Robust Latent Variable Modeling Using Probabilistic Slow Feature Analysis

by

Lei Fan

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

Data-driven modeling approaches have been widely studied and applied to the process industries for inferential sensor development, process monitoring and fault detection and early warnings, etc. Essential information of process, like dynamic and relationships between process variables are buried in the massive archived historical data. They are often with high dimensionality and corrupted by diffident kinds of data irregularities, e.g. outliers, missing and multi-rate samples, uncertain time delays, etc. To address all these data irregularities and build a computational efficient modeling approach, the latent variable modeling has become a preferred and successful method. In most chemical processes, the process condition does not vary too fast and often contains large inertia. It is naturally considered that the features with small varying velocity are informative and carry most of the information of the process. With a probabilistic formulation, dynamic latent variable models, based on extracting slowly varying features, are developed in this thesis to address the aforementioned data irregularities, thus give reliable prediction results of quality variables that are otherwise difficult to measure.

Outliers are observations that are distant from other observations and they are common in process variable measurements. A robust dynamic latent feature extraction model is first proposed in this thesis to handle the outlier issue. By assuming the observations following the Student's t-distribution that has heavier tails, more weights can be assigned to the outliers thus they can be properly accounted for during modeling process. In feature extraction phase, a weighted Kalman gain is proposed since it violates the Gaussian assumption of the traditional Kalman filter. Smoother and slower features can be extracted and the impact of outliers is alleviated by the latent variance scale.

The next contribution of this thesis is to develop a semi-supervised model based

on probability slow feature analysis to include the information from quality variables in the extracted latent features while accounting for the missing data issues in quality variables. An approach by augmenting both input and output variables is proposed. It can deal with the different missing data issues, i.e. either missing at random or multi-rate sampling. In latent feature extracting process, the quality variable samples can be utilized whenever they are available. The compensation by the past quality variable samples leads to better predictability of its future samples.

Another irregular property of the lab samples of quality variable is its uncertain time delays. In many cases, the quality variables are sampled and analyzed manually by operators if the real-time on-line analysis is not possible. Various factors during manual sampling, i.e. human errors, manual sample, lab analysis and data recording procedures, etc can result in time-varying time delays on the quality variable samples. Another latent variable, delay indicator which evolves following a hidden Markov model, is introduced in the variational Bayesian framework to address this issue. The preference of model parameters is given as their prior distributions. More accurate and meaningful dynamic latent features can be extracted using the shifted samples of quality variables.

Time-varying time delays not only exist in the quality variables, but also in the fast-sampled process variables since their distributed locations in the plant. The changes of process conditions, varying velocity of flows, changing viscosity of transmission materials, etc., will cause the changes of delay to the target quality variable. The generalization formulation of the earlier work is proposed to address this issue. Multiple Markov chains are introduced to represent the different time-varying time delay sequences for different process variables. Dynamic latent features are extracted using both the shift process variables and scattered quality variable samples. With the consideration of the shifted observations, better prediction results of quality variable are provided.

The validity and practicality of these proposed probabilistic latent variable modeling approaches are verified through numerical examples, benchmark simulations, experimental studies and industrial applications. Specifically, the application to the SAGD well pair water content prediction performance is improved by applying proposed methods when data irregularities are considered.

Preface

This thesis is an original work done by Lei Fan under the supervision of Dr. Biao Huang and is funded in part by Natural Sciences and Engineering Research Council (NSERC) of Canada. Part of chapter 2 of this thesis has been published as: Fan L, Kodamana H, Huang B. Identification of robust probabilistic slow feature regression model for process data contaminated with outliers. Chemometrics and Intelligent Laboratory Systems. 2018 Feb 15;173:1-3. Part of chapter 3 of this thesis has been published as: Fan L, Kodamana H, Huang B. Semi-supervised dynamic latent variable modeling: I/O probabilistic slow feature analysis approach. AIChE Journal. 2019 Mar;65(3):964-79. Part of chapter 4 of this thesis has been submitted as: Fan L, Huang B. Dynamic Latent Variable Modeling with Output Time-varying Time Delays to Journal of Process Control. Part of chapter 5 of this thesis will be submitted as: Fan L, Huang B. Dynamic Latent Variable Modeling with Input Time-varying Time Delays and Application to SAGD Process.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Biao Huang for the guidance and inspiration that led me exploring the path of research and industrial projects. I greatly appreciate his patience and investment of time in helping me and providing useful critiques when I am in trouble with innovative ideas, derivations and simulations, etc. His creative thoughts and positive attitude always give me hope and support me to complete this journey. I am honored to have not only a scholarly supervisor but also a wise life mentor like him.

I wish to show my gratitude to Dr. Hariprasad Kodamana and it is always a pleasure to work with him. He showed tremendous patience when he guided me for the first two years of research, sometimes even as detail as how to derive a specific formula. His selfless assistant and optimistic attitude inspired not only me but also many people in our group.

A significant part of the work in this thesis and achievements on industrial projects benefit from the close collaboration with Dr. Yanjun Ma in Conputer Process Control (CPC) group and experts from oil industry. Dr. Yanjun Ma elevated my research with genius advices and provided good references. I also appreciated the great help from Yamei Liu, Tawanda Mutasa, Yanjun Ma and Haitao Zhang provided in solving complex industrial problems and their rich industrial experiences and professional attitude always encourage me to learn more in engineering field.

Moreover, I would like to thank my colleagues in CPC group during past years, including Fadi Ibrahim, Yujia Zhao, Shunyi Zhao, Nabil Magbool Jan, Rahul Raveendran, Agustin Vicente, Shabnam Sedghi, Mengqi Fang, Guoyang Yan, Atefeh Daemi, Hashem Alighardashi, Rui Nian, Xunyuan Yin, Seraphina Kwak and many others who provided help to me. I want to acknowledge National Science and Engineering Research Council (NSERC) of Canada for their financial support, and the Department

of Chemical and Materials Engineering at the University of Alberta for providing a pleasant environment to pursue my Ph.D. degree.

At last, I want to express my special thanks to my family and my friends for all the encouragement, and especially to my wife for taking care of two kids and the family chores. Without her support as always and encouragement in the tough time, the journey cannot be completed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Research Overview

In this chapter, the motivations of this thesis are introduced first and then the literatures relate to data irregularities and latent variable modeling are reviewed.

1.1.1 Motivation

In modern process industries, successful implementation of advanced control technologies and process monitoring techniques, especially for key quality variables, heavily rely on timely on-line measurements. Sometimes, development or installations of physical measuring instruments are impossible due to inadequacy of measurement techniques, harsh environments or economic infeasibility. Thus, on-line acquisition of these data is difficult if not impossible. One way to solve this problem is the development of inferential sensors, also called soft sensors, from off-line laboratory samples. However, a challenge is that the laboratory data has discontinuity, large delays and missing information.

Soft sensors usually take available process variables as inputs to estimate key quality variables that are not possible or very difficult to measure by physical sensors on-line. Soft sensor has many advantages such as: (i) cost-effectiveness, (ii) easy implementation, and (iii) providing insights of the process [1]. The process to build a soft sensor is equivalent to building a model between real-time process variables and key quality variables on the basis of their correlations. Generally, there are three types of models: (i) First principles models based on mass, energy and momentum balances; (ii) Black box models based on input-output data; and (iii) Grey box models

combining physical laws and process data. First principles model can give more insight of the process and has a wider range of validity, but usually difficult to build due to inadequate process knowledge. Since most chemical processes are complex and there is no or limited process information, grey and black box model approaches are often the common choices. However, the existing industrial practices have imposed some challenges on grey or black model development.

Since black box model does not incorporate any prior process knowledge other than the information in data, the performance of the model will highly depend on data quality and the adopted method of model development. Consider the oilsands industry as an example, it is known for its harsh production environment and it is important to maintain the safe, stable and sustainable operation in the plants. The challenge is that production process involves uncertainties due to various factors and these uncertainties will eventually be reflected in the data as irregularities. Irregularities in data will cause biased or even wrong estimation of the parameters of the target model. Therefore, irregularities in data have to be seriously and systematically accounted for while obtaining the system models and robust modeling strategy has to be adopted. Bayesian methods provide a natural way to combine prior information with data, hence a natural choice for grey box modeling. When new data is available, the prior information can be updated based on new process scenarios. Bayesian methods provide convenient ways to estimate model parameters and also can handle the missing data and non-Gaussian distribution problems. Thus, Bayesian approach is a powerful tool for modeling of the process. The other major hurdle in developing models based on the process data is its high dimensionality. The essential information that is useful in developing prediction models of quality variables is normally buried in those high-dimensional data. Thus how to extract the latent information becomes a popular research subject. Various methods have been developed in literature to address the issue of dimensionality of the process data based on latent variable models. Hence, a successful modeling paradigm should be able to account for both the irregularities and high dimensionality of the industrial process data simultaneously. Hence, in this thesis we seek to develop robust latent variable models which simultaneously address above mentioned problems. Specially, we develop latent variable models which are based on Probabilistic Slow Feature Analysis and develop various

strategies to address the irregularities in the process data, thereby developing a robust modeling paradigm for handling high dimensional data.

1.1.2 Robustness Issues in Process Modeling

Robust system identification methods are the identification approaches capable of handling various data irregularities. The following section presents detailed review about data irregularities and approaches for handling them.

1.1.2.1 Irregularities in Data Magnitudes

Irregularities in data values, also known as outliers, are observations that are distant from other normal observations. Outliers may occur due to disturbances, instrument failures, wrong indicator readings or sudden changes in an operational mode. To address these issues, various outlier detection techniques and robust regression models have been developed [2]. During the data preprocessing phase, some methods directly remove the detected outliers [3] and fit the good data in the classical way. But not all outliers are harmful to the identification process. Some of them may be the influential observations while others may contain useful information about system dynamics. Thus, the outliers should not be arbitrarily removed, but accommodated in the identification process in a systematic fashion.

Robust identification methods addressing outlier issues have been well discussed in literature. Most of the methods deal with such situations by choosing an appropriate noise model. One of the commonly chosen distributions for noise model is Student's t-distribution [4–7]. A smaller degree of freedom in Student's t-distribution represents a longer tail which can accommodate larger outliers. Based on this property, reference [6] proposed a robust multi-model linear parameter varying (LPV) approach to identify non-linear process contaminated with outliers. Besides Student's t-distribution, mixture Gaussian distribution [8, 9] has also been used to address robustness to outliers. Reference [8] proposed two types of mixture Gaussian distribution to tackle scale outliers and locations outliers respectively, which are generated by a shift in the scale (variability) or in the location (mean) of measurement noise. Other distributions like Laplace distribution [10] and skewed distribution [11] have also been applied to handle the outliers. For the classification problems, refer-

ence [12] proposed outlier robust Gaussian process classifiers (GPCs) which is based on the Expectation Propagation (EP) method.

1.1.2.2 Irregularities in Data Availability

Irregularities in the data availability are very common in chemical processes. There are two main types of measurements in process industries: sensor measurements and laboratory measurements. Sensor measurements are often conducted by installing specific hardware to measure process variables, such as: level, flow, pressure and temperature, etc. The measured values are sampled and transferred to Distributed Control System (DCS) in a pre-programmed sampling rate. Laboratory measurements are normally collected from process units for analyzing in laboratory manually. Although on-line measurements are automatically sampled and collected, they also suffer from irregular sampling problems from a data availability point of view, e.g. multi-rate sampling, uneven rate sampling. Sometimes one may encounter unexpected situation of missing data due to hardware malfunction, data acquisition system failure and scheduled maintenance, etc. Besides all above scenarios, data from laboratory measurements can be unavailable due to large delays, human errors (errors in recording the values and time). The mechanism of missing data can be summarized and categorized into three classes [13]: (i) Missing completely at random (MCAR); (ii) Missing at random (MAR); (iii) Non ignorable mechanism (NI). All these situations can lead to information loss, biased estimation of model parameters and finally result in inaccurate predictions. Irregular sampling problems can also be treated as missing data problem. A popular method to deal with missing data is to assume data is missing at random and then solve it with Expectation Maximization (EM) algorithm [1, 14]. Under EM framework, a multiple model technique has been applied on LPV systems [15] and nonlinear parameter varying systems [16]. Instead of using EM algorithm, Generalized EM (GEM) algorithm has also been applied on LPV system [17, 18] to handle the data that are missing at random. In addition, reference [19] proposed a system identification method based on a subspace formulation and used the trace norm heuristic for structured low-rank matrix approximation with partially missing inputs and outputs. Multi-rate sampling problem is another kind of missing data problem. An ad hoc way to solve the multi-rate sampling problem is by

down sampling data which will inevitably results in loss of information. Slow Feature Analysis (SFA) is a technique which can to certain extent relieve this problem [20].

1.1.2.3 Irregularities in Time Delays

Time delay is a common phenomenon in process industries and it can exist in both fast-sampled process variables and slow-sampled laboratory variable. In process industries, the time delay in fast-sampled process variables is mainly caused by the measurement sensors that are installed in various areas of the plant and the transportation of the materials need different amount of time [21]. In addition, due to changes of process condition and the properties of the transmission materials, e.g. viscosity, components composition and transmission velocity etc., the time delay could change along with time [22, 23]. Beside above reasons, time delay in laboratory data is often affected by human errors or sampling and lab analysis procedures. For example, it may take different time to analyze different batch of samples or inconsistent of execution of sampling process from different operators, etc. There are many methods to estimate the time delays. It is common to consider it as a deterministic parameter, which is the simplest case and easiest way to estimate it, either through experimental approach [24] or data-driven approaches [25, 26]. For non-linear systems, a sliding mode method has been developed to identify time delay within certain boundaries [27]. A modified least squares method is used in on-line identification case [28]. For the state-space formulation, identifiability of time delay has been investigated [29]. In the more challenging case of identifying time-varying time delays, the randomness shown in time delays could impact the identification process [21–23,30]. To resolve the problem, time delay can be identified as a sequence of unknown parameters [31, 32]. In this case, probabilistic models can be utilized to deal with the randomness in time delays [30,33,34] and also be able to incorporate modeling preferences under Bayesian framework [30].

In industrial processes, data obtained with the irregularities are difficult to use. Use of these data leads to poor estimation of parameters. To address data irregularities, a robust modeling strategy needs to be developed so that the model can still make accurate prediction in spite of irregularities in data. In addition to these data irregularities, the issue of high data dimensionality is another challenge. Application

of latent variable modeling techniques, which can handle high dimensional data are reviewed in next subsection.

1.1.3 Modeling of High Dimensional Process Data

Nowadays, the big data analysis has become popular in machine learning. Massive amount of data with large dimensionality have been accumulated in industries. What information these data contain and how to extract useful information to reduce the dimensionality of the data is of a great interest. If the dimensionality of the archived process data set is large, one might need to develop very high dimensional models to describe the behaviour of the data. Development of such models can be computationally difficult. In addition, the historical data ofter suffers from information redundancy [1] since many high correlated process variables have the same variation patterns as they may originate from the same source. For example, a level variation in upstream process will cause the occurrences of the similar variation in many process variables in downstream since the associated control strategy tries to suppress this variation. In such scenarios, in order to develop models in a more efficient way, development of lower dimensional latent variable models is a more practical alternative. A latent variable is a variable that is not directly observed from process but inferred from other variables that can be observed directly. The model that relates the set of observations to the latent variables are called latent variable model. Employment of latent variable models often results in dimensionality reduction since the dimension of latent variables is much less than raw data dimension.

The success of a learning algorithm relies heavily on the choice of data representations (features) extracted from original data. Representation learning is thus becoming a rapidly developing area that provides a new perspective when building new classifiers or predictors [35]. From the probabilistic modeling perspective, the question of representation learning can be interpreted as an attempt to recover a parsimonious set of latent random variables that describe a distribution over the observed data. Typical latent variable models are Principal Component Analysis (PCA) [36, 37], Partial Least Square (PLS) [38–40], Independent Component Analysis (ICA) [41, 42], and Slow Feature Analysis (SFA) [43] etc. All these methods project higher dimensional original data space to lower dimensional latent space.

Each technique extracts different data representations to represent information from explanatory factors hidden in the data, e.g. PCA extracts features that have the maximum variance. These algorithms are extended in a probabilistic sense to account for various noise distributions. The general structure of such models is [44–47]:

$$\begin{cases} x_k = W z_k + \mu_1 + e_{1,k} \\ y_k = Q z_k + \mu_2 + e_{2,k} \end{cases} \tag{1.1}$$

where, x, y and z are inputs, outputs and latent variables respectively and μ_i and $e_{i,k}$ are mean values and noise terms of the model. Q and W are latent variable mapping matrices.The aforementioned latent variable models except SFA are described in following paragraphs and SFA will be introduced in the mathematical fundamental subsection.

PCA is a well-established technique for dimensionality reduction, data compression, visualization and feature extraction [48]. PCA uses an orthogonal transformation to transfer a set of correlated variables into a set of linearly uncorrelated variables (principal components). The principle of PCA is to project a set of observed data vector to orthonormal principal axes in order to maximizes the variance in the projected space [37]. If the training data is corrupted by outliers or any other irregularities, robust PCA [49, 50] can be applied, which computes the principal components using a robust estimator for covariance matrix. The limitation of above conventional PCA is the absence of an associated probabilistic description for the observed data. To address this issue, Probabilistic Principal Component Analysis (PPCA) was proposed by [44], considering latent variables as Gaussian random variables. This method is closely related to statistical factor analysis. In PPCA, the principal axes are the maximum likelihood estimates of the parameters, which can be calculated by eigendecomposition. By specifying proper prior distribution to the noise terms in (1.1), PPCA can also be robust to data irregularities as mentioned above [51–53].

The Partial Least Square methods was first developed by Herman Wold in chemometrics and econometrics fields in the 1960's [38] and has since been widely applied in chemometrics [54]. An overview of PLS application to different data analysis problems is provided in [55] and the connections among PCA, Canonical Correlation Analysis (CCA) and PLS are also discussed. PLS tries to find the fundamental relations between inputs and outputs matrices, i.e. a latent variable approach to model

the covariance structures in both inputs and outputs spaces. Comparing with PCA approach, PLS incorporates information not only from inputs but also from outputs. PLS regression (PLSR) gives solution of linear regression by carrying out orthogonal projections from input space to latent space [39]. And also, PLS can be used to handle the collinearities among the independent variables X in multiple regression process [54, 56]. A recursive PLS regression method [40] was also proposed for on-line system identification and circumventing ill-conditioned problem. Based on PLSR and PPCA, a generative form of the Probabilistic PLSR (PPLSR) model was proposed by [45, 57] for quantitative analysis for Raman spectroscopy data. It provides a probabilistic view of the traditional PLSR model and explains the relationship between two variables and the latent variable. It also provides a foundation to develop robust and more accurate PPLSR models in a Bayesian framework, in order to solve the over-fitting problem. Bayesian framework not only incorporates the prior knowledge but also can automatically deal with the model complexity to avoid over-fitting issue [58].

Independent Component Analysis (ICA) [41, 42] is another widely used latent variable model. ICA is used to separate a multivariate signal into several sources. The premise is assuming these sources to follow non-Gaussian distribution and to be statistically independent, or as independent as possible. Such a representation can capture the essential structures of the data in many applications, including feature extraction and signal separation [59]. A classical application of ICA is the speech recognition problem. ICA can also be used in image preprocessing [41] and finding hidden factors in financial data [60]. These conventional ICA methods fail to take advantage of the statistical properties of the signals. Thus, several Probabilistic ICA (PICA) methods were developed to handle this problem. PICA assumes a small number of independent components with a residual term that is modeled as Gaussian noise [61]. Under the probabilistic framework, Sparse Code Shrinkage [62] was proposed to denoise the non-Gaussian data by maximum likelihood estimation. In PICA, when the number of sources, M, is less than the number of sensors, N, it will lead to the so-called non-square mixing, where the 'extra' sensor observations are explained as observation noise. The likelihood calculation of non-square models is intractable as it involves an integral, which can be represented by a Laplace approximation [63]. An example of PICA on non-square mixing process is the application of functional

MRI (FMRI) data with Gaussian noise [46]. Recently, a unified probabilistic model for PCA and ICA was also proposed [64] which tries to model PPCA if components are Gaussian and PICA if the components are non-Gaussian.

1.2 Mathematical Fundamentals

In this section, a brief introduction of SFA and Probabilistic SFA (PSFA) is provided first. Then two solution tools, EM algorithm or Variational Bayesian (VB) algorithm, will be introduced.

1.2.1 Slow Feature Analysis and Probabilistic Slow Feature Analysis

Slow Feature Analysis is an unsupervised learning method for modeling invariant or slowly varying features from input signals. The procedure of SFA not only reduces the dimensionality of input signals, but also removes the noisy components in the signals which are uninformative for identification. The remaining components are informative and possess some desired properties: zero mean, unit variance and uncorrelated with each other.

Deterministic SFA Deterministic SFA was first proposed by [43] and it aims to find a set of non-linear functions $g(x) = \{g_1(x(t)), \cdots, g_q(x(t))\}$ to map an I-dimensional input vector $x(t)$ to a q-dimensional feature space $\mathcal{F}$. The Slow Features (SFs) are expressed as the output of these functions: $s_j(t) \triangleq g_j(x(t))(1 \leq j \leq q)$. The SFs should be as slow as possible, therefore:

$$\min_{g_j(\cdot)} \Delta(\cdot) \triangleq \min_{g_j(\cdot)} \left\langle \dot{s}_j^2(t) \right\rangle_t \tag{1.2}$$

subject to:

$$\left\langle s_j(t) \right\rangle_t = 0, \text{(zero mean)} \tag{1.3}$$

$$\left\langle s_j^2(t) \right\rangle_t = 1, \text{(unit variance)} \tag{1.4}$$

$$\forall i \neq j, \langle s_i(t) s_j(t) \rangle = 0, \text{(decorrelation and order)} \tag{1.5}$$

where, $\langle f(t) \rangle_t = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t) dt$ stands for the expectation over time and $\dot{s}_j$ refers to the speed of j-th slow feature. Constraints (1.3) and (1.4) help avoid the trivial

solution $s_j(t) = \text{const}$. Constraints (1.5) guarantees that different output signal components carry different information and do not simply reproduce each other. SFs are mutually independent and resulting in a natural descending order of $s_j, 1 \leq j \leq q$, in which the feature having the lowest index represents the slowest one. Thus, $\Delta(s_j) \leq \Delta(s_{j'})$ if $j < j'$. When we do the mapping using non-linear function $g(x) = \{g_1(x(t)), \cdots, g_q(x(t))\}$, we adopt the similar technique as support vector machine [65] to turn a non-linear problem into a linear one. Each component in $g(x)$ is a weighted sum over a set of K non-linear functions $h_k(x)$: $g_j(x) = \sum_{k=1}^{K} w_{jk} h_k(x)$, usually $K > \max(I, q)$ and the weighting vector $w_j = [w_{j,1}, \cdots, w_{j,K}]^T$ is to be estimated. Then, the j-th output component is given by $s_j(t) = g_j(x(t)) = w_j^T h(x(t)) = w_j^T z(t)$, which is in a linear-in-parameter form. Thus our objective is now to minimize:

$$\min_{g_j(\cdot)} \left\langle \dot{s}_j^2(t) \right\rangle_t = \min_{g_j(\cdot)} w_j^T \langle \dot{z}\dot{z}^T \rangle_t w_j \tag{1.6}$$

In the same time, the zero mean, unit variance and decorrelation constraints are still hold by deriving non-linear functions h_k from an arbitrary set h_k' using sphering technique [66].

SFA is based on the strong belief that the real process dynamics (especially in chemical processes) often change slowly and those features that change fast can be treated as noise.

Probabilistic SFA Based on above concepts, SFA has been extended to probabilistic framework (PSFA) [47]. The PSFA model is a first order Markov linear-Gaussian dynamic model (or AR(1) autoregressive model):

$$\begin{aligned} p(s_t | s_{t-1}, \lambda_{1:q}, \sigma_{1:q}^2) &= \prod_{j=1}^{q} p(s_{j,t} | s_{j,t-1}, \lambda_j, \sigma_j^2) \\ p(s_{j,t} | s_{j,t-1}, \lambda_j, \sigma_j^2) &= \mathcal{N}(\lambda_j s_{j,t-1}, \sigma_j^2) \\ p(s_{j,1} | \sigma_{j,1}^2) &= \mathcal{N}(0, \sigma_{j,1}^2) \end{aligned} \tag{1.7}$$

where λ is the AR(1) coefficient and σ_j^2 is variance. From eq. (1.7) we can see that, λ_j controls the strength of the correlation between the latent variables at different time points and therefore their slowness. If $\lambda_j = 0$, then successive latent variables are uncorrelated with previous latent variables, and the process varies randomly and

rapidly. When $\lambda_j \to 1$, the process becomes more temporally correlated, hence slower. Thus, we can conclude that if $|\lambda_j| < 1$, the latent series settles into a stationary state asymptotically ($t \to \infty$). In addition to the prior distribution of the latent variables $s(t)$, the complete specification of a generative model requires a probabilistic mapping from latent variables to the observations. The matrix of generative weights is the inverse of the recognition matrix w which is composed of the weighting vector w_j:

$$p(x_t|s_t, w, \sigma_x) = \mathcal{N}(w^{-1}s_t, \sigma_x^2\mathbf{I}) \tag{1.8}$$

Formulation in eq. (1.7) and (1.8) complete the specification of the probabilistic model of SFA and it is a linear Gaussian state-space model.

As an innovative latent variable model, the probabilistic SFA has been employed in inferential sensor design [20]. The mathematical formulation for PSFA is given as:

$$s(t) = Fs(t-1) + e(t), e(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \tag{1.9}$$

$$x(t) = Hs(t) + e_x(t), e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \tag{1.10}$$

where F, Λ and Σ are diagonal matrices defined as:

$$F = diag\left\{\lambda_1, \cdots, \lambda_q\right\}, \Lambda = diag\left\{1-\lambda_1^2, \cdots, 1-\lambda_q^2\right\}, \Sigma = diag\left\{\sigma_1^2, \cdots, \sigma_m^2\right\} \tag{1.11}$$

Under this formulation, each slow feature is consistently corrupted by an independent noise $e_j(t)$ following Gaussian distribution. The decorrelation nature in constraints (1.5) is characterized by the independence assumption of SFs. We can easily verify the properties of zero mean and unit variance for each SF:

$$E[s_j(t)] = 0, Var\{s_j(t)\} = 1, 1 \leq j \leq q \tag{1.12}$$

which are consistent with constraints (1.3) and (1.4). We have already known that the slowness of each slow feature is governed by λ_j. In fact, the slowness measurement $\Delta(\cdot)$ can be calculated as $\Delta(s_j) = 2(1-\lambda_j)$, which verifies that a large λ_j implies a strong correlation between $s_j(t)$ and $s_j(t-1)$, and indicates that $s_j(t)$ tends to have slow variations with a small $\Delta(\cdot)$, and vice versa.

The parameters to be estimated in a PSFA model are: $\theta \triangleq \{\lambda_j, 1 \leq j \leq q, H, \Sigma, \}$. These parameters can be estimated by maximizing the likelihood function $p(x_{1:T}|\theta)$.

In many cases, direct optimization of the incomplete likelihood is intractable. Reference [20] employed EM algorithm to estimate θ by maximizing the expectation of complete data log likelihood: $\log p(x, s|\theta)$. In order to predict outputs, we need to build a regression model using the selected SFs and output variables. The latent state $\boldsymbol{s}_j(t)$ can be inferred through Kalman smoothing recursions. First, forward recursions are used to calculate the posterior distribution $P(s(t)|x(1), \cdots, x(t), \theta^{old}) \sim \mathcal{N}(\mu_t, P(t))$. Then the parameters of the posterior distribution $P(s|x, \theta^{old})$ are obtained by backward recursions. After inferring SFs, we can choose appropriate number of SFs for regression, for which two criteria can be adopted. One is slowness-based criterion, that is to choose the M ($M \leq m$) slowest features for regression. This can be done by choosing them before their λ_j shows an apparent drop, which means that features after the drop become significantly faster and contain much more noise information. An alternative criterion is correlation-based, which will evaluate the correlation coefficient between each SF and output, and the M slow features with the highest correlation with the outputs will be chosen. With selected M slow features, a regression model can be fitted between outputs and SFs, as:

$$y(t) = b^T s_{1:M}(t) + c + \epsilon \tag{1.13}$$

where $y(t)$ is the target output, $b \in \mathbb{R}^M$ are regression coefficients, and c is bias term.

1.2.2 Expectation-Maximization Algorithm

EM algorithm can be used for point estimation of unknown parameters and it works best when the fraction of missing information is small and the dimensionality of data is not too large. The high dimension of data can dramatically slow down the E-Step. The rate of convergence is typically good in the first few steps but can be slow when approaching a (local or global) optimal point.

Maximum Likelihood (ML) Estimation Expectation Maximization [14] is a technique used for point estimation of parameters. Given a set of observation data D_o, we want to estimate parameters Θ in the model, with missing observations and latent (hidden) variables D_{mis}. D_o and D_{mis} together form the complete data set D. The target of EM algorithm is to find an optimal solution of unknown parameters by

iteratively maximizing the expectation of logarithm likelihood function (also known as Q function) through expectation step (E-Step) and maximization step (M-Step) with respect to missing observations and hidden features simultaneously. Assuming the general model structure is defined as follows:

$$y_k = f(x_k, \theta) + e_k \tag{1.14}$$

$X \triangleq [x_1, x_2, \cdots, x_n]^T$ is the regressor which includes past inputs and outputs and y is a scalar output. Parameters for noise e_k are θ_e and overall unknown parameters are denoted as $\Theta = \{\theta, \theta_e\}$. Then Θ can be identified by maximizing the log likelihood of observation y with respect to the unknown parameters:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \log p(y|X, \Theta) \tag{1.15}$$

Sometimes the likelihood function is complex and difficult to be maximized directly and we may not be able to find a solution for it. In such situations, EM can be used to estimate parameters by iteratively maximizing the lower bound of the likelihood function. When the noise distribution belongs to the exponential family, it is equivalent to maximizing the log likelihood function. For the case that noise distribution does not belong to exponential family, i.e. student t-distribution, it may be decomposed into a Gaussian distribution and a gamma-distribution by introducing the variance scale R_k [6] and R_k can be treated as latent variable. EM algorithm includes two major steps:

1. **E-Step:** In this step, the objective is to find the expected value of log likelihood with respect to the latent variables, which is also known as the Q-function:

$$Q(\Theta|\Theta^{(n)}) = E_{D_{mis}|D_o,\Theta^{(n)}}\{\log p(D_o, D_{mis}|\Theta)\} \tag{1.16}$$

   where, $\Theta^{(n)}$ represents the n-th iterative value of Θ.

2. **M-Step:** In this step, the updated Θ is obtained by maximizing above Q function:

$$\Theta^{(n+1)} = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta|\Theta^{(n)}) \tag{1.17}$$

   where, $\Theta^{(n+1)}$ is updated parameters at $n+1$-th iteration.

Then $\Theta^{(n+1)}$ is used in E-Step to update the Q-function, and then obtain $\Theta^{(n+2)}$ in M-Step again. This is an iterative procedure until convergence.

EM can be useful due to its conceptual simplicity, and easy to implement. It has been applied in many fields, such as multi-models [68], HMM [67], PCA [44], missing data problems [17], etc. One limitation of EM is that it is sensitive to initial conditions and poor initial condition may cause the algorithm to diverge.

Maximum A Posteriori (MAP) Estimation The EM algorithm can be extended to maximizing the posteriori $p(\Theta|D)$(or joint distribution $p(\Theta, D)$) when the prior $p(\Theta)$ is available with hyper-parameters. In this case, the E-Step is still used to evaluate $p(D_{mis}|D_o, \Theta^{(n)})$. In M-Step, instead of maximizing $Q(\Theta|\Theta^{(n)})$, we maximize $Q(\Theta|\Theta^{(n)}) + \ln p(\Theta)$. Thus, we need to carefully choose the prior to make the maximization process tractable. No matter maximizing the likelihood or posteriori, EM algorithm is a non-Bayesian method and its result gives a point estimation of unknown parameters Θ and posterior distribution over latent variables and missing data. It can incorporate prior information but cannot give the posterior distribution of Θ. In this case, we can use Bayesian approach to get the full posteriori over Θ as well as latent variables. Variational Bayesian method is one of the most commonly used method for Bayesian inference. It will be introduced in next section.

1.2.3 Variational Bayesian Inference

Comparing with EM algorithm, VB is a full Bayesian version of maximum likelihood or a posteriori estimation. It can incorporate prior information of parameters and handle incomplete data. VB iterates over free distributions of each latent variable (including unknown parameters) and optimize them one at a time by minimizing log marginal likelihood. Finally, VB provides the approximated posteriori distribution of unknown parameters and latent variables. With posteriori estimation $P(\Theta|D)$ provided, VB can also be used for model selection. Further, the factorization of free distribution is an important step in VB. Simpler factorization will take less computation time and also result in less accuracy in posteriori estimation. On the contrary, more complex factorization will yield tighter lower bound of marginal log likelihood that leads to more accurate posteriori estimation, but with more computation.

Bayesian Inference Bayesian inference is a statistical inference method in which Bayes' rule is used to calculate the probability of a hypothesis given observation data. Bayesian inference has been applied in various fields, i.e. engineering, science, sport, phychology, medicine, etc. The core principle Bayesian inference used is Bayes' rule

$$p(\Theta|D) = \frac{p(D|\Theta)p(\Theta)}{p(D)} \tag{1.18}$$

where:

- Θ is the hypothesis which represents all the unknown parameters we want to estimate.
- D represents observed data.
- $p(\Theta|D)$ is the posterior distribution of the hypothesis when taking into account the observed data.
- $p(D|\Theta)$ is the distribution of the observed data conditional on the hypothesis. It is also termed as likelihood and is a function of the hypothesis (parameters).
- $p(\Theta)$ is the prior distribution of the hypothesis before any data is observed. It can be viewed as priori knowledge or experiences about the hypothesis.
- $p(D)$ is a normalizing constant which represents model evidence.

When no prior experience of hypothesis $p(\Theta)$ is available, $p(\Theta)$ can be treated by uniform distribution, and to maximize the likelihood is equivalent to maximizing the posterior, i.e. $\Theta^{ML} = \Theta^{MAP}$.

Variational Bayesian Approach In contrast to EM, Variational Bayesian approach is an approximate approach to explicitly calculate the posterior distribution of parameters as well as latent variables. For a problem with model structure M, observed data D_o, non-observed data set D_{mis} (missing data) and unknown model parameters Θ, where we can denote all unobserved variables as $Z = \{D_{mis}, \Theta\}$, the posterior distribution of Z is:

$$p(Z|D_o, M) = \frac{p(D_o|Z, M)p(Z|M)}{p(D_o|M)} = \frac{p(D_o|Z, M)p(Z|M)}{\int_Z p(D_o|Z, M)p(Z|M)dZ} \tag{1.19}$$

where $p(D_o|Z, M)$ is the data likelihood. For most practical models, it is complicated and intractable to calculate the integral part in the denominator of eq. (1.19) within polynomial time. One of the solutions to deal with problems stated above is to find a simpler and more tractable distribution $q(Z)$ to approximate the true posterior distribution $p(Z|D_o, M)$, i.e. $p(Z|D_o, M) \approx q(Z)$. Then two problems arises:

(1) if such a $q(Z)$ exists, how to measure the similarity between $p(Z|D_o, M)$ and $q(Z)$?

(2) how to get a simpler $q(Z)$?

For problem (1), we already have such an index to measure the dissimilarity between two probability distributions P and Q, which is called Kullback-Leibler divergence, defined as:

$$KL(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{1.20}$$

For problem (2), a common practice is to factorize the approximate posterior $q(Z)$ into independent partitions, i.e. $q(Z) = q_{D_{mis}}(D_{mis})q_\Theta(\Theta)$ [69], which is known as mean field approximation. Adopting above methods, VB can make the integral part in the denominator of the eq. (1.19) tractable by introducing the approximate posterior $q(Z)$. Then applying Jensen's inequality yields,

$$\begin{aligned} \ln p(D_o|M) &= \ln \int_Z q(Z) \frac{p(D_o, Z|M)}{q(Z)} dZ \geq \int_Z q(Z) \ln \frac{p(D_o, Z|M)}{q(Z)} dZ \\ &= \int_{D_{mis},\Theta} q_{D_{mis}}(D_{mis}) q_\Theta(\Theta) \ln \frac{p(D_o, D_{mis}, \Theta|M)}{q_{D_{mis}}(D_{mis}) q_\Theta(\Theta)} dD_{mis} d\Theta \\ &\triangleq F_M(q_\Theta(\Theta), q_{D_{mis}}(D_{mis}), D_o) \end{aligned} \tag{1.21}$$

$F_M(q(\Theta), q(D_{mis}), q(D_o))$ is the lower bound of the marginal log likelihood $\ln p(D_o|M)$. The VB algorithm will iteratively maximize the likelihood by indirectly maximizing the lower bound $F_M(,,)$ in terms of the free distribution $q_{D_{mis}}(D_{mis})$ and $q_\Theta(\Theta)$. Taking functional derivatives of eq. (1.21) results in the following update equations to be solved iteratively, forming the Variational Bayesian EM algorithm [69]:

(1) updating missing data posterior:

$$q_{D_{mis}}^{(i+1)}(D_{mis}) \propto \exp\left[\int_\Theta \ln p(D_o, D_{mis}|\Theta, M) q_\Theta^{(i)}(\Theta) d\Theta\right] \tag{1.22}$$

(2) updating parameter posterior:

$$q_{\Theta}^{(i+1)}(\Theta) \propto P(\Theta|M) \exp\left[\int_{D_{mis}} \ln p(D_o, D_{mis}|\Theta, M) q_{D_{mis}}^{(i)}(D_{mis}) dD_{mis}\right] \quad (1.23)$$

Above two iterative updating steps will be carried out until convergence. Often, step (1) and (2) are referred to as variational Bayesian E-step and M-step.

The difference between the log marginal likelihood and its lower bound is the Kullback-Leibler divergence as we have introduced above:

$$\ln p(D_o|M) - F_M(q_{\Theta}(\Theta), q_{D_{mis}}(D_{mis}), D_o) = KL(q||p) \quad (1.24)$$

It is to be noted that, VB approaches require prior distribution to be a conjugate distribution of the likelihood for tractability.

1.3 Contributions and Thesis Outline

The rest of the thesis is organized as follows:

In Chapter 2, we consider the identification of robust probabilistic slow feature analysis in presence of outliers. A novel regression model RPSFA is proposed to address the outlier issue in process data by assuming the measurements follow Student's t-distribution, while reducing the dimension of slow features that are used for regression of quality variables. The problem is solved using EM algorithm, in which the extracted slow features and variance scale factor are considered as hidden variables. In RPSFA, a weighted gain Kalman filter is proposed as the noise violates the Normal distribution assumption. Based on the validation results of case study datasets, the proposed approach shows its strength for feature extraction and prediction ability of quality variables. This work has been published as: *Fan L, Kodamana H, Huang B. Identification of robust probabilistic slow feature regression model for process data contaminated with outliers. Chemometrics and Intelligent Laboratory Systems. 2018 Feb 15;173:1-3.*

In Chapter 3, a semi-supervised dynamic latent variable modeling approach, IOPSFA, is proposed to consider the information contained in quality variable. It overcome the drawback in conventional PSFA that it only account for the information carried by input variables, not by output variables. The extracted latent features

using IOPSFA approach are proved to have better prediction ability of quality variables than PSFA. IOSPFA can be applied to the datasets that contain a wild range of missing data, regardless of its missing mechanism, missing at random or multi-rate sampling problem. It can use the output information as soon as it is available. The efficacy has been demonstrated through an industrial application and an experiment case study. This work has been published as: *Fan L, Kodamana H, Huang B. Semi-supervised dynamic latent variable modeling: I/O probabilistic slow feature analysis approach. AIChE Journal. 2019 Mar;65(3):964-79.*

In Chapter 4, the time-varying time delay problem of quality variables is investigated. In this problem, the observation of outputs is assumed to have uncertain time delays due to sampling procedure, human errors and operating condition changes, etc. Based on the formulation of IOSPFA, a probabilistic model, IOPSFA_VTD is proposed and in this model, the outputs are reconstructed by shifting the observations. A delay indicator is defined as a latent variable that follows a hidden Markov model. The proposed model is solved under variational Bayesian framework, so process knowledges can be incorporated as the priors of the unknown parameters. The missing data problem, same as in IOSPFA, can also be accommodated in IOSFAP_VTD as they are under the similar formulation. Through a numerical simulation and a benchmark CSTR simulation, the proposed model is validated and proved to have advantages comparing with IOSPFA and the fixed time delay cases. This work has been submitted to the Journal of Process Control and it is currently under review.

In Chapter 5, a more general case of time-varying time delay problem than Chapter 4 is investigated. Instead of only considering that the time-varying time delay exists in output, we consider the case that all input variables have different time-varying time delays with reference to the output, which is closer to the practical situation of processes since the sensors in plant are often distributed, i.e. at different locations across the whole plant. As a result, multiple time delay indicators are defined and the input measurements are shifted according to them to reconstruct the delay-free observations. The latent dynamic features are extracted from these delay-free observations. By eliminating the effect of delays, the extracted features are proved to have better prediction ability for quality variables than IOSPFA_VTD, IOSPFA and fixed delay cases.

In Chapter 6, concluding remarks are presented, and the possible future work and further improvements are discussed.

Chapter 2

Identification of Robust Probabilistic Slow Feature Regression Model for Process Data Contaminated with Outliers *

Modeling of high dimensional dynamic process is considered as a challenging task. In this regard, probabilistic Slow Feature Analysis (PSFA), a dynamic latent variable model, is proven to be a useful tool which extracts temporally correlated dynamic features from the high-dimensional raw measurements. The extracted latent Slow Features (SFs) can capture process variations which are useful in developing dynamic models. Often times industrial data is affected by outliers, and modeling such data could result in inferior prediction performance. To deal with such scenarios, we propose a robust PSFA (RPSFA) based regression model that models outliers in the observation data using the Student's t-distribution. To estimate the parameters in RPSFA and to extract reduced dimension of SFs, we employ Expectation-Maximization (EM) algorithm under the Maximum Likelihood Estimation (MLE) framework considering SFs as hidden variables. To estimate the hidden SFs we propose a weighted gain Kalman filter based approach as the Normal distribution assumption of the observations is no longer valid. The validity and merits of the proposed approach are demonstrated though a simulated example, an industrial application and an experimental study.

*Part of this chapter has been published as: Fan L, Kodamana H, Huang B. Identification of robust probabilistic slow feature regression model for process data contaminated with outliers. Chemometrics and Intelligent Laboratory Systems. 2018 Feb 15;173:1-3.

2.1 Introduction

Data based system identification methods are widely used to develop process models from industrial data [70, 71]. Very often, many plants have years of historical data archived in their database, which contain information about the process dynamics and relationship between different process variables. With the rapid development of information technology, historical data can be favorably used for process modeling, optimization and inference. Data-driven soft sensor development is one of the successful areas where archived industrial data is employed to develop models to predict the variables that are difficult to measure by hard-wired instruments. In contrast to physics based first principles models, which require a thorough understanding of the complicated underlying physics, data-driven models aim to extract and learn the process features from historical data, thereby enabling us to build models with less difficulty.

When developing models using data driven methods, data quality plays a crucial role in the modeling process and the final prediction performance. Further, complexity, high dimensionality and unpredictable uncertainties also impose enormous challenges in process modeling. Usually, raw data is messy, i.e., contaminated by noise, outliers and bad data, due to sensor malfunctions or human errors. There are many different ways to extract useful features from raw data, meanwhile reducing its dimension. In general, high dimensional data is composed of highly correlated variables, which contain the redundant information of the process. In such cases, key underlying features which are called latent features, can be extracted from the observations and can be represented in the form of latent variables (LVs). From a certain point of view, LVs represent the inherent common causes of the variations of raw data [20] in a lower-dimension space. The popular latent variable models used for prediction include principal component regression (PCR) [72] and partial least square regression (PLS) [54,56] among many others [73] A drawback of traditional PCR and PLS is that they cannot describe the dynamic relationship between samples at different time instants. Dynamic PCR (DPCR) and dynamic PLS (DPLS) infer latent LVs using lagged observations, but the lagged observations will increase the dimension of observation space [74], which further results in the increase of complexity.

While modeling dynamic data, the latent features that represent the intrinsic information of processes are expected to be temporal correlated. However, the LV methods such as PCR and PLS could only be used to capture the static characteristics. To deal with such scenarios, slow feature analysis (SFA), which is an unsupervised learning method for modeling slowly varying features from the signals, developed by [43] proved to be useful [75]. SFA can learn temporal correlated latent features from the data and has been successfully used for process monitoring [76] and modeling [20]. SFA is based on the belief that the real process dynamics, especially in chemical processes, often change slowly and the features that change fast can be treated as noise. Its extension to probabilistic framework, probabilistic slow feature analysis (PSFA) [47], uses state space form to describe the process dynamics. Also, it models distributions of latent features.

PSFA models a process assuming Gaussian distribution in the noise and has been successfully employed to predict quality variables in chemical processes [20]. However, Gaussian noise model fails in modeling outlier contaminated data [77]. Outliers in data may occur due to large disturbances, instrument failures, wrong indicator readings or sudden changes in an operational mode and the impact of outliers on parameter estimation and prediction results can be significant if they are not handled appropriately [2]. However, outliers cannot be arbitrarily removed either, since they likely contain some dynamic information that is useful to process modeling. The most intuitive way to handle outliers is to choose an appropriate distribution other than Gaussian distribution to model the noise. In literature, several distributions have been utilized for modeling outliers. For example, Gaussian mixture distribution has been adopted by [8] for Autoregressive Exogenous (ARX) model to make it robust to outliers. In addition, modeling using Student's t-distribution has been employed as a more general approach by many researchers to model outlier contaminated observations [5, 6, 77] as it overcomes the limitation of Gaussian mixtures which can only deal with a certain class of outliers.

In this study, to simultaneously address issues in process data such as high dimensionality, dynamic characteristics as well as issues due to outliers while modeling, we propose robust PSFA (RPSFA) by assuming that the observation noise follows Student's t-distribution. For the estimation of parameters in the proposed model, we

apply Expectation-Maximization (EM) algorithm, under the Maximum Likelihood Estimation (MLE) Framework, where the SFs are considered as hidden variables. In additional to that, one more hidden parameter - variance scale [78] is introduced to decompose the Student's t-distribution. To estimate the hidden variables, a weighted gain Kalman filter technique is applied to correct the mismatch between true noise distribution and Gaussian distribution. Compared with PSFA and other commonly used modeling approaches such as, MLR, PCR and PLS, etc, the proposed method is envisaged to accommodate outliers in the SFs extraction process to obtain slower and smoother features, for diminishing the impact of outliers. After extracting the desired SFs, we develop models between the desired output variable and the selected lower dimensional SFs to have RPSFA based regression model. Three examples, namely, Monte Carlo simulations are conducted on classic Tennessee Eastman (TE) process under different percentage of outliers, an industrial case study is performed on a water content soft sensor in oilsands Steam-assisted gravity drainage (SAGD) process and an experimental study of hybrid tanks is employed to validate performance of RPSFA.

The rest of the chapter is organized as follows. In section 2, prerequisite including Student's t-distribution, definition and properties of SFA and PSFA is briefly introduced. Section 3 and section 4 give the formulation of RPSFA and detailed derivation of EM algorithm for parameter estimation, respectively. Following that, three case studies are presented in section 5, to show the efficacy of the proposed approach: (i) the Monte Carlo simulations conducted on TE process in different outlier percentage scenarios; (ii) a soft sensor development for SAGD process and (iii) an experiment on hybrid tanks system. In section 6, the conclusions from the studies are reported.

2.2 Revisit of SFA and Probabilistic SFA

SFA is an unsupervised learning method for extracting features from signals according to their slowness. SFA not only reduces the dimensionality of signals, but also removes the fast components in signals which are usually uninformative to process identification.

SFA: SFA was first proposed by [43] and it aims to find a set of non-linear func-

tions $g(x) = \{g_1(x(t)), \cdots, g_q(x(t))\}$ to map a m-dimensional input vector $x(t)$ to a q-dimensional feature space $\mathcal{F}$. The SFs are expressed as the outputs of these functions: $s_j(t) \triangleq g_j(x(t))(1 \leq j \leq q)$. The following formulation is used to extract SFs from the raw data [43]:

$$\min_{g_j(\cdot)} \Delta(\cdot) \triangleq \min_{g_j(\cdot)} \left\langle \dot{s}_j^2(t) \right\rangle_t \tag{2.1}$$

subject to:

$$\left\langle s_j(t) \right\rangle_t = 0, \quad \text{(zero mean)} \tag{2.2}$$

$$\left\langle s_j^2(t) \right\rangle_t = 1, \quad \text{(unit variance)} \tag{2.3}$$

$$\forall i \neq j, \langle s_i(t) s_j(t) \rangle = 0, \quad \text{(decorrelation and order)} \tag{2.4}$$

where, $\Delta(\cdot)$ is the defined measurement index of features' varying speed and i, j represent the number of slow features, e.g. s_i is the i-th slow feature. A smaller $\Delta(\cdot)$ represents a slower-speed feature. $\langle s_j(t) \rangle_t = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} s_j(t) dt$ stands for the expectation over time and $\dot{s}_j(t) = s_j(t) - s_j(t-1)$, which refers to the rate of change, hence speed, of j-th SF. Constraints (2.2) and (2.3) help avoid the trivial solution $s_j(t) = \text{const}$. Constraints set (2.4) guarantee that different features report different aspects of the stimulus [47]. SFs are mutually independent of each other and solving (2.1) to (2.4) results in natural descending order of $s_j, 1 \leq j \leq q$, in which the feature having the lowest index represents the slowest one, that is, $\Delta(s_j) \leq \Delta(s_{j'})$ if $j < j'$.

Linear SFA: When the mapping functions from input space to feature space are linear, we can derive SFs in linear form:

$$s(t) = W^T x(t) \tag{2.5}$$

where $W = [W_1 W_2 \cdots W_q] \in \mathbb{R}^{m \times q}$ is the mapping matrix. Thus, our objective becomes [43]:

$$\min_{W_j} \left\langle \dot{s}_j^2(t) \right\rangle_t = \min_{W_j} W_j^T \langle \dot{x} \dot{x}^T \rangle_t W_j \tag{2.6}$$

satisfying the constraints (2.2) to (2.4). When we want to derive the same number of SFs as that of inputs, i.e. $q = m$, the above optimization problem (2.6) leads to a generalized eigenvalue problem as follows [79]:

$$\left\langle \dot{x} \dot{x}^T \right\rangle_t W = \left\langle x x^T \right\rangle_t W \Omega \tag{2.7}$$

The mapping matrix W in (2.7) is composed of all eigenvectors and Ω is a diagonal matrix of eigenvalues, which in turn represent the varying speed of derived features.

PSFA: SFA has been extended in a probabilistic framework to PSFA [47] which is a first order Markov linear-Gaussian dynamic system (or AR(1) autoregressive model) as given below:

$$p(s(t))|s(t-1), \lambda_{1:q}) = \prod_{j=1}^{q} p(s_j(t)|s_j(t-1), \lambda_j) \tag{2.8}$$

$$p(s_j(t)|s_j(t-1), \lambda_j) = \mathcal{N}(\lambda_j s_j(t-1), 1-\lambda_j^2) \tag{2.9}$$

$$p(s_j(1)) = \mathcal{N}(0, \mathbf{I}_q) \tag{2.10}$$

where λ_j and σ_j^2 are the AR(1) coefficient and variance for the j-th dimension of SF, respectively. The notation $\lambda_{1:q} \triangleq \{\lambda_1, \cdots, \lambda_q\}$ represents the vector composed of AR(1) models' coefficients and $\mathcal{N}(\cdot, \cdot)$ represents the Gaussian distribution with parameters mean and variance. From (2.9) we observe that, λ_j controls the strength of the correlation between the SFs at different time points and therefore their slowness. If $\lambda_j = 0$, then successive latent variables are uncorrelated and the process varies randomly and rapidly. When $\lambda_j \to 1$, the process becomes more temporally correlated, hence slower. Thus, we can conclude that if $|\lambda_j| < 1$, the extracted features will settle into stationary states after a long time ($t \to \infty$). In addition to the prior distribution on the latent variables $s(t)$, the complete specification of a generative model requires a probabilistic mapping from latent variables to the observations.

$$p(x(t)|s(t), H, \Sigma) = \mathcal{N}(Hs(t), \Sigma) \tag{2.11}$$

Equations (2.8) to (2.11) completely specify the probabilistic model of PSFA and it is a linear Gaussian state-space model. The mathematical formulation of PSFA is given as:

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), & e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ x(t) = Hs(t) + e_x(t), & e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \end{cases} \tag{2.12}$$

where transition matrix F, latent states and observations noise covariance matrices Λ, Σ are diagonal and defined as:

$$F = \text{diag}\left\{\lambda_1, \cdots, \lambda_q\right\}, \Lambda = \text{diag}\left\{1-\lambda_1^2, \cdots, 1-\lambda_q^2\right\}, \Sigma = \text{diag}\left\{\sigma_1^2, \cdots, \sigma_m^2\right\} \tag{2.13}$$

$H \in \mathbb{R}^{m\times q}$ is an emission matrix, and m is the dimension of the observation space. Under this framework, each SF is consistently corrupted by an independent noise $e_j(t)$, which follows a Gaussian distribution. The decorrelation nature in constraints (2.4) is characterized by the independence assumption of SFs. We can easily verify the properties of zero mean and unit variance for each SF:

$$\mathbb{E}[s_j(t)] = 0, Var\{s_j(t)\} = 1, 1 \leq j \leq q \tag{2.14}$$

which is consistent with constraints (2.2) and (2.3). The slowness measurement $\Delta(\cdot)$ can be calculated as $\Delta(s_j) = 2(1-\lambda_j)$ such that a large λ_j implies a strong correlation between $s_j(t)$ and $s_j(t-1)$. This further indicates that $s_j(t)$ tends to have slower variation with a smaller $\Delta(\cdot)$, and vice versa.

2.3 Robust Formulation of Probabilistic SFA

In order to develop RPSFA for modeling industrial processes data contaminated with outliers, we proposed to adopt Student's t-distribution to model noise. Next, we will revisit Student's t-distribution and following that, the formulation of RPSFA is provided.

Multivariate Student's t-distribution: The probability density function (pdf) of multivariate Student's t-distribution is defined as follows:

$$\mathcal{S}t(x|\mu,\Sigma,\nu) = \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})}\frac{1}{(\nu\pi)^{\frac{d}{2}}}\frac{1}{\sqrt{|\Sigma|}}\left(1+\frac{1}{\nu}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)^{-\frac{d+\nu}{2}} \tag{2.15}$$

where, $\Gamma(\cdot)$ denotes the Gamma function, d is the dimension of the feature space, $\mu \in \mathbb{R}^{d\times 1}$ is the location parameter, Σ is the covariance matrix, and ν is the degree of freedom. A smaller ν corresponds to a heavier tail, and when $\nu \to \infty$, t-distribution collapses to Gaussian distribution. By introducing a latent variable R, pdf of multivariate t-distribution can be decomposed as [80]:

$$\mathcal{S}t(x|\mu,\Sigma,\nu) = \int_0^{+\infty} \mathcal{N}(x|\mu,R^{-1}\Sigma)\mathcal{G}\Big(R|\frac{\nu}{2},\frac{\nu}{2}\Big)dR \tag{2.16}$$

where, $R > 0$ and the pdf of Gaussian and Gamma distribution are given, respectively, as follows:

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{2.17}$$

$$\mathcal{G}(R|\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} R^{\alpha-1}\exp(-\beta R) \tag{2.18}$$

Under this decomposition, the Student's t-distribution can be viewed as an infinite mixture of Gaussian distributions with the same mean and varying variances scaled by the scaling factor R which follows a Gamma distribution.

Proposed Formulation of RPSFA: In robust formulation, measurement noise is assumed to follow Student's t-distribution to account for the outlying values in observations. The robust formulation is given as follows:

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), & e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ x(t) = Hs(t) + e_x(t), & e_x(t) \sim \mathcal{S}t(\mathbf{0}, \Sigma, \nu) \end{cases} \tag{2.19}$$

where the definition of F, Λ and Σ are the same as that of PSFA shown in (2.13). Since all SFs are independent to each other, the state equation (2.19) can be further decomposed into individual components. Then j-th SF has following representation:

$$s_j(t) = \lambda_j s_j(t-1) + e_j(t), e_j(t) \sim \mathcal{N}(0, 1-\lambda_j^2), 1 \leq j \leq q \tag{2.20}$$

Each feature is formulated as an auto-regressive AR(1) process which is also governed by the Markov property and the initial state of SFs follows Gaussian distribution as in (2.10) while observations follow Student's t-distribution:

$$p(x(t)|s(t), H, \Sigma, \nu) = \mathcal{S}t(Hs(t), \Sigma, \nu) \tag{2.21}$$

Formulation of RPSFA satisfies the independent assumption of all SFs $\{s_j(t)\}$ and each SF also has zero mean and unit variance:

$$\mathbb{E}[s_j(t)] = 0, Var\{s_j(t)\} = 1, 1 \leq j \leq q \tag{2.22}$$

Applying (2.16), (2.21) can be further decomposed as:

$$p(x(t)|s(t), H, \Sigma, R(t)) = \mathcal{N}(Hs(t), \Sigma/R(t)) \tag{2.23}$$

$$p(R(t)|\nu) = \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2}) \tag{2.24}$$

Now that we have presented the problem formulation, next section presents the parameter estimation of the RPSFA model (2.19) using the EM algorithm.

2.4 Parameter Estimation of RPSFA Using the EM Algorithm

The parameter estimation and SFs extraction in PSFA have been solved under the maximum-likelihood estimation (MLE) framework [20, 47] using the EM algorithm [14]. Given a set of observation data D_o, the EM algorithm finds an optimal solution of unknown parameters Θ by iteratively maximizing the expectation of logarithm of the likelihood function (also known as Q-function) through expectation step (E-Step) and maximization step (M-Step) with respect to missing observations and hidden variables D_{hid} simultaneously. D_o and D_{hid} together form the complete data set D.

2.4.1 Parameterization in the EM algorithm: E-Step

Let complete data be composed of observations $D_o = \{x\} = \{x(1), \cdots, x(T)\}$ and latent variables $D_{hid} = \{s, R\} = \{s(1), \cdots, s(T), R(1), \cdots, R(T)\}$, where R is the variance scale factors that is used to decompose t-distribution in (2.23) and (2.24). Then, log-likelihood of the complete data can be written as:

$$\begin{aligned}\log P(X, s, R|\Theta) =& \log P(x|s, R, \Theta)P(s|R, \Theta)P(R|\Theta) \\ =& \underbrace{\sum_{t=1}^{T} \log P\big(x(t)|s(t), R(t), \Theta\big)}_{A} + \underbrace{\log P\big(s(1)|\Theta\big)}_{B} \\ &+ \underbrace{\sum_{t=2}^{T} \log P\big(s(t)|s(t-1), \Theta\big)}_{C} + \underbrace{\sum_{t=1}^{T} \log P(R(t)|\nu)}_{D}\end{aligned} \tag{2.25}$$

where the parameters to be estimated in RPSFA are denoted as $\Theta = \{\lambda_{1:q}, H, \Sigma, \nu\}$. Due to the decomposition presented in (2.23) and (2.24), the components in the under-braced term A follow the scaled Gaussian distributions (2.23), the components in the under-braced term B follow the Gaussian distributions (2.10), the components in the under-braced term C follow the Gaussian distributions (2.9), and the components in the under-braced term D follow the Gamma distribution (2.24). From here onwards, we enumerate each term in (2.25) sequentially.

Substituting (2.23) into term A, term A can be expanded as:

$$\sum_{t=1}^{T} \log P\big(x(t)|s(t), R(t), \Theta\big) \tag{2.26}$$

$$= -\frac{mT}{2}\log 2\pi + \frac{m}{2}\sum_{t=1}^{T}\log R(t) - \frac{T}{2}\log|\Sigma| - \frac{1}{2}\sum_{t=1}^{T}\Big(x(t) - Hs(t)\Big)^T \Sigma^{-1} R(t)\Big(x(t) - Hs(t)\Big)$$

Substituting (2.10) into term B, term B becomes:

$$\log P\big(s(1)|\Theta\big) = -\frac{q}{2}\log 2\pi - \frac{1}{2}s(1)^T s(1) \tag{2.27}$$

Substituting (2.9) into term C and using the property of decorrelation between different features (2.4), term C can be calculated as:

$$\sum_{t=2}^{T} \log P\big(s(t)|s(t-1), \Theta\big)$$

$$= -\frac{q(T-1)}{2}\log 2\pi - \frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t) - \lambda_j s_j(t-1)\Big)^2 \tag{2.28}$$

Substituting (2.24) into term D, then term D can be written as:

$$\sum_{t=1}^{T}\log P(R(t)|\nu) = \sum_{t=1}^{T}\Big[-\log\Gamma(\frac{\nu}{2}) + \frac{\nu}{2}\log\frac{\nu}{2} + \frac{\nu}{2}\Big(\log R(t) - R(t)\Big) - \log R(t)\Big] \tag{2.29}$$

Below, we present the complete data log-likelihood in (2.25) by taking summation of (2.26)$\sim$(2.29) and performing some straightforward algebraic manipulations:

$$\begin{aligned}
&\log P(X, s, R|\Theta) \\
=& -\frac{(m+q)T}{2}\log 2\pi - \frac{T}{2}\log|\Sigma| - \frac{1}{2}s(1)^T s(1) - \frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) \\
&- \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t) - \lambda_j s_j(t-1)\Big)^2 - \frac{1}{2}\sum_{t=1}^{T}\Big(x(t) - Hs(t)\Big)^T\Sigma^{-1}R(t)\Big(x(t) - Hs(t)\Big) \\
&- T\log\Gamma(\frac{\nu}{2}) + \frac{T\nu}{2}\log\frac{\nu}{2} + \Big(\frac{m+\nu}{2} - 1\Big)\sum_{t=1}^{T}\log R(t) - \frac{\nu}{2}\sum_{t=1}^{T}R(t)
\end{aligned} \tag{2.30}$$

Now we can compute the Q-function as:

$$\begin{aligned}
Q(\Theta|\Theta^{(n)}) =& \mathbb{E}_{s,R|x,\Theta^{(n)}}\Big\{\log P(X, s, R|\Theta)\Big\} \\
=& \mathbb{E}_{s|x,\Theta^{(n)}}\Big\{\mathbb{E}_{R|s,x,\Theta^{(n)}}\Big[\log P(X, s, R|\Theta)\Big]\Big\}
\end{aligned} \tag{2.31}$$

In order to maximize the Q-function, the posterior distribution of latent variable R needs to be derived using Bayes' theorem [6]:

$$\begin{aligned} & P(R(t)|s,x,\Theta^{(n)}) \\ =& \frac{P(x(t)|R(t),s(t),\Theta^{(n)})P(R(t)|s(t),\Theta^{(n)})}{P(x(t)|s(t),\Theta^{(n)})} \\ =& \mathcal{G}(\frac{\nu^{(n)}+d}{2}, \frac{\nu^{(n)}+\delta(x(t)|H^{(n)}s(t),\Sigma^{(n)})}{2}) \end{aligned} \tag{2.32}$$

As we can see in (2.32), the posterior distribution of R follows a Gamma distribution with different set of parameters. In (2.32), $\delta(x(t)|H^{(n)}s(t),\Sigma^{(n)})$ is the squared Mahalanobis distance between observation $x(t)$ and the Gaussian distribution with mean $H^{(n)}s(t)$ and variance $\Sigma^{(n)}$:

$$\delta(x(t)|H^{(n)}s(t),\Sigma^{(n)}) = \big[x(t)-H^{(n)}s(t)\big]^T\big(\Sigma^{(n)}\big)^{-1}\big[x(t)-H^{(n)}s(t)\big] \tag{2.33}$$

In the proceeding steps, we also use the fact that, if a random variable $x \sim \mathcal{G}(\alpha,\beta)$, then:

$$\mathbb{E}(x|\alpha,\beta) = \frac{\alpha}{\beta} \tag{2.34}$$

$$\mathbb{E}(\log x|\alpha,\beta) = \psi(\alpha) - \log\beta = \psi(\alpha) - \log\alpha + \log\alpha - \log\beta = \psi(\alpha) - \log\alpha + \log\frac{\alpha}{\beta} \tag{2.35}$$

where $\psi(\alpha)$ is the digamma function, that is, $\psi(\alpha) = \frac{d}{d\alpha}\ln(\Gamma(\alpha))$ [4, Chapter 7]. The expectation value of $R(t)$ can be computed as:

$$\mathbb{E}(R(t)|s(t),x(t),\Theta^{(n)}) = \frac{\nu^{(n)}+1}{\nu^{(n)}+\delta(x(t)|H^{(n)}s(t),\Sigma^{(n)})} \triangleq r^{(n)}(t) \tag{2.36}$$

and after applying property (2.35), we have,

$$\mathbb{E}(\log R(t)|s(t),x(t),\Theta^{(n)}) = \psi\Big(\frac{\nu^{(n)}+1}{2}\Big) - \log\Big(\frac{\nu^{(n)}+1}{2}\Big) + \log r^{(n)}(t) \tag{2.37}$$

Substituting a posteriori expressions (2.36), (2.37) and log-likelihood (2.30) into

(2.31), the Q-function can be rewritten as:

$$
\begin{aligned}
&Q(\Theta|\Theta^{(n)}) \\
=&\mathbb{E}_{s|x,\Theta^{(n)}}\Bigg\{ -\frac{(m+q)T}{2}\log 2\pi - \frac{T}{2}\log|\Sigma| - \frac{1}{2}s(1)^T s(1) - \frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) \\
&-\frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t)-\lambda_j s_j(t-1)\Big)^2 - \frac{1}{2}\sum_{t=1}^{T}\Big(x(t)-Hs(t)\Big)^T\Sigma^{-1}r^{(n)}(t)\Big(x(t)-Hs(t)\Big) \\
&-T\log\Gamma(\frac{\nu}{2}) + \frac{T\nu}{2}\log\frac{\nu}{2} + \Big(\frac{m+\nu}{2}-1\Big)\sum_{t=1}^{T}\Big[\psi\Big(\frac{\nu^{(n)}+1}{2}\Big) - \log\Big(\frac{\nu^{(n)}+1}{2}\Big) + \log r^{(n)}(t)\Big] \\
&-\frac{\nu}{2}\sum_{t=1}^{T}r^{(n)}(t)\Bigg\} \qquad (2.38)
\end{aligned}
$$

after performing suitable mathematical manipulations, Q-function in (2.38) can be represented as:

$$Q(\Theta|\Theta^{(n)}) = Q_1(\lambda_j) + Q_2(H,\Sigma) + Q_3(\nu) + C \qquad (2.39)$$

where,

$$Q_1(\lambda_j) = \mathbb{E}_{s|x,\Theta^{(n)}}\Bigg\{ -\frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t)-\lambda_j s_j(t-1)\Big)^2\Bigg\} \qquad (2.40)$$

$$Q_2(H,\Sigma) = \mathbb{E}_{s|x,\Theta^{(n)}}\Bigg\{ -\frac{T}{2}\log|\Sigma| - \frac{1}{2}\sum_{t=1}^{T}\Big(x(t)-Hs(t)\Big)^T\Sigma^{-1}r^{(n)}(t)\Big(x(t)-Hs(t)\Big)\Bigg\} \qquad (2.41)$$

$$
\begin{aligned}
Q_3(\nu) = \mathbb{E}_{s|x,\Theta^{(n)}}\Bigg\{ &-T\log\Gamma(\frac{\nu}{2}) + \frac{T\nu}{2}\log\frac{\nu}{2} - \frac{\nu}{2}\sum_{t=1}^{T}r^{(n)}(t) \\
&+\Big(\frac{m+\nu}{2}-1\Big)\sum_{t=1}^{T}\Big[\psi\Big(\frac{\nu^{(n)}+1}{2}\Big) - \log\Big(\frac{\nu^{(n)}+1}{2}\Big) + \log r^{(n)}(t)\Big]\Bigg\} \qquad (2.42)
\end{aligned}
$$

$$C = -\frac{(m+q)T}{2}\log 2\pi - \frac{1}{2}s(1)^T s(1) \qquad (2.43)$$

where, each term of the above equations is related to the corresponding unknown parameter. With this well formulated Q-function, the updated expressions of all parameters will be derived in the M-step and are presented in the next subsection. The expectation terms in E-step will be computed in a later subsection.

2.4.2 M-Step

The updated expression of each parameter is derived by computing the derivatives of the Q-function with respect to the corresponding parameter, thereby maximizing the Q-function. For simplicity, we use $\mathbb{E}(\cdot)$ instead of $E_{s|x,\Theta^{(n)}}(\cdot)$ in the following derivation.

Updating λ_j: Parameter λ_j is updated by taking derivative of $Q_1(\lambda_j)$ with respect to λ_j and equating it to zero:

$$\begin{aligned}
&\frac{\partial Q_1(\lambda_j)}{\partial \lambda_j} = 0 \\
=&\frac{\lambda_j(T-1)}{1-\lambda_j^2} - \frac{1}{(1-\lambda_j^2)^2}\Bigg\{\lambda_j \sum_{t=2}^{T}\mathbb{E}\Big[s_j^2(t) - 2\lambda_j s_j(t)s_j(t-1) + \lambda_j^2 s_j^2(t-1)\Big] \\
&+ \lambda_j(1-\lambda_j^2)\sum_{t=2}^{T}\mathbb{E}\Big[s_j^2(t-1)\Big] - (1-\lambda_j^2)\sum_{t=2}^{T}\mathbb{E}\Big[s_j(t)s_j(t-1)\Big]\Bigg\} \qquad (2.44)
\end{aligned}$$

Let us define:

$$\tau_1 \triangleq \sum_{t=2}^{T}\mathbb{E}\Big[s_j^2(t)\Big], \tau_2 \triangleq \sum_{t=2}^{T}\mathbb{E}\Big[s_j^2(t-1)\Big], \tau_{12} \triangleq \sum_{t=2}^{T}\mathbb{E}\Big[s_j(t)s_j(t-1)\Big] \qquad (2.45)$$

Then updating equation (2.44) can be simplified as:

$$(T-1)\lambda_j^3 - \tau_{12}\lambda_j^2 + (\tau_1 + \tau_2 - T + 1)\lambda_j - \tau_{12} = 0 \qquad (2.46)$$

The updated λ_j can be calculated by solving (2.46) while constraining the roots in the range [0,1). The expectation calculation in (2.45) can be obtained using Kalman Filter and is presented in later subsection.

Updating H: Parameter H is updated by taking derivative of $Q_2(H,\Sigma)$ with respect to H and equating it to zero:

$$\frac{\partial Q_2(H,\Sigma)}{\partial H} = 0 \Rightarrow H^{(n+1)} = \Bigg(\sum_{t=1}^{T} x(t)\mathbb{E}\big[r^{(n)}(t)s^T(t)\big]\Bigg)\Bigg(\sum_{t=1}^{T}\mathbb{E}\big[r^{(n)}(t)s(t)s^T(t)\big]\Bigg)^{-1} \qquad (2.47)$$

where, the terms $\mathbb{E}\big[r^{(n)}(t)s^T(t)\big]$ and $\mathbb{E}\big[r^{(n)}(t)s(t)s^T(t)\big]$ are calculated using Kalman Filter.

Updating Σ: Parameter Σ is updated by taking derivative of $Q_2(H, \Sigma)$ with respect to Σ and equating it to zero:

$$\begin{aligned}
&\frac{\partial Q_2(H,\Sigma)}{\partial \Sigma} = 0 \\
\Rightarrow &(\Sigma^T)^{(n+1)} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[r^{(n)}(t)x(t)x^T(t) - r^{(n)}(t)x(t)s^T(t)(H^T)^{(n+1)} \\
&- H^{(n+1)}r^{(n)}(t)s(t)x^T(t) + H^{(n+1)}r^{(n)}(t)s(t)s^T(t)(H^T)^{(n+1)}\Big]
\end{aligned} \tag{2.48}$$

As $(\Sigma^T)^{(n+1)} = \Sigma^{(n+1)}$:

$$\begin{aligned}
\Sigma^{(n+1)} = &\frac{1}{T}\sum_{t=1}^{T}\Big\{x(t)x^T(t)\mathbb{E}\Big[r^{(n)}(t)\Big] - x(t)\mathbb{E}\Big[r^{(n)}(t)s^T(t)\Big](H^T)^{(n+1)} \\
&- H^{(n+1)}\mathbb{E}\Big[r^{(n)}(t)s(t)\Big]x^T(t) + H^{(n+1)}\mathbb{E}\Big[r^{(n)}(t)s(t)s^T(t)\Big](H^T)^{(n+1)}\Big\}
\end{aligned} \tag{2.49}$$

where, terms $\mathbb{E}\Big[r^{(n)}(t)\Big]$, $\mathbb{E}\Big[r^{(n)}(t)s^T(t)\Big]$, $\mathbb{E}\Big[r^{(n)}(t)s(t)\Big]$ and $\mathbb{E}\Big[r^{(n)}(t)s(t)s^T(t)\Big]$ are calculated using Kalman Filter. We also notice that:

$$\mathbb{E}\Big[r^{(n)}(t)s^T(t)\Big] = \Big(\mathbb{E}\Big[r^{(n)}(t)s(t)\Big]\Big)^T \tag{2.50}$$

Updating ν: Parameter ν is updated by taking derivative of $Q_3(\nu)$ with respect to ν and equating it to zero:

$$\begin{aligned}
&\frac{\partial Q_3(\nu)}{\partial \nu} = 0 \\
\Rightarrow &-\psi(\frac{\nu}{2}) + 1 + \log(\frac{\nu}{2}) + \psi\Big(\frac{\nu^{(n)}+1}{2}\Big) - \log\Big(\frac{\nu^{(n)}+1}{2}\Big) + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\log r^{(n)}(t) - r^{(n)}(t)\Big] = 0
\end{aligned} \tag{2.51}$$

As (2.51) does not yield a closed form solution, numerical solution needs to be sought such that $\nu = [0, \infty]$.

2.4.3 Computation of Posteriors in the E-Step

In the E-Step, we need to calculate the posterior distributions of latent states $P(s|x, \Theta^{(n)})$. Hence, the following expectation terms that appear in (2.46) to (2.49) and (2.51) need

to be calculated:

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)\big] \tag{2.52}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)s^T(t)\big] \tag{2.53}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)s^T(t-1)\big] \tag{2.54}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[r^{(n)}(t)\big] \tag{2.55}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[\log r^{(n)}(t)\big] \tag{2.56}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[r^{(n)}(t)s(t)\big] \tag{2.57}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[r^{(n)}(t)s(t)s^T(t)\big] \tag{2.58}$$

As (2.36) indicates, $r^{(n)}(t)$ is the function of $s(t)$. Hence, the expectation terms in (2.55) $\sim$ (2.58) including $r^{(n)}(t)$, are also functions of $s(t)$. These terms cannot be calculated analytically in terms of posterior distribution $P(s|x,\Theta^{(n)})$ as they are intrinsically related. However, we approximate these terms by considering $r^{(n)}(t)$ as constant in terms of $s(t)$, since the slowly varying features $s(t-1)$ and $s(t)$ are expected to be numerically very close to each other, thereby allowing us to treat $r^{(n)}(t)$ as a constant. This also helps to take $r^{(n)}(t)$ out of expectation operator, as applicable. Further, validations on process case studies also reveal that the approximation of $r^{(n)}(t)$ to be a constant is well justified as indicated in Section 2.5.1. After the approximation, we only need to calculate terms (2.52) to (2.54) and parameters update equations (2.47), (2.49) and (2.51) can be simplified as follows:

$$H^{(n+1)} = \Big(\sum_{t=1}^{T} x(t)r^{(n)}(t)\mathbb{E}\big[s^T(t)\big]\Big)\Big(\sum_{t=1}^{T} r^{(n)}(t)\mathbb{E}\big[s(t)s^T(t)\big]\Big)^{-1} \tag{2.59}$$

$$\begin{aligned}\Sigma^{(n+1)} =& \frac{1}{T}\sum_{t=1}^{T}\Big\{x(t)x^T(t)r^{(n)}(t) - x(t)r^{(n)}(t)\mathbb{E}\big[s^T(t)\big](H^T)^{(n+1)} \\ &- H^{(n+1)}r^{(n)}(t)\mathbb{E}\big[s(t)\big]x^T(t) + H^{(n+1)}r^{(n)}(t)\mathbb{E}\big[s(t)s^T(t)\big](H^T)^{(n+1)}\Big\}\end{aligned} \tag{2.60}$$

$$-\psi(\frac{\nu}{2})+1+\log(\frac{\nu}{2})+\psi\Big(\frac{\nu^{(n)}+1}{2}\Big)-\log\Big(\frac{\nu^{(n)}+1}{2}\Big)+\frac{1}{T}\sum_{t=1}^{T}\Big(\log r^{(n)}(t)-r^{(n)}(t)\Big)=0 \tag{2.61}$$

The estimation of the dynamic latent states $s(t)$ requires two steps: first, Kalman filtering, so called forward recursion and second, Kalman smoothing, so called back-

ward recursion [73], as they follow state space dynamics (2.19). The Kalman filtering step requires the posterior distribution to be Gaussian and the following posterior distribution can be obtained in forward recursion:

$$P(s(t)|x(1),\cdots,x(t),\Theta^{(n)}) \sim \mathcal{N}(\mu(t),V(t)) \tag{2.62}$$

For the calculation of terms in (2.52) $\sim$ (2.54), if noise follows the Gaussian distribution, the Kalman filter and smoother framework can be used to optimally estimate the value of latent states $s(t)$ [73]. As the Student's t-distribution assumption of noise violates the Gaussian assumption, the Kalman filter cannot be employed to do the same. Hence, we suitably modify the existing Kalman filter framework employing a weighted Kalman gain approach to account for the outlying information. We introduce a correction factor $w(t)$ to the Kalman filter gain to adjust distribution mismatch, as follows:

$$\begin{aligned} w(t) &= \frac{P_{Gauss}\big(x(t)|\mu(t-1),\Theta^{(n)}\big)}{P_t\big(x(t)|\mu(t-1),\Theta^{(n)}\big)} \\ &= \frac{\mathcal{N}\big(x(t)|HF\mu(t-1),\Sigma\big)}{\mathcal{S}t\big(x(t)|HF\mu(t-1),\Sigma,\nu\big)} \end{aligned} \tag{2.63}$$

where $P_{Gauss}(\cdot)$ and $P_t(\cdot)$ represent the probability density of measurement $x(t)$ assuming the Gaussian distribution and the Student's t-distribution, respectively, with currently estimated parameters. Now, the new weighted gain is calculated as: $K_w(t) = w(t) \cdot K(t)$, where $K(t)$ is Kalman gain [81]. Kalman gain is larger when the measurements have smaller variance or the estimates have larger variance, and vice versa. The weight $w(t)$ defined in (2.63) will attribute a small weight to the gain if the observation lies farther away from the mean. That is to say, weighted Kalman gain will further compensate the normal observations by giving heavier weights to them and giving lighter weights to outliers complying with the principle of Kalman filter. Therefore, we use the weighted gain $K_w(t)$ as a replacement of $K(t)$ in Kalman filtering to obtain the parameters by sequentially executing following steps:

$$\begin{aligned} P(t-1) &= FV(t-1)F^T + \Lambda \\ \mu(t) &= F\mu(t-1) + K_w(t)\big[x(t) - HF\mu(t-1)\big] \\ V(t) &= \big[\mathbf{I} - K(t)H\big]P(t-1) \\ K_w(t) &= P(t-1)H^T\big[HP(t-1)H^T + \Sigma\big]^{-1} \end{aligned} \tag{2.64}$$

with the initial conditions:

$$\begin{aligned} \mu(1) &= K_w(1)x(1) \\ V(1) &= \mathbf{I} - K_w(1)H \\ K_w(1) &= H^T\big[HH^T + \Sigma\big]^{-1} \end{aligned} \tag{2.65}$$

After forward recursion, we need to find the posterior distribution of $s(t)$ given all the observations x by performing backward recursion as [73]:

$$\begin{aligned} \hat{\mu}(t) &= \mu(t) + J(t)\big[\hat{\mu}(t+1) - F\mu(t)\big] \\ \hat{V}(t) &= V(t) + J(t)\big[\hat{V}(t+1) - P(t)\big]J^T(t) \\ J(t) &= V(t)F^T P(t)^{-1} \end{aligned} \tag{2.66}$$

The backward recursion is initialized by the value calculated in the last step in forward recursion: $\hat{\mu}(T) = \mu(T)$ and $\hat{V}(T) = V(T)$. After performing the forward and backward recursions, the evaluation of expectation terms in (2.52) to (2.54) can be calculated as follows [73]:

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)\big] = \hat{\mu}(t) \tag{2.67}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)s^T(t)\big] = \hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t) \tag{2.68}$$

$$\mathbb{E}_{s|x,\Theta^{(n)}}\big[s(t)s^T(t-1)\big] = J(t-1)\hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t-1) \tag{2.69}$$

With this modified weighted Kalman gain approach and Kalman smoother, parameter updating equations (2.46) to (2.49) and (2.51) are iteratively solved to obtain the updated parameters. The latent state SFs can be inferred as the mean value $\mu(t)$ of the conditional probability distribution in (2.62).

2.4.4 Regression Model Based on SFs

After extracting the SFs, regression models can be built on a selected subset of the derived SFs. The SFs used for regression are selected based on their slowness as the slowly varying features carry more process information while the fast varying features are noise manifestations. After computing q-dimensional SFs and their corresponding slownesses $\lambda_j, 1 \leq j \leq q$, we select p ($< q$) slowest features $s_{1:p}(t)$ as the predictors to

build a linear regression model in following form:

$$y(t) = B^T s_{1:p}(t) + a \tag{2.70}$$

where B and a are model regression coefficients, which can be obtained using ordinary least square (OLS) algorithm. The SFs used for building regression model are selected according to their slowness λ_j and the trend of the SFs. The Chosen criteria are: (1) Slowness: if there is a sharp drop of the slowness, it basically means the remaining SFs contain obvious fast SFs. Then we can choose all SFs until the sharp dropping point; (2) Trend: we can monitor the trend of the SFs. Sometimes even the SFs contain relatively noisy fast SFs but they still have clear trends which contain system dynamic information and can therefore be still included in the regression model. Also, it is not difficult enumerate all possible combinations and choose the one with best regression performance.

In the next section, a simulation example, an industrial application and an experimental case study are employed to show the efficacy of the proposed approach.

2.5 Case Studies

2.5.1 Simulation Example: Tennessee Eastman Process

In this section, Tennessee Eastman (TE) process data is employed to verify the effectiveness of proposed RPSFA algorithm. The employed TE data is from [82]. As given below, five variables are chosen as inputs and composition of reactor feed A as output.

x_1 :Reactor Pressure (kPa)

x_2 :Stripper Temperature (degC)

x_3 :Stripper Steam Flow (kg/h)

x_4 :Compressor Power (kw)

x_5 :Component C in Purge Gas (Mole %)

y :Component A in Reactor Feed (Mole %)

In this simulation, certain percentage of outliers are added in the input variables. These outliers will apparently affect the model identification and prediction perfor-

mance. We consider two simulation cases. In case 1, we perform simulations by adding certain percentage (10%) outliers to investigate the nature of extracted SFs, the convergence of degree of freedom ν in RPSFA and the prediction results. In case 2, Monte Carlo simulations are conducted to investigate the prediction performances under different percentages of outliers.

2.5.1.1 Case 1: Input Variables with 10% Outliers

In this case, 10% outliers are randomly added to each dimension of input variables. Figure 2.1 shows the input variables contaminated by outliers, which lie beyond the range of three standard deviation of the normal operating data.

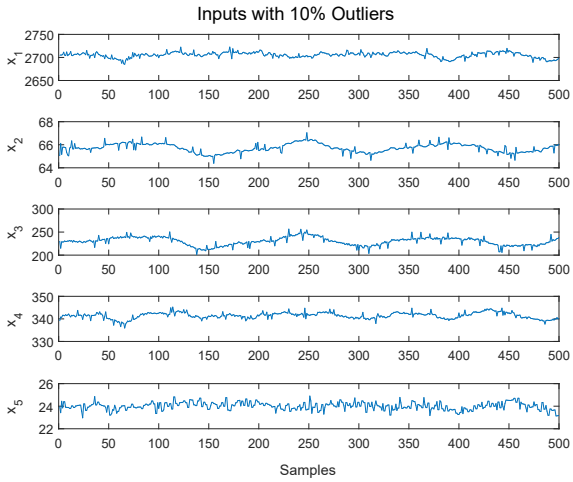

Figure 2.1: Inputs with 10% outliers on each dimension

With these input variables, we can extract SFs by PSFA and RPSFA, respectively as shown in Figure 2.2. The sub-figures in left column are SFs derived by PSFA and right column by RPSFA. Only the first three slowest features for each algorithm are shown for the sake of brevity in the illustration. As we can see, SFs from PSFA are badly impacted by outliers and SFs derived from RPSFA are robust hence less sensitive to the outliers. These first three slowest features for each algorithm will be used to build regression model to predict the output shortly.

The degree of freedom ν of Student's t-distribution controls the size of the tails and as iterations in EM algorithm go along, ν will converge to a value which reflects the extent of the outliers in inputs. Figure 2.3 shows convergence profile of the ν for

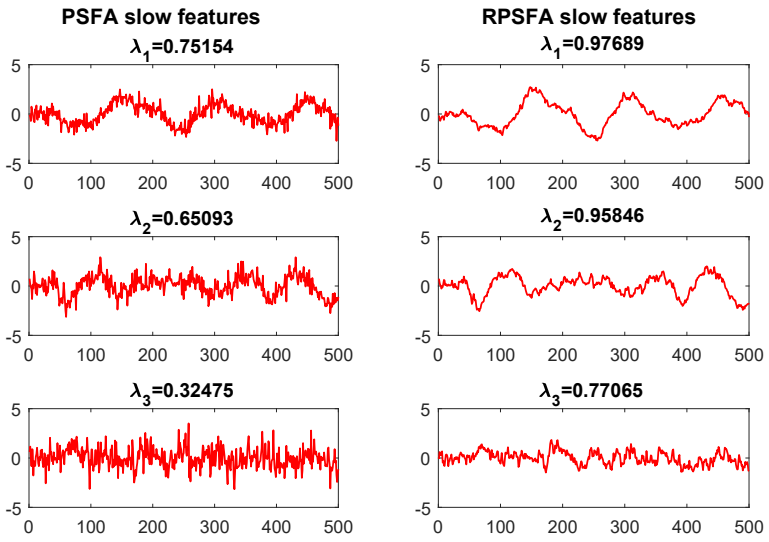

Figure 2.2: Comparison of SFs derived by PSFA and RPSFA λ_i represents the slowness of according feature. The larger λ_i, the slower the feature is.

the data set considered and we can see that ν eventually converges to a value around 16.4, indicating the extent of the outliers (infinity implies no outliers), hence the data quality of the input variables. It also justifies our choice of Student's t-distribution to model the outlier contaminated data.

Further, with the derived SFs from PSFA and RPSFA, we can build a SF regression model as discussed in section 4:

$$y(t) = B^T s_{1:3}(t) + a \tag{2.71}$$

After getting PSFA based regression (PSFR) and RPSFA based regression (RPSFR) models, we compare the performance of these two models with other models that are widely used in literature, namely, Multivariate Linear Regression (MLR), Principal Component Regression (PCR) and Partial Least Square (PLS) models. In PCR and PLS, the dimension of latent space is chosen to obtain the best prediction performance. Figure 2.4 shows the prediction results of all the five approaches. The corresponding mean square error (MSE), Pearson correlation coefficients (Corr) and Concordance correlation (ρ_c) between the predicted output and real output are also reported in each sub-figure. From these results, we can conclude that RPSFR outperforms other algorithms in terms of prediction accuracy as indicated by the smallest

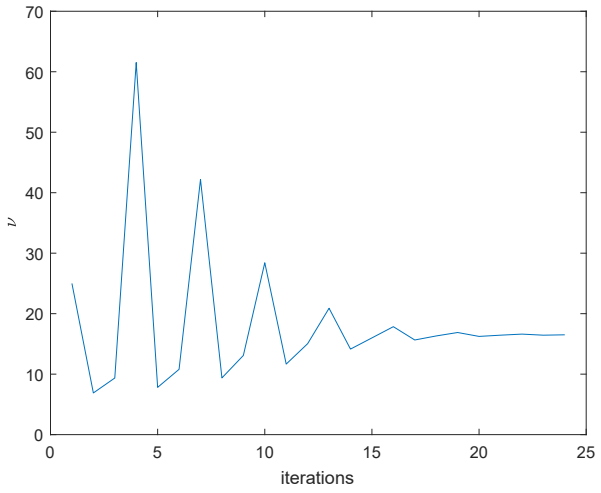

Figure 2.3: degree of freedom converges along iterations

MSE and the largest correlation.

Further, we also test the validity of the assumption of $r^{(n)}(t)$ being constant. According to the definition of $r^{(n)}(t)$ in (2.36), $r^{(n)}(t)$ is the function of $s(t)$ and $x(t)$, and all other terms in the definition are constant values which are calculated from the last iteration in the EM algorithm. Since we want to extract the slow features, we expect that $s(t)$ will change slowly. That is to say, the value of $s(t)$ will not change too much from the value of $s(t-1)$, the value from the previous time point. The variation in $r^{(n)}(t)$ comparing to $r^{(n)}(t-1)$ is mainly caused by the variation of $x(t)$ rather than $s(t)$. This can be verified by Figure 2.5. We manually added 3% outliers on each channel of inputs. Figure 2.5 shows the variation of $r^{(n)}(t)$ and input variables. As we can see, $r^{(n)}(t)$ remains constant in most of the time except at the time points in which outliers appear in the inputs. Also, comparing $r^{(n)}(t)$ in Figure 2.5 and Figure 2.6, it can be proved that variation of $r^{(n)}(t)$ is of less fluctuation compared to $s(t)$. Further, the SFs from RPSFA are smoother than that of conventional PSFA and they do not contain many abrupt changes which can cause variations in $r^{(n)}(t)$. In summary, the variation in $r^{(n)}(t)$ is mainly caused by the outliers in input variables and it remains constant when there are no outliers as SFs derived from RPSFA are smoother and slower. In addition, the value of $r^{(n)}(t)$ reflects the data quality. When outliers appear, as shown in Figure 2.5, $r^{(n)}(t)$ will increase the variance of the Gaussian distribution

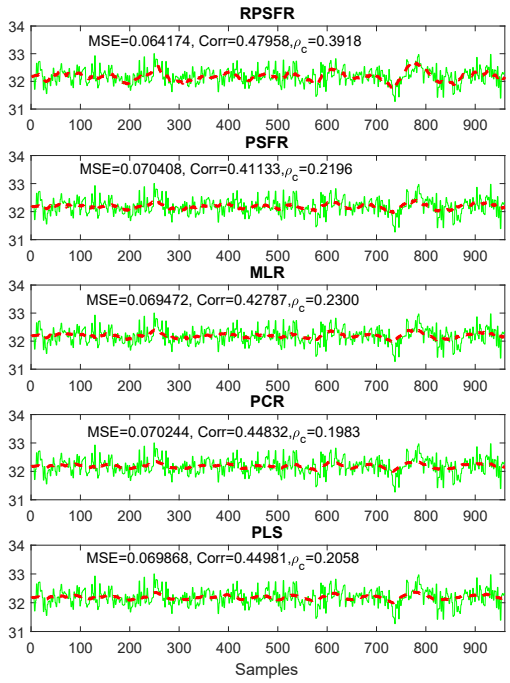

Figure 2.4: Comparison of prediction results of different algorithms for TE process. Green lines represent real values and red dot lines represent prediction values

that is decomposed from the student's t-distribution in (2.16), hence yields heavier tails. Thus, the impact of outliers is absorbed by the latent variable R.

2.5.1.2 Case 2: Monte Carlo Simulation Under Different Percentage of Outliers

In order to compare the performance of each method in presence of different percentage of outliers, we conduct simulations under eight different scenarios, i.e. 1%, 3%, 5%, 8%, 10%, 12%, 15% and 20% outlier contaminated data. Under each scenario, 20 Monte Carlo simulations are performed since the generation of outliers would be random. For each t stamps, the occurrence of the outliers are totally random and the magnitude of the generated outliers is within the range $[2\sigma,\ 3.5\sigma\ ]$, where σ is the standard deviation of the input to which the outliers are added. The average values of MSE and Corr are also calculated. Figure 2.7 and Figure 2.8 show that Corr for all five algorithms decrease and MSE increase when we increase the percentage of outliers as more outliers in the data make these models less accurate. However, simulation results indicate that, when comparing between these algorithms, under

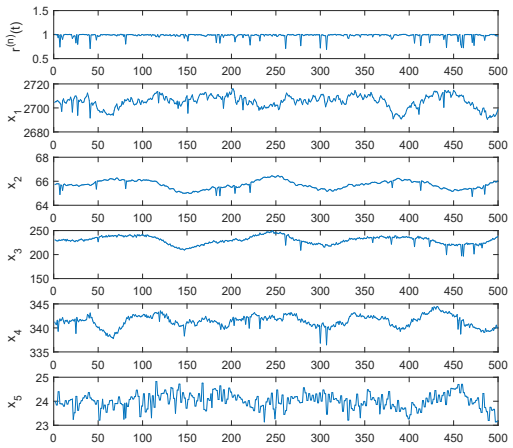

Figure 2.5: $r^{(n)}(t)$ variations

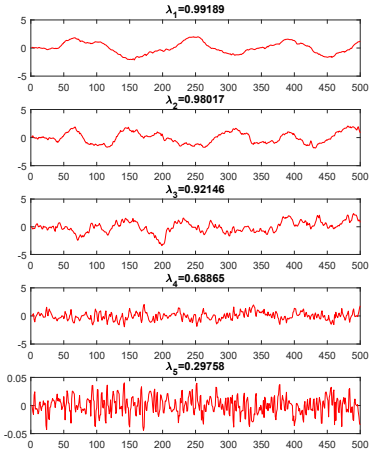

Figure 2.6: SFs extracted by RPSFA when 3% outliers

different outliers percentages, RPSFR has superior performance to others, showing its ability to handle outliers.

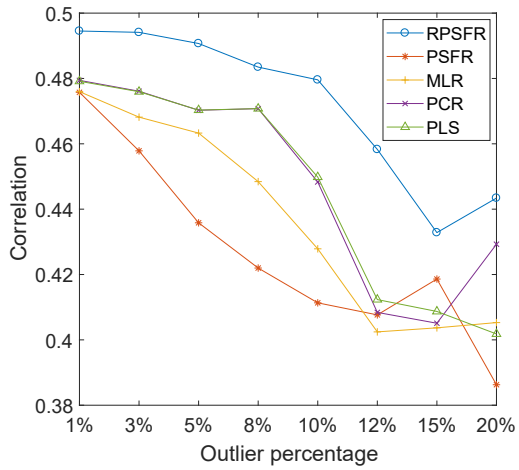

Figure 2.7: Monte Carlo simulation: comparison of Corr between predicted output and real output

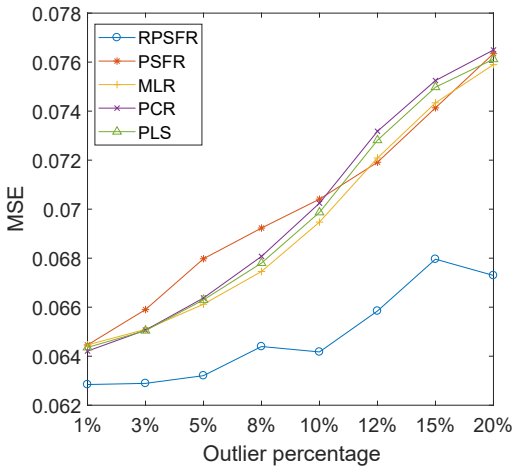

Figure 2.8: Monte Carlo simulation: comparison of MSE between predicted output and real output

We can also observe the profile of converged ν under different percentage of outliers in Figure 2.9. As the number of outliers increases, the converged ν decreases since the Student's t-distribution needs to have heavier tails to describe larger outliers in which heavier tail corresponds to a smaller value of ν.

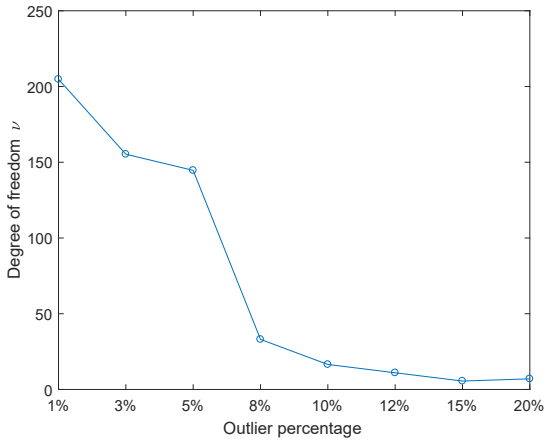

Figure 2.9: Monte Carlo simulation results: degree of freedom variation trend

2.5.2 Industrial Case Study: SAGD Water Content Soft Sensor

2.5.2.1 System Introduction

In this section, we develop a soft sensor for the water content measurement of one production well in a SAGD well pair to further demonstrate the efficacy of the proposed algorithm. Alberta province in Canada is the largest producer of heavy crude oil from oilsands. SAGD process is a novel in-situ oil recovery technology for bitumen extraction by employing steam injection. Figure 2.10 is the process flow diagram of a well pair in a typical oil extraction section of SAGD process. For each well pair, two wells are drilled into the underground. The injection well (upper well) is used to inject steam into the underground reservoir to heat the oilsands and to make the oilsands in the chamber area softer. Then the heated emulsion (mixture of oil and water) is drained into the production well (lower well) and pumped out from the chamber. The pumped out emulsion contains water due to the condensation of the injected steam. Accurate real time measurement of water content at the production wellheads will help to improve the downstream separation performance. It can be measured relatively accurately by an on-line analyzer called VX meter, which is a costly hardware. Hence, there is an interest to develop a soft sensor to infer the water content. The objective of soft sensor development is to provide an accurate and real-time estimation of this quality variable in an economical and viable manner.

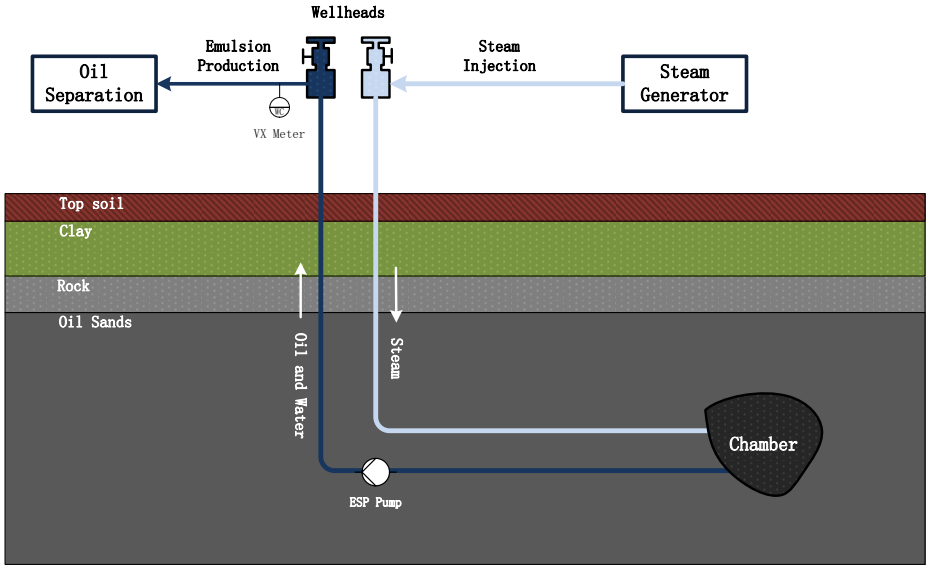

Figure 2.10: Process diagram of SAGD well pair

By performing correlation analysis, seven influential variables are selected as input variables for predictive model. After data preprocessing, the water content and input variables are represented as 5-hour averaged values, which can be considered as applying mean filter on the raw data. We can see from Figure 2.11 that all the inputs x and output water content are noisy and some of the inputs contain outliers as well even after taking data average, i.e. x_2 and x_4. Those outliers that are left out after averaging are to be handled by RPSFA. For proprietary reasons the attributes of input variables are not disclosed and all data have been normalized.

2.5.2.2 Model Development and Prediction Results

In order to compare the prediction results of proposed algorithm with PSFA and other regression models, we need to first extract SFs and build regression models using extracted SFs. Left column in Figure 2.12 shows the SFs extracted with the corresponding λ values and we can observe that RPSFA successfully eliminates most noise and outliers as seen in the first three SFs, compared to PSFA.

Next, regression models of the form (2.71) are built with the extracted SFs for prediction. We select the first four SFs for building PSFR and RPSFR models and

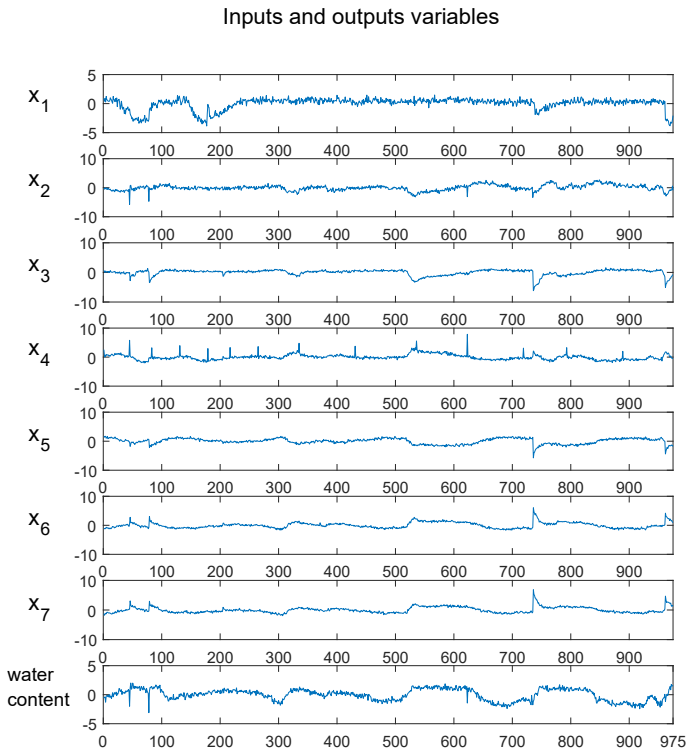

Figure 2.11: Seven input variables and water content

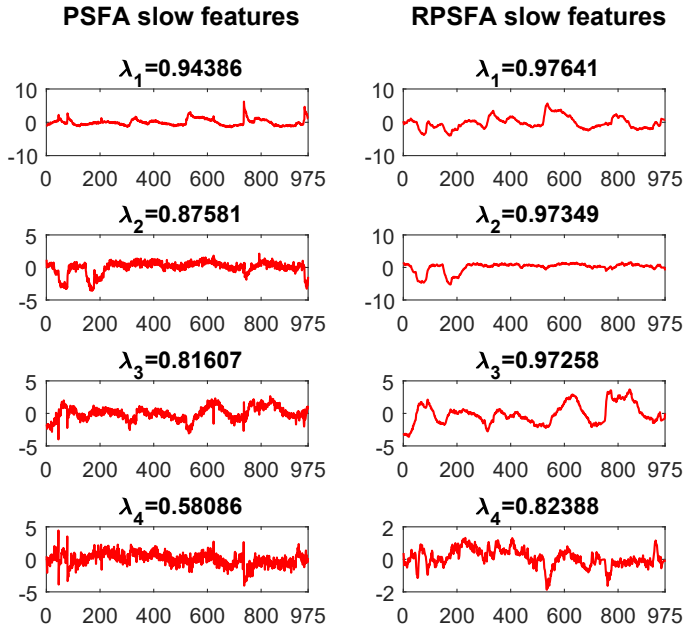

Figure 2.12: SFs comparison for PSFA and RPSFA

compare the prediction results with MLR, PCR and PLS. Figure 2.13 shows the prediction results of five methods, and the corresponding MSE and correlation values are also listed in each sub-figure. It is apparent that RPSFR outperforms other methods in all MSE and correlation indices as it has the smallest MSE and largest correlation among all five algorithms. The Corr of RPSFR exceeds 90%, which means it can capture the trend of the water content measurements fairly well, meanwhile restraining the impact of outliers.

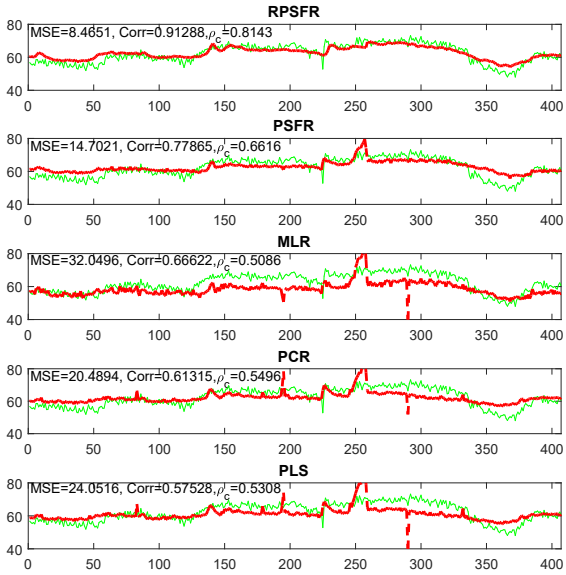

Figure 2.13: Prediction results of water content in for one SAGD well pair. Green lines represent real values and red dot lines represent prediction values

2.5.3 Experimental Case Study: Hybrid Tanks System

2.5.3.1 Hybrid Tanks System Configurations

In this section, an experiment is conducted using the hybrid tanks apparatus to further prove the efficacy of the robust PSFA algorithm. The hybrid tanks system, shown in Figure 2.14, is located in the process control laboratory in University of Alberta. The system is consist of:

- Three vertical tanks: left tank, mid-tank and right tank
- Storage tank: storage of the water and the water can flow into or out of the vertical tanks

- Two inlet pumps: left pump can fill water from storage tank into left tank and right pump into right tank
- Nine valves V1 $\sim$ V9: V1 $\sim$ V4, V6 and V8 are interconnection valves, which are used to connect mid-tank with side tanks. Water can flow between each tanks through interconnection valves. V5, V7 and V9 are outlet valves that connect vertical tanks and storage tank

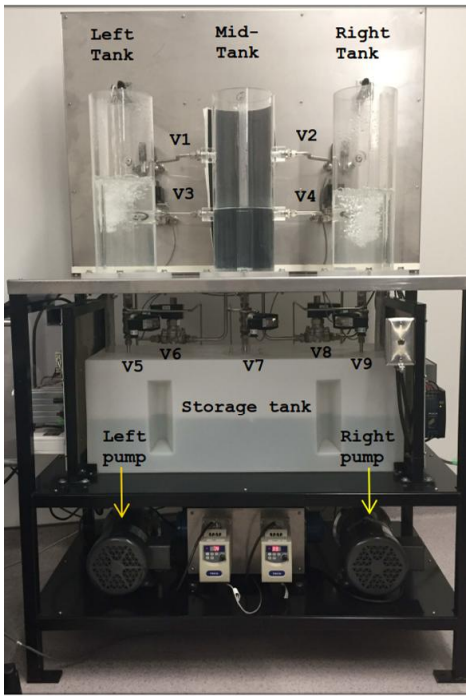

Figure 2.14: Hybrid tanks system

In the experiment, V1 and V2 are kept closed and the rest of the valves are kept open. To constrain the process, we have controlled the mid-tank level at around 48%, which is between V1 and V3. During the experimental process, the mid-tank level was never allowed to fall below 40%, the height of V3 and V4. The mid-tank level is controlled simultaneously by two cascade control loops, and in each control loop, the primary level controller's output is the set point of pump outlet flow controller, which works as the slave controller. Our desired variable to be predicted is the mid-tank level. We choose eight input variables in the process: two side tank levels, two pump outlet flow rates, two pump speeds and two slave controller outputs which set the pump speeds

and are shown in Figure 2.15. Due to the sensor problems of both pump speeds and flow rates, we observe a lot of outlying measurements as indicated in Figure 2.15, which makes this case study suitable for the validation of RPSFA algorithm. The complete data set has 4320 samples and we use the first 2160 samples for training and last 2160 samples for testing.

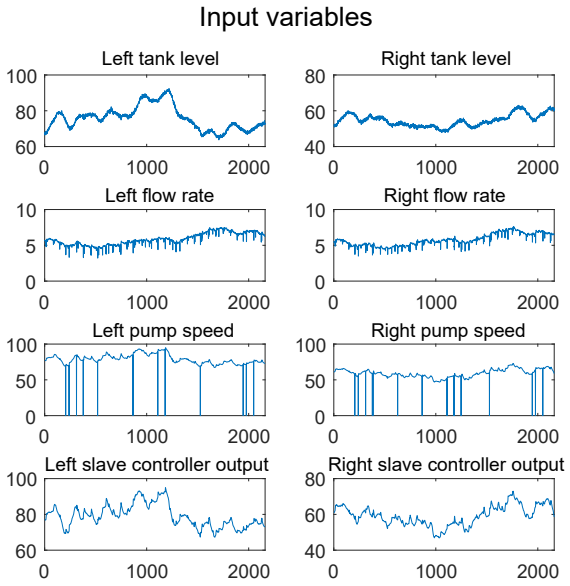

Figure 2.15: Input variables for Hybrid Tanks system

In order to compare the differences between the SFs extracted from RPSFA and PSFA, six SFs from each algorithm are presented in Figure 2.16, from which we can see, $s_1(t)$ and $s_2(t)$ of both algorithms are very slow and they all have the similar slowness. $s_{3:5}(t)$ of PSFA contain either more noise or outliers compared to that of RPSFA. The slowness of $s_6(t)$ has a sudden drop for both algorithms. Hence, we choose $s_{1:5}(t)$ of RPSFA as predictors to build regression model. For the selection of SFs of PSFA, we tried following three scenarios: (1) Only use $s_{1:3}(t)$ as predictors since they are slower and smoother, and do not contain obvious outliers and noise; (2) Exclude $s_4(t)$ and use $s_{1:3,5}(t)$ as predictors since $s_4(t)$ contains obvious outliers, which may impact the model accuracy; (3) Use $s_{1:5}(t)$, since $s_{4,5}(t)$ contain outliers and noise, the useful information carried by them may be favorable for prediction performance. After testing above three scenarios, adoption of $s_{1:5}(t)$ yields the best prediction results, which will be shown together with the prediction performance of

RPSFA, MLR, PCR and PLS in the next sub-section.

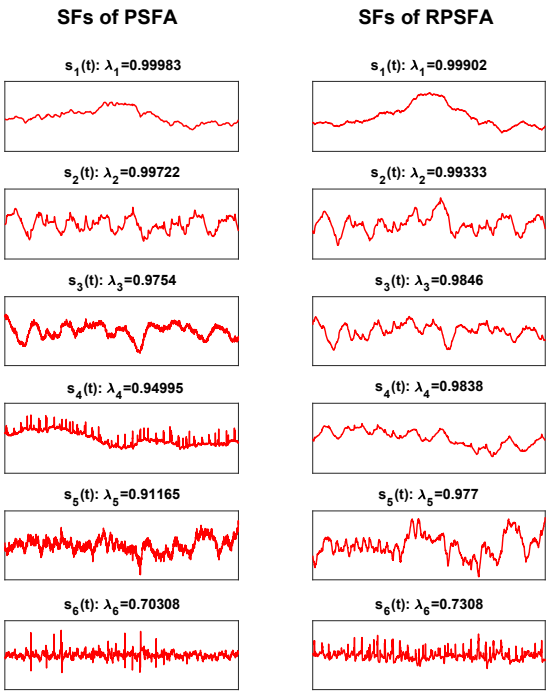

Figure 2.16: SFs extracted from process variables for PSFA and RPSFA

2.5.3.2 Prediction Results

In this sub-section, we compare the prediction performance of models built based on RPSFR, PSFR, MLR, PCR and PLS algorithms. The SFs $s_{1:5}(t)$ are selected for both RPSFR and PSFR as stated above. The number of principal components of PCR and PLS is selected by enumerating all possible number of components and choosing the set that shows the best results. For comparison purposes, we still use MSE, Corr and ρ_c as evaluation indices. The comparison results are shown in Figure 2.17 and the corresponding MSE and correlation values are also indicated on each sub-figure. As indicated by MSE and correlation, RPSFR outperforms other methods in terms of prediction accuracy. Visually, we can also see the prediction results of RPSFR are more accurate and smoother than other methods. It also proves the validity of our belief that as the real process dynamics often change slowly, the extracted SFs can effectively represent the information carried by output variables.

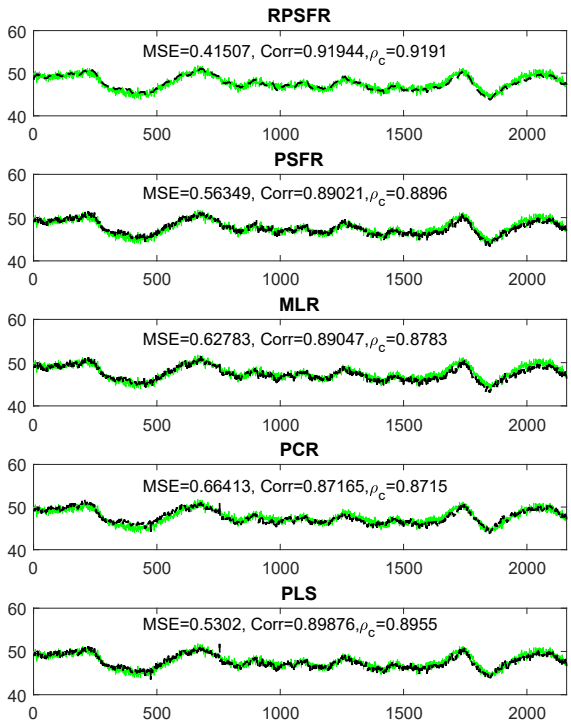

Figure 2.17: Hybrid tanks experiment: prediction results and performance comparison of RPSFA, PSFA, MLR, PCR and PLS. Green lines represent real values and black dot lines represent prediction values

2.6 Conclusions

In this chapter, we have proposed a robust PSFA for the modeling of dynamics and high dimensional data that contains outliers by modelling the measurements noise as the Student's t-distribution. RPSFA can induce slower and smoother latent features in presence of outliers, compared to the existing PSFA in literature. The parameters of the RPSFA are estimated using EM algorithm by introducing variance scale as hidden variable. A weighted gain Kalman filter is proposed to estimate the hidden states, as the observations follow the Student's t-distribution. Then models are built by regressing the extracted features with the measured outputs. Simulation results on various case studies have demonstrated the superiority of the proposed approach.

Chapter 3

Semi-supervised Dynamic Latent Variable Modeling: I/O Probabilistic Slow Feature Analysis Approach *

Modeling of high dimensional dynamic data is a challenging task. The high dimensionality problem in process data is usually accounted for using latent variable models. Probabilistic slow feature analysis (PSFA) is an example of such an approach that accounts for high dimensionality while simultaneously capturing the process dynamics. However, PSFA also suffers from a drawback that it cannot use output information when determining the latent slow features. To address this lacunae, extension of the PSFA by incorporating outputs, resulting in Input-Output PSFA (IOPSFA) is proposed. IOPSFA can use both input and output information for extracting latent variables. Hence, inferential models based on IOPSFA are expected to have better predictive ability. The efficacy of the proposed approach with an industrial and a laboratory scale soft sensing case studies that have both complete and incomplete output measurements is evaluated, respectively.

*Part of this chapter has been published as: Fan L, Kodamana H, Huang B. Semi-supervised dynamic latent variable modeling: I/O probabilistic slow feature analysis approach. AIChE Journal. 2019 Mar;65(3):964-79.

3.1 Introduction

Nowadays, inferential models are widely used in chemical processes in lieu of hard-wired sensors because many important quality variables, whose information is critical to the process, are difficult to measure online. Quality variables are those variables that define the quality or the specification of the product and usually measured by using online analyzer or offline laboratory analysis. Even though online analyzers can provide real-time values, they may be unreliable or expensive to use. Laboratory analysis can provide more accurate measurements but it normally encompasses large sampling intervals and introduces large and uncertain time delays. When the laboratory data is available, the process variables at the same time instance are considered to be labelled. Inferential model describes the relationship between desired quality variables and easily measurable process variables using first principles or historical data [83]. Building a trustable first principles model for real-time inference requires in-depth understanding of underlying physics, which is often difficult, especially for complex chemical processes. Under such circumstances, data-driven methods for inferential modeling have become a popular choice. The advent of improved methods of data collection and warehousing has enabled chemical plants to archive large amount of historical data in their databases. These data contain useful information about process dynamics and relationships between various process and quality variables, and therefore are useful in developing data based inference of key quality variables [84]. Popular latent variable approaches for data based inferential modeling include: Principal Component Analysis/Regression (PCA/PCR) [48, 85], Probabilistic PCR (PPCR) [86, 87], Partial Least Squares (PLS) [40, 54, 56], among others. To enhance the modeling capability, semi-supervised models, which can use both labelled and unlabelled data, have been proposed by various researchers for inferential modeling [88] and process monitoring purposes [89]. A detailed treatment of various semi-supervised learning appraoches, e.g. transductive support vector machine, self-training, entropy regularization, graph-based models are presented in [90, 91].

Missing data is a key factor that influence the development of empirical inferential models. Missing data problem is an common problem in industrial settings such as sensor failure, communication interruption, difficulty in obtaining the measurements,

among others. It has significant impact on the modeling and finally the conclusions that are drawn based on data. Normally, inferential modeling requires complete set of input and output data, that is, every input sample needs to be labelled with a corresponding output sample. Practically, quality variables that are measured by laboratory analysis can only be sampled at a different (normally larger) interval, compared to other process variables that are sampled faster. Besides this multi-rate problem, some samples may also be randomly missing due to sensor failure or transmission problem [1]. Since these two cases possess different patterns of data missing, these different patterns will lead to different ways of estimating the missing data [92]. One thing in common in both cases is that, we can only obtain partial samples of outputs, which imposes challenges in building inferential models. The multi-rate problem [93], where the data missing follows a certain pattern, can be treated as a special case of missing data problem. Many approaches have been proposed in literature for developing models in the presence of missing data [92, 94, 95], e.g. to handle missing data using PCR/PLS [96], maximum likelihood approaches [97–99], regression based approaches [100, 101], multi-resolution approaches [102] and uncertainty-based approach [98, 103]. Among them, Expectation-Maximization (EM) algorithm [14] is one of the most commonly employed methods, for instances, the readers can refer to references [104–106].

To accurately infer the quality variables, we need to extract the essential and useful dynamical information from available process variables. Usually, process variables may contain redundant information and show similar correlation with the desired quality variables. On one hand this redundancy may prove to be useful in some applications; however it increases the computational load while handling many variables and may introduce ill condition problem. Hence, researchers have employed many robust ways to extract essential information from data, e.g. multiscale approaches [107–110], Subspace State-Space (SSS) [111–113] and latent variable models [114], etc. As the industrial systems become more complicated, the collected data also presents multiscale nature of the systems, the interacting elements from which can vary from fine scales to coarse scales. To address this challenge, several multiscale modeling approaches have also been developed in [107–110]. Due to issues related to data handling and warehousing, we need to compress available data to

capture essential information. Among the above mentioned approaches, latent variable models (LVM) [115] have the unique ability to compress the redundant input information that leads to the reduction of the dimensionality. LVMs can identify and extract useful and exclusive information using the reduced dimension latent variables, leading to computational benefits. These latent variables can also be treated as the common sources of the input variables and quality variables [116]. The extracted latent variables are the representations of the observations in favor of certain features of data, e.g. PCA extract latent variables in terms of maximum variance and slow feature analysis (SFA) in terms of the slowness of latent variables, etc. Traditionally, latent variables are extracted by latent variables models only from input variables, for instances, PCA [72], PPCA [44], SFA [43], PSFA [47], independent component analysis (ICA) [41, 42], singular spectrum analysis (SSA) [110, 117, 118], among others. Among the listed above, PLS can extract latent variables using both input and output information. However, being a static model, PCA or PLS cannot describe the dynamics in variables. To address this issue, various useful extensions are proposed for PCA as Dynamic PCA (DPCA) [119, 120]. DPCA incorporates the dynamic information by using lagged observations with limited window length [119], thereby compromising on the computational efficiency. In contrast to that, SFA and its probabilistic counterpart PSFA model the system dynamics by extracting temporally and slowly varying latent variables, so called slow features (SFs) [20]. This enables them to capture all the dynamic information contained in observations during the modeling process, resulting in a reduced dimensional model that contains less noise since noise is normally included in the fast varying features. The main difference between PSFA and the other methods to extract process dynamical trends, for example, SSS, is that PSFA includes probabilistic interpretation and the prior of process knowledge by constraining the features to have large inertia, which are more informative for modeling process. Apart from that, PSFA has also shown to be effective for outlier handling [121] and process monitoring [76]. As an unsupervised method [75], PSFA can only extract latent variables from input observations without considering the quality variables and further, a regression model needs to be built to predict the quality variables using the extracted SFs.

Even though PSFA is an effective method to model dynamic systems, the modeling

paradigm does not account for any output information during the latent variables extraction. We intend to extract the latent features that vary slowly and carry the dominant and intrinsic varying trends for the processes, bearing close resemblance with physics of the process. In order to address this issue, we propose a semi-supervised approach for PSFA termed as Input-Output PSFA (IOPSFA). It is a probabilistic latent variable model, in which the whole observation dataset, rather than only input variables, are used to extract SFs. If the full set of output observations are available, IOPSFA treats combined inputs and outputs as augmented inputs. In a case where only partial observations of outputs can be obtained, the available observations are used to train the model. In both cases, models are trained by employing EM algorithm under maximum likelihood estimation (MLE) framework. In this work, the above mentioned two cases, both randomly missing outputs and multi-rate problems, are investigated for the proposed IOPSFA. Inherited from PSFA, SFs extracted from IOPSFA can separate slowly (intrinsic) and fast (noisy) varying latent variables and more importantly, SFs include output information, hence are expected to be more explanatory for the outputs. Further, predictive models are built based on the selected lower dimensional SFs and the effectiveness of the models is validated through one industrial application along with an experimental case study. In a related work , reference [122] have proposed Dynamic Probabilistic Latent Variable Model (DPLVM) by incorporating both inputs and outputs in a Linear Dynamical System (LDS) framework, which bears a similar formulation as that of PSFA. However, the above work neither extracts slowly varying features from the data nor considers missing data problem.

The rest of the chapter is organized as follows. In the next section, as preliminaries, SFA and PSFA are briefly revisited. In the following two sections, detailed formulation and parameter estimation steps of IOPSFA using EM algorithm are presented along with the procedure for inferential modeling. Following that, a case study on an oilsands process, namely, steam-assisted gravity drainage (SAGD) process along with a laboratory experiment on tanks system are provided. Finally, conclusions from the studies are drawn.

3.2 Preliminaries: SFA and Probabilistic SFA

3.2.1 SFA

Slow Feature Analysis was proposed by [43] and it is a novel method to learn the slowly varying latent variables or invariant properties from input signals. SFA is a dimensionality reduction method which extracts useful information from input signals by separating out fast components which are usually uninformative. SFA tries to find a set of non-linear mapping functions $\{g_j(\cdot), 1 \leq j \leq q\}$ from a m-dimensional input space $\mathcal{X}$ to a q-dimensional latent variable space $\mathcal{F}$. The SFs can be extracted by solving the following optimization problem [43]:

$$\min_{g_j(\cdot)} \Delta(\cdot) \triangleq \min_{g_j(\cdot)} \left\langle \dot{s}_j^2(t) \right\rangle_t \tag{3.1}$$

subject to:

$$\left\langle s_j(t) \right\rangle_t = 0, \quad \text{(zero mean)} \tag{3.2}$$

$$\left\langle s_j^2(t) \right\rangle_t = 1, \quad \text{(unit variance)} \tag{3.3}$$

$$\forall i \neq j, \langle s_i(t) s_j(t) \rangle = 0, \quad \text{(decorrelation)} \tag{3.4}$$

where, $s_j(t)$ represents the extracted j-th dimension latent variable. $\langle \cdot \rangle_t$ is the expectation operator over time and $\dot{s}_j(t)$ refers to the temporal difference: $\dot{s}_j(t) = s_j(t) - s_j(t-1)$, hence speed, of the j-th SF. $\Delta(\cdot)$ represents the slowness of the latent variable and the smaller $\Delta(\cdot)$, the slower variation of the latent variable. Equations (3.2) $\sim$ (3.4) impose three properties of the extracted slow features: each dimension of slow features has zero mean, unit variance and de-correlated from each other. When the mapping functions are linear, these functions simply reduce to a mapping matrix $W \in \mathbb{R}^{m \times q}$. The SFs are calculated as:

$$s(t) = W^T x(t) \tag{3.5}$$

where $x(t)$ is the observations. When we want to derive the same number of SFs as that of inputs, i.e. $q = m$, the optimization problem of Linear SFA reduces to a generalized eigenvalue problem:

$$\left\langle \dot{x}\dot{x}^T \right\rangle_t W = \left\langle x x^T \right\rangle_t W \Omega \tag{3.6}$$

where $\dot{x}(t) = x(t) - x(t-1)$ and Ω is a diagonal matrix composed of eigenvalues that represent the slowness of the extracted latent variables and the mapping matrix W is composed of corresponding eigenvectors.

3.2.2 Probabilistic SFA

Reference [47] extended SFA to a probabilistic generative model framework and provided a probabilistic interpretation of SFA. Reference [20] provided detailed methodology of estimating PSFA model parameters by MLE using the EM algorithm. The probabilistic model not only provides better insights to the model structures and parameters, it can also give the distributions of the latent variables and observations. The model of PSFA is in linear Gaussian state-space form and is given as below:

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), & e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ x(t) = Hs(t) + e_x(t), & e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \end{cases} \tag{3.7}$$

where Λ and Σ are the state and observation noise covariance matrices, respectively, and state transition matrix F is a diagonal matrix composed of $\lambda_{1:q} \triangleq \{\lambda_1, \cdots, \lambda_q\}$, where λ_j controls the strength of the correlation between the j-th dimension SF at different time points and, therefore, the slowness. Emission matrix $H \in \mathbb{R}^{m \times q}$ is a full matrix where m is the dimension of the input space. The SFs and observations are assumed to be corrupted by independent and identical Gaussian distributed noise, hence, the matrices Λ and Σ become diagonal such that:

$$\Lambda = \text{diag}\left\{1 - \lambda_1^2, \cdots, 1 - \lambda_q^2\right\}, \Sigma = \text{diag}\left\{\sigma_1^2, \cdots, \sigma_m^2\right\} \tag{3.8}$$

where, σ_j^2 is the variance of the j-th dimension SF. The decorrelation property (3.4) of SFs will still hold due to the independent assumption of state noise. It can also be easily verified that SFs derived from PSFA have zero mean and unit variance as well:

$$\mathbb{E}[s_j(t)] = 0, Var\{s_j(t)\} = 1, 1 \leq j \leq q \tag{3.9}$$

The slowness of each SF can be calculated as:

$$\Delta(s_j) = 2(1 - \lambda_j) \tag{3.10}$$

such that a larger λ_j implies a stronger correlation between $s_j(t)$ and $s_j(t-1)$, hence a slower variation of $s_j(t)$ [20].

The formulation of PSFA in (3.7) shows that the latent variable $s(t)$ is derived only from input variables $x(t)$. Hence, $s(t)$ is meant to capture the dynamics in $x(t)$ only. The output variables could also contain dynamics which are beneficial to the prediction of future outputs. If we can also conveniently extract the useful information that outputs carry while deriving latent variables, the SFs are expected to have better predictability than SFs derived from inputs alone, as in PSFA. In the next section, an approach to extract SFs from input with consideration of output is proposed.

3.3 Proposed Formulation of Input-Output PSFA (IOPSFA)

In order to fuse the output information in SFs and to provide better prediction to output variables, it is possible to extract SFs in a semi-supervised fashion. In our proposed formulation of IOPSFA, output measurements are additionally included into the formulation of PSFA in (3.7). Hence, the proposed formulation of IOPSFA is given as:

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ x(t) = Hs(t) + e_x(t), e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \\ y(t) = Us(t) + e_y(t), e_y(t) \sim \mathcal{N}(\mathbf{0}, \Gamma) \end{cases} \tag{3.11}$$

where, $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ are measurement noise covariance matrices for inputs and outputs, respectively. The dimensions of latent variable space $\mathcal{F}$, inputs space $\mathcal{X}$ and outputs space $\mathcal{Y}$ are q, m and l, respectively. Similar to PSFA, the diagonal components of F: $\{\lambda_j, 1 \leq j \leq q\}$ control the varying speed of each slow features. Augmenting the input and output equation in (3.11) yields:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} H \\ U \end{bmatrix} s(t) + \begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix} \tag{3.12}$$

where,

$$\begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Gamma \end{bmatrix}\right) \tag{3.13}$$

Hence by treating $\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ as augmented inputs with dimension $m + l$, we can apply PSFA to extract the q-dimensional SFs from the augmented inputs. The detailed

steps and derivations are provided in the next section. For simplicity in the following derivation, we define $\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \triangleq [x(t); y(t)]$ and $\begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Gamma \end{bmatrix} \triangleq D$. The unknown parameter set is $\Theta = \{\lambda_j, 1 \leq j \leq q, H, U, \Sigma, \Gamma\}$. Further, it can be easily verified that properties of independence, zero mean and unit variance for SFs will still hold for IOPSFA.

3.4 Parameter Estimation of IOPSFA Using the EM Algorithm

Given observations D, MLE can be used to estimate the parameters Θ of a statistical model by maximizing the likelihood function $L(D; \Theta) = p(\Theta|D)$. The EM algorithm is one of the most popular methods under the MLE framework that can be used if hidden/latent variables are present [14]. EM algorithm iterates between two steps: (1) Maximization step (M-Step): to calculate the model parameters by maximizing the expectation of log-likelihood function, so called Q-function, parameterized by the current value of parameters; (2) Expectation step (E-Step): to find the expected log-likelihood function in terms of latent variables that need to be calculated for estimating the unknown parameters. The iteration between two steps continues until convergence.

The parameter estimation and SFs extraction using the IOPSFA method can be solved under the MLE framework using the EM algorithm. Let us denote the complete data set $D = \{D_o, D_{hid}\}$, where the observation data set $D_o = \left\{ \left[x(t); y(t)\right] \right\} = \left\{ \left[x(1); y(1)\right], \cdots, \left[x(t); y(T)\right] \right\}$ and the hidden data set is composed of latent variables, which are the slow features: $D_{hid} = \{s(1), \cdots, s(t)\}$. We assume that there are missing values in the output data set. So, the whole time series can be divided into two parts: $[1 : T] = \{T_{obs}, T_{mis}\}$, in which, T_{obs} are the time stamps at which inputs are labelled and T_{mis} are the time stamps at which inputs are unlabelled, i.e, outputs are missing.

The missing outputs at time t, denoted as $y_{mis}(t)$, can have any possible realization value but unknown to us. EM algorithm helps to obtain approximate MLE estimates by maximizing the expected joint log-likelihood of all available observations and latent

variables. In order to proceed further, the joint log-likelihood can be derived as:

$$\log P\big(\,[x;y]\,,s|\Theta\big) = \log P(s|\Theta)\cdot P\big(\,[x;y]\,|s,\Theta\big) \tag{3.14}$$
$$= \sum_{t=1}^{T} \log P\big(\,[x(t);y(t)]\,|s(t),\Theta\big) + \log P\big(s(1)|\Theta\big) + \sum_{t=2}^{T} \log P\big(s(t)|s(t-1),\Theta\big)$$

Since the output values at T_{mis} are missing, the first term in (3.14) can be expanded, as they follow multivariate Gaussian distribution, as follows:

$$\sum_{t=1}^{T} \log P\big(\,[x(t);y(t)]\,|s(t),\Theta\big)$$
$$= -\frac{(m+l)T}{2}\log 2\pi - \frac{T}{2}\log|D| - \frac{1}{2}\sum_{t\in T_{mis}} \Big(x(t)-Hs(t)\Big)^T \Sigma^{-1}\Big(x(t)-Hs(t)\Big)$$
$$-\frac{1}{2}\sum_{t\in T_{obs}} \Big(\,[x(t);y(t)]-[H;U]\,s(t)\Big)^T D^{-1}\Big(\,[x(t);y(t)]-[H;U]\,s(t)\Big) \tag{3.15}$$

where, at instances in T_{mis}, we only use information from available inputs. Assuming that the initial state distribution is standard Gaussian distribution: $p(s(1)) = \mathcal{N}(\mathbf{0},\mathbf{I}_q)$, the second term in (3.14) is derived as:

$$\log P\big(s(1)|\Theta\big) = -\frac{q}{2}\log 2\pi - \frac{1}{2}s(1)^T s(1) \tag{3.16}$$

Further, state propagation term is determined as:

$$\sum_{t=2}^{T} \log P\big(s(t)|s(t-1),\Theta\big) \tag{3.17}$$
$$= -\frac{q(T-1)}{2}\log 2\pi - \frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t)-\lambda_j s_j(t-1)\Big)^2$$

After taking summation of (3.15) to (3.17) and performing some straightforward mathematical manipulations, the complete data log-likelihood (3.14) becomes:

$$\begin{aligned}\log P\Big(\,[x;y]\,,s|\Theta\Big) = &-\frac{(m+l+q)T}{2}\log 2\pi - \frac{T}{2}\log|D| - \frac{1}{2}s(1)^T s(1)\\ &-\frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t)-\lambda_j s_j(t-1)\Big)^2\\ &-\frac{1}{2}\sum_{t\in T_{obs}}\Big(\,[x(t);y(t)]-[H;U]\,s(t)\Big)^T D^{-1}\Big(\,[x(t);y(t)]-[H;U]\,s(t)\Big)\\ &-\frac{1}{2}\sum_{t\in T_{mis}}\Big(x(t)-Hs(t)\Big)^T\Sigma^{-1}\Big(x(t)-Hs(t)\Big)\end{aligned} \tag{3.18}$$

After obtaining the joint log-likelihood of complete data set, the Q-function is formulated for application of the EM algorithm:

$$Q(\Theta|\Theta^{(n)}) = \mathbb{E}_{s|X,Y,\Theta^{(n)}}\Big\{ \log P\Big([x;y], s|\Theta \Big) \Big\} \tag{3.19}$$

where, $\Theta^{(n)}$ represents the unknown parameters estimated in the last (n-th) iteration in the EM algorithm. For simplicity, we use $\mathbb{E}$ instead of $\mathbb{E}_{s|x,y,\Theta^{(n)}}$ in the following derivation. Substituting (3.18) into (3.19), the Q-function can be decomposed into three parts:

$$Q(\Theta|\Theta^{(n)}) \triangleq Q_{Const} + Q_1(\lambda_j) + Q_2(U, H, \Sigma, \Gamma) \tag{3.20}$$

where each part is presented as below:

$$Q_{Const} = -\frac{(m+l+q)T}{2}\log 2\pi - \frac{1}{2}\mathbb{E}\Big\{ s(1)^T s(1) \Big\} \tag{3.21}$$

$$Q_1(\lambda_j) = \mathbb{E}\Big\{ -\frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2) - \frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\Big(s_j(t) - \lambda_j s_j(t-1) \Big)^2 \Big\} \tag{3.22}$$

$$Q_2(U, H, \Sigma, \Gamma) = \mathbb{E}\Big\{ -\frac{T}{2}\log|D| - \frac{1}{2}\sum_{t=1}^{T}\Big[\big(x(t) - Hs(t)\big)^T \Sigma^{-1} \big(x(t) - Hs(t)\big) \Big] - \frac{1}{2}\sum_{t\in T_{obs}}\Big[\big(y(t) - Us(t)\big)^T \Gamma^{-1} \big(y(t) - Us(t)\big) \Big] \Big\} \tag{3.23}$$

Now that we have formulated the Q-function, we can take derivative of it with respect to each unknown parameter to obtain iterative parameter update expressions (M-step). Further, this also involves the calculation of expectation terms in various equations (E-step). These steps are presented in detail in the upcoming subsections.

3.4.1 M-Step

In M-Step, we can derive the update equation for unknown parameters $\Theta = \{\lambda_j, 1 \leq j \leq q, H, U, \Sigma, \Gamma\}$ by computing the derivatives and letting them equal to zero.

Updating λ_j: Parameter λ_j is updated by taking derivative of $Q_1(\lambda_j)$ with respect to λ_j and equating it to zero,

$$\frac{\partial Q_1(\lambda_j)}{\partial \lambda_j} \Rightarrow (T-1)\lambda_j^3 - \sum_{t=2}^{T} \mathbb{E}\big[s_j(t)s_j(t-1)\big]\lambda_j^2 + \Big(\sum_{t=2}^{T} \mathbb{E}\big[s_j^2(t)\big] + \sum_{t=2}^{T} \mathbb{E}\big[s_j^2(t-1)\big] - T + 1\Big)\lambda_j - \sum_{t=2}^{T} \mathbb{E}\big[s_j(t)s_j(t-1)\big] = 0 \tag{3.24}$$

The updated λ_j can be calculated by solving the nonlinear equation (3.24) numerically while constraining the roots in the range [0,1).

Updating H,U,Σ and Γ: Parameter H, U, Σ and Γ are updated by taking derivative of $Q_2(U, H, \Sigma, \Gamma)$ with respect to each parameter and equating them to zero, leading to,

$$H^{(n+1)} = \Big(\sum_{t=1}^{T} \mathbb{E}\big[x(t)s^T(t)\big]\Big) \cdot \Big(\sum_{t=1}^{T} \mathbb{E}\big[s(t)s^T(t)\big]\Big)^{-1} \tag{3.25}$$

$$U^{(n+1)} = \Big(\sum_{t\in T_{obs}} \mathbb{E}\big[y(t)s^T(t)\big]\Big) \cdot \Big(\sum_{t\in T_{obs}} \mathbb{E}\big[s(t)s^T(t)\big]\Big)^{-1} \tag{3.26}$$

$$\begin{aligned}\Sigma^{(n+1)} =& \frac{1}{T}\sum_{t=1}^{T}\Big(x(t)x^T(t) - x(t)\mathbb{E}\big[s^T(t)\big](H^T)^{(n+1)} \\ &- H^{(n+1)}\mathbb{E}\big[s(t)\big]x^T(t) + H^{(n+1)}\mathbb{E}\big[s(t)s^T(t)\big](H^T)^{(n+1)}\Big)\end{aligned} \tag{3.27}$$

$$\begin{aligned}\Gamma^{(n+1)} =& \frac{1}{T}\sum_{t\in T_{obs}}\Big[y(t)y^T(t) - y(t)\mathbb{E}\big[s^T(t)\big](U^T)^{(n+1)} \\ &- U^{(n+1)}\mathbb{E}\big[s(t)\big]y^T(t) + U^{(n+1)}\mathbb{E}\big[s(t)s^T(t)\big](U^T)^{(n+1)}\Big]\end{aligned} \tag{3.28}$$

As the equations in (3.11) form a state-space model with Gaussian noises, the expectation terms in (3.24) $\sim$ (3.28) can be computed using Kalman Filtering and Kalman smoothing techniques and are presented in a later subsection.

3.4.2 E-Step

In M-Step, parameter update equations (3.24) $\sim$ (3.28) require computation of expectation terms: $\mathbb{E}\big[s(t)\big], \mathbb{E}\big[s(t)s^T(t)\big]$, and $\mathbb{E}\big[s(t)s^T(t-1)\big]$ with respect to the posterior distribution $P\big(s|x, y, \Theta^{(n)}\big)$. Since IOPSFA pertains to the class of LDS, these

terms are calculated in E-Step by adopting Kalman filtering (forward recursions) and smoothing techniques (backward recursions) [73]. The posterior distribution of SFs follows a normal distribution:

$$P\Big(s(t)|\left[x(1);y(1)\right],\cdots,\left[x(t);y(t)\right],\Theta^{(n)}\Big)\sim\mathcal{N}(\mu(t),V(t)) \tag{3.29}$$

where $\mu(t)$ and $V(t)$ are the mean and covariance matrix of SFs respectively and $s(t)$ is calculated as $\mu(t)$. The predicted state also follows Gaussian distribution:

$$P\Big(s(t)|\left[x(1);y(1)\right],\cdots,\left[x(t-1);y(t-1)\right],\Theta^{(n)}\Big)\sim\mathcal{N}(F\mu(t-1),P(t-1)) \tag{3.30}$$

$P(t-1)$ is the covariance matrix of one-step prediction given observations of $s(t)$ up to time $t-1$. With full information of outputs, the forward recursions of Kalman filtering are carried out in (3.31) in a sequential manner, as below:

$$\begin{aligned}
P(t-1) &= FV(t-1)F^T+\Lambda \\
K(t) &= P(t-1)\left[H;U\right]^T\Big(\left[H;U\right]P(t-1)\left[H;U\right]^T+D\Big)^{-1} \\
\mu(t) &= F\mu(t-1)+K(t)\Big(\left[x(t);y(t)\right]-\left[H;U\right]F\mu(t-1)\Big) \\
V(t) &= \Big(\mathbf{I}-K(t)\left[H;U\right]\Big)P(t-1)
\end{aligned} \tag{3.31}$$

The calculated $\mu(t)$ and $V(t)$ will be used in the Kalman smoothing process later. The initialization conditions for the forward recursions are given as:

$$\begin{aligned}
K(1) &= \left[H;U\right]^T\Big(\left[H;U\right]\left[H;U\right]^T+D\Big)^{-1} \\
\mu(1) &= K(1)\left[x(1);y(1)\right] \\
V(1) &= \mathbf{I}-K(1)\left[H;U\right]
\end{aligned} \tag{3.32}$$

If outputs are missing at time t, only inputs information can be used to carry out the forward recursions, then the Kalman gain, mean and covariance matrix are calculated as:

$$\begin{aligned}
K_{mis}(t) &= P(t-1)H^T\big[HP(t-1)H^T+\Sigma\big]^{-1} \\
\mu_{\mathbf{mis}}(t) &= F\mu_{\mathbf{mis}}(t-1)+K_{\mathbf{mis}}(t)\big[x(t)-HF\mu_{\mathbf{mis}}(t-1)\big] \\
V_{mis}(t) &= \big[\mathbf{I}-K(t)H\big]P(t-1)
\end{aligned} \tag{3.33}$$

and the initialization conditions become:

$$\begin{aligned} K_{mis}(1) &= H^T\big[HH^T + \Sigma\big]^{-1} \\ \mu_{mis}(1) &= K_{mis}(1)x(1) \\ V_{mis}(1) &= \mathbf{I} - K_{mis}(1)H \end{aligned} \tag{3.34}$$

where the subscript 'mis' refers to the missing data case. In both cases, mean $\mu(t)$ and covariance matrix $V(t)$ have the dimensions $q \times 1$ and $q \times q$, respectively. In full data case, the Kalman gain $K(t)$ has the dimension of $q \times (m+l)$ and in the missing data case, its dimension become $q \times m$. However, the dimension change of Kalman gain does not impact the calculation of SFs. After forward recursions, backward recursions can be carried out sequentially, as given below:

$$\begin{aligned} \hat{\mu}(t) &= \mu(t) + J(t)\big[\hat{\mu}(t+1) - F\mu(t)\big] \\ \hat{V}(t) &= V(t) + J(t)\big[\hat{V}(t+1) - P(t)\big]J^T(t) \\ J(t) &= V(t)F^T P(t)^{-1} \end{aligned} \tag{3.35}$$

where $\hat{\mu}(t)$ and $\hat{V}(t)$ are the mean and covariance matrix of posterior distribution $P\big(s|x, y, \Theta^{(n)}\big)$, respectively. The backward recursions are initialized by the estimations in the last step of forward recursion, that is, $\hat{\mu}(T) = \mu(T)$ and $\hat{V}(T) = V(T)$. After performing the forward and backward recursions, the expectation terms in (3.24) $\sim$ (3.28) can be calculated as follows [73]:

$$\mathbb{E}\big[s(t)\big] = \hat{\mu}(t) \tag{3.36}$$

$$\mathbb{E}\big[s(t)s^T(t)\big] = J(t-1)\hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t-1) \tag{3.37}$$

$$\mathbb{E}\big[s(t)s^T(t-1)\big] = \hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t) \tag{3.38}$$

The expectation values calculated in (3.36) $\sim$ (3.38) complete the Q-function computation. Now, it can iteratively be maximized to obtain the updated estimation of parameters in the M-Step in the next iteration.

It is worth noticing that, in this study, we do not explicitly distinguish the randomly missing case and multi-rate case. Due to the flexibility in Kalman filtering step, IOPSFA does not have any requirements on the missing pattern of the data as it directly aims to maximize the expected joint log-likelihood of the latent variables

and all available observations. This feature makes IOPSFA more adaptable to various missing data cases.

The selection of the initial value of each parameter and latent variable is critical to the EM algorithm's performance. In this work, we adopt an improved initialization strategy that utilizes the result of Linear SFA [20]. The initial slowness value of each SF is calculated as the diagonal elements of Ω in (3.6) and the initial value of λ_j are obtained using (3.10). When $q < m+l$, W^{-T} can be divided into $W^{-T} = \begin{bmatrix} W_1^T & W_2^T \end{bmatrix}^T$, where W_1 contains the first q rows of W^{-1} and W_2 contains the last $m+l-q$ rows of W^{-1} and they can be used to recover the observations from SFs $s(t)$ in LSFA in (3.39).

$$[x(t); y(t)] = W_1^{-T} s_{1:q}(t) + W_2^{-T} s_{q+1:m+l}(t) \tag{3.39}$$

The first term is calculated by the first q slowest features and it describes the dominant trends of the observations. The second term is calculated by the last $m+l-q$ slowest features which mainly contain the noise. The initial guess of $[H; U]$ would be W_1^T and the initial guess of the covariance matrix Σ and Γ are calculated as the variance of each elements in the second term.

3.4.3 Prediction Using the Model

For soft sensor application, only the past output measurements are available to us for the prediction of future outputs. At the time t, we can obtain SFs: $\{s(1), \cdots, s(t-1)\}$ using (3.29) since observations $\{x(1), \cdots, x(t), y(1), \cdots, y(t-1)\}$ are now available. To predict $y(t)$, one step ahead prediction of $s(t)$ needs to be performed according to the Kalman filter recursions. Since $y(t)$ is not available, we can treat it as the missing $y(t)$ case and predict $s(t)$ using Kalman filter equation (3.33):

$$\hat{s}(t) = Fs(t-1) + K(t)\big[x(t) - HFs(t-1)\big] \tag{3.40}$$

where $\hat{s}(t)$ is the one step ahead prediction of SFs. Then, the prediction of quality variables $\hat{y}(t)$ can be estimated by substituting (3.40) into the output measurements equation in (3.11):

$$\hat{y}(t) = U\hat{s}(t) + e_y(t) = UFs(t-1) + UK(t)\big[x(t) - HFs(t-1)\big] + e_y(t) \tag{3.41}$$

Since $e_y(t)$ is unknown, $y(t)$ can be estimated as the mean value of $\hat{y}(t)$. Given that $e(t)$ has zero mean, $y(t)$ can be evaluated as follows:

$$y(t) = \text{mean}\{\hat{y}(t)\} = UFs(t-1) + UK(t)\big[x(t) - HFs(t-1)\big] \tag{3.42}$$

Only current $x(t)$ and past slow feature $s(t-1)$ are used in the prediction of $\hat{s}(t)$ in (3.40) and $\hat{s}(t)$ does not include any information from $y(t)$ yet. As the latent variable derived from IOPSFA should include information from both x and y, the predicted latent variables in (3.40) can not be considered as complete. Once $y(t)$ is obtained, the complete set of latent variables $s(t)$ can be calculated using Kalman filter equation (3.31):

$$s(t) = Fs(t-1) + K(t)\bigg([x(t); y(t)] - [H; U]\, Fs(t-1) \bigg) \tag{3.43}$$

Then the complete set $s(t)$ can be used in (3.40) $\sim$ (3.42) to predict $y(t+1)$.

In the next section, an industrial application and an experimental study are employed to showcase the efficacy of the proposed approach.

3.5 Industrial Case Study: SAGD Process Well Pair Water Content Soft Sensor Design

In this section, we employ industrial data from a SAGD process to illustrate the validity of the proposed algorithm. SAGD process is an innovative in-situ oil recovery technology to extract heavy oil or bitumen from oil sands that are buried deep in underground [123, 124].

Figure 3.1 shows one typical well pair for oil extraction section of SAGD process and illustrates how emulsion, mixture of oil, water and gas, are extracted from underground. For each well pair, two horizontal wells are drilled into the underground. The upper well, i.e. injection well, is used to inject high temperature and pressure steam to soften the oil sands. This results in the formulation of a oil-water emulsion which is flowable and transmissible. The lower well, i.e. production well, is used to pump out the heated emulsion from the underground chamber. The outlet emulsion contains a few gas components and a lot of water due to the condensation of the injected steam. The composition of the emulsion, especially the water content, is an

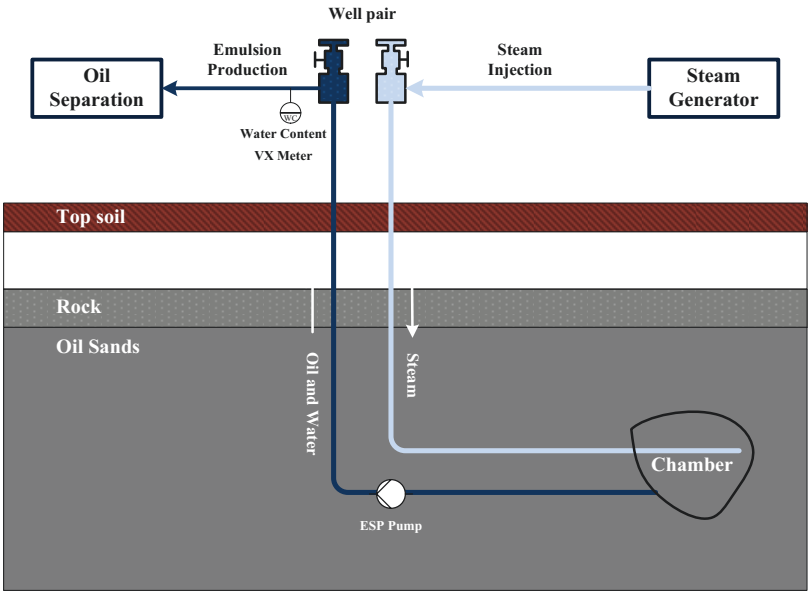

Figure 3.1: SAGD process well pair diagram

important variable that determines the amount of chemicals the needs to be injected in the downstream process in order to produce oil that meets the specifications. On-line measurement of water content is possible by using an expensive instrument called VX meter and hence cannot be installed for all well pairs due to economic considerations. This calls for the development of a soft sensor for estimating estimate water content in real time.

The first sub-figure in Figure 3.2 shows the profile of the quality variable, which is water content and the rest are seven selected influential input variables by performing correlation analysis. The raw data of water content and selected influential variables are sampled every 10 min. After data pre-processing they are represented as 2-hour averaged values and used for further analysis. The data includes 2642 samples in total wherein the first 1422 samples are used for training and the last 1220 samples for testing. For proprietary reasons, the attributes of input variables are not disclosed and all data have been normalized. In the following sub-section, we provide the details of predictive model development by using SFs extracted from all inputs and output considering the following three cases: 1) no missing output; 2) output with randomly missing values; 3) multi-rate case, which is a special case of 2).

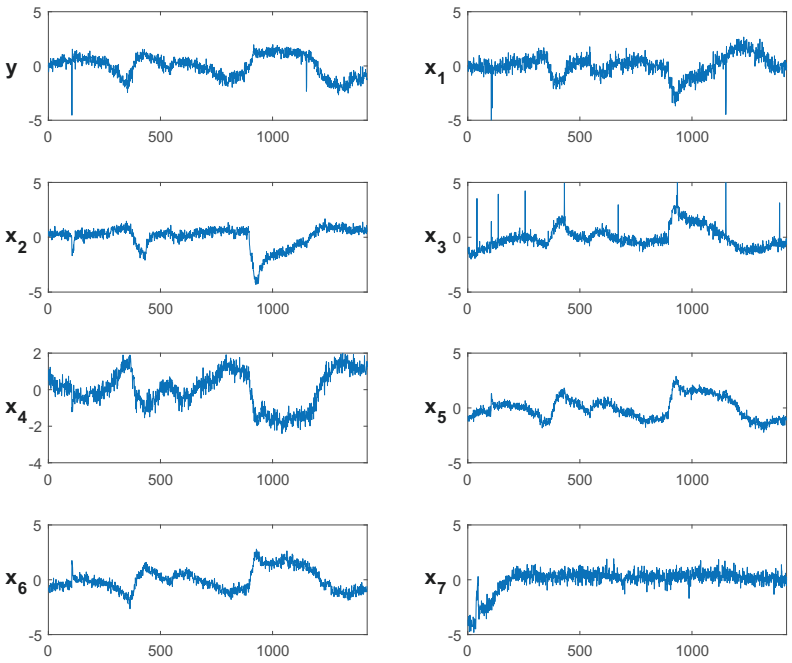

Figure 3.2: Process measurements (normalized)

3.5.1 Case 1: No Missing Output

In this case, we do not consider any missing data in outputs and all measurement values of inputs and outputs are available for model building. Since the plant has one VX Meter installed for the well pair, we can obtain accurate and real-time measurements for the water content. So water content has the same sampling rate as other influential variables. By combining all inputs and output variables as augmented inputs and then implementing IOPSFA algorithm, we can extract SFs and the resulting seven SFs are shown in the left column of Figure 3.3. For comparison purpose, the seven SFs extracted only from inputs by PSFA are also shown in the right column of Figure 3.3. SFs used for building predictive model are chosen according to their slowness values and trends. It can be seen that there is a sudden decrease from λ_4 to λ_5 in both methods and the resulting SFs $s_{6:7}(t)$ only contain noise and show no obvious trends. This means $s_{6:7}(t)$ contain very little system dynamic information and may not be useful in building predictive models. Although $s_5(t)$ extracted from both methods are very noisy, they nevertheless show certain trend which may be useful in building the model. So in this case study, we choose $q = 5$. Then, predictive models are built based on IOPSFA and PSFA algorithms, respectively.

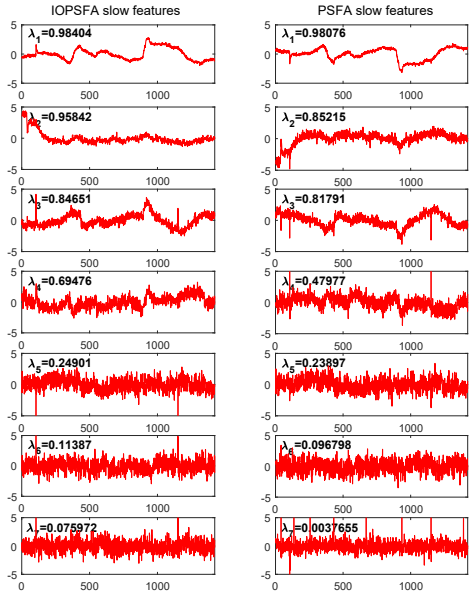

Figure 3.3: SFs extracted by IOPSFA and PSFA for water content soft sensor

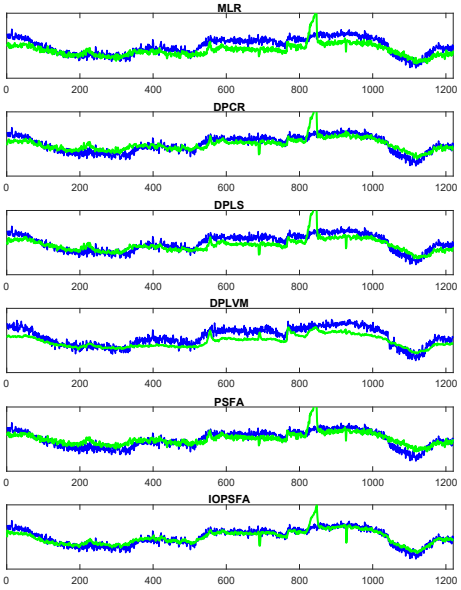

Figure 3.4: Prediction performance without missing outputs

Table 3.1: Prediction results without missing outputs in SAGD case

	MLR	DPCR	DPLS	DPLVM	PSFA	IOPSFA
MSE	22.3593	13.3051	18.2976	16.7858	14.3875	6.701
CORR	0.74765	0.76958	0.72223	0.8995	0.75201	0.89114

We compare the prediction results of these models with other popular regression methods, i.e. Multiple Linear Regression (MLR), Dynamic Principal Component Regression (DPCR), Dynamic Partial Least Square (DPLS), PSFA and DPLVM [122] on the testing data set. The time delay for the DPCR and DPLS is chosen using the method presented in [125] with $d_{max} = 2$. This represents about maximum 4 hours delay, which makes sense in real SAGD application since we use 2-hour time average data. It is easy to enumerate all the possible time shifts for all seven process variables. The optimal combination of time delay is d=[0,1,0,0,2,2,1], in which each element represents the corresponding variable's time delay. The reduction order of DPLVM is considered to be 3 so that the best performance on testing data is obtained. The prediction trends are shown in Figure 3.4 and corresponding mean squared error (MSE) and Pearson correlation coefficient (CORR) are tabulated in Table 3.1.

We can see from Table 3.1 that, IOPSFA outperforms other methods in predicting the quality variable in the sense of MSE. DPLVM has the best performance in the sense of CORR, while IOPSFA results in CORR which is very close to DPLVM and the improvement is only marginal. Next, we consider the case where outputs contain randomly missing values, as they impose challenges in building the predictive models.

3.5.2 Case 2: Randomly Missing Output

In this subsection, we test the IOPSFA algorithm when water content has random missing values. We randomly generate certain percentage of time stamps and treat the output values at these time stamps as missing values. All the previously methods except for DPLVM were compared using the 2-hour averaged data, as DPLVM has not been developed for missing data scenarios. In IOPSFA, SFs are extracted using the available water content measurements along with other seven process variables. In PSFA, SFs are extracted only from seven process variables, then the predictive model is built based on the extracted SFs. Under each missing scenario, 20 Monte Carlo simulations have been conducted. In each Monte Carlo simulation, the missing time stamps are randomly generated and they are different from one simulation run to the other. The mean values of prediction MSE and CORR are calculated and shown in Figure 3.5 for comparison.

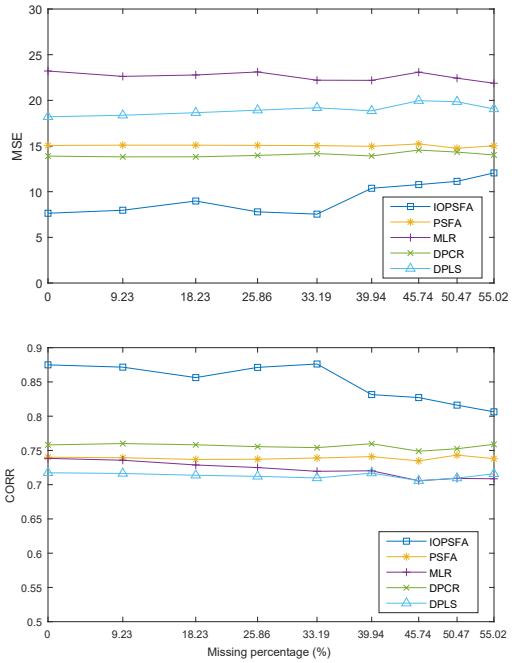

Figure 3.5: Mean value of the MSE and CORR for different missing percentages

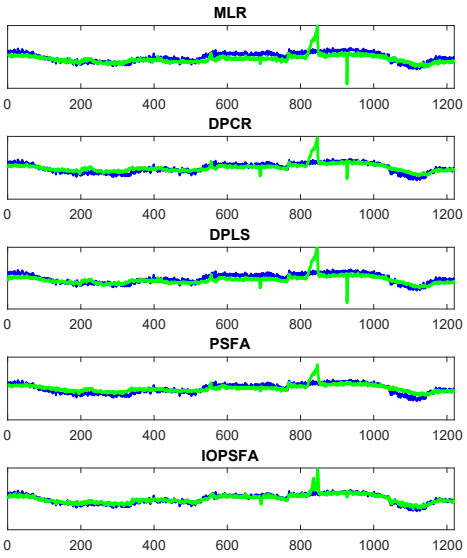

Figure 3.6: Prediction trends with 55.02% missing output samples

Table 3.2: Prediction results with 55.02% missing outputs and standard deviation of MSE and CORR in 20 Monte Carlo simulations in SAGD case

		MLR	DPCR	DPLS	PSFA	IOPSFA
Simulation instance performance (55.02% missing output)	MSE	24.7077	16.1349	23.3119	14.3075	8.9147
	CORR	0.67091	0.7274	0.67881	0.75579	0.85602
20 Monte Carlo simulations Std of performance indicators	MSE	2.8856	1.6312	2.4049	0.7650	1.5789
	CORR	0.0326	0.0277	0.0270	0.0140	0.0315

As illustrated in Figure 3.5, IOPSFA achieves the higher CORR and lower MSE compared to other methods. To intuitively compare the prediction performance among all methods, the predictions of water content for the Monte Carlo simulations with 55.02% missing data are shown in Figure 3.6, where blue line represents the real measurements without missing data and green line represents the predicted water content. The predicted values and full measurements of output are used when calculating CORR and MSE. Table 3.2 summarizes the simulation results with 55.02% missing output samples along with the standard deviation (Std) of performance indicators (MSE and CORR) to assess the uncertainty/variability of the Monte Carlo simulations. Figure 3.6 indicates that prediction from IOPSFA can track the trend of real measurements of water content better than other methods.

3.5.3 Case 3: Multi-rate Problem

In industrial settings, multi-rate measurements, a special case of missing data problem, are common. Normally, in multi-rate sampling, the quality variable has larger sampling interval than other process variables. The difference is that it has periodic missing pattern in contrast to random missing. When the sampling interval of quality variable is significantly larger than that of process variables, it imposes more challenges in building the predictive models in general. The proposed IOPSFA framework can however handle the multi-rate problem in the similar way as randomly missing output problem. In this example, we have considered water content measurements at 13 different re-sampling coefficients (in the range of [1,40]). For example, if the re-sampling coefficient is β ($1 \leq \beta \leq 40$), it means the sampling interval of quality variables is β times of the sampling interval of process variables and there are β ways to re-sample the water content measurements with a possible starting time $i \in [1, \beta]$. Suppose the starting point is $i = 3$ and $\beta = 10$, then the re-sampled time stamp series is $\{3, 3+\beta, 3+2\beta, \cdots\} = \{3, 13, 23, \cdots\}$. Similarly, if the starting point is $i = 4$, then the re-sampled time stamp series is $\{4, 14, 24, \cdots\}$. We have enumerate all possible i for each re-sampling coefficient and calculated the average value of MSE and CORR as comparison indices, which are shown in Figure 3.7.

Table 3.3: Prediction results when re-sampling coefficient=10 and standard deviation of MSE and CORR in 20 Monte Carlo simulations in SAGD case

		MLR	DPCR	DPLS	PSFA	IOPSFA
Simulation instance (re-sampling coefficient=10)	MSE	39.7082	16.9902	22.8753	14.0789	11.1184
	CORR	0.72708	0.70718	0.68188	0.75173	0.8133
20 Monte Carlo simulations performance Std of performance indicators	MSE	19.4432	7.9788	14.6634	10.7380	3.9522
	CORR	0.1095	0.0865	0.1384	0.0981	0.0970

Figure 3.7 shows that the performance of all methods decreases (MSE increases and CORR decreases) as the re-sampling coefficient becomes larger, and among those considered methods, IOPSFA has the smallest MSE and largest CORR. DPCR and DPLS generate latent variables to maximize the variance and therefore unable to model the slowly varying water content profile. When the re-sampling coefficient becomes very large, i.e. 35 and 40, performance of DPLS decreases very fast since there are only very few output samples that can be used in training. Performance of IOPSFA does not decrease significantly because the SFs can still be extracted from

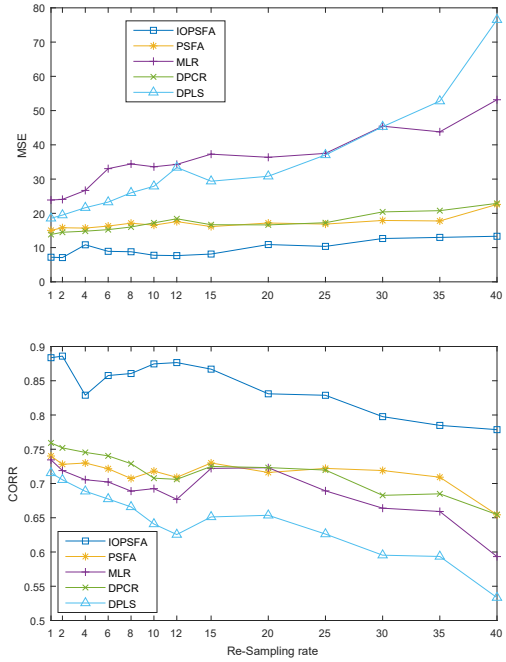

Figure 3.7: Mean value of the MSE and CORR for different missing percentages

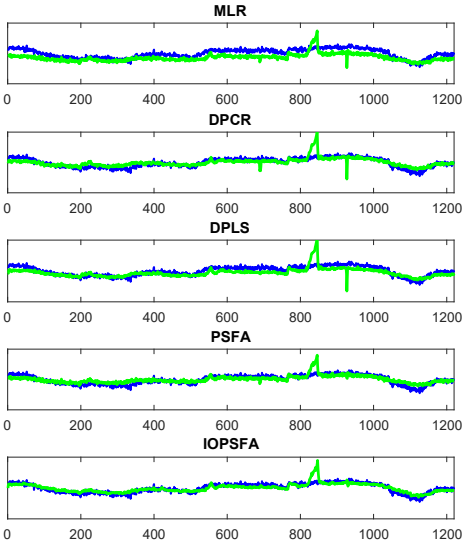

Figure 3.8: Mean value of the prediction trends when re-sampling coefficient = 10

the input data even though there are very few output samples. Figure 3.8 shows the prediction trends when re-sampling coefficient is equal to 10, where blue lines represent the true value of water content measurements and the green lines represent the predicted value. Table 3.3 summarizes the prediction performance of all methods when re-sampling coefficient is 10 along with the standard deviations of the Monte Carlo simulations, respectively.

3.6 Experimental Study: Tanks System

To further validate the efficacy of the proposed IOPSFA algorithm, an experiment is conducted on a tanks system located in the Process Control Laboratory of University of Alberta. The tanks system is composed of three cylindrical tanks, one storage tank, two pumps, three level sensors, two flow sensors and nine block valves. The schematic of apparatus is shown in Figure 3.9. Three cylindrical tanks are connected by six interconnection values: $V_1 \sim V_4$, V_6 and V_8 at the top, middle and bottom levels, respectively, V_5, V_7 and V_9 are drainage valves for cylindrical tanks. Two side pumps are used to pump the water from storage tank to the left and right cylindrical tanks, respectively. The flow rates from two pumps and the levels of three tanks are measured by flow and level sensors, respectively.

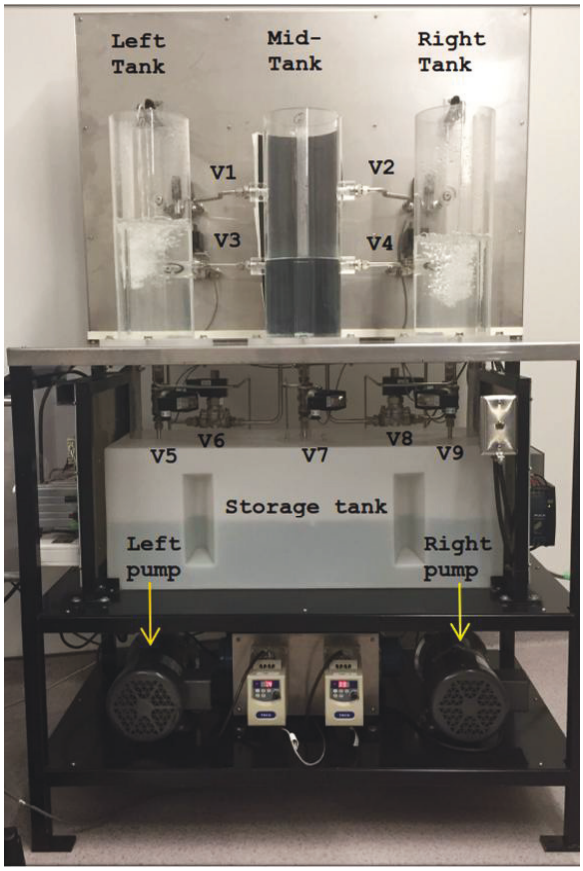

Figure 3.9: Tanks system

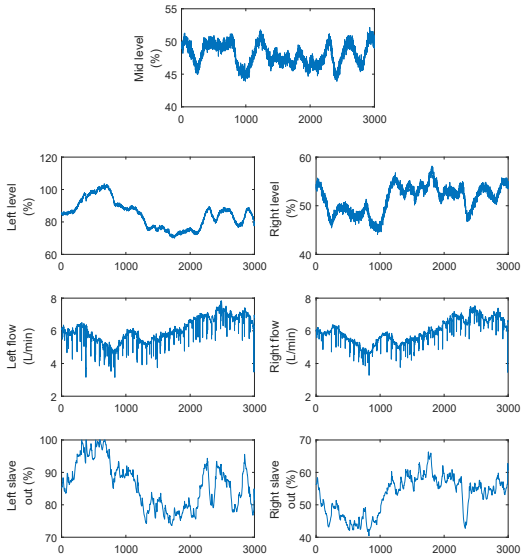

Figure 3.10: Inputs and outputs for tanks system experiment

In the tanks system, different combinations of open/closed status of the block valves can create different system dynamics. In this experiment, we kept V_1 and V_2 closed and rest of the valves are kept open. The tank levels are controlled to obtain a unique working mode. The quality variable to be predicted is the mid-tank level and six variables are chosen as inputs: two side tank levels, two pump outlet flow rates and two slave controller outputs that set the pump speeds. The quality variable and all input variables are shown in Figure 3.10. The data includes 3000 samples in total wherein the first 1500 samples are used for training and the last 1500 samples for testing.

3.6.1 Case 1: No Missing Values in Outputs

In this case, IOPSFA uses full information of quality variable to extract SFs. Six SFs extracted by PSFA and IOPSFA are displayed in Figure 3.11. When selecting the SFs for building the predictive model, considering the slowness values and dynamic trends, the following three scenarios are tested: 1) Choose $s_{1:3}(t)$ - as they are slower and smoother compared to other SFs and do not contain obvious outliers. 2) Choose $s_{1:4}(t)$ - as $s_4(t)$ has clear trend and does not contain obvious outliers. 3) Choose $s_{1:5}(t)$

- as $s_5(t)$ has clear trend although it is noisy. After trying above three scenarios, we find that choosing $s_{1:5}(t)$ yields the best performance. So, we fix $q = 5$ in this case study.

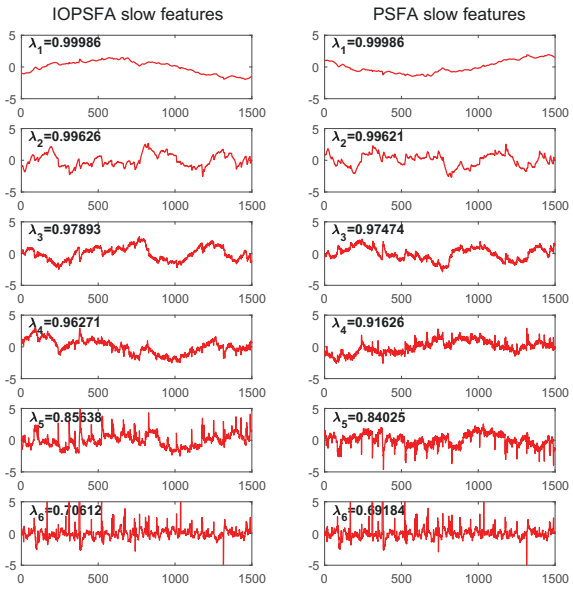

Figure 3.11: SFs extracted by IOPSFA and PSFA for tanks system

Table 3.4: Prediction results without missing outputs in Tanks system case

	MLR	DPCR	DPLS	DPLVM	PSFA	IOPSFA
MSE	0.56997	0.55148	0.56541	0.5388	1.0329	0.38943
CORR	0.87732	0.87789	0.87782	0.9055	0.87695	0.91518

The prediction results are shown in Figure 3.12 and Table 3.4 where MLR, DPCR, DPLS, DPLVM, PSFA and IOPSFA are compared. We have also determined the maximum time shift for DPCA and DPLS [125] and the optimal value is obtained as $d_{max} = 0$, which means DPLS and DPCR achieve the best performance when there exists no delay. So the delay vector $d = [0, 0, 0, 0, 0, 0, 0]$. The reduction order of DPLVM is 2 in this case. We can clearly observe that performance of IOPSFA is superior to other approaches considered.

3.6.2 Case 2: Randomly Missing Output

In this subsection, we manipulate the missing percentage of quality variables by randomly generating certain percentage of time stamps and removing the quality

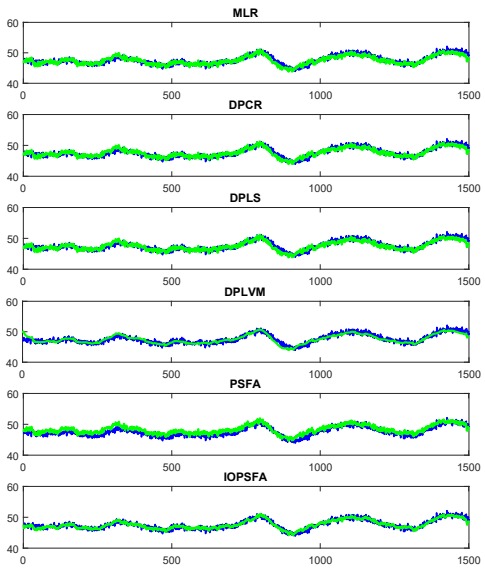

Figure 3.12: Prediction trends without missing outputs

variables' measurements corresponding to the time stamps. Under each missing percentage of outputs, we conduct 20 Monte Carlo simulations and the mean values of prediction MSE and CORR are calculated. DPCR and DPLS are also applied and we observe that they achieved the best performance when $d = [0, 0, 0, 0, 0, 0]$. The overall performance is shown in Figure 3.13. The performance of IOPSFA reduces when missing percentage exceeds 40% although it is far superior to other methods in terms of MSE and CORR. To intuitively demonstrate the prediction performances, predicted trends for 55.25% missing samples are presented in Figure 3.14 and Table 3.5 summarizes the numerical results.

Table 3.5: Prediction results with 55.25% missing outputs and standard deviation of MSE and CORR in 20 Monte Carlo simulations in Tanks system case

		MLR	DPCR	DPLS	PSFA	IOPSFA
Simulation instance performance (55.25% missing output)	MSE	0.61631	0.58439	0.6131	1.0601	0.47042
	CORR	0.87245	0.87279	0.87272	0.87133	0.89833
20 Monte Carlo simulations Std of performance indicators	MSE	0.0178	0.0115	0.0142	0.0378	0.0074
	CORR	0.0025	0.0021	0.0022	0.0032	0.0017

3.6.3 Case 3: Multi-rate Problem

For generating the multi-rate case, we manually re-sample the quality variables by β times, $1 \leq \beta \leq 40$. We also enumerate all re-sampling possibilities and calculate the average MSE and CORR for comparison. The comparison of MSE and CORR under

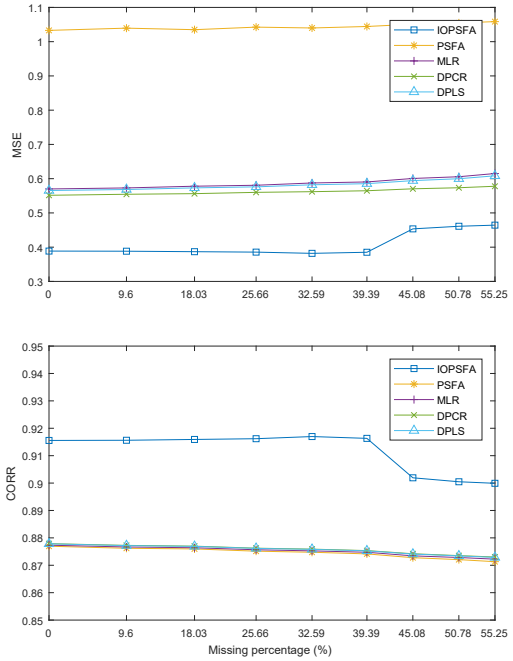

Figure 3.13: Mean value of the MSE and CORR for different missing percentages

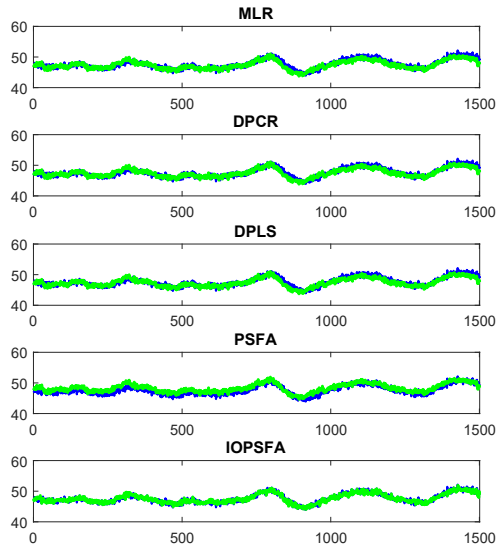

Figure 3.14: Prediction trends with 55.25% missing output samples

different re-sampling coefficients are shown in Figure 3.15.

As the re-sampling coefficient increases, the performances of all methods tend to deteriorate. However, IOPSFA still shows the best performance with the smallest MSE and the highest CORR, among the other methods. Figure 3.16 and Table 3.6 show the results of the simulation instance when re-sampling coefficient is 10. Blue lines represent the true value of middle tank level measurements and the green lines represent the predicted level measurements.

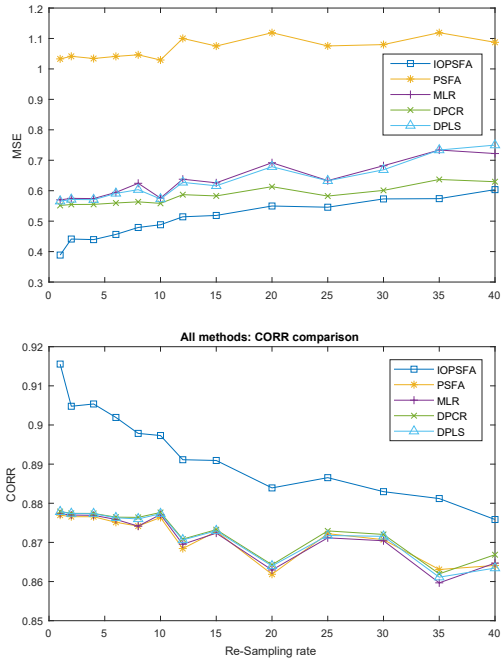

Figure 3.15: Mean value of the MSE and CORR under different re-sampling coefficients

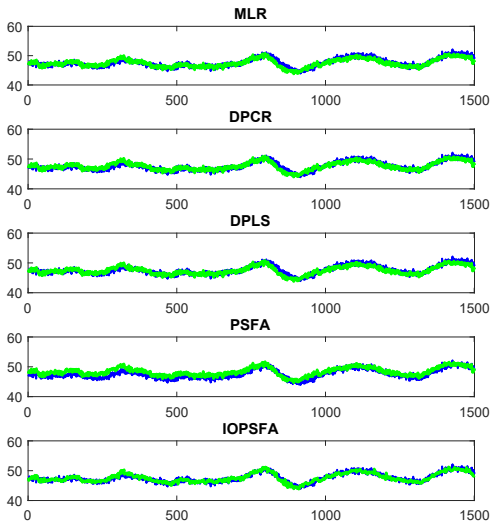

Figure 3.16: Mean value of the prediction trends when re-sampling coefficient = 10

Table 3.6: Prediction results when re-sampling coefficient=10 and standard deviation of MSE and CORR in 20 Monte Carlo simulations in Tanks system case

		MLR	DPCR	DPLS	PSFA	IOPSFA
Simulation instance performance (Re-sampling coefficient=10)	MSE	0.63615	0.61678	0.636	1.1665	0.47497
	CORR	0.86273	0.86265	0.86267	0.86127	0.8993
20 Monte Carlo simulations Std of performance indicators	MSE	0.1539	0.0680	0.1724	0.1997	0.0848
	CORR	0.0164	0.0159	0.0237	0.0295	0.0208

3.7 Conclusions

This study proposed an enhancement of PSFA by incorporating information of quality variables, leading to a new modeling paradigm termed as IOPSFA. The extracted latent variables using IOPSFA can represent the intrinsic properties of process in a better way, thus, providing a better prediction performance to the quality variables. We also considered the case of missing data while developing IOPSFA model and the results indicate that IOPSFA is robust to a wide range of missing data. We have tested the proposed framework by: 1) using an industrial case study, namely: a SAGD well pair water content soft sensor modeling, and 2) a tanks system experiment. The results of the two case studies have demonstrated that IOPSFA based soft sensors can provide improved prediction results for modeling dynamic data.

Chapter 4

Dynamic Latent Variable Modeling with Output Time-varying Time Delays *

Modeling the time-varying time delays in process industry has been a challenging task. Failure to estimate the delays may result in poor performance in system identification. Measuring or predicting of quality variables is critical to real-time control and process monitoring. However, time-varying time delays often exist in the quality variables of interest which are normally used as references in modeling process. To address time delay estimation problem, a probabilistic modeling approach is proposed and solved using variational Bayesian method in this study. The proposed method, along with the variational learning algorithm, not only uses the information of the reference samples but also address time-varying time-delay problem. The improved prediction performance is verified through a numerical example and a simulated chemical process.

4.1 Introduction

In process industry, measurements of quality variables are important both for real-time control and for product quality assurance. The methods to measure the quality variables can be generally categorized into two types: on-line analyzer and lab analysis. On-line analyzers can provide fast measurements, so they can be used for the

*Part of this chapter has been submitted as: Fan L, Huang B. Dynamic Latent Variable Modeling with Output Time-varying Time Delays to Journal of Process Control.

real-time control. But analyzers typically have reliability issues due to varying operation conditions and high cost to install or maintain them. On the contrary, lab analysis can provide more accurate measurements but often with large sampling interval, uncertain time delays or unpredictable human errors. This makes the lab analysis not appropriate for real-time control. An effective way to utilize the lab data is through soft sensing techniques [1, 83]. Soft sensor can provide estimations of quality variables using the fast-sampled process data along with the slow-sampled lab data. The lab analysis results play important roles in the soft sensor development in the sense that they are used as references for building data-driven soft sensor models [88] or calibrating the developed models. A soft sensor is essentially an inferential model, which describes the relationship between the fast-sampled process variables and desired typically slow-sampled quality variables [126]. Normally, there are three categories of inferential models: the first principle model, data-driven model, and their combination [83]. Each has its advantages and disadvantages. An accurate first principle model requires detailed process knowledge, which is often difficult to obtain. A data-driven model does not require accurate process knowledge although its performance highly depends on data quality, and many advanced data processing techniques are available.

In consideration of the difficulties to develop an accurate first principle model, and with the help of the well-developed data collection and storage techniques in process industries, inferential modeling using data-driven methods has gained a lot of attention and is becoming a common choice for soft sensor modeling. The measurement of a quality variable is normally obtained from the lab analysis, which is usually sparse and irregularly sampled with varying time delay [127]. This imposes a big challenge on developing a model. A process variable is considered to be labeled when the lab data is available; otherwise, it is called unlabeled. Semi-supervised modeling, which can use both labeled and unlabeled data, enhances the prediction ability of the inferential models by using both labeled and unlabeled data [128]. On the other side, additional challenges exist in the semi-supervised modeling methods. In addition to the collinearity between process variables, the sparsity, and uncertainty of lab data, i.e. human errors and varying time delay, make many traditional modeling methods not applicable [127]. To overcome the above difficulties, latent variable models

(LVM) have become popular in modeling practical data [115, 122, 129]. LVM methods make good use of a large amount of historical data by projecting them linearly or nonlinearly to a latent feature space. The latent features are more informative and normally have lower dimension than the original data space, i.e. reducing the redundant and non-informative information, such as noise. Different constraints are applied on the projection to make latent features possess desired properties in different LVM approaches, e.g. maximum variance or slowest variation, etc. Popular LVM approaches that have been widely adopted in process data analysis include but not limit to: Principle Component Analysis (PCA) [48, 130], Partial Least Square (PLS) [131, 132], Independent Component Analysis (ICA) [42], Slow Feature Analysis (SFA) [43] and their probabilistic counterparts, Probabilistic PCA (PPCA) [44], Probabilistic PLS (PPLS) [133], Probabilistic ICA (PICA) [46] and Probabilistic SFA (PSFA) [20, 47], etc. Latent features capture the underlying causes of variations in process variables. Each latent feature is expected to capture unique information or relationship between inputs and outputs. All latent features are expected to capture most or even all of the information carried by the data. Thus, large-dimensional process variables are compressed into small-dimensional features without losing essential information. However, most aforementioned approaches only extract latent features from input variables without accounting for any information from output variables, i.e. latent features are extracted in an unsupervised way. Although PLS and PPLS can utilize output information, they cannot reveal the dynamics in variables, eventually leading to a static model. However, IOPSFA, by incorporating both input and output information in a semi-supervised fashion, can extract dynamic latent features and in addition, can deal with the missing data problem.

A major challenge when dealing with lab data, as explained earlier, is the input-output time delay due to the procedure of lab sample collection and analysis. Time delay may be constant or time-varying. Many methods have been proposed to estimate the time delay, e.g. prediction error methods [34], impulse response methods [24] and adaptive methods. They all dealt with constant time delays. However, in reality, time delay can be time-varying. Certain distribution must be assumed to account for the uncertainty of time delay, e.g. uniform distribution [33], multinomial distribution [127], etc. In reality, the time delay may not only be time-varying,

but also correlated sequentially since material transportation is continuous. In this case, Markovian transition property may be considered to describe time delay variations [34]. In order to address the multi-rate problem and time-varying time delay problem at the same time when utilizing the lab data to build inferential models, IOPSFA with varying time delay named IOPSFA_VTD is proposed in this work under probabilistic and Bayesian framework. Categorical distribution is used to account for the uncertainty of time delays and a Markov chain is used to define the transition pattern of varying delay. The time delay sequence is considered as a set of latent variables and will be estimated along with latent slow features (SFs) and unknown parameters using variational Bayesian (VB) approach [116], which can provide the estimation of the posterior of both latent variables and unknown parameters. By using the estimated time delay between inputs and outputs, slow features can be extracted using the shifted output values without delays.

The remainder of the chapter is organized as follows. In the next section, PSFA and IOPSFA are briefly reviewed. In the third section, the formulation of IOSPFA_VTD is proposed and the detailed estimation steps to solve this problem using VB are presented. Following that, a numerical case study and simulation on the continuous stirred tank reactor (CSTR) process are provided to verify the proposed method. Conclusion is given in the final section.

4.2 Preliminary of Probabilistic Slow Feature Analysis

As a novel dynamic latent feature extraction method, PSFA extracts latent features that have temporal correlations and sorts features according to their slowness. Instead of describing the process dynamics in the observations directly, PSFA describes the process dynamics in the latent space. Since the latent features are treated as the common causes of the inputs and outputs, the dynamics in the observations can also be described through the emission relationship between latent features and observations. Turner and Sahani extended the probabilistic interpretation of SFA [47], and the following formulation is widely used in describing the dynamics in latent space and the relationships between observations and latent features.

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ X(t) = Hs(t) + e_x(t), e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \end{cases} \tag{4.1}$$

where Λ and Σ are the state and observation noise covariance matrices, respectively. With this Linear Dynamic System (LDS) formulation, PSFA assumes that each latent feature varies with a certain degree of slowness and latent variables are independent of each other. This independence property also imposes the state transition matrix F and state noise covariance matrix to be diagonal. The elements of F correspond to the varying speed of latent features. To ensure the extracted slow features having unit variance, PSFA constraints F and Λ with the following relationship

$$F^2 + \Lambda = I \tag{4.2}$$

where I is the identity matrix. The slow features can be extracted by maximizing the likelihood $p(X|F, H, \Sigma)$ through Expectation-Maximization (EM) method [20]. Considering the uncertainty of the parameters, a Bayesian approach has been proposed [116]. Beta distribution has been utilized as the prior distribution for the elements of F and the preference can be defined by manipulating the hyper-parameters of Beta distribution to make the elements of F as close to 1 as possible (the closer to 1, the slower the feature is [20]). Thus the posterior of each unknown parameter can be estimated; meanwhile, the slow features can be extracted. Based on this, the semi-supervised IOPSFA extends the observations that are used to extract slow features by adding the following equation in the formulation of PSFA

$$Y(t) = Us(t) + e_y(t), e_y(t) \sim \mathcal{N}(\mathbf{0}, \Gamma) \tag{4.3}$$

This makes IOPSFA capable of extracting SFs that have better predictability than the SFs that are extracted only from inputs. If outputs are available at all time instances, IOSPFA essentially is of the same procedure as PSFA and it becomes a supervised learning method. The merit of IOSPFA lies mostly in taking advantage of the information carried by outputs and dealing with the missing data or multi-rate data. The formulation of IOPSFA captures the dynamics in outputs, but it assumes inputs are correctly labeled by outputs with no time delays between inputs and outputs which is restrictive when dealing with practical data that has time delays.

In the next section, a new formulation is proposed to account for the input-output time delay and the extracted features are expected to have better predictability than the existing methods.

4.3 Modeling and Variational Inference of IOPSFA with Output Time-varying Time Delays

4.3.1 Formulation of IOPSFA with Output Time-varying Time Delays

In order to model time delays between input and output in the presence of the latent variables, the proposed formulation of IOPSFA with output time-varying time delay is given below by introducing the delay-free outputs $\tilde{Y}$

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ X(t) = Hs(t) + e_x(t), e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \\ \tilde{Y}(t) = Us(t) + e_y(t), e_y(t) \sim \mathcal{N}(\mathbf{0}, \Gamma) \end{cases} \tag{4.4}$$

where, $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ are measurement noise covariance matrices for inputs and outputs, respectively. The diagonal structure of Σ and Γ shows the independence between each input and output variables. The dimensions of feature space $\mathcal{S}$, inputs space $\mathcal{X}$ and outputs space $\mathcal{Y}$ are q, m and l, respectively. Similar to PSFA, the diagonal components of F: $\{\lambda_j, 1 \leq j \leq q\}$ control the varying speed of slow features. $\Lambda = \text{diag}\left\{1 - \lambda_1^2, \cdots, 1 - \lambda_q^2\right\}$ is the states covariance matrix. Slow features derived from IOPSFA_VTD also have unit variance, i.e. $F^2 + \Lambda = I_q$. The latent variable s connects the input X and delay-free output $\tilde{Y}$. Since the delays considered in this work occur in the output, as a result, we can consider that there is no delay between s and X and the delays only exist between s and $\tilde{Y}$, which is equivalent to the input-output delay scenario. In the above formulation (4.4), $\tilde{Y}(t)$ is the measurement reconstructed from the raw measurements: $Y(t-K), \cdots, Y(t)$. K is the maximum possible time delay among all elements of $Y(t)$. Then $\tilde{Y}(t)$ can be reconstructed as follows

$$\tilde{Y}(t) = [Y_1(t + k_1), \cdots, Y_j(t + k_j), \cdots, Y_l(t + k_l)], \quad 1 \leq j \leq l \tag{4.5}$$

where, Y_j represents j-th element of Y and $k_j \in \{0, 1, 2, \cdots, K\}$ is the time delay for Y_j with reference to $s(t)$.

Most of the time, Y represents the quality variables and it only has one dimension. And if Y has more than one dimension and time delays exist between different dimensions, we can decompose Y into multiple vectors and build a separate model for each element of Y.

4.3.1.1 Time Delay Indicator I

With the reconstructed $\tilde{Y}(t)$, slow features $s(t)$ can be extracted from $X(t)$ and $\tilde{Y}(t)$ following the IOPSFA procedure [128]. To indicate the time delay between $\tilde{Y}(t)$ and $Y(t)$, an indicator variable $I \in \mathbb{R}^{l\times(K+1)\times T}$ is introduced, and T is the total number of samples. At time t, $I(t) = [I_1(t)^T, I_2(t)^T, \cdots, I_l(t)^T]^T$ and j-th row of $I_j(t) = [I_j^{(0)}(t), I_j^{(1)}(t), \cdots, I_j^{(K)}(t)], 1 \leq j \leq l$ indicates the time delay of j-th dimension of outputs. The structure of $I(t)$ is shown in Figure 4.1 $I(t)$ has the

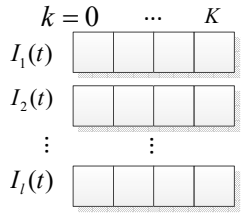

Figure 4.1: Graphical structure of the indicator variable $I(t)$

following property:

$$\forall j \in \{1, \cdots, l\}, \sum_{k=0}^{K} I_j^{(k)}(t) = 1, I_j^{(k)}(t) \in \{0, 1\} \tag{4.6}$$

The prior of the initial time delay indicator $I_j(1)$ is represented as π_j:

$$\pi_j = \{\pi_j^{(k)}\} : \pi_j^{(k)} = p(I_j^{(k)}(1) = 1) \tag{4.7}$$

From the definition and property of $I(t)$, only one component of $I_j(t)$ can take value 1, e.g. $I_j^{(k)}(t) = 1$ indicates that k is the time delay between $Y_j(t)$ and $X(t)$. It means $X(t-k)$ will take k sampling time to impact the output $Y_j(t)$

$$X(t-k) \xrightarrow[I_j^{(k)}(t)=1]{\text{delay}=k} Y_j(t) \tag{4.8}$$

And the corresponding time delay indicator is represented in Figure 4.2 Since the

	0	...	k	...	K
$I_j(t)$	0	0	1	0	0

Figure 4.2: Representation of $I_j(t)$ when delay $= k$

time delay for each output varies along time and practically, time delay could increase, decrease or stay at the same value. For example, at time t, the time delay for an output is d, and at time $t+1$, time delay could stay at value d in a higher chance since at most of the time, the process is continuous and steady. However, there are also chances that time delay could increase or decrease due to process disturbances, operation condition changes or measurement procedure variant. If outputs are quality variables that are analyzed through the lab, these samples are typically slow rate samples and have uncertain delays comparing to the fast rate process variables. In both cases, we assume the time delay follows a markovian chain and a transition matrix M_j for j-th-dimensional output can be utilized to describe this transition behavior. The elements of M_j represent the transition probability from one time delay value to the next

$$M_j^{(d_{t-1},d_t)} = p(I_j^{(d_t)}(t) = 1 | I_j^{(d_{t-1})}(t-1) = 1),$$
$$1 \leq j \leq l,\ 0 \leq d_{t-1}, d_t \leq K \tag{4.9}$$

where d_t and d_{t-1} represent the time delay at time t and $t-1$, respectively. For example, $M_j^{(1,2)}$ represents the probability that time delay changes from 1 at time $t-1$ to 2 at time t. Transition matrix M_j could be known as a prior by incorporating process knowledge or its elements can also be estimated as unknown parameters in the proposed algorithm. The elements in M_j are constrained by the following relationship

$$\sum_{d_t=0}^{K} M_j^{(d_{t-1},d_t)} = 1, \tag{4.10}$$

which means each row vector of M_j has the summation to 1.

4.3.1.2 Probabilistic Dependencies

A) Prior Assignment

In this problem, the unknown parameter set is $\Theta = \{F, H, U, \Sigma, \Gamma, M\}$ and the latent variable set is $L = \{s, \pi_j, I_j, 1 \leq j \leq l\}$. The unobserved variable set combines them, denoted as $Z = \{\Theta, L\}$. To extract the features and estimate the time delay sequence I and other unknown parameters under the Bayesian framework, proper priors need to be assigned to all unobserved variables. Priors represent the process knowledge that we want to incorporate in models and they also provide the preferences in the modeling process. The priors of time delay variables $I(t)$, π and parameter M are assigned and explained in this subsection. The priors of other unknown parameters are similarly defined as in work [116]. Here we provide the priors assignment details for unknown parameters:

- Feature transition matrix parameter F:
  Since the elements of F, i.e. $\lambda_{1:q}$, control the varying speed of the latent features, the prior of each $\lambda_j, 1 \leq j \leq q$ is chosen as Beta distribution [116] since the varying speed, represented by the eigenvalues, is normally restricted in (0,1), where 0 represents no varying and 1 represents random walk. The shape of the probability density function (pdf) of beta distribution can be manipulated by tuning the shape parameters $\alpha_{\lambda_0}, \beta_{\lambda_0}$ to have the preference of $\lambda_j \to 1$ to make the extracted features as slow as possible:

$$p(\lambda_j|\alpha_{\lambda_0}, \beta_{\lambda_0}) = Beta(\lambda_j|\alpha_{\lambda_0}, \beta_{\lambda_0}) \tag{4.11}$$

  And the conditional distribution of $s_j(t)$ is

$$p(s(t)|s(t-1), F, \Lambda) = \mathcal{N}(s(t)|Fs(t-1), \Lambda) \tag{4.12}$$

- Observation (emission) matrices H and U:
  If H is parameterized as row vectors, i.e. $H = [h_1, \cdots, h_m]^T$, the normal distribution is used for the prior of each row h_i

$$p(h_i|0, \Sigma_{h_0}) = \mathcal{N}(h_i|0, \Sigma_{h_0}) \tag{4.13}$$

  Similarly, since $U = [u_1, \cdots, u_l]^T$, the prior of each row u_i is expressed as

$$p(u_i|0, \Sigma_{u_0}) = \mathcal{N}(u_i|0, \Sigma_{u_0}) \tag{4.14}$$

where, Σ_{h_0} and Σ_{u_0} are the hyper-parameters for the normal distribution, respectively.

- Observation noise covariance matrices Σ and Γ:
  Since $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and the observation $X_i(t)$ follows the normal distribution with fixed mean (zero mean), its conjugate prior is inverse gamma distribution

$$p(\sigma_i^2|\alpha_{\sigma_0}, \beta_{\sigma_0}) = Inv\text{-}Gamma(\sigma_i^2|\alpha_{\sigma_0}, \beta_{\sigma_0}) \tag{4.15}$$

  Similarly, since $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$, the prior of γ_j^2 is also inverse gamma distribution

$$p(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) = Inv\text{-}Gamma(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) \tag{4.16}$$

- Latent variable $I(t)$, π and parameter M:
  Latent variable $I(t)$Time delay transits in a markovian pattern, defined by M_j. For each $I_j, 1 \leq j \leq l$:

$$p(I_j(1)|\pi_j) = Cat(I_j(1)|\pi_j) = \prod_{k=0}^{K} [\pi_j^{(k)}]^{I_j^{(k)}(1)} \tag{4.17}$$

$$p(I_j|\pi_j) = \prod_{t=2}^{T} p(I_j(t)|I_j(t-1), M_j) \cdot p(I_j(1)|\pi_j) \tag{4.18}$$

  where, π_j is the hyper-parameter of the Categorical distribution and it follows the Dirichlet distribution. π_j is a vector with the same dimensionality as $I_j(t)$ and the sum of all the elements of π_j is equal to one. The element of π_j represents the probability of the corresponding value of time delay, so all the elements of π_j take positive values. The value of π_j can be known by incorporating the process knowledge if available. Alternatively, Dirichlet distribution can be assigned to it as prior since it is the conjugate prior of the categorical distribution.

$$p(\pi_j|\alpha_{\pi_0}) = Dir(\pi_j|\alpha_{\pi_0}) = \frac{1}{B(\alpha_{\pi_0})} \prod_{k=0}^{K} (\pi_j^{(k)})^{\alpha_{\pi_0}^{(k)}-1} = \frac{\Gamma(\sum_{k=0}^{K} \alpha_{\pi_0}^{(k)})}{\prod_{k=0}^{K} \Gamma(\alpha_{\pi_0}^{(k)})} \prod_{k=0}^{K} (\pi_j^{(k)})^{\alpha_{\pi_0}^{(k)}-1} \tag{4.19}$$

Choose to use the symmetric Dirichlet priors with a fixed strength $f^{(\pi_j)}$:

$$\alpha_{\pi_0} = \Big[\frac{f^{(\pi_j)}}{K+1}, \cdots, \frac{f^{(\pi_j)}}{K+1}\Big], \quad s.t.\ f^{(\pi_j)} = \sum_{k=0}^{K} \alpha_{\pi_0}^{(k)} \tag{4.20}$$

Similarly, each row of the transition matrix M_j follows a Dirichlet distribution:

$$p(M_j|\alpha_{M_0}) = \prod_{k=0}^{K} Dir(\{M_j^{(k,0)}, \cdots, M_j^{(k,K)}\}|\{\alpha_{M_0}^{(k,0)}, \cdots, \alpha_{M_0}^{(k,K)}\})$$

with strength $f^{(M_j)}$:

$$\alpha_{M_0} = \Big[\frac{f^{(M_j)}}{K+1}, \cdots, \frac{f^{(M_j)}}{K+1}\Big], \quad s.t.\ f^{(M_j)} = \sum_{k=0}^{K} \alpha_{M_0}^{(k)} \tag{4.21}$$

B) Probabilistic Graphic Model

To summarize all the probabilistic dependencies defined above, the probabilistic graphic model is presented in Figure 4.3. The grey circles represent the observations

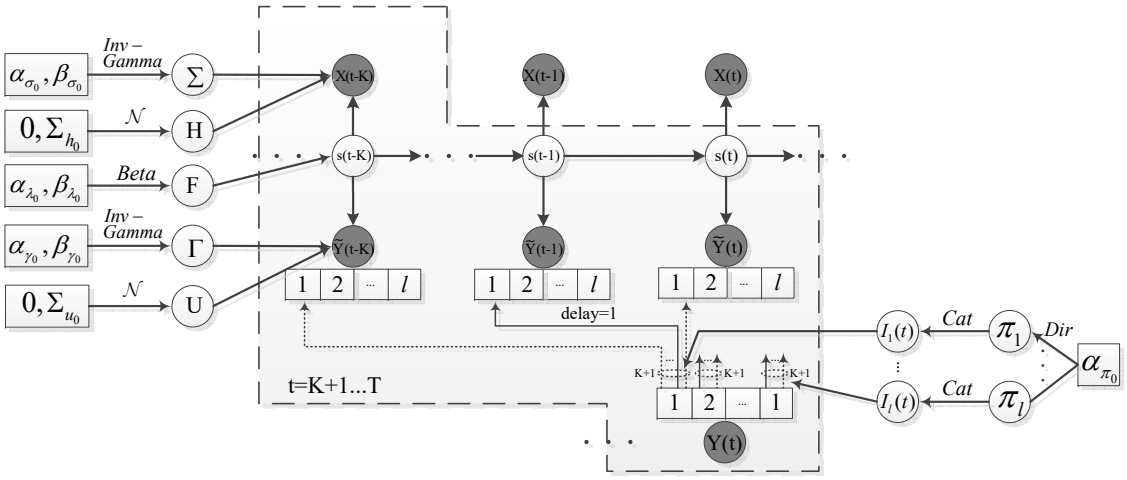

Figure 4.3: Graphical structure of IOPSFA with time-varying time delays

and white ones represent unobserved variables to be estimated. Rectangles represent the known hyper-parameters for the distributions of unobserved variables. $Y_j(t)$ could take value from $\tilde{Y}_j(t-K)$ to $\tilde{Y}_j(t)$ due to the assumption of maximum K delays, e.g. $Y_1(t)$ in Figure 4.3, could be delayed from the samples of : $\{\tilde{Y}_1(t-K), \tilde{Y}_1(t-K+1), \cdots, \tilde{Y}_1(t)\}$. The solid line from $Y_1(t)$ represents the actual time delay and there is only one solid line for each sample of $Y_j(t)$ according to the property of I in (4.6). Figure 4.3 illustrates the case that the delay time for $Y_1(t)$ is 1.

In this time-varying time delay problem, time delay could decrease or increase, which could cause the following two issues: missing values in $\tilde{Y}$ or conflict values in $\tilde{Y}$. Taking one-dimensional Y as an example, these two scenarios are illustrated in Figure 4.4 and Figure 4.5.

1. Delay decrease causing missing value in $\tilde{Y}$:
   As shown in Figure 4.4, delay for $Y(t+1)$ decreases by 1 comparing with $Y(t)$, which causes the missing of $\tilde{Y}(t-1)$. The missing sample will be skipped when inferring the latent feature $s(t-1)$. It is worth noticing that, in this case, there is no measurement information missing in inferring feature s, since all measurement of Y are used when reconstructing $\tilde{Y}(t-2)$ and $\tilde{Y}(t)$.

2. Delay increase causing conflict in $\tilde{Y}$:
   As shown in Figure 4.5, delay for $Y(t+1)$ increases by 1 comparing with $Y(t)$, which causes the conflict value in $\tilde{Y}(t-2)$ since it takes information from both measurements. The proposed model has the ability to resolve this issue by fusing two observation samples under the probabilistic formulation.

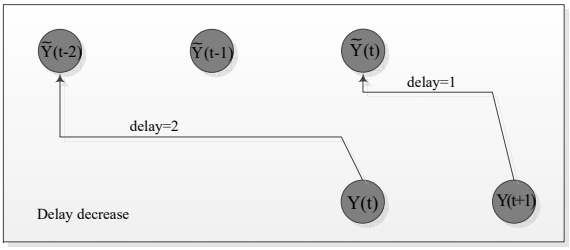

Figure 4.4: Delay decrease

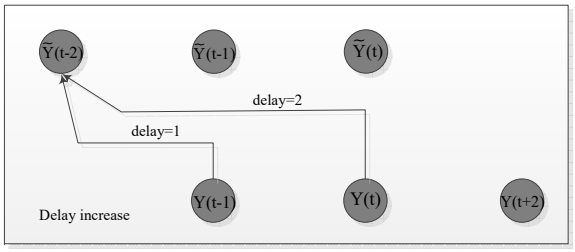

Figure 4.5: Delay increase

4.3.2 Variational Bayesian Inference

In order to extract slow features and learn the posterior distribution of unknown parameters at the same time, variational Bayesian inference method is used instead of maximum a posterior (MAP) estimation, which can only obtain the point estimation of unknown parameters. The approximation strategy is adopted in the Bayesian inference procedure for computational feasibility. In order to model the log-evidence, we introduce a variational distribution of unobserved variables $q(Z)$ and decompose

the log evidence

$$\begin{aligned}\ln p(X,Y) &= \int_{q(Z)} q(Z) \ln \frac{p(X,Y,Z)}{p(Z|X,Y)} \mathrm{d}Z \\ &= \int_{q(Z)} q(Z) \ln \frac{p(X,Y,Z)}{q(Z)} \mathrm{d}Z + \int_{q(Z)} q(Z) \ln \frac{q(Z)}{p(Z|X,Y)} \mathrm{d}Z \\ &= F(q(Z)) + KL(q(Z)||p(Z|X,Y)).\end{aligned} \tag{4.22}$$

The log evidence has been decomposed into two terms: the first term $F(q(Z))$, also called variational free energy, is the lower bound of the log evidence. The second term $KL\big(q(Z)||p(Z|X,Y)\big)$ is the Kullback-Leibler (KL) divergence between the proposal variational distribution $q(Z)$ and true posterior distribution $p(Z|X,Y)$. Since the log evidence is constant, minimizing the KL divergence is equivalent to maximizing the variational free energy. To achieve this, the proposal distribution can be factorized according to the mean-field approximation to approximate the posterior distribution $P(Z|X,Y)$ [73].

$$q(Z) = \prod_j q_j(Z_j) \xrightarrow{approx.} P(Z|X,Y) \tag{4.23}$$

in which, Z_j represents a variable group that contains one or several unobserved variables. Normally, maximizing the variational free energy is complicated and not analytically solvable due to multiple integrations involved. It is difficult to obtain the optimal value for all observed variables simultaneously. Taking advantage of the mean-field approximation, we can maximize $F(q(Z))$ with respect to one group of unobserved variables while fixing all others with the following factorization [134] in each iteration

$$q(Z) = q(H) \cdot q(U) \cdot q(\Sigma) \cdot q(\Gamma) \cdot q(I,\pi,M) \cdot q(s) \cdot q(\lambda_{1:q}) \tag{4.24}$$

Apparently, we assume that each variable group contributes independently to the target multivariate posterior [135]. With such kind of factorization, the general solution is given by [73]

$$q_j^*(Z_j) \propto \exp\Big(\mathbb{E}_{q_{\setminus j}(Z_{\setminus j})}\big[\ln p(X,Y,Z)\big]\Big) \tag{4.25}$$

where $q_j^*(Z_j)$ represents the optimal solution of $q_j(Z_j)$. $Z_{\setminus j}$ represents all other unobserved variables except Z_j. For convenience, notation $q_j(Z_j)$ is simplified as q_j in the following derivations. Next, the optimal solution for each variables will be provided.

4.3.2.1 Inference of H, U, Σ, and Γ

The derivations of the update equations for H, U, Σ and Γ are straight forward and will be provided here. As mentioned before, the posterior distribution of H and U follow Normal distribution and Σ and Γ follow Inverse-Gamma distribution. They can be iteratively updated by the equations derived in this subsection.

- Inference of H

Since $H = [h_1, \cdots, h_m]^T \in \mathbb{R}^{m\times q}$, and for each dimension $h_i \in \mathbb{R}^{q\times 1}, 1 \leq i \leq m$ has a normal distribution as prior in (4.13) and its posterior to be inferred is:

$$\begin{aligned} q^*(h_i) =& \mathcal{N}(h_i|\hat{\mu}_{h_i}, \hat{\Sigma}_{h_i}) \\ \Rightarrow \ln q^*(h_i) =& -\frac{1}{2}(h_i - \hat{\mu}_{h_i})^T\hat{\Sigma}_{h_i}^{-1}(h_i - \hat{\mu}_{h_i}) - \frac{1}{2}\ln(2\pi)^d\hat{\Sigma}_{h_i} \end{aligned} \tag{4.26}$$

where, the parameters with $\hat{}$ represent the hyper-parameters of the optimal posterior distribution of the corresponding unknown parameter. According to the model formulation in (4.4), only the i-th dimension of X depends on h_i: $p(X_i(t)|h_i, s(t)) = \mathcal{N}(h_i^T s(t), \sigma_i^2)$. Applying the general solution in (4.25)

$$\begin{aligned} \ln q_{h_i}^* \propto& \mathbb{E}_{q_{\setminus h_i}(Z_{\setminus h_i})}\big[\ln p(X, Y, Z)\big] \\ =& \Big\langle \big[\ln p(X_i|h_i, s, \sigma_i^2)p(h_i|0, \Sigma_{h_0})\big] \Big\rangle \\ =& \Big\langle \Big[\sum_{t=1}^{T}\sum_{k=0}^{K} \ln\mathcal{N}(X_i(t-k)|h_i^T s(t), \sigma_i^2) + \ln\mathcal{N}(0, \Sigma_{h_0})\Big] \Big\rangle \\ =& \Big\langle \Big[\sum_{t=1}^{T}\sum_{k=0}^{K} \Big(-\frac{1}{2}\big(X_i(t-k) - h_i^T s(t)\big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t-k) - h_i^T s(t)\big) - \frac{1}{2}\ln(2\pi)\sigma_i^2\Big) \\ & - \frac{1}{2}h_i^T\Sigma_{h_0}^{-1}h_i^T - \frac{1}{2}\ln(2\pi)^q\Sigma_{h_0}\Big] \Big\rangle \\ =& \Big\langle \Big[-\frac{1}{2}h_i^T\Sigma_{h_0}^{-1}h_i^T - \frac{1}{2}\ln(2\pi)^q\Sigma_{h_0} - \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big\{\Big[X_i(t-k)X_i(t-k) + \big(h_i^T s(t)\big)^T\big(h_i^T s(t)\big) \\ & - X_i(t-k)h_i^T s(t) - \big(h_i^T s(t)\big)^T X_i(t-k)\Big] \cdot \sigma_i^{-2} - \frac{1}{2}\ln(2\pi)\sigma_i^2\Big\}\Big] \Big\rangle \\ =& \Big\langle \Big[-\frac{1}{2}h_i^T\Big(\Sigma_{h_0}^{-1} + \sigma_i^{-2}\sum_{t=1}^{T}\sum_{k=0}^{K} s(t)s^T(t)\Big)h_i - \sum_{t=1}^{T}\sum_{k=0}^{K} h_i s^T(t)X_i(t-k)\sigma_i^{-2} + \cdots\Big] \Big\rangle \end{aligned} \tag{4.27}$$

Comparing the quadratic term and linear term with respect to h_i in (4.26) and (4.27), the update equation for the hyper-parameters of h_i are derived as:

$$\hat{\Sigma}_{h_i}^{-1} = \Sigma_{h_0}^{-1} + \Big\langle \sigma_i^{-2} \sum_{t=1}^{T} \sum_{k=0}^{K} \big[s(t) s^T(t) \big] \Big\rangle = \Sigma_{h_0}^{-1} + \big\langle \sigma_i^{-2} \big\rangle \cdot \sum_{t=1}^{T} \big\langle s(t) s^T(t) \big\rangle \tag{4.28}$$

$$\hat{\mu}_{h_i} = \hat{\Sigma}_{h_i} \cdot \big\langle \sigma_i^{-2} \big\rangle \cdot \sum_{t=1}^{T} \sum_{k=0}^{K} \big\langle s(t) X_i(t-k) \big\rangle = \hat{\Sigma}_{h_i} \cdot \big\langle \sigma_i^{-2} \big\rangle \cdot \sum_{t=1}^{T} X_i(t) \big\langle s(t) \big\rangle \tag{4.29}$$

here, $\langle \cdot \rangle$ is the expectation operator and we will use it to simplify the derivation in following sections.

- Inference of Σ

Since $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and σ_i^2 has inverse gamma distribution as its prior in (4.15) and its posterior to be inferred is:

$$\begin{aligned} q^*(\sigma_i^2) =& \textit{Inv-Gamma}(\sigma_i^2 | \hat{\alpha}_{\sigma_i}, \hat{\beta}_{\sigma_i}) \\ \Rightarrow \ln q^*(\sigma_i^2) =& \hat{\alpha}_{\sigma_i} \ln \hat{\beta}_{\sigma_i} - (\hat{\alpha}_{\sigma_i} + 1) \ln \sigma_i^2 - \hat{\beta}_{\sigma_i} \sigma_i^{-2} - \ln \Gamma(\hat{\alpha}_{\sigma_i}) \end{aligned} \tag{4.30}$$

The likelihood follows a normal distribution:

$$p(X_i(t) | h_i, s, \sigma_i^2) = \mathcal{N}(X_i(t) | h_i^T s(t), \sigma_i^2) \tag{4.31}$$

According to general solution in (4.25), the optimal posterior distribution of σ_i^2 can be derived as:

$$\begin{aligned} \ln q_\Sigma^* \propto& \mathbb{E}_{q_{\setminus\Sigma}(Z_{\setminus\Sigma})} \big[\ln p(X, Y, Z) \big] \\ =& \Big\langle \ln p(X_i | h_i, s, \sigma_i^2) \cdot p(\sigma_i^2 | \alpha_{\sigma_0}, \beta_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \mathcal{N}(X_i(t) | h_i^T s(t), \sigma_j^2) + \ln \textit{Inv-Gamma}(\alpha_{\sigma_0}, \beta_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \Big[-\frac{1}{2} \big(X_i(t) - h_i^T s(t) \big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t) - h_i^T s(t) \big) - \frac{1}{2} \ln(2\pi) + \frac{1}{2} \ln \sigma_i^{-2} \Big] \\ & + \alpha_{\sigma_0} \ln \beta_{\sigma_0} - (\alpha_{\sigma_0} + 1) \ln \sigma_i^2 - \beta_{\sigma_0} \sigma_i^{-2} - \ln \Gamma(\alpha_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \Big(\Big[-\frac{1}{2} \big(X_i(t) - h_i^T s(t) \big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t) - h_i^T s(t) \big) \Big] - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln \sigma_i^2 \Big) \\ & + \alpha_{\sigma_0} \ln \beta_{\sigma_0} - (\alpha_{\sigma_0} + 1) \ln \sigma_i^2 - \beta_{\sigma_0} \sigma_i^{-2} - \ln \Gamma(\alpha_{\sigma_0}) \Big\rangle \end{aligned} \tag{4.32}$$

Comparing the coefficients of $\ln \sigma_i^2$ term and linear term of σ_i^{-2} term for (4.30) and (4.32), the update equation for the hyper-parameters of Σ are derived as:

$$-(\hat{\alpha}_{\sigma_i}+1) = -\frac{T}{2} - (\alpha_{\sigma_0}+1)k \Rightarrow \hat{\alpha}_{\sigma_i} = \frac{T}{2} + \alpha_{\sigma_0} \tag{4.33}$$

And,

$$\begin{aligned}
-\hat{\beta}_{\sigma_i} &= -\beta_{\sigma_0} + \Big\langle \sum_{t=1}^{T} \Big[-\frac{1}{2}\big(X_i(t) - h_i^T s(t)\big)^T \cdot \big(X_i(t) - h_i^T s(t)\big)\Big]\Big\rangle \\
\Rightarrow \hat{\beta}_{\sigma_i} &= \beta_{\sigma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T} \Big[X_i(t)X_i(t) - 2X_i(t)h_i^T s(t) + s(t)^T h_i h_i^T s(t)\Big]\Big\rangle \\
\Rightarrow \hat{\beta}_{\sigma_i} &= \beta_{\sigma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T} \Big[X_i(t)X_i(t) - 2X_i(t)h_i^T s(t) + tr\big(s(t)^T h_i h_i^T s(t)\big)\Big]\Big\rangle \\
&= \beta_{\sigma_0} + \frac{1}{2}\sum_{t=1}^{T} \Big[X_i(t)X_i(t) - 2X_i(t)\big\langle h_i^T\big\rangle\big\langle s(t)\big\rangle + tr\big[\big\langle h_i h_i^T\big\rangle \cdot \big\langle s(t)s^T(t)\big\rangle\big]\Big]
\end{aligned} \tag{4.34}$$

- Inference of U

Since $U = [u_1, \cdots, u_l]^T \in \mathbb{R}^{l\times q}$, and for each dimension $u_j \in \mathbb{R}^{q\times 1}, 1 \le j \le l$ has normal distribution as prior in (4.14) and its posterior to be inferred is:

$$\begin{aligned}
q^*(u_j) =& \mathcal{N}(u_j|\hat{\mu}_{u_j}, \hat{\sigma}_{u_j}^2) \\
\Rightarrow \ln q^*(u_j) =& -\frac{1}{2}(u_j - \hat{\mu}_{u_j})^T \hat{\sigma}_{u_j}^{-2}(u_j - \hat{\mu}_{u_j}) - \frac{1}{2}\ln(2\pi)^q \hat{\sigma}_{u_j}^2
\end{aligned} \tag{4.35}$$

Only j-th dimension of Y depends on u_j: $p(Y_j(t)|u_j, s(t), I_j)$ and according to the

general solution in (4.25)

$$
\begin{aligned}
\ln q^*_{u_j} \propto & \mathbb{E}_{q_{\setminus u_j}(Z_{\setminus u_j})}\big[\ln p(X,Y,Z)\big] \\
=& \Big\langle \ln p(Y_j|u_j,s,I_j,\gamma_j^2)p(u_j|0,\sigma^2_{u_0})\Big\rangle \\
=& \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K}\big[\ln \mathcal{N}(Y_j(t+k)|u_j^T s(t),\gamma_j^2)\big]^{I_j^{(k)}(t+k)} + \ln \mathcal{N}(u_j|0,\sigma^2_{u_0})\Big\rangle \\
=& \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K}\Big(-\frac{1}{2}\big(Y_j(t+k)-u_j^T s(t)\big)^T\cdot\gamma_j^{-2}\cdot\big(Y_j(t+k)-u_j^T s(t)\big)-\frac{1}{2}\ln(2\pi)\gamma_j^2\Big) \\
& \cdot I_j^{(k)}(t+k) - \frac{1}{2}u_j^T\sigma^{-2}_{u_0}u_j^T - \frac{1}{2}\ln(2\pi)^q\sigma^2_{u_0}\Big\rangle \\
=& \Big\langle -\frac{1}{2}u_j^T\sigma^{-2}_{u_0}u_j^T - \frac{1}{2}\ln(2\pi)^q\sigma^2_{u_0} - \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big\{\Big[Y_j(t+k)Y_j(t+k)+\big(u_j^T s(t)\big)^T\big(u_j^T s(t)\big) \\
& - Y_j(t+k)u_j^T s(t) - \big(u_j^T s(t)\big)^T Y_j(t+k)\Big]\cdot\gamma_j^{-2} - \frac{1}{2}\ln(2\pi)\gamma_j^2\Big\}\cdot I_j^{(k)}(t+k)\Big\rangle \\
=& \Big\langle -\frac{1}{2}u_j^T\Big(\sigma^{-2}_{u_0}+\gamma_j^{-2}\sum_{t=1}^{T}\sum_{k=0}^{K}s(t)s^T(t)I_j^{(k)}(t+k)\Big)u_j \\
& - \sum_{t=1}^{T}\sum_{k=0}^{K}u_j^T s^T(t)Y_j(t+k)\gamma_j^{-2}I_j^{(k)}(t+k)+\cdots\Big\rangle
\end{aligned}
\tag{4.36}
$$

Comparing the quadratic term and linear term with respect to u_j for (4.35) and (4.36), the update equation for the hyper-parameters of u_j are derived as:

$$
\begin{aligned}
\hat{\sigma}^{-2}_{u_j} =& \sigma^{-2}_{u_0} + \Big\langle \gamma_j^{-2}\sum_{t=1}^{T}\sum_{k=0}^{K}\big[s(t)s^T(t)\big]I_j^{(k)}(t+k)\Big\rangle \\
=& \sigma^{-2}_{u_0} + \big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t=1}^{T}\sum_{k=0}^{K}\big\langle s(t)s^T(t)\big\rangle\big\langle I_j^{(k)}(t+k)\big\rangle
\end{aligned}
\tag{4.37}
$$

$$
\hat{\mu}_{u_j} = \hat{\sigma}^2_{u_j}\cdot\big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t=1}^{T}\sum_{k=0}^{K}Y_j(t+k)\big\langle s(t)\big\rangle\big\langle I_j^{(k)}(t+k)\big\rangle \tag{4.38}
$$

Since the output Y(t) is not always available, we adopt the similar technique as IOPSFA, in that T can be divided into two parts, $T = \{T_{obs}, T_{mis}\}$, in which, T_{obs} represents the time stamps at which inputs are labeled and T_{mis} represents the time stamps at which inputs are unlabeled, i.e., outputs are missing. The updating equa-

tion (4.37) and (4.38) become:

$$\hat{\sigma}_{u_j}^{-2} = \sigma_{u_0}^{-2} + \langle \gamma_j^{-2} \rangle \cdot \sum_{t \in T_{obs}} \sum_{k=0}^{K} \langle s(t) s^T(t) \rangle \langle I_j^{(k)}(t+k) \rangle \tag{4.39}$$

$$\hat{\mu}_{u_j} = \hat{\sigma}_{u_j}^2 \cdot \langle \gamma_j^{-2} \rangle \cdot \sum_{t \in T_{obs}} \sum_{k=0}^{K} Y_j(t+k) \langle s(t) \rangle \langle I_j^{(k)}(t+k) \rangle \tag{4.40}$$

- Inference of Γ

Since $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ and γ_j^2 has inverse gamma distribution as its prior in (4.16) and its posterior to be inferred is:

$$\begin{aligned} q^*(\gamma_j^2) =& Inv\text{-}Gamma(\gamma_j^2 | \hat{\alpha}_{\gamma_j}, \hat{\beta}_{\gamma_j}) \\ \Rightarrow \ln q^*(\gamma_j^2) =& \hat{\alpha}_{\gamma_j} \ln \hat{\beta}_{\gamma_j} - (\hat{\alpha}_{\gamma_j} + 1) \ln \gamma_j^2 - \hat{\beta}_{\gamma_j} \gamma_j^{-2} - \ln \Gamma(\hat{\alpha}_{\gamma_j}) \end{aligned} \tag{4.41}$$

The likelihood follows a normal distribution:

$$p(Y_j | u_j, s, \gamma_j^2, I_j) = \sum_{t=1}^{T} \sum_{k=0}^{K} \mathcal{N}(Y_j(t+k) | u_j^T s(t), \gamma_j^2)^{I_j^{(k)}(t+k)} \tag{4.42}$$

According to general solution in (4.25), the optimal posterior distribution of γ_j^2 can be derived as:

$$\begin{aligned} \ln q_\Gamma^* \propto& \mathbb{E}_{q_{\setminus\Gamma}(Z_{\setminus\Gamma})} \big[\ln p(X, Y, Z) \big] \\ =& \Big\langle \ln p(Y_j | u_j, s, \gamma_j^2, I_j) \cdot p(\gamma_j^2 | \alpha_{\gamma_0}, \beta_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \sum_{k=0}^{K} \mathcal{N}(Y_j(t+k) | u_j^T s(t), \gamma_j^2)^{I_j^{(k)}(t+k)} + \ln Inv\text{-}Gamma(\alpha_{\gamma_0}, \beta_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \sum_{k=0}^{K} \Big[-\frac{1}{2} \big(Y_j(t+k) - u_j^T s(t) \big)^T \cdot \gamma_j^{-2} \cdot \big(Y_j(t+k) - u_j^T s(t) \big) - \frac{1}{2} \ln(2\pi) + \frac{1}{2} \ln \gamma_j^{-2} \Big] \\ & \cdot I_j^{(k)}(t+k) + \alpha_{\gamma_0} \ln \beta_{\gamma_0} - (\alpha_{\gamma_0} + 1) \ln \gamma_j^2 - \beta_{\gamma_0} \gamma_j^{-2} - \ln \Gamma(\alpha_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{k=0}^{K} \Big(\sum_{t=1}^{T} \Big[-\frac{1}{2} \big(Y_j(t+k) - u_j^T s(t) \big)^T \cdot \gamma_j^{-2} \cdot \big(Y_j(t+k) - u_j^T s(t) \big) \Big] - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln \gamma_j^2 \Big) \\ & \cdot I_j^{(k)}(t+k) + \alpha_{\gamma_0} \ln \beta_{\gamma_0} - (\alpha_{\gamma_0} + 1) \ln \gamma_j^2 - \beta_{\gamma_0} \gamma_j^{-2} - \ln \Gamma(\alpha_{\gamma_0}) \Big\rangle \end{aligned} \tag{4.43}$$

Comparing the coefficients of $\ln \gamma_j^2$ and linear term of γ_j^{-2} term for (4.41) and (4.43), the update equation for the hyper-parameters of Γ are derived as:

$$-(\hat{\alpha}_{\gamma_j} + 1) = -\frac{1}{2} \sum_{t=1}^{T} \sum_{k=0}^{K} \langle I_j^{(k)}(t+k) \rangle - (\alpha_{\gamma_0} + 1) \Rightarrow \hat{\alpha}_{\gamma_j} = \frac{1}{2} \sum_{t=1}^{T} \sum_{k=0}^{K} \langle I_j^{(k)}(t+k) \rangle + \alpha_{\gamma_0} \tag{4.44}$$

And,

$$
\begin{aligned}
-\hat{\beta}_{\gamma_j} =& -\beta_{\gamma_0} + \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K}\Big[-\frac{1}{2}\big(Y_j(t+k)-u_j^T s(t)\big)^T\cdot\big(Y_j(t+k)-u_j^T s(t)\big)\Big]\cdot I_j^{(k)}(t+k)\Big\rangle \\
\Rightarrow \hat{\beta}_{\gamma_j} =& \beta_{\gamma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[Y_j(t+k)Y_j(t+k) - 2Y_j(t+k)u_j^T s(t) + s(t)^T u_j u_j^T s(t)\Big]\cdot I_j^{(k)}(t+k)\Big\rangle \\
\Rightarrow \hat{\beta}_{\gamma_j} =& \beta_{\gamma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[Y_j(t+k)Y_j(t+k) - 2Y_j(t+k)u_j^T s(t) \\
& + tr\big(s(t)^T u_j u_j^T s(t)\big)\Big]\cdot I_j^{(k)}(t+k)\Big\rangle \\
=& \beta_{\gamma_0} + \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[Y_j(t+k)Y_j(t+k) - 2Y_j(t+k)\big\langle u_j^T\big\rangle\big\langle s(t)\big\rangle \\
& + tr\big[\big\langle u_j u_j^T\big\rangle\cdot\big\langle s(t)s^T(t)\big\rangle\big]\Big]\cdot\big\langle I_j^{(k)}(t+k)\big\rangle
\end{aligned}
\tag{4.45}
$$

The update equation for $\hat{\beta}_{\gamma_j}$ (4.45) reduces to the following when there are missing observations in Y:

$$
\begin{aligned}
\hat{\beta}_{\gamma_j} =& \beta_{\gamma_0} + \frac{1}{2}\sum_{t\in T_{obs}}^{T}\sum_{k=0}^{K}\Big[Y_j(t+k)Y_j(t+k) - 2Y_j(t+k)\big\langle u_j^T\big\rangle\big\langle s(t)\big\rangle \\
& + tr\big[\big\langle u_j u_j^T\big\rangle\cdot\big\langle s(t)s^T(t)\big\rangle\big]\Big]\cdot\big\langle I_j^{(k)}(t+k)\big\rangle
\end{aligned}
\tag{4.46}
$$

4.3.2.2 Inference of I_j, π_j, and M_j

The posterior of π_j and M_j can be derived as follows according to the general solution of (4.25)

$$
\begin{aligned}
q^*(\pi_j) =& Dir(\{\pi_j^{(0)},\cdots,\pi_j^{(K)}\}|\hat{\alpha}_\pi) \qquad (4.47)\\
& \text{with: } \hat{\alpha}_\pi = \alpha_{\pi_0} + \big\langle I_j(t)\big\rangle \\
q^*(M_j) =& \prod_{k'=0}^{K} Dir\Big(\{M_j^{(0,k')},\cdots,M_j^{(K,k')}\}|\{\hat{\alpha}_{M_j}^{(0,k')},\cdots,\hat{\alpha}_{M_j}^{(K,k')}\}\Big) \\
\text{with: } \hat{\alpha}_{M_j}^{(k,k')} =& \alpha_{M_0}^{(k,k')} + \sum_{t=2}^{T}\big\langle I_j^{(k)}(t)\cdot I_j^{(k')}(t-1)\big\rangle \qquad (4.48)
\end{aligned}
$$

Then the following statistics are calculated and used in (4.51)

$$\langle M_j^{(k,k')} \rangle = \frac{\hat{\alpha}_{M_j}^{(k,k')}}{\sum_{k=0}^{K} \hat{\alpha}_{M_j}^{(k,k')}} \tag{4.49}$$

$$\langle \ln M_j^{(k,k')} \rangle = \psi(\hat{\alpha}_{M_j}^{(k,k')}) - \psi(\sum_{k=0}^{K} \hat{\alpha}_{M_j}^{(k,k')}) \tag{4.50}$$

The optimal solution of the posterior of I_j can be derived as follows:

$$\begin{aligned}
\ln q_{I_j}^* \propto & \mathbb{E}_{q_{\setminus I_j}(Z_{\setminus I_j})}\big[\ln p(X,Y,Z)\big] \\
= & \Big\langle \ln p(X_i|s, I_j, h_i, \sigma_i^2) + \ln p(I_j|\pi_j, M_j) \Big\rangle \\
= & \Big\langle \ln \prod_{t=1}^{T}\prod_{k=0}^{K} p(X_i(t)|h_i^T, s(t+k), I_j^{(k)}(t), \sigma_i^2) \\
& + \ln \prod_{t=2}^{T} p(I_j(t)|I_j(t-1), M_j) + \ln p(I_j(1)|\pi_j) \Big\rangle \\
= & \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K} \ln \Big[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)\Big]^{I_j^{(k)}(t)} \\
& + \sum_{t=2}^{T} \ln \Big[\prod_{k=0}^{K}\prod_{k'=0}^{K} \big(M_j^{(k,k')}\big)^{I_j^{(k)}(t)\cdot I_j^{(k')}(t)}\Big] + \sum_{k=0}^{K} \ln(\pi_j^{(k)})^{I_j^{(k)}(1)} \Big\rangle \\
= & \sum_{t=1}^{T}\sum_{k=0}^{K} I_j^{(k)}(t) \cdot \Big\langle \ln \mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) \Big\rangle \\
& + \sum_{t=2}^{T}\sum_{k=0}^{K}\sum_{k'=0}^{K} I_j^{(k)}(t) \cdot I_j^{(k')}(t) \Big\langle \ln M_j^{(k,k')} \Big\rangle + \sum_{k=0}^{K} I_j^{(k)}(1) \Big\langle \ln \pi_j^{(k)} \Big\rangle
\end{aligned} \tag{4.51}$$

The approximated posterior of I_j has the similar formulation as the complete-data likelihood in Hidden Markov Model (HMM) [134] except the expectation is now taken on the logarithm of the parameters. In order to use the HMM forward-backward algorithm in the reference of I_j, we utilize the corollary 2.2 in [134], which can be explained briefly as follows as an example: for a unknown parameter set that consists of three parameters: $\theta = \{\theta_1, \theta_2, \theta_3\}$ and the natural (logarithm of the) parameter set is: $\phi(\theta) = \{\ln\theta_1, \ln\theta_2, \ln\theta_3\}$. The expected natural parameters set used in approximate posterior is: $\langle\phi(\theta)\rangle = \{\langle\ln\theta_1\rangle, \langle\ln\theta_2\rangle, \langle\ln\theta_3\rangle\}$ and the modified parameter set is: $\tilde{\theta} = \{\exp\langle\ln\theta_1\rangle, \exp\langle\ln\theta_2\rangle, \exp\langle\ln\theta_3\rangle\}$. We can use $\ln\tilde{\theta}$ derived above instead of $\langle\ln\theta\rangle$ since they generate the same logarithm likelihood according

to the Corollary 2.2 in [134]:

$$\langle \ln p(X|\theta) \rangle = \ln \tilde{p}(X|\tilde{\theta}) \tag{4.52}$$

So, the following set of modified parameters in (4.53) are defined since they generate the same logarithm likelihood as the original ones.

$$\begin{aligned}
&\tilde{\mathcal{N}}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) = \exp\Big\langle \ln \mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) \Big\rangle \\
&\tilde{M}_i^{(k,k')} = \exp\Big\langle \ln M_j^{(k,k')} \Big\rangle \\
&\tilde{\pi}_i^{(k)} = \exp\Big\langle \ln \pi_j^{(k)} \Big\rangle
\end{aligned} \tag{4.53}$$

Substituting the modified parameters to (4.51), the solution of I_j becomes

$$\begin{aligned}
\ln q_{I_j}^* \propto & \ln \prod_{t=1}^{T} \prod_{k=0}^{K} \tilde{p}(X_i(t)|h_i^T, s(t+k), I_j^{(k)}(t), \sigma_i^2) \\
& + \ln \prod_{t=2}^{T} \tilde{p}(I_j(t)|I_j(t-1), \tilde{M}_i) + \ln \tilde{p}(I_j(1)|\tilde{\pi}_i) \\
= & \sum_{t=1}^{T} \sum_{k=0}^{K} I_j^{(k)}(t) \ln \tilde{\mathcal{N}}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) \\
& + \sum_{t=2}^{T} \sum_{k=0}^{K} \sum_{k'=0}^{K} I_j^{(k)}(t) I_j^{(k')}(t) \ln \tilde{M}_i^{(k,k')} + \sum_{k=0}^{K} I_j^{(k)}(1) \ln \tilde{\pi}_i^{(k)}
\end{aligned} \tag{4.54}$$

where, $\tilde{p}$ represents the probability density function with the modified parameters. Getting rid of the expectation operators, we can use the forward-backward algorithm of HMM to infer I_j. In order to calculate equation (4.47) and (4.48), the expectation term $\big\langle I_j(t) \big\rangle$ and $\big\langle I_j^{(k)}(t) \cdot I_j^{(k')}(t-1) \big\rangle$ need to be calculated. In forward recursion, the posterior over I_j given the observed sequence up to and including current time t is defined as:

$$\begin{aligned}
\tilde{\alpha}_t(I_j(t)) = & \tilde{p}(I_j(t)|X_i(1:t)) \\
= & \frac{\tilde{p}(X_i(t)|I_j(t), X_i(1:t-1)) \cdot \tilde{p}(I_j(t)|X_i(1:t-1))}{\tilde{p}(X_i(t)|X_i(1:t-1))} \\
= & \frac{1}{\tilde{p}(X_i(t)|X_i(1:t-1))} \sum_{I_j(t-1)} \tilde{p}(X_i(t)|I_j(t)) \cdot \tilde{p}(I_j(t)|I_j(t-1)) \\
& \cdot \tilde{p}(I_j(t-1)|X_i(1:t-1)) \\
= & \frac{1}{\xi(X_i(t))} \Bigg[\sum_{I_j(t-1)} \tilde{\alpha}_{t-1}(I_j(t-1)) \cdot \tilde{p}(I_j(t)|I_j(t-1)) \Bigg] \tilde{p}(X_i(t)|I_j(t))
\end{aligned} \tag{4.55}$$

where, $\xi(X_i(t)) = \tilde{p}(X_i(t)|X_i(1:t-1))$ is the normalization constant. And also, $\tilde{\beta}_t(I_j(t))$ is also defined in the backward recursion, which is carried out from $t = T, \cdots, 1$:

$$\begin{aligned}\tilde{\beta}_t(I_j(t)) =& \tilde{p}(X_i(t+1:T)|I_j(t)) \\ =& \sum_{I_j(t+1)} \tilde{p}(X_i(t+2:T)|I_j(t+1))\tilde{p}(I_j(t+1)|I_j(t))\tilde{p}(X_i(t+1)|I_j(t+1)) \\ =& \sum_{I_j(t+1)} \tilde{\beta}_{t+1}(I_j(t+1))\tilde{p}(I_j(t+1)|I_j(t))\tilde{p}(X_i(t+1)|I_j(t+1)) \end{aligned} \tag{4.56}$$

with the initial condition $\tilde{\beta}_t(I_j(t)) = 1$. The posterior distribution can be written as:

$$\tilde{p}(I_j(t)|X_i(1:T)) \propto \tilde{\alpha}_t(I_j(t)) \cdot \tilde{\beta}_t(I_j(t)) \tag{4.57}$$

and the posterior of the joint distribution is

$$\begin{aligned}&\tilde{p}(I_j(t-1), I_j(t)|X_i(1:T)) \\ \propto& \tilde{\alpha}_{t-1}(I_j(t-1)) \cdot \tilde{p}(I_j(t)|I_j(t-1)) \cdot \tilde{p}(X_i(t)|I_j(t)) \cdot \tilde{\beta}_t(I_j(t)) \end{aligned} \tag{4.58}$$

In above equations, we need to calculate two terms: $\tilde{p}(X_i(t)|h_i^T, s, I_j^{(k)}(t), \sigma_i^2)$ and $\tilde{p}(I_j(t)|I_j(t-1), M_j)$. From corollary 2.2 in [134], we can obtain:

$$\begin{aligned}&\ln \tilde{p}(X_i(t)|h_i^T, s, I_j^{(k)}(t), \sigma_i^2) \\ =& \Big\langle \ln p(X_i(t)|h_i^T, s, I_j^{(k)}(t), \sigma_i^2) \Big\rangle \\ =& \sum_{k=0}^{K} I_j^{(k)}(t) \cdot \Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \end{aligned} \tag{4.59}$$

So,

$$\begin{aligned}&\tilde{p}(X_i(t)|h_i^T, s, I_j^{(k)}(t), \sigma_i^2) \\ =& \exp\Big\{ \sum_{k=0}^{K} I_j^{(k)}(t) \cdot \Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \Big\} \\ =& \prod_{k=0}^{K} I_j^{(k)}(t) \cdot \exp\Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \end{aligned} \tag{4.60}$$

Similarly,

$$\begin{aligned}
&\ln \tilde{p}(I_j(t)|I_j(t-1), M_j) \\
=&\Big\langle \ln p(I_j(t)|I_j(t-1), M_j) \Big\rangle \\
=&\sum_{k=0}^{K}\sum_{k'=0}^{K} I_j^{(k)}(t)\cdot I_j^{(k')}(t)\cdot \Big\langle \ln M_j^{(k,k')} \Big\rangle
\end{aligned} \tag{4.61}$$

And,

$$\begin{aligned}
&\tilde{p}(I_j(t)|I_j(t-1), M_j) \\
=&\exp\Big\{ \sum_{k=0}^{K}\sum_{k'=0}^{K} I_j^{(k)}(t)\cdot I_j^{(k')}(t)\cdot \Big\langle \ln M_j^{(k,k')} \Big\rangle \Big\} \\
=&\prod_{k=0}^{K}\prod_{k'=0}^{K} I_j^{(k)}(t)\cdot I_j^{(k')}(t)\cdot \exp\Big\langle \ln M_j^{(k,k')} \Big\rangle
\end{aligned} \tag{4.62}$$

by substituting (4.60) and (4.62), HMM forward and backward factor $\tilde{\alpha}_t(I_j(t))$ and $\tilde{\beta}_t(I_j(t))$ can be obtained in (4.55) and (4.56). Having that, the posterior distribution of $I_j(t)$ and joint posterior distribution of $\big\{I_j(t), I_j(t-1)\big\}$ are calculated using (4.57) and (4.58), respectively. Then we can calculate the expectation terms: $\big\langle I_j^{(k)}(t)\cdot I_j^{(k')}(t-1)\big\rangle$ and $\big\langle I_j(t)\big\rangle$ that are needed in (4.47) and (4.48):

$$\big\langle I_j^{(k)}(t)\big\rangle = \frac{\tilde{\alpha}_t(I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))}{\sum_{k'=0}^{K}\tilde{\alpha}_t(I_j^{(k')}(t))\tilde{\beta}_t(I_j^{(k')}(t))} \tag{4.63}$$

$$\big\langle I_j^{(k)}(t)\cdot I_j^{(k')}(t-1)\big\rangle = \frac{\tilde{\alpha}_{t-1}(I_j^{(k')}(t-1))\tilde{M}_j^{(k,k')}\tilde{p}(X_i(t)|I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))}{\sum_{k=0}^{K}\sum_{k'=0}^{K}\tilde{\alpha}_{t-1}(I_j^{(k')}(t-1))\tilde{M}_j^{(k,k')}\tilde{p}(X_i(t)|I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))} \tag{4.64}$$

4.3.2.3 Unified Inference of s

According to the general solution, we can write the optimal posterior distribution of latent variable s as:

$$
\begin{aligned}
\ln q_s^* \propto & \mathbb{E}_{q_{\setminus s}(Z_{\setminus s})}\big[\ln p(X,Y,Z)\big] \qquad (4.65)\\
=&\Big\langle \sum_{t=1}^{T}\sum_{i=1}^{m}\ln p(X_i(t)|s(t),H,\Sigma)+\sum_{t=1}^{T}\sum_{j=1}^{l}\sum_{k=0}^{K}\ln p(Y_j(t+k)|s(t),U,\Gamma,I_j)\\
&+\sum_{t=1}^{T}\ln p(s(t)|s(t-1),F)\Big\rangle\\
=&\sum_{t=1}^{T}\sum_{i=1}^{m}-\frac{1}{2}\Big\langle[X_i(t)-h_i^Ts(t)]^T\cdot\frac{1}{\sigma_i^2}\cdot[X_i(t)-h_i^Ts(t)]\Big\rangle\\
&-\frac{1}{2}\sum_{t=1}^{T}\sum_{j=1}^{l}\sum_{k=0}^{K}\Big\langle[Y_j(t+k)-u_j^Ts(t)]^T\cdot\frac{1}{\gamma_j^2}\cdot[Y_j(t+k)-u_j^Ts(t)]\cdot I_j^{(k)}(t-k)\Big\rangle\\
&+\sum_{t-1}^{T}-\frac{1}{2}s^T(t)\Big\langle\frac{1}{1-F^2}\Big\rangle s(t)+s^T(t-1)\Big\langle\frac{F}{1-F^2}\Big\rangle s(t)\\
&-\frac{1}{2}s^T(t-1)\Big\langle\frac{F^2}{1-F^2}\Big\rangle s(t-1)+const\\
=&\sum_{t=1}^{T}\sum_{i=1}^{m}\Big[-\frac{1}{2}s^T(t)\Big\langle h_ih_i^T\frac{1}{\sigma_i^2}\Big\rangle s(t)+s^T(t)\Big\langle h_i\frac{1}{\sigma_i^2}X_i(t)\Big\rangle\Big]\\
&+\sum_{t=1}^{T}\sum_{j=1}^{l}\sum_{k=0}^{K}\Big[-\frac{1}{2}s^T(t)\Big\langle u_ju_j^T\frac{1}{\gamma_j^2}I_j^{(k)}(t-k)\Big\rangle s(t)+s^T(t)\Big\langle u_j\frac{1}{\gamma_j^2}Y_j(t+k)I_j^{(k)}(t-k)\Big\rangle\Big]\\
&+\sum_{t-1}^{T}-\frac{1}{2}s^T(t)\Big\langle\frac{1}{1-F^2}\Big\rangle s(t)+s^T(t-1)\Big\langle\frac{F}{1-F^2}\Big\rangle s(t)\\
&-\frac{1}{2}s^T(t-1)\Big\langle\frac{F^2}{1-F^2}\Big\rangle s(t-1)+const\\
=&\sum_{t=1}^{T}-\frac{1}{2}s^T(t)\Big[\sum_{i=1}^{m}\Big\langle h_ih_i^T\Big\rangle\Big\langle\sigma_i^2\Big\rangle^{-1}+\sum_{j=1}^{l}\sum_{k=0}^{K}\Big\langle u_ju_j^T\Big\rangle\Big\langle\gamma_j^2\Big\rangle^{-1}\Big\langle I_j^{(k)}(t-k)\Big\rangle\Big]s(t)\\
&+s^T(t)\Big[\sum_{i=1}^{m}\Big\langle h_i\Big\rangle\Big\langle\sigma_i^2\Big\rangle^{-1}\Big\langle X_i(t)\Big\rangle+\sum_{j=1}^{l}\sum_{k=0}^{K}\Big\langle u_j\Big\rangle\Big\langle\gamma_j^2\Big\rangle^{-1}\Big\langle Y_j(t+k)\Big\rangle\Big\langle I_j^{(k)}(t-k)\Big\rangle\Big]\\
&+\sum_{t-1}^{T}-\frac{1}{2}s^T(t)\Big\langle\frac{1}{1-F^2}\Big\rangle s(t)+s^T(t-1)\Big\langle\frac{F}{1-F^2}\Big\rangle s(t)\\
&-\frac{1}{2}s^T(t-1)\Big\langle\frac{F^2}{1-F^2}\Big\rangle s(t-1)+const
\end{aligned}
$$

The inference of latent feature s brings up a state estimation problem given observations. If the model parameter is deterministic and the model is linear, Kalman filtering and smoothing solution is optimal [73, 136]. With missing data, the filtering and smoothing processes can adopt the same strategy as IOPSFA in EM algorithm [128]. However, under Bayesian framework, parameters are estimated as posterior distributions with consideration of uncertainties, which introduce the difficulties in calculating the optimal value of s [116]. The original Linear Gaussian State Space Model (LGSSM) in (4.4) is transferred to an augmented LGSSM to which the standard Kalman filtering and smoothing can be applied. In the augmented LGSSM, the optimal solution should have the following form as in (4.66) by introducing fluctuation terms $\tilde{F}_A(t)$ and $\tilde{F}_B(t)$ [137, 138]

$$\begin{aligned}\ln q_s^* = &\sum_{t=1}^{T} -\frac{1}{2} s^T(t)\tilde{F}_A(t)s(t) + \sum_{t=1}^{T} s^T(t)\tilde{F}_B(t) \\ &+ \sum_{t=1}^{T} -\frac{1}{2}[s(t) - \tilde{F}s(t-1)]^T\tilde{\Lambda}^{-1}[s(t) - \tilde{F}s(t-1)] + \text{const}\end{aligned} \tag{4.66}$$

In (4.66), the first two terms correspond to the emission equation and the third term corresponds to the state transition equation. We set parameters $\tilde{\Lambda}, \tilde{F}, \tilde{F}_A(t), \tilde{F}_B(t)$ in (4.67) $\sim$ (4.70)

$$\tilde{\Lambda} = \Big\langle \frac{1}{1-F^2} \Big\rangle^{-1} \tag{4.67}$$

$$\tilde{F} = \Big\langle \frac{F}{1-F^2} \Big\rangle \Big\langle \frac{1}{1-F^2} \Big\rangle^{-1} \tag{4.68}$$

$$\tilde{F}_A(t) = \begin{cases} \sum_{i=1}^{m} \Big\langle h_i h_i^T \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} + \sum_{j=1}^{l}\sum_{k=0}^{K} \Big\langle u_j u_j^T \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1} \Big\langle I_j^{(k)}(t+k) \Big\rangle - \tilde{F}\tilde{\Lambda}^{-1}\tilde{F} + \Big\langle \frac{F^2}{1-F^2} \Big\rangle, t = 1:T-1 \\ \sum_{i=1}^{m} \Big\langle h_i h_i^T \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} + \sum_{j=1}^{l}\sum_{k=0}^{K} \Big\langle u_j u_j^T \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1} \Big\langle I_j^{(k)}(t+k) \Big\rangle \qquad , t = T \end{cases} \tag{4.69}$$

$$\tilde{F}_B(t) = \textstyle\sum_{i=1}^{m} \Big\langle h_i \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} \Big\langle X_i(t) \Big\rangle + \sum_{j=1}^{l}\sum_{k=0}^{K} \Big\langle u_j^T \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1} \Big\langle Y_j(t+k) \Big\rangle \Big\langle I_j^{(k)}(t+k) \Big\rangle, t = K+1:T \tag{4.70}$$

Set $I_j^{(k)}(t+k) = 0$, if $t + k \leq 0$ in (4.69) and (4.70). By substituting the transformed parameters (4.67) $\sim$ (4.70) in (4.66), we can get the optimal solution of the

approximate posterior of s as in (4.65) and the augmented LGSSM can be formulated as

$$\begin{cases} s(t) = \tilde{F}s(t-1) + \tilde{e}_s(t), \tilde{e}_s(t) \sim \mathcal{N}(\mathbf{0}, \tilde{\Lambda}) \\ D(t) = \tilde{H}s(t) + \tilde{e}_d(t), \tilde{e}_d(t) \sim \mathcal{N}(\mathbf{0}, \tilde{\Sigma}) \end{cases} \tag{4.71}$$

where D represents the observations including both inputs and outputs and the standard Kalman filtering and smoothing techniques can be applied.

(1). Filtering step

With the augmented model in (4.71), the distributions for s in different steps are defined in (4.72) to (4.74)

$$p(s(t)|D(1:t), \theta^{old}) = \mathcal{N}(\mu(t), V(t)) \tag{4.72}$$

$$p(s(t)|D(1:t-1), \theta^{old}) = \mathcal{N}(\mu_t^{t-1}, V_t^{t-1}) \tag{4.73}$$

$$p(s(t)|D(1:T), \theta^{old}) = \mathcal{N}(\hat{\mu}(t), \hat{V}(t)) \tag{4.74}$$

where, μ_t^{t-1}, V_t^{t-1} are the mean and variance of the Gaussian distribution in the prediction step (from time $t-1$ to t). $\mu(t)$ and $V(t)$ are the mean and variance of the posterior distribution in filtering step. $\hat{\mu}(t)$ and $\hat{V}(t)$ are mean and variance of the posterior distribution in smoothing step. In order to calculate $\mu(t)$ and $V(t)$, we derive the logarithm posterior distribution and using Bayes rule as follows

$$\begin{aligned} &\ln p(s(t)|D(1:t)) \\ =&\ln p(D(t)|s(t)) + \ln p(s(t)|D(1:t-1)) + \text{const} \\ =&-\frac{1}{2}\Big[D(t) - \tilde{H}s(t)\Big]^T \tilde{\Sigma}^{-1}\Big[D(t) - \tilde{H}s(t)\Big] \\ &-\frac{1}{2}\Big[s(t) - \tilde{F}s(t-1)\Big]^T (V_t^{t-1})^{-1}\Big[s(t) - \tilde{F}s(t-1)\Big] + \text{const} \\ =&-\frac{1}{2}s^T(t)\Big(\tilde{H}^T\tilde{\Sigma}^{-1}\tilde{H} + (V_t^{t-1})^{-1}\Big)s(t) \\ &+ s^T(t)\Big(\tilde{H}^T\tilde{\Sigma}^{-1}D(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\Big) + \text{const} \end{aligned} \tag{4.75}$$

To comply with the format in (4.72), we can get:

$$V(t) = \Big(\tilde{H}^T\tilde{\Sigma}^{-1}\tilde{H} + (V_t^{t-1})^{-1}\Big)^{-1} \tag{4.76}$$

$$\mu(t) = V(t) \cdot \Big(\tilde{H}^T\tilde{\Sigma}^{-1}D(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\Big) \tag{4.77}$$

$\tilde{H}$ and $\tilde{\Sigma}$ have no explicit expression in the augmented system (4.71) and they can be derived by comparing the corresponding terms in (4.66) and (4.75)

$$\tilde{F}_A(t) = \tilde{H}^T \tilde{\Sigma}^{-1} \tilde{H} \tag{4.78}$$

$$\tilde{F}_B(t) = \tilde{H}^T \tilde{\Sigma}^{-1} D(t) \tag{4.79}$$

Substituting $\tilde{F}_A(t)$ and $\tilde{F}_B(t)$ into (4.76) and (4.77) results in:

$$V(t) = \left[\tilde{F}_A(t) + (V_t^{t-1})^{-1}\right]^{-1} \tag{4.80}$$

$$\mu(t) = V(t) \cdot \left[\tilde{F}_B(t) + (V_t^{t-1})^{-1} \tilde{F}\mu(t-1)\right] \tag{4.81}$$

Above two equations constitute the update step in Kalman filtering for the augmented system (4.71), in which all the needed items are calculated in (4.67) $\sim$ (4.70) and the following prediction equations

$$\mu_t^{t-1} = \tilde{F} \cdot \mu(t-1) \tag{4.82}$$

$$V_t^{t-1} = \tilde{F} \cdot V(t-1) \cdot \tilde{F}^T + \tilde{\Lambda} \tag{4.83}$$

(2) Smoothing step

In smoothing step, the standard Kalman smoothing procedure can be adopted

$$\hat{\mu}(t) = \mu(t) + J(t)\left[\hat{\mu}(t+1) - \tilde{F}\mu(t)\right] \tag{4.84}$$

$$\hat{V}(t) = V(t) + J(t)\left[\hat{V}(t+1) - V_{t+1}^t\right] J^T(t) \tag{4.85}$$

$$J(t) = V(t)\tilde{F}^T (V_{t+1}^t)^{-1} \tag{4.86}$$

with initializations:

$$\hat{\mu}(T) = \mu(T) \tag{4.87}$$

$$\hat{V}(T) = V(T) \tag{4.88}$$

In summary, (4.80) $\sim$ (4.88) constitute the complete Kalman filtering and smoothing procedure and the following sufficient statistics can be calculated

$$\langle s(t) \rangle = \hat{\mu}(t) \tag{4.89}$$

$$\langle s(t)s(t) \rangle = \hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t) \tag{4.90}$$

$$\langle s(t+1)s(t) \rangle = J(t)\hat{V}(t+1) + \hat{\mu}(t+1)\hat{\mu}^T(t) \tag{4.91}$$

4.3.2.4 Inference of λ_j: Importance Sampling

Since the prior of each λ_j is of Beta distribution and it is not conjugate to the likelihood of s, which is of Gaussian distribution, the target posterior of λ_j cannot be derived analytically. Importance sampling method is employed to solve this problem [116]. The optimal posterior of λ_j is

$$\begin{aligned}\ln q^*_{\lambda_j} \propto & \mathbb{E}_{q_{\setminus\lambda_j}(Z_{\setminus\lambda_j})}\big[\ln p(X,Y,Z)\big] \\ = & \Big\langle \ln p(s_j|\lambda_j) p(\lambda_j|\alpha_{\lambda_0},\beta_{\lambda_0}) \Big\rangle + const \\ = & \Big\langle \ln p(s_j|\lambda_j) \Big\rangle \cdot p(\lambda_j|\alpha_{\lambda_0},\beta_{\lambda_0}) + const \end{aligned} \tag{4.92}$$

And, the expectation of likelihood function can be derived with initial distribution of $s(1) \sim \mathcal{N}(0,1)$:

$$\begin{aligned} & \Big\langle \ln p(s_j|\lambda_j) \Big\rangle \\ = & \Big\langle \ln \mathcal{N}(0,1) \cdot \sum_{t=2}^{T} \mathcal{N}\big(s_j(t)|\lambda_j s_j(t-1), 1-\lambda_j^2\big) \Big\rangle \\ = & \Big\langle -\frac{1}{2}\ln 2\pi - \frac{1}{2}s^2(1) - \frac{1}{2}\sum_{t=2}^{T}\big[\ln 2\pi + \ln(1-\lambda_j^2)\big] + \big[s_j^2(t) - \lambda_j s_j^2(t-1)\big]^2 \cdot \frac{1}{1-\lambda_j^2} \Big\rangle \\ = & -\frac{T}{2}\ln\pi - \frac{T-1}{2}\ln(1-\lambda_j^2) - \frac{1}{2}\Big\langle s^2(1) \Big\rangle \\ & -\frac{1}{2}\Big\langle \sum_{t=2}^{T} s_j^2(t) \Big\rangle \frac{1}{1-\lambda_j^2} + \Big\langle \sum_{t=2}^{T} s_j(t-1)s_j(t) \Big\rangle \frac{\lambda_j}{1-\lambda_j^2} - \frac{1}{2}\Big\langle \sum_{t=1}^{T-1} s_j^2(t) \Big\rangle \frac{\lambda_j^2}{1-\lambda_j^2} \end{aligned} \tag{4.93}$$

In importance sampling method, let's assume the target distribution is $p(x)$ and the sampling distribution is $q(x)$, then the sample weights can be calculated as $w(x) = \frac{p(x)}{q(x)}$. In this model, Beta distribution $Beta(\alpha_{\lambda_0}, \beta_{\lambda_0})$ is chosen as the sampling distribution to generate sample weights. The weights of each sample can be calculated as the likelihood in (4.93). Then the three needed statistics of the posterior of

λ_j in (4.67)$\sim$(4.69) can be derived using the value of each sample:

$$\left\langle \frac{1}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{1}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{4.94}$$

$$\left\langle \frac{\lambda_j}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{\lambda_j^{(n)}}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{4.95}$$

$$\left\langle \frac{\lambda_j^2}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{\left(\lambda_j^{(n)}\right)^2}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{4.96}$$

where $\lambda_j^{(n)}$ is the n-th sample drawn from sampling distribution and N is the total number of samples. When $N \to \infty$, the expectation values in (4.94) $\sim$(4.96) will approximate the corresponding statistics of the optimal posterior in (4.92).

4.3.3 On-line Prediction Using the Model

For on-line implementation, only the past output measurements are available for the prediction of future outputs. If the target output has missing values and only partial measurements are available, e.g. in the case of lab samples, we can only use the available samples in the filtering step to obtain the latent feature s. Take one-dimensional output as an example, to predict $y(t)$, one step ahead prediction of $s(t)$ needs to be performed according to the Kalman filter recursions. The predicted $\hat{s}(t)$ can be calculated using the results from the Kalman filtering step:

$$\hat{s}(t) = \mu(t) = V(t) \cdot \left[\tilde{F}_B(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\right] \tag{4.97}$$

Since $y(t)$ and all future y are not available, only the first term of $\tilde{F}_B$ in (4.70) can be calculated. After calculating $\hat{s}(t)$, the prediction of output $\hat{y(t)}$ can be estimated as follows

$$\hat{y}(t) = \sum_{k=0}^{K} U\hat{s}(t-k) \cdot I^{(k)}(t) + e_y(t) \tag{4.98}$$

Given that $e_y(t)$ has zero mean, $y(t)$ is evaluated as:

$$y(t) = \text{mean}\{\hat{y}(t)\} = \sum_{k=0}^{K} U\hat{s}(t-k) \cdot I^{(k)}(t) \tag{4.99}$$

If $y(t)$ is available, we can use it to update previous and current latent features, i.e. $\{s(t-K), \cdots, s(t-1), s(t)\}$. Because in the feature prediction equation (4.97), the term $\tilde{F}_B$ includes information from $y(t)$ to $y(t+K)$, for any sample available in $\{y(t), \cdots y(t+K)\}$, we need to update the current and previous K samples of s. For example, $y(t)$ is available at time t, the update equations of $\{s(t-K), \cdots, s(t)\}$ are as follows

$$
\begin{aligned}
&[s(t)]_0 = [\mu(t)]_0 \\
&= V(t) \cdot \left[\tilde{F}_B(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\right], \\
&y(t) \text{ in } \tilde{F}_B(t) \text{ is available,} \qquad (4.100) \\
&[s(t-1)]_1 = [\mu(t-1)]_0 \\
&= V(t-1) \cdot \left[\tilde{F}_B(t-1) + (V_{t-1}^{t-2})^{-1}\tilde{F}\mu(t-2)\right], \\
&y(t) \text{ in } \tilde{F}_B(t-1) \text{ is available,} \qquad (4.101) \\
&\cdots \\
&[s(t-K)]_K = [\mu(t-K)]_0 \\
&= V(t-K) \cdot \left[\tilde{F}_B(t-K) + (V_{t-K}^{t-K-1})^{-1}\tilde{F}\mu(t-K-1)\right], \\
&y(t) \text{ in } \tilde{F}_B(t-K) \text{ is available.} \qquad (4.102)
\end{aligned}
$$

The subscript i in $[s(t)]_i$ is the update times of $s(t)$. If there is no update of $s(t)$, then $[s(t)]_i$=$[s(t)]_{i-1}$. Also, it is worth noticing that, the prediction of $y(t)$ in (4.98) always uses the most updated $\{\hat{s}(k), 0 \leq k \leq K\}$. In this way, we can utilize the information of all available y to infer latent feature s to make it as accurate as possible.

4.4 Applications

In this section, the prediction ability of the proposed method is demonstrated with two simulations. First, a numerical case is utilized to illustrate the prediction ability in two scenarios: output with no missing data and with multi-rate samples. Second, application to a simulated CSTR process is conducted to demonstrate the prediction ability in the presence of a slow-sampled quality variable. In both simulation studies, the proposed method is compared with the IOPSFA algorithm to demonstrate the performance improvement by considering the time delay, and it is also compared

with the case that only considers the fixed time delay to illustrate the benefits by considering the time-varying time delay.

4.4.1 Numerical Case Study

In the numerical case, a linear state space model is considered as shown in (4.103)

$$\begin{cases} s(t) = \begin{bmatrix} 0.995 & 0 \\ 0 & 0.85 \end{bmatrix} s(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(0, \begin{bmatrix} 1-0.995^2 & 0 \\ 0 & 1-0.85^2 \end{bmatrix}) \\ X(t) = \begin{bmatrix} 1.5 & 3 \\ 0.5 & -0.5 \\ -0.3 & -1 \end{bmatrix} s(t) + e_x(t), e_x(t) \sim \mathcal{N}(0, \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.06 \end{bmatrix}) \\ Y(t) = \begin{bmatrix} 2 & -1 \end{bmatrix} s(t) + e_y(t), e_y(t) \sim \mathcal{N}(0, 0.05) \end{cases} \tag{4.103}$$

In this case, we use the one-dimension output as an example. As mentioned before, if more than one quality variable needs to be inferred, they can be decomposed into multiple one-dimensional output as shown in (4.103). We assume that the maximum time delay $K = 4$, so the time delay transition matrix M is constructed by a 5×5 matrix in (4.104)

$$M = \begin{bmatrix} 0.95 & 0.01 & 0.01 & 0.01 & 0.02 \\ 0.01 & 0.95 & 0.01 & 0.02 & 0.01 \\ 0.01 & 0.02 & 0.90 & 0.03 & 0.04 \\ 0.01 & 0.01 & 0.01 & 0.95 & 0.02 \\ 0.01 & 0.02 & 0.01 & 0.01 & 0.95 \end{bmatrix} \tag{4.104}$$

in which, the diagonal elements are larger than the other elements in the same row, since in reality, most processes are continuous and steady and the time delay values are not expected to change too frequently owing to the inertia of the process. To generate data, first, the two-dimensional latent features are generated according to the given λ: $\lambda_1 = 0.995$ and $\lambda_2 = 0.85$ in Figure 4.6. Then input and output data are generated according to the emission equations in Figure 4.7. In order to generate the delayed outputs, time delay sequence is generated using the Markov transition matrix M as shown in Figure 4.8. At last, the delayed output data can be determined by shifting each sample with the values according to the generated delay sequence. The generated data consisting of 10,000 samples. The first 5000 samples are used for training and the last 5000 samples for validation.

Next, we provide the details of the modeling process and illustrate the performance through two cases: 1) no missing output; 2) multi-rate case.

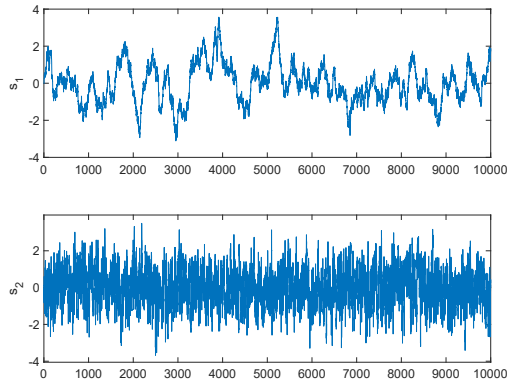

Figure 4.6: Simulated slow features s

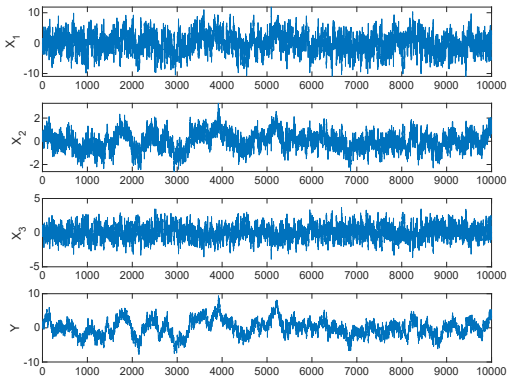

Figure 4.7: Simulated inputs X and ouyput Y

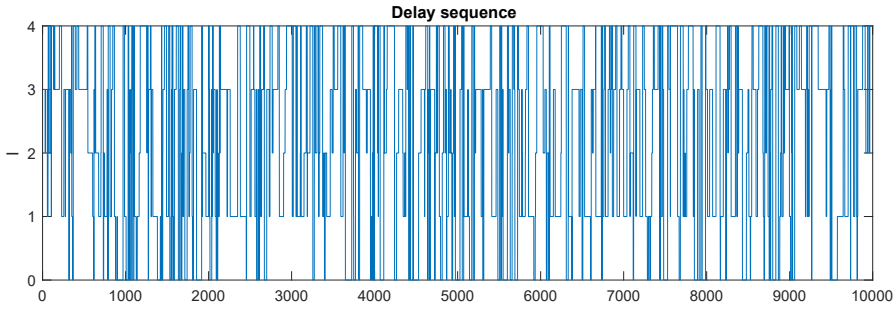

Figure 4.8: Simulated delay sequence I

Case 1: No missing output

In this case, we assume all output observations are available and slow features are extracted from non-missing observations of both inputs and outputs. This is corresponding to the scenario that an accurate on-line analyzer is installed in the plant to measure the quality variable, e.g. VX Meter installed to measure water content in Steam-assisted gravity drainage (SAGD) process [128]. The developed soft sensor model will be useful when the on-line analyzer is out of service, i.e. damaged, under maintenance or becoming inaccurate due to long time service or harsh operation conditions, etc. Mean Absolute Error (MAE) is used to compare the difference between the predicted output $\hat{Y}$ and observed output Y

$$MAE = \text{mean}(|\hat{Y} - Y|) \tag{4.105}$$

To illustrate the performance of the developed model, first, we compare the proposed method with IOPSFA to demonstrate that IOPSFA_VTD can better extract the SFs from data and the performance is improved when time delay in Y is considered. Figure 4.9 shows the extracted SFs by IOPSFA_VTD and IOPSFA. The blue dashed line in each sub-figure is the real SF that we have generated through the simulated model in (4.103). The MAE for the extracted SFs comparing with real SFs is calculated and labeled in the corresponding sub-figure. As we can see, the MAE for the SFs extracted by IOPSFA_VTD is smaller than SFs extracted by IOPSFA. It means SFs extracted by IOPSFA_VTD is closer to the real SFs.

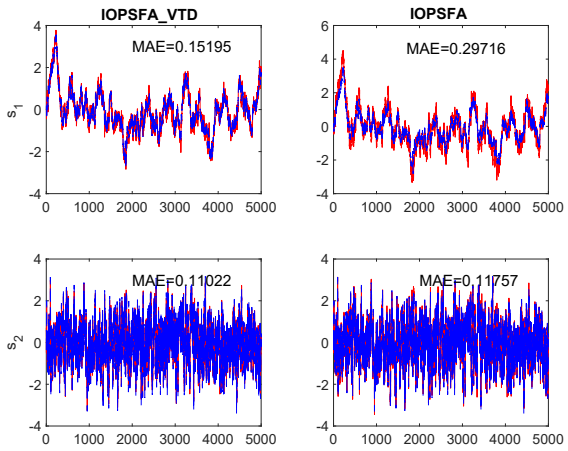

Figure 4.9: Comparison of SFs extracted by IOPSFA_VTD and IOSPFA

The model learned by IOPSFA_VTD is as follows

$$
\begin{aligned}
F &= \begin{bmatrix} 0.9702 & 0 \\ 0 & 0.8492 \end{bmatrix}, H = \begin{bmatrix} 1.3977 & 3.1012 \\ 0.3814 & -0.4252 \\ -0.3015 & -1.0057 \end{bmatrix} \\
U &= \begin{bmatrix} 1.6382 & -0.8207 \end{bmatrix} \\
\Sigma &= \begin{bmatrix} 0.2118 & 0 & 0 \\ 0 & 0.0548 & 0 \\ 0 & 0 & 0.0601 \end{bmatrix}, \Gamma = 0.0341
\end{aligned}
\tag{4.106}
$$

Next, with the above model, we compare prediction performance of the IOPSFA_VTD and IOSFPA in the fixed time delay cases to demonstrate the performance improvement when considering time-varying time delay. A part of prediction results of output is shown in Figure 4.10. The blue dashed lines represent the actual measurements

and the red lines are the predicted values. The first sub-figure is the prediction results for IOPSFA_VTD, the second one is for IOSPFA and the rest of the sub-figures correspond to the fixed time delay cases, i.e. time delay is fixed as 1, 2, 3 and 4, respectively. MAE is calculated for the prediction of each method and summarized in Table 4.1.

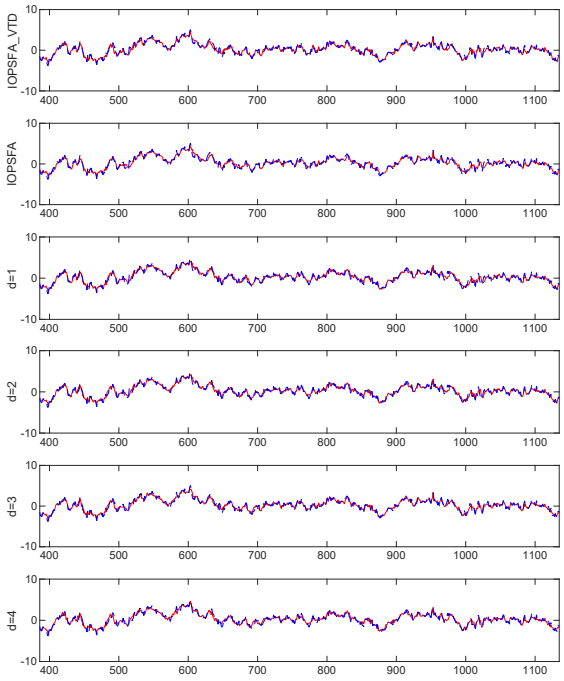

Figure 4.10: Prediction trends without missing outputs

Table 4.1: Prediction results without missing outputs

	IOPSFA_VTD	IOPSFA	delay=1	delay=2	delay=3	delay=4
MAE	0.3588	0.4059	0.3793	0.3670	0.3636	0.3719

The above results show that IOPSFA_VTD approach produces the smallest MAE comparing with other methods. So, considering fixed time delay is still better than not considering any time delay, i.e. IOPSFA approach. Also, when the time delay is fixed as 1 or 4, the performance is worse than the case that time delay is 2 or 3. To understand this, we can calculate the average time delay that is around 2.2 in this simulation. This is consistent with the above results in the sense that, the closer to the average time delay, the smaller MAE it can produce.

Case 2: Multi-rate case

In industries, we are often short of ways to measure quality variables accurately and timely like other regular process variables. For example, even if we have the option to install an on-line analyzer like VX Meter in oil sands extraction process, as we mentioned before, installing and then maintaining it are expensive. So a commonly adopted approach is to sample the quality variable manually and analyze the samples in the lab periodically. The collected samples, therefore, have larger sampling intervals than other process variables and it imposes challenges in building a predictive model. In this case, we adopt the similar idea to handle the multi-rate problem as IOPSFA [128]. In this case study, we down-sample the output to simulate the multi-rate scenario. The down-sampling ratio is 20, that is to say, one sample is kept for every 20 samples. IOPSFA_VTD can provide on-line prediction when the output is not available and it can also update the extracted slow features whenever the output sample is available. The estimated model parameters are as follows

$$
\begin{aligned}
F &= \begin{bmatrix} 0.9901 & 0 \\ 0 & 0.8790 \end{bmatrix}, H = \begin{bmatrix} 1.5856 & 3.5273 \\ 1.1984 & -0.6567 \\ -0.2422 & -1.1915 \end{bmatrix} \\
U &= \begin{bmatrix} 2.1270 & -0.5560 \end{bmatrix} \\
\Sigma &= \begin{bmatrix} 0.2497 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.0572 \end{bmatrix}, \Gamma = 0.0342
\end{aligned}
\tag{4.107}
$$

And the sample prediction trends for IOPSFA_VTD and trends from its comparing methods are shown in Figure 4.11

The prediction performance indexed in terms of MAE is summarized in Table 4.2.

Table 4.2: Prediction results with missing outputs, down-sample rate=20

	IOPSFA_VTD	IOPSFA	delay=1	delay=2	delay=3	delay=4
MAE	0.3751	0.5068	0.4434	0.4140	0.3872	0.3977

The predicted trends in Figure 4.11 show that all approaches can catch the trend of real output. However, IOPSFA_VTD has the best results among all methods and the results from the fixed delay cases achieve better performance than IOPSFA. The latter does not consider the time delay. We can also notice that the MAE in multi-rate

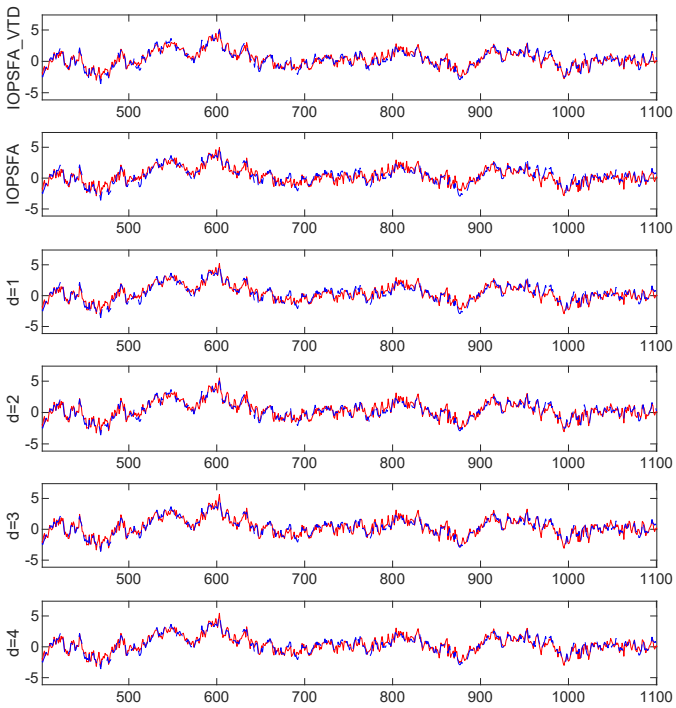

Figure 4.11: Prediction trends with missing outputs, down-sample rate=20

cases is larger than the corresponding no-missing-output cases. This is because in the training phase, the output is not always available. Fewer output samples contain less information, which leads to less accuracy in predicting the output.

4.4.2 Continuously Stirred Tank Reactor

In this section, we use a simulated chemical process to further validate the efficacy of the proposed IOPSFA_VTD algorithm. The CSTR is widely used in many industries, especially in chemical industries. In a single-phase CSTR, chemical reaction takes place and converts component A, which is not desired in the downstream process or not appropriate to discharge to the environment, to component B, which is desired or safe to discharge. A CSTR is normally equipped with an agitator driven by a motor to mix the components in the reactor. A typical CSTR diagram is shown in Figure 4.12. The feed flow is mostly composed of undesired component A and the product contains much less component A. F_A, T_A, C_{Ai} are the flow rate, temperature and concentration of component A into the reactor, respectively. T_J and T_R are cooling jacket temperature and reactor temperature, respectively. C_{Ao} is the concentration

of component A in the product. The feed concentration is around 0.8 and we want to reduce it as much as possible in the product through the reaction. In this case study, F_A, T_A, C_{Ai}, T_J and T_R are selected as input variables and the sampling time is 1. The product concentration C_{Ao} is the quality variable of interest. We assume C_{Ao} cannot be measured on-line, but only be sampled in a larger interval, and in this case, we first generate the output with the same sampling interval as other process variables, then down-sample it with the down-sample ratio = 20.

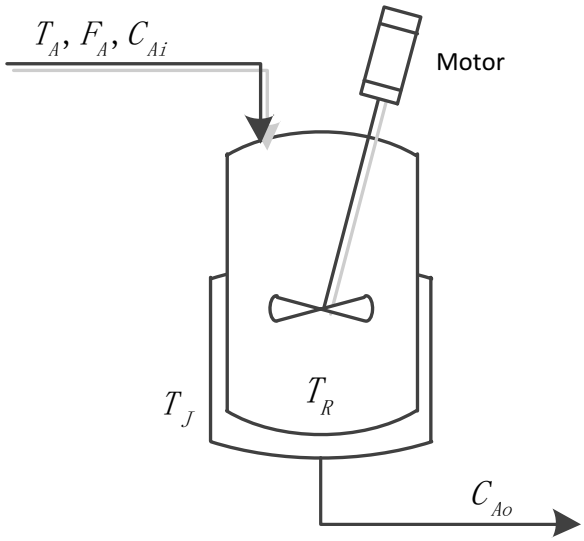

Figure 4.12: CSTR Diagram

The maximum delay time is selected as 3 and total of 7000 samples are collected. The first 5000 samples are used for training and the rest 2000 samples are used for validation. The SFs extracted from IOPSFA_VTD and IOPSFA are compared in Figure 4.13. The slowness of SFs extracted is indicated by the value of λ on top of each sub-figure. The number of SFs, namely q that is used to build the regression model is selected based on their slowness and trends as that selected in IOPSFA approach [128]. In this case, apparently, the fifth feature of each approach mostly contains noise and the slowness measure has a sudden drop comparing to the first four features. So we select $q = 4$ for both approaches. After plotting the extracted features, the same feature number selection criterion is applied to other cases, i.e. fixed delay =1,2,3 and 4. In all cases, we select the first four features as regressors.

The prediction results for all scenarios are shown in Figure 4.14. The blue dashed lines are the real outputs as references and the red lines are the predicted values. The

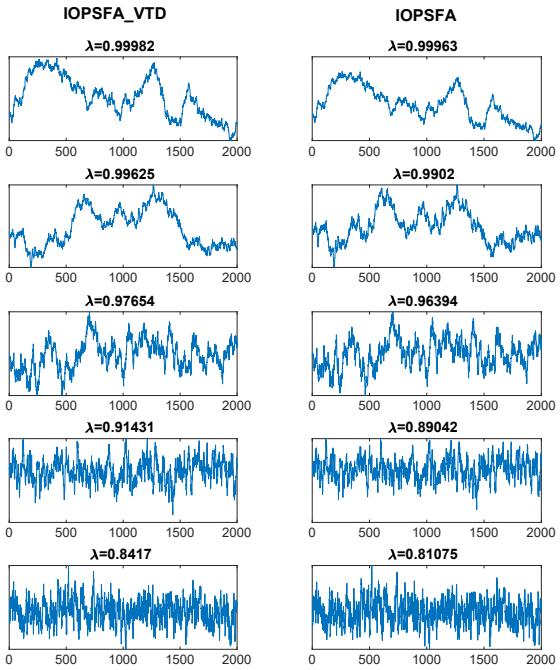

Figure 4.13: CSTR: Comparison of SFs extracted by IOPSFA VTD and IOSPFA

performance indices and the performance improvement of IOPSFA_VTD comparing with other algorithms in terms of MAE is shown in Table 4.3.

Table 4.3: CSTR: Prediction results with missing outputs, down-sample rate=20

	IOPSFA_VTD	IOPSFA	delay=1	delay=2	delay=3
MAE	0.0045	0.0049	0.0059	0.0047	0.0071
Improvement(%)	-	8.16%	23.73%	4.26%	36.62%

The outputs shown in blue dashed lines are fast rate samples and MAE is also calculated using these fast rate samples. From the above results, we can tell that IOPSFA_VTD outperforms other algorithms since it considers time-varying delays in its modeling while the algorithm with the fixed delay = 3 gives the worst performance.

4.5 Conclusions

In this chapter, an enhanced approach based on IOPSFA, termed as IOPSFA_VTD, is proposed by considering the time-varying time delay, under the variational Bayesian framework. Process knowledge can be incorporated as prior distributions of model parameters. The proposed algorithm has the ability to address the time-varying

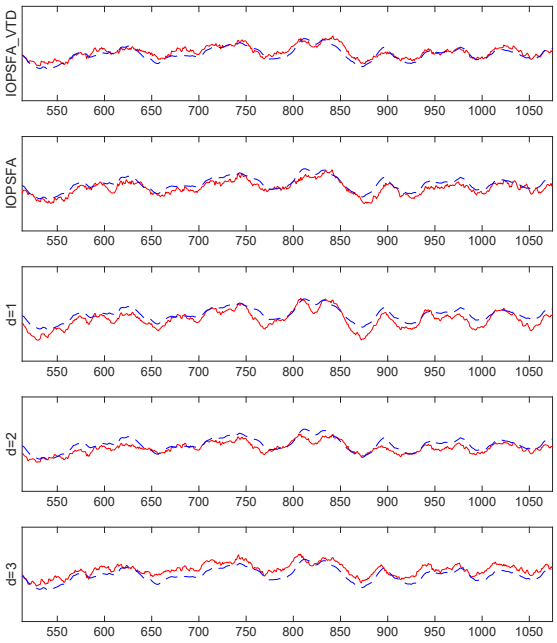

Figure 4.14: CSTR: Prediction trends with missing outputs, down-sample rate=20

time delay which is common in industrial processes. It can extract dynamic latent features using both inputs and the delayed outputs. The output with larger sampling interval can be effectively utilized in extracting latent features. The extracted latent features have improved ability in predicting the desired key variable and the results are validated through a simulated numerical example along with a CSTR example. From the simulation results, it is observed that without considering time delay, the quality variables cannot be estimated as accurately as the proposed method that has considered time delay appropriately.

Chapter 5

Dynamic Latent Variable Modeling with Input Time-varying Time Delays and Application to SAGD Process *

Existence of uncertain time delay is a challenging problem in system identification. Accurate estimates of input-output delay can significantly improve the accuracy of parameter estimation. However, time-varying time delays often exist in the measurements of industrial processes and different measurements often possess different time delays in reference to a target quality variable due to various reasons such as distributed locations of sensors. To address this problem, a probabilistic inferential model is developed under variational Bayesian framework with consideration that different process variables have different time delays. Each delay sequence is properly estimated and utilized in the prediction of the target quality variable. The efficacy of the proposed method is demonstrated through a numerical example and an industrial application.

5.1 Introduction

In modern process industries, successful implementation of advanced control technologies and process monitoring techniques, especially for key quality variables, heavily rely on timely on-line measurements. Sometimes, development or installations of

*Part of this chapter will be submitted as: Fan L, Huang B. Dynamic Latent Variable Modeling with Input Time-varying Time Delays and Application to SAGD Process.

physical measuring instruments are impossible due to inadequacy of measurement techniques, harsh environments or economic infeasibility. Thus, on-line acquisition of these data is difficult if not impossible. One way to solve this problem is to develop inferential sensors, also called soft sensors, which usually take available process variables as inputs to estimate key quality variables that are not possible or very difficult to measure by physical sensors in real-time. Soft sensor has many advantages such as: (i) cost-effectiveness, (ii) easy implementation, and (iii) providing insights into the process [1]. The process to build a soft sensor is equivalent to building a model between real-time process variables and key quality variables on the basis of their correlations. Generally, a model can be built based on process mechanisms, driven by process data or their combinations, i.e. first principles model, black box model or grey box model, respectively. Models with accurate process mechanisms can give more insight of the process and has a wider range of validity, but usually difficult to build due to inadequate process knowledge. For the processes that are complex and have limited process knowledge available, grey box modeling approaches are often the common choices.

Bayesian methods provide a natural way to combine prior information with data, hence a natural choice for grey box modeling. When new data is available, the prior information can be updated based on new process scenarios. Thus, Bayesian approach is a powerful tool for modeling of the process. The other major hurdle in developing models based on process data is its high dimensionality. One might need to develop high dimensional models to describe the behaviour of the data. Development of such models can be challenging. The historical data ofter suffers from information redundancy since many highly correlated process variables have the same variation patterns as they may originate from the same source. In such scenarios, in order to develop models in a more efficient manner, development of lower dimensional latent variable models is a more practical alternative. A latent variable is a variable that is not directly observed from process but inferred from other variables that can be observed directly. The model that relates the set of observations to the latent variables is called latent variable model (LVM). Employment of latent variable models often results in dimensionality reduction since the dimension of latent variables is much less than raw data dimension. Thus how to extract the latent information becomes

a popular research subject. Various methods have been developed in the literature to address the issue of dimensionality of the process data based on latent variable models.

The success of a learning algorithm relies heavily on the choice of features [35]. Typical latent variable models include Principal Component Analysis (PCA) [36,48], Partial Least Square (PLS) [38–40], Independent Component Analysis (ICA) [41,42], and Slow Feature Analysis (SFA) [43] etc. All these methods project higher dimensional original data space to lower dimensional latent space. Each technique extracts different data representations to capture information from explanatory factors hidden in the data. PCA extracts features that have the maximum variances. It uses an orthogonal transformation to transfer a set of correlated variables into a set of linearly uncorrelated variables i.e. principal components [37]. PLS extracts features that have maximum covariance between inputs and outputs space [38,54]. Comparing with PCA approach, PLS incorporates information not only from inputs but also from outputs. ICA is used to separate a multivariate signal into several sources, which are assumed to follow non-Gaussian distribution and to be statistically independent, or as independent as possible [59]. SFA extracts features that vary slowly. It not only reduces the dimensionality of input signal, but also removes the noisy components in the signals which are uninformative for identification.

These algorithms are extended in a probabilistic sense to account for various noise distributions [44–47]. In Probabilistic PCA (PPCA), the principal axes are the maximum likelihood estimates of the parameters, which can be calculated by eigendecomposition. By specifying proper prior distribution in the noise terms, PPCA can also be robust to data irregularities [51–53]. Based on PLS Regression (PLSR) [39] and PPCA, a generative form of the Probabilistic PLSR (PPLSR) model was proposed by [45,57]. It provides a probabilistic view of the traditional PLSR model and explains the relationship among input, output and latent variables. Several Probabilistic ICA (PICA) methods were developed to take advantage of the statistical properties of the signals. PICA assumes a small number of independent components with a residual term that is modeled as Gaussian noise [61]. In PICA, when the number of sources is less than the number of sensors, it leads to the so-called non-square mixing, where the 'extra' sensor observations are explained as observation

noise. The probabilistic SFA (PSFA) describes the process dynamics in the latent space instead of observations by assigning temporal correlation to the extracted features. Then the dynamics can be propagated to the observations from their common causes, i.e. slow features, through the emission matrix between latent features and observations [20, 47]. However, comparing with PLS and PPLS, PSFA can only extract latent features from input variables. To utilize the output observations, a semi-supervised learning method, Input-Output PSFA (IOPSFA) [128] is able to extract dynamic latent features using both input and output information and can also deal with the missing data problem.

A major challenge in developing a LVM is time delay. A wrong time delay results in use of observations at wrong time instance to extract inaccurate latent features. Often, time delay has significant influence on system identification and time delay estimation has been one of the most active research topics. A lot of work has been conducted considering constant time delays [24, 34, 139, 140] and many methods have been proposed, e.g. prediction error methods [34], impulse response methods [24] and adaptive methods, etc. However, time delay can be time-varying. In chemical processes, for example, time delay is often caused by the transportation time of the material in the process. The material transportation rate can change due to the change of working conditions and material components, etc. This makes the time delay estimation problem more challenging. Many approaches have been proposed to address this problem, e.g. using adaptive filtering [141, 142] and quadratic convex approach [143, 144], where delays were considered to vary between some known lower and upper bounds. Considering the uncertainties of time delay, certain distributions can be assumed to account for the varying time delays e.g. uniform distribution [87], multinomial distribution [8], etc. A Bayesian method based on IOPSFA, termed as IOPSFA_VTD is proposed to infer time delay sequence that follows a Categorical distribution. Furthermore, occurrence of time delay at each time instance may not be random and may follow a certain dynamic stochastic pattern. To describe the dynamics of time delay sequence a stochastic correlation model such as hidden Markov model (HMM) is often utilized [26, 30, 34].

Time delay is often distributed. For example, there are many sensors installed across a chemical plant. Operation units are connected by a network of pipelines.

This results in different time spent for the materials to be transported to the locations where the quality variables are sampled, i.e. different time delays from different process variables to the quality variables. In IOPSFA_VTD, only the output variable is considered to have time-varying delays which is equivalent to assuming all input variables have the same time delay in reference to the output. The proposed method in this work, IOPSFA_VTD algorithm is generalized by considering different time-varying delays for different input variables, which is more common in the industrial processes due to the distributed locations of sensors. In addition, in this study, the modeling process is conducted under the variational Bayesian framework so that proper distribution can be utilized to model the uncertainties in unknown parameters and latent variables. As a multi-dimensional problem, the time delay indicator for each input variable is described by a separate Markov chain and probability transition matrix. As a result, multiple time delay sequences are to be identified. The latent dynamic features extracted using IOPSFA_InVTD are expected to have better prediction ability for quality variables than IOSPFA_VTD as it considers more practical scenario.

The remainder of this chapter is organized as follows. In the next section, fundamentals of SFA and PSFA are briefly reviewed. A detailed formulation and variational inferential procedure of the proposed input time-varying time delay problem is presented in section 5.2. Section 5.3 gives a numerical example and an industrial application study to validate the proposed method. The conclusion is drawn in the final section.

5.2 Modeling and Variational Inference of IOPSFA with Input Time-varying Time Delays

5.2.1 Mathematical Formulation

We consider the time delays between different input variables and the quality variable to be different and also time-varying in this work. The input time-varying time delay

problem based on IOPSFA (IOPSFA_InVTD) can be formulated as below:

$$\begin{cases} s(t) = Fs(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(\mathbf{0}, \Lambda) \\ \tilde{X}(t) = Hs(t) + e_x(t), e_x(t) \sim \mathcal{N}(\mathbf{0}, \Sigma) \\ Y(t) = Us(t) + e_y(t), e_y(t) \sim \mathcal{N}(\mathbf{0}, \Gamma) \end{cases} \tag{5.1}$$

where, $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ are measurement noise covariance matrices for inputs and outputs, respectively. The diagonal structure of Σ and Γ shows the independence of measurement noises among different input and output variables. The dimensions of feature space $\mathcal{S}$, inputs space $\mathcal{X}$ and outputs space $\mathcal{Y}$ are q, m and l, respectively. Similar to PSFA, the diagonal components of F: $\{\lambda_j, 1 \leq j \leq q\}$ control the varying speed of each slow features. $\Lambda = \text{diag}\left\{1 - \lambda_1^2, \cdots, 1 - \lambda_q^2\right\}$ is states covariance matrix. We can derive that:

$$F^T F + \Lambda = I_q \tag{5.2}$$

where, I_q is the identity matrix. This constraint imposes the unit variance property on slow feature s(t). s is a latent variable, which connects the input $\tilde{X}$ and output Y. We assume there is no delay between s and Y and the delays exist between s and $\tilde{X}$. Based on the above assumptions, in the formulation (5.1), $\tilde{X}(t)$ is the measurement inputs reconstructed from the raw input measurements: $X(t-K), \cdots, X(t)$. K is the maximum possible time delay for all dimensions of $X(t)$. Then $\tilde{X}(t)$ can be reconstructed as follows:

$$\tilde{X}(t) = [X_1(t-k_1), \cdots, X_i(t-k_i), \cdots, X_m(t-k_m)], \quad 1 \leq i \leq m \tag{5.3}$$

where, X_i represents i-th dimension of X and $k_i \in \{0, 1, 2, \cdots, K\}$ is the time delay for X_i in reference to output Y.

As we assume there is no time delay between each dimension of Y and s, the time delays only exist between Y and each dimension of X. More often, Y represents the quality variables and it normally has one dimension. In the event that Y has more than one dimension and time delays exist in each dimension, we can decompose Y into multiple single outputs and then build multiple models, one for each dimension of Y.

5.2.1.1 Time Delay Indicator I

With the reconstructed $\tilde{X}(t)$ using the delayed raw measurements in (5.3), slow features $s(t)$ can be extracted from both $\tilde{X}(t)$ and $Y(t)$ following the IOPSFA procedure [128]. To better indicate the time delay between $\tilde{X}(t)$ and $X(t)$, an indicator variable $I \in \mathbb{R}^{m\times(K+1)\times T}$ is introduced, where T is the total number of samples. At time t, $I(t) = [I_1(t)^T, I_2(t)^T, \cdots, I_m(t)^T]^T$ and each row of $I(t)$ is: $I_i(t) = [I_i(t)^{(0)}, I_i(t)^{(1)}, \cdots, I_i(t)^{(K)}], 1 \leq i \leq m$. The structure of $I(t)$ is represented in Figure 5.1: $I(t)$ has the following property:

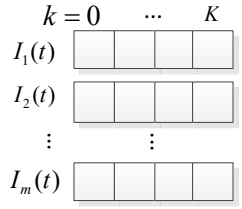

Figure 5.1: Graphical structure of indicator variable $I(t)$

$$\forall, i \in \{1, \cdots, m\}, \sum_{k=0}^{K} I_i^{(k)}(t) = 1, I_i^{(k)}(t) \in \{0, 1\} \tag{5.4}$$

The prior of the initial time delay indicator $I_i(1)$ is represented as π_i:

$$\pi_i = \{\pi_i^{(k)}\} : \pi_i^{(k)} = p(I_i^{(k)}(1) = 1) \tag{5.5}$$

From the definition and property of $I(t)$, only one component of $I_i(t)$ can take value 1, e.g. $I_i^{(k)}(t) = 1$ indicates that k is the time delay between X_i and $\tilde{X}_i/s/Y$. It means $X_i(t)$ will take k sampling time to impact the output, i.e. $Y(t+k)$:

$$X_i(t) \xrightarrow[I_i^{(k)}(t)=1]{\text{delay}=k} Y(t+k) \tag{5.6}$$

Since the time delay for each process variable varies along with time and practically, time delay could increase, decrease or stay the same value as the previous time instant. For example, at time t, the time delay for a process variable is d, and at time $t+1$, time delay could stay at value d in a good chance the process is continuous and abrupt changes are not expected. There are some chances that time delay increase to $d+1$ or decrease to $d-1$ due to process disturbances, operating condition changes

or measurement errors. Assuming the time delay can only increase or decease by 1 at the current time instance relative to the last time instance, then $I_i(t)$ may be described by a markovian chain where a transition matrix M_i can be defined for i-th process variable to describe the Markov behaviour. The elements of M_i represent the transition probability from one time delay value to the next:

$$M_i^{(d_{t-1},d_t)} = p(I_i^{(d_t)}(t) = 1 | I_i^{(d_{t-1})}(t-1) = 1) \tag{5.7}$$

where, d_t and d_{t-1} represent the time delay at time t and $t-1$, respectively. For example, $M_i^{(1,2)}$ represents the probability that time delay increases from 1 at time $t-1$ to 2 at time time t. Transition matrix M_i may be known as a priori by incorporating process knowledges or otherwise its elements can be estimated as unknown parameters in the proposed algorithm. The elements in M_i are constrained by following relationship:

$$\sum_{d_t=0}^{K} M_i^{(d_{t-1},d_t)} = 1 \tag{5.8}$$

which means each row vector of M_j has the sum of 1.

5.2.1.2 Probabilistic Graphical Model

The probabilistic graphic model is presented in Figure 5.2. The grey circles represent the observations and white ones represent unknown variables or parameters that need to be estimated. Rectangles represent the known hyper-parameters for unknown variables or parameters. Above each reconstructed observation $\tilde{X}(t)$, the number $1 \cdots m$ represent the m-dimension input variables. Each sample of observation X_i could impact $K+1$ samples in $\tilde{X}_i$ due to the assumption of maximum K delays. For example, $X_i(t-K)$ in Figure 5.2, could impact several samples of $\tilde{X}_i$: $\{\tilde{X}_i(t-K), \tilde{X}_i(t-K+1), \cdots, \tilde{X}_i(t)\}$. The solid line from each $X_i(t-K)$ represents the time delay that actually takes effect and there is only one solid line from each sample of $X_i(t-K)$ according to the property of indicator variable I in (5.4).

Time delay can decrease or increase in this time-varying time delay problem, which can cause following two issues: missing value in $\tilde{X}$ or conflict value in $\tilde{X}$. Taking one dimensional X as an example, these two scenarios are illustrated in Figure 5.3 and Figure 5.4 as following:

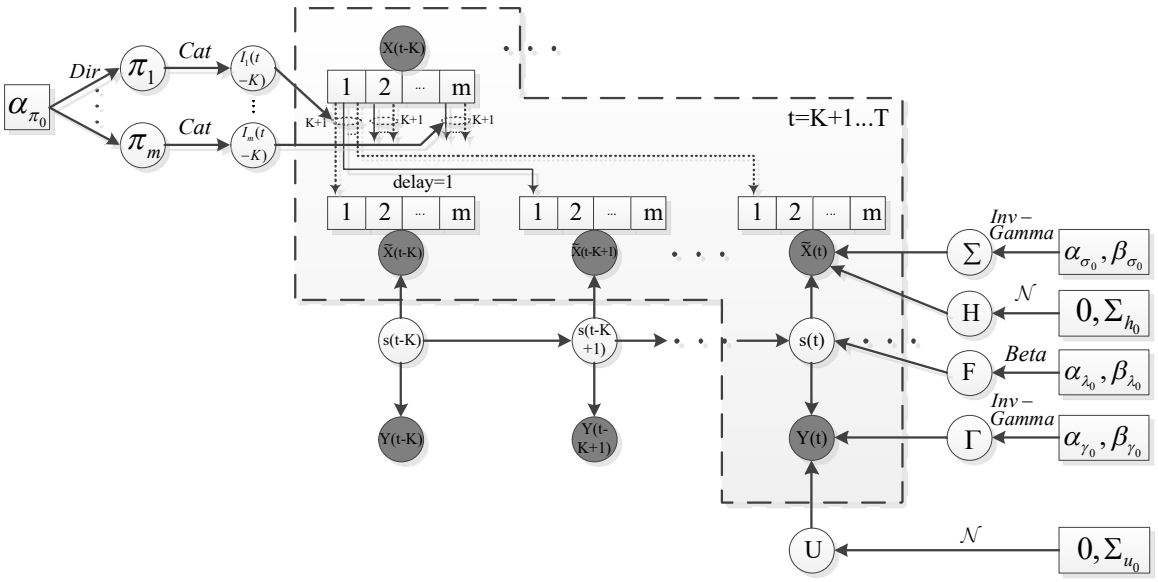

Figure 5.2: Graphical structure of IOPSFA with time-varying time delays

1. Delay decrease causing conflict in $\tilde{X}(t)$:

   In Figure 5.3, delay for $X(t-1)$ decreases by 1 comparing to $X(t-2)$, which causes the conflict in $\tilde{X}(t)$ since it assumes values from both $X(t-2)$ and $X(t-1)$. The proposed model will be designed to fuse observations under probabilistic formulation.

2. Delay increase causing missing value in $\tilde{X}(t)$:

   In Figure 5.4, delay for $X(t)$ increases by 1 comparing to $X(t-1)$, which causes the missing value of $\tilde{X}(t+1)$, namely $\tilde{X}(t+1)$ does not correspond to any physical value. Under probabilistic formulation, the proposed model will fuse the $X(t+1)$ and previous K observation samples as $\tilde{X}(t+1)$ when inferring the latent feature $s(t+1)$. It is worth noting that there is no information missing in the inference of s since all observation samples are now used under probabilistic formulation.

5.2.1.3 Prior Assignment

In this problem, the unknown parameter set is $\Theta = \{F, H, U, \Sigma, \Gamma\}$ and the latent variable set is: $L = \{s, \pi_i, I_i, 1 \leq i \leq m\}$. The unobserved variable set combines them together and is denoted as $Z = \{\Theta, L\} = \{\Theta, s, \pi_i, I_i, 1 \leq i \leq m\}$. To solve this

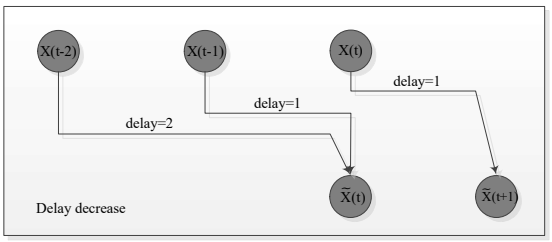

Figure 5.3: Delay decrease scenario

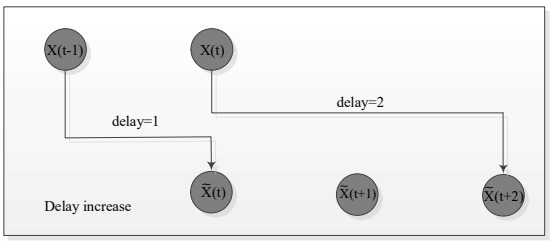

Figure 5.4: Delay increase scenario

problem using variational Bayesian (VB) approach, proper priors need to be assigned to each unknown parameters and latent variables.

- Latent variable s(t):

$$p(s(t)|s(t-1), F, \Lambda) = \mathcal{N}(s(t)|Fs(t-1), \Lambda) \tag{5.9}$$

- Latent variable $I(t)$, π and parameter M:
  For each $I_i, 1 \leq i \leq m$:

$$p(I_i|\pi_i) = \prod_{t=2}^{T} p(I_i(t)|I_i(t-1), M_i) \cdot p(I_i(1)|\pi_i) \tag{5.10}$$

  where, π_i is the hyper-parameter of the Categorical distribution and it is modeled by the Dirichlet distribution. π_i is a vector with the same dimensionality as $I_i(t)$ and the summation of all the elements of π_i is equal to one. The elements of π_i represents the probabilities of the corresponding time delay case, so all the elements of π_i take positive values. The value of π_i can be defined as known by incorporating priori process knowledges. Alternatively, Dirichlet distribution can be assigned to it as the prior since it is the conjugate prior of the Categorical distribution.

$$p(\pi_i|\alpha_{\pi_0}) = Dir(\pi_i|\alpha_{\pi_0}) = \frac{1}{B(\alpha_{\pi_0})} \prod_{k=0}^{K} (\pi_i^{(k)})^{\alpha_{\pi_0}^{(k)}-1} = \frac{\Gamma(\sum_{k=0}^{K} \alpha_{\pi_0}^{(k)})}{\prod_{k=0}^{K} \Gamma(\alpha_{\pi_0}^{(k)})} \prod_{k=0}^{K} (\pi_i^{(k)})^{\alpha_{\pi_0}^{(k)}-1} \tag{5.11}$$

  We use the symmetric Dirichlet priors with a fixed strength $f^{(\pi_i)}$:

$$\alpha_{\pi_0} = \Big[\frac{f^{(\pi_i)}}{K+1}, \cdots, \frac{f^{(\pi_i)}}{K+1}\Big], \quad s.t.\ f^{(\pi_i)} = \sum_{k=0}^{K} \alpha_{\pi_0}^{(k)} \tag{5.12}$$

Similarly, each column of the transition matrix M_i follows a Dirichlet distribution:

$$p(M_i) = \prod_{k=0}^{K} Dir(\{M_j^{(k,0)}, \cdots, M_j^{(k,K)}\} | \{\alpha_{M_0}^{(k,0)}, \cdots, \alpha_{M_0}^{(k,K)}\}) \tag{5.13}$$

with strength $f^{(M_i)}$:

$$\alpha_{M_0} = \Big[\frac{f^{(M_i)}}{K+1}, \cdots, \frac{f^{(M_i)}}{K+1}\Big], \quad s.t.\ f^{(M_i)} = \sum_{k=0}^{K} \alpha_{M_0}^{(k)} \tag{5.14}$$

- Unknown parameter F:

  $F = \text{diag}\{\lambda_1, \cdots, \lambda_q\}$, and the prior of each λ_j is chosen as Beta distribution since it is commonly applied to model the random variables that distribute within finite intervals. In this case, λ_i, indicating the varying speed of s_i, is constrained in the interval [0,1). The shape of the probability density function (pdf) of beta distribution can be manipulated by tuning the shape parameters $\alpha_{\lambda_0}, \beta_{\lambda_0}$ to have the preference $\lambda_j \to 1$ (namely slowness):

  $$p(\lambda_j | \alpha_{\lambda_0}, \beta_{\lambda_0}) = Beta(\lambda_j | \alpha_{\lambda_0}, \beta_{\lambda_0}) \tag{5.15}$$

- Unknown parameter H:

  $H = [h_1, \cdots, h_m]^T$, and the prior of each row h_i has normal distribution

  $$p(h_i | 0, \Sigma_{h_0}) = \mathcal{N}(h_i | 0, \Sigma_{h_0}) \tag{5.16}$$

  Σ_{h_0} is the hyperparameter which can be set as, e.g. $\Sigma_{h_0} = \text{diag}\{0.001, \cdots, 0.001\}_{q \times q}$.

- Unknown parameter U:

  $U = [u_1, \cdots, u_l]^T$, and the prior of each row u_i has normal distribution

  $$p(u_i | 0, \Sigma_{u_0}) = \mathcal{N}(u_i | 0, \Sigma_{u_0}) \tag{5.17}$$

  Σ_{u_0} is the hyperparameter which can be set as, e.g. $\Sigma_{u_0} = \text{diag}\{0.001, \cdots, 0.001\}_{q \times q}$.

- Unknown parameter Σ:

  $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and the observation $X_i(t)$ follows normal distribution with fixed mean (zero mean). Its conjugate prior is inverse gamma distribution

  $$p(\sigma_i^2 | \alpha_{\sigma_0}, \beta_{\sigma_0}) = Inv\text{-}Gamma(\sigma_i^2 | \alpha_{\sigma_0}, \beta_{\sigma_0}) \tag{5.18}$$

- Unknown parameter Γ:

  $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ and the observation $Y_j(t)$ follows normal distribution with fixed mean (zero mean). Its conjugate prior is also inverse gamma distribution.

$$p(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) = Inv\text{-}Gamma(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) \tag{5.19}$$

5.2.2 Variational Bayesian Inference

After assigning priors to all latent variables and unknown parameters, in order to maximize the model log-evidence, we introduce a variational distribution of unobserved variables $q(Z)$ and decompose the log model evidence:

$$\begin{aligned} \ln p(X,Y) &= \int_{q(Z)} q(Z) \ln \frac{p(X,Y,Z)}{p(Z|X,Y)} \mathrm{d}Z \\ &= \int_{q(Z)} q(Z) \ln \frac{p(X,Y,Z)}{q(Z)} \mathrm{d}Z + \int_{q(Z)} q(Z) \ln \frac{q(Z)}{p(Z|X,Y)} \mathrm{d}Z \\ &= F(q(Z)) + KL(q(Z)||p(Z|X,Y)) \end{aligned} \tag{5.20}$$

From (5.20), the log model evidence is decomposed into two terms: the first term $F(q(Z))$ is called variational free energy, which is the lower bound of the log model evidence. The second term $KL\big(q(Z)||p(Z|X,Y)\big)$ is the Kullback-Leibler (KL) divergence between the proposed variational distribution $q(Z)$ and true posterior distribution $p(Z|X,Y)$. Since the log model evidence is constant, to minimize the KL divergence is equivalent to maximizing the variational free energy. To achieve this, the proposal distribution can be factorized according to the mean-field theory to approximate the posterior distribution $P(Z|X,Y)$ as:

$$q(Z) = \prod_j q_j(Z_j) \xrightarrow{approx.} P(Z|X,Y) \tag{5.21}$$

in which, Z_j represents a variable group that contains one or several variables of the unobserved variables data set Z. Then we use VBEM to optimize each group of variables in turn while fixing all other variables. Normally, the maximization of the variational free energy is complicated and not analytically solvable due to the multiple integration involved. It is unlikely to obtain the optimal value for all observed variables at the same time. Taking advantage of the mean-field approximation, we

can maximize $F(q(Z))$ with respect to one group of unobserved variables while fixing all others with following factorization [134] in each iteration:

$$q(Z) = q_j(Z_j) \prod_{Z_{\setminus j}} q_{\setminus j}(Z_{\setminus j}) \tag{5.22}$$

$Z_{\setminus j}$ represents all other unobserved variables except Z_j. When $F(q(Z))$ is maximized with respect to Z_j, all other unobserved variables are fixed. Next, $F(q(Z))$ can be maximized with respect to another unobserved variable in $Z_{\setminus j}$. Apparently, we make the assumption that each variable group contributes independently to the target multivariate posterior [135]. The general solution is given by [73]:

$$q_j^*(Z_j) \propto \exp\Big(\mathbb{E}_{q_{\setminus j}(Z_{\setminus j})}\big[\ln p(X, Y, Z)\big]\Big) \tag{5.23}$$

In the following subsections, the optimal solution for each variables will be derived respectively.

5.2.2.1 Inference of H, U, Σ, and Γ

- Inference of H

Since $H = [h_1, \cdots, h_m]^T \in \mathbb{R}^{m \times q}$, and for each dimension $h_i \in \mathbb{R}^{q \times 1}, 1 \leq i \leq m$ has a normal distribution as prior:

$$p(h_i|0, \Sigma_{h_0}) = \mathcal{N}(h_i|0, \Sigma_{h_0}) \tag{5.24}$$

and its posterior to be inferred:

$$\begin{aligned} q^*(h_i) =& \mathcal{N}(h_i|\hat{\mu}_{h_i}, \hat{\Sigma}_{h_i}) \\ \Rightarrow \ln q^*(h_i) =& -\frac{1}{2}(h_i - \hat{\mu}_{h_i})^T \hat{\Sigma}_{h_i}^{-1}(h_i - \hat{\mu}_{h_i}) - \frac{1}{2}\ln(2\pi)^d \hat{\Sigma}_{h_i} \end{aligned} \tag{5.25}$$

where, the parameters with $\hat{}$ represent the hyper-parameters of the optimal posterior distribution of the corresponding unknown parameter. According to the model formulation in (5.1), only the i-th dimension of X depends on h_i: $p(X_i(t)|h_i, s(t)) = \mathcal{N}(h_i^T s(t), \sigma_i^2)$. Applying the general solution in (5.23), then the optimal posterior of h_i is given in (5.26)

$$
\begin{aligned}
\ln q^*_{h_i} \propto & \mathbb{E}_{q_{\setminus h_i}(Z_{\setminus h_i})}\big[\ln p(X,Y,Z)\big] \\
= & \Big\langle \big[\ln p(X_i|h_i,I_i,s,\sigma_i^2)p(h_i|0,\Sigma_{h_0})\big]\Big\rangle \\
= & \Big\langle \Big[\sum_{t=1}^{T}\sum_{k=0}^{K} \ln \mathcal{N}(X_i(t-k)|h_i^T s(t),\sigma_i^2)^{I_i^{(k)}(t-k)} + \ln \mathcal{N}(0,\Sigma_{h_0})\Big]\Big\rangle \\
= & \Big\langle \Big[\sum_{t=1}^{T}\sum_{k=0}^{K} \Big(-\frac{1}{2}\big(X_i(t-k)-h_i^T s(t)\big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t-k)-h_i^T s(t)\big) - \frac{1}{2}\ln(2\pi)\sigma_i^2\Big) \\
& \cdot I_i^{(k)}(t-k) - \frac{1}{2}h_i^T\Sigma_{h_0}^{-1}h_i^T - \frac{1}{2}\ln(2\pi)^q\Sigma_{h_0}\Big]\Big\rangle \\
= & \Big\langle \Big[-\frac{1}{2}h_i^T\Sigma_{h_0}^{-1}h_i^T - \frac{1}{2}\ln(2\pi)^q\Sigma_{h_0} - \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big\{\Big[X_i(t-k)X_i(t-k) + \big(h_i^T s(t)\big)^T\big(h_i^T s(t)\big) \\
& - X_i(t-k)h_i^T s(t) - \big(h_i^T s(t)\big)^T X_i(t-k)\Big]\cdot\sigma_i^{-2} - \frac{1}{2}\ln(2\pi)\sigma_i^2\Big\}\cdot I_i^{(k)}(t-k)\Big]\Big\rangle \\
= & \Big\langle \Big[-\frac{1}{2}h_i^T\Big(\Sigma_{h_0}^{-1} + \sigma_i^{-2}\sum_{t=1}^{T}\sum_{k=0}^{K} s(t)s^T(t)I_i^{(k)}(t-k)\Big)h_i \\
& - \sum_{t=1}^{T}\sum_{k=0}^{K} h_i s^T(t)X_i(t-k)\sigma_i^{-2}I_i^{(k)}(t-k) + \cdots\Big]\Big\rangle
\end{aligned} \tag{5.26}
$$

Comparing the quadratic term and linear term with respect to h_i in (5.25) and (5.26), the update equations for the hyper-parameters of h_i are derived as:

$$
\begin{aligned}
\hat{\Sigma}_{h_i}^{-1} = & \Sigma_{h_0}^{-1} + \Big\langle \sigma_i^{-2}\sum_{t=1}^{T}\sum_{k=0}^{K}\big[s(t)s^T(t)I_i^{(k)}(t-k)\big]\Big\rangle \\
= & \Sigma_{h_0}^{-1} + \big\langle\sigma_i^{-2}\big\rangle\cdot\sum_{t=1}^{T}\sum_{k=0}^{K}\big\langle s(t)s^T(t)\big\rangle\big\langle I_i^{(k)}(t-k)\big\rangle \qquad (5.27) \\
\hat{\mu}_{h_i} = & \hat{\Sigma}_{h_i}\cdot\big\langle\sigma_i^{-2}\big\rangle\cdot\sum_{t=1}^{T}\sum_{k=0}^{K}\big\langle s(t)X_i(t-k)I_i^{(k)}(t-k)\big\rangle \\
= & \hat{\Sigma}_{h_i}\cdot\big\langle\sigma_i^{-2}\big\rangle\cdot\sum_{t=1}^{T}\sum_{k=0}^{K}X_i(t-k)\big\langle s(t)\big\rangle\big\langle I_i^{(k)}(t-k)\big\rangle \qquad (5.28)
\end{aligned}
$$

where, $\langle\cdot\rangle$ is the expectation operator and we will use it to simplify the derivation in following sections.

- Inference of Σ

Since $\Sigma = \text{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}$ and σ_i^2 has inverse gamma distribution as its prior:

$$p(\sigma_i^2|\alpha_{\sigma_0}, \beta_{\sigma_0}) = Inv\text{-}Gamma(\sigma_i^2|\alpha_{\sigma_0}, \beta_{\sigma_0}) \tag{5.29}$$

and its posterior to be inferred:

$$\begin{aligned} q^*(\sigma_i^2) =& Inv\text{-}Gamma(\sigma_i^2|\hat{\alpha}_{\sigma_i}, \hat{\beta}_{\sigma_i}) \\ \Rightarrow \ln q^*(\sigma_i^2) =& \hat{\alpha}_{\sigma_i} \ln \hat{\beta}_{\sigma_i} - (\hat{\alpha}_{\sigma_i} + 1) \ln \sigma_i^2 - \hat{\beta}_{\sigma_i} \sigma_i^{-2} - \ln \Gamma(\hat{\alpha}_{\sigma_i}) \end{aligned} \tag{5.30}$$

The likelihood follows a normal distribution:

$$p(X_i|h_i, s, \sigma_i^2, I_i) = \sum_{t=1}^{T} \sum_{k=0}^{K} \mathcal{N}(X_i(t-k)|h_i^T s(t), \sigma_i^2)^{I_i^{(k)}(t-k)} \tag{5.31}$$

According to general solution in (5.23), the optimal posterior distribution of σ_i^2 can be derived as:

$$\begin{aligned} \ln q_\Sigma^* \propto& \mathbb{E}_{q_{\setminus\Sigma}(Z_{\setminus\Sigma})}\big[\ln p(X, Y, Z)\big] \\ =& \Big\langle \ln p(X_i|h_i, s, \sigma_i^2, I_i) \cdot p(\sigma_i^2|\alpha_{\sigma_0}, \beta_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \sum_{k=0}^{K} \mathcal{N}(X_i(t-k)|h_i^T s(t), \sigma_j^2)^{I_i^{(k)}(t-k)} + \ln Inv\text{-}Gamma(\alpha_{\sigma_0}, \beta_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \sum_{k=0}^{K} \Big[-\frac{1}{2} \big(X_i(t-k) - h_i^T s(t)\big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t-k) - h_i^T s(t)\big) - \frac{1}{2} \ln(2\pi) \\ & + \frac{1}{2} \ln \sigma_i^{-2} \Big] \cdot I_i^{(k)}(t-k) + \alpha_{\sigma_0} \ln \beta_{\sigma_0} - (\alpha_{\sigma_0} + 1) \ln \sigma_i^2 - \beta_{\sigma_0} \sigma_i^{-2} - \ln \Gamma(\alpha_{\sigma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \Big(\sum_{k=0}^{K} \Big[-\frac{1}{2} \big(X_i(t-k) - h_i^T s(t)\big)^T \cdot \sigma_i^{-2} \cdot \big(X_i(t-k) - h_i^T s(t)\big) \Big] - \frac{1}{2} \ln(2\pi) \\ & - \frac{1}{2} \ln \sigma_i^2 \Big) \cdot I_i^{(k)}(t-k) + \alpha_{\sigma_0} \ln \beta_{\sigma_0} - (\alpha_{\sigma_0} + 1) \ln \sigma_i^2 - \beta_{\sigma_0} \sigma_i^{-2} - \ln \Gamma(\alpha_{\sigma_0}) \Big\rangle \end{aligned} \tag{5.32}$$

Comparing the coefficients of $\ln \sigma_i^2$ term and linear term of σ_i^{-2} term for (5.30) and (5.32), the update equation for the hyper-parameters of Σ are derived as:

$$\begin{aligned} -(\hat{\alpha}_{\sigma_i} + 1) =& -\frac{1}{2} \sum_{t=1}^{T} \sum_{k=0}^{K} \big\langle I_j^{(k)}(t-k) \big\rangle - (\alpha_{\sigma_0} + 1)k \\ \Rightarrow \hat{\alpha}_{\sigma_i} =& \frac{1}{2} \sum_{t=1}^{T} \sum_{k=0}^{K} \big\langle I_j^{(k)}(t-k) \big\rangle + \alpha_{\sigma_0} \end{aligned} \tag{5.33}$$

And,

$$
\begin{aligned}
-\hat{\beta}_{\sigma_i} &= -\beta_{\sigma_0} + \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K}\Big[-\frac{1}{2}\big(X_i(t-k) - h_i^T s(t)\big)^T \cdot \big(X_i(t-k) - h_i^T s(t)\big)\Big] \cdot I_i^{(k)}(t-k)\Big\rangle \\
\Rightarrow \hat{\beta}_{\sigma_i} &= \beta_{\sigma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[X_i(t-k)X_i(t-k) - 2X_i(t-k)h_i^T s(t) \\
&\quad + s(t)^T h_i h_i^T s(t)\Big] \cdot I_i^{(k)}(t-k)\Big\rangle \\
\Rightarrow \hat{\beta}_{\sigma_i} &= \beta_{\sigma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[X_i(t-k)X_i(t-k) - 2X_i(t-k)h_i^T s(t) \\
&\quad + tr\big(s(t)^T h_i h_i^T s(t)\big)\Big] \cdot I_i^{(k)}(t-k)\Big\rangle \\
&= \beta_{\sigma_0} + \frac{1}{2}\sum_{t=1}^{T}\sum_{k=0}^{K}\Big[X_i(t-k)X_i(t-k) - 2X_i(t-k)\big\langle h_i^T\big\rangle\big\langle s(t)\big\rangle \\
&\quad + tr\big[\big\langle h_i h_i^T\big\rangle \cdot \big\langle s(t)s^T(t)\big\rangle\big]\Big] \cdot \big\langle I_i^{(k)}(t-k)\big\rangle
\end{aligned} \tag{5.34}
$$

- Inference of U

Since $U = [u_1, \cdots, u_l]^T \in \mathbb{R}^{l\times q}$, and for each dimension $u_j \in \mathbb{R}^{q\times 1}, 1 \le j \le l$ has normal distribution as prior:

$$
p(u_j|0, \sigma_{u_0}^2) = \mathcal{N}(u_j|0, \sigma_{u_0}^2) \tag{5.35}
$$

and its posterior to be inferred:

$$
\begin{aligned}
q^*(u_j) =& \mathcal{N}(u_j|\hat{\mu}_{u_j}, \hat{\sigma}_{u_j}^2) \\
\Rightarrow \ln q^*(u_j) =& -\frac{1}{2}(u_j - \hat{\mu}_{u_j})^T \hat{\sigma}_{u_j}^{-2}(u_j - \hat{\mu}_{u_j}) - \frac{1}{2}\ln(2\pi)^q \hat{\sigma}_{u_j}^2
\end{aligned} \tag{5.36}
$$

Only j-th dimension of Y depends on u_j: $p(Y_j(t)|u_j, s(t)) = \mathcal{N}(u_j^T s(t), \gamma_j^2)$ and ac-

cording to the general solution in (5.23)

$$
\begin{aligned}
\ln q^*_{u_j} \propto & \mathbb{E}_{q_{\setminus u_j}(Z_{\setminus u_j})}\big[\ln p(X,Y,Z)\big] \\
=& \Big\langle \ln p(Y_j|u_j,s,\gamma_j^2)p(u_j|0,\sigma^2_{u_0})\Big\rangle \\
=& \Big\langle \sum_{t=1}^{T} \ln \mathcal{N}(Y_j(t)|u_j^T s(t),\gamma_j^2) + \ln \mathcal{N}(u_j|0,\sigma^2_{u_0})\Big\rangle \\
=& \Big\langle \sum_{t=1}^{T} \Big(-\frac{1}{2}\big(Y_j(t)-u_j^T s(t)\big)^T \cdot \gamma_j^{-2} \cdot \big(Y_j(t)-u_j^T s(t)\big) - \frac{1}{2}\ln(2\pi)\gamma_j^2\Big) \\
& -\frac{1}{2}u_j^T\sigma_{u_0}^{-2}u_j^T - \frac{1}{2}\ln(2\pi)^q\sigma^2_{u_0}\Big\rangle \\
=& \Big\langle -\frac{1}{2}u_j^T\sigma_{u_0}^{-2}u_j^T - \frac{1}{2}\ln(2\pi)^q\sigma^2_{u_0} - \frac{1}{2}\sum_{t=1}^{T}\Big\{\Big[Y_j(t)Y_j(t) + \big(u_j^T s(t)\big)^T\big(u_j^T s(t)\big) \\
& - Y_j(t)u_j^T s(t) - \big(u_j^T s(t)\big)^T Y_j(t)\Big]\cdot\gamma_j^{-2} - \frac{1}{2}\ln(2\pi)\gamma_j^2\Big\}\Big\rangle \\
=& \Big\langle -\frac{1}{2}u_j^T\Big(\sigma_{u_0}^{-2} + \gamma_j^{-2}\sum_{t=1}^{T} s(t)s^T(t)\Big)u_j - \sum_{t=1}^{T} u_j^T s^T(t)Y_j(t)\gamma_j^{-2} + \cdots\Big\rangle \qquad (5.37)
\end{aligned}
$$

Comparing the quadratic term and linear term with respect to u_j for (5.36) and (5.37), the update equation for the hyper-parameters of u_j are derived as:

$$\hat{\sigma}_{u_j}^{-2} = \sigma_{u_0}^{-2} + \Big\langle \gamma_j^{-2}\sum_{t=1}^{T}\big[s(t)s^T(t)\big]\Big\rangle = \sigma_{u_0}^{-2} + \big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t=1}^{T}\big\langle s(t)\cdot s^T(t)\big\rangle \qquad (5.38)$$

$$\hat{\mu}_{u_j} = \hat{\sigma}_{u_j}^2\cdot\big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t=1}^{T}\big\langle s(t)\cdot Y_j(t)\big\rangle \qquad (5.39)$$

Since the output Y(t) is not always available, we adopt the similar technique as deriving the IOPSFA that T can be divided into two parts, $T = \{T_{obs}, T_{mis}\}$, in which, T_{obs} are the time stamps at which inputs are labeled and T_{mis} are the time stamps at which inputs are unlabeled, i.e., outputs are missing. The updating equation (5.38) and (5.39) become:

$$\hat{\sigma}_{u_j}^{-2} = \sigma_{u_0}^{-2} + \big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t\in T_{obs}}\big\langle s(t)s^T(t)\big\rangle \qquad (5.40)$$

$$\hat{\mu}_{u_j} = \hat{\sigma}_{u_j}^2\cdot\big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t\in T_{obs}}\big\langle s(t)\cdot Y_j(t)\big\rangle = \hat{\sigma}_{u_j}^2\cdot\big\langle\gamma_j^{-2}\big\rangle\cdot\sum_{t\in T_{obs}} Y_j(t)\big\langle s(t)\big\rangle \qquad (5.41)$$

- Inference of Γ

Since $\Gamma = \text{diag}\{\gamma_1^2, \cdots, \gamma_l^2\}$ and γ_j^2 has inverse gamma distribution as its prior:

$$p(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) = \textit{Inv-Gamma}(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) \tag{5.42}$$

and its posterior to be inferred:

$$\begin{aligned} q^*(\gamma_j^2) =& \textit{Inv-Gamma}(\gamma_j^2|\hat{\alpha}_{\gamma_j}, \hat{\beta}_{\gamma_j}) \\ \Rightarrow \ln q^*(\gamma_j^2) =& \hat{\alpha}_{\gamma_j} \ln \hat{\beta}_{\gamma_j} - (\hat{\alpha}_{\gamma_j} + 1) \ln \gamma_j^2 - \hat{\beta}_{\gamma_j} \gamma_j^{-2} - \ln \Gamma(\hat{\alpha}_{\gamma_j}) \end{aligned} \tag{5.43}$$

The likelihood follows a normal distribution:

$$p(Y_j(t)|u_j, s(t), \gamma_j^2) = \mathcal{N}(Y_j(t)|u_j^T s(t), \gamma_j^2) \tag{5.44}$$

According to general solution in (5.23), the optimal posterior distribution of γ_j^2 can be derived as:

$$\begin{aligned} \ln q_\Gamma^* \propto& \mathbb{E}_{q_{\backslash\Gamma}(Z_{\backslash\Gamma})}\big[\ln p(X, Y, Z)\big] \\ =& \Big\langle \ln p(Y_j|u_j, s, \gamma_j^2) \cdot p(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \ln \mathcal{N}(u_j^T s(t), \gamma_j^2) + \ln \textit{Inv-Gamma}(\gamma_j^2|\alpha_{\gamma_0}, \beta_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \Big[-\frac{1}{2}\big(Y_j(t) - u_j^T s(t)\big)^T \cdot \gamma_j^{-2} \cdot \big(Y_j(t) - u_j^T s(t)\big) - \frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln \gamma_j^{-2} \Big] \\ & + \alpha_{\gamma_0} \ln \beta_{\gamma_0} - (\alpha_{\gamma_0} + 1) \ln \gamma_j^2 - \beta_{\gamma_0} \gamma_j^{-2} - \ln \Gamma(\alpha_{\gamma_0}) \Big\rangle \\ =& \Big\langle \sum_{t=1}^{T} \Big[-\frac{1}{2}\big(Y_j(t) - u_j^T s(t)\big)^T \cdot \gamma_j^{-2} \cdot \big(Y_j(t) - u_j^T s(t)\big) \Big] - \frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln \gamma_j^{-2} \\ & + \alpha_{\gamma_0} \ln \beta_{\gamma_0} - (\alpha_{\gamma_0} + 1) \ln \gamma_j^2 - \beta_{\gamma_0} \gamma_j^{-2} - \ln \Gamma(\alpha_{\gamma_0}) \Big\rangle \end{aligned} \tag{5.45}$$

Comparing the coefficients of $\ln \gamma_j^2$ and linear term of γ_j^{-2} term for (5.43) and (5.45), the update equations for the hyper-parameters of Γ are derived as:

$$-(\hat{\alpha}_\gamma + 1) = -\frac{T}{2} - (\alpha_{\gamma_0} + 1) \Rightarrow \hat{\alpha}_{\gamma_j} = \frac{T}{2} + \alpha_{\gamma_0} \tag{5.46}$$

$$
\begin{aligned}
-\hat{\beta}_{\gamma_j} &= -\beta_{\gamma_0} + \Big\langle \sum_{t=1}^{T} \Big[-\frac{1}{2}\big(Y_j(t) - u_j^T s(t)\big)^T \cdot \big(Y_j(t) - u_j^T s(t)\big)\Big]\Big\rangle \\
\Rightarrow \hat{\beta}_{\gamma_j} &= \beta_{\gamma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T} \Big[Y_j(t)Y_j(t) - 2Y_j(t)u_j^T s(t) + s(t)^T u_j u_j^T s(t)\Big]\Big\rangle \\
\Rightarrow \hat{\beta}_{\gamma_j} &= \beta_{\gamma_0} + \Big\langle \frac{1}{2}\sum_{t=1}^{T} \Big[Y_j(t)Y_j(t) - 2Y_j(t)u_j^T s(t) + tr\big(s(t)^T u_j u_j^T s(t)\big)\Big]\Big\rangle \\
&= \beta_{\gamma_0} + \frac{1}{2}\sum_{t=1}^{T} \Big[Y_j(t)Y_j(t) - 2Y_j(t)\big\langle u_j^T\big\rangle\big\langle s(t)\big\rangle + tr\big[\big\langle u_j u_j^T\big\rangle \cdot \big\langle s(t)s^T(t)\big\rangle\big]\Big]
\end{aligned}
\tag{5.47}
$$

The update equation for $\hat{\beta}_\gamma$ (5.47) reduces to the following when there are missing observations in Y:

$$
\hat{\beta}_{\gamma_j} = \beta_{\gamma_0} + \frac{1}{2}\sum_{t\in T_{obs}}^{T} \Big[Y_j(t)Y_j(t) - 2Y_j(t)\big\langle u_j^T\big\rangle\big\langle s(t)\big\rangle + tr\big[\big\langle u_j u_j^T\big\rangle \cdot \big\langle s(t)s^T(t)\big\rangle\big]\Big]
\tag{5.48}
$$

5.2.2.2 Inference of I_i, π_i and M_i

The posterior of π_i and M_i can be derived as follows according to the general solution of (5.23):

$$
\begin{aligned}
q^*(\pi_i) =& Dir(\{\pi_i^{(0)}, \cdots, \pi_i^{(K)}\}|\hat{\alpha}_\pi) && (5.49)\\
& \text{with: } \hat{\alpha}_\pi = \alpha_{\pi_0} + \big\langle I_i(t)\big\rangle \\
q^*(M_i) =& \prod_{j=0}^{K} Dir(\{M_i^{(0,j)}, \cdots, M_i^{(K,j)}\}|\{\hat{\alpha}_{M_i}^{(0,j)}, \cdots, \hat{\alpha}_{M_i}^{(K,j)}\}) && (5.50)\\
& \text{with: } \hat{\alpha}_{M_i}^{(k,j)} = \alpha_{M_0}^{(k,j)} + \sum_{t=2}^{T} \big\langle I_i^{(k)}(t)\cdot I_i^{(j)}(t-1)\big\rangle
\end{aligned}
$$

Then the following statistic items can be calculated as:

$$
\big\langle M_i^{(k,j)}\big\rangle = \frac{\hat{\alpha}_{M_i}^{(k,j)}}{\sum_{k=0}^{K}\hat{\alpha}_{M_i}^{(k,j)}} \tag{5.51}
$$

$$
\big\langle \ln M_i^{(k,j)}\big\rangle = \psi(\hat{\alpha}_{M_i}^{(k,j)}) - \psi(\sum_{k=0}^{K}\hat{\alpha}_{M_i}^{(k,j)}) \tag{5.52}
$$

And the posterior of I_i can be derived as follows:

$$
\begin{aligned}
\ln q^*_{I_i} \propto & \mathbb{E}_{q_{\setminus I_i}(Z_{\setminus I_i})}\big[\ln p(X,Y,Z)\big] \\
= & \Big\langle \ln p(X_i|s,I_i,h_i,\sigma_i^2) + \ln p(I_i|\pi_i,M_i) \Big\rangle \\
= & \Big\langle \ln \prod_{t=1}^{T}\prod_{k=0}^{K} p(X_i(t)|h_i^T, s(t+k), I_i^{(k)}(t), \sigma_i^2) \\
& + \ln \prod_{t=2}^{T} p(I_i(t)|I_i(t-1),M_i) + \ln p(I_i(1)|\pi_i) \Big\rangle \\
= & \Big\langle \sum_{t=1}^{T}\sum_{k=0}^{K} \ln \Big[\mathcal{N}(X_i(t)|h_i^T\cdot s(t+k),\sigma_i^2)\Big]^{I_i^{(k)}(t)} \\
& + \sum_{t=2}^{T} \ln \Big[\prod_{k=0}^{K}\prod_{k'=0}^{K} \big(M_i^{(k,k')}\big)^{I_i^{(k)}(t)\cdot I_i^{(k')}(t)}\Big] + \sum_{k=0}^{K} \ln (\pi_i^{(k)})^{I_i^{(k)}(1)} \Big\rangle \\
= & \sum_{t=1}^{T}\sum_{k=0}^{K} I_i^{(k)}(t)\cdot \Big\langle \ln \mathcal{N}(X_i(t)|h_i^T\cdot s(t+k),\sigma_i^2) \Big\rangle \\
& + \sum_{t=2}^{T}\sum_{k=0}^{K}\sum_{k'=0}^{K} I_i^{(k)}(t)\cdot I_i^{(k')}(t) \Big\langle \ln M_i^{(k,k')} \Big\rangle + \sum_{k=0}^{K} I_i^{(k)}(1) \Big\langle \ln \pi_i^{(k)} \Big\rangle
\end{aligned} \tag{5.53}
$$

The approximated posterior of I_i has a similar form as the HMM's complete-data likelihood [134] except the expectation is now taken on the logarithm of the parameters. In order to use the HMM forward-backward algorithm in the inference of I_i, we utilize the corollary 2.2 in [134], which can be explained briefly as follows as an example. For a unknown parameter set that consists of three parameters: $\theta = \{\theta_1, \theta_2, \theta_3\}$ and the natural (logarithm of the) parameter set is: $\phi(\theta) = \{\ln\theta_1, \ln\theta_2, \ln\theta_3\}$. The expected natural parameters set used in approximate posterior is: $\langle\phi(\theta)\rangle = \{\langle\ln\theta_1\rangle, \langle\ln\theta_2\rangle, \langle\ln\theta_3\rangle\}$ and the modified parameter set is: $\tilde{\theta} = \{\exp\langle\ln\theta_1\rangle, \exp\langle\ln\theta_2\rangle, \exp\langle\ln\theta_3\rangle\}$. We can use $\ln\tilde{\theta}$ derived above instead of $\langle\ln\theta\rangle$ since they generate the same logarithm likelihood according to the Corollary 2.2 in [134]:

$$
\langle \ln p(X|\theta)\rangle = \ln \tilde{p}(X|\tilde{\theta}) \tag{5.54}
$$

So, we set the modified parameters as:

$$\begin{aligned}\tilde{\mathcal{N}}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) &= \exp\Big\langle \ln \mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)\Big\rangle \\ \tilde{M}_i^{(k,k')} &= \exp\Big\langle \ln M_i^{(k,k')}\Big\rangle \\ \tilde{\pi}_i^{(k)} &= \exp\Big\langle \ln \pi_i^{(k)}\Big\rangle\end{aligned} \tag{5.55}$$

Substituting the new parameters in (5.55) to (5.53):

$$\begin{aligned}\ln q_{I_i}^* \propto & \ln \prod_{t=1}^{T}\prod_{k=0}^{K} \tilde{p}(X_i(t)|h_i^T, s(t+k), I_i^{(k)}(t), \sigma_i^2) \\ & + \ln \prod_{t=2}^{T} \tilde{p}(I_i(t)|I_i(t-1), \tilde{M}_i) + \ln \tilde{p}(I_i(1)|\tilde{\pi}_i) \\ = & \sum_{t=1}^{T}\sum_{k=0}^{K} I_i^{(k)}(t) \ln \tilde{\mathcal{N}}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2) \\ & + \sum_{t=2}^{T}\sum_{k=0}^{K}\sum_{k'=0}^{K} I_i^{(k)}(t) I_i^{(k')}(t) \ln \tilde{M}_i^{(k,k')} + \sum_{k=0}^{K} I_i^{(k)}(1) \ln \tilde{\pi}_i^{(k)}\end{aligned} \tag{5.56}$$

where, $\tilde{p}$ represents the probability density function with the modified parameters. Getting rid of the expectation operator in (5.56), we can use the forward-backward algorithm of HMM to infer I_i. In forward recursion, the posterior over I_i given the observed sequence up to and including current time t is defined as:

$$\begin{aligned}\tilde{\alpha}_t(I_i(t)) =& \tilde{p}(I_i(t)|X_i(1:t)) \\ =& \frac{\tilde{p}(X_i(t)|I_i(t), X_i(1:t-1)) \cdot \tilde{p}(I_i(t)|X_i(1:t-1))}{\tilde{p}(X_i(t)|X_i(1:t-1))} \\ =& \frac{1}{\tilde{p}(X_i(t)|X_i(1:t-1))} \sum_{I_i(t-1)} \tilde{p}(X_i(t)|I_i(t)) \\ & \cdot \tilde{p}(I_i(t)|I_i(t-1)) \cdot \tilde{p}(I_i(t-1)|X_i(1:t-1)) \\ =& \frac{1}{\xi(X_i(t))}\Bigg[\sum_{I_i(t-1)} \tilde{\alpha}_{t-1}(I_i(t-1)) \cdot \tilde{p}(I_i(t)|I_i(t-1))\Bigg] \tilde{p}(X_i(t)|I_i(t))\end{aligned} \tag{5.57}$$

where, $\xi(X_i(t)) = \tilde{p}(X_i(t)|X_i(1:t-1))$ is the normalization constant. Also, $\tilde{\beta}_t(I_i(t))$ is also defined in the backward recursion, which is carried out from $t = T, \cdots, 1$:

$$\begin{aligned}\tilde{\beta}_t(I_i(t)) =& \tilde{p}(X_i(t+1:T)|I_i(t)) \\ =& \sum_{I_i(t+1)} \tilde{p}(X_i(t+2:T)|I_i(t+1))\tilde{p}(I_i(t+1)|I_i(t))\tilde{p}(X_i(t+1)|I_i(t+1)) \\ =& \sum_{I_i(t+1)} \tilde{\beta}_{t+1}(I_i(t+1))\tilde{p}(I_i(t+1)|I_i(t))\tilde{p}(X_i(t+1)|I_i(t+1)) \end{aligned} \tag{5.58}$$

with the initial condition $\tilde{\beta}_t(I_i(t)) = 1$. So the posterior distribution can be written as:

$$\tilde{p}(I_i(t)|X_i(1:T)) \propto \tilde{\alpha}_t(I_i(t)) \cdot \tilde{\beta}_t(I_i(t)) \tag{5.59}$$

and the posterior of the joint distribution is

$$\tilde{p}(I_i(t-1), I_i(t)|X_i(1:T)) \propto \tilde{\alpha}_{t-1}(I_i(t-1)) \cdot \tilde{p}(I_i(t)|I_i(t-1)) \cdot \tilde{p}(X_i(t)|I_i(t)) \cdot \tilde{\beta}_t(I_i(t)) \tag{5.60}$$

In above equations, we need to calculate two terms: $\tilde{p}(X_i(t)|h_i^T, s, I_i^{(k)}(t), \sigma_i^2)$ and $\tilde{p}(I_i(t)|I_i(t-1), M_i)$. From corollary 2.2 in [134], we can obtain:

$$\begin{aligned} & \ln \tilde{p}(X_i(t)|h_i^T, s, I_i^{(k)}(t), \sigma_i^2) \\ =& \Big\langle \ln p(X_i(t)|h_i^T, s, I_i^{(k)}(t), \sigma_i^2) \Big\rangle \\ =& \sum_{k=0}^{K} I_i^{(k)}(t) \cdot \Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \end{aligned} \tag{5.61}$$

So,

$$\begin{aligned} & \tilde{p}(X_i(t)|h_i^T, s, I_i^{(k)}(t), \sigma_i^2) \\ =& \exp\Big\{ \sum_{k=0}^{K} I_i^{(k)}(t) \cdot \Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \Big\} \\ =& \prod_{k=0}^{K} I_i^{(k)}(t) \cdot \exp\Big\langle \ln[\mathcal{N}(X_i(t)|h_i^T \cdot s(t+k), \sigma_i^2)] \Big\rangle \end{aligned} \tag{5.62}$$

Similarly,

$$\begin{aligned} & \ln \tilde{p}(I_i(t)|I_i(t-1), M_i) \\ =& \Big\langle \ln p(I_i(t)|I_i(t-1), M_i) \Big\rangle \\ =& \sum_{k=0}^{K} \sum_{k'=0}^{K} I_i^{(k)}(t) \cdot I_i^{(k')}(t) \cdot \Big\langle \ln M_i^{(k,k')} \Big\rangle \end{aligned} \tag{5.63}$$

And,

$$\begin{aligned}
&\tilde{p}(I_i(t)|I_i(t-1), M_i) \\
=&\exp\Big\{\sum_{k=0}^{K}\sum_{k'=0}^{K} I_i^{(k)}(t)\cdot I_i^{(k')}(t)\cdot\Big\langle \ln M_i^{(k,k')}\Big\rangle\Big\} \\
=&\prod_{k=0}^{K}\prod_{k'=0}^{K} I_i^{(k)}(t)\cdot I_i^{(k')}(t)\cdot\exp\Big\langle \ln M_i^{(k,k')}\Big\rangle
\end{aligned} \tag{5.64}$$

By substituting (5.62) and (5.64), HMM forward and backward factor $\tilde{\alpha}_t(I_i(t))$ and $\tilde{\beta}_t(I_i(t))$ can be obtained in (5.57) and (5.58). Having obtained that, the posterior distribution of $I_i(t)$ and joint posterior distribution of $\big\{I_i(t), I_i(t-1)\big\}$ are calculated using (5.59) and (5.60), respectively. Then we can calculate the expectation terms: $\big\langle I_i^{(k)}(t)\cdot I_i^{(j)}(t-1)\big\rangle$ and $\big\langle I_i(t)\big\rangle$, which are needed in (5.49) and (5.50):

$$\big\langle I_j^{(k)}(t)\big\rangle = \frac{\tilde{\alpha}_t(I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))}{\sum_{k'=0}^{K}\tilde{\alpha}_t(I_j^{(k')}(t))\tilde{\beta}_t(I_j^{(k')}(t))} \tag{5.65}$$

$$\big\langle I_j^{(k)}(t)\cdot I_j^{(k')}(t-1)\big\rangle = \frac{\tilde{\alpha}_{t-1}(I_j^{(k')}(t-1))\tilde{M}_j^{(k,k')}\tilde{p}(X_i(t)|I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))}{\sum_{k=0}^{K}\sum_{k'=0}^{K}\tilde{\alpha}_{t-1}(I_j^{(k')}(t-1))\tilde{M}_j^{(k,k')}\tilde{p}(X_i(t)|I_j^{(k)}(t))\tilde{\beta}_t(I_j^{(k)}(t))} \tag{5.66}$$

5.2.2.3 Unified Inference of s

According to the general solution, we can write the optimal posterior distribution of latent variable s as:

$$
\begin{aligned}
\ln q_s^* \propto & \mathbb{E}_{q_{\setminus s}(Z_{\setminus s})}\big[\ln p(X,Y,Z)\big] \qquad (5.67)\\
=&\Big\langle \sum_{t=1}^{T}\sum_{i=1}^{m}\sum_{k=0}^{K}\ln p(X_i(t-k)|s(t),H,\Sigma,I_i)\\
&+\sum_{t=1}^{T}\sum_{j=1}^{l}\ln p(Y_j(t)|s(t),U,\Gamma)+\sum_{t=1}^{T}\ln p(s(t)|s(t-1),F)\Big\rangle\\
=&\sum_{t=1}^{T}\sum_{i=1}^{m}\sum_{k=0}^{K}-\frac{1}{2}\Big\langle[X_i(t-k)-h_i^T s(t)]^T\cdot\frac{1}{\sigma_i^2}\cdot[X_i(t-k)-h_i^T s(t)]\cdot I_i^{(k)}(t-k)\Big\rangle\\
&-\frac{1}{2}\sum_{t=1}^{T}\sum_{j=1}^{l}\Big\langle[Y_j(t)-u_j^T s(t)]^T\cdot\frac{1}{\gamma_j^2}\cdot[Y_j(t)-u_j^T s(t)]\Big\rangle\\
&+\sum_{t-1}^{T}-\frac{1}{2}s^T(t)\Big\langle\frac{1}{1-F^2}\Big\rangle s(t)+s^T(t-1)\Big\langle\frac{F}{1-F^2}\Big\rangle s(t)\\
&-\frac{1}{2}s^T(t-1)\Big\langle\frac{F^2}{1-F^2}\Big\rangle s(t-1)+const\\
=&\sum_{t=1}^{T}-\frac{1}{2}s^T(t)\Big[\sum_{i=1}^{m}\sum_{k=0}^{K}\Big\langle h_i h_i^T\Big\rangle\Big\langle\sigma_i^2\Big\rangle^{-1}\Big\langle I_i^{(k)}(t-k)\Big\rangle+\sum_{j=1}^{l}\Big\langle u_j u_j^T\Big\rangle\Big\langle\gamma_j^2\Big\rangle^{-1}\Big]s(t)\\
&+s^T(t)\Big[\sum_{i=1}^{m}\sum_{k=0}^{K}\Big\langle h_i\Big\rangle\Big\langle\sigma_i^2\Big\rangle^{-1}\Big\langle X_i(t-k)\Big\rangle\Big\langle I_i^{(k)}(t-k)\Big\rangle+\sum_{j=1}^{l}\Big\langle u_j\Big\rangle\Big\langle\gamma_j^2\Big\rangle^{-1}\Big\langle y_j(t)\Big\rangle\Big]\\
&+\sum_{t-1}^{T}-\frac{1}{2}s^T(t)\Big\langle\frac{1}{1-F^2}\Big\rangle s(t)+s^T(t-1)\Big\langle\frac{F}{1-F^2}\Big\rangle s(t)\\
&-\frac{1}{2}s^T(t-1)\Big\langle\frac{F^2}{1-F^2}\Big\rangle s(t-1)+const
\end{aligned}
$$

The posterior distribution of s is Gaussian distribution and in order to utilize the standard Kalman filtering and smoothing technique in inference of s, unified inference technique [137] is used. The original LGSSM is transferred to an augmented LGSSM for which the standard Kalman filtering and smoothing can be used. In the augmented

LGSSM, the optimal solution should have following form [137, 138]:

$$\begin{aligned}\ln q_s^* =& \sum_{t=1}^{T} -\frac{1}{2} s^T(t)\tilde{F}_A(t)s(t) + \sum_{t=1}^{T} s^T(t)\tilde{F}_B(t) \\ &+ \sum_{t=1}^{T} -\frac{1}{2}[s(t) - \tilde{F}s(t-1)]^T\tilde{\Lambda}^{-1}[s(t) - \tilde{F}s(t-1)] + \text{const} \\ =& \sum_{t=1}^{T} -\frac{1}{2} s^T(t)\tilde{F}_A(t)s(t) + \sum_{t=1}^{T} s^T(t)\tilde{F}_B(t) \\ &+ \sum_{t=1}^{T} \Big[-\frac{1}{2} s^T(t)\tilde{\Lambda}^{-1}s(t) - \frac{1}{2} s^T(t-1)\tilde{F}^T\tilde{\Lambda}^{-1}\tilde{F}s(t-1) + s^T(t)\tilde{\Lambda}^{-1}\tilde{F}s(t-1)\Big] \\ &+ \text{const} \end{aligned} \tag{5.68}$$

In (5.68), the first two terms correspond to the emission equation and the third term corresponds to the state transition equation.

We set the value of $\tilde{\Lambda}, \tilde{F}, \tilde{F}_A(t), \tilde{F}_B(t)$ as follows:

$$\tilde{\Lambda} = \Big\langle \frac{1}{1-F^2} \Big\rangle^{-1} \tag{5.69}$$

$$\tilde{F} = \Big\langle \frac{F}{1-F^2} \Big\rangle \Big\langle \frac{1}{1-F^2} \Big\rangle^{-1} \tag{5.70}$$

$$\tilde{F}_A(t) = \begin{cases} \sum_{i=1}^{m}\sum_{k=0}^{K} \Big\langle h_i h_i^T \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} \Big\langle I_i^{(k)}(t-k) \Big\rangle + \sum_{j=1}^{l} \Big\langle u_j u_j^T \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1} \\ \qquad - \tilde{F}\tilde{\Lambda}^{-1}\tilde{F} + \Big\langle \frac{F^2}{1-F^2} \Big\rangle, t = 1:T-1 \\ \sum_{i=1}^{m}\sum_{k=0}^{K} \Big\langle h_i h_i^T \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} \Big\langle I_i^{(k)}(t-k) \Big\rangle + \sum_{j=1}^{l} \Big\langle u_j u_j^T \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1}, t = T \end{cases} \tag{5.71}$$

$$\tilde{F}_B(t) = \sum_{i=1}^{m}\sum_{k=0}^{K} \Big\langle h_i \Big\rangle \Big\langle \sigma_i^2 \Big\rangle^{-1} \Big\langle I_i^{(k)}(t-k) \Big\rangle \Big\langle X_i(t-k) \Big\rangle + \sum_{j=1}^{l} \Big\langle u_j \Big\rangle \Big\langle \gamma_j^2 \Big\rangle^{-1} \Big\langle y_j(t) \Big\rangle, \quad t = K+1:T \tag{5.72}$$

Set $I_i^{(k)}(t-k) = 0$, if $t - k \leq 0$ in (5.71) and (5.72). Substituting equations (5.69) $\sim$ (5.72) to (5.68), then we can get the optimal solution of the approximate posterior of s in (5.67).

(1) Filtering step

With the augmented parameters in (5.69) $\sim$ (5.72), we assume the augmented

LGSSM has following form:

$$\begin{cases} s(t) = \tilde{F}s(t-1) + \tilde{e}_s(t), \tilde{e}_s(t) \sim \mathcal{N}(\mathbf{0}, \tilde{\Lambda}) \\ D(t) = \tilde{H}s(t) + \tilde{e}_d(t), \tilde{e}_d(t) \sim \mathcal{N}(\mathbf{0}, \tilde{\Sigma}) \end{cases} \tag{5.73}$$

where D represents the observations including both inputs and outputs. The definition of forward and backward path is:

$$p(s(t)|D(1:t), \theta^{old}) = \mathcal{N}(\mu(t), V(t)) \tag{5.74}$$

$$p(s(t)|D(1:t-1), \theta^{old}) = \mathcal{N}(\mu_t^{t-1}, V_t^{t-1}) \tag{5.75}$$

$$p(s(t)|D(1:T), \theta^{old}) = \mathcal{N}(\hat{\mu}(t), \hat{V}(t)) \tag{5.76}$$

where, μ_t^{t-1}, V_t^{t-1} are the mean and variance of the Gaussian distribution in the prediction step (from time $t-1$ to t). $\mu(t), V(t)$ are the mean and variance of the posterior distribution in filtering step. $\hat{\mu}(t), \hat{V}(t)$ are mean and variance of the posterior distribution in smoothing step. In order to calculate $\mu(t), V(t)$, we derive the logarithm posterior distribution as:

$$\begin{aligned}
&\ln p(s(t)|D(1:t)) \\
=&\ln p(s(t)|D(1:t-1), D(t)) \\
=&\ln p(D(t)|s(t), D(1:t-1)) + \ln p(s(t)|D(1:t-1)) + \text{const} \\
=&\ln p(D(t)|s(t)) + \ln p(s(t)|D(1:t-1)) + \text{const} \\
=&-\frac{1}{2}\Big[D(t) - \tilde{H}s(t)\Big]^T \tilde{\Sigma}^{-1}\Big[D(t) - \tilde{H}s(t)\Big] \\
&-\frac{1}{2}\Big[s(t) - \tilde{F}s(t-1)\Big]^T (V_t^{t-1})^{-1}\Big[s(t) - \tilde{F}s(t-1)\Big] + \text{const} \\
=&-\frac{1}{2}s^T(t)\Big(\tilde{H}\tilde{\Sigma}^{-1}\tilde{H}\Big)s(t) + s^T(t)\Big(\tilde{H}^T\tilde{\Sigma}^{-1}D(t)\Big) \\
&-\frac{1}{2}\Big[s(t) - \tilde{F}s(t-1)\Big]^T (V_t^{t-1})^{-1}\Big[s(t) - \tilde{F}s(t-1)\Big] + \text{const} \qquad (5.77) \\
=&-\frac{1}{2}s^T(t)\Big(\tilde{H}\tilde{\Sigma}^{-1}\tilde{H} + (V_t^{t-1})^{-1}\Big)s(t) \\
&+ s^T(t)\Big(\tilde{H}^T\tilde{\Sigma}^{-1}D(t) + (V_t^{t-1})^{-1}\tilde{F}s(t-1)\Big) + \text{const} \qquad (5.78)
\end{aligned}$$

Since $p(s(t)|D(1:t), \theta^{old}) = \mathcal{N}(\mu(t), V(t))$, the logarithm posterior distribution can also be written as:

$$\ln p(s(t)|D(1:t)) = -\frac{1}{2}s^T(t)V(t)^{-1}s(t) + s^T(t)\Big(V(t)^{-1}\mu(t)\Big) + \text{const} \tag{5.79}$$

Comparing (5.78) and (5.79), we can get:

$$V(t) = \left(\tilde{H}^T\tilde{\Sigma}^{-1}\tilde{H} + (V_t^{t-1})^{-1}\right)^{-1} \tag{5.80}$$

$$\mu(t) = V(t) \cdot \left(\tilde{H}^T\tilde{\Sigma}^{-1}D(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\right) \tag{5.81}$$

Since $\tilde{H}$ and $\tilde{\Sigma}$ have no explicit expression in the augmented system (5.73), we can get rid of them by comparing the first two terms in equations (5.68) and (5.77), respectively and obtain:

$$\tilde{F}_A(t) = \tilde{H}^T\tilde{\Sigma}^{-1}\tilde{H} \tag{5.82}$$

$$\tilde{F}_B(t) = \tilde{H}^T\tilde{\Sigma}^{-1}D(t) \tag{5.83}$$

Substituting (5.82) and (5.83) into (5.80) and (5.81) yields:

$$V(t) = \left[\tilde{F}_A(t) + (V_t^{t-1})^{-1}\right]^{-1} \tag{5.84}$$

$$\mu(t) = V(t) \cdot \left[\tilde{F}_B(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\right] \tag{5.85}$$

Above two equations compose the update step in Kalman filtering for the augmented system (5.73), in which all the needed items are calculated in (5.70) $\sim$ (5.72) and following prediction equations:

$$\mu_t^{t-1} = \tilde{F} \cdot \mu(t-1) \tag{5.86}$$

$$V_t^{t-1} = \tilde{F} \cdot V(t-1) \cdot \tilde{F}^T + \tilde{\Lambda} \tag{5.87}$$

(2) Smoothing step

The augmented system can use the standard Kalman smoothing procedure as below:

$$\hat{\mu}(t) = \mu(t) + J(t)\left[\hat{\mu}(t+1) - \tilde{F}\mu(t)\right] \tag{5.88}$$

$$\hat{V}(t) = V(t) + J(t)\left[\hat{V}(t+1) - V_{t+1}^t\right]J^T(t) \tag{5.89}$$

$$J(t) = V(t)\tilde{F}^T(V_{t+1}^t)^{-1} \tag{5.90}$$

with initializations:

$$\hat{\mu}(T) = \mu(T) \tag{5.91}$$

$$\hat{V}(T) = V(T) \tag{5.92}$$

In summary, the equations (5.84) $\sim$ (5.90) compose the complete Kalman filtering and smoothing steps and we can calculate the following sufficient statistics:

$$\langle s(t)\rangle = \hat{\mu}(t) \tag{5.93}$$

$$\langle s(t)s(t)\rangle = \hat{V}(t) + \hat{\mu}(t)\hat{\mu}^T(t) \tag{5.94}$$

$$\langle s(t+1)s(t)\rangle = J(t)\hat{V}(t+1) + \hat{\mu}(t+1)\hat{\mu}^T(t) \tag{5.95}$$

5.2.2.4 Inference of λ_j: Importance Sampling

Since the prior of each λ_j is of Beta distribution and it is not conjugate to the likelihood of s, Gaussian distribution, the target posterior of λ_j cannot be derived analytically. Importance sampling method can be employed to solve this problem. According to the general solution, the optimal posterior of λ_j is:

$$\begin{aligned}
\ln q^*_{\lambda_i} \propto & \mathbb{E}_{q_{\setminus\lambda_i}(Z_{\setminus\lambda_i})}\big[\ln p(X,Y,Z)\big] \\
= & \Big\langle \ln p(s_i|\lambda_i)p(\lambda_i|\alpha_{\lambda_0},\beta_{\lambda_0})\Big\rangle + \text{const} \\
= & \Big\langle \ln p(s_i|\lambda_i)\Big\rangle \cdot p(\lambda_i|\alpha_{\lambda_0},\beta_{\lambda_0}) + \text{const}
\end{aligned} \tag{5.96}$$

And, the expectation of likelihood function can be derived starting from initial distribution of $s_j \sim \mathcal{N}(0,1)$:

$$\begin{aligned}
& \Big\langle \ln p(s_j|\lambda_j)\Big\rangle \\
= & \Big\langle \ln \mathcal{N}(s_j(1)|0,1) \cdot \sum_{t=2}^{T} \mathcal{N}\big(s_j(t)|\lambda_j s_j(t-1), 1-\lambda_j^2\big)\Big\rangle \\
= & \Big\langle -\frac{1}{2}\ln 2\pi - \frac{1}{2}s^2(1) - \frac{1}{2}\sum_{t=2}^{T}\big[\ln 2\pi + \ln(1-\lambda_j^2)\big] + \big[s_j^2(t) - \lambda_j s_j^2(t-1)\big]^2 \cdot \frac{1}{1-\lambda_j^2}\Big\rangle \\
= & -\frac{T}{2}\ln\pi - \frac{T-1}{2}\ln(1-\lambda_j^2) - \frac{1}{2}\Big\langle s^2(1)\Big\rangle \\
& - \frac{1}{2}\Big\langle \sum_{t=2}^{T} s_j^2(t)\Big\rangle \frac{1}{1-\lambda_j^2} + \Big\langle \sum_{t=2}^{T} s_j(t-1)s_j(t)\Big\rangle \frac{\lambda_j}{1-\lambda_j^2} - \frac{1}{2}\Big\langle \sum_{t=1}^{T-1} s_j^2(t)\Big\rangle \frac{\lambda_j^2}{1-\lambda_j^2}
\end{aligned} \tag{5.97}$$

In the importance sampling method, let us assume the target distribution is $p(x)$ and the sampling distribution is $q(x)$, then the sample weights can be calculated as $w(x) = \frac{p(x)}{q(x)}$. In this model, the sampling distribution is chosen to be the same as prior distribution $Beta(\alpha_{\lambda_0}, \beta_{\lambda_0})$, so the weights of each sample can be calculated as

the likelihood in (5.97). Then the three needed statistics of the posterior of λ_j in (5.69)$\sim$(5.71) can be derived using the value of each sample:

$$\left\langle \frac{1}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{1}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{5.98}$$

$$\left\langle \frac{\lambda_j}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{\lambda_j^{(n)}}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{5.99}$$

$$\left\langle \frac{\lambda_j^2}{1-\lambda_j^2} \right\rangle = \sum_{n=1}^{N} \frac{\left(\lambda_j^{(n)}\right)^2}{1-\left(\lambda_j^{(n)}\right)^2} \cdot w\left(\lambda_j^{(n)}\right) \tag{5.100}$$

where $\lambda_j^{(n)}$ is the n-th sample drawn from sampling distribution and N is the total number of samples. When $N \to \infty$, the expectation values in (5.98) $\sim$(5.100) will approximate the corresponding statistics of the optimal posterior in (5.96).

5.2.3 On-line Prediction Using the Model

In on-line implementation, future observations are not available. Thus Kalman smoothing step will not be used. If partial measurements of the target output are available, e.g. in the case of lab samples, only available samples can be used in the filtering step to obtain the latent feature s. One step ahead prediction of $s(t)$ can be obtained through the Kalman filter recursions:

$$\hat{s}(t) = \mu(t) = V(t) \cdot \left[\tilde{F}_B(t) + (V_t^{t-1})^{-1}\tilde{F}\mu(t-1)\right] \tag{5.101}$$

Since $Y(t)$ and all future y are not available, only the first term of $\tilde{F}_B$ in (5.72) can be calculated. That is to say, when $Y(t)$ is available, $s(t)$ is estimated from both $X(t-K:t)$ and $Y(t)$. In contrast, when $Y(t)$ is not available, $s(t)$ is only estimated from $X(t-K:t)$ until next slow sample of Y is available. After calculating $\hat{s}(t)$, the prediction of unsampled output $\hat{Y}(t)$ can be estimated as follows

$$\hat{Y}(t) = U\hat{s}(t) + e_y(t) \tag{5.102}$$

Given that $e_y(t)$ has zero mean, $Y(t)$ is evaluated as:

$$Y(t) = \text{mean}\{\hat{Y}(t)\} = U\hat{s}(t) \tag{5.103}$$

5.3 Applications

In this section, the prediction ability of the proposed method is demonstrated with a simulation study and an industrial application. First, a numerical example is utilized to illustrate the prediction ability in two scenarios: output with no missing data and with multi-rate samples. Second, application to a SAGD process is conducted to demonstrate the prediction ability of well pair water content in both scenarios: missing data and in presence of a slow-sampled quality variable. In both simulation and industrial application studies, the proposed method is compared with the IOPSFA algorithm to demonstrate the performance improvement by considering the time delay, and compared with the case that only considers the fixed time delay to illustrate the benefits by considering the time-varying time delay. In addition, in the numerical example, the proposed IOPSFA_InVTD method is also compared with the IOPSFA_VTD to demonstrate the benefits by considering different time delays for different process variables.

5.3.1 Numerical Case Study

In this case study, the following linear state space model is considered

$$\begin{cases} s(t) = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.8 \end{bmatrix} s(t-1) + e_s(t), e_s(t) \sim \mathcal{N}(0, \begin{bmatrix} 1-0.95^2 & 0 \\ 0 & 1-0.8^2 \end{bmatrix}) \\ X(t) = \begin{bmatrix} 0.3 & 0.4 \\ 0.35 & -0.3 \\ -0.2 & -0.65 \end{bmatrix} s(t) + e_x(t), e_x(t) \sim \mathcal{N}(0, \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.06 \end{bmatrix}) \\ Y(t) = \begin{bmatrix} 2 & -1 \end{bmatrix} s(t) + e_y(t), e_y(t) \sim \mathcal{N}(0, 0.3) \end{cases} \tag{5.104}$$

In this example, we use the one-dimension output as an example. As mentioned before, if more than one quality variable needs to be inferred, they can be decomposed into multiple one-dimensional output as shown in (5.104). We assume that the maximum time delay $K = 3$ for simplicity of illustration, so the time delay transition

matrix M_i for each process variable X_i can be constructed by a 4×4 matrix in (5.105)

$$
\begin{aligned}
M_1 &= \begin{bmatrix} 0.95 & 0.02 & 0.02 & 0.01 \\ 0.02 & 0.95 & 0.02 & 0.01 \\ 0.0 & 0.02 & 0.96 & 0.02 \\ 0.01 & 0.02 & 0.02 & 0.95 \end{bmatrix}, \\
M_2 &= \begin{bmatrix} 0.98 & 0.01 & 0.01 & 0.00 \\ 0.01 & 0.97 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.95 & 0.03 \\ 0.01 & 0.01 & 0.02 & 0.96 \end{bmatrix}, \\
M_3 &= \begin{bmatrix} 0.95 & 0.02 & 0.02 & 0.01 \\ 0.02 & 0.95 & 0.01 & 0.02 \\ 0.00 & 0.01 & 0.97 & 0.02 \\ 0.00 & 0.01 & 0.01 & 0.98 \end{bmatrix}
\end{aligned}
\tag{5.105}
$$

in which, for the practical reason, the diagonal elements are larger than the other elements in the same row. To generate data, first, the two-dimensional latent features are generated according to the given λ: $\lambda_1 = 0.95$ and $\lambda_2 = 0.8$ in Figure 5.5. Then input and output data are generated according to the emission equations in Figure 5.6. In order to generate the delayed inputs, time delay sequences, shown in Figure 5.7, are generated using the Markov transition matrix M_i for each process input X_i. At last, the delayed input data can be determined by shifting each sample with the values according to the generated delay sequence. The generated data consist of 6,000 samples. The first 3000 samples are used for training and the last 3000 samples for validation.

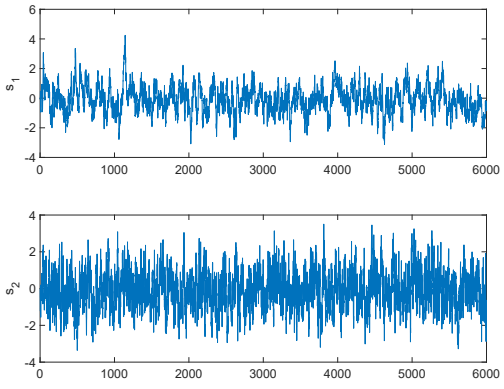

Figure 5.5: Simulated slow features s

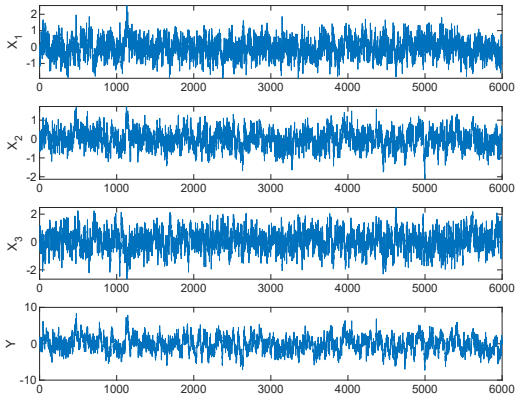

Figure 5.6: Simulated inputs X and ouyput Y

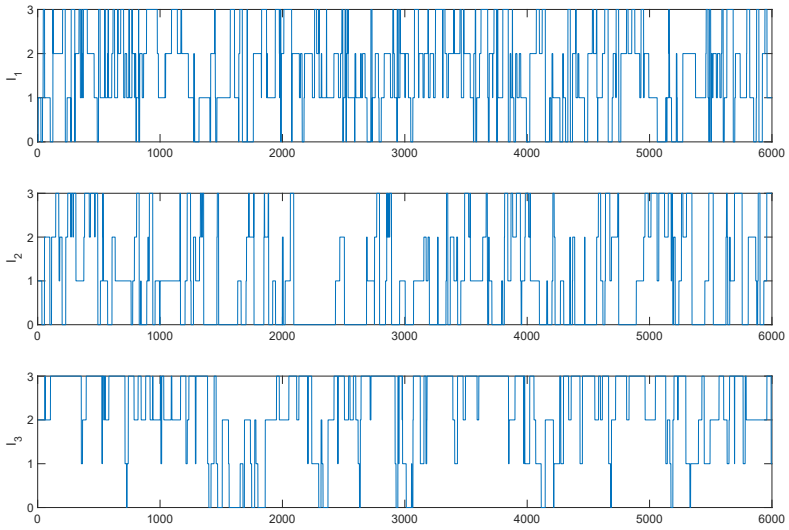

Figure 5.7: Simulated delay sequence I

Next, we provide the details of the modeling process and illustrate the performance through two cases: 1) no missing output data; 2) multi-rate sampling.

Case 1: No missing output data

In this case, we assume all output observations are available and slow features are extracted from non-missing observations of both inputs and outputs. This corresponds to the scenario that an accurate on-line analyzer is installed in the plant to measure the quality variable, e.g. VX Meter installed to measure water content in Steam-assisted gravity drainage (SAGD) process [128]. The developed soft sensor model will be useful when the on-line analyzer is out of service, i.e. damaged, under maintenance or becoming inaccurate due to long time service or harsh operation conditions, etc. We will use MAE to compare the difference between the predicted output $\hat{Y}$ and observed output Y.

To illustrate the performance of the developed model, first, we compare the extracted SFs from IOPSFA_InVTD and IOPSFA_VTD to demonstrate their ability to extract the SFs from data and the performance is improved for IOPSFA_InVTD when different time delays in different process variables are considered. Figure 5.8

shows the extracted SFs by IOPSFA_InVTD and IOPSFA_VTD. The blue dashed line in each sub-figure is the real SF that we have generated through the simulated model in (5.104) and the red line is the extracted SF. The MAE for the extracted SFs comparing with real SFs is calculated and labeled in the corresponding sub-figure. As we can see, the MAE for the SFs extracted by IOPSFA_InVTD is smaller than SFs extracted by IOPSFA. It means SFs extracted by IOPSFA_InVTD is closer to the real SFs.

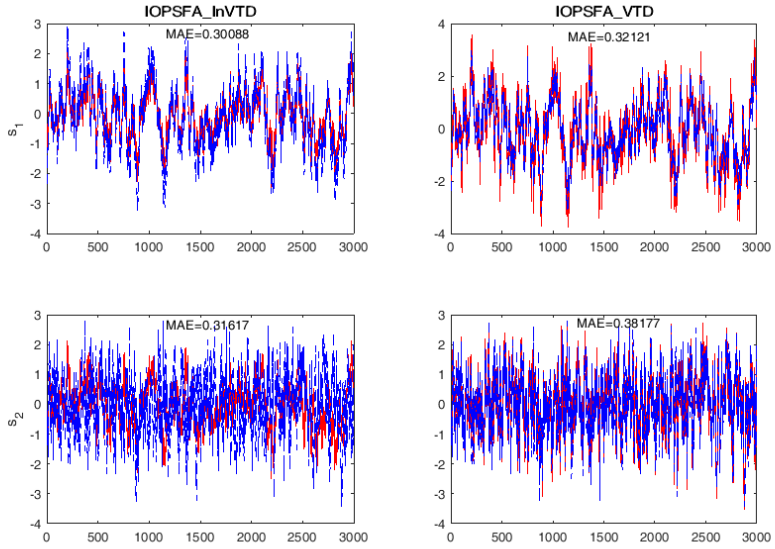

Figure 5.8: Comparison of SFs extracted by IOPSFA_InVTD and IOSPFA_VTD

The model learned by IOPSFA_InVTD is as follows

$$
\begin{aligned}
F &= \begin{bmatrix} 0.9127 & 0 \\ 0 & 0.8426 \end{bmatrix}, H = \begin{bmatrix} 0.2231 & 0.3635 \\ 0.3635 & -0.2089 \\ -0.1551 & -0.5642 \end{bmatrix} \\
U &= \begin{bmatrix} 1.9057 & -0.9667 \end{bmatrix} \\
\Sigma &= \begin{bmatrix} 0.0872 & 0 & 0 \\ 0 & 0.0454 & 0 \\ 0 & 0 & 0.0577 \end{bmatrix}, \Gamma = 0.2830
\end{aligned} \tag{5.106}
$$

Next, with the above model, we compare prediction performance of the IOPSFA_InVTD, IOSFPA_VTD and IOPSFA_InVTD in the fixed time delay cases to demonstrate the performance improvement when considering time-varying time delays for different process variables. In the fixed time delay cases of IOPSFA_InVTD, it is assumed that

the time delays are constant and same across different process variables, e.g. $X_{1:m}$ all have time delay 1 and it corresponds to the fixed time delay case $d = 1$. A part of prediction results of output is shown in Figure 5.9.

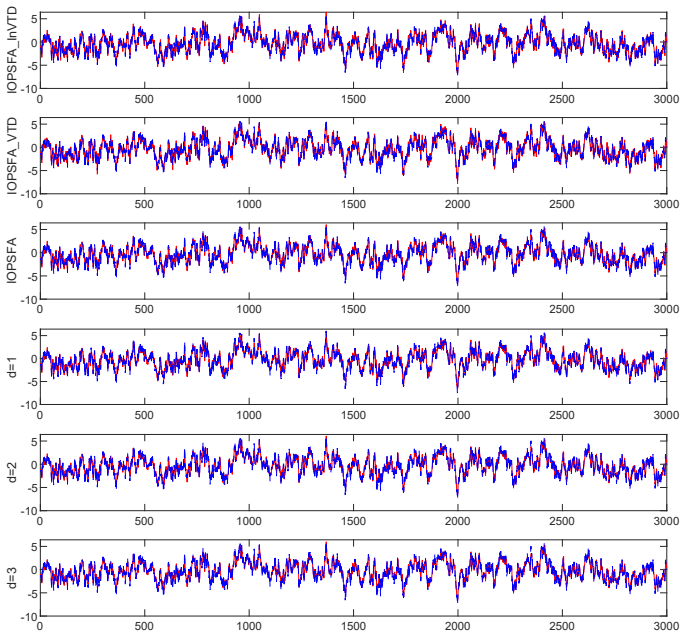

Figure 5.9: Prediction trends without missing outputs

The blue dashed lines represent the actual measurements and the red lines are the predicted values. The first sub-figure is the prediction results for IOPSFA_InVTD, the second one is for IOSPFA_VTD and the rest of the sub-figures correspond to the fixed time delay cases, i.e. time delay is fixed as 1, 2 and 3, respectively. MAE is calculated for the prediction of each method and summarized in Table 5.1.

Table 5.1: Prediction results without missing outputs

	IOPSFA_InVTD	IOPSFA_VTD	IOSPFA	delay=1	delay=2	delay=3
MAE	0.3718	0.3782	0.4133	0.3952	0.4052	0.4262

The above results show that IOPSFA_InVTD approach produces the smallest MAE comparing with other methods, i.e. IOSPFA_VTD and IOPSFA. It means when different time-varying time delays are considered for different process variables, the performance is better than just considering time-varying time delay for output (IOPSFA_VTD).

But IOSPFA_VTD is still better than just considering the fixed time delay cases, like IOPSFA that corresponds to no consideration of time delay case and $d = 1, 2, 3$ cases. This makes sense because when different assumptions are made on different cases, the further the assumption deviates from the real condition (the real condition in this case study is known to us since it is a simulated system), the worse results are expected.

Case 2: Multi-rate case

In this case, we also adopt the idea to handle the multi-rate problem similar to IOPSFA [128]. In this case study, we down-sample the output to simulate the multi-rate scenario. The down-sampling ratio is 10, that is to say, one sample is kept for every 10 samples. IOPSFA_InVTD can provide on-line prediction when the output is not available and it can also update the extracted slow features whenever the output sample is available. The estimated model parameters are as follows

$$\begin{aligned} F &= \begin{bmatrix} 0.9058 & 0 \\ 0 & 0.8278 \end{bmatrix}, H = \begin{bmatrix} 0.2490 & 0.3389 \\ 0.3326 & -0.2551 \\ -0.1792 & -0.5462 \end{bmatrix} \\ U &= \begin{bmatrix} 1.7035 & -0.8654 \end{bmatrix} \\ \Sigma &= \begin{bmatrix} 0.1301 & 0 & 0 \\ 0 & 0.0424 & 0 \\ 0 & 0 & 0.0532 \end{bmatrix}, \Gamma = 0.0385 \end{aligned} \tag{5.107}$$

The sample prediction trends for IOPSFA_InVTD and the comparative methods are shown in Figure 5.10. The prediction performance in terms of MAE is summarized in Table 5.2. The predicted trends in Figure 5.10 show that all approaches can catch the trend of the real output. However, IOPSFA_InVTD has the best results among all methods. We can also notice that the MAE in multi-rate cases is larger than the corresponding no missing data cases. This is because in the training phase, the output data is not always available. Fewer output samples would contain less information, which leads to less accuracy in predicting the output.

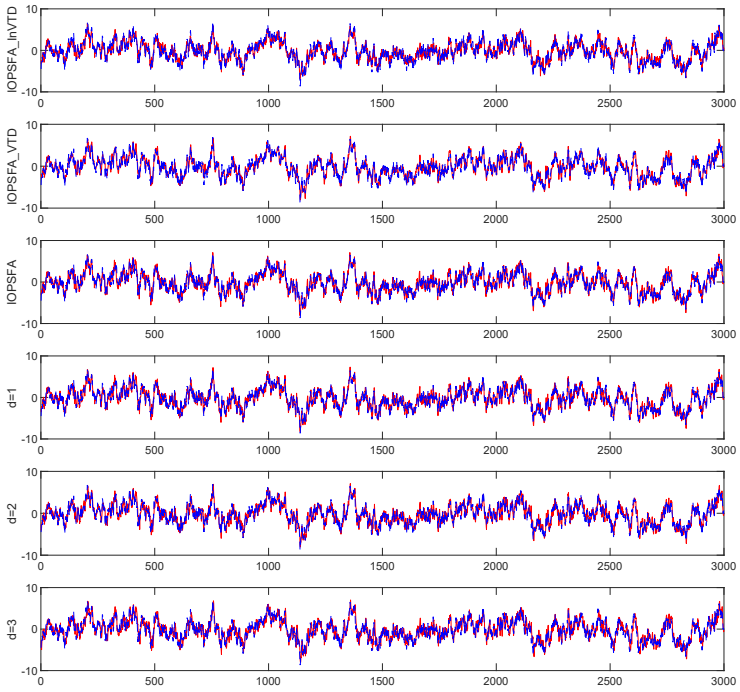

Figure 5.10: Prediction trends with missing outputs, down-sample rate=10

Table 5.2: Prediction results with missing outputs, down-sample rate=10

	IOPSFA_InVTD	IOPSFA_VTD	IOSPFA	delay=1	delay=2	delay=3
MAE	0.6108	0.7059	0.8385	0.7329	0.7894	0.9465

5.3.2 Industrial Case Study: SAGD Process Well Pair Water Content Soft Sensor Design

In this section, we also employ industrial data from a SAGD process to illustrate the practicality of the proposed algorithm. As we introduced earlier, SAGD process is an innovative in-situ oil recovery technology to extract heavy oil or bitumen from oil sands that are buried deep in underground [123, 124]. Figure 5.11 shows one typical well pair for oil extraction section of SAGD process and illustrates how emulsion, mixture of oil, water and gas, is extracted from underground. For each well pair, two horizontal wells are drilled into the underground. The upper well, i.e. injection well, is used to inject high temperature and pressure steam to soften the oil sands. This results in the formulation of oil-water emulsion which is flowable and transmissible.

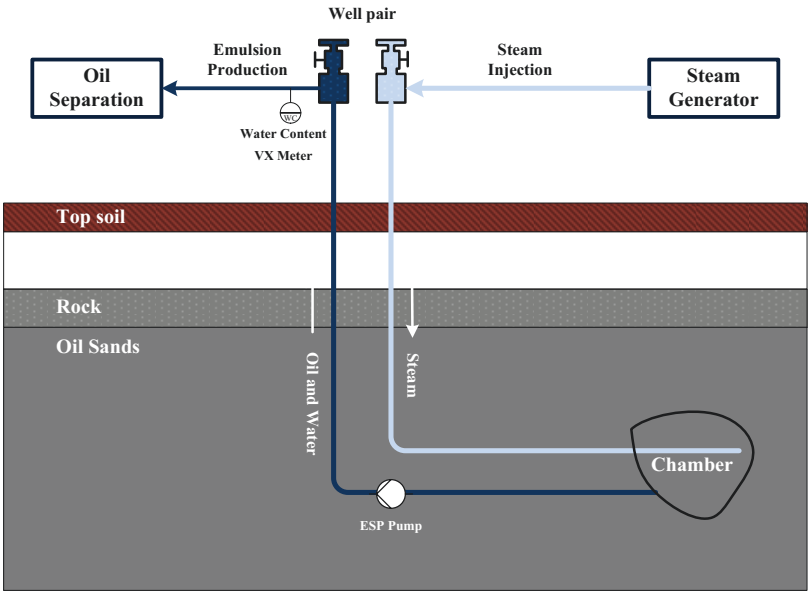

Figure 5.11: SAGD process well pair diagram

The lower well, i.e. production well, is used to pump out the heated emulsion from the underground chamber. The outlet emulsion contains a few gas components and a lot of water due to the condensation of the injected steam. The composition of the emulsion, especially the water content, is an important variable that determines the amount of chemicals that need to be injected in the downstream process in order to produce oil that meets the specifications. On-line measurement of water content is possible by using an instrument called VX meter that is costly and hence cannot be installed for all well pairs due to economic considerations. Another way to accurately measure the water content in outlet emulsion is to use a test separator, which is used to separate the emulsion into liquid and vapor components, and it has the ability to measure the water content continuously and accurately. There is normally one test separator installed for the whole well pad, which contains several well pairs, so it rotates between different well pairs and tests emulsion from one well pair at a time. Thus, the measurement of water content for one specific well pair is only available for a limit time. All these factors lead to the necessity to develop a soft sensor for estimating water content in real time.

The first four sub-figures in Figure 5.12 are four selected influential input variables by performing correlation analysis and the last subfigure shows the profile of the

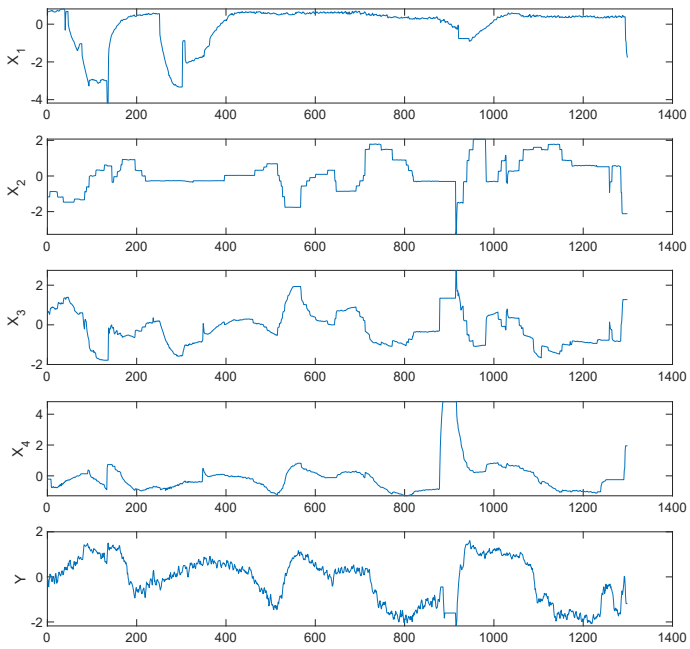

Figure 5.12: Well pair water content measurements of X and Y

quality variable, which is water content. The raw data of water content and selected influential variables are sampled and stored every 10 min. After data pre-processing they are 3-hour averaged values and used for further analysis. The data set includes 1299 samples in total wherein the first 711 samples are used for training and the last 588 samples for validation. For proprietary reasons, the attributes of all variables are not disclosed and all data have been normalized. In the following sub-section, we provide the details of predictive model development by using SFs extracted from all inputs and output considering the following two cases: 1) no missing output; 2) multi-rate case and with down-sample ratio 4, which is equivalent to the scenario that one accurate water content measurement sample is available for every 12 hours. In both cases, the maximum delay is considered as 3.

Case 1: No missing output data

In this case, we do not consider any missing data in outputs and all measurement

values of inputs and outputs are available for model building. Since the plant has one VX Meter installed for the well pair, we can obtain accurate and real-time measurements for the water content. So water content has the same sampling rate as other influential variables. The developed soft sensor model will be useful when theVX meter is out of service, i.e. damaged, under maintenance or becoming inaccurate due to long time service or harsh operation conditions, etc. MAE is used to compare the difference between the predicted output $\hat{Y}$ and observed output Y. The comparison is conducted between proposed IOPSFA_InVTD, IOPSFA and fixed time delay cases using IOPSFA_InVTD, i.e. d=1, 2, and 3. Figure 5.13 shows the prediction results for all cases and performance in terms of MAE is summarized in Table 5.3.

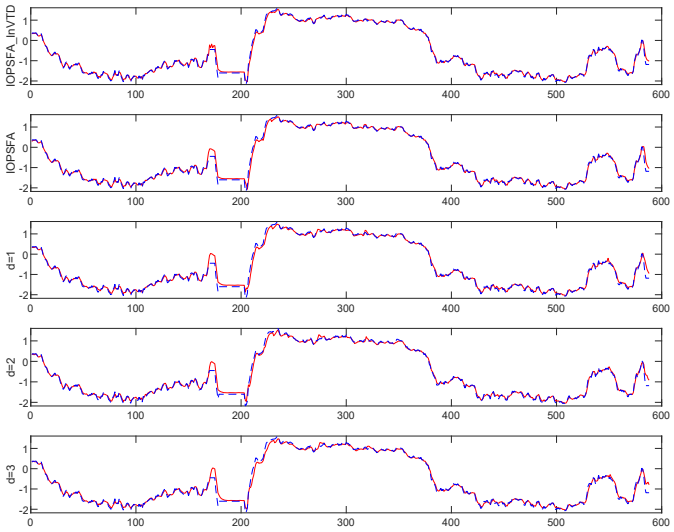

Figure 5.13: Well pair water content: prediction trends without missing outputs

Table 5.3: Well pair water content: prediction results without missing outputs

	IOPSFA_InVTD	IOPSFA	delay=1	delay=2	delay=3
MAE	0.0445	0.0673	0.0644	0.0572	0.0660
Improvement(%)	-	33.88%	30.90%	22.20%	32.58%

In the above figure, the blue dashed lines are the real outputs as references and the red lines are the predicted values. The performance indices and the performance improvement of IOPSFA_InVTD in comparison with other algorithms in terms of MAE are shown in Table 5.3. As we can see, the proposed method has the best performance

comparing to other method or cases in terms of MAE. The performance increases at least by 20%. Also it is worth noticing that, IOPSFA has worse performance than the fixed delay cases. It means although considering all the input variables to have the same fixed time delay does not comply with the practical situation, it is still better than not considering time delay at all, as in IOPSFA. The following stacked Figure 5.14 illustrates the time delay transition of each input variable. At each time instant, the value bar is composed of four probability values, which correspond to the probability that the time delay equals 0,1,2,3 at this time instant, respectively. Different colors are used to represent different time delays and the sum of these four probability values is 1.

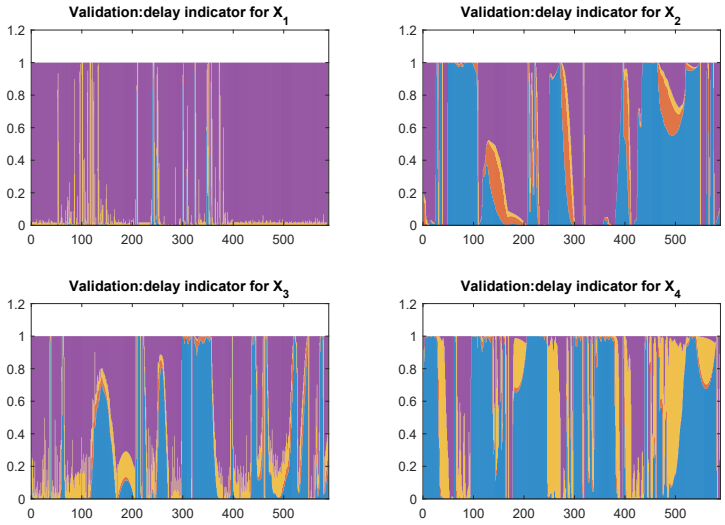

Figure 5.14: Well pair water content: estimated time delays for X

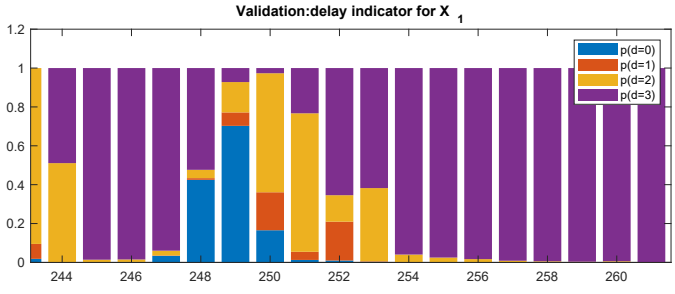

Figure 5.15: Well pair water content: estimated time delays for X_1

To make it clearer to readers, a zoom-in illustration is given in Figure 5.15. At

each time instant, if we take the one with the largest probability as the time delay, then we can see from Figure 5.15 that: at $t = 245 \sim 248$, time delay is 3 and transits to $d = 0$ at $t = 249$, then to $d = 2$ at $t = 250, 251$, and then to $d = 3$ again from $t = 252$ and remains there. This reveals the possible varying time delay in practical situation and under this situation, the predicted water content achieves the smallest MAE among all comparative methods as shown in Table 5.3.

Case 2: Multi-rate case

In this case, we simulate the scenario in industrial settings in which the quality variable has larger sampling interval than other process variables. This corresponds to the situation that installation of an on-line analyzer, like VX meter for each well pair is not feasible due to economical reason or harsh condition on site. The samples need to be collected manually periodically and analyzed in the lab. In this case, we adopt the similar idea to handle the multi-rate problem as IOPSFA [128]. In this case study, the quality variable is down-sampled to simulate the multi-rate scenario. The down-sampling ratio is 4, considering that the 3-hour time average data is used, which means the accurate water content measurement sample is available for every 12 hours. The maximum possible time delay is also set to 3. The predicted trends of all comparing methods are shown in Figure 5.16. The blue dashed lines represent the fast rate samples and MAE is also calculated using these fast rate samples. The red lines are the predicted trends. As we can see, all methods can catch the trend of water content represented by multi-rate sampling lab data. The prediction performance is still measured using MAE, which are calculated and summarized in 5.4. From the above results, we can see that IOPSFA_InVTD outperforms other algorithms since it considers time-varying delays in its modeling while the algorithm with the fixed delay = 3 gives the worst performance.

Table 5.4: Well pair water content: prediction results with missing outputs, down-sample rate=10

	IOPSFA_InVTD	IOPSFA	delay=1	delay=2	delay=3
MAE	0.0484	0.0696	0.0668	0.0773	0.0816
Improvement(%)	-	30.46%	27.54%	37.39%	40.69%

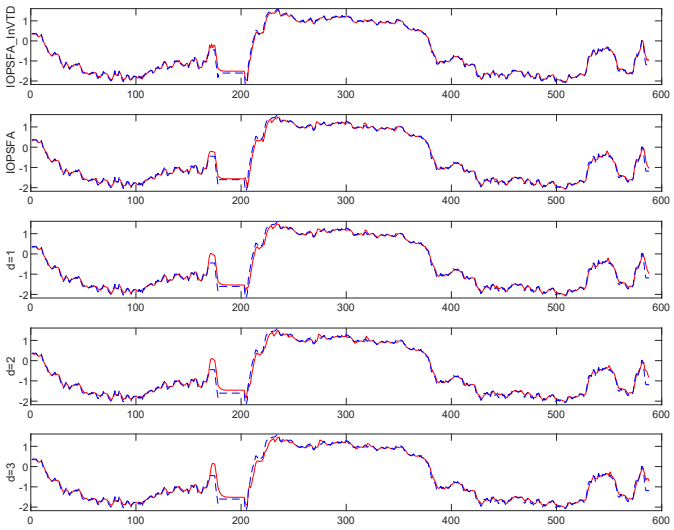

Figure 5.16: Well pair water content: prediction trends with missing outputs, down-sample rate=10

The stacked plot of time delay transition is given in Figure 5.17, which presents a similar pattern as in no missing output case with some differences. For example, in multi-rate case, the time delays estimated for X_4 present more occurrences of $d = 3$ and less occurrences of $d = 2$ than no missing output data case. This is probably caused by the multi-rate sampled output because we cannot get the lab data timely. It can only use the nearest available reference, which is later than in the no missing output case, to estimate time delay.

5.4 Conclusions

In this work, an enhanced approach based on IOSPFA, termed as IOPSFA_InVTD, is proposed by considering the inputs time-varying time delays which are different among different process variables. Comparing with IOPSFA_VTD in chapter 4, IOPSFA_InVTD is a more general method and it has the ability to address the time delay problem caused by the scattered locations where each process variable measurement device resides. Under the variational Bayesian framework, dynamic latent features can be extracted using delayed process variables and quality variable. The extracted latent features have better prediction ability than the case that only

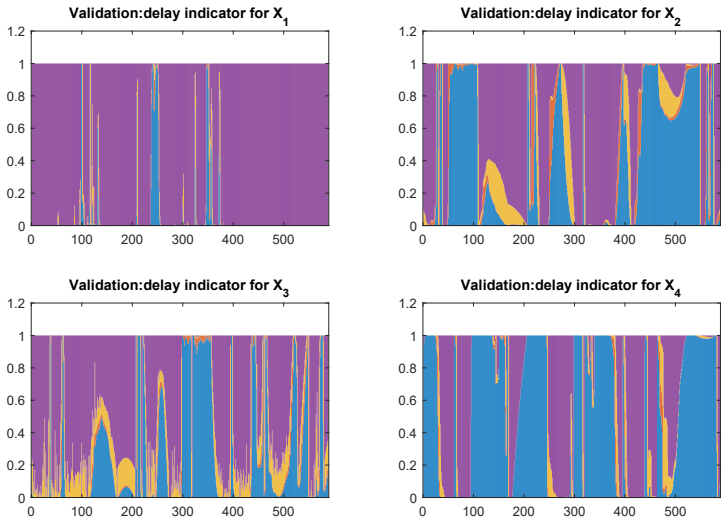

Figure 5.17: Well pair water content: estimated time delays for X, down-sample rate=10

considering fixed time delays and simultaneous varying time delays. The performance results are validated through a simulated numerical example along with an industrial application.

Chapter 6

Conclusions and Future work

In this chapter, the conclusions of this thesis are provided for the preceding chapters of this thesis. Furthermore, possible studies of future research are also discussed.

6.1 Conclusions

The main topic of this thesis is inferential modeling by extracting dynamic latent features from process data with various irregularities. The proposed solutions are solved under the Maximum Likelihood estimation and variational Bayesian framework. The inferential models are learned with the extracted dynamic latent features in presence of various uncertainties separately or combined. The advantages of the developed models have been demonstrated through multiple simulations, experiments and industrial applications.

In Chapter 2, a robust PSFA method has been proposed for the modeling of the dynamics and high dimensional data which contains outliers. The Student's t-distribution is utilized to address the outliers since its heavier tails increase the weight of the measurements beyond the normal range. EM algorithm is employed to estimate the parameters in RPSFA formulation, in which a variance scale is introduced as a hidden variable. A weighted gain Kalman filter technique is proposed to compensate the Kalman gain due to the non-Gaussian assumption of measurement noise. Slower and smoother latent features can be extracted using the proposed RPSFA comparing with conventional PSFA algorithm. The effectiveness of the proposed approach is illustrated using TE benchmark process, an industrial application and an experimental case study.

In Chapter 3, an enhanced approach, termed as IOPSFA, to extract dynamic latent features is proposed. The conventional PSFA can only extract latent features from input process variables while not considering the past measurements of quality variables that need to be predicted. IOSPFA, by contract, is able to extract dynamic latent features that incorporate information from quality variables, providing a better prediction performance to the quality variables. The latent features extracted using IOPSFA algorithm are more interpretable for the intrinsic properties of the process, thus leading to a better prediction of the quality variables. In the procedure of building an inferential model using IOSPFA, randomly missing and multi-rate sampling of quality variable are considered. It is proved that IOPSFA algorithm is robust to a wild range of missing data and can be applied to different types of missing data scenarios. The validity and performance improvement of the proposed approach are demonstrated through a SAGD well pair water content soft sensor application and a tank system experiment.

In Chapter 4, we consider another common problem in process industries, time delay. Instead of identifying a fixed time delay, an improved IOPSFA-based approach that can address time-varying time delays in quality variables is proposed. By describing the time delay sequence as a dynamic feature, reconstructed observation functions of quality variables are introduced in the IOPSFA formulation, as proposed in Chapter 3, by shifting the observations according to the time delay sequence. The time delay sequence essentially follows a hidden Markov model, which is governed by a transition matrix that can be estimated. The proposed method IOPSFA_VTD is formulated under variational Bayesian framework. It incorporates prior process knowledges and can provide more accurate estimation of the posterior distribution for unknown parameters. The output with larger sampling interval and uncertain time delays can be effectively utilized in extracting latent features. The improved ability in predicting the desired key variables when considering time-varying time delays is validated through a numerical example along with a CSTR example.

In Chapter 5, the proposed method, IOPSFA_InVTD is a generalized algorithm of IOPSFA_VTD proposed in Chapter 4. In IOPSFA_VTD, considering the output variable has time-varying delays is equivalent to assuming all input variables has the same time delay at each time instant in reference to the output. IOPSFA_InVTD

extend this by considering different time-varying delays for different input variables, which is more common in the industrial processes due to the distributed locations of sensors. In the formulation of IOPSFA_InVTD, shifted input observations are reconstructed according to multiple time delay sequences and these delay sequences are governed by different hidden Markov model transition matrices, respectively. In this way, the extracted slow features have better prediction ability than IOPSFA_VTD and fixed delay cases. The performance has been validated through a simulated numerical example along with an industrial application.

To summarize, this thesis has addressed various irregular properties of process data in building inferential models by extracting dynamic latent features under MLE and variational Bayesian framework, i.e. outlier problem in Chapter 2, randomly missing and multi-rate problem in Chapter 3, and time-varying time delay for output and input variables problems in Chapter 4 and Chapter 5, respectively.

6.2 Future work

Each method discussed above is based on certain assumptions of the processes. There exist many different processes with different properties and conditions, the followings are two of them that may be further explored.

- **Unknown time delay range** In Chapter 4 and Chapter 5, the time delay range is assumed to be known and determined by our process knowledges. The faulty assumptions of time delay may result in inaccurate even wrong feature extraction and parameter estimation. For example, the minimum time delay may not necessarily start with zero and maximum value may be unknown and should be determined by the estimation.

- **Non-slowly-varying process modeling** In this thesis, we mainly discussed the modeling methods for slowly-varying process, which can apply to most of the chemical processes. If the slowest features are selected to build regression or monitoring models for fast-varying processes, it may lead to the unsatisfied results. For example, to predict flooding and weeping events of a distillation tower is difficult by using the slowest varying features. Some of the flooding

or weeping events occur due to the slow accumulation of heat or condensed flow. But there are other events occurring very fast due to some unexpected and sudden interruptions. For those events that happen fast and suddenly, choosing a group of latent features with certain varying speed (determined by λ) may be beneficial. In the modeling process, how to choose the varying speed according to the event properties and causes with the help of causality analysis and frequency domain analysis is an interesting topic to study.

- **Transfer learning between similar problems** Regarding to the missing data problem, sometimes the measurements are neither missing at random nor multi-rate sampled, it is completely unavailable. For example, in the industrial application of SAGD well pair water content soft sensing problem, often there are no VX meters installed for many well pads, but only a test separator available to separate the emulsion into vapor, oil and gas and measure the water content at the same time. The test separator rotates between different well pairs. It can only take intake flow from one well pair at a time. So the referenced water content values are only available for the specific well pair for a limited time and for most of the time there are no measurements for it. Previously, we can use the available water content values to build separate models for each well pair and the model cannot be updated until next time test separator is available. By using transfer learning concepts, the well pair models without test separator can still be updated using the well pair model with test separator if some common interruptions happen to all well pairs. Thus, we can utilize the knowledges learned from well pair with test separator to maintain a relatively accurate water content prediction for other well pairs. Otherwise, large bias in prediction will be introduced for most of the well pairs if unexpected interruption happens.

Bibliography

[1] Khatibisepehr S, Huang B, Khare S. Design of inferential sensors in the process industry: A review of Bayesian methods. Journal of Process Control. 2013 Nov 1;23(10):1575-96.

[2] Rousseeuw PJ, Leroy AM. Robust regression and outlier detection. John wiley & sons; 2005 Feb 25.

[3] Zeng JS, Gao CH. Improvement of identification of blast furnace ironmaking process by outlier detection and missing value imputation. Journal of Process Control. 2009 Oct 1;19(9):1519-28.

[4] McLachlan, G.J. and Peel, D., 2004. Finite mixture models. John Wiley & Sons.

[5] Gerogiannis, D., Nikou, C. and Likas, A., 2009. The mixtures of Student's t-distributions as a robust framework for rigid registration. Image and Vision Computing, 27(9), pp.1285-1294.

[6] Lu, Y. and Huang, B., 2014. Robust multiple-model LPV approach to nonlinear process identification using mixture t distributions. Journal of Process Control, 24(9), pp.1472-1488.

[7] Lu Y, Huang B, Khatibisepehr S. A variational Bayesian approach to robust identification of switched ARX models. IEEE transactions on cybernetics. 2015 Dec 1;46(12):3195-208.

[8] Khatibisepehr, S. and Huang, B., 2013. A Bayesian approach to robust process identification with ARX models. AIChE Journal, 59(3), pp.845-859.

[9] Jin X, Huang B. Robust identification of piecewise/switching autoregressive exogenous process. AIChE journal. 2010 Jul;56(7):1829-44.

[10] Kotz S, Kozubowski T, Podgorski K. The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. Springer Science & Business Media; 2012 Dec 6.

[11] Zhao C, Yang J. A Robust Skewed Boxplot for Detecting Outliers in Rainfall Observations in Real-Time Flood Forecasting. Advances in Meteorology. 2019;2019.

[12] Kim HC, Ghahramani Z. Outlier robust gaussian process classification. InJoint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) 2008 Dec 4 (pp. 896-905). Springer, Berlin, Heidelberg.

[13] Rubin DB. Inference and missing data. Biometrika. 1976 Dec 1;63(3):581-92.

[14] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society Series B (methodological). 1977:1–38.

[15] Jin X, Wang S, Huang B, Forbes F. Multiple model based LPV soft sensor development with irregular/missing process output measurement. Control Engineering Practice. 2012 Feb 1;20(2):165-72.

[16] Deng J, Huang B. Identification of nonlinear parameter varying systems with missing output data. AIChE journalnal. 2012 Nov;58(11):3454-67.

[17] Yang X, Huang B, Zhao Y, Lu Y, Xiong W, Gao H. Generalized expectation–maximization approach to LPV process identification with randomly missing output data. Chemometrics and Intelligent Laboratory Systems. 2015 Nov 15;148:1-8.

[18] Yang X, Xiong W, Huang B, Gao H. Identification of linear parameter varying systems with missing output data using generalized expectation-maximization algorithm. IFAC Proceedings Volumes. 2014 Jan 1;47(3):9364-9.

[19] Liu Z, Hansson A, Vandenberghe L. Nuclear norm system identification with missing inputs and outputs. Systems & Control Letters. 2013 Aug 1;62(8):605-12.

[20] Shang C, Huang B, Yang F, Huang D. Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling. AIChE Journal. 2015;61(12):4126-4139.

[21] Cooke KL, Grossman Z. Discrete delay, distributed delay and stability switches. Journal of mathematical analysis and applications. 1982 Apr 1;86(2):592-627.

[22] Rao G, Sivakumar L. Identification of time-lag systems via Walsh functions. IEEE Transactions on Automatic Control. 1979 Oct;24(5):806-8.

[23] Guzmán JL, Garcia P, Hägglund T, Dormido S, Albertos P, Berenguel M. Interactive tool for analysis of time-delay systems with dead-time compensators. Control Engineering Practice. 2008 Jul 1;16(7):824-35.

[24] Björklund S. Experimental evaluation of some cross correlation methods for time-delay estimation in linear systems. Linköping University Electronic Press; 2003.

[25] Söderström T, Stoica P, editors. System identification. Prentice-Hall, Inc.; 1988 Jan 1.

[26] Richard JP. Time-delay systems: an overview of some recent advances and open problems. Automatica. 2003 Oct 1;39(10):1667-94.

[27] Zheng G, Polyakov A, Levant A. Delay estimation via sliding mode for nonlinear time-delay systems. Automatica. 2018 Mar 1;89:266-73.

[28] Ren XM, Rad AB, Chan PT, Lo WL. Online identification of continuous-time systems with unknown time delay. IEEE Transactions on Automatic Control. 2005 Sep 12;50(9):1418-22.

[29] Anguelova M, Wennberg B. State elimination and identifiability of the delay parameter for nonlinear time-delay systems. Automatica. 2008 May 1;44(5):1373-8.

[30] Zhao Y, Fatehi A, Huang B. Robust estimation of ARX models with time varying time delays using variational Bayesian approach. IEEE transactions on cybernetics. 2017 Jan 10;48(2):532-42.

[31] Mazenc F. Stability analysis of time-varying neutral time-delay systems. IEEE Transactions on Automatic Control. 2014 Jul 23;60(2):540-6.

[32] Han QL. On robust stability of neutral systems with time-varying discrete delay and norm-bounded uncertainty. Automatica. 2004 Jun 1;40(6):1087-92.

[33] Xie L, Yang H, Huang B. FIR model identification of multirate processes with random delays using EM algorithm. AIChE Journal. 2013 Nov;59(11):4124-32.

[34] Zhao Y, Fatehi A, Huang B. A data-driven hybrid ARX and Markov chain modeling approach to process identification with time-varying time delays. IEEE Transactions on Industrial Electronics. 2016 Aug 2;64(5):4226-36.

[35] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013 Mar 7;35(8):1798-828.

[36] Pearson K. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901 Nov 1;2(11):559-72.

[37] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of educational psychology. 1933 Sep;24(6):417.

[38] Wold H. Estimation of principal components and related models by iterative least squares. Multivariate analysis. 1966:391-420.

[39] Geladi P, Kowalski BR. Partial least-squares regression: a tutorial. Analytica chimica acta. 1986 Jan 1;185:1-7.

[40] Qin SJ. Partial least squares regression for recursive system identification. In: Proceedings of the 32nd IEEE Conference on Decision and Control. 1993;3:2617-2622.

[41] Jutten C, Herault J. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal processing. 1991;24(1):1–10.

[42] Comon P. Independent component analysis, A new concept? Signal Processing. 1994;36(3):287-314.

[43] Laurenz Wiskott, Sejnowski TJ. Slow feature analysis: Unsupervised learning of invariances. Neural computation. 2002;14(4):715–770.

[44] Tipping ME, Bishop C. Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B. 1999;61(3):611-622.

[45] Li S, Gao J, Nyagilo JO, Dave DP. Probabilistic partial least square regression: A robust model for quantitative analysis of raman spectroscopy data. In2011 IEEE International Conference on Bioinformatics and Biomedicine 2011 Nov 12 (pp. 526-531). IEEE.

[46] Beckmann CF, Smith SM. Probabilistic independent component analysis for functional magnetic resonance imaging. IEEE transactions on medical imaging. 2004 Feb 6;23(2):137-52.

[47] Turner R, Sahani M. A maximum-likelihood interpretation for slow feature analysis. Neural computation. 2007;19(4):1022–1038.

[48] Jolliffe, IT. Principal Component Analysis. 2nd ed. Springer. 2002.

[49] Candès EJ, Li X, Ma Y, Wright J. Robust principal component analysis?. Journal of the ACM (JACM). 2011 Jun 9;58(3):1-37.

[50] Tiwari A, Dutertre B, Jovanović D, de Candia T, Lincoln PD, Rushby J, Sadigh D, Seshia S. Safety envelope for security. InProceedings of the 3rd international conference on High confidence networked systems 2014 Apr 15 (pp. 85-94).

[51] Ge Z, Song Z, Kano M. External analysis-based regression model for robust soft sensing of multimode chemical processes. AIChE Journal. 2014 Jan;60(1):136-47.

[52] Zhu J, Ge Z, Song Z. Robust supervised probabilistic principal component analysis model for soft sensing of key process variables. Chemical Engineering Science. 2015 Jan 27;122:573-84.

[53] Sadeghian A, Huang B. Robust probabilistic principal component analysis for process modeling subject to scaled mixture Gaussian noise. Computers & Chemical Engineering. 2016 Jul 12;90:62-78.

[54] Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems. 2001;58(2):109-130.

[55] Rosipal R, Krämer N. Overview and recent advances in partial least squares. In-International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection" 2005 Feb 23 (pp. 34-51). Springer, Berlin, Heidelberg.

[56] Wold S, Ruhe A, Wold H, Dunn WJ. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. SIAM Journal on Scientific and Statistical Computing. 1984;5(3):735–743.

[57] Li S, Gao J, Nyagilo JO, Dave DP, Zhang B, Wu X. A unified probabilistic PLSR model for quantitative analysis of surface-enhanced Raman spectrum (SERS). InThe Proceedings of the Second International Conference on Communications, Signal Processing, and Systems 2014 (pp. 1095-1103). Springer, Cham.

[58] Ma M, Khatibisepehr S, Huang B. A Bayesian framework for real-time identification of locally weighted partial least squares. AIChE Journal. 2015 Feb;61(2):518-29.

[59] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. Neural networks. 2000 Jun 1;13(4-5):411-30.

[60] Back AD, Weigend AS. A first application of independent component analysis to extracting structure from stock returns. International journal of neural systems. 1997 Aug;8(04):473-84.

[61] Allassonniere S, Younes L. A stochastic algorithm for probabilistic independent component analysis. The Annals of Applied Statistics. 2012;6(1):125-60.

[62] Takala J, Hyvarinen A. Sparse code shrinkage: De-noising of non Gaussian data by maximum likelihood estimation. Neural Comput. 1999;11(7):1739-68.

[63] Penny WD. ICA: model order selection and dynamic. Independent component analysis: Principles and practice. 2001:299.

[64] Hyvärinen A. A unified probabilistic model for independent and principal component analysis. InAdvances in Independent Component Analysis and Learning Machines 2015 Jan 1 (pp. 75-82). Academic Press.

[65] Vapnik V. The nature of statistical learning theory. Springer science & business media; 2013 Jun 29.

[66] Wiskott L, Berkes P, Franzius M, Sprekeler H, Wilbert N. Slow feature analysis. Scholarpedia. 2011 Apr 12;6(4):5282.

[67] Jin X, Huang B. Identification of switched Markov autoregressive eXogenous systems with hidden switching state. Automatica. 2012 Feb 1;48(2):436-41.

[68] Chen L, Huang B, Ding Y. Multiple model LPV approach to identification of nonlinear dual-rate system with random time delay. IFAC-PapersOnLine. 2015 Jan 1;48(28):1220-5.

[69] Bernardo JM, Bayarri MJ, Berger JO, Dawid AP, Heckerman D, Smith AF, West M. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. Bayesian statistics. 2003 Sep;7:453-64.

[70] Ljung, L., 1999. System identification. Wiley Encyclopedia of Electrical and Electronics Engineering, pp.1-19.

[71] Nelles, O., 2013. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer Science & Business Media.

[72] Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometrics and intelligent laboratory systems. 1987;2(1-3):37-52.

[73] Christopher MB. Pattern recognition and machine learning. Springer-Verlag New York, 2006.

[74] Negiz, A. and Çlinar, A., 1997. Statistical monitoring of multivariable dynamic processes with state-space models. AIChE Journal, 43(8), pp.2002-2020.

[75] Shang C, Yang F, Gao X, Huang D. Extracting latent dynamics from process data for quality prediction and performance assessment via slow feature regression. In: American Control Conference (ACC), 2015. IEEE; 2015:912–917.

[76] Guo F, Shang C, Huang B, Wang K, Yang F, Huang D. Monitoring of operating point and process dynamics via probabilistic slow feature analysis. Chemometrics and Intelligent Laboratory Systems. 2016;151:115-125.

[77] Svensénn, M. and Bishop, C.M., 2005. Robust Bayesian mixture modelling. Neurocomputing, 64, pp.235-252.

[78] Peel, D. and McLachlan, G.J., 2000. Robust mixture modelling using the t distribution. Statistics and computing, 10(4), pp.339-348.

[79] Mitchison, G., 1991. Removing time variation with the anti-Hebbian differential synapse. Neural Computation, 3(3), pp.312-320.

[80] Liu, C., 1997. ML Estimation of the MultivariatetDistribution and the EM Algorithm. Journal of Multivariate Analysis, 63(2), pp.296-312.

[81] Anderson, B.D. and Moore, J.B., 1979. Optimal filtering Englewood cliffs. NJ: Pretice-Hall.

[82] Braatz, Richard. TE data download website, url: http://web.mit.edu/braatzgroup/links.html

[83] Fortuna L, Graziani S, Rizzo A, Xibilia MG. Soft sensors for monitoring and control of industrial processes. Springer Science & Business Media, 2007.

[84] Ge Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. Chemometrics and Intelligent Laboratory Systems. 2017;171:16-25.

[85] Hartnett MK, Lightbody G, Irwin GW. Dynamic inferential estimation using principal components regression (PCR). Chemometrics and Intelligent Laboratory Systems. 1998;40(2):215-224.

[86] Tipping ME, Bishop CM. Mixtures of Probabilistic Principal Component Analyzers. Neural Computation. 1999;11(2):443-482.

[87] Ge Z, Gao F, Song Z. Mixture probabilistic PCR model for soft sensing of multimode processes. Chemometrics and Intelligent Laboratory Systems. 2011;105(1):91-105.

[88] Ge Z, Song Z. Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples. AIChE Journal. 2011 Aug;57(8):2109-19.

[89] Zhou L, Chen J, Song Z, Ge Z. Semi-supervised PLVR models for process monitoring with unequal sample sizes of process variables and quality variables. Journal of Process Control. 2015;26:1-16.

[90] Chapelle O, Schölkopf B, Zien A. Semi-Supervised Learning. Cambridge, Mass: MIT Press, 2006.

[91] Zhu X. Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison. 2006;2(3):4.

[92] Little RJ, Rubin DB. Statistical analysis with missing data. John Wiley & Sons, 2014.

[93] Lu N, Yang Y, Gao F, Wang F. Multirate dynamic inferential modeling for multivariable processes. Chemical Engineering Science. 2004;59(4):855-864.

[94] Grung B, Manne R. Missing values in principal component analysis. Chemometrics and Intelligent Laboratory Systems. 1998;42(1):125-139.

[95] Woodbury MA. A missing information principle: Theory and applications. Duke University Medical Center Durham United States, 1970.

[96] Walczak B, Massart DL. Dealing with missing data: Part I. Chemometrics and Intelligent Laboratory Systems. 2001;58(1):15-27.

[97] Walczak B, Massart DL. Dealing with missing data: Part II. Chemometrics and Intelligent Laboratory Systems. 2001;58(1):29-42.

[98] Andrews DT, Wentzell PD. Applications of maximum likelihood principal component analysis: incomplete data sets and calibration transfer. Analytica Chimica Acta. 1997;350(3):341-352.

[99] Reis MS, Saraiva PM. Integration of data uncertainty in linear regression and process optimization. AIChE Journal. 2005;51(11):3007-3019.

[100] Arteaga F, Ferrer A. Dealing with missing data in MSPC: several methods, different interpretations, some examples. Journal of Chemometrics. 2002;16(8-10):408-418.

[101] Nelson PR, Taylor PA, MacGregor JF. Missing data methods in PCA and PLS: Score calculations with incomplete observations. Chemometrics and Intelligent Laboratory Systems. 1996;35(1):45-65.

[102] Rato TJ, Reis MS. Multiresolution soft sensors: A new class of model structures for handling multiresolution data. Ind Eng Chem Res. 2017;56(13):3640-3654.

[103] Reis MS, Rendall R, Chin S-T, Chiang L. Challenges in the Specification and Integration of Measurement Uncertainty in the Development of Data-Driven Models for the Chemical Processing Industry. Industrial & Engineering Chemistry Research. 2015;54(37):9159-9177.

[104] Digalakis V, Rohlicek JR, Ostendorf M. ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. IEEE Transactions on Speech and Audio Processing. 1993;1(4):431-442.

[105] Chen L, Khatibisepehr S, Huang B, Liu F, Ding Y. Nonlinear process identification in the presence of multiple correlated hidden scheduling variables with missing data. AIChE J. 2015;61(10):3270-3287.

[106] Jin X, Wang S, Huang B, Forbes F. Multiple model based LPV soft sensor development with irregular/missing process output measurement. Control Engineering Practice. 2012;20(2):165-172.

[107] Bakshi BR, Nounou MN, Goel PK, Shen X. Multiscale bayesian rectification of data from linear steady-state and dynamic systems without accurate models. Ind Eng Chem Res. 2001;40(1):261-274.

[108] Daoudi K, Frakt AB, Willsky AS. Multiscale autoregressive models and wavelets. IEEE Transactions on Information Theory. 1999;45(3):828-845.

[109] Marco S. Reis. A multiscale empirical modeling framework for system identification. Journal of Process Control. 2009;19(9):1546-1557.

[110] Jemwa GT, Krishnannair S, Aldrich C. Multiscale process monitoring with singular spectrum analysis. IFAC Proceedings Volumes. 2007;40(11):167-172.

[111] Favoreel W, De Moor B, Van Overschee P. Subspace state space system identification for industrial processes. Journal of Process Control. 2000;10(2):149-155.

[112] Juricek BC, Seborg DE, Larimore WE. Identification of Multivariable, Linear, Dynamic Models: Comparing Regression and Subspace Techniques. Ind Eng Chem Res. 2002;41(9):2185-2203.

[113] Van Overschee P, De Moor B. A unifying theorem for three subspace system identification algorithms. Automatica. 1995;31(12):1853-1864.

[114] Everett B. An Introduction to Latent Variable Models. Springer Netherlands, 1984.

[115] Raveendran R, Kodamana H, Huang B. Process monitoring using a generalized probabilistic linear latent variable model. Automatica. 2018 Oct 1;96:73-83.

[116] Ma Y, Huang B. Bayesian learning for dynamic feature extraction with application in soft sensing. IEEE Transactions on Industrial Electronics. 2017;64(9):7171-7180.

[117] Tzagkarakis G, Papadopouli M, Tsakalides P. Trend forecasting based on Singular Spectrum Analysis of traffic workload in a large-scale wireless LAN. Performance Evaluation. 2009;66(3-5):173-190.

[118] Lu Y, Saniie J. A comparative study of singular spectrum analysis and empirical mode decomposition for ultrasonic NDE. In: 2016 IEEE International Ultrasonics Symposium (IUS). 2016:1-4.

[119] Ku W, Storer RH, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems. 1995;30(1):179-196.

[120] Rato TJ, Reis MS. Advantage of using decorrelated residuals in dynamic principal component analysis for monitoring large-scale systems. Ind Eng Chem Res. 2013;52(38):13685-13698.

[121] Fan L, Kodamana H, Huang B. Identification of robust probabilistic slow feature regression model for process data contaminated with outliers. Chemometrics and Intelligent Laboratory Systems. 2018;173:1-13.

[122] Ge Z, Chen X. Dynamic probabilistic latent variable model for process data modeling and regression application. IEEE Transactions on Control Systems Technology. 2017 Nov 13;27(1):323-31.

[123] Butler RM. Thermal recovery of oil and bitumen. Prentice Hall, 1991.

[124] Tohidi Hosseini SM, Esfahani S, Hashemi Doulatabadi M, Hemmati Sarapardeh A, Mohammadi AH. On the evaluation of steam assisted gravity drainage in naturally fractured oil reservoirs. Petroleum. 2017;3(2):273-279.

[125] Rato TJ, Reis MS. Defining the structure of DPCA models and its impact on process monitoring and prediction activities. Chemometrics and Intelligent Laboratory Systems. 2013;125:74-86.

[126] De Assis AJ, Maciel Filho R. Soft sensors development for on-line bioreactor state estimation. Computers & Chemical Engineering. 2000 Jul 15;24(2-7):1099-103.

[127] Ma Y, Kwak S, Fan L, Huang B. A variational bayesian approach to modelling with random time-varying time delays. In2018 Annual American Control Conference (ACC) 2018 Jun 27 (pp. 5914-5919). IEEE.

[128] Fan L, Kodamana H, Huang B. Semi-supervised dynamic latent variable modeling: I/O probabilistic slow feature analysis approach. AIChE Journal. 2019 Mar;65(3):964-79.

[129] Everett B. An introduction to latent variable models. Springer Science & Business Media; 2013 Mar 7.

[130] Russell EL, Chiang LH, Braatz RD. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometrics and intelligent laboratory systems. 2000 May 8;51(1):81-93.

[131] Wold, H. Partial least squares. Encyclopedia of statistical sciences. 2006.

[132] Qin SJ. Recursive PLS algorithms for adaptive data modeling. Computers & Chemical Engineering. 1998 Jan 1;22(4-5):503-14.

[133] Zheng J, Song Z, Ge Z. Probabilistic learning of partial least squares regression model: Theory and industrial applications. Chemometrics and Intelligent Laboratory Systems. 2016 Nov 15;158:80-90.

[134] Beal MJ. Variational algorithms for approximate Bayesian inference. London: university of London; 2003 May.

[135] Ostwald D, Kirilina E, Starke L, Blankenburg F. A tutorial on variational Bayes for latent linear stochastic time-series models. Journal of Mathematical Psychology. 2014 Jun 1;60:1-9.

[136] Rauch HE, Tung F, Striebel CT. Maximum likelihood estimates of linear dynamic systems. AIAA journal. 1965 Aug;3(8):1445-50.

[137] Barber D, Chiappa S. Unified inference for variational Bayesian linear Gaussian state-space models. InAdvances in Neural Information Processing Systems 2007 (pp. 81-88).

[138] Ma Y, Huang B. Extracting dynamic features with switching models for process data analytics and application in soft sensing. AIChE Journal. 2018 Jun;64(6):2037-51.

[139] Chen L, Han L, Huang B, Liu F. Parameter estimation for a dual-rate system with time delay. ISA transactions. 2014 Sep 1;53(5):1368-76.

[140] Gao X, Xie F, Hu H. Enhancing the security of electro-optic delayed chaotic system with intermittent time-delay modulation and digital chaos. Optics Communications. 2015 Oct 1;352:77-83.

[141] Boudreau D, Kabal P. Joint time-delay estimation and adaptive recursive least squares filtering. IEEE Transactions on Signal processing. 1993 Feb;41(2):592-601.

[142] Boudreau D, Kabal P. Joint gradient-based time delay estimation and adaptive filtering. InIEEE International Symposium on Circuits and Systems 1990 May 1 (pp. 3165-3169). IEEE.

[143] Zhang H, Yang F, Liu X, Zhang Q. Stability analysis for neural networks with time-varying delay based on quadratic convex combination. IEEE transactions on neural networks and learning systems. 2013 Jan 14;24(4):513-21.

[144] Sirisongkol R, Liu X. Stability analysis of recurrent neural networks with time-varying delay and disturbances via quadratic convex technique. InFifth International Conference on Intelligent Control and Information Processing 2014 Aug 18 (pp. 130-137). IEEE.