measurement at time k

- C is the measurement matrix
- v_k is the measurement noise with covariance R

The KF involves two steps: the prediction step and the update step. In the prediction step, we predict the state and its uncertainty based on the previous state estimate and current control input. The predicted state estimate $\hat{x}_k^-$ and the predicted error covariance P_k^- can be computed as follows:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{2.13}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{2.14}$$

In the update step, we incorporate the measurement to refine the state estimate based on the measurement's reliability and the predicted state estimate. The Kalman gain K_k is computed as:

$$K_k = P_{k|k-1} - C^T(CP_{k|k-1}C^T + R)^{-1} \tag{2.15}$$

Using the Kalman gain, the updated state estimate $\hat{x}_{k|k-1}$ and the updated error covariance $P_{k|k}$ are given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \tag{2.16}$$

$$P_{k|k} = (I - K_kC)P_{k|k-1} \tag{2.17}$$

These equations reflect the essential steps of the KF. The filter starts with an initial state estimate $\hat{x}_{0|0}$ and error covariance $P_{0|0}$, and then iterates through the prediction and update steps for each time step k.

The KF provides an optimal solution for estimating the state of a dynamic system based on noisy measurements. By combining predictions and measurements while considering uncertainties, it allows for accurate and adaptive state estimation. With its broad applicability and solid theoretical foundations, the KF serves as a fundamental tool in state estimation and data fusion applications.

2.3 Image Feature Extraction

Images are valuable sources of information with diverse applications. They offer insights in areas like medical diagnostics, agriculture, security, creative arts, and artificial intelligence. The visual nature of images makes them indispensable for understanding complex phenomena and making informed decisions. The extraction of image features is an integral part of computer vision and image processing, involving the capture of relevant and distinctive information from digital images. In order to interpret visual content efficiently, raw pixel data must be converted into a compact and meaningful representation. As a result of this extraction of key features, a variety of tasks are possible, including object recognition, image classification, image analysis, and image retrieval. For feature extraction, a number of techniques have been developed, such as edge detection, color histograms, texture feature extraction, and deep learning-based techniques. Computer vision applications across diverse industries and domains are empowered by image feature extraction, which identifies key patterns, shapes, textures, and other distinctive characteristics.

2.3.1 Common Image Feature Extractors

Image feature extractors can be categorized into different groups based on their underlying methodologies and applications. Following is a categorization of some commonly used image feature extractors:

1. **Edge detection:** Edge detection stands as a technique for extracting features, aiming to pinpoint object boundaries within an image. This procedure bears resemblance to the notion of enhancing image clarity. As noted, pixel intensity values tend to exhibit consistency within regions occupied by objects of the same kind. Substantial alterations occur solely along object perimeters, where distinct changes become evident. The following is an example of an edge

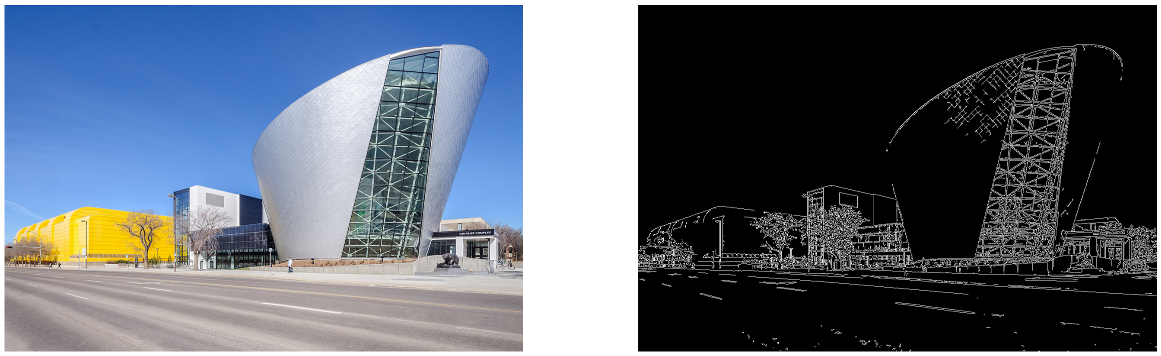

(a) Original image (b) Feature map after edge detection

Figure 2.1: Image edge detection

detector:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

where G_x and G_y are the gradients in the horizontal and vertical directions respectively, and I represents the image. The magnitude of the gradient can be obtained as:

$$\text{Gradient Magnitude} = \sqrt{G_x^2 + G_y^2}$$

This magnitude highlights regions with significant changes in intensity, corresponding to edges. By utilizing such equations, edge detection algorithms can effectively identify object boundaries based on these intensity fluctuations. Figure 2.1 shows the results of edge detection on a sample image using the above equations.

2. **Handcrafted feature extractors:**

- **Gray level co-occurrence matrix (GLCM):** It is a statistical method used in image processing for texture analysis. A wide range of texture features can be extracted from digital images using this technique. GLCM captures the spatial relationships between pixel intensities by analyzing the co-occurrence of gray levels within different pixel positions within an image.
- **Local binary pattern (LBP):** LBP determines the local texture patterns of an image, making it useful for identifying texture patterns and analyzing facial features.
- **Gabor Filters:** These filters are spatial frequency filters that analyze image regions at different orientations and scales to capture texture information.

3. **Frequency domain techniques:**
    - **Fast Fourier transform (FFT):** This method converts images into frequency domain representations, highlighting patterns that may not be easily seen in the spatial domain. Compression and denoising of images are examples of its use.
    - **Wavelet transform:** This technique is useful for detecting edges and analyzing textures, since it breaks down an image into different frequency components.

4. **Deep learning-based extractors:**
    - **Convolutional neural networks (CNNs):** These deep neural networks use convolutional layers to automatically learn features from images. In image processing and image analysis, CNNs are widely used for tasks such as image classification, object detection, and semantic segmentation. VGG, ResNet, and MobileNet are networks developed based on CNNs.

- **Siamese networks:** Siamese networks are used to compare and identify similarities between pairs of images, often in tasks such as similarity analysis.

5. **Transfer learning and pre-trained models:**

   - **Feature extractors from pre-trained CNNs:** Transfer learning involves using pre-trained CNNs as feature extractors for other tasks. For example, features from intermediate layers of networks like VGG or ResNet can be used for various tasks.

In this thesis, we use GLCM and a pre-trained CNN, VGG16, for the extraction of image features. Hence, we provide a detailed description of these two techniques.

2.3.2 Gray Level Co-occurrence Matrix

Texture analysis aims in finding a unique way of representing the underlying characteristics of textures and represent them in some simpler but unique form, so that they can be used for robust, accurate classification and segmentation of objects. The statistical texture analysis determines texture features by analyzing how intensities vary at each position relative to others in an image. There are three types of texture statistics: first order, second order, and higher order statistics. GLCM is used to extract second-order statistical texture features from images [54]. The GLCM operates on the principle of examining the spatial relationships between pixel intensity values in an image. GLCM calculates the frequency of pairs of pixels with a specific value and offset within a gray-level image, and it is a $G*G$ matrix where G is the number of gray levels in the image. The matrix element $p(i, j|\Delta x, \Delta y)$ is the relative frequency where i and j represent the intensity and both are separated by a pixel distance $\Delta x, \Delta y$. Image textural features such as energy, entropy, contrast, homogeneity, correlation, dissimilarity, inverse difference moment, and maximum probability can be computed using GLCM. The numerical features computed from GLCM can be used to represent, compare, and classify textures. Therefore, the GLCM is used in this paper to extract flotation froth image textural features. Once the GLCM is computed, a variety of statistical measures including entropy, contrast, energy and homogeneity can be derived from it. The following is a description of these features. These measures quantify the relationship between pixel intensity pairs and reveal essential textural information about the image [55].

$$\begin{aligned}
\text{Energy} &= \sum_{i=1}^{G}\sum_{j=1}^{G} p(i,j)^2 \\
\text{Contrast} &= \sum_{i=1}^{G}\sum_{j=1}^{G} (i-j)^2 \cdot p(i,j) \\
\text{Homogeneity} &= \sum_{i=1}^{G}\sum_{j=1}^{G} \frac{1}{1+|i-j|} \cdot p(i,j) \\
\text{Entropy} &= -\sum_{i=1}^{G}\sum_{j=1}^{G} p(i,j) \cdot \log(p(i,j)) \qquad (2.18)
\end{aligned}$$

Here G is the number of gray levels in the image. Also, $p(i,j)$ is the probability of the co-occurrence of pixel intensities i and j in the GLCM matrix.

2.3.3 Convolutional Neural Network

CNNs, inspired by the human visual cortex, have revolutionized computer vision tasks, especially in image processing. A CNN is a type of deep neural network mainly used to analyze visual images. In contrast to a feed-forward neural network, where each input element has its unique set of weights, CNN operates by sharing parameters across the input pixels. Their hierarchical feature learning and translation invariance enable them to extract meaningful patterns from raw data. A typical CNN architecture usually consists of convolutional layers, activation layer, pooling layer, and fully connected layer, as shown in Figure 2.2. The layers of the CNN are listed below, along with a description of their functions.

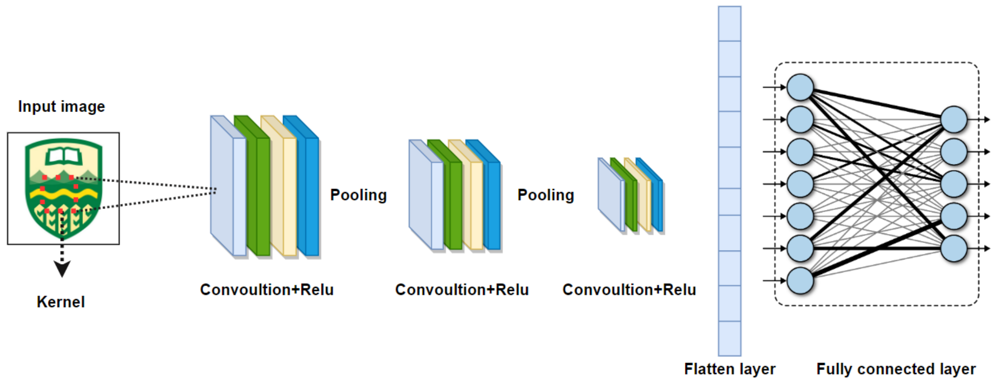


Figure 2.2: An architecture of a CNN

- **Convolutional layer:** In the convolution layers, trainable filters (kernels) are convoluted with the input image to extract specific features, starting from basic characteristics like edges and shapes in initial layers to more complex patterns in deeper ones. Proper adjustment of convolution kernel parameters is essential to match the input image's size and architecture of the network [56]. The sparse connectivity of neurons in the convolution layer allows them to share weights in response to overlapping receptive field regions, ensuring comprehensive coverage of the visual area.

- **Activation layer:** This layer introduces non-linearity to the network by applying an activation function to the output of the convolutional layer. The most commonly used activation function is the Rectified Linear Unit (ReLU), which replaces negative values with zero, allowing the network to learn complex relationships between features.

- **Pooling layer:** Also known as downsampling layer, it reduces the spatial dimensions of the feature maps while retaining their essential information [57]. Pooling helps decrease the computational complexity and makes the network more robust to variations in object position and size. Common pooling methods include max-pooling and average-pooling.

- **Fully connected (FC) layer:** After several convolutional and pooling layers, the feature maps are flattened and fed into one or more FC layers. The FC layers are responsible for generating the final output by combining the abstract features learned from the previous layers. In classification tasks, the FC layer produces the probabilities for each class.

It should be noticed that when only feature extraction is desired, the FC layer needs to be removed from the network.

In summary, CNNs offer several advantages that have made them popular, including:

1. The sharing of weights in convolutional layers substantially reduces the number of trainable parameters in the network. This makes CNNs computationally efficient, making them applicable to large-scale datasets.

2. During training, CNNs automatically learn relevant features from the input data and discover hierarchical representations for complex visual tasks, rather than relying on handcrafted features. This allows CNNs to understand complex relationships within the data.

3. Training CNNs on large datasets (*e.g.*, ImageNet) can be fine-tuned and used on other tasks, reducing the need for massive amounts of data.

4. CNNs can also be used just as feature extractors which can then be used for any other further applications.

Chapter 3

Predictive Model for Switching Multirate Processes with Varying Delays and Integrated Measurements [1]

3.1 Introduction

Multirate data, which can be seen as a special case of the missing data problem, is a common problem in practice. Modeling of multirate processes with time-varying time-delays has been addressed in numerous studies [24, 28, 29, 31, 58, 59]. The EM algorithm is commonly used for solving the parameter estimation problem in the presence of hidden variable(s) [60–62]. For instance, it was used to estimate the unknown parameters of a fast-rate FIR model in [28], output error (OE) model in [24, 58], state-space model in [59], and a slow-rate ARX model in [29]. While [24, 28, 58, 59] assumed that the measurement time-delays follow a uniform distribution, [29] claimed that the variations in time delays are not completely random, and there is a correlation between subsequent delays. Hence, a Markov chain was used to model the delays and the correlations between them. In [31], a fast-rate ARX model was proposed for multirate processes without relying on a prior distribution for modeling

[1]This chapter has been published as: **Y. Salehi** and B. Huang, "Offline and Online Parameter Learning for Switching Multirate Processes with Varying Delays and Integrated Measurements," in *IEEE Transactions on Industrial Electronic*, vol. 69, no. 7, pp. 7213-7222, July 2022.

the delays. In that work, the contribution of the different delays was determined by introducing a set of occurrence probabilities.

In addition, in certain processes such as in the cement industry and chemical processes, the samples of quality variables are often collected over a significant time interval and then transported to the lab for analysis. These integrated samples contain information of material over the time interval of collection, not a single instant. The time interval for collecting the material which might vary, is defined as the integration period. In most of the current multirate process identification practices, the integrated measurement problem is not considered. Although integrated measurement is common in practice, there exist limited theoretical studies on this subject [37, 39, 40, 63–65].

Furthermore, processes are typically operated in different operating modes [66], thus a single model may not perform well in capturing all the dynamics of the multimode processes. Data-driven process modeling techniques and detection of different operating conditions have been explored in the literature [67, 68]. For example, this problem has been considered in [69] where the data classification and parameter estimation problem are solved simultaneously. Also, adaptive methods are well-known approaches to update the parameters when the operating conditions of the systems are changing [70]. In this category, the recursive form of the EM algorithm [71] is a popular method which can perform efficiently in dealing with changing operating conditions. In [40], an ARX model was selected to predict the fast-rate values of the slow-rate sampled variables where the EM algorithm was used to estimate the unknown parameters. The main shortcomings of the work are, however, that the slow-rate measurements are assumed to be delay free, as well as having a fixed, known integration period. Moreover, the system is assumed to work in a single operating mode with no changes in the operating conditions.

The results in [37], [39], and [40] show the importance of considering integration periods in multirate modeling and control. Furthermore, in most of the existing literature for multirate systems, the time-delays are modeled by specific probability distribution functions. However, the delays may not follow the same distribution for the entire process. Hence, it is desirable to develop a nonparametric-based approach to deal with time-varying time-delays. Thus, the main contributions of this work are:

1. Formulating the multirate process modeling considering time-varying integration periods, time-varying time-delays, and switching operating points, simultaneously.
2. Using a nonparametric form of distribution with no requirement of prior knowledge to model the integration periods, delays, and switching of different operating modes.
3. Updating the model parameters in real time using a recursive EM algorithm.

3.2 Problem Statement

The following is the mathematical representation of a dual-rate augmented regression model in the presence of varying time-delays along with varying integration periods:

$$x_t = \phi_{(1)t}\theta^m_{(1)} + \phi_{(2)t}\theta^m_{(2)} + \cdots + \phi_{(R)t}\theta^m_{(R)} \tag{3.1}$$

$$y_{T_i} = \frac{1}{\ell_i}\sum_{s=0}^{\ell_i-1} x_{T_i-\lambda_i-s} + v_{T_i} \tag{3.2}$$

where $\phi_{(r)t} = \left[u_{(r)t}, u_{(r)t-1}, \cdots, u_{(r)t-n_r}\right]_{r=1,2,\ldots,R}$ (*i.e.* each input variable is augmented by its lagged values) and R is the number of input variables. Note that $\{u_{(r)t}\}_{t=1,\cdots,N_f}$ denotes the rth input variable which is available at every fast sampling time t_f, and x_t is the fast-rate output of interest. $\{\theta^m_{(r)}\}^{m=1,2,\cdots,M}_{r=1,2,\cdots,R} \in \mathbb{R}^{1\times(n_r+1)}$ denotes the parameter vector where n_r is the known order of the augmented regressor associated with the rth input variable, and M stands for the number of operating modes that is given. $\{y_{T_i}\}_{i=1,2,\cdots,N_s}$ represents the slow-rate integrated output that is available only at every $T_i \times t_f$. N_f and N_s are the number of the fast-rate input data (for each input variable) and slow-rate output data, respectively, where $N_f/N_s = \Delta$. Also, l_i and λ_i are, respectively, the varying integration periods and the varying time delays. The measurement noise term υ_{T_i} is an independent, identically distributed ($i.i.d$) Gaussian sequence with mean zero and an unknown variance σ^2. Figure 3.1 shows the system with delayed integrated measurements.

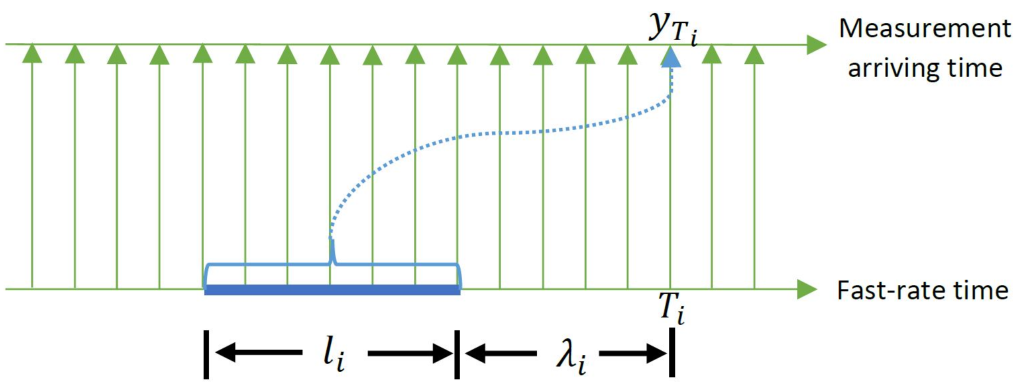


Figure 3.1: System with delayed integrated measurements

For simplicity in notation, the model representation in (3.1) and (3.2) can be rewritten as follows:

$$x_t = \tilde{\phi}_t \bar{\theta}^m \tag{3.3}$$

$$y_{T_i} = \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i} \bar{\theta}^m + \upsilon_{T_i} \tag{3.4}$$

where

$$\tilde{\phi}_t = \left[\phi_{(1)t}, \phi_{(2)t}, \cdots, \phi_{(R)t}\right] \tag{3.5}$$

$$\bar{\theta}^m = \left[\theta^m_{(1)}, \theta^m_{(2)}, \cdots, \theta^m_{(R)}\right]^T \tag{3.6}$$

$$\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i} = \frac{1}{\ell_i}\sum_{s=0}^{\ell_i-1} \tilde{\phi}_{T_i-\lambda_i-s} \tag{3.7}$$

Most of the existing modeling approaches for varying random delays rely on a specific form of the probabilistic distribution. Thus, the results may not be reliable when the delays follow a distribution that is different from the assumed distribution.

Despite the previous research such as [28, 58], and [29], the integer measurement delays arising from transportation time and manual processing do not have to follow a specific distribution. Therefore, in this work, the unknown varying time-delays are modeled by a nonparametric distribution which is more flexible. The upper possible bound for the delays, represented by J, is assumed to be known but it does not need to be precise. For each of the possible delays $j \in \{1, 2, \cdots, J\}$, an occurrence probability β_j is assigned that shows the probability for specific delay value j to occur. The set of the time-delay occurrence probabilities is indicated by $\beta = \{\beta_1, \beta_2, \cdots, \beta_J\}$ where $\sum\limits_{j=1}^{J} \beta_j = 1$ and $\beta_j \in [0, 1]$. For the extreme cases, $\beta_j = 1$ indicates that a delay equal to j is the only possible value. In contrast, $\beta_j = 0$ means that the delay equal to j does not occur.

Also, the integration periods are usually unknown and time varying. In this regard, the same proposed method for modeling the delays is used to model the varying integration periods. A set of occurrence probabilities $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$ is assigned to the integration periods $k \in \{1, 2, \cdots, K\}$ with the constraint that $\sum\limits_{k=1}^{K} \alpha_k = 1$. To comply with the physical possibility, it is assumed that $K + J \leq \Delta$ holds. However, if $K + J \leq \Delta$ increases, the number of parameters to estimate increases, making it more difficult to calculate parameters using the parameter estimation algorithms. Thus, if this upper bound is high, traditional methods based on prior distributions may be preferable since they require fewer parameters.

Furthermore, the systems usually switch between different modes. To solve the parameter estimation problem, the data need to be classified into different modes. We will classify the data using the same idea as for the delays and integration periods. Thus, a set of occurrence probabilities $\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$ is assigned to the operating points $m \in \{1, 2, \cdots, M\}$ with the constraint that $\sum\limits_{m=1}^{M} \gamma_m = 1$. The true upper bound for the number of modes, *i.e.* M, is assumed to be known, and the identity of each local model is denoted by I. In this setting, the observed variables

(D_{obs}), hidden variables (D_{hid}) and parameters to be estimated (Θ), are defined as

$$D_{obs} = \{Y, U\} = \{y_{T_1:T_{N_s}}, u_{(1)1:N_f}, \cdots, u_{(R)1:N_f}\} \tag{3.8}$$

$$D_{hid} = \{I, \Lambda, L\} = \{I_{1:N_s}, \lambda_{1:N_s}, \ell_{1:N_s}\} \tag{3.9}$$

$$\Theta = \{\bar{\theta}^m, \sigma^2, \gamma, \beta, \alpha\}_{m=1,2,\ldots,M} \tag{3.10}$$

A list of the key notation used in the problem formulation along with the corresponding descriptions are given in Table 3.1.

The goal is to develop an EM algorithm to solve the parameter estimation problem for multirate sampled processes subject to varying time-delays along with varying integration periods in the presence of multiple operating modes. Then, a recursive EM algorithm will be developed to update the parameters to predict the variables of interest in real time and at a fast rate.

Table 3.1: List of variables and their corresponding description

Notation	Description
x	Unmeasured fast-rate output
$u_{(r)}$	rth input variable
n_r	Order of dynamics in the rth input
y	Slow-rate integrated measurement
v	Measurement noise
λ	varying time-delay
β_j	Occurrence probability of $\lambda = j$
ℓ	Varying integration period
α_k	Occurrence probability of $\ell = k$
$\bar{\theta}^m$	Model parameter vector
$\tilde{\phi}$	Augmented regressor vector
I	Identity of each local model
γ_m	Occurrence probability of $I = m$

3.3 Offline Parameter Learning

In this section, a batch EM algorithm is derived for switching augmented regression models with the delayed and integrated output measurements to estimate the unknown parameters.

The EM algorithm is an iterative optimization algorithm to compute the MLE or MAP estimate of the unknown parameters in the presence of incomplete data [48]. In the EM algorithm, starting with an initial value of the unknown parameters, the E-step and M-step are repeated until the convergence of the parameters. Convergence of the EM algorithm has been proven in [48, 49]. Assume that a complete data set consists of an observed part D_{obs} and a hidden part D_{hid}. Then, the mathematical formulation of the EM algorithm can be expressed through the following steps.

3.3.1 Expectation Step

In this step, the expectation of the log-likelihood function of the whole data, which is known as the Q-function, including the missing data, with respect to the conditional distribution of the missing data given the observed data and the parameters estimated at the previous iteration is

$$Q(\Theta|\Theta^h) = \mathbb{E}_{D_{hid}|D_{obs},\Theta^h}\left[\log(p(D_{hid}, D_{obs}|\Theta))\right] \tag{3.11}$$

where $\mathbb{E}$ is the expectation operator and Θ^h is the estimated parameters at the previous iteration. According to the definition of the Q-function in (3.11) and the complete data set represented by (3.8)-(3.10), the following Q-function is obtained for the problem of interest, that is,

$$Q(\Theta|\Theta^h) = \mathbb{E}_{I,\Lambda,L|Y,U,\Theta^h}\left[\log(p(Y, U, I, \Lambda, L|\Theta))\right] \tag{3.12}$$

Based on the probability chain rule, the likelihood function of the complete data set can be decomposed as

$$p\left(Y,U,I,\Lambda,L|\Theta\right) = p\left(Y|U,I,\Lambda,L,\Theta\right)p\left(I|\Lambda,L,U,\Theta\right) \times p\left(\Lambda|L,U,\Theta\right)p\left(L|U,\Theta\right)p(U|\Theta) \quad (3.13)$$

Since the input values are known and deterministic, the last term in (3.13) is independent of Θ. Thus, this term can be denoted as $p(U|\Theta) \equiv C$ where C_1 is a constant with respect to Θ.

For the set of N_s observations, according to (3.4), the probability of the observations is independent of each other and takes the distribution of the measurement noise which can be expressed by the product of all output marginal distributions as

$$p\left(Y|U,I,\Lambda,L,\Theta\right) = \prod_{i=1}^{N_s} p(y_{T_i}|\bar{\phi}_{T_i-\lambda_i}^{(\ell_i)},\lambda_i,\ell_i,\Theta^{I_i}) \quad (3.14)$$

where every individual marginal distribution takes the form of

$$p\left(y_{T_i}|\bar{\phi}_{T_i-\lambda_i}^{(\ell_i)},\lambda_i,\ell_i,\Theta^{I_i}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(y_{T_i}-\bar{\phi}_{T_i-\lambda_i}^{(\ell_i)}\bar{\theta}^{I_i}\right)^2}{2\sigma^2}\right) \quad (3.15)$$

Furthermore, the modes, measurement time-delays, and integration periods can be considered jointly independent of each other and independent of the inputs as well owing to the fact that these variables do not affect each other, and their corresponding nonparametric distributions are given as

$$p\left(I|\Lambda,L,U,\Theta\right) = \prod_{i=1}^{N_s} p(I_i|\Theta) \quad (3.16)$$

$$p\left(\Lambda|L,U,\Theta\right) = \prod_{i=1}^{N_s} p(\lambda_i|\Theta) \quad (3.17)$$

$$p\left(L|U,\Theta\right) = \prod_{i=1}^{N_s} p(\ell_i|\Theta) \quad (3.18)$$

where γ_m, β_j, and α_k are, respectively, the probability that the operating mode m, the delay j, and the integration period k occur. So, for each slow-rate data point i we have

$$p(I_i = m) = \gamma_m \quad (3.19)$$

$$p(\lambda_i = j) = \beta_j \tag{3.20}$$

$$p(\ell_i = k) = \alpha_k \tag{3.21}$$

Substituting (3.14) and (3.16)-(3.18) into (3.12) and noting that the expectation operator can be moved into the summation yields

$$\begin{aligned} Q(\Theta|\Theta^h) =& \mathbb{E}_{I,\Lambda,L|Y,U,\Theta^h}\Bigg[\log(C)\prod_{i=1}^{N_s} p(y_{T_i}|\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, \Theta^{I_i})p(I_i)p(\lambda_i)p(\ell_i)\Bigg] \\ &= \sum_{i=1}^{N_s} \mathbb{E}_{I,\Lambda,L|Y,U,\Theta^h}\Bigg[\log(p(y_{T_i}|\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, \Theta^{I_i})) + \log(p(I_i)) \\ &+ \log(p(\lambda_i)) + \log(p(\ell_i))\Bigg] + \log(C) \end{aligned} \tag{3.22}$$

Substituting (3.15) and (3.19)-(3.21) into (3.22) gives

$$\begin{aligned} Q(\Theta|\Theta^h) =& \sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K}(p(I_i = m, \lambda_i = j, \ell_i = k)|y_{T_i}, \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, (\Theta^{I_i})^h) \\ &\times \Bigg[\log(p(y_{T_i})|\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, \lambda_i = j, \ell_i = k, \Theta^{I_i}, I_i = m)\Bigg] \\ &+ \sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K}(p(I_i = m, \lambda_i = j, \ell_i = k)|y_{T_i}, \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, (\Theta^{I_i})^h) \\ &\times \Bigg[\log p(I_i = m) + \log p(\lambda_i = j) + \log p(\ell_i = k)\Bigg] + \log(C) \end{aligned} \tag{3.23}$$

In order to further simplify the Q-function obtained in (3.23), the weight w_{imjk} denotes the joint posterior distribution of the hidden variables as

$$\begin{aligned} w_{imjk} &= p((I_i = m, \lambda_i = j, \ell_i = k)|y_{T_i}, \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}, \Theta^h) \\ &= \frac{p\left(y_{T_i}|\bar{\phi}^{(k)}_{T_i-j}, \lambda_i=j, \ell_i=k, (\Theta^{I_i=m})^h\right)\gamma^h_m\beta^h_j\alpha^h_k}{\sum\limits_{m=1}^{M}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} p\left(y_{T_i}|\bar{\phi}^{(k)}_{T_i-j}, \lambda_i=j, \ell_i=k, (\Theta^{I_i=m})^h\right)\gamma^h_m\beta^h_j\alpha^h_k} \end{aligned} \tag{3.24}$$

Thus, by considering the defined weight for the joint posterior distribution of the hidden variables in (3.24), as well as substituting (3.15) and (3.19)-(3.21) into (3.23), the Q-function becomes

$$\begin{aligned} Q(\Theta|\Theta^h) =& \sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}\Bigg[-\log\sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}[y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}^m]^2\Bigg] \\ &+ \sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}[\log\gamma_m + \log\beta_j + \log\alpha_k] + \log(C) \end{aligned} \tag{3.25}$$

3.3.2 Maximization Step

In this step, the parameters are updated by maximizing the Q-function with respect to each corresponding parameter, that is,

$$\Theta^{h+1} = \arg\max_{\Theta} Q(\Theta|\Theta^h) \tag{3.26}$$

Therefore, by taking the derivative of the Q-function with respect to the corresponding parameters and setting them equal to zero, that is,

$$\frac{\partial Q}{\partial \bar{\theta}^m} = \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}\Big[-\frac{1}{\sigma^2}(\bar{\phi}^{(k)}_{T_i-j})^T[y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}^m]\Big] = 0, \tag{3.27}$$

the update equation for the unknown model parameters is obtained as

$$(\bar{\theta}^m)^{h+1} = \Big[\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}(\bar{\phi}^{(k)}_{T_i-j})^T\bar{\phi}^{(k)}_{T_i-j}\Big]^{-1}\Big[\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}(\bar{\phi}^{(k)}_{T_i-j})^T y_{T_i}\Big] \tag{3.28}$$

Also, for the noise variance we have

$$\frac{\partial Q}{\partial \sigma^2} = \sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}\Big[-\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4}[y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}^m]^2\Big] = 0 \tag{3.29}$$

which gives the following update equation:

$$(\sigma^2)^{h+1} = \tfrac{1}{N_s}\Big[\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}[y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}^m]^2\Big] \tag{3.30}$$

To calculate the update equation of the occurrence probabilities, *i.e.* γ, β and α, we need to, respectively, include the constraints $\sum_{m=1}^{M}\gamma_m = 1$, $\sum_{j=1}^{J}\beta_j = 1$ and $\sum_{k=1}^{K}\alpha_k = 1$, and then solve a constrained optimization problem. Thus, we have

$$\frac{\partial}{\partial \gamma_m}\Big\{\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}[\log\gamma_m + \log\beta_j + \log\alpha_k] + L_\gamma\Big(\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K}\gamma_m - 1\Big)\Big\} = 0 \tag{3.31}$$

$$\frac{\partial}{\partial \beta_j}\Big\{\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}[\log\gamma_m + \log\beta_j + \log\alpha_k] + L_\beta\Big(\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{k=1}^{K}\beta_j - 1\Big)\Big\} = 0 \tag{3.32}$$

$$\frac{\partial}{\partial \alpha_k}\Bigg\{\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{imjk}[\log\gamma_m + \log\beta_j + \log\alpha_k] + L_\alpha\Bigg(\sum_{i=1}^{N_s}\sum_{m=1}^{M}\sum_{j=1}^{J}\alpha_k - 1\Bigg)\Bigg\} = 0 \quad (3.33)$$

where L_γ, L_β and L_α are the Lagrangian multipliers. Solving the above optimization problems provides the update equations for the modes, delays, and integration periods occurrence probabilities, respectively, as

$$\gamma_m^{h+1} = \frac{\sum\limits_{i=1}^{N_s}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{imjk}}{\sum\limits_{i=1}^{N_s}\sum\limits_{m=1}^{M}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{imjk}} \quad (3.34)$$

$$\beta_j^{h+1} = \frac{\sum\limits_{i=1}^{N_s}\sum\limits_{m=1}^{M}\sum\limits_{k=1}^{K} w_{imjk}}{\sum\limits_{m=1}^{M}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K}\sum\limits_{i=1}^{N_s} w_{imjk}} \quad (3.35)$$

$$\alpha_k^{h+1} = \frac{\sum\limits_{i=1}^{N_s}\sum\limits_{m=1}^{M}\sum\limits_{j=1}^{J} w_{imjk}}{\sum\limits_{i=1}^{N_s}\sum\limits_{m=1}^{M}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{imjk}} \quad (3.36)$$

It is important to note that in the proposed method, the above solutions are the probabilities of occurrences. If the point estimation of the actual mode, delay, and integration period is desired, using the idea of MAP, they can be determined by finding the largest w_{imjk} among all the possible values.

The proposed offline identification algorithm, which we call batch EM (B-EM), is:

1. Initialization: Set $h = 0$. Assign random values to $\Theta^h = \{\bar{\theta}^m, \sigma^2\,\beta, \alpha\}_{m=1,\cdots,M}$ and a small positive value to ε.

2. Compute w_{imjk}, $i = 1, 2, ...N_s$, $m = 1, 2, \cdots, M$, $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$ using (3.24).

3. Compute $(\bar{\theta}^m)^{h+1}, (\sigma^2)^{h+1}$ using (3.28) and (3.30), respectively.

4. Compute $\{\gamma_m^{h+1}\}_{m=1,2,\cdots,M}$, $\{\beta_j^{h+1}\}_{j=1,2,\cdots,J}$ and $\{\alpha_k^{h+1}\}_{k=1,2,\cdots,K}$ using (3.34), (3.35), and (3.36), respectively.

5. Set $h = h + 1$. If $\frac{||\Theta^{h+1}-\Theta^h||^2}{||\Theta^h||^2} \leq \varepsilon$, terminate the procedure; otherwise, go back to step 2.

3.4 Online Parameter Learning

The solution obtained so far addresses offline parameter estimation. It may be more desirable to predict the fast-rate values of variables with slow measurements in real time. In this regard, an online parameter learning algorithm is developed to update the parameters once a new measurement has arrived. Thus, changes in the parameters arising from changes in the operating conditions can be captured in real time. In this work, based on the idea of EM algorithm in [71] for latent data models, an iterative version of recursive EM, which makes better use of each data point [29], is obtained. The main idea of the proposed algorithm in [71] is to replace the E-step with a stochastic approximation step, while the M-step remains unchanged. Consider the following recursive Q-function:

$$\hat{Q}_{n+1}(\Theta) = \hat{Q}_n(\Theta) + \eta_{n+1}\left(\mathbb{E}_{D_{hid}^{n+1}}[\log f(X_{n+1}, \Theta)|Y_{n+1}] - \hat{Q}_n(\Theta)\right) \tag{3.37}$$

where Y_{n+1} is the new data point that has arrived at the $n + 1^{th}$ slow-rate sampling instant, X_{n+1} is the complete data and η is the step size. In this formulation, the expectation of the log distribution of the new data point is taken with respect to the hidden variables, given the updated parameters $\hat{\Theta}_n$ for the observation Y_{n+1}. The Q-function introduced in (3.37) is the lower bound of the log-likelihood, and maximizing it with respect to the parameters leads to parameter estimation. To use an iterative version of this recursive algorithm, we can set $\Theta_{n+1}^0 = \hat{\Theta}_n$ and iteratively maximize $\hat{Q}_{n+1}(\Theta)$ to improve parameter estimation at each iteration [29].

$$\begin{aligned}
\hat{Q}_{n+1}(\Theta|\Theta^{h}_{n+1}) =& \prod_{p=2}^{n+1}(1-\eta_p)\sum_{j=1}^{J}\sum_{k=1}^{K} w_{1jk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_1} - \bar{\phi}^{(k)}_{T_1-j}\bar{\theta}\Big)^2\Bigg] \\
&+\prod_{p=2}^{n+1}(1-\eta_p)\sum_{j=1}^{J}\sum_{k=1}^{K} w_{1jk}\Bigg[\log p(\lambda_1=j)+\log p(\ell_1=k)\Bigg] \\
&+\sum_{i=2}^{n}\sum_{p=i+1}^{n+1}(1-\eta_p)\eta_i\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}\Big)^2\Bigg] \\
&+\sum_{i=2}^{n}\sum_{p=i+1}^{n+1}(1-\eta_p)\eta_i\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Bigg[\log p(\lambda_i=j)+\log p(\ell_i=k)\Bigg] \\
&+\eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_{n+1}} - \bar{\phi}^{(k)}_{T_{n+1}-j}\bar{\theta}\Big)^2\Bigg] \\
&+\eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\Bigg[\log p(\lambda_{n+1}=j)+\log p(\ell_{n+1}=k)\Bigg] \qquad (3.38)
\end{aligned}$$

3.4.1 Expectation Step

According to the aforementioned solution, the recursive Q-function for the multirate processes with varying time-delays and integration periods is constructed as

$$\begin{aligned}
\hat{Q}_{n+1}(\Theta|\Theta^{h}_{n+1}) =&(1-\eta_{n+1})\hat{Q}_n(\Theta) \\
&+\eta_{n+1}\mathbb{E}_{D^{n+1}_{hid}}\big[\log p(y_{T_{n+1}},\bar{\phi}^{(\ell_{n+1})}_{T_{n+1}-\lambda_{n+1}},\Theta|y_{T_{n+1}},\bar{\phi}^{(\ell_{n+1})}_{T_{n+1}-\lambda_{n+1}})\big] \qquad (3.39)
\end{aligned}$$

By expanding the previous Q-function, the $n+1^{th}$ recursive Q-function can be written as (3.40).

$$
\begin{aligned}
\hat{Q}_{n+1}(\Theta|\Theta^h_{n+1}) = &\prod_{p=2}^{n+1}(1-\eta_p)\sum_{j=1}^{J}\sum_{k=1}^{K} w_{1jk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_1} - \bar{\phi}^{(k)}_{T_1-j}\bar{\theta}\Big)^2\Bigg] \\
&+ \prod_{p=2}^{n+1}(1-\eta_p)\sum_{j=1}^{J}\sum_{k=1}^{K} w_{1jk}\Bigg[\log p(\lambda_1 = j) + \log p(\ell_1 = k)\Bigg] \\
&+ \sum_{i=2}^{n}\sum_{p=i+1}^{n+1}(1-\eta_p)\eta_i\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_i} - \bar{\phi}^{(k)}_{T_i-j}\bar{\theta}\Big)^2\Bigg] \\
&+ \sum_{i=2}^{n}\sum_{p=i+1}^{n+1}(1-\eta_p)\eta_i\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Bigg[\log p(\lambda_i = j) + \log p(\ell_i = k)\Bigg] \\
&+ \eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\Bigg[-\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\Big(y_{T_{n+1}} - \bar{\phi}^{(k)}_{T_{n+1}-j}\bar{\theta}\Big)^2\Bigg] \\
&+ \eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\Bigg[\log p(\lambda_{n+1} = j) + \log p(\ell_{n+1} = k)\Bigg] \qquad (3.40)
\end{aligned}
$$

3.4.2 Maximization Step

The derivative of the recursive Q-function in (3.40), with respect to the model parameters, is set equal to zero. Then, the updated parameter estimates are calculated as

$$\bar{\theta}^{h+1}_{n+1} = (\bar{\theta}^{h+1}_{n+1,Den})^{-1}(\bar{\theta}^{h+1}_{n+1,Num}) \qquad (3.41)$$

where

$$(\bar{\theta}^{h+1}_{n+1,Num}) = (1-\eta_{n+1})(\bar{\theta}^{h}_{n+1,Num}) + \eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\ \ (\bar{\phi}^{(k)}_{T_{n+1}-j})^T y_{T_{n+1}}$$

$$(\bar{\theta}^{h+1}_{n+1,Den}) = (1-\eta_{n+1})(\bar{\theta}^{h}_{n+1,Den}) + \eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}(\bar{\phi}^{(k)}_{T_{n+1}-j})^T(\bar{\phi}^{(k)}_{T_{n+1}-j})$$

The subscripts Num and Den denote the numerator and denominator, respectively.

The variance of measurement noise is updated by taking the derivative of the Q-function in (3.40), with respect to the noise variance, and setting it equal to zero gives

$$(\sigma^2)^{h+1}_{n+1} = (1-\eta_{n+1})(\sigma^2)^{h}_{n+1} + \eta_{n+1}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{n+1jk}\big(y_{T_{n+1}} - \bar{\phi}^{(k)}_{T_{n+1}-j}\bar{\theta}^{h+1}_{n+1}\big) \qquad (3.42)$$

Also, the delay occurrence probabilities are updated using

$$(\beta_j)_{n+1}^{h+1} = [(\beta_j)_{n+1,Den}^{h+1}]^{-1}[(\beta_j)_{n+1,Num}^{h+1}] \tag{3.43}$$

where

$$(\beta_j)_{n+1,Num}^{h+1} = (1-\eta_{n+1})(\beta_j)_{n,Num} + \eta_{n+1}\sum_{i=1}^{N_s}\sum_{k=1}^{K} w_{ijk} \tag{3.44}$$

$$(\beta_j)_{n+1,Den}^{h+1} = (1-\eta_{n+1})(\beta_j)_{n,Den} + \eta_{n+1}\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk} \tag{3.45}$$

Moreover, the integration period occurrence probabilities are updated by

$$(\alpha_k)_{n+1}^{h+1} = [(\alpha_k)_{n+1,Den}^{h+1}]^{-1}[(\alpha_k)_{n+1,Num}^{h+1}] \tag{3.46}$$

where

$$(\alpha_k)_{n+1,Num}^{h+1} = (1-\eta_{n+1})(\alpha_k)_{n,Num} + \eta_{n+1}\sum_{i=1}^{N_s}\sum_{j=1}^{J} w_{ijk}$$

$$(\alpha_k)_{n+1,Den}^{h+1} = (1-\eta_{n+1})(\alpha_k)_{n,Den} + \eta_{n+1}\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}$$

We call this algorithm recursive EM (R-EM). The proposed R-EM is a computationally more efficient method since in the E-step, it does not require to calculate the Q-function using historical data. Also, in the M-step, only a simple update of the unknown parameters is required. In contrast to the B-EM, no prior information is required regarding the number of operating modes except that the modes should not switch before the minimum number of data required for parameter estimation have arrived.

3.5 Experimental Verification

The developed methods are tested on a lab-scale hybrid three-tank process. The experimental setup of the hybrid three-tank system is shown in Figure 3.2. The system consists of three tanks connected through six valves, *i.e.*, V1 - V4, V6, and V8. At the bottom of Tanks 1, 2, and 3, three outlet valves, *i.e.*, V5, V7 and V9, are provided. The two pumps feed the water into the two side tanks. In this experiment, all the valves are open. The inlet flow from left-hand-side pump and right-hand-side pump are considered as the inputs which are sampled at a fast rate. The level of the middle Tank 2 is the output sampled at a slow rate.

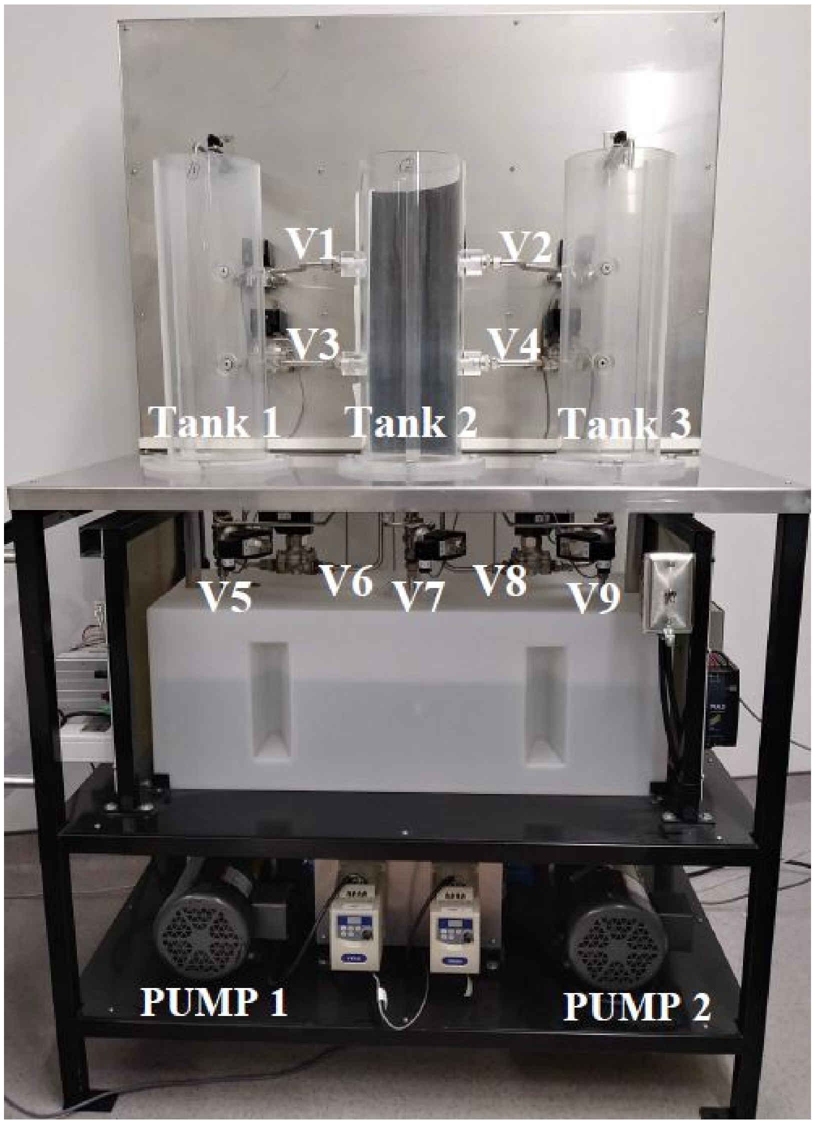


Figure 3.2: The hybrid three-tank system experimental setup

In the study, the fast-rate sampling interval is 20 s. The right-hand-side input signal is selected as a pseudorandom binary signal (PRBS) with levels [-0.3, 0.3] and with two different steady-state values 4.5 ℓ/min and 5.5 ℓ/min. Also, the left-hand-side input signal is selected as a PRBS with levels [-0.3 0.3] with a fixed steady-state value equal to 5.5 ℓ/min. Thus, due to the change in the steady-state value of the inlet flow of the right-hand-side input, the system switches between two different operating points. In total, there exist 3500 fast-rate input training data and 350 slow-rate output training data where 55.7% and 44.3% belong to the first and second mode, respectively. The output data are integrated and delayed with $\lambda_i \in \{1, 2, 3\}$ and $\ell_i \in \{3, 4, 5, 6\}$ where the generated delays and integration periods vary frequently. The true occurrence probabilities of the delays and integration periods are given in Table 3.3. Also, a zero-mean Gaussian noise with variance 0.001 is added to the delayed integrated outputs to generate the lab measurement noise. The collected fast-rate and slow-rate training data points are shown in Figure 3.3. As described earlier, ignoring integration periods and assuming constant delays for lab measurements is common in the literature, which contradicts reality. To see the significance of considering integration periods and delays in multirate systems, the results are compared with the case in which integration periods are neglected and the delays are considered as a constant. We call this conventional method the neglected integration fixed delay EM (NIFD-EM). In this study, to have a fair comparison, the most likely delay value, *i.e.*, $j = 1$, is used as the fixed delay for NIFD-EM. Moreover, both algorithms start from the same random initial values for the unknown model parameters. Like the persistence excitation order of the input signals, selection of the initial values for the unknown parameters can affect the convergence of the estimated parameters. For evaluation, the root mean square error (RMSE) as well as the coefficient of determination (R^2) values are used. The definition of these metrics are:

$$R^2 = 1 - \frac{\sum_{i=1}^{N_f} (x_i - \hat{x}_i)^2}{\sum_{i=1}^{N_f} (x_i - \bar{x})^2} \tag{3.47}$$

$$RMSE = \sqrt{\frac{1}{N_f}\sum_{i=1}^{N_f}(x_i - \hat{x}_i)^2} \tag{3.48}$$

where x_i is the true value, $\hat{x}_i$ is the estimated output value, and $\bar{x}$ denotes the mean value of the true output.

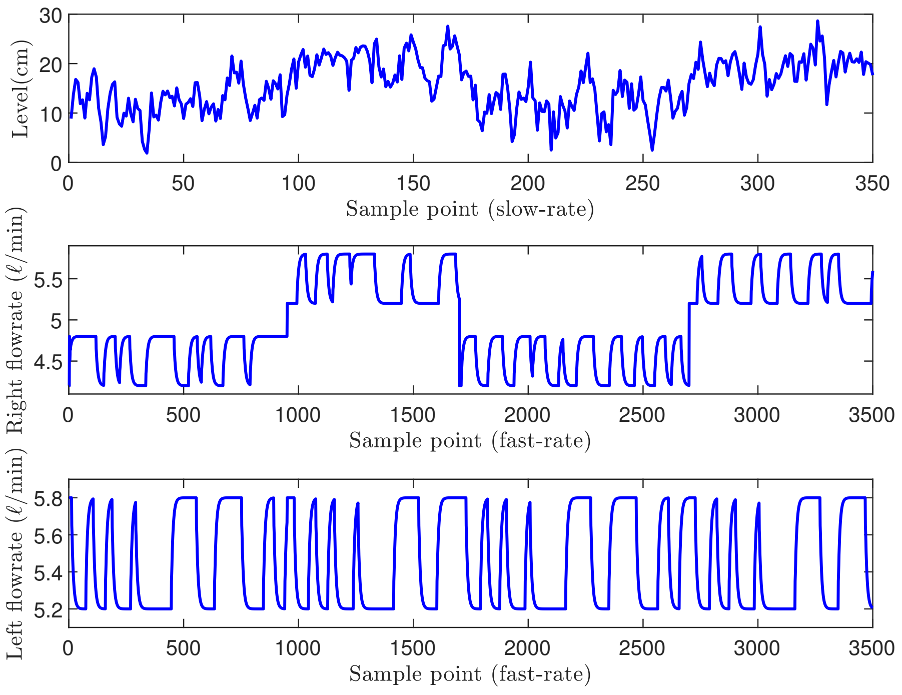


Figure 3.3: Training input and output data points

3.5.1 Parameter Learning Using the B-EM Algorithm

Applying the proposed B-EM algorithm and considering an augmented regression model with the lagged value of order 1 for each input, the validation results for the predicted fast-rate output samples are shown in Figure 3.4. In this study, while the true upper bounds of delays and integration periods are 3 and 6, respectively, we assume that the upper bounds are not exactly known. So, here $J = 4$ and $K = 7$ are arbitrarily chosen as the assumed upper bounds. From Figure 3.4 and the reported RMSE and R^2 values in Table 3.2, it can be seen that the model obtained can predict the unmeasured fast-rate outputs well. From Figure 3.4, it can also be seen that the output prediction in mode 2 is slightly less accurate than mode 1 due to fewer training data being available in mode 2. Comparing the RMSE and R^2 values with the NIFD-EM method shows the advantages of the proposed B-EM method, due to the consideration of the integration periods and varying time-delays. In this example, the R^2 is improved by 4.1%. This improvement will be clearer for larger values of integrations and delays. Nevertheless, the data with larger values of integrations and delays may degenerate the performance of both algorithms since more parameters need to be estimated. It is worth noting that even if mode switching occurs with higher frequencies, like that of delays and integration periods, the offline algorithm is still able to handle the problem efficiently. Furthermore, Table 3.3 shows the evolution of estimated occurrence probabilities of integration periods, delays, and modes. The initial values selected for parameters of each particular hidden variable are set equal to each other. Table 3.3 shows that, after around 80 iterations, all the parameters converge to the neighborhood of the true values. Also, we can see that the occurrence probabilities of the nonexisting integrations, *i.e.*, $k = \{1, 2, 7\}$, and the nonexisting delays, *i.e.*, $j = 4$, converge to zero. This means even if the lower and upper bounds are not exactly known, the algorithm can identify them, and the bounds can be adjusted accordingly. The adjusted bounds can be useful for the online algorithm. As the initially assumed bounds may not be close to their true values, the estimation results can be degraded considerably. However, using the proposed algorithm, the wrongly assumed bounds can be further adjusted and then the algorithm can be used again to improve the results.

Table 3.2: RMSE and R^2 of offline fast-rate output prediction

	Proposed B-EM	NIFD-EM
RMSE	0.079	0.093
R^2	0.894	0.853

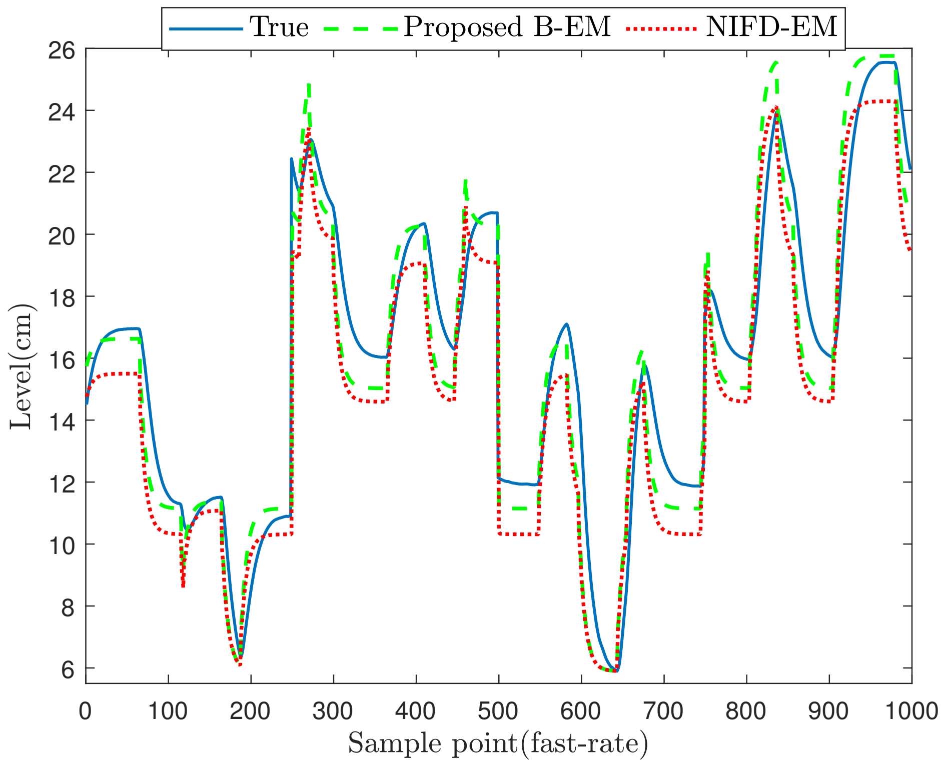


Figure 3.4: Offline fast-rate output prediction using the B-EM

Table 3.3: Estimated occurrence probabilities of the hidden variables

Hidden variables	Parameters	Iteration 1	20	40	60	80	True values
Integration periods	α_1	0.142	0.102	0.020	0.000	0.000	0
	α_2	0.142	0.129	0.052	0.000	0.000	0
	α_3	0.142	0.209	0.144	0.078	0.099	0.1
	α_4	0.142	0.233	0.278	0.201	0.192	0.2
	α_5	0.142	0.188	0.368	0.349	0.305	0.3
	α_6	0.142	0.093	0.115	0.364	0.403	0.4
	α_7	0.142	0.044	0.021	0.006	0.000	0
Delays	β_1	0.25	0.586	0.644	0.520	0.501	0.5
	β_2	0.25	0.257	0.235	0.309	0.311	0.3
	β_3	0.25	0.133	0.118	0.170	0.187	0.2
	β_4	0.25	0.023	0.002	0.000	0.000	0
Modes	γ_1	0.5	0.501	0.242	0.455	0.527	0.553
	γ_2	0.5	0.498	0.757	0.544	0.472	0.447

3.5.2 Real-time Output Prediction Using the R-EM Algorithm

In this evaluation, the data set from the previous section is used and there is no need to partition the data into training and validation parts. The algorithm starts with the assumption that 20 slow-rate output data points are already available, and the updating rate η_{n+1} is fixed and equal to 0.03. First, the derived B-EM is used for the available data and then the developed R-EM is applied to the subsequent data points. The predicted and true fast-rate outputs are shown in Figure 3.5. As it can be seen in Figure 3.5, the proposed R-EM has a reliable performance in real time prediction of outputs at a fast rate. Once the mode jumps to a new condition, it takes some time for the parameters to converge. Thus, the predicted results do not reach the true ones immediately after jumping to a new mode. As time passes and more observations arrive, the prediction is improved. This observation can be seen in Figure 3.5. The results for the RMSE and R^2 of the R-EM and NIFD-EM methods are given in Table 3.4 which shows the advantages of the R-EM algorithm developed in this study. Furthermore, the online estimates of the occurrence probabilities of integration periods and delays are shown in Figure 3.6, where it can be seen that the estimate is improved as more measurements become available. However, there is still bias in the estimation of these parameters, as the proposed algorithm is an approximated method, and the number of parameters to be estimated is relatively large.

Table 3.4: RMSE and R^2 of online fast-rate output prediction

	Proposed R-EM	NIFD-EM
RMSE	0.005	0.008
R^2	0.939	0.898

3.6 Conclusion

This study explored a data-driven modeling approach for fast-rate identification of multirate systems described by augmented regression models. The problem was formulated by taking into consideration important practical issues including switching

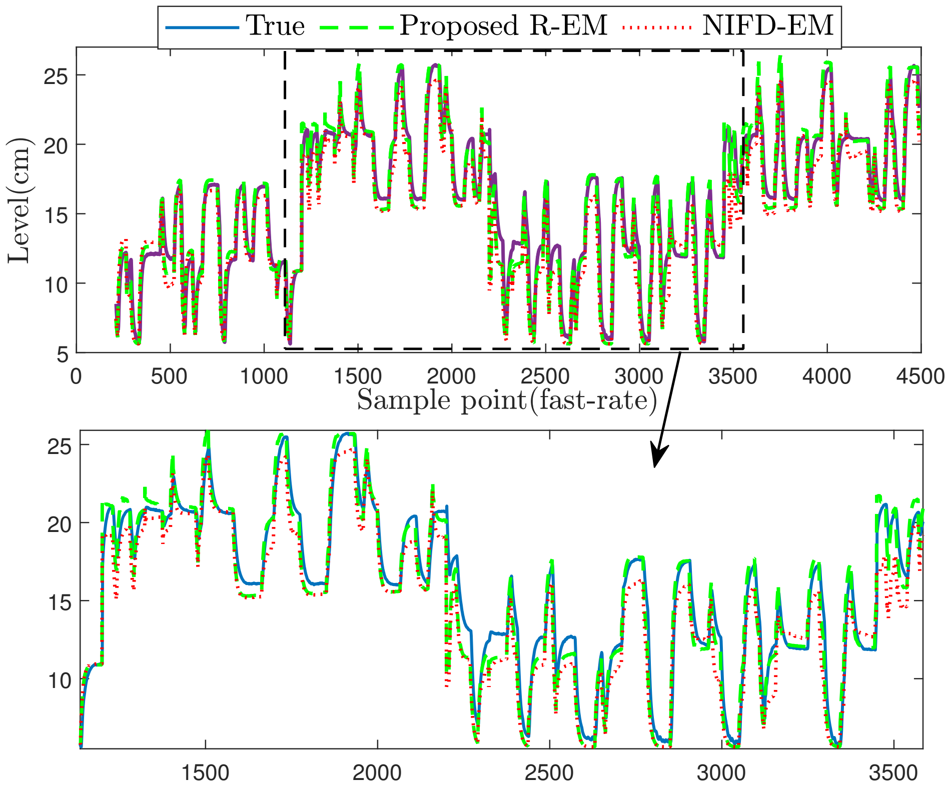


Figure 3.5: Online fast-rate output prediction using R-EM

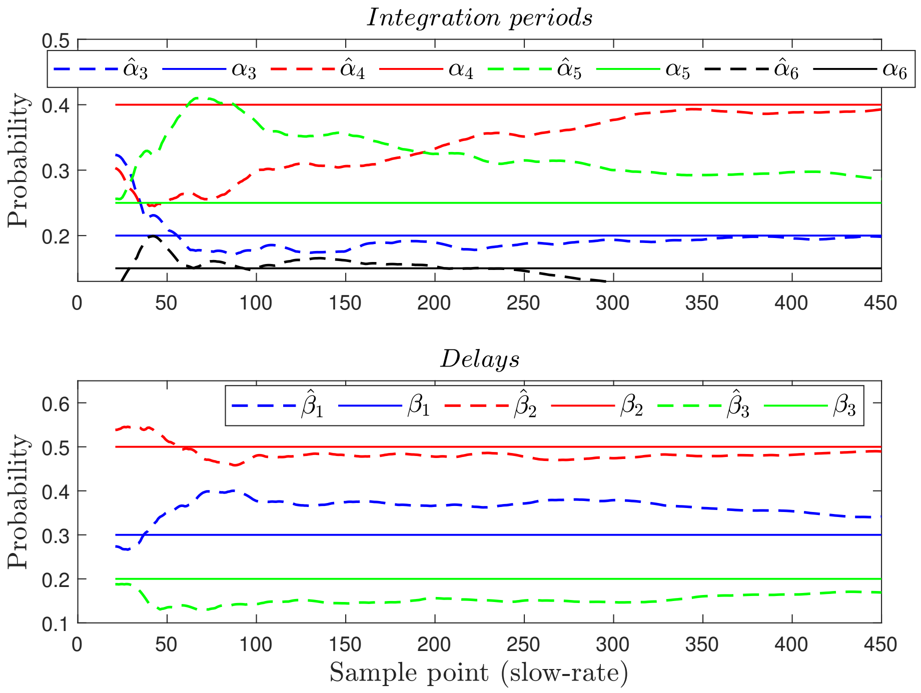


Figure 3.6: Online estimated occurrence probabilities of the hidden variables

operating modes, unknown varying measurement delays, and unknown varying integration periods. First, an offline algorithm was developed to classify the data and estimate the unknown parameters related to the local models and the hidden variables. The identified local models are capable of predicting the slow-rate sampled variables at fast rates. Furthermore, an online algorithm was proposed to recursively update the parameters and then predict the fast-rate outputs in real time. Without being confined by a specific probability distribution, a nonparametric-distribution-based method was proposed in modeling the hidden variables. The experimental results show that the R^2 value for the fast-rate output prediction improves by 4.1%. This improvement can be more significant for the processes with larger integration periods and delays.

Chapter 4

Robust Variational Bayesian-Based Predictive Model for LPV Processes with Delayed and Integrated Output Measurements [1]

4.1 Introduction

The lack of frequent measurements for the quality variables can deteriorate the performance of control and optimization and also may make them infeasible. Hence, a soft sensor model for predicting quality variables is essential for a variety of control applications. In addition to multirate data, a switching multi-model soft sensor [72] cannot capture the transitions between the modes. However, LPV modeling is a promising technique which consists of two general approaches [73], namely local approach and global approach. In the local approach, the model parameters are estimated based on an interpolation between multiple models. In contrast, in the global approach, the LPV model is identified directly using a global dataset sampled from the process working on a wide range of operating conditions. The former is preferred when the dataset is collected near the steady-state conditions, but the latter is suggested when the working conditions change frequently [74]. In this work, the global approach is

[1]This chapter has been published as **Y. Salehi** and B. Huang, "Robust Variational Bayesian-Based Soft Sensor Model for LPV Processes with Delayed and Integrated Output Measurements," in *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-10, 2022, Art no. 1006310, doi: 10.1109/TIM.2022.3200098.

considered.

The authors in [60] proposed an EM algorithm to develop a local-based LPV model in the presence of missing output measurements. Also, [61] used an EM algorithm to estimate the parameters of an FIR model for LPV multirate data. The model in [62] was an OE model, and it employed the GEM algorithm. Additionally, a robust version of the EM algorithm was proposed in [24] to address the outliers problem using Laplace distribution since it has adjustable-longer tails compared to Gaussian distribution. Also, [75] proposed a robust version of the EM algorithm to derive an LPV soft sensor model for predicting the slow-rate variable in multirate process modeling. t-distribution was chosen for the slow-rate output distribution to improve the robustness against outliers in the work. The chief limitation in the above works is that the developed EM-based algorithms only give the point estimate of the unknown parameters. However, the VB algorithm can also consider the uncertainties of the unknown parameters. In [31], a VB-based algorithm was proposed for multirate ARX models. However, it assumed that the process operates at only a single operating point. Then, [76] proposed a VB-based algorithm for the switching FIR models with missing output data. Further, [50] extended this work to LPV processes considering outliers and following a global approach.

The integration period problem in multirate process modeling was first addressed in [63] and later further considered in [40] using the EM algorithm. In [40], the improvement of the parameter estimation resulting from considering the integration periods for the slow-rate variables is confirmed. However, [40] addressed only LTI systems and delay-free measurements with a known and fixed integration period. Later, [77] proposed an EM algorithm for nonlinear multirate processes based on particle filter. In that work, the slow-rate integrated measurements were used as additional data, which are considered to be more accurate, to improve the estimation results. Recently, the parameter estimation of a regression model for the multirate processes was further developed by taking into account other critical practical issues including switching operating conditions, unknown and varying integration periods and delays [72]. The work also used a non-parametric approach to model the delays and integration periods, which is more flexible.

One shortcoming of [72], however, is that the transition between the operating conditions has been disregarded. Omitting the effect of outliers is another deficiency in that work. Also, as the EM algorithm is used in the work, the uncertainties of the model parameters have not been considered. Thus, the current chapter aims at developing a predictive model for quality variables considering all the above challenges simultaneously. The main contributions of this work, which extend that of [72], are listed as

1. Developing a global LPV predictive model for the processes with slow-rate integrated output measurements.
2. Improving robustness to outlying output observations using t-distribution.
3. Considering the time-varying integration periods, time-varying delays, and uncertainties of the unknown parameters, simultaneously, using the VB algorithm.

4.2 Problem Statement

An augmented regression model (each input variable contains its lagged values) for LPV multirate processes is given below:

$$x_t = \sum_{n=0}^{n_b} b_{(m)n}(s_t) u_{(m)t-n} \tag{4.1}$$

$$y_{T_i} = \frac{1}{\ell_i} \sum_{p=0}^{\ell_i - 1} x_{T_i - \lambda_i - p} + e_{T_i} \tag{4.2}$$

where

$$b_{(m)n}(s_t) = b_{(m)n,0} + \sum_{g=1}^{n_\beta} b_{(m)n,g} \eta_g(s_t), \qquad n = 1, 2, \cdots, n_b. \tag{4.3}$$

Table 4.1 contains the notations used in the problem formulation. The input variables (u_t) and the scheduling variable (s_t) are assumed to be available at each frequent (fast) sampling time instant t_f. The output measurements $\{y_{T_i}\}_{i=1,2,\cdots,N_s}$ are available only at every $T_i \times t_f$. N_f and N_s are the number of samples for each of the input variables and the slow-rate output, respectively. Also, $\eta_g(s_t)$ is a meromorphic function, such as polynomial function, of the scheduling variable.

To clarify the formulation in (4.1)-(4.3), consider the following illustrative example:

$$x_t = (1.1+2.3s_t)u_{(1)t} - (0.5+0.8s_t)u_{(1)t-1} + (1.5+3.4s_t)u_{(2)t} - (0.7+0.9s_t)u_{(2)t-1} \tag{4.4}$$

$$y_{T_i} = \frac{1}{\ell_i}\sum_{p=0}^{\ell_i-1} x_{T_i-\lambda_i-p} + e_{T_i}, \quad \ell_i \in \{1,2,3\},\ \lambda_i \in \{1,2\} \tag{4.5}$$

The augmented regression model in (4.4) contains two input variables, *i.e.* $u_{(1)}$ and $u_{(2)}$, with one lagged value for each of the inputs. Also, the regression model coefficients are linear combinations of the first order polynomial functions and of the scheduling variable.

For notation simplicity, (4.1) and (4.2) can be rewritten as follows.

$$x_t = \phi_t\theta \tag{4.6}$$

Table 4.1: List of the notations used in the formulation

Notation	Description
x	Unmeasured fast-rate output
$u_{(m)t}$	mth input variable at time instant t
n	Lag of historical data for the mth input
M	Total number of input variables
$b_{(m)n}$	LPV model coefficients for mth input variable
s	Scheduling variable
$\eta_g(s_t)$	Meromorphic function
y	Slow-rate integrated output
e	Measurement noise
λ	Delay
ℓ	integration period

$$y_{T_i} = \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta + e_{T_i} \tag{4.7}$$

where

$$\phi_t = [\phi_{(1)t}, \phi_{(2)t}, \cdots, \phi_{(M)t}] \tag{4.8}$$

$$\begin{aligned}\phi_{(m)t} =&[u_{(m)t}, \eta_1(s_t)u_{(m)t} \cdots \eta_{n_\beta}(s_t)u_{(m)t} \cdots u_{(m)t-1}, \eta_2(s_t)u_{(m)t-1} \cdots \eta_{n_\beta}(s_t)u_{(m)t-1},\\ &u_{(m)t-n_b} \cdots \eta_{n_\beta}(s_t)u_{(m)t-n_b}]\end{aligned} \tag{4.9}$$

$$\theta = [\theta_{(1)}, \theta_{(2)}, \cdots, \theta_{(M)}]^T \tag{4.10}$$

$$\theta_{(m)} = [b_{(m)1,0:n_\beta} \quad b_{(m)2,0:n_\beta} \quad \cdots \quad b_{(m)n_b,0:n_\beta}] \tag{4.11}$$

$$\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i} = \frac{1}{\ell_i}\sum_{p=0}^{\ell_i-1}\phi_{T_i-\lambda_i-p} \tag{4.12}$$

As an illustration, based on the rewritten formulation, the augmented regression vector and model parameters of the illustrative example introduced in (4.4)-(4.5) can be written as:

$$\phi_t = [u_{(1)t} \;\; s_t u_{(1)t} \;\; u_{(1)t-1} \;\; s_t u_{(1)t-1} \;\; \cdots u_{(2)t} \;\; s_t u_{(2)t} \;\; u_{(2)t-1} \;\; s_t u_{(2)t-1}] \tag{4.13}$$

$$\theta = [1.1 \quad 2.3 \quad -0.5 \quad -0.8 \quad 1.5 \quad 3.4 \quad -0.7 \quad -0.9]^T \tag{4.14}$$

Gaussian distribution is widely used in noise characterization. However, it is sensitive to the outliers owing to its short distribution tails. To make the modeling robust against outliers, the measurement noise e_{T_i} is assumed to follow a t-distribution with the mean zero, unknown precision δ, and degrees of freedom v. Accordingly, the slow-rate output variable follows a t-distribution, that is:

$$p(y_{T_i}|\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta, \delta, v) = \int \mathcal{N}(y_{T_i}|\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta, \frac{1}{\delta r_i})G(r_i|\frac{v}{2}, \frac{v}{2})dr_i \tag{4.15}$$

where $\mathcal{N}$ and G denote the Gaussian distribution and Gamma distribution, respectively.

The delays are often modeled based on some probabilistic distributions like uniform distribution [58]. Also, hidden Markov model is commonly used to characterize the correlations between the sequential delays [29]. However, in reality, the lab measurement delays may not necessarily follow such a specific form. Therefore, in the current work, the delays are modeled through a nonparametric-based method which is preferred for modeling arbitrary delay distributions [72]. It is assumed that an upper bound, denoted by J, for the delays is known. If it is unknown, a more conservative larger bound may be chosen. An occurrence probability β_j is assigned to each of the possible delay values $j \in \{1, 2, \cdots, J\}$ to indicate the probability of occurring a specific delay value j. The set of the time-delay occurrence probabilities is indicated by $\beta = \{\beta_1, \beta_2, \cdots, \beta_J\}$ where $\sum_{j=1}^{J} \beta_j = 1$ and $\beta_j \in [0, 1]$.

The same procedure is followed to model the integraion intervals. So, a set of occurrence probabilities $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$ is assigned to the different integration periods $k \in \{1, 2, \cdots, K\}$ subject to $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_j \in [0, 1]$.

4.3 Robust LPV Predictive Model

For the problem of interest, the observed variables (D_{obs}) and the hidden variables (D_{hid}) are defined as

$$D_{obs} = \{Y, U, S\} = \{y_{T_1:T_{N_s}}, u_{1:N_f}, s_{1:N_f}\} \tag{4.16}$$

$$D_{hid} = \{R, \Lambda, L\} = \{r_{1:N_s}, \lambda_{1:N_s}, \ell_{1:N_s}\} \tag{4.17}$$

Also, the unknown parameters are $\tilde{\Theta} = \{\Theta, v, \beta, \alpha\}$ where $\Theta = \{\theta, \delta\}$, $\beta = \{\beta_1, \beta_2, \ldots, \beta_J\}$, and $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$. The joint prior distribution for the unknown parameters can be written as follows.

$$p(\Theta) = p(\theta|b)p(\delta|c, d) \tag{4.18}$$

with

$$p(\theta|b) = \mathcal{N}(0, bI_{Rn_\beta(n_b+1)}),\ p(\delta|c, d) = G(c, d) \tag{4.19}$$

where b, c, and d are constant values, and $I_{R.n_\beta.(n_b+1)}$ is an identity matrix. To ensure that a close form solution can be obtained, the prior distributions are chosen as given in (4.19).

4.3.1 VB Algorithm

The VB algorithm can be used to estimate the posterior distributions of the unknown parameters and the hidden variables. This algorithm introduces the aforementioned posterior distributions by maximizing the log-likelihood function [78], that is:

$$\log p(D_{obs}) = \log \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \frac{p(D_{obs}, \Lambda, L, \Theta)}{q(L, \Lambda) q(\Theta)} dR d\Theta \tag{4.20}$$

However, solving this problem directly is difficult and may not be feasible. By applying Jensen's inequality, we have

$$\begin{aligned} \log p(D_{obs}) \geq & \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \log \frac{p(D_{obs}, \Lambda, L, \Theta)}{q(R, L, \Lambda) q(\Theta)} dR d\Theta \\ := & F[q(R, \Lambda, L), q(\Theta)] \end{aligned} \tag{4.21}$$

Thus, the problem turns to maximizing the lower bound denoted by $F[q(\Lambda, L), q(\Theta)]$ rather than the original log-likelihood function. To begin with, the lower bound in (4.21) can be further decomposed into the summation of several terms using the probability chain rule as

$$\begin{aligned} F[q(R, \Lambda, L), q(\Theta)] = & \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \log p(D_{obs} | R, \Lambda, L, \Theta) dR d\Theta \\ & + \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \log p(\Lambda | \beta) dR d\Theta \\ & + \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \log p(L | \alpha) dR d\Theta \\ & + \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) q(\Theta) \log p(R | v) dR d\Theta + \int q(\Theta) \log p(\Theta) d\Theta \\ & - \sum_{\Lambda} \sum_{L} \int q(R, \Lambda, L) \log q(R, \Lambda, L) dR d\Theta - \int q(\Theta) \log q(\Theta) d\Theta \end{aligned} \tag{4.22}$$

The VB algorithm iterates between two steps. In the first step, the lower bound is maximized with respect to the posterior distribution of hidden variables, *i.e.*

$q(R, \Lambda, L)$, assuming the posterior distribution over the unknown parameters, *i.e.* $q(\Theta)$, is fixed. In the second step, the lower bound is maximized with respect to $q(\Theta)$, by assuming that $q(R, \Lambda, L)$ is fixed. These iterations are executed until convergence [79]. Before the derivations, the likelihood terms are required to be determined, that is:

$$p(D_{obs}|R, \Lambda, L, \Theta) = \prod_{i=1}^{N_s} p(y_{T_i}|U, S, r_i, \lambda_i, \ell_i, \Theta) \tag{4.23}$$

$$p(R|v) = \prod_{i=1}^{N_s} p(r_i|v) \tag{4.24}$$

$$p(\Lambda|\beta) = \prod_{i=1}^{N_s} p(\lambda_i|\beta) \tag{4.25}$$

$$p(L|\alpha) = \prod_{i=1}^{N_s} p(\ell_i|\alpha) \tag{4.26}$$

Also, the probability density function (*pdf*) for the slow-rate output, noise variance, delay and integration period at each sampling instant i is given as (4.27)-(4.30), respectively.

$$p(y_{T_i}|U, S, r_i, \lambda_i, \ell_i, \Theta) = \frac{\sqrt{\delta r_i}}{\sqrt{2\pi}} \exp\Big(- \frac{\delta r_i}{2}[y_{T_i} - \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta]^2\Big) C_{U,S} \tag{4.27}$$

$$p(r_i|v) = \frac{(\frac{v}{2})^{\frac{v}{2}}(r_i)^{(\frac{v}{2}-1)}}{\Gamma(\frac{v}{2})} \exp\big(- \frac{v}{2} r_i\big) \tag{4.28}$$

$$p(\lambda_i = j) = \beta_j, \qquad j = 1, ..., J \tag{4.29}$$

$$p(\ell_i = k) = \alpha_k, \qquad k = 1, ..., K \tag{4.30}$$

where $C_{U,S}$ is the probability of data sampled from input and scheduling variables that is a constant term with respect to the unknown parameters and hidden variables.

Updating posterior distributions of the hidden variables

In this step, by treating $q(\Theta)$ as a fixed distribution, the lower bound in (4.21) can be written as

$$
\begin{aligned}
F[q(R,\Lambda,L),q(\Theta)] = & \sum_{\Lambda}\sum_{L}\int q(R,\Lambda,L)\langle \log p(D_{obs}|R,\Lambda,L,\Theta)\rangle_{q(\Theta)} \\
& + \sum_{\Lambda}\sum_{L}\int q(R,\Lambda,L)\log p(R|v)dR \\
& + \sum_{\Lambda}\sum_{L}\int q(R,\Lambda,L)\log p(L|\alpha)dR \\
& + \sum_{\Lambda}\sum_{L}\int q(R,\Lambda,L)\log p(\Lambda|\beta)dR \\
& - \sum_{\Lambda}\sum_{L}\int q(R,\Lambda,L)\log q(R,\Lambda,L)dR + C_{\Lambda,L,R} \qquad (4.31)
\end{aligned}
$$

where $\langle . \rangle_{q(\Theta)}$ indicates the expectation operator over $q(\Theta)$ and $C_{\Lambda,L,R}$ represents the terms that are independent of the hidden variables which is treated as a constant value. By maximizing $F[q(R,\Lambda,L),q(\Theta)]$ in (4.21) with respect to $q(R,\Lambda,L)$ such that $\sum\limits_{\Lambda}\sum\limits_{L}\int q(R,\Lambda,L)dR = 1$, we obtain

$$
q(R,\Lambda,L) = \frac{p(R|v)p(\Lambda|\beta)p(L|\alpha)\exp\big(B\big)}{\sum\limits_{\Lambda}\sum\limits_{L}\int p(R|v)p(\Lambda|\beta)p(L|\alpha)\exp\big(B\big)dR} \qquad (4.32)
$$

where $B = \langle \log p(D_{obs}|R,\Lambda,L,\Theta)\rangle_{q(\Theta)}$. Next, the posterior distribution of each hidden variable is derived.

- **Variational posterior of R given** Λ, L: By integrating out R from the joint density $q(R,\Lambda,L)$, the joint density of Λ and L is obtained as

$$
q(\Lambda,L) = \frac{\int p(R|v)p(\Lambda|\beta)p(L|\alpha)\exp\big(B\big)dR}{\sum\limits_{\Lambda}\sum\limits_{L}\int p(R|v)p(\Lambda|\beta)p(L|\alpha)\exp\big(B\big)dR} \qquad (4.33)
$$

Therefore, the conditional density $q(R|\Lambda,L)$ is calculated as

$$
q(R|\Lambda,L) = \frac{q(R,\Lambda,L)}{q(\Lambda,L)} = \frac{p(R|v)\exp\big(B\big)}{\int p(R|v)\exp\big(B\big)dR} = \prod_{i=1}^{N_s}\frac{p(r_i|v)\exp\big(B_i\big)}{\int p(r_i|v)\exp\big(B_i\big)dr_i} \qquad (4.34)
$$

To simplify (4.34), we need to take expectation of the log-likelihood in (4.27) at time instant i, resulting in:

$$
B_i = -\log\sqrt{2\pi} + \frac{1}{2}\tilde{\delta} + \log\sqrt{r_i} - \frac{\bar{\delta}r_i}{2}g_{ijk} + \log C_{U,S} \qquad (4.35)
$$

where

$$\tilde{\delta} = \langle \log \delta \rangle_{q(\delta)} \tag{4.36}$$

$$\bar{\delta} = \langle \delta \rangle_{q(\delta)} \tag{4.37}$$

$$g_{ijk} = {y_{T_i}}^2 - 2y_{T_i}\bar{\phi}^{(k)}_{T_i-j}\bar{\theta} + \bar{\phi}^{(k)}_{T_i-j}\langle \theta\theta^T \rangle_{q(\Theta)}(\bar{\phi}^{(k)}_{T_i-j})^T \tag{4.38}$$

$$\bar{\theta} = \langle \theta \rangle_{q(\theta)} \tag{4.39}$$

$$\begin{aligned} q(r_i|\lambda_i = j, \ell_i = k) &= \frac{p(r_i|v)\exp\big(B_i\big)}{\int p(r_i|v)\exp\big(B_i\big)dr_i} = \frac{\exp\bigg(- r_i\frac{v+\bar{\delta}g_{ijk}}{2}\bigg)(r_i)^{(\frac{v+1}{2}-1)}}{\int \exp\bigg(- r_i\frac{v+\bar{\delta}g_{ijk}}{2}\bigg)(r_i)^{(\frac{v+1}{2}-1)}dr_i} \\ &= \frac{1}{\Gamma(\frac{v+1}{2})}\bigg(\frac{v+\bar{\delta}g_{ijk}}{2}\bigg)^{\frac{v+1}{2}}(r_i)^{\frac{v+1}{2}-1}\exp\bigg(- r_i\frac{v+\bar{\delta}g_{ijk}}{2}\bigg) \\ &\sim G\bigg(\frac{v+1}{2}, \frac{v+\bar{\delta}g_{ijk}}{2}\bigg) \end{aligned} \tag{4.40}$$

Therefore, the variational posterior of r_i given $\lambda_i = j$ and $\ell_i = k$ is derived as (4.40). Based on (4.40) and according to the property of Gamma distribution, we can obtain the expectation of the conditional posterior distribution over r_i and $\log r_i$, which yields

$$\bar{r}_{ijk} = \langle r_i \rangle_{q(r_i|\lambda_i=j,\ell_i=k)} = \frac{v+1}{v+\bar{\delta}g_{ijk}} \tag{4.41}$$

$$\tilde{r}_{ijk} = \langle \log r_{ijk} \rangle_{q(r_i|\lambda_i=j,\ell_i=k)} = \Psi\bigg(\frac{v+1}{2}\bigg) - \log\bigg(\frac{v+\bar{\delta}g_{ijk}}{2}\bigg) \tag{4.42}$$

where Ψ is the derivative of the logarithm of the gamma function, *i.e.*, $\Psi(v) = \frac{\partial\Gamma(v)}{\partial v}\frac{1}{\Gamma(v)}$.

- **Variational joint posterior of the delay and integration period:**

Now, based on (4.33), the variational joint posterior of $\{\lambda_i = j, \ell_i = k\}$ for $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$, which we indicate by w_{ijk}, is obtained as

$$w_{ijk} := q(\lambda_i = j, \ell_i = k) = \frac{\int p(r_i|v)p(\lambda_i = j)p(\ell_i = k)\exp\big(B_i\big)dr_i}{\sum\limits_{k=1}^{K}\sum\limits_{j=1}^{J}\int p(r_i|v)p(\lambda_i = j)p(\ell_i = k)\exp\big(B_i\big)dr_i}$$
$$= \frac{\int p(r_i|v)\exp\big(B_i\big)dr_i\beta_j\alpha_k}{\sum\limits_{k=1}^{K}\sum\limits_{j=1}^{J}\int p(r_i|v)\exp\big(B_i\big)dr_i\beta_j\alpha_k} \tag{4.43}$$

where for each time instant i

$$\int p(r_i|v)\exp\big(B_i\big)dr_i = C_{r_i}\int\bigg(-r_i\frac{v+\bar{\delta}g_{ijk}}{2}\bigg)(r_i)^{\frac{v-1}{2}}dr_i = C_{r_i}\bigg(\frac{v+\bar{\delta}g_{ijk}}{2}\bigg)^{-\frac{v+1}{2}} \tag{4.44}$$

where C_{r_i} contains the constant terms.

Updating posterior distribution of the model parameters

So far, the posterior distributions of the hidden variables, *i.e.* noise variance, delay and integration period have been obtained. The objective of this step is to maximize the lower bound in (4.21) with respect to $q(\Theta)$ by fixing $q(R, \Lambda, L)$ and finally obtain the posterior distributions of the unknown parameters. Then, we get

$$F[q(R, \Lambda, L), q(\Theta)] = \int q(\Theta)\langle\log p(D_{obs}|R, \Lambda, L, \Theta)\rangle_{q(R,\Lambda,L)}d\Theta$$
$$+\int q(\Theta)\log p(\Theta)d\Theta - \int q(\Theta)\log q(\Theta)d\Theta + C_\Theta \tag{4.45}$$

where C_Θ includes all the terms that are constant with respect to Θ. Here, for notation simplicity, we define

$$D = \langle\log p(D_{obs}|R, \Lambda, L, \Theta)\rangle_{q(R,\Lambda,L)} \tag{4.46}$$

By maximizing $F[q(R, \Lambda, L), q(\Theta)]$ with respect to $q(\Theta)$ such that $\int q(\Theta)d\Theta = 1$, we have

$$q(\Theta) = \frac{p(\Theta)\exp\big(D\big)}{\int p(\Theta)\exp\big(D\big)d\Theta} \tag{4.47}$$

where D can be computed as

$$D = \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Big[-\log\sqrt{2\pi} + \log\sqrt{\delta} + \frac{1}{2}\tilde{r}_{ijk} - \frac{\delta\bar{r}_{ijk}}{2}(y_{T_i} - \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta)^2 + \log C_{U,S}\Big] \tag{4.48}$$

To obtain $q(\Theta)$, we take the derivative of $F[q(R,\Lambda,L), q(\Theta)]$ considering $\int q(\Theta)d\Theta = 1$, and then get

$$\begin{aligned} q(\theta) &= \frac{1}{C_\theta} p(\theta|b)\exp\big(\langle D\rangle_{q(\delta)}\big) \\ &= \frac{1}{C_\theta} p(\theta|b)\exp\Big(\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}(-\frac{\bar{\delta}\bar{r}_{ijk}}{2})(y_{T_i} - \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta)^2\Big) \\ &= \frac{1}{C_\theta}\exp\Big(-\frac{1}{2b}\theta^T I\theta\Big)\exp\Big(\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}(-\frac{\bar{\delta}\bar{r}_{ijk}}{2})(y_{T_i} - \bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\theta)^2\Big) \\ &= \frac{1}{C_\theta}\exp\Big(-\frac{1}{2}\theta^T\big[b^{-1}I + \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\bar{\delta}\bar{r}_{ijk}\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\big]\theta \\ &+ \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\bar{\delta}\bar{r}_{ijk}y_{T_i}\theta^T(\bar{\phi}^{(k)}_{T_i-j})^T\Big) \end{aligned} \tag{4.49}$$

This expression indicates that $q(\theta)$ is amount to a Gaussian distribution with the following mean and variance, respectively.

$$\bar{\theta} = var(\theta)\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\bar{\delta}\bar{r}_{ijk}y_{T_i}(\bar{\phi}^{(k)}_{T_i-j})^T \tag{4.50}$$

$$var(\theta) = \Big[b^{-1}I + \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\bar{\delta}\bar{r}_{ijk}(\bar{\phi}^{(k)}_{T_i-j})^T\bar{\phi}^{(\ell_i)}_{T_i-\lambda_i}\Big]^{-1} \tag{4.51}$$

Therefore, we have

$$\langle\theta\theta^T\rangle_{q(\theta)} = var(\theta) + \bar{\theta}\bar{\theta}^T \tag{4.52}$$

The same procedure is followed to maximize $F[q(R,\Lambda,L), q(\Theta)]$ with respect to $q(\delta)$ subject to $\int q(\delta)d\delta = 1$. Solving this maximization problem yields the following expression for $q(\delta)$.

$$\begin{aligned}
q(\delta) &= \frac{p(\delta|c,d)}{C_\delta} \exp\big(\langle D\rangle_{q(\Theta)}\big) \\
&= \frac{p(\delta|c,d)}{C_\delta} \exp\Big(\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}[\frac{1}{2}\log\delta - \frac{\delta\bar{r}_{ijk}g_{ijk}}{2}]\Big) \\
&= \frac{d^c\delta^{c-1}\exp\big(-d\delta\big)}{C_\delta\Gamma(c)}\delta^{\frac{N_s}{2}} \exp\Big(\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}(-\frac{\delta\bar{r}_{ijk}g_{ijk}}{2})\Big) \\
&= \frac{\delta^{c+\frac{1}{2}N_s-1}}{C_\delta} \exp\Big(-[d+\frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\bar{r}_{ijk}g_{ijk}]\delta\Big) \qquad (4.53)
\end{aligned}$$

The obtained expression for $q(\delta)$ indicates a Gamma density function with the parameters given below:

$$\bar{\delta} = \frac{2c+N_s}{2d+\sum\limits_{i=1}^{N_s}\sum\limits_{j=}^{J}\sum\limits_{k=1}^{K} w_{ijk}\bar{r}_{ijk}g_{ijk}} \qquad (4.54)$$

$$\tilde{\delta} = \psi(\frac{2c+N_s}{2}) - \log(\frac{2d+\sum\limits_{i=1}^{N_s}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{ijk}\bar{r}_{ijk}g_{ijk}}{2}) \qquad (4.55)$$

Furthermore, to update the hidden variables-related parameters, *i.e.* $\{v,\beta,\alpha\}$, the lower bound in (4.45) can be expressed as given in (4.56).

$$\begin{aligned}
&F[q(R,\Lambda,L),q(\Theta)] \\
&= \langle\log p(R|v)\rangle_{q(R|\Lambda,L)q(\Lambda,L)} + \langle\log p(L|\alpha)\rangle_{q(L|\Lambda)q(\Lambda)} + \langle\log p(\Lambda|\beta)\rangle_{q(\Lambda)} + C_{v,\beta,\alpha} \\
&= \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\Big(-\log\Gamma(\frac{v}{2}) + \frac{v}{2}\log(\frac{v}{2}) + (\frac{v}{2}-1)\tilde{r}_{ijk} - \frac{v}{2}\bar{r}_{ijk}\Big) \\
&+ \sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}(\log\alpha_k + \log\beta_j) + C_{v,\beta,\alpha} \qquad (4.56)
\end{aligned}$$

By taking the derivative of (4.56) with respect to β and α subject to constraints $\sum\limits_{j=1}^{J}\beta_j = 1$ and $\sum\limits_{k=1}^{K}\alpha_k = 1$, respectively, we have

$$\frac{\partial}{\partial\beta_j}\Big\{\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\log\beta_j + L_\beta\Big(\sum_{j=1}^{J}\beta_j - 1\Big)\Big\} = 0 \qquad (4.57)$$

$$\frac{\partial}{\partial \alpha_k}\left\{\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\log\alpha_k + L_\alpha\left(\sum_{k=1}^{K}\alpha_k - 1\right)\right\} = 0 \tag{4.58}$$

Then, solving the above optimization problems, in which L_α and L_β are the Lagrangian multipliers, gives the point estimate of α_k and β_j as follows.

$$\beta_j = \frac{\sum\limits_{i=1}^{N_s}\sum\limits_{k=1}^{K} w_{ijk}}{\sum\limits_{i=1}^{N_s}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{ijk}} \tag{4.59}$$

$$\alpha_k = \frac{\sum\limits_{i=1}^{N_s}\sum\limits_{j=1}^{J} w_{ijk}}{\sum\limits_{i=1}^{N_s}\sum\limits_{j=1}^{J}\sum\limits_{k=1}^{K} w_{ijk}} \tag{4.60}$$

Solving the derivative of the lower bound in (4.56) with respect to v yields

$$\sum_{i=1}^{N_s}\sum_{j=1}^{J}\sum_{k=1}^{K} w_{ijk}\left[-\psi(\frac{v}{2}) + \log(\frac{v}{2}) + 1 + \tilde{r}_{ijk} - \bar{r}_{ijk}\right] = 0 \tag{4.61}$$

Since (4.61) is a nonlinear equation, it needs to be solved numerically. We have used the MATLAB nonlinear solver *fslove* for this purpose. v then is updated.

The procedure for the proposed robust VB-based algorithm is summarized as given in Algorithm 1.

Algorithm 1 Robust VB-based LPV predictive model

```
1: Set $h = 0$. Assign random values to $\Theta^h$, and random positive values to $v, b, c, d$.
2: Update $\bar{r}_{ijk}$ and $\tilde{r}_{ijk}$, using (4.41) and (4.42), respectively.
3: Update the joint posterior distribution of the delays and integration periods, $q(\lambda_i = j, \ell_i = k)$, using (4.43).
4: Update $\bar{\theta}$, $var(\theta)$ and $\langle\theta\theta^T\rangle_{q(\Theta)}$ using (4.50)-(4.52), respectively.
5: Update $\bar{\delta}$ and $\tilde{\delta}$ using (4.54) and (4.55), respectively.
6: Update the occurring probabilities $\{\beta_j\}_{j=1,2,\cdots,J}$ and $\{\alpha_k\}_{k=1,2,\cdots,K}$ using (4.59) and (4.60), respectively.
7: Update the degree of freedom, $v$, by solving (4.61).
8: If $\frac{||\Theta^{h+1}-\Theta^h||^2}{||\Theta^h||^2} \leq \varepsilon$, terminate the procedure; otherwise, let $h = h + 1$ and go back to step 2.
```

4.4 Experimental Verification

In this section, validity of the proposed algorithm for developing a robust predictive model for LPV multirate processes is explored through the same lab-scale hybrid three-tank system described in Chapter 3. The figure and details about the hybrid three-tank system were given in 3.5. As shown in Figure 3.2, this system consists of two pumps, nine valves and three tanks. Water is fed into tank 1 and tank 3 via pump 1 and pump 2, respectively. Also, water is fed into the middle tank (tank2) as it is connected to the side tanks through the valves V1-V4, V6 and V9. Valves V5, V7 and V9 connect the tanks to the storage tank located at the bottom of the tanks. Water level in the tank relative to the locations of the valves can put the system on different operating conditions. Thus, the system behavior varies depending on the water level in the tanks.

In this experiment, it is assumed that all the valves are open except V6 and V8. The inlet flow rates of the pumps are considered as the input variables and their data are recorded every 10 seconds. The right-hand-side flow rate is a PRBS. The flow rate signal is varying between two steady-state values equal to 4.9 ℓ/min and 6 ℓ/min, respectively, with same variations $[-0.3,\ 0.3]$. Moreover, the left-hand-side flow rate signal is chosen as a PRBS with a fixed steady-state value equal to 5.5 ℓ/min with variations $[-0.3,\ 0.3]$. Thus, the system experiences two operating modes with a transition in between. In this study, the scheduling variable is water level in tank 3 as it reflects the changes in operating modes. Also, water level in tank 2 is the output of interest recorded every 100 seconds, *i.e.* it is updated at a slow-rate. Then, the slow-rate data are artificially integrated over $\ell_i \in \{2, 3, 4, 5\}$ and are measured with the artificially introduced delays $\lambda_i \in \{1, 2, 3\}$. The generated integration periods and delays change randomly between the given values and their actual occurrence probabilities are reported in Table 4.3. In this study, it is assumed that the upper bounds for the integration period and delay are $K = 5$ and $J = 4$, respectively.

Also, to evaluate the robustness of the developed model against the outliers, the slow-rate output is corrupted with 2% and 10% outliers, respectively. The data used for training, which include 3000 fast-rate data and 300 slow-rate ones, are shown in Figure 4.1. While 59% of the training data belong to the first operating condition, 37% belong to the second operating condition. The rest of the data are sampled from the transitions. To verify the performance of the proposed model, the coefficient of determination (R^2) and RMSE are used.

Using the proposed robust VB-based model for LPV multirate processes contaminated with the stated practical issues above, the fast-rate output for the testing dataset is reconstructed as shown in Figure 4.2. From Figure 4.2 it can also be seen that the accuracy of predictions at the second working condition is less than the first one due to fewer training data available in the second mode. The RMSE and R^2 values are reported in Table 4.2. For 2% outlier ratio, it is observed that the VB-based model gives $\mathrm{R}^2 = 86.15\%$ while the robust one further improves it to 89.41%. It can also be seen that the estimation accuracy is decreased when there exist 10% outliers among the process output data. However, the superiority of the proposed robust soft sensor model over the other model becomes further highlighted. The results have also been compared with a VB-based soft sensor model that does not consider the integration periods and treats the delay as a fixed value (called VB-ignore in this paper). This comparison is conducted since ignoring existence of the integration period and assuming a fixed delay is a common practice in most of the existing techniques. For a fair comparison, the most occurring delay, *i.e.* $\lambda = 2$, has been selected for the VB-ignore algorithm and same initial values have been selected for the parameters of the models. Table 4.2 shows that the estimation accuracy is considerably degraded by disregarding the integration period and using a fixed delay.

Table 4.3 shows the occurrence probabilities for each of the possible integration period and delay values after the convergence. The results verify a satisfactory estimation for the parameters. It is seen that some of the probabilities are estimated

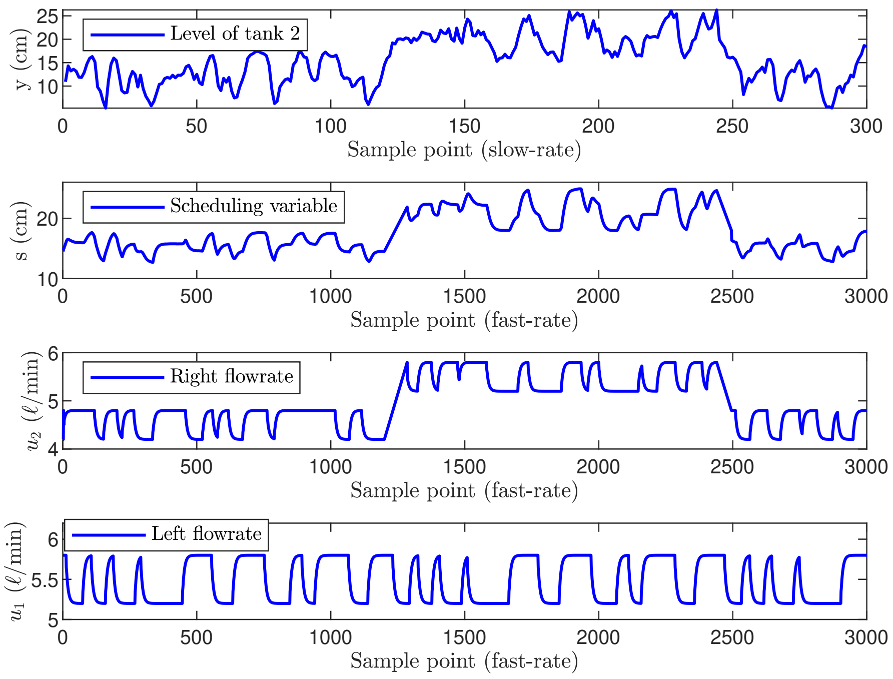


Figure 4.1: Input and output data points for training

with a slight bias because we have assumed that the true upper and lower bounds are not known. Furthermore, the number of unknown parameters is to some extent large in this study. Being an advantage of the developed soft sensor model, the existing integration periods and delays along with their occurrence probabilities are obtained even if their actual bounds are not known. This observation can be seen in Table 4.3 where the estimated occurrence probabilities of the non-existing integration periods, *i.e.* $k = 1$, and delays, *i.e.* $j = 4$, are close to zero. Now that once the true bounds have been estimated, these values can be updated and then run the algorithm again to improve the results as now the number of unknown parameters have decreased.

4.5 Conclusion

This study developed a robust VB-based soft sensor model for LPV processes with slow-rate integrated output measurements. Several important practical issues including unknown varying integration period, unknown varying time-delay and presence

Table 4.2: RMSE and R^2 values of the output prediction at fast rate

Outlier ratio	Index	VB ignore	VB	Robust VB
2%	RMSE	0.1322	0.1185	0.1012
	R^2	0.7763	0.8615	0.8941
10%	RMSE	0.1612	0.1485	0.1187
	R^2	0.7119	0.8017	0.8622

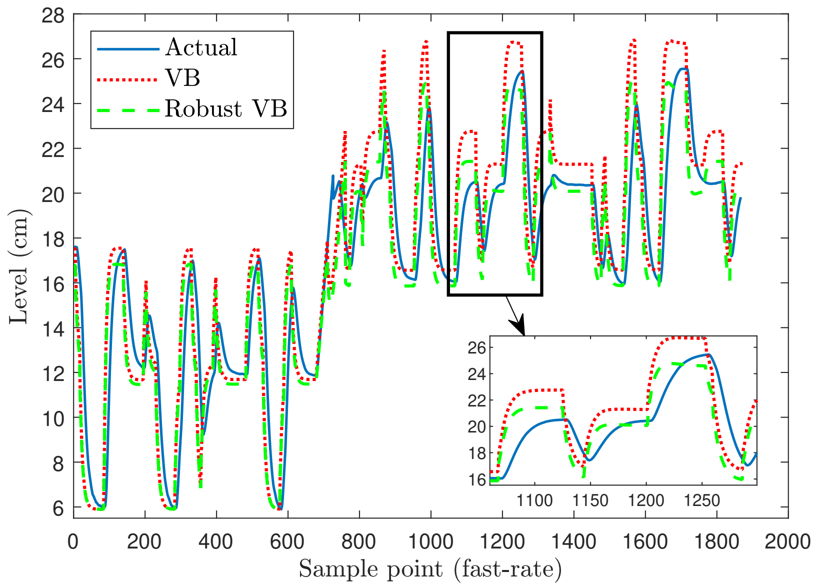


Figure 4.2: Output prediction at the fast rate

Table 4.3: Estimated parameters of the hidden variables in presence of 10% outlier output data

Hidden variables	Parameters	True values	VB	Robust VB
integration periods	α_1	0	2×10^{-4}	4×10^{-5}
	α_2	0.05	0.032	0.039
	α_3	0.1	0.073	0.083
	α_4	0.15	0.115	0.122
	α_5	0.7	0.790	0.761
Delays	β_1	0.2	0.141	0.163
	β_2	0.5	0.540	0.531
	β_3	0.3	0.252	0.262
	β_4	0	0.061	0.042
Relative error			0.158	0.107

of outliers were considered in the problem. The proposed VB algorithm estimates the parameters of the predictive model and noise variance along with their uncertainties. Moreover, the algorithm is robust against outlier measurements by using a t-distribution rather than the commonly used Gaussian distribution for measurement noise characterization. Furthermore, the probability of occurrence of each of potential integration period and time-delay was estimated based on a nonparametric-based approach. The validation results on a hybrid three-tank lab-scale system showed a reliable prediction for the quality variables at a fast rate. While large integration periods and time-delays increase the number of unknown parameters, the importance of considering these issues is evident from this work.

Chapter 5

A Vision Predictive Model with Application to Quality Variable Prediction in Flotation Process [1]

5.1 Introduction

A flotation process is commonly used in mining industry to separate valuable minerals from the unwanted ones. The properties of the froth, such as its stability, color and texture, can have a significant impact on the efficiency of the flotation process and the final product quality. In Figure 5.1, a schematic of a flotation process can be seen. In oil sands operation, which is one of the main oil resources in Canada, a primary separation cell (PSC) separates bitumen from the gangue using water-based gravity separation. In consequence, PSC generates froth, middling, and tailings. The froth, middling, and tailings layers should contain only bitumen, water, and solids, respectively. However, depending on the process operating conditions, the three layers are often a mixture of bitumen and gangue. It is common for the froth layer to contain contaminants such as water and solids. For further separation, the froth needs to be sent to a flotation cell. The tailings also contain residual bitumen. Residual bitumen

[1]This chapter has been accepted for publication as **Yousef Salehi**, Kaiyu Zhou, Biao Huang, Xuehua Zhang, "Image restoration and analysis with application to quality variable prediction in flotation process", in *Journal of Process Control*, Sept. 2023.
A short version of this chapter has also been accepted for publication as **Yousef Salehi**, Amir Mohseni, Kaiyu Zhou, Biao Huang, Xuehua Zhang, "A computer vision system for bitumen content estimation in flotation froth with degraded images", in *IFAC World Congress*, 2023.

can contribute to bitumen loss and emissions during disposal of the waste [80, 81]. Thus, tailings are sent to a flotation cell to extract residual bitumen. Bitumen should be extracted from the flotation cell's froth as much as possible while preserving the flotation stability. To accomplish this, many variables need to be adjusted timely (*e.g.* air flow rate, reagent dosage, froth depth). It is therefore essential to measure the concentration of the froth components on a regular basis.

It is often difficult and costly to measure the bitumen concentration of the froth (froth of the flotation cell), which is relatively small. A lab analysis is commonly performed on samples collected across the batch for measuring the concentration. Although lab result is accurate, it can take several hours to complete [82]. The long delay associated with lab analysis will directly affect the effectiveness of the process control solutions. A visualization of froth also provides valuable information about the concentrate grade and the flotation performance. Operators adjust the operation variables based on visual features, such as the color and size of the bubbles. But maintaining optimal flotation performance over long periods of time is difficult since continuous human monitoring is required [83]. There may also be errors associated with this type of monitoring since different operators may interpret the same conditions differently.

The use of computer vision has become popular for the analysis of different flotation froth processes using image features [84–86]. Essentially, various features can be extracted to either build a regression model or classify the froth concentration grade [16]. Color, texture, bubble size distribution, froth stability and velocity are among the numerous features that can be extracted from the froth images [86]. Thus, many tools have been developed to extract features from images. A classical way to extract texture features such as homogeneity, entropy, and contrast is through the use of gray-level co-occurrence matrix (GLCM) [54]. Moreover, CNNs can automatically extract numerous low and high-level features. However, the deep features lack physical meanings, are heavily dependent on data sample size, and require a lot of computation. This may not be feasible due to the high costs of labeling the froth concentration. To avoid the training from scratch, a pre-trained CNN on an existing large data set can be used for the task by using transfer learning [87]. In this regard, numerous CNN-based machine vision systems have been developed to monitor and control the flotation process [86, 88]. In [89], AlexNet, a pre-trained CNN, was used to extract froth features to predict the froth condition. Further, [85] designed a feature reconstruction with a weight-shared kernel network, and the features then were used as inputs for a fully connected network to monitor the flotation froth performance. In [84], the authors investigated the importance of image frames in a sequence as well as appearance and motion features of froth for developing soft sensor models. To improve the prediction accuracy, [90] designed a hybrid neural network combined with three different CNNs to extract the deep features. A real-time soft sensor model based on the extracted image features was developed in [82] to predict the bitumen froth grade and recovery rate. However, it ignored the possible froth image degradation. Further, a commercial software was used to directly provide certain features such as froth velocity, stability and so on. Despite the benefits and multiple features that commercial systems offer [83], their higher cost and the necessity for regular calibration and maintenance to ensure accuracy and reliability can be limiting factors. An overview of the recent advances in froth image analysis can be found in [15]. While most of the discussed works assume high-quality images, it is essential to acknowledge that images can be affected by various sources including camera noise, varying lighting conditions, marks, unwanted objects and so on. In this regard, an image restoration should be performed before feature extraction. For instance, illumination variation of the froth surface was considered in [91]. In addition, errors due to instruments malfunction or humans may lead to inaccurate labels of froth concentration. Therefore, a modeling process may need to take into account froth concentration observations outliers.

In the present study, we estimate the froth concentration using computer vision

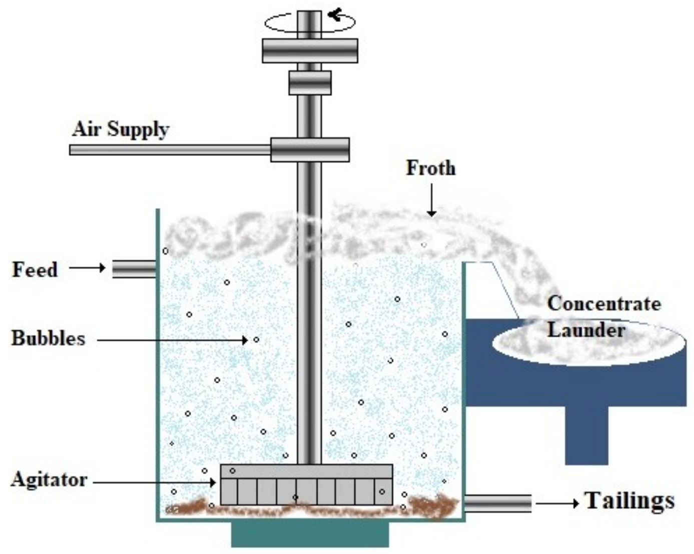


Figure 5.1: Schematic drawing of a flotation column

technology, taking into account contaminated images and outliers. A model with multiple image features is considered to estimate the concentration. An EM algorithm is used to estimate the unknown parameters of the model. The principal contributions of this study are as follows.

1. Restoration of the contaminated images with bright lighting spots using a modified spatial-based KF.

2. Extraction of deep froth image features using CNN and transfer learning, and integrating features over the batch with unique attention weights assigned to successive frames.

3. Development of a robust-to-outlier predictive model for cumulative bitumen content of each batch based on image features.

5.2 Image Restoration Using Kalman Filter

Various factors can degrade an image, including environmental conditions, transmission channel interference, and compression artifacts. Therefore, a restoration step is necessary to improve the quality of the image for further processing. Due to the two-dimensional nature of images, it is challenging to directly apply classical filtering approaches [92], such as KF. Thus, it is necessary to formulate a proper state vector to filter the images [93]. In this section, a state-space framework is introduced to model a two-dimensional contaminated gray-scale image following the approach proposed in [93]. Afterward, the state-space model is used to restore an estimate of true image from the image contaminated by bright lighting spots and camera noise.

5.2.1 Image Model

Consider an M by N image where M and N are the number of rows and columns of the image frame, respectively. A state at any given time instant is represented by nine pixels shown in Figure 5.2, and the process of horizontal state propagation can be seen in Figure 5.3. Once the state propagation in the horizontal direction from left to right is completed for each strip, the same procedure will be followed for the next strip by sliding one row down. The goal is to estimate the middle pixel, *i.e.* the blank pixel in Figure 5.2, since it can use neighboring pixels' estimations. The local state-space model for the image is written as:

$$X(k+1) = AX(k) + W(k) \tag{5.1}$$

$$Y(k) = CX(k) + V(k) \tag{5.2}$$

that is

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ . \\ . \\ x_9(k+1) \end{pmatrix} = A \begin{pmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ x_9(k) \end{pmatrix} + \begin{pmatrix} w_1(k) \\ w_2(k) \\ . \\ . \\ w_9(k) \end{pmatrix} \tag{5.3}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{6,5} & A_{6,6} & A_{6,7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{7,7} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{8,5} & 0 & A_{8,7} & A_{8,8} & 0 \\ A_{9,1} & A_{9,2} & A_{9,3} & A_{9,4} & A_{9,5} & A_{9,6} & 0 & A_{9,8} & A_{9,9} \end{bmatrix} \tag{5.4}$$

$$C = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1] \tag{5.5}$$

and, X is the current state vector, while Y indicates the observed image. The state noise and observation noise, which are assumed to be white Gaussian noises, are represented by W and V, respectively.

The unusual numbering of these blocks is solely for the sake of easier formulating and programming [93]. It is assumed that pixels 1, 2, 3, and 4 only repeat the values in their previous spatial position while their corresponding state noise is zero. To make it clearer, for example, $x_1(k+1) = x_2(k)$. The support pixels for other pixels are given in Table 5.1. Thus, the transition matrix A is obtained as given in (5.4).

Table 5.1: Region of support for the pixels

Pixel	Support pixels
$x_6(k+1)$	$x_i(k), i = 5, 6, 7$
$x_7(k+1)$	$x_i(k), i = 7$
$x_8(k+1)$	$x_i(k), i = 5, 7, 8$
$x_9(k+1)$	$x_i(k), i = 1, 2, 3, 4, 5, 6, 8, 9$

The strip is advanced one row down in each strip. As the state propagates from left to right, blocks 1, 2, 3, 4, and 5 are shifted from the previous state. Blocks 6, 7, 8, and 9 are evaluated using four concurrent estimators. It should be noted that the filtered estimate of pixel 9 with a full plane of support is the only pixel estimate that is saved [93].

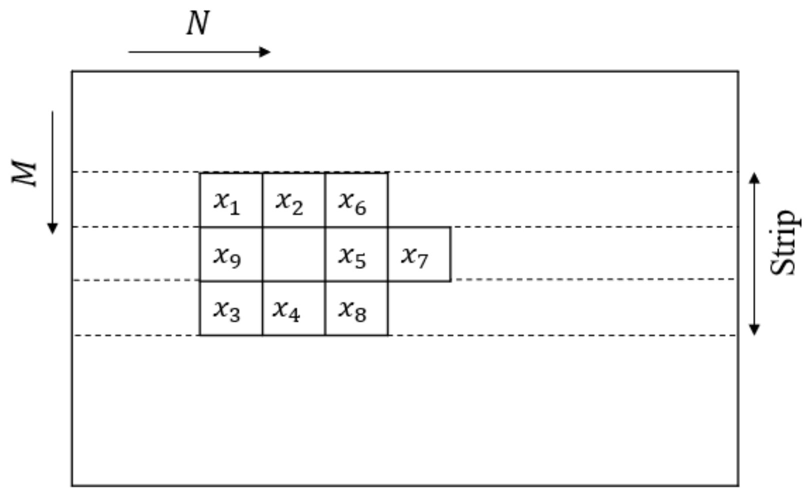


Figure 5.2: Image scanning

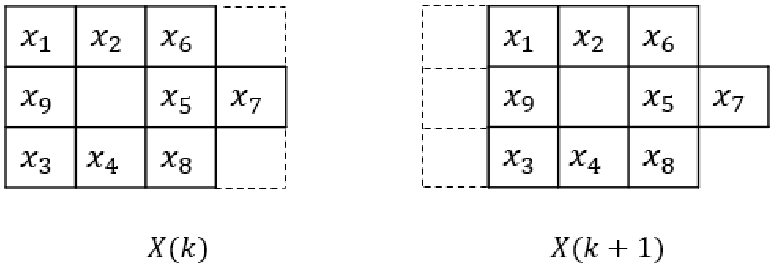


Figure 5.3: State movement by one block

5.2.2 Image Model Parameter Estimation

The parameters of the state-space image model introduced in the previous section can be obtained based on Yule-Walker equations [93]. Among these parameters are some elements of the state transition matrix and their corresponding state noise covariances. This section describes how to estimate the parameters of the state-space model. From (5.3) we have:

$$x_7(k+1) = A_{7,7}x_7(k) + w_7(k) \tag{5.6}$$

Post-multiplying (5.6) by $x_7(k)$, taking the expectation, and using the following orthogonality principle:

$$E[w_7(k)x_7(k-1)] = 0 \tag{5.7}$$

yields

$$\rho_{7,7}(t) = A_{7,7}\rho_{7,7}(t-1) + Q_{w_{7,7}}\delta(t) \tag{5.8}$$

where $\mathbb{E}$ indicates the expectation operator, and $\rho_{i,j}(t) \triangleq \mathbb{E}[x_i(n)x_j(n-t)]$ is the correlation between pixels i and j. Also, $Q_{w_{i,j}}$ denotes the variance of the state noise, *i.e.* variance of $w_{i,j}$. Additionally, δ is the Dirac delta function.

Transposing (5.8) and using the property $\rho_{i,j}^T(t) = \rho_{j,i}(-t)$ gives

$$\rho_{7,7}(-t) = \rho_{7,7}(1-t)A_{7,7}^T + Q_{w_{7,7}}\delta(t) \tag{5.9}$$

Plugging $t = 0, 1$ in this equation gives the following Yule-Walker equation

$$\begin{bmatrix} \rho_{7,7}(0) & \rho_{7,7}(1) \\ \rho_{7,7}^T(1) & \rho_{7,7}(0) \end{bmatrix} \begin{bmatrix} 1 \\ -A_{7,7}^T \end{bmatrix} = \begin{bmatrix} Q_{w_{7,7}} \\ 0 \end{bmatrix} \tag{5.10}$$

The solution to (5.10) yields $A_{7,7}$ and $Q_{w_{7,7}}$.

Similarly, we have

$$x_6(k+1) = \begin{bmatrix} A_{6,5} & A_{6,6} & A_{6,7} & A_{6,8} \end{bmatrix} \begin{bmatrix} x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \end{bmatrix} + w_6(k) \tag{5.11}$$

Yule-Walker's vector equation can be expressed as follows using the orthogonality principle.

$$\begin{bmatrix} \rho_{6,6}(0) & \rho_{6,5}(1) & \rho_{6,6}(1) & \rho_{6,7}(1) & \rho_{6,8}(1) \\ \rho_{6,5}(1) & \rho_{5,5}(0) & \rho_{5,6}(0) & \rho_{5,7}(0) & \rho_{5,8}(0) \\ \rho_{6,6}(0) & \rho_{6,5}(0) & \rho_{6,6}(0) & \rho_{6,7}(0) & \rho_{6,8}(0) \\ \rho_{6,7}(1) & \rho_{7,5}(0) & \rho_{7,6}(0) & \rho_{7,7}(0) & \rho_{7,8}(0) \\ \rho_{6,8}(1) & \rho_{8,5}(0) & \rho_{8,6}(0) & \rho_{8,7}(0) & \rho_{8,8}(0) \end{bmatrix} \begin{bmatrix} 1 \\ -A_{6,5}^T \\ -A_{6,6}^T \\ -A_{6,7}^T \\ -A_{6,8}^T \end{bmatrix} = \begin{bmatrix} Q_{w_{6,6}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.12}$$

The solution of (5.12) provides $Q_{w_{6,6}}$ and $A_{6,5:8}$. The same procedure is followed for the rest of the parameters. It should be noted that

$$\rho_{i,i}(0) = \rho_{j,j}(0), \qquad i \neq j \tag{5.13}$$

$$\rho_{i,j}(-t) = \rho_{j,i}^T(-t), \qquad t \text{ any integer} \tag{5.14}$$

By using these equivalences, considerable computation time can be saved. So far, we have explained how to estimate the elements of A and the diagonal elements of Q. To complete the parameter estimation process, the off-diagonal elements of Q need to be determined. For $Q_{w_{6,7}}$, post-multiplying both sides of (5.11) by $x_7^T(k)$ and using the orthogonality principle, $E[w_7(k)x_7^T(k-1)] = 0$, gives

$$Q_{w_{6,7}} = \rho_{6,7}(0) - \begin{bmatrix} A_{6,5} & A_{6,6} & A_{6,7} & A_{6,8} \end{bmatrix} \begin{bmatrix} \rho_{5,7}(-1) \\ \rho_{6,7}(-1) \\ \rho_{7,7}(-1) \\ \rho_{7,8}(-1) \end{bmatrix} \tag{5.15}$$

The procedure for determining $Q_{w_{8,7}}$ and $Q_{w_{6,8}}$ is similar. For $Q_{w_{6,9}}$ we obtain

$$Q_{w_{6,9}} = \rho_{6,9}(0) - [\rho_{6,1}(1) \;\; \rho_{6,2}(1) \;\; \rho_{6,3}(1) \;\; \rho_{6,4}(1)\rho_{6,5}(1) \;\; \rho_{6,6} \;\; \rho_{6,7}(1) \;\; \rho_{6,8}(1) \;\; \rho_{6,9}]A_9^T \tag{5.16}$$

The same applies to $Q_{w_{8,9}}$ and $Q_{w_{7,9}}$. The remaining off-diagonal elements can be obtained by the relationship $Q_{w_{j,i}} = Q_{w_{i,j}}^T$. For $i \in [0,4]$, $j \in [1,9]$ and $i \in [6,9], j \in [1,5]$ we have $Q_{w_{i,j}} = 0$ since the shifting process produces the corresponding blocks. This work uses measurement noise covariance as a tuning parameter. A detailed explanation can be found in [93].

5.2.3 KF Algorithm in the Presence of Missing Observations

Many approaches have been proposed to solve state estimation problems in literature. Among them, the KF gives the optimal state estimation for linear systems when the state noise and measurement noise both follow a Gaussian distribution with zero mean and variances Q and R, respectively, with appropriate dimensions. In the current study, however, some parts of the images are obscured by bright lighting spots. The image pixels corrupted by the bright lighting spots are considered missing pixels. To restore the degraded image, we adopt a modified version of the KF algorithm that can handle missing observations [94]. Consider η as a binary matrix with the same size as the image where its elements are equal to one except for the spatial position of the bright lighting spots where zero is assigned to the corresponding elements, meaning the corresponding pixel observation is missed. The steps for image restoration using a modified KF are provided in (5.17)-(5.21).

$$\hat{X}_{k+1|k} = A\hat{X}_{k|k} \tag{5.17}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q \tag{5.18}$$

$$K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1} \tag{5.19}$$

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + \eta_{k+1}K_{k+1}(y_{k+1} - C\hat{X}_{k+1|k}) \tag{5.20}$$

$$P_{k+1|k+1} = P_{k+1|k} - \eta_{k+1}K_{k+1}CP_{k+1|k} \tag{5.21}$$

where P and K denote the estimation error covariance matrix and Kalman gain, respectively.

As shown in (5.20), when the current pixel observation is missing, only the previously estimated value is used as an update. A convergence analysis of the KF in the presence of missing observations has been given in [94]. The one-directional recursion of the above image restoration algorithm is one of its limitations. In this work, we propose to use three other KFs simultaneously to restore the image by propagating the state along different directions, *i.e.* from right to left, top to bottom, and bottom to top. When the four KFs have been applied, the final image quality is determined by fusing the four filtered images together. For each restored image, the inverse of the error covariance matrix is used as the weight [35] as given below.

$$\hat{X} = \frac{P_{LR}^{-1}\hat{X}_{LR} + P_{RL}^{-1}\hat{X}_{RL} + P_{TB}^{-1}\hat{X}_{TB} + P_{BT}^{-1}\hat{X}_{BT}}{P_{LR}^{-1} + P_{RL}^{-1} + P_{TB}^{-1} + P_{BT}^{-1}} \tag{5.22}$$

where the subscripts LR, RL, TB, and BT indicate left to right, right to left, top to bottom and bottom to top, respectively.

5.3 Image Feature Extraction

Various visual features can be extracted from the captured froth images. This section explains how color and statistical texture features can be extracted. A CNN-based network is also described as a tool for extracting other complex visual features. A combination of the extracted features serves as inputs for the following regression model.

5.3.1 Color and Statistical Features

Color of the froth is a basic, yet informative feature about the froth concentration. This study takes into consideration the HSV color space for its more robustness against illumination variations when defining a color image. Mean values of the three channels, *i.e.* H, S, and V, are set as the color features. Based on the intensity of each position in the image, texture features are determined. Three types of texture statistics exist: first order, second order, and higher order. The GLCM can be used to extract image statistical texture features such as energy, entropy, contrast, homogeneity, correlation, dissimilarity, and inverse difference moment [54].

5.3.2 CNN-Based Features

CNN models have greatly improved computer vision tasks, particularly for image processing [95]. VGG16 [96] is a pre-trained CNN, trained on a large data set of images, allowing it to learn to recognize a wide variety of objects and scenes. This pre-trained model can be used as a starting point for other computer vision tasks, such as fine-tuning it on a smaller data set for a specific task or using its learned features to train another model. This is a common practice in the field of computer vision and it is known as transfer learning, which can save training time and computational resources. As the convolution layers progress, more complex features are extracted, with a fully connected layer of the VGG16 network being used to extract global semantic information of an image. However, we do not use the fully connected layer and only use the extracted features in this work.

5.4 Model Development

We consider a batch wise image processing for a froth sampling process. Froth is mixture of oil and water. In this work, we are interested in building a model to predict froth concentration using images. In the experiment setup, froth is collected batch-wise, and its concentration is measured through lab analysis. Each batch contains only one label for the froth concentration of the entire batch and J consecutive image frames taken within the batch. Thus, for every input variable (*i.e.*, feature extracted from image) to a model, there are J values. For each feature, an integration of the feature values across the J frames is calculated using a unique attention weight. Due to the fact that froth concentration decreases as batch progresses (see Figure 5.4), the weight is assumed to follow an exponential decaying function [81]. The fast-rate inputs are therefore transferred to the slow-rate inputs by integrating each input over the batch with its corresponding frame's weight. Then, a regression model at a slow rate is built between the output and integrated inputs as:

$$U_i^m = \frac{1}{\sum_{j=1}^{J} c_j} \sum_{j=1}^{J} c_j u_{i,j}^m \tag{5.23}$$

$$y_i = \phi_i^T \theta + e_i \tag{5.24}$$

where

$$c_j = \lambda_1 + \exp\big(-\lambda_2 j\big), \; j = 1, 2, \ldots, J \tag{5.25}$$

$$\phi_i^T = [U_i^1 \; U_i^2 \; \ldots \; U_i^M \; 1] \tag{5.26}$$

$$\theta^T = [\theta_1 \; \theta_2 \; \ldots \; \theta_M \; \theta_{bias}] \tag{5.27}$$

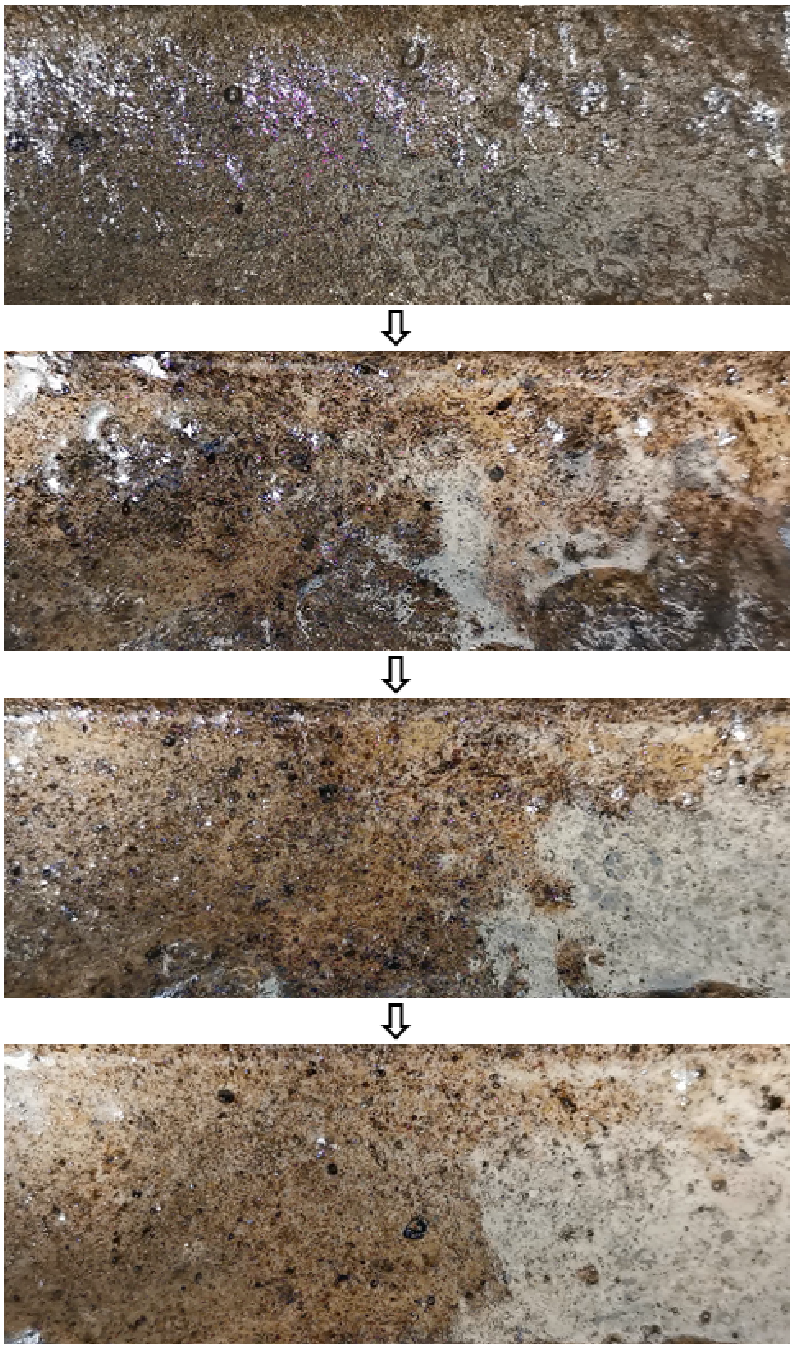


Figure 5.4: Sequential froth images recorded every 1 min

Table 5.2: List of variables in the problem formulation

Notation	Description
y_i	Output measurement for batch i.
$u_{i,n}^m$	mth feature of the nth frame in batch i
U_i^m	mth integrated feature in batch i
M	Total number of features (input variables)
θ	Model parameters
e	Measurement noise
σ^2	Measurement noise variance
υ	Degree of freedom
r	Measurement noise scaling parameter
λ_1, λ_2	Tuning parameters of the attention weights assigned to image frames in a batch
c_j	The attention weight assigned to jth image frame
J	Total number of frames in a batch

c_j is the associated attention weight for the jth image frame. Also, e shows the output measurement noise. Gaussian distribution is widely used in noise characterization. However, it cannot handle outliers due to its short tails. For robustness against outliers, the measurement noise e is assumed to follow a t-distribution with the mean zero, variance σ^2, and the degree of freedom υ [72]. A list of the notation used in the problem formulation is provided in Table 5.2. For the EM algorithm problem formulation, the observed variables (D_{obs}), hidden variables (D_{hid}) and parameters to be estimated (Θ), are identified as

$$D_{obs} = \{Y, U\} = \{y_{1:N}, U_{1:N}^1, \cdots, U_{1:N}^M\} \tag{5.28}$$

$$D_{hid} = \{R\} = \{r_{1:N}\} \tag{5.29}$$

$$\Theta = \{\theta, \sigma^2, \upsilon\} \tag{5.30}$$

We aim to develop an EM algorithm that solves the parameter estimation problem for the regression model in the presence of outlying output observations. The EM algorithm is a powerful method for finding the maximum likelihood estimate (MLE) of the parameters in a statistical model. It is particularly useful when the data is incomplete. The algorithm is an iterative method that alternates between the E-step and M-step. The E-step estimates the expected value of the complete data log-likelihood function given the current estimates of the parameters. The M-step maximizes the expected value of the complete data log-likelihood function computed in the E-step with respect to the parameters. This step estimates new parameters that maximize the expected likelihood of the data. Beginning with an initial value of the unknown parameters, the EM algorithm repeats its steps until convergence is achieved. The EM algorithm has been shown to converge [49].

5.4.1 Expectation Step

The expectation of the log-likelihood function of the complete data defined in (5.28)-(5.29), known as Q-function, is

$$Q(\Theta|\Theta^h) = \mathbb{E}_{D_{hid}|D_{obs},\Theta^h}\left[\log(p(D_{hid}, D_{obs}|\Theta))\right] \tag{5.31}$$

where Θ^h shows the estimated parameters at the hth iteration. According to the definition of the Q-function in (5.31), we obtain Q-function for the problem of interest as follows:

$$Q(\Theta|\Theta^h) = \mathbb{E}_{R|Y,U,\Theta^h}\left[\log(p(Y, U, R|\Theta))\right] \tag{5.32}$$

It is possible to decompose the likelihood function of the complete data set as follows using the probability chain rule.

$$p\left(Y, U, R|\Theta\right) = p\left(Y|U, R, \Theta\right) p\left(R|\Theta\right) p(U|\Theta) \tag{5.33}$$

Given the set of N observations, the probability of the observations is independent of each other and assumes the measurement noise distribution, which can be expressed as the product of the marginal distributions for each output, that is,

$$p\left(Y|U, R, \Theta\right) = \prod_{i=1}^{N} p(y_i|\phi_i^T, r_i, \Theta) \tag{5.34}$$

The marginal distributions are each expressed as follows:

$$p\left(y_i|\phi_i^T, r_i, \Theta\right) = \mathcal{N}(y_i|\phi_i\theta, \frac{\sigma^2}{r_i}) \tag{5.35}$$

where

$$p\left(r_i|\upsilon\right) = G(\frac{\upsilon}{2}, \frac{\upsilon}{2}) \tag{5.36}$$

and, $\mathcal{N}$ and G represent the Gaussian distribution and Gamma distribution, respectively. Substituting (5.33)-(5.36) into (5.32) yields (5.37).

$$\begin{aligned}
Q(\Theta|\Theta^h) &= \mathbb{E}_{R|Y,U,\Theta^h}\left[\log(p(Y,U,R|\Theta))\right] = \sum_{i=1}^{N}\int p(r_i|D_{obs},\Theta^h)\log p(y_i|\phi_i^T, r_i,\Theta)dr_i \\
&+ \sum_{i=1}^{N}\int p(r_i|D_{obs},\Theta^h)\log p(r_i|\upsilon)dr_i + \sum_{i=1}^{N}\int p(r_i|D_{obs},\Theta^h)\log p(U_i|\Theta^h)dr_i \\
&= \sum_{i=1}^{N}\int p(r_i|D_{obs},\Theta^h)\Big(-\frac{1}{2}\log 2\pi\sigma^2 + \frac{1}{2}r_i - \frac{(y_i-\phi\theta)^2}{2\sigma^2}\Big)dr_i \\
&+ \sum_{i=1}^{N}\Big(\frac{\upsilon}{2}\log\frac{\upsilon}{2} + (\frac{\upsilon}{2}-1)\int p(r_i|D_{obs},\Theta^h)\log r_i dr_i \\
&- \frac{\upsilon}{2}\int p(r_i|D_{obs},\Theta^h)r_i dr_i - \log\Gamma(\frac{\upsilon}{2})\Big) + C
\end{aligned} \tag{5.37}$$

where Γ denotes Gamma function.

As the input values are known and deterministic, the last term in (5.37) is independent of Θ. Thus, $p(U|\Theta) \equiv C_1$ where C_1 is a constant value with respect to Θ. Furthermore, we have:

$$\langle r_i\rangle = \int p(r_i|D_{obs},\Theta^h)r_i dr_i = \frac{1+\upsilon^h}{\frac{(y_i-\phi_i\theta^h)^2}{(\sigma^2)^h} + \upsilon^h} \tag{5.38}$$

$$\langle\log r_i\rangle = \int p(r_i|D_{obs},\Theta^h)\log r_i dr_i = -\log\left(\frac{\frac{(y_i-\phi_i\theta^h)^2}{(\sigma^2)^h} + \upsilon^h}{2}\right) + \psi\left(\frac{1+\upsilon^h}{2}\right) \tag{5.39}$$

where $\langle.\rangle$ indicates the expectation and ψ is the derivative of the logarithm of the Gamma function, *i.e.*, $\psi(\upsilon) = \frac{\partial\Gamma(\upsilon)}{\partial\upsilon}\frac{1}{\Gamma(\upsilon)}$.

The Q function in (37) will then be rewritten as follows:

$$\begin{aligned} Q(\Theta|\Theta^h) = &-\frac{N}{2}\log 2\pi\sigma^2 + \frac{1}{2}\sum_{i=1}^{N}\left(\langle \log r_i\rangle - \frac{\langle r_i\rangle(y_i-\phi\theta)^2}{\sigma^2}\right) \\ &+ \frac{Nv}{2}\log\frac{v}{2} - N\log\Gamma(\frac{v}{2}) + \frac{v}{2}\sum_{i=1}^{N}\Big(\langle \log r_i\rangle - \langle r_i\rangle\Big) \\ &- \sum_{i=1}^{N}\langle \log r_i\rangle \ + C \end{aligned} \tag{5.40}$$

where C is a constant term.

5.4.2 Maximization Step

To update the parameters, we maximize the Q-function with respect to each parameter, that is,

$$\Theta^{h+1} = \operatorname*{argmax}_{\Theta} Q(\Theta|\Theta^h) \tag{5.41}$$

As a result, when we take the derivative of the Q-function with respect to the model parameters and set them to zero, we obtain:

$$\theta^{h+1} = \left[\sum_{i=1}^{N}\langle r_i\rangle\phi_i^T\phi_i\right]^{-1}\left[\sum_{i=1}^{N}\langle r_i\rangle\phi_i^T y_i\right] \tag{5.42}$$

The noise variance can also be estimated by taking gradient with respect to σ^2 and setting it to zero, resulting in

$$(\sigma^2)^{h+1} = \frac{1}{N}\sum_{i=1}^{N}\langle r_i\rangle(y_i-\phi_i\theta)^2 \tag{5.43}$$

In addition, the degree of freedom can be estimated by taking the gradient of Q with respect to v and equating it to zero.

$$\sum_{i=1}^{N}(\langle \log r_i\rangle - \langle r_i\rangle) + N\bigg(\log\frac{v}{2} - \psi(\frac{v}{2}) + 1\bigg) = 0 \tag{5.44}$$

Solving the nonlinear equation in (5.44) gives v. We also need to estimate the attention weight parameters, λ_1 and λ_2. Taking the derivative of (5.45) with respect to λ_1 and λ_2, respectively, yields λ_1 and λ_2.

$$F(\lambda_1,\lambda_2) = \sum_{i=1}^{N}\left(\langle r_i\rangle(y_i-\phi_i\theta)^2\right) \tag{5.45}$$

Equations (5.23) and (5.26) show that ϕ is a function of λ_1 and λ_2. We have solved (5.45) using the nonlinear functions in MATLAB.

5.5 Experimental Verification

The massive mining production of oil sands ores has generated a large amount of oil sands tailings. Residual bitumen in the tailings is a source of greenhouse gases emissions. Efficient residual bitumen recovery benefits the environment as well as the oil recovery. The proposed method is evaluated in this section by estimating bitumen concentration in a lab-scale tailings flotation plant. A pipeline loop was utilized to recover the residual bitumen from the oil sands tailings, which is composed of 0.2 wt % bitumen, 50 wt % solids and 49.8 wt % process water was prepared. Figure 5.5 shows a schematic of a lab-scale oil sands flotation used in the experimental work [97]. On the right side of the loop, there is a power unit with a progressive cavity pump that allows the tailings to circulate steadily. In this experiment, the tailings stream was controlled to flow at 2 m/s. The glass trough near the power unit has multiple functions, including the inlet of oil sands tailings, froth sample collection and image capturing of the recovered froth. The experimental oil sands tailings were injected to the loop from the trough first. The heat exchanger at the bottom was set to the desired temperature of 42°C to heat the circulated tailings for approximate 20 minutes to achieve 42°C. After that, 30% hydrogen peroxide solution of 6 mL and 12 mL was injected into the tailings from the trough at various rates to investigate the bitumen recovery. The froth formed at the top of the trough as a result of continuous decomposition of bitumen, solids, and water. The recovered froth containing bitumen, solids, and entrapped water was formed in the trough. A camera was positioned at a height of 30 cm above the recovered froth surface. In addition, one piece of tinfoil was used to minimize light reflection on the froth surface by shading the scattered light in the environment. However, the images were still unavoidably affected by lighting conditions. Subsequently, the collected froth was transferred to the lab to

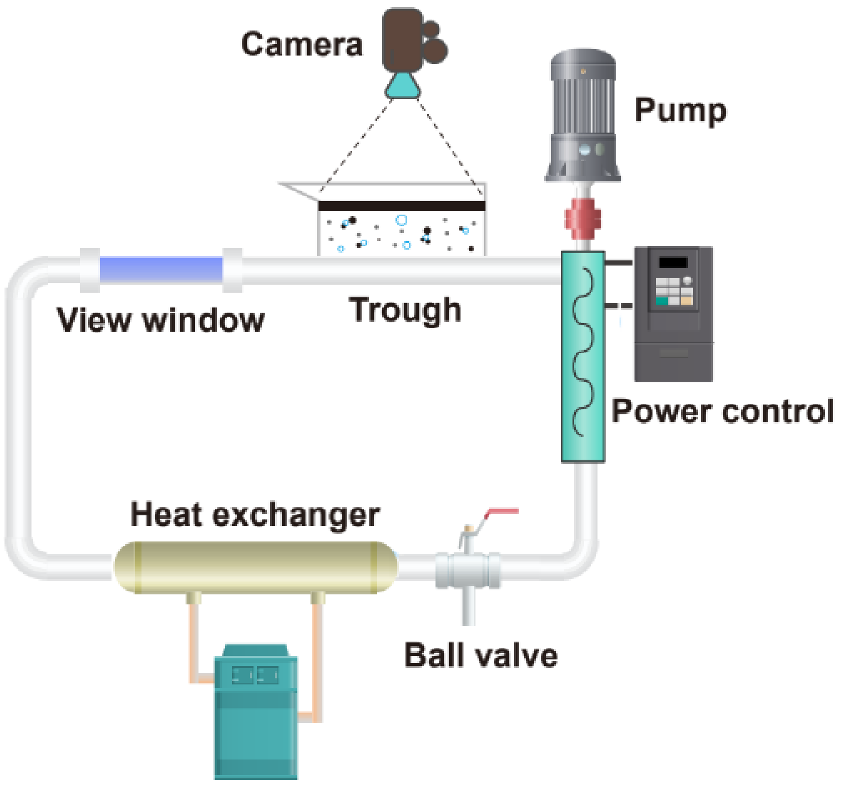


Figure 5.5: Schematic of the experimental setup

determine the solids, water, and bitumen contents for labeling using Dean-Stark. Figure 5.4 shows the visual changes in the froth over time within a batch. There is clearly more bitumen at the beginning of the batch, and less bitumen as the batch progresses. Hence, we assign a unique attention weight to each frame as considered in the proposed model in (5.23).

5.5.1 Data Expansion

The experiment involved 106 flotation batches, each lasting for eight minutes. In every batch, the operator tried ten times to collect the bitumen from the froth's surface and poured it into a container each time. Meanwhile, images were continuously captured throughout each batch. However, in this study, we only used the images taken right before each of the ten bitumen collections, *i.e.* we used ten images per batch, as only those sub-samples were accumulated and analyzed in the laboratory. Consequently, the concentration label of the cumulative froth sample over each batch used in the study is related with those ten images. To generate additional images artificially from the collected images, many classical and deep learning-based methods exist [98]. This study uses several traditional methods, including flipping, scaling, adding noise, and rotating images. The utilization of these conventional techniques led to the creation of 212 additional batches.

5.5.2 Results

Mean values of the three color channels are considered as color features. In addition, homogeneity, contrast, energy, and correlation are extracted from each image using GLCM, representing textural features. To further enhance the analysis of the images, VGG16 is used to extract deep features. Due to label limitations that may deteriorate the parameter estimation accuracy, we have selected only 10 mostly correlated deep features using principal component analysis (PCA). We use the coefficient of determination (R^2) and RMSE to test the performance of the proposed soft sensor model.

All the images need to be restored. The restoration process begins by automatically detecting bright lighting pixels and setting their corresponding η value to zero. Then the proposed KF in (5.17)-(5.22) is applied. Figure 5.6 displays a contaminated image along with its corresponding restored image. As can be seen, the froth image pixels affected by the lighting have been filtered satisfactorily. It is important to note that the restoration process becomes challenging when there is a significant absence of connected pixels, as the KF applies spatial filtering technique.Fig. 5.7 and Fig. 5.8 exhibit two additional examples used to assess the performance of the proposed restoration algorithm when images are subject to a Gaussian noise with a variance of 0.04 and the presence of marks on the image surface, respectively. These figures demonstrate that the algorithm effectively mitigates the impact of degradation with satisfactory accuracy. Once the images are restored, the proposed EM algorithm is applied and parameters of the regression model are estimated. The proposed vision model is validated using leave-one-out cross-validation. Cross-validation with leave-one-out is a special type of cross-validation that equating the number of folds by the number of instances in the data set. Using this method, the learning algorithm is applied to every instance, with the other instances acting as training sets and the selected instance acting as test. Fig. 5.9 shows the scatter plot for both predicted and true (laboratory result) bitumen concentration. The corresponding R^2 and RMSE values are given in Table 5.3. It is seen that $R^2 = 64.12\%$ is obtained using the proposed robust EM (REM) algorithm. We also compared the results with the REM algorithm applied to the degraded images without using restoration. It was noted that excluding the restoration process resulted in a decrease in accuracy of 4.08%. In order to illustrate the impact of considering deep features, we used the REM algorithm but with only the color and GLCM features (called REM-GLCM in this paper). As a result, the accuracy exhibited a decrease of 3.01% in comparison to the developed REM with image restoration. Further, we compared the results with those obtained when Gaussian distribution was used rather than the t-distribution. An R^2 of 58.61% was achieved using the regular least squares (LS) algorithm. Additionally, the proposed model, including the image restoration step, can estimate the cumulative bitumen content over a batch in less than two minutes using MATLAB R2019b on a CPU of frequency 2.60 GHz, eight cores, and 16-GB memory instead of taking a couple of hours as using Dean-Stark in the laboratory. The only associated cost is to fine-tune the model.

Table 5.3: RMSE and R^2 of the validation output estimation

Index	LS	REM-GLCM	REM without restoration	Proposed REM
RMSE	0.0539	0.0512	0.0558	0.0492
R^2	0.5861	0.6111	0.6004	0.6412

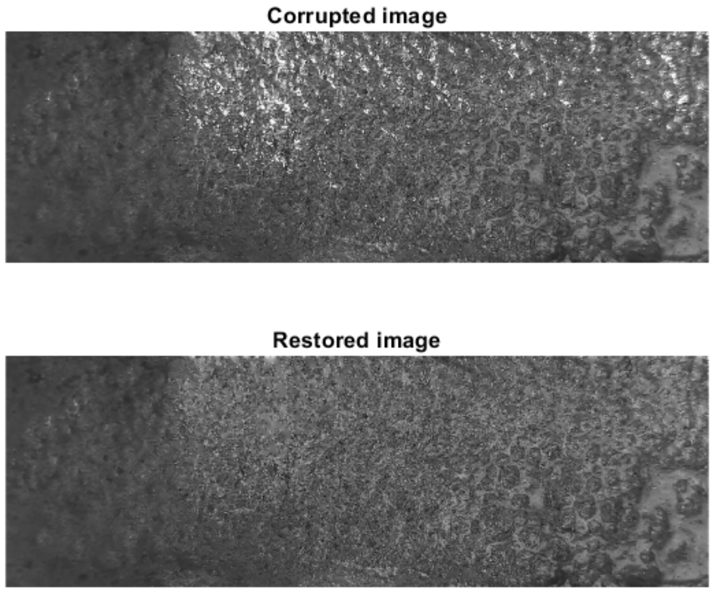


Figure 5.6: Reducing the effects of bright lighting spots using the modified KF

5.6 Conclusion

We propose a computer vision model to estimate the froth concentration grade in batch flotation processes. Due to the common disturbances such as lighting and camera noise, a modified KF-based algorithm is used to restore the images. Froth color, texture and deep features are then extracted and integrated from the restored images using GLCM and a pre-trained deep learning algorithm, VGG16. The highly correlated features are selected using PCA and used to construct a model to estimate the froth concentration. To estimate the model parameters, an EM algorithm is applied. Moreover, to improve the model robustness against outliers, the t-distribution is used in modeling the noise. A laboratory-scale bitumen flotation experiment data is used to assess the proposed algorithm's efficiency. An R^2 of 64.12% is achieved in estimating the concentration of the froth. Employing the identical algorithm without the restoration step leads to a decrease in accuracy, resulting in an R^2 of 60.04% for estimating the concentration. Moreover, excluding only deep features from the pro-

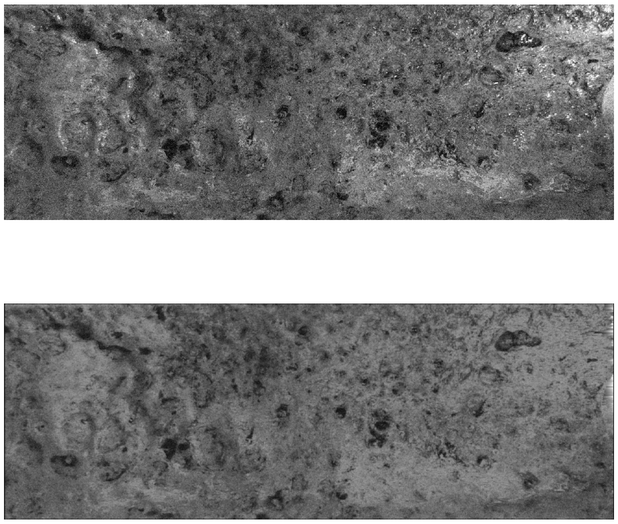


Figure 5.7: Noise reduction result using the modified KF

posed model reduces the R^2 to 61.11% which supports the importance of considering deep features. Further, the proposed vision based algorithm is more time efficient compared to the laboratory analysis.

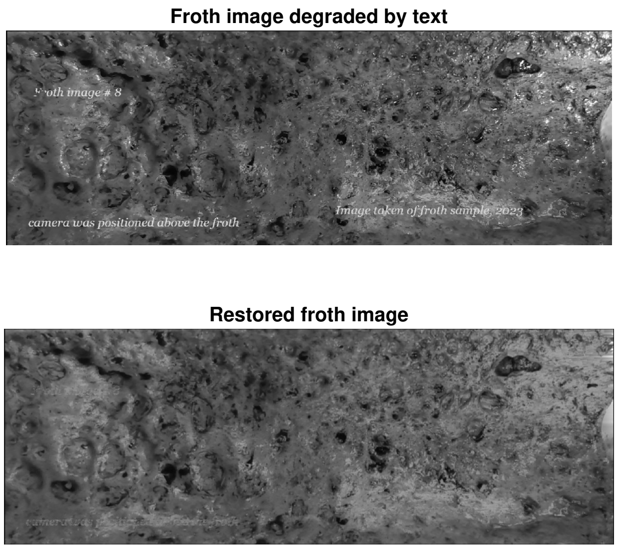

Froth image degraded by text
Froth image # 8
camera was positioned above the froth
Image taken of froth sample, 2023
Restored froth image

Figure 5.8: Text removal result using the modified KF

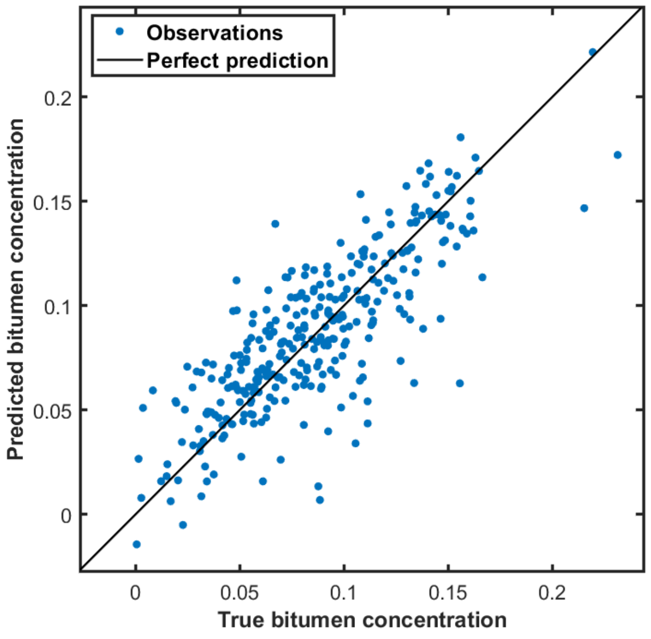


Figure 5.9: Scatter plot for the validation results

Chapter 6

Robust Computer Vision Model for Predicting Process Variables with Slow-Rate Measurements [1]

6.1 Introduction

Computer vision models are powerful tools in process monitoring and control [11–13]. Vision models leverage image processing and machine learning techniques to predict process variables that are not directly measured due to technical or economic constraints. The detection of certain objects or variables in real-time is another industrial application of vision-based models [99, 100]. Through the analysis of images, the predictive models can relate the relevant features or patterns to the target process variables. The advantage of vision predictive models lies in their non-intrusive nature, as they do not require physical contact with the subject. Additionally, they can capture spatial and temporal information at a lower cost, enabling real-time monitoring and control. However, developing accurate vision predictive models requires appropriate images of sufficient quality. In practical applications, an image can be corrupted by a variety of sources, resulting in a degraded visual representation. Physical damage, such as marks or scratches can distort images. Furthermore, environmental factors such as lighting conditions, steam, fog, dust, and rain can have a significant impact

[1] This chapter is under review as **Yousef Salehi**, Ranjith Chiplunkar and Biao Huang, "Robust-to-Occlusion Machine Vision Model for Predicting Quality Variables with Slow-Rate Measurements", in *Computers and Chemical Engineering*, 2023.

on image quality by reducing clarity and introducing unwanted artifacts, leading to a decline in the visual information [44]. Understanding the common sources of image degradation is essential to mitigate such issues and employ appropriate methods for image restoration. In this regard, image inpainting involves restoring the completeness of an image by filling in missing or degraded regions. Image inpainting methods aim to reconstruct the missing or damaged regions in a way that seamlessly blends with the surrounding areas, creating visually coherent and convincing results [101]. Traditional image inpainting approaches rely on patch-based or texture synthesis methods to infer missing information based on the surrounding context [102]. However, recent advancements in deep learning, particularly with the use of CNN, have revolutionized the field of image inpainting [103, 104]. The deep learning methods have shown remarkable capabilities in capturing the high-level semantics and structures of images, enabling them to generate more realistic and visually satisfying inpainted results. Autoencoders have also proven to be effective tools for image inpainting tasks [105]. Autoencoders provide an end-to-end framework for inpainting, allowing them to directly learn to fill in missing regions without manual feature engineering. Autoencoders-based inpainted images are often visually coherent and semantically meaningful because they preserve important features and structures in the original image. Furthermore, autoencoders consider relationships between different regions to produce realistic and cohesive inpaintings. This study uses autoencoders to complete the degraded images caused by environmental factors.

Images and, therefore, their features are commonly available at a fast rate, but the desired process variable (output) is available only at a slow rate. A variety of model structures have been considered as predictive models where ARX model captures underlying dynamics while preserving simplicity of model structure [31]. The model can provide predictions of the output variable based on the input(s) and past output values. In this study, we develop a fast-rate ARX model for the output variable in the presence of multirate data with inclusion of images. To estimate the unknown

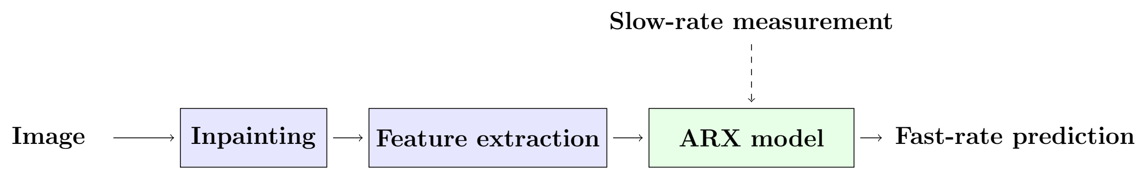


Figure 6.1: Flowchart of the proposed computer vision model

parameters of the model, we use the EM algorithm that is commonly used [106]. Figure 6.1 is a flowchart of the proposed method.

In summary, we propose a computer vision model for predicting slow-rate sampled variables at a fast rate. The main contributions of this chapter include:

1. Development of an autoencoder-based algorithm to inpaint images with relatively large missing portions.
2. Development of an ARX model using image data and slow-rate sampled output to predict the output at a fast rate.
3. Integration of the Rauch-Tung-Striebel (RTS) smoother into the KF framework, coupled with the utilization of the EM algorithm, thereby enabling estimation of unknown model parameters.
4. Evaluation of the proposed method through an experimental study.

6.2 Image Inpainting Using Autoencoder

Image inpainting techniques address the task of filling in relatively large missing regions in images. By training on sufficient amounts of available images, deep learning models can learn the underlying patterns, textures, and structures of images, enabling them to generate coherent and realistic content for the missing regions. Autoencoders have proven to be effective tools for image inpainting tasks. An autoencoder is a type of neural network architecture that is primarily designed to learn efficient representations of input data. In the context of image inpainting, an autoencoder can be trained to reconstruct complete and visually coherent images from partially degraded input images. An architecture of autoencoder is presented in Figure 6.2. The encoder part of the network learns to extract relevant features from the available image information, while the decoder fills in the missing or degraded regions based on the learned features. By training the autoencoder on a dataset of ground-truth images and their corresponding degraded versions, it learns to capture the underlying patterns and structures of the images, enabling it to reconstruct the missing regions. The use of autoencoders for image inpainting has demonstrated impressive results in various applications, including removing unwanted objects, restoring damaged areas, and filling in occluded regions.

In this work, the encoder consists of two convolutional layers where the first layer has 32 kernels of size (3, 3), and rectified linear unit (ReLU) activation. The second convolutional layer has 64 kernels of size (3, 3), and ReLU activation. After each convolutional layer, there is a max-pooling layer with a pool size of (2, 2).

The decoder mirrors the encoder architecture with two convolutional layers and two upsampling layers where the first and second convolutional layer has 64 and 32 kernels, respectively, of size (3, 3) followed by ReLU activation. After each convolutional layer, there is an upsampling layer with a size of (2,2). The output layer of the decoder has 3 kernels of size (3, 3) and sigmoid activation. It is designed to match the number of channels in the input image. An autoencoder is trained using the principle of

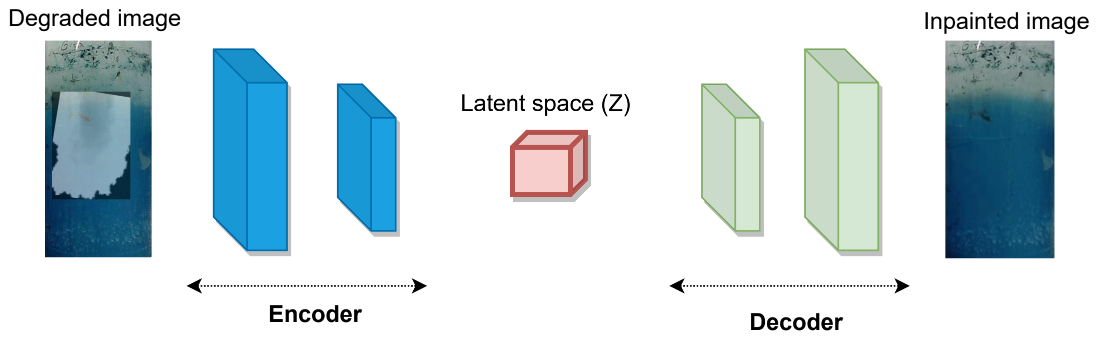


Figure 6.2: Architecture of autoencoder for image inpainting

minimizing the reconstruction loss. In other words, if we denote the ground-truth images as X and the reconstructed images as $\hat{X}$, the goal of autoencoder is to find the optimal parameters of the neural networks, denoted by θ_{ae}, that minimize the difference between the ground truth and reconstructed images. The reconstruction loss is commonly measured using the mean squared error (MSE) between X and $\hat{X}$:

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \tag{6.1}$$

where n is the number of image samples.

To update the network parameters θ_{ae}, we use gradient descent. The gradient of the reconstruction loss with respect to the parameters can be computed using back-propagation. Let E represent the error between X and $\hat{X}$:

$$E = X - \hat{X} \tag{6.2}$$

The gradients of the reconstruction loss with respect to the parameters can be computed as

$$\frac{\partial L_{\text{MSE}}}{\partial \theta_{ae}} = \frac{2}{n}\sum_{i=1}^{n} E_i \frac{\partial \hat{X}_i}{\partial \theta_{ae}} \tag{6.3}$$

$$\frac{\partial \hat{X}_i}{\partial \theta_{ae}} = \frac{\partial \hat{X}_i}{\partial Z_i}\frac{\partial Z_i}{\partial \theta_{ae}} \tag{6.4}$$

where Z represents the latent space of the autoencoder.

To complete the training process, we update the parameters using the gradient descent update rule:

$$\theta_{ae}^{\text{new}} = \theta_{ae}^{\text{old}} - \eta \frac{\partial L_{\text{MSE}}}{\partial \theta_{ae}} \tag{6.5}$$

where η is the learning rate.

6.3 ARX Predictive Model

In this section, a dual-rate ARX model is formulated where the color and texture features of the images are available at a faster rate, but only slow-rate measurements are available for the output. The objective is to predict the fast-rate values of the slow-rate sampled output. Thus, we have:

$$x_k = \sum_{n=1}^{n_a} a_n x_{k-n} + \sum_{m=1}^{M}\sum_{n=1}^{n_b} b_{m,n} u_{m,k-n} + w_k \tag{6.6}$$

$$y_{T_i} = x_{T_i} + v_{T_i} \tag{6.7}$$

In the problem formulation, x_k is the fast-rate output that is hidden. The index k, ranging from 1 to N_f, represents the instant of fast sampling. We have assumed M input variables (image features) are available and the term $\{u_{m,k}\}_{k=1,\cdots,N_f}$ denotes the mth input variable, which is available at every fast sampling time, *i.e.* $k \times t$, with $b_{m,n}$ being their corresponding coefficients. Further, $\{a_n\}_{n=1:n_a}$ is the coefficient for the nth lag of x. $\{y_{T_i}\}_{i=1,2,\cdots,N}$ represents the slow-rate output that is available only at every $T_i \times t$. N_f and N are the number of fast-rate images and slow-rate output data, respectively. The process noise w_k and measurement noise v_{T_i} are assumed to be independent, identically distributed (*i.i.d*) Gaussian sequences with mean zero and unknown variances σ_w^2 and σ_v^2, respectively.

For the notation simplicity, we rewrite (6.6) as:

$$x_k = \phi_k^T \theta + w_k \tag{6.8}$$

where,

$$\phi_k^T = [x_{k-1} \cdots x_{k-n_a}, u_{1,k-1:k-n_b}, u_{2,k-1:k-n_b} \cdots, u_{M,k-1:k-n_b}] \tag{6.9}$$

$$\theta = [a_1, a_2 \cdots a_{n_a}, b_{1,1:n_b}, \cdots, b_{M,1:n_b}]^T \tag{6.10}$$

Thus, a complete representation of the data is as follows:

$$D_{obs} = \{Y, U\} = \{y_{1:N}, u_{1,1:n}, u_{2,1:N_f}, \cdots, u_{M,1:N_f}\} \tag{6.11}$$

$$D_{hid} = X = \{x_{1:N_f}\} \tag{6.12}$$

$$\Theta = \{a_{1:n_a}, b_{1:M,1:n_b}, \sigma_w^2, \sigma_v^2\} \tag{6.13}$$

where D_{obs} is the observed data, D_{hid} is the hidden data and Θ denotes the unknown parameters. We use the EM algorithm to estimate the unknown parameters given the incomplete data.

6.3.1 Expectation-Maximization Algorithm

The EM algorithm performs MLE of the parameters of a model with missing or hidden data. Due to the missing data, maximizing the likelihood as such is not possible, and hence a lower bound to the likelihood function is maximized. The complete data likelihood which includes both D_{obs} and D_{hid} is expressed as follows:

$$p\left(Y, X, U|\Theta\right) = p(Y|X, U, \Theta)p(X|U, \Theta)p(U|\Theta) \tag{6.14}$$

In this work, U is independent of Θ, and hence $p(U|\Theta)$ is a constant and can be dropped from the above expression. As mentioned earlier, EM algorithm involves maximizing a lower bound of the above likelihood, and the details are presented in the subsequent sections.

6.3.2 Expectation Step

This step involves evaluating the expected value of the logarithm of the likelihood function given in section (6.14) with respect to the posterior distribution of the hidden variables. The resulting function is called the Q-function and is expressed as follows:

$$\begin{aligned} Q(\Theta|\Theta^h) =& \mathbb{E}_{D_{hid}|D_{obs},\Theta^h}\left[\log(p(D_{hid}, D_{obs}|\Theta))\right] \\ =& \mathbb{E}_{X|Y,U,\Theta^h}\left[\log(p(Y, X|\Theta, U))\right]. \end{aligned} \tag{6.15}$$

Here, Θ^h represents the estimated parameters at hth iteration of the EM algorithm.

With w_k and v_{T_i} being Gaussian white noise, the following results are obtained.

$$p\left(Y|X, U, \Theta\right) = \prod_{i=1}^{N} \mathcal{N}(y_{T_i}; x_{T_i}, \sigma_v^2) \tag{6.16}$$

$$p\left(X|U, \Theta\right) = \prod_{k=n^*+1}^{N_f} \mathcal{N}(x_k; \phi_k^T\theta, \sigma_w^2) \cdot \mathcal{N}(\tilde{x}; \mu_0, \Sigma_0) \tag{6.17}$$

Here, $n^* = \max(n_a, n_b)$ and $\mathcal{N}(x; \mu, \Sigma)$ represents a Gaussian distribution for variable x with mean μ and covariance Σ. Additionally, $\tilde{x} = x_{n^*:n^*+1-n_a}$ represents the initial conditions. The complete data log-likelihood can be written as follows:

$$\begin{aligned}\log p(Y, X|\Theta, U) = \quad & \underbrace{-N \log(\sigma_v) - \frac{1}{2}\sum_{i=1}^{N} \frac{(y_{T_i} - x_{T_i})^2}{\sigma_v^2}}_{q_1} \\ & \underbrace{-(N_f - n^*) \log(\sigma_w) - \frac{1}{2}\sum_{k=n^*+1}^{N_f} \frac{(x_k - \phi_k^T\theta)^2}{\sigma_w^2}}_{q_2} \\ & \underbrace{-\frac{1}{2}\log(\det(\Sigma_0)) - \frac{1}{2}(\tilde{x} - \mu_0)^T \Sigma_0^{-1} (\tilde{x} - \mu_0)}_{q_3} + c \end{aligned} \tag{6.18}$$

Here, c accounts for the proportionality constants of the involved distributions. Finally, the expected value of the above expression is taken which results in the following expressions. For simplicity, $\mathbb{E}_{X|Y,U,\Theta^h}[\cdot]$ is represented as $\langle \cdot \rangle$.

$$Q_1 = \langle q_1 \rangle = -N \log(\sigma_v) - \frac{1}{2}\sum_{i=1}^{N} \frac{y_{T_i}^2 - 2y_{T_i}\langle x_{T_i}\rangle + \langle x_{T_i}^2\rangle}{\sigma_v^2} \tag{6.19}$$

$$Q_2 = \langle q_2 \rangle = -(N_f - n^*) \log(\sigma_w) - \frac{1}{2}\sum_{k=n^*+1}^{N_f} \frac{\langle x_k^2\rangle - 2\langle x_k\phi_k^T\rangle\theta + \theta^T\langle \phi_k\phi_k^T\rangle\theta}{\sigma_w^2} \tag{6.20}$$

$$Q_3 = \langle q_3 \rangle = -\frac{1}{2}\log(\det(\Sigma_0)) - \frac{1}{2}\left(\mathrm{Tr}(\Sigma_0^{-1}\langle \tilde{x}\tilde{x}^T\rangle)\right) - \frac{1}{2}\left(-2\mu_0^T\Sigma_0^{-1}\langle \tilde{x}\rangle + \mu_0^T\Sigma_0^{-1}\mu_0\right) \tag{6.21}$$

Here $\mathrm{Tr}(\cdot)$ represents the trace operator. It can be observed that $Q = Q_1 + Q_2 + Q_3 + c$.

6.3.3 Evaluating the Posterior Distribution

It must be noted that the expressions in (6.19) - (6.21) contain the expected value of the hidden variables with respect to the posterior distribution of $p(X|Y, U, \Theta)$. This is essentially a state estimation problem with the state evolving according to an ARX model. It can be recast into the standard state estimation problem by converting the

ARX model into the state-space form as expressed by the following equations.

$$\begin{aligned} z_k &= Az_{k-1} + B\tilde{u}_{k-1} + \tilde{w}_k \\ y_{T_i} &= Cz_{T_i} + v_{T_i} \end{aligned} \tag{6.22}$$

The definitions of the states and parameters in (6.22) are given as below:

$$z_k = \begin{bmatrix} z_k^{(1)} \\ z_k^{(2)} \\ \vdots \\ z_k^{(n_a)} \end{bmatrix}_{n_a\times 1} = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-n_a+1} \end{bmatrix}_{n_a\times 1} \quad ; \quad \tilde{u}_{k-1} = \begin{bmatrix} u_{1,k-1:k-n_b} \\ u_{2,k-1:k-n_b} \\ \vdots \\ u_{M,k-1:k-n_b} \end{bmatrix} \tag{6.23}$$

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_{n_a} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & 0 \end{bmatrix}_{n_a\times n_a} \qquad C = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}_{1\times n_a}$$

$$B = \begin{bmatrix} b_{1,1:n_b} & b_{2,1:n_b} & \dots & b_{M,1:n_b} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \dots & \mathbf{0} \end{bmatrix}_{n_a\times(M\cdot n_b)} \tag{6.24}$$

Here, $\mathbf{0}$ represents a n_b dimensional row vector of zeros. The noise $\tilde{w}_k$ is Gaussian white noise defined as follows:

$$\tilde{w}_k \sim \mathcal{N}(\tilde{w}_k; [0]_{n_a\times 1}, \Sigma_w = \text{diag}([\sigma_w^2 \ 0 \ \dots \ 0])_{n_a\times n_a}) \tag{6.25}$$

With the above definitions, one can use the standard KF and Rauch-Tung-Striebel (RTS) smoother to obtain the posterior distribution of the states z_k [107]. The filtering and smoothing algorithms are summarized in the following set of equations.

Kalman filtering:

$$
\begin{aligned}
&\hat{z}_{k|k-1} = A\hat{z}_{k-1|k-1} + B\tilde{u}_{k-1} \\
&\hat{P}_{k|k-1} = A\hat{P}_{k-1|k-1}A^T + \Sigma_w \\
\\
&\text{if } y_k \text{ is available,} \\
&K_k = \hat{P}_{k|k-1}C^T(C\hat{P}_{k|k-1}C^T + \sigma_v^2)^{-1} \\
&\hat{z}_{k|k} = \hat{z}_{k|k-1} + K_k(y_k - C\hat{z}_{k|k-1}) \\
&\hat{P}_{k|k} = (I - K_kC)\hat{P}_{k|k-1} \\
\\
&\text{if } y_k \text{ is unavailable,} \\
&\hat{z}_{k|k} = \hat{z}_{k|k-1} \\
&\hat{P}_{k|k} = \hat{P}_{k|k-1}
\end{aligned}
\tag{6.26}
$$

RTS smoothing:

$$
\begin{aligned}
&J_k = \hat{P}_{k|k}A^T\hat{P}_{k+1|k}^{-1} \\
&\hat{z}_{k|N} = \hat{z}_{k|k} + J_k(\hat{z}_{k+1|N} - \hat{z}_{k+1|k}) \\
&\hat{P}_{k|N} = \hat{P}_{k|k} + J_k(\hat{P}_{k+1|N} - \hat{P}_{k+1|k})J_k^T \\
&V_{k|N} = J_{k-1}\hat{P}_{k|N}
\end{aligned}
\tag{6.27}
$$

Here, $\hat{z}$, $\hat{P}$, K, and V represent the estimated mean of the states, the estimated covariance matrix of the states, Kalman gain, and covariance of z_k and z_{k-1} respectively. It must be noted that the identification of the ARX model, also discussed in [31] and [108], uses a modified KF to estimate the hidden states. The modified KF is equivalent to the standard KF implementation after recasting the ARX model into a state-space model. Unlike the approach taken in [31] and [108], we have integrated the RTS smoothing step into the filtering process to refine the state estimation results.

Finally, with the distribution of the states estimated as given in (6.27), the expected value of the quantities related to the hidden variables in (6.19) - (6.21) are estimated as given in the following equations.

- It can be noted that x_k is nothing but the first element of z_k. Thus, the moments of x_k are represented as follows:

$$\begin{aligned}\langle x_k\rangle &= \hat{z}^{(1)}_{k|N}\\ \langle x_k^2\rangle &= \hat{P}^{(1,1)}_{k|N} + \langle x_k\rangle^2\end{aligned} \tag{6.28}$$

- Form the definition of ϕ_k in (6.9), it can be noted that

$$\phi_k = \begin{bmatrix} z_{k-1} \\ \tilde{u}_{k-1}\end{bmatrix} \tag{6.29}$$

Thus, we have the following results for expectations of quantities related to ϕ_k.

$$\begin{aligned}\langle x_k\phi_k^T\rangle &= \begin{bmatrix}\langle x_k z_{k-1}^T\rangle & \langle x_k\rangle\tilde{u}_{k-1}\end{bmatrix}\\ \langle x_k z_{k-1}^T\rangle &= \hat{V}^{(1,:)}_{k|N} + \langle x_k\rangle \hat{z}^T_{k-1|N}\\ \langle \phi_k\phi_k^T\rangle &= \begin{bmatrix}\langle z_{k-1}z_{k-1}^T\rangle & \hat{z}_{k-1|N}\tilde{u}_{k-1}^T\\ \tilde{u}_{k-1}\hat{z}^T_{k-1|N} & \tilde{u}_{k-1}\tilde{u}_{k-1^T}\end{bmatrix}\\ \langle z_{k-1}z_{k-1}^T\rangle &= \hat{P}_{k-1|N} + \hat{z}_{k-1|N}\hat{z}^T_{k-1|N}\end{aligned} \tag{6.30}$$

Substituting the above expressions in (6.19) - (6.21) completes the E-step of the algorithm.

6.3.4 Maximization Step

In the M-step, the Q-function is maximized with respect to the parameters of the model as represented below.

$$\Theta^{h+1} = \arg\max_{\Theta} Q(\Theta|\Theta^h) \tag{6.31}$$

The derivation of the M-step to obtain Θ^{h+1} is summarized in the following points.

- **Updating $\boldsymbol{\sigma_v^2}$:** Since σ_v^2 appears only in Q_1,

$$\frac{\partial Q_1}{\partial \sigma_v} = 0 \implies \sigma_v^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_{T_i}^2 - 2y_{T_i}\langle x_{T_i}\rangle + \langle x_{T_i}^2\rangle\right) \tag{6.32}$$

- **Updating $\boldsymbol{\sigma_w^2}$:** Similarly, the expression for σ_w^2 is obtained as

$$\sigma_w^2 = \frac{1}{N_f - n^*}\sum_{k=n^*+1}^{N_f}\left(\langle x_k^2\rangle - 2\langle x_k\phi_k^T\rangle\theta + \theta^T\langle\phi_k\phi_k^T\rangle\theta\right) \tag{6.33}$$

- **Updating $\boldsymbol{\theta}$:** It can be observed that θ appears only in $Q2$. Thus, the following result is obtained.

$$\frac{\partial Q_2}{\partial \theta} = 0 \implies \theta = \left(\sum_{n^*+1}^{N}\langle\phi_k\phi_k^T\rangle\right)^{-1}\left(\sum_{n^*+1}^{N}\langle\phi_k x_k\rangle\right) \tag{6.34}$$

In addition to the model parameters, the parameters of the distribution of the initial states $\tilde{x}$ can also be estimated. This is achieved by differentiating Q_3 with respect to μ_0 and Σ_0 which results in the following expressions

$$\begin{aligned}\mu_0 &= \langle\tilde{x}\rangle \\ \Sigma_0 &= \langle\tilde{x}\tilde{x}^T\rangle - 2\mu\langle\tilde{x}^T\rangle + \mu_0\mu_0^T\end{aligned} \tag{6.35}$$

It can be observed that $\tilde{x}$ is nothing but z_{n^*}. All the expectations in (6.31) - (6.35) are evaluated as given in 6.3.3.

6.4 Experimental Verification

This section presents an assessment of the proposed method through the estimation of interface level in a lab-scale PSC. The experimental setup is illustrated in Figure 6.3 where oil and distilled water are used to create an insoluble interface. The figure depicts a main cell and two side tanks containing water and oil. The pumps located at the bottom of the setup enable water and oil to be pumped into and out of the main cell. Due to its lower density compared to water, the oil forms a top layer analogous to the froth layer found in a typical indutria-scale PSC. Conversely, the distilled water

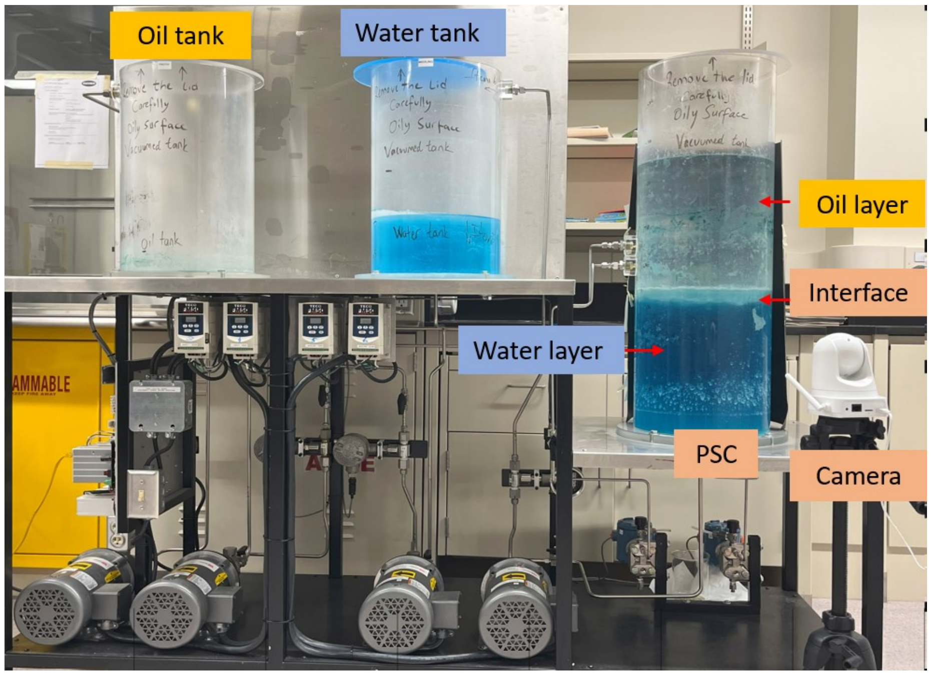


Figure 6.3: PSC lab experimental setup

is analogous to the middlings layer of the PSC. The level of the interface between the two phases can be regulated by controlling the inlet and outlet flows of both water and oil. There is a camera installed adjacent to the PSC tank to capture color images at a resolution of 1280×720. To control the process and record data, MATLAB and OPTO22 are used as the implementation platforms. In this experiment, the PSC images were recorded every five seconds for two and a half consecutive hours, resulting in 1800 images. To avoid unnecessary processing and reduce computation time, the region of interest (ROI) is limited to the main cell only. Additionally, the cropped images are resized to 530×260 pixels to enhance computational efficiency. The recorded images have relatively small stains and noise, so they are used as the ground truth. Afterwards, all the images were occluded by relatively large artificial masks with a combination of dark and bright colors that mimicked real-life objects affecting images. A series of ground-truth and occluded images of the experiments are shown in Figure 6.4. The color and texture features of the training images are significantly altered by the occlusions, making it difficult to visualize the interface

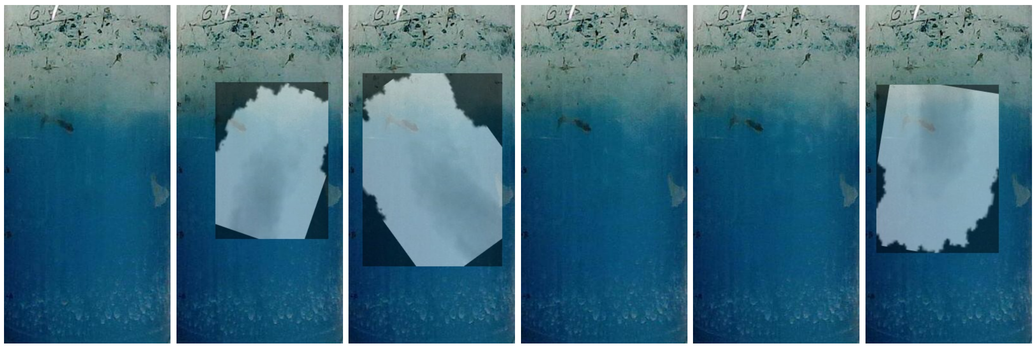
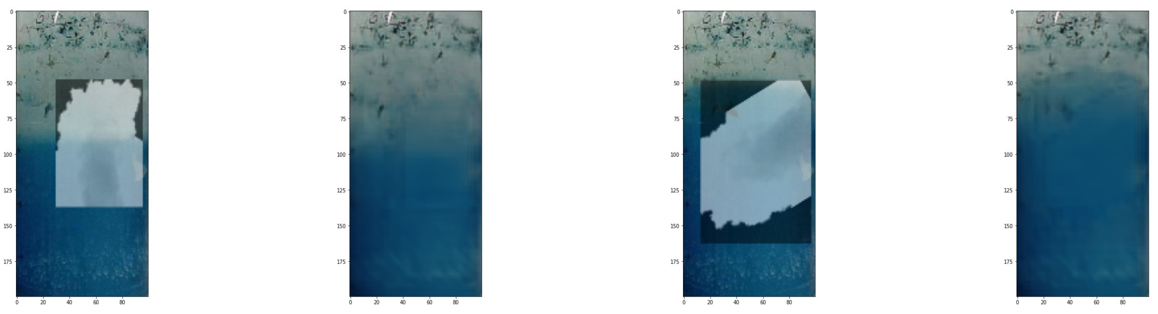


Figure 6.4: Six sequential images of PSC with distortion in some images

level. In the testing phase, the first objective is to inpaint the testing images so that clear images of the PSC are obtained. The ultimate goal is to predict interface level at a fast rate from the inpainted images.

6.4.1 Image Inpainting Results

In this subsection, results of the proposed image inpainting algorithm are presented. A total of 1350 pairs of ground-truth and degraded images were used for training the autoencoder, and 450 pairs independent of the training images were used for testing. By training the autoencoder on the partially occluded images along with their corresponding ground-truth images, we enable the model to learn meaningful representations and reconstruct images that closely resemble the original one. The training process was implemented by Keras in Python. Two typical examples of the validations results are shown in Figure 6.5. As can be seen, the algorithm performs satisfactorily in filling in the degraded regions of images. Thus, the algorithm can indeed recover the relatively large missing areas. Moreover, the transitions between inpainted regions and surrounding areas in the inpainted images are smooth. In addition to recovering the missing regions, the inpainting algorithm also removes the minor noises and stains on the image. The experimental outcomes highlight the potential of autoencoder-based image inpainting as a promising solution for image inpainting tasks. The algorithm, however, cannot recover an entirely blocked image, as expected. To further demonstrate the effectiveness of the inpainting algorithm, we have conducted a comparison of the mean values of three color features, *i.e.* red, green, and blue (RGB) values, for testing images where 15% of images are randomly degraded and the rest remain intact. Namely, we compare between contaminated images and their ground truth. The obtained results are presented in Figure 6.6. The visual evidence indicates that the inpainted images closely resemble their corresponding ground-truth images in terms of color characteristics. In contrast, the non-inpainted images show noticeable variations in the color features when compared to their corresponding ground-truth images.



(a) (b) (c) (d)

Figure 6.5: Sample inpainting results. (a) Degraded. (b) Inpainted of a. (c) Degraded. (d) Inpainted of c.

6.4.2 Interface Prediction

As described earlier, the primary objective of this paper is to predict the slow-measured variables at a fast-rate using an ARX model and image features. RGB values of the images were used as inputs to the dual-rate ARX model. As in the inpainting algorithm, 1350 images were used for training and the rest for testing. We manually labeled the corresponding interface level at every five sampling instants. This slow-rate labeling mimics the lab analysis in industrial settings. Gaussian noise with mean zero and standard deviation of 0.025 was added to the slow-rate measurements. The model uses mean of RGB values of each image as image features. Using the slow-rate interface level labels, fast-rate image features, and applying the EM

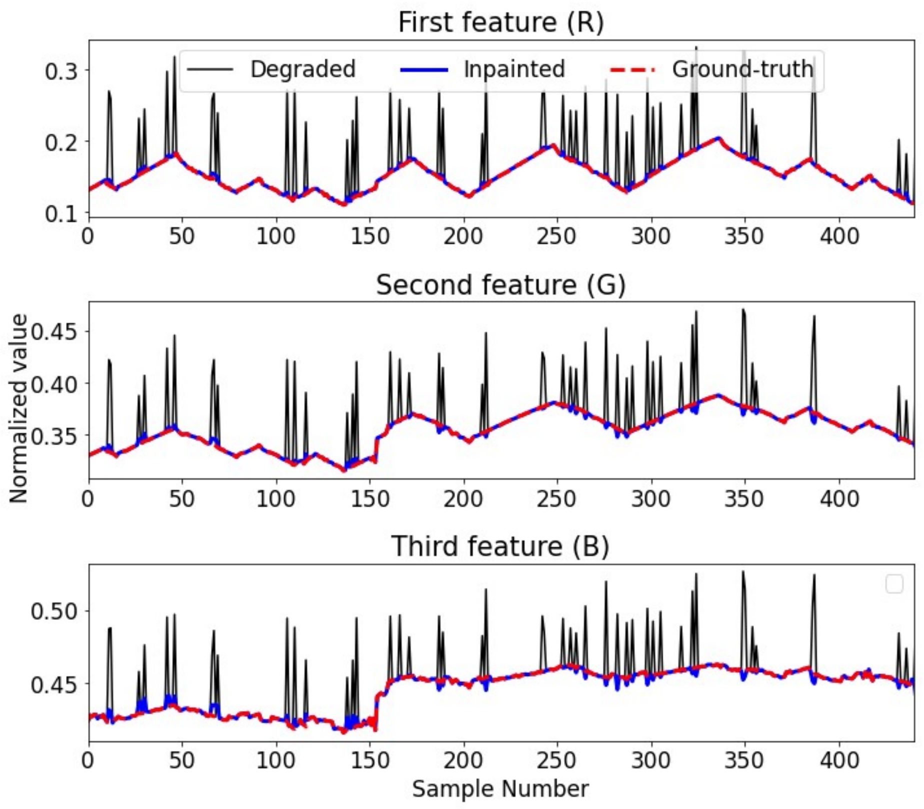


Figure 6.6: Inpainting results for the sequence of validation images

algorithm introduced in 6.3, the following ARX model was obtained.

$$\hat{x}_k = 0.90\hat{x}_{k-1} - 0.85u_{1,k} - 1.03u_{1,k-1} + 1.46u_{1,k-2} + 1.04u_{2,k} - 1.24u_{2,k-1} \\ + 1.16u_{2,k-2} - 0.18u_{3,k} + 1.46u_{3,k-1} - 1.83u_{3,k-2} \tag{6.36}$$

To evaluate the performance of the prediction model, we randomly introduced degradation in 15% of the testing images, while the remaining 85% are left unaffected. The model is used for predicting the PSC interface level at a fast rate, as shown in Figure 6.7. A promising performance can be seen in the tracking of interface level by the proposed ARX model with inpainting. The results have also been compared with the same ARX modeling algorithm but without inpainting the degraded images. We observed that lack of the inpainting step significantly degraded the prediction accuracy. The code was run 50 times with different noise realizations. Table 6.1 presents the average RMSE and coefficient of determination (R^2) values for

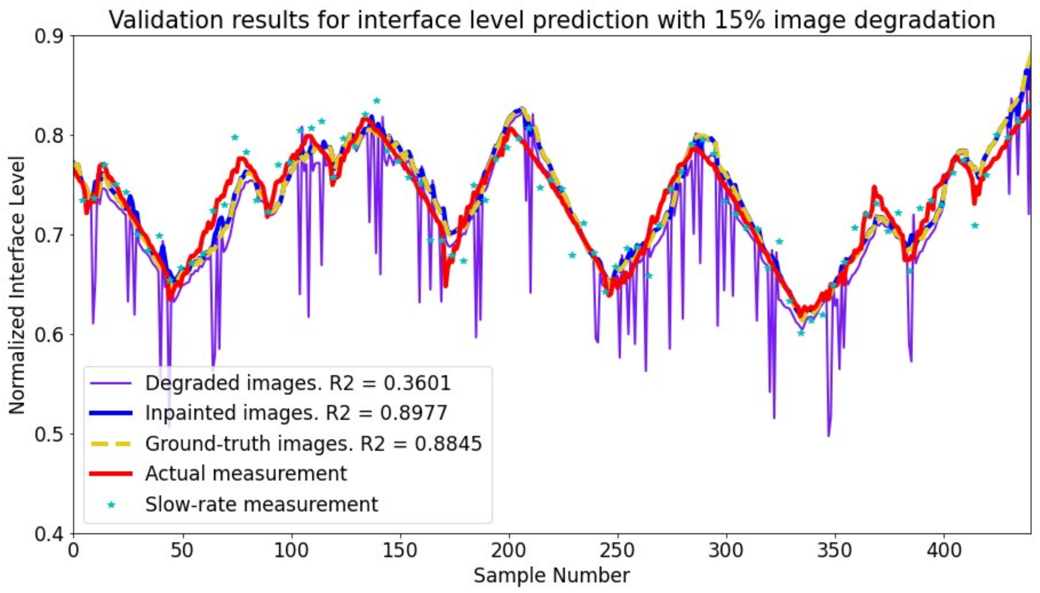


Figure 6.7: Validation results for interface level prediction in PSC with 15% degraded images

the competing algorithms discussed. In the presence of 15% degraded images, it can be observed that the proposed algorithm achieves an accuracy of 87.78%, whereas excluding the inpainting process results in an accuracy of 38.01%. We have also compared the results when the ground-truth images are used as the testing images directly. It is evident from Table 6.1 that the results are almost the same. The slight superiority of the ARX with inapinting is due to removing the small noises from the ground-truth images as the ground-truth images in this experiment are not flawless. To further evaluate the performance, we considered a larger percentage of degraded testing images, which is 25%. As a result, the performance of the model without inpainting is drastically affected. However, the model with inpainting images still shows satisfactory robustness against image deterioration, highlighting the importance of inpainting. A comparison of two different image degradation ratios shows that as the image degradation ratio increases, the prediction accuracy decreases but the one with inpainting shows more robustness.

Table 6.1: Validation results for the fast-rate prediction

Image degradation ratio	Metric	ARX with ground-truth	ARX with inpainting	ARX without inpainting
15%	RMSE	0.0179	0.0177	0.0421
	R^2	0.8722	0.8778	0.3801
25%	RMSE	0.0179	0.0240	0.1612
	R^2	0.8722	0.7853	0.0901

6.5 Conclusion

In this work, we proposed a computer vision model that enables fast-rate prediction of variables that are measured only at a slow rate. Considering that images can be affected by various sources including environmental factors, we used an autoencoder algorithm to inpaint images with relatively large missing portions. Then, the inpainted image features were extracted and utilized as input variables for an ARX multirate model. The unknown parameters of the model were estimated using an EM algorithm coupled with the KF and RTS smoother. Experimental validation conducted on a lab-scale PSC proved the effectiveness of the image inpainting process. Furthermore, the proposed vision model was able to predict the interface level accurately.

Chapter 7

Concluding Remarks

This chapter outlines the conclusions drawn from the proposed methods and their implementations in the experimental case studies. Additionally, potential future directions of research in this domain are also outlined.

7.1 Conclusion

In conclusion, this thesis develops four models for predicting quality variables while taking important practical challenges into account. In the first two contributions, predictive models are developed using regular sensor data, while the last two use image data. Both types of data are available at a fast-rate and used for building the respective predictive models. Accordingly, important practical challenges are addressed to make them useful for actual applications.

The first contribution focuses on formulating a statistical multirate model identification problem, considering integration periods, time-delays, and switching operating points. By employing a non-parametric distribution, the model captures unknown and time-varying integration periods, delays, and operating modes with no requirement of prior knowledge. This model enables fast-rate offline and online prediction of the slow-rate measured variables. The accuracy of this model has been verified by experiments on a hybrid three-tank system.

To incorporate the transition between operating modes, the second contribution extends the first one by developing a global LPV model. The model also enhances robustness to outlying output observations by utilizing the t-distribution and considers uncertainties of unknown parameters using the VB algorithm. This model shows promising results in experiments.

The third contribution focuses on developing a computer vision model that utilizes image features to predict the quality variables in batch processes. The proposed model uses a modified KF to restore the degraded images caused by factors like lighting and noise. Further, a robust-to-outlier EM algorithm for parameter estimation is presented. Validation of the proposed vision model on a flotation column demonstrated its effectiveness in predicting the cumulative bitumen content over each batch.

The last contribution is about the development of a computer vision model that enables fast-rate prediction of quality variables. To overcome the impact of environmental factors on images, an image inpainting algorithm is utilized to restore significant missing regions in the images. The restored images along with the slow-rate sampled measurements, are then used in conjunction with the EM algorithm to construct a hybrid vision-statistical predictive model, offering an efficient approach for quality variable prediction. An experimental evaluation confirms the method's reliability.

Overall, the contributions of this thesis have advanced the field of predictive modeling for quality variables by addressing various key practical problems and incorporating novel techniques. The proposed models and methodologies have demonstrated promising performance in their respective applications, paving the way for further advancements in process optimization and control.

7.2 Future Scope

7.2.1 Advanced Image Inpainting Techniques

It is important to take into account additional image degradation sources that may be encountered in industrial applications, such as image reflections. In industrial settings, reflections can often occur due to the presence of shiny or reflective surfaces, which can adversely affect the quality and accuracy of captured images. In addition, there are several other common sources of image degradation encounters in various applications. These include blurred images, low resolution images and color distortions.

To address such problems, it is suggested to explore more advanced image restoration techniques like GANs as they have shown promising results in generating superior and realistic inpainted outputs. Through adversarial training, GANs encourage the generator to create high-quality inpainted images that align visually with the surrounding context. Compared to traditional methods, GAN-based inpainting techniques excel in capturing intricate details, preserving textures, and delivering visually appealing results, significantly improving the quality and accuracy of inpainted image.

7.2.2 Video Inpainting

It is recommended to incorporate temporal information along with the spatial information during the process of image inpainting. This is known as video inpainting, and it leverages temporal data to improve the inpainting results, especially when most of the image is missing. In such cases, temporal information becomes invaluable for restoring the missing content. Relying solely on spatial information may not be sufficient to achieve effective inpainting results.

7.2.3 Classification Approach

In future work, it is recommended to consider developing classification models for assessing quality variables, considering the limited availability of labeled quality data in practical applications. By categorizing quality variables into classes or levels, classification models can provide acceptable practical predictions based on the available data. To enhance the performance of these models, exploring more advanced techniques such as ensemble learning, deep learning, and transfer learning can be beneficial.

References

[1] S. Khatibisepehr and B. Huang, "Dealing with irregular data in soft sensors: Bayesian method and comparative study," *Industrial & Engineering Chemistry Research*, vol. 47, no. 22, pp. 8713–8723, 2008.

[2] V. Prasad, M. Schley, L. P. Russo, and B. W. Bequette, "Product property and production rate control of styrene polymerization," *Journal of Process Control*, vol. 12, no. 3, pp. 353–372, 2002.

[3] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Computers & chemical engineering*, vol. 33, no. 4, pp. 795–814, 2009.

[4] P. Facco, F. Doplicher, F. Bezzo, and M. Barolo, "Moving average pls soft sensor for online product quality estimation in an industrial batch polymerization process," *Journal of Process Control*, vol. 19, no. 3, pp. 520–529, 2009.

[5] S. B. Chitralekha and S. L. Shah, "Support vector regression for soft sensor design of nonlinear processes," in *18th Mediterranean Conference on Control and Automation, MED'10*, IEEE, 2010, pp. 569–574.

[6] Y. Liu, T. Chen, and J. Chen, "Auto-switch gaussian process regression-based probabilistic soft sensors for industrial multigrade processes with transitions," *Industrial & Engineering Chemistry Research*, vol. 54, no. 18, pp. 5037–5047, 2015.

[7] X. Yuan, S. Qi, and Y. Wang, "Stacked enhanced auto-encoder for data-driven soft sensing of quality variable," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7953–7961, 2020.

[8] X. Yuan, L. Li, Y. A. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based lstm for industrial soft sensor model development," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4404–4414, 2020.

[9] D. Wang, J. Liu, and R. Srinivasan, "Data-driven soft sensor approach for quality prediction in a refining process," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 11–17, 2010. DOI: 10.1109/TII.2009.2025124.

[10] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[11] N. B. Khazaei, T. Tavakoli, H. Ghassemian, M. H. Khoshtaghaza, and A. Banakar, "Applied machine vision and artificial neural network for modeling and controlling of the grape drying process," *Computers and electronics in agriculture*, vol. 98, pp. 205–213, 2013.

[12] M Massinaei, A Jahedsaravani, E Taheri, and J Khalilpour, "Machine vision based monitoring and analysis of a coal column flotation circuit," *Powder Technology*, vol. 343, pp. 330–341, 2019.

[13] D. Zhang, X. Gao, and W. Qi, "Soft sensor of iron tailings grade based on froth image features for reverse flotation," *Transactions of the Institute of Measurement and Control*, p. 01 423 312 221 096 450, 2022.

[14] O. Dogru, R. Chiplunkar, and B. Huang, "Reinforcement learning with constrained uncertain reward function through particle filtering," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 7491–7499, 2022. DOI: 10.1109/TIE.2021.3099234.

[15] C. Aldrich, E. Avelar, and X. Liu, "Recent advances in flotation froth image analysis," *Minerals Engineering*, vol. 188, p. 107 823, 2022.

[16] M Zarie, A Jahedsaravani, and M Massinaei, "Flotation froth image classification using convolutional neural networks," *Minerals Engineering*, vol. 155, p. 106 443, 2020.

[17] R. Gopaluni, "A particle filter approach to identification of nonlinear processes under missing observations," *The Canadian Journal of Chemical Engineering*, vol. 86, no. 6, pp. 1081–1092, 2008.

[18] F. Ding, P. X. Liu, and H. Yang, "Parameter identification and intersample output estimation for dual-rate systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 4, pp. 966–975, 2008.

[19] J. Deng and B. Huang, "Identification of nonlinear parameter varying systems with missing output data," *AIChE Journal*, vol. 58, no. 11, pp. 3454–3467, 2012.

[20] N. Lu, Y. Yang, F. Gao, and F. Wang, "Multirate dynamic inferential modeling for multivariable processes," *Chemical Engineering Science*, vol. 59, no. 4, pp. 855–864, 2004.

[21] M. Kano, N. Showchaiya, S. Hasebe, and I. Hashimoto, "Inferential control of distillation compositions: Selection of model and control configuration," *Control Engineering Practice*, vol. 11, no. 8, pp. 927–933, 2003.

[22] H. Raghavan, A. K. Tangirala, R. B. Gopaluni, and S. L. Shah, "Identification of chemical processes with irregular output sampling," *Control Engineering Practice*, vol. 14, no. 5, pp. 467–480, 2006.

[23] H. Lin and S. Sun, "An overview of multirate multisensor systems: Modelling and estimation," *Information Fusion*, vol. 52, pp. 335–343, 2019.

[24] X. Yang and S. Yin, "Robust global identification and output estimation for LPV dual-rate systems subjected to random output time-delays," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2876–2885, 2017.

[25] W. Xiong, X. Yang, B. Huang, and B. Xu, "Multiple-model based linear parameter varying time-delay system identification with missing output data using an expectation-maximization algorithm," *Industrial & Engineering Chemistry Research*, vol. 53, no. 27, pp. 11 074–11 083, 2014.

[26] X. Yang and H. R. Karimi, "Identification of LTI time-delay systems with missing output data using GEM algorithm," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[27] J. Chen, J. Ma, Y. Liu, and F. Ding, "Identification methods for time-delay systems based on the redundant rules," *Signal Processing*, vol. 137, pp. 192–198, 2017.

[28] L. Xie, H. Yang, and B. Huang, "FIR model identification of multirate processes with random delays using EM algorithm," *AIChE Journal*, vol. 59, no. 11, pp. 4124–4132, 2013.

[29] Y. Zhao, A. Fatehi, and B. Huang, "A data-driven hybrid ARX and markov chain modeling approach to process identification with time-varying time delays," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4226–4236, 2016.

[30] Y. Zhao, A. Fatehi, and B. Huang, "Robust estimation of ARX models with time varying time delays using variational Bayesian approach," *IEEE transactions on cybernetics*, vol. 48, no. 2, pp. 532–542, 2017.

[31] J. Chen, B. Huang, F. Ding, and Y. Gu, "Variational Bayesian approach for arx systems with missing observations and varying time-delays," *Automatica*, vol. 94, pp. 194–204, 2018.

[32] X. Liu and X. Liu, "An improved identification method for a class of time-delay systems," *Signal Processing*, vol. 167, p. 107 308, 2020.

[33] F. Wu and K. M. Grigoriadis, "LPV systems with parameter-varying time delays: Analysis and control," *Automatica*, vol. 37, no. 2, pp. 221–229, 2001.

[34] Y. Guo, Y. Zhao, and B. Huang, "Development of soft sensor by incorporating the delayed infrequent and irregular measurements," *Journal of Process Control*, vol. 24, no. 11, pp. 1733–1739, 2014.

[35] A. Fatehi and B. Huang, "Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay," *Journal of Process Control*, vol. 53, pp. 15–25, 2017.

[36] A. Fatehi and B. Huang, "State estimation and fusion in the presence of integrated measurement," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2490–2499, 2017.

[37] Y. Salehi, A. Fatehi, and M. A. Nayebi, “State estimation of slow-rate integrated measurement systems in the presence of parametric uncertainties,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3983–3991, 2018.

[38] F. Yaghoobi, A. Fatehi, and M. Moghaddasi, “Bayesian state estimation in the presence of slow-rate integrated measurement,” *International Journal of Systems Science*, pp. 1–17, 2020.

[39] E. R. Khosro and A. Fatehi, “Design of state and output feedback pole placement controller in the presence of slow-rate integrated measurement,” *Journal of Process Control*, vol. 85, pp. 214–226, 2020.

[40] A. Kheirandish, A. Fatehi, and M. S. Gheibi, “Identification of slow-rate integrated measurement systems using expectation–maximization algorithm,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9477–9484, 2020.

[41] Y. Fang and M. K. Jeong, “Robust probabilistic multivariate calibration model,” *Technometrics*, vol. 50, no. 3, pp. 305–316, 2008.

[42] X. Jin and B. Huang, “Robust identification of piecewise/switching autoregressive exogenous process,” *AIChE journal*, vol. 56, no. 7, pp. 1829–1844, 2010.

[43] J. Zhu, Z. Ge, and Z. Song, “Robust semi-supervised mixture probabilistic principal component regression model development and application to soft sensors,” *Journal of Process Control*, vol. 32, pp. 25–37, 2015.

[44] P. Milanfar, “A tour of modern image filtering: New insights and methods, both practical and theoretical,” *IEEE signal processing magazine*, vol. 30, no. 1, pp. 106–128, 2012.

[45] H. Tang, G. Hu, W. He, and Q. Tu, “Improved anomaly detection based on image reconstruction and global template features for industrial products,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2166, 2022, p. 012 062.

[46] B. Rasti, Y. Chang, E. Dalsasso, L. Denis, and P. Ghamisi, “Image restoration for remote sensing: Overview and toolbox,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 201–230, 2022. DOI: 10.1109/MGRS.2021.3121761.

[47] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, pp. 1–12, 2019.

[48] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[49] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions.* John Wiley & Sons, 2007, vol. 382.

[50] X. Liu and X. Yang, "Robust variational inference for LPV dual-rate systems with randomly delayed outputs," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[51] R. Chiplunkar and B. Huang, "Modeling and Bayesian inference for processes characterized by abrupt variations," *Journal of Process Control*, vol. 127, p. 103 001, 2023.

[52] D. Ostwald, E. Kirilina, L. Starke, and F. Blankenburg, "A tutorial on variational Bayes for latent linear stochastic time-series models," *Journal of Mathematical Psychology*, vol. 60, pp. 1–19, 2014.

[53] S. Roshany-Yamchi, M. Cychowski, R. R. Negenborn, B. De Schutter, K. Delaney, and J. Connell, "Kalman filter-based distributed predictive control of large-scale multi-rate systems: Application to power networks," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 27–39, 2011.

[54] P Mohanaiah, P Sathyanarayana, and L GuruKumar, "Image texture feature extraction using GLCM approach," *International journal of scientific and research publications*, vol. 3, no. 5, pp. 1–5, 2013.

[55] D. Kumar *et al.*, "Feature extraction and selection of kidney ultrasound images using glcm and pca," *Procedia Computer Science*, vol. 167, pp. 1722–1731, 2020.

[56] Y. Liu, H. Pu, and D.-W. Sun, "Efficient extraction of deep image features using convolutional neural network (cnn) for applications in detecting and analysing complex food matrices," *Trends in Food Science & Technology*, vol. 113, pp. 193–204, 2021.

[57] M. Paoletti, J. Haut, J Plaza, and A Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[58] X. Yang and X. Yang, "Local identification of LPV dual-rate system with random measurement delays," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1499–1507, 2017.

[59] Y. Gu, J. Liu, X. Li, Y. Chou, and Y. Ji, "State space model identification of multirate processes with time-delay using the expectation maximization," *Journal of the Franklin Institute*, vol. 356, no. 3,

[60] X. Jin, S. Wang, B. Huang, and F. Forbes, "Multiple model based LPV soft sensor development with irregular/missing process output measurement," *Control Engineering Practice*, vol. 20, no. 2, pp. 165–172, 2012.

[61] X. Yang, H. Sun, L. Bai, and X. Liu, "Identification of linear parameter varying system with dual-rate sampled data and uncertain measurement delay," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, IEEE, 2016, pp. 95–99.

[62] X. Yang and X. Yang, "Local identification of LPV dual-rate system with random measurement delays," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1499–1507, 2017.

[63] Y. Guo and B. Huang, "State estimation incorporating infrequent, delayed and integral measurements," *Automatica*, vol. 58, pp. 32–38, 2015.

[64] R. Han, Y. Salehi, B. Huang, and V. Prasad, "State estimation for multirate measurements in the presence of integral term and variable delay," *IEEE Transactions on Control Systems Technology*, pp. 1–11, 2020. DOI: 10.1109/TCST.2020.3039737.

[65] Y. Liu, Z. Wang, and D. Zhou, "State estimation and fault reconstruction with integral measurements under partially decoupled disturbances," *IET Control Theory & Applications*, vol. 12, no. 10, pp. 1520–1526, 2018.

[66] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Industrial & Engineering Chemistry Research*, vol. 52, no. 10, pp. 3543–3562, 2013.

[67] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[68] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[69] H. Nakada, K. Takaba, and T. Katayama, "Identification of piecewise affine systems based on statistical clustering technique," *Automatica*, vol. 41, no. 5, pp. 905–913, 2005.

[70] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, "Online adaptive parameter identification and state-of-charge coestimation for lithium-polymer battery cells," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2053–2061, 2013.

[71] O. Cappé and E. Moulines, "On-line expectation–maximization algorithm for latent data models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 593–613, 2009.

[72] Y. Salehi and B. Huang, "Offline and online parameter learning for switching multirate processes with varying delays and integrated measurements," *IEEE Transactions on Industrial Electronics*, 2021.

[73] A. Golabi, N. Meskin, R. Toth, and J. Mohammadpour, "A Bayesian approach for LPV model identification and its application to complex processes," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2160–2167, 2017. DOI: 10.1109/TCST.2016.2642159.

[74] D. Li, S. L. Shah, and T. Chen, "Identification of fast-rate models from multirate data," *International Journal of Control*, vol. 74, no. 7, pp. 680–689, 2001.

[75] X. Liu, X. Yang, P. Zhu, and Y. Wang, "Robust multimodel identification of LPV systems with missing observations based on t-distribution," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[76] X. Liu, X. Yang, and M. Yu, "Identification of switched FIR systems with random missing outputs: A variational Bayesian approach," *Journal of the Franklin Institute*, vol. 358, no. 1, pp. 1136–1151, 2021.

[77] R. Han, Y. Salehi, B. Huang, and V. Prasad, "Parameter estimation for nonlinear systems with multirate measurements and random delays," *AIChE Journal*, e17327, 2021.

[78] S. Gao, S. Zhao, X. Luan, and F. Liu, "Intelligent state estimation for continuous fermenters using variational Bayesian learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8429–8437, 2021. DOI: 10.1109/TII.2021.3057421.

[79] W. R. Jacobs, T. Baldacchino, T. Dodd, and S. R. Anderson, "Sparse Bayesian nonlinear system identification using variational inference," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4172–4187, 2018. DOI: 10.1109/TAC.2018.2813004.

[80] K. Kasperski, "A review of properties and treatment of oil sands tailings," *AOSTRA Journal of Research (Alberta Oil Sands Technology Authority);(Canada)*, vol. 8, no. 1, 1992.

[81] K. Zhou, S. G. Sontti, J. Zhou, P. Esmaeili, and X. Zhang, "Microbubble-enhanced bitumen separation from tailing slurries with high solid contents," *Industrial & Engineering Chemistry Research*, vol. 61, no. 47, pp. 17 327–17 341, 2022.

[82] K. Popli, V. Maries, A. Afacan, Q. Liu, and V. Prasad, "Development of a vision-based online soft sensor for oil sands flotation using support vector regression and its application in the dynamic monitoring of bitumen extraction," *The Canadian Journal of Chemical Engineering*, vol. 96, no. 7, pp. 1532–1540, 2018.

[83] P. Holtham and K. Nguyen, "On-line analysis of froth surface in coal and mineral flotation using jkfrothcam," *International Journal of Mineral Processing*, vol. 64, no. 2-3, pp. 163–180, 2002.

[84] M. Ai, Y. Xie, Z. Tang, J. Zhang, and W. Gui, "Two-stream deep feature-based froth flotation monitoring using visual attention clues," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–14, 2020.

[85] H. Zhang, Z. Tang, Y. Xie, Q. Chen, X. Gao, and W. Gui, “Feature reconstruction-regression network: A light-weight deep neural network for performance monitoring in the froth flotation,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8406–8417, 2020.

[86] C. Aldrich, C Marais, B. Shean, and J. Cilliers, “Online monitoring and control of froth flotation systems with machine vision: A review,” *International Journal of Mineral Processing*, vol. 96, no. 1-4, pp. 1–13, 2010.

[87] S. Niu, Y. Liu, J. Wang, and H. Song, “A decade survey of transfer learning (2010–2020),” *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[88] A Jahedsaravani, M. Marhaban, M Massinaei, M. Saripan, and N Mehrshad, “Development of a new algorithm for segmentation of flotation froth images,” *Mining, Metallurgy & Exploration*, vol. 31, no. 1, pp. 66–72, 2014.

[89] Y. Fu and C. Aldrich, “Froth image analysis by use of transfer learning and convolutional neural networks,” *Minerals Engineering*, vol. 115, pp. 68–78, 2018.

[90] D. Zhang and X. Gao, “Soft sensor of flotation froth grade classification based on hybrid deep neural network,” *International Journal of Production Research*, vol. 59, no. 16, pp. 4794–4810, 2021.

[91] J. Liu *et al.*, “Illumination-invariant flotation froth color measuring via wasserstein distance-based cyclegan with structure-preserving constraint,” *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 839–852, 2020.

[92] Yousefi, Siamak and Kehtarnavaz, Nasser and Cao, Y and Razlighi, Qolamreza R, “Bilateral Markov mesh random field and its application to image restoration,” *Journal of Visual Communication and Image Representation*, vol. 23, no. 7, pp. 1051–1059, 2012.

[93] S. Citrin and M. R. Azimi-Sadjadi, “A full-plane block Kalman filter for image restoration,” *IEEE Transactions on Image Processing*, vol. 1, no. 4, pp. 488–495, 1992.

[94] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[95] X. Peng, X. Zhang, Y. Li, and B. Liu, “Research on image feature extraction and retrieval algorithms based on convolutional neural network,” *Journal of Visual Communication and Image Representation*, vol. 69, p. 102 705, 2020.

[96] J. Duan and X. Liu, “Online monitoring of green pellet size distribution in haze-degraded images based on vgg16-lu-net and haze judgment,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–16, 2021. DOI: 10.1109/TIM.2021.3052018.

[97] A. Motamed Dashliborun, J. Zhou, P. Esmaeili, and X. Zhang, “Microbubble-enhanced recovery of residual bitumen from the tailings of oil sands extraction in a laboratory-scale pipeline,” *Energy & Fuels*, vol. 34, no. 12, pp. 16 476–16 485, 2020.

[98] N. E. Khalifa, M. Loey, and S. Mirjalili, “A comprehensive survey of recent trends in deep learning for digital images augmentation,” *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2351–2377, 2022.

[99] A. Vicente *et al.*, “Computer vision system for froth-middlings interface level detection in the primary separation vessels,” *Computers & Chemical Engineering*, vol. 123, pp. 357–370, 2019.

[100] F. Amjad, S. K. Varanasi, and B. Huang, “Kalman filter-based convolutional neural network for robust tracking of froth-middling interface in a primary separation vessel in presence of occlusions,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–8, 2021.

[101] T. Ruzic and A. Pizurica, “Context-aware patch-based image inpainting using Markov random field modeling,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 444–456, 2015. DOI: 10.1109/TIP.2014.2372479.

[102] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, “Patch-based image inpainting via two-stage low rank approximation,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 6, pp. 2023–2036, 2017.

[103] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, and H. Yu, “Deep learning for image inpainting: A survey,” *Pattern Recognition*, vol. 134, p. 109 046, 2023.

[104] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.

[105] U. Demir and G. Unal, “Inpainting by deep autoencoders using an advisor network,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 2017, pp. 1–4. DOI: 10.1109/SIU.2017.7960317.

[106] N. Sammaknejad, Y. Zhao, and B. Huang, “A review of the Expectation Maximization algorithm in data-driven process identification,” *Journal of Process Control*, vol. 73, pp. 123–136, 2019.

[107] D. Simon and Y. S. Shmaliy, “Unified forms for Kalman and finite impulse response filtering and smoothing,” *Automatica*, vol. 49, no. 6, pp. 1892–1899, 2013.

[108] X. Liu and X. Yang, “Robust variational inference for LPV dual-rate systems with randomly delayed outputs,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021. DOI: 10.1109/TIM.2021.3067242.