Enhanced Probabilistic Slow Feature Analysis - Dealing with Complexities in Industrial Process Data

by

Vamsi Krishna Puli

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

Doctor of Philosophy

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

In modern industrial processes, the measurement and storage of thousands of correlated process variables have become commonplace. Dimensionality reduction techniques are often employed to extract underlying informative patterns called features by discarding redundant information. Slow feature analysis is one such technique that focuses on extracting slowly varying patterns. A probabilistic extension was proposed to address data corruption caused by measurement noise. However, industrial process data is fraught with additional complexities, including periodic patterns from plant-wide oscillations, non-stationarity due to aging equipment, non-linearities, skewed noise distribution etc. The estimated parameter error will be large when a conventional probabilistic slow feature model is applied to complex industrial data. Hence, this thesis focuses on enhancing the probabilistic slow feature model to accommodate various industrial complexities.

Oscillatory behavior commonly arises in measured data as a result of poor controller tuning, stiction, and external oscillatory disturbances. Identifying and analyzing these oscillatory patterns is vital for monitoring control loops and diagnosing faults. Unfortunately, the presence of significant measurement noise prevents the conventional slow feature analysis from effectively extracting these patterns due to limitations in the model structure. Therefore, the primary contribution of this thesis is to develop an enhanced slow feature model that overcomes this limitation by relaxing the diagonal structure of the state-transition matrix and incorporating a block-diagonal matrix structure to accommodate complex poles. As a result, the enhanced slow feature analysis is called complex slow feature analysis in this thesis.

Further, the drift-type non-stationary characteristics in measured variables also pose significant challenges for conventional slow feature extraction methods as the corresponding slow features are assumed stationary. Consequently, the second contribution of this thesis is to address this issue by introducing an additional latent variable that compensates for the drift-type non-stationary behaviour, thereby ensuring the stationarity of the extracted slow features. Due to the inherent non-linearity observed in complex industrial processes, we enhance the second contribution by incorporating an extended gated recurrent neural network architecture.

Process data commonly suffer from measurement issues like outliers and asymmetric noise distributions, impacting the quality and performance of extracted slow features. Our fourth contribution proposes a robust complex slow feature model that assumes a skewed t-distribution for measurement noise, rather than a Gaussian distribution. Model parameters are jointly estimated using the expectation-maximization algorithm. Additionally, high-dimensional datasets often stem from a low-dimensional latent space, and not all latent features influence all measured variables. Hence, it is crucial to ensure that only a subset of latent variables influences each measured variable. To address this, our fifth contribution introduces a novel model that automatically determines the optimal latent space dimension by employing a Laplace distribution to model the emission matrix, resulting in a sparse model. The conventional black-box nature of the slow feature model and its numerous extensions may lead to inconsistent or unacceptable results at the physical boundaries. Therefore, the final contribution integrates process knowledge into the probabilistic slow feature model to extract features that adhere to physical laws/limits.

The efficiency of all contributions is demonstrated through simulations and industrial/experimental case studies in which they are compared to state-of-the-art methods. This comparison yields conclusive evidence of their effectiveness.

Preface

This thesis is an original work conducted by Vamsi Krishna Puli under the supervision of Dr. Biao Huang and is funded in part by Natural Sciences and Engineering Research Council (NSERC) of Canada. Portions of the thesis have been published in peer-reviewed journals. Some of the chapters may have some overlaps, which is to ensure each chapter is self-contained.

1. Chapter 3 of this thesis has been published as: **V. K. Puli**, R. Raveendran, and B. Huang, "Complex probabilistic slow feature extraction with applications in process data analytics". *Computers and Chemical Engineering*, 154, 107456, 2021

2. Chapter 4 of this thesis has been published as: **V. K. Puli** and B. Huang, "Variational Bayesian Approach to Nonstationary and Oscillatory Slow Feature Analysis With Applications in Soft Sensing and Process Monitoring," in *IEEE Transactions on Control Systems Technology*, doi: 10.1109/TCST.2023.3240980, 2023

3. Chapter 7 of this thesis has been published as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Sparse Robust Dynamic Feature Extraction using Bayesian Inference", *IEEE Transactions on Industrial Electronics*, 2023.

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

1. Chapter 5 of this thesis has been submitted as: **V. K. Puli** and B. Huang, "Nonlinear Slow Feature Analysis for Oscillating Characteristics under Deep Encoder-Decoder Framework", *IEEE Transactions on Industrial Informatics*, 2023

2. Chapter 6 of this thesis has been published as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Robust Complex Probabilistic Slow Feature Analysis in the Presence of Skewed Measurement Noise", *IFAC-PapersOnLine*, 56(2), 10947-10952, 2023

3. Chapter 8 of this thesis has been submitted as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Physics Informed Probabilistic Slow Feature Extraction", *Automatica*, 2023

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

Credit Authorship Contribution Statement in applicable co-authored journal publications:

- **Vamsi Krishna Puli:** Conceptualization, Data curation, Formal analysis, Algorithm development, Investigation, Methodology, Software development, Validation, Visualization, Writing –original draft, Writing –review & editing.

- **Ranjith Chiplunkar:** Providing support in algorithm development, Resources, and Writing –review & editing.

- **Rahul Raveendran:** Formal Analysis, Resources, and Writing –review & editing.

- **Biao Huang:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Resources, Supervision, Writing – review & editing.

Dedicated to

My loving parents Venkata Ramana Puli and Padmavati Puli, and all my teachers.

Acknowledgements

With utmost gratitude, I acknowledge Prof. Biao Huang, my thesis advisor, for his guidance, motivation, and freedom to explore diverse research topics while ensuring my timely focus on tasks and promptly reviewing my work. Dr. Huang's expertise in the field, his keen eye for detail, and his commitment to excellence have continuously pushed me to strive for the highest standards in my work. His constant encouragement and constructive feedback have been instrumental in shaping my research direction and refining my skills. I am indebted to him for entrusting me with challenging industrial problems, instilling confidence, and providing invaluable financial support. I also extend my heartfelt appreciation to Dr. Vinay Prasad and Dr. Zukui Li, members of the supervisory committee, for their invaluable feedback during my candidacy exam, as it significantly contributed to the enhancement of my work.

I am sincerely thankful to Suncor Energies for providing me with invaluable opportunities to engage with practical refinery soft-sensing challenges. Collaborating with esteemed professionals like Dr. Yanjun Ma, Dr. Dian Wang, and Dr. Haitao Zhang at Suncor Energies has been an immensely rewarding experience, from which I have gained extensive knowledge and genuine satisfaction.

I express my heartfelt gratitude to my esteemed friend, Dr. Ranjith Chiplunkar, for his invaluable assistance in securing a residence for me in Edmonton prior to my arrival in September 2019. Our research collaboration commenced with an engaging project on industrial heat exchanger modeling, allowing me to gain comprehensive insights into the intricacies of the SAGD process. The countless long and enlightening discussions and debates we shared have significantly enriched my understanding of statistical concepts, such as expectation maximization and variational inference, offering a fresh and unique perspective.

I am grateful to Dr. Rahul Raveendran for his invaluable guidance and support in comprehending the intricacies of deep neural networks and steering my research endeavors during the critical early stages of my Ph.D. His assistance played a pivotal role in my preparation for the candidacy exam, a significant milestone in my Ph.D. journey.

I am sincerely grateful to the National Science and Engineering Research Council (NSERC) of Canada for their generous financial support, which has enabled me to pursue my Doctoral degree in the Department of Chemical Engineering at the University of Alberta. I gratefully acknowledge the financial support provided by the University of Alberta, which includes the UofA Ph.D. scholarship, Alberta Innovates Graduate Student Scholarship, Doctoral Recruitment Scholarship, DB Robinson Graduate Scholarship, and Captain Thomas Farrell Greenhalgh Memorial Graduate Scholarship. They have not only provided me with exceptional resources but have also cultivated a warm and supportive environment.

I am grateful for the invaluable assistance provided by numerous members of the Process Data Analytics and Smart Automation group, without whom this work would not have been possible. Special thanks are extended to my fellow Ph.D. mate, Yousef Salehi, with whom fond memories are shared, from playing table tennis to presenting at the IFAC World Congress, 2023. My regards are extended to Dr. Junyao Xie, Dr. Mengqi Fang, Dr. Santosh Kumar Varanasi, Dr. Seshu Kumar Damarla, Chaitanya Manchikatla, Dr. Purushottama Rao Dasari, Dr. Anahita Sadeghian, Dr. Lei Fan, Dr. Kirubakaran Velswamy, Dr. Hongtian Chen, Arun S. Sundaramoorthy, Parashara Kodati, Khizer Mohamed and Dr. Oguzhan Dogru for their support. I would also like to express my sincere appreciation to Mrs. Terry Runyon and Dr. Fadi Ibrahim for their unwavering support throughout my journey.

I express my heartfelt gratitude to my dear friends Kiran Raviprakash, Sarupa Debnath and Alireza Memarian, whose presence ensured that my initial days were not chaotic. I extend my thanks to my badminton partners Surya Chandra Lagisetti, Kiran Penchala, Chandana Reddy and Vedasree Mudireddy, as our outings together have created unforgettable memories. I would like to give a special mention to Nived Madhusoodhanan, Dr. Mani Hemanth Dhullipalla, Hari Krishna Rao and Abhishek Muralidharan for reviving my childhood memories through cricket. I take this opportunity to express my appreciation to my roommates Dr. Gokul Subraveti, Sanat

Kanekar, Janish Kumar Rajput, Rahul Chetri, and Gaurav Rasaily, whose affectionate and brotherly care made my stay in Edmonton truly enjoyable. A special mention to my long-time friends Neha Meka, Rahul Dev Appapogu, Naveen Shankar Mitikiri and Shanmukh Srinivas Battula, whose unwavering support and motivation pushed me to excel in every situation. I express my gratitude to two significant individuals from my alma mater: Dr. Arun K. Tangirala, my masters supervisor, for instilling the essence of data science within me and encouraging my research pursuits, and Dr. Sudhakar Kathari, who displayed exceptional care towards me, treating me as his own younger brother.

Lastly, I am immensely grateful for my family's love, support, and selfless sacrifices. Their unconditional love and unwavering support have meant the world to me.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

In recent years, the study of data-driven modeling [2–4] has experienced a surge in popularity, primarily driven by the abundance of historical data that has been made accessible through the utilization of cutting-edge measurement techniques and advanced data storage technologies. Industrial operational tasks, such as predictive modelling [5], fault diagnosis [6–8], quality monitoring, plant-wide oscillation detection [9–12], causality analysis [13–16], are greatly simplified. Within this vast sea of data, there exist significant interconnections among the various measured variables. Consequently, the resultant input-output models derived from such data often find themselves afflicted with a common pitfall known as over-fitting [17–20].

To counteract this issue and enhance the quality of process modeling, it has become commonplace to employ dimensionality reduction techniques [21–23] as a preliminary step. This strategic approach aims to eliminate redundant information while simultaneously extracting the most informative variables, aptly referred to as features, embedded within the data. By effectively reducing the dimensionality, subsequent modeling efforts between these extracted features and the desired outputs become notably less computationally burdensome. This reduction in computational complexity is primarily owed to the fact that the extracted features possess a comparatively lower dimensionality.

Given the tangible benefits of dimensionality reduction through feature extraction, it is no surprise that this methodology has garnered significant attention and intrigue from many scientific disciplines. Notably, domains such as econometrics [24, 25], health sciences [26–28], and process industry [29–33] have all recognized the potential and utility of feature extraction in their respective fields of study. As a result, researchers and practitioners alike have invested considerable effort into exploring and exploiting the power of feature extraction techniques to comprehend, model, and optimize complex systems across diverse domains. Some of the most popular linear latent variable models are Principal Component Analysis (PCA) [34,35], Slow Feature Analysis (SFA) [36], Partial Least Squares (PLS) [37–39], Independent Component Analysis (ICA) [40].

1. PCA is a technique that identifies a linear combination of the original variables that explains the maximum amount of variance. The resulting principal components provide a lower-dimensional representation of the data while preserving the most important information.

2. SFA, on the other hand, focuses on extracting the slowest varying feature from the input data. By identifying features that change slowly over time, SFA aims to capture the underlying dynamics and temporal relationships within the data.

3. PLS is a method that aims to identify a feature that exhibits maximum covariance between the input and output variables. It seeks to establish a predictive relationship between the two sets of variables by finding a latent variable that maximizes the covariance between them.

4. ICA, in contrast, aims to find features that exhibit maximum statistical independence. It assumes that the observed data is a linear combination of independent source signals and aims to estimate these sources by solving an optimization problem.

These models all employ different optimization criteria to project the high-dimensional data onto a lower-dimensional space. By extracting meaningful features from the data, they provide insights into the underlying structure, relationships, dynamics, and predictive capabilities of the variables under consideration. This thesis examines a specific latent variable model called slow feature analysis. In chemical process industries, slow variations in measured industrial data offer valuable insights into underlying patterns and relationships, aiding in the comprehension, control, and enhancement of chemical processes. SFA focuses on capturing these slower variations, which typically represent significant process information, while faster variations typically indicate transient events, disturbances, or noise. SFA's relevance in the chemical process industries is highlighted below.

1. Process Monitoring [41–46]: In process industries, it is crucial to monitor various process variables to ensure optimal performance, safety, and quality. SFA can be used to identify and extract slowly varying features from process data, which may correspond to important underlying dynamics or trends. By monitoring these features, process operators can detect deviations, anomalies, or gradual changes in the process behavior, enabling them to take timely corrective actions and prevent potential issues.

2. Process Understanding and Control [47]: SFA can provide valuable insights into the underlying dynamics and behavior of chemical processes. This understanding can be utilized to develop advanced control strategies, improve process design, or optimize control parameter settings.

3. Fault Diagnosis [48, 49]: When a process exhibits unexpected behavior or malfunctions, it is essential to diagnose the root cause accurately and quickly. The extracted slow features can reveal dependencies and interactions that may not be apparent in the original dataset. This can support the identification of specific variables or combinations of variables that contribute to process faults or deviations.

4. Applications to sustainable and recycling processes: SFA is applied successfully to detect icing faults in wind turbine blades, which can be subtle and evolve over time [50]. Further, an enhanced kernel slow feature approach is utilized to detect and identify faults in a nonlinear Air Handling Unit system [51]. Recycling facilities often deal with diverse materials, including different types of plastics, metals, paper, and glass. SFA could be employed to extract slow features from sensor data collected during the sorting process. These slow features could represent distinctive patterns or characteristics, making it possible to classify and sort recyclables more efficiently.

5. Data Visualization and Interpretation [52, 53]: Visualizing high-dimensional data is challenging, and interpreting complex patterns can be even more difficult. SFA can aid in data visualization by reducing the dimensionality while preserving relevant slow variations. By mapping the data onto a lower-dimensional space, SFA can provide visual representations that facilitate data interpretation, exploration, and communication among stakeholders in the chemical process industries.

1.1 Deterministic Slow Feature Analysis (DSFA):

Given an input sequence $X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_N \end{bmatrix}$, $\boldsymbol{x}_k \in \mathbb{R}^p$, the optimization problem shown in (1.1)-(1.5) can be solved to obtain slow features in the order of increasing velocities.

$$\min_{W} \quad \langle \dot{s}_k^{(i)^2} \rangle \tag{1.1}$$

$$\text{s.t} \quad \boldsymbol{s}_k = W^T \boldsymbol{x}_k \tag{1.2}$$

$$\langle s_k^{(i)} \rangle = 0 \tag{1.3}$$

$$\langle s_k^{(i)^2} \rangle = 1 \tag{1.4}$$

$$\forall i \neq j, \langle s_k^{(i)} \cdot s_k^{(j)} \rangle = 0 \tag{1.5}$$

where $\langle \dot{s}_k^{(i)^2} \rangle = \frac{1}{N-1} \sum_{k=2}^{N} (s_k^{(i)} - s_{k-1}^{(i)})^2$ denotes the squared average velocity, $W \in \mathbb{R}^{p \times m}$ indicates the projection matrix and $\langle \cdot \rangle$ stands for the average over data samples. N, m, and p denote the number of sampling points, latent features and observed variables, respectively. Equation (1.3) - (1.4) are applied to each slow feature to avoid trivial solutions, whereas (1.5) ensures zero correlation among the extracted features. The solution to the optimization problem can be readily obtained by performing singular value decomposition (SVD) twice, as shown in the following steps.

- **Whitening:** It is a transformation that converts a group of random variables into a new set of random variables, ensuring that their covariance becomes the identity matrix.

$$\boldsymbol{z}_k = Q \boldsymbol{x}_k \tag{1.6}$$

where Q represents whitening matrix that is defined as follows.

$$Q = \Lambda^{-\frac{1}{2}} V^T$$

Here Λ and V are obtained by performing SVD on the input data covariance matrix, as shown below.

$$\langle \boldsymbol{x}_k \boldsymbol{x}_k^T \rangle = V \Lambda V^T$$

Therefore, the slow feature equation (1.2) can be written as

$$\begin{aligned} \boldsymbol{s}_k = W^T \boldsymbol{x}_k = W^T Q^{-1} \boldsymbol{z}_k &= P^T \boldsymbol{z}_k && (1.7) \\ \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle = P^T \langle \boldsymbol{z}_k \boldsymbol{z}_k^T \rangle P &= P^T P && (1.8) \end{aligned}$$

- The objective of the slow feature analysis can be rewritten as

$$\begin{aligned} \min_{W} \quad & \langle \dot{s}_k^{(i)^2} \rangle \\ \text{s.t} \quad & \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle = \mathrm{I} \end{aligned} \qquad \Longrightarrow \qquad \begin{aligned} \min_{P} \quad & P^T \langle \dot{\boldsymbol{z}}_{\boldsymbol{k}} \dot{\boldsymbol{z}}_{\boldsymbol{k}}{}^T \rangle P \\ \text{s.t} \quad & P^T P = \mathrm{I} \end{aligned}$$

  The solution to this optimization problem can be obtained by performing SVD on the covariance matrix $\langle \dot{z}_k \dot{z}_k^T \rangle$.

However, this two-step solution can be avoided by just solving the generalized eigenvalue problem, as shown in 1.9.

$$AW = BW\Omega \tag{1.9}$$

where $A = \langle \dot{\boldsymbol{x}}_{\boldsymbol{k}} \dot{\boldsymbol{x}}_{\boldsymbol{k}}{}^T \rangle$, $B = \langle \boldsymbol{x}_k \boldsymbol{x}_k^T \rangle$, and Ω is a diagonal matrix whose diagonal entries represent the optimal velocities of each slow feature. Both solutions are equivalent.

1.2 Does Dynamic PCA achieve the same result as SFA?

Consider the following latent variable from dynamic PCA

$$\begin{aligned} t_k &= p^T x \\ &= \begin{bmatrix} p_c^T & p_p^T \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} \end{aligned}$$

$$t_k = \underbrace{p_c^T x_k}_{t_c} + \underbrace{p_p^T x_{k-1}}_{t_p}$$

So the resultant latent variable has two components. One is the static component "t_c" and the other is the dynamic component "t_p". The objective of PCA is to maximize the variance of the latent variable. So

$$\begin{array}{ll}\max\limits_{p} & t_k^T t_k \\ \text{s.t.} & p^T p = 1\end{array} \Longrightarrow \begin{array}{ll}\max\limits_{p} & (t_c + t_p)^T (t_c + t_p) \\ \text{s.t.} & p^T p = 1\end{array} \Longrightarrow \begin{array}{ll}\max\limits_{p} & t_c^T t_c + t_p^T t_p + 2 t_c^T t_p \\ \text{s.t.} & p^T p = 1\end{array}$$

$$\begin{array}{ll}\max\limits_{p_c, p_p} & p_c^T x_k x_k^T p_c + p_p^T x_{k-1} x_{k-1}^T p_p + 2 p_c^T x_k x_{k-1}^T p_p \\ \text{s.t.} & p_c^T p_c + p_p^T p_p = 1\end{array}$$

Therefore, PCA finds the latent variable t_k such that it maximizes the

- Variance of the static contribution t_c.
- Variance of the dynamic contribution t_p.
- Correlation between static and dynamic contributions.

However, SFA extracts latent variable $s_k = c^T x_k$ with a different objective as shown below.

$$\begin{array}{ll}\min\limits_{c} & (s_k - s_{k-1})^T (s_k - s_{k-1}) \\ \text{s.t.} & s_k^T s_k = 1\end{array} \Longrightarrow \begin{array}{ll}\min\limits_{c} & 2(1 - s_k^T s_{k-1}) \\ \text{s.t.} & s_k^T s_k = 1\end{array} \Longrightarrow \begin{array}{ll}\max\limits_{c} & c^T x_k x_{k-1}^T c \\ \text{s.t.} & c^T x_k x_k^T c = 1\end{array}$$

Note: The prediction of the DPCA latent variable t_k is made utilizing x_k and past values x_{k-1} rather than its own previous values t_{k-1}. Consequently, latent variable dimension reduction in terms of dynamics may not be enforced by the DPCA model. In general, the dynamics are diffused across all PCs. Conversely, SFA explicitly maximizes the auto-correlation of the latent variable. Therefore, the dimensionality reduction and feature ranking according to the dynamics are accomplished by SFA.

1.3 Probabilistic Slow Feature Analysis (PSFA):

DSFA is a method that aims to extract slowly varying features from a given dataset. However, DSFA assumes that the underlying factors causing the data variations are deterministic, which may not always hold true in real-world scenarios. The probabilistic formulation [54, 55] is presented in (1.10) - (1.11).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(0, Q) \tag{1.10}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(0, R) \tag{1.11}$$

where $A \in \mathbb{R}^{m\times m}$, $C \in \mathbb{R}^{p\times m}$, $Q \in \mathbb{R}^{m\times m}$, and $R \in \mathbb{R}^{p\times p}$ are the state-transition matrix, the emission matrix, the state-noise covariance matrix, and the measurement noise covariance matrix, respectively. In the above framework, $\boldsymbol{s}_k$ is a weighted function of $\boldsymbol{s}_{k-1}$ and an independent noise $\boldsymbol{w}_k$ that is drawn from Gaussian distribution. The probabilistic formulation achieves an enhanced temporal description for sequential process data since it imparts dynamics to the hidden features. Here m and p refer to number of slow features and the number of input variables, respectively. Several constraints are employed directly from deterministic slow feature analysis, as discussed below.

- The problem of interest is to obtain the slow features which are uncorrelated, and hence (A, Q) imbibe diagonal structures. Therefore, this formulation is limited to system with only real poles.

- The slow features are assumed to be stationary, *i.e.*, the state covariance matrix is independent of the time k. Also, the covariance matrix of latent variables is forced to be an identity in accordance with (1.4).

- Applying covariance on both sides of Eq. (1.10) to obtain a specific form of discrete algebraic Lyapunov equation,

  $$AA^T + Q = \mathrm{I}_m \tag{1.12}$$

  where I_m represents m dimensional Identity matrix. Since A and Q are diagonal, Eq. (1.12) reduces to:

  $$a_i^2 + q_i = 1 \tag{1.13}$$

  where a_i and q_i are the ith diagonal entries of A and Q respectively.

- For the positive definite requirement of the process noise covariance matrix Q, all the diagonal entries must be greater than zero ($q_i > 0$). Using Eq. (1.13), the bound constraint for a_i can be defined as shown in (1.14).

$$a_i \in \begin{pmatrix}-1 & 1\end{pmatrix} \tag{1.14}$$

  Eq. (1.14) can also be obtained by applying discrete state-space systems stability condition on the state-transition matrix. It is further restricted to $\begin{pmatrix}0 & 1\end{pmatrix}$ to avoid switching every sample.

- The measurement noise covariance matrix R is assumed to be diagonal.

The probabilistic graphical model is shown in Fig. 1.1. The advantages of PSFA are summarized below.

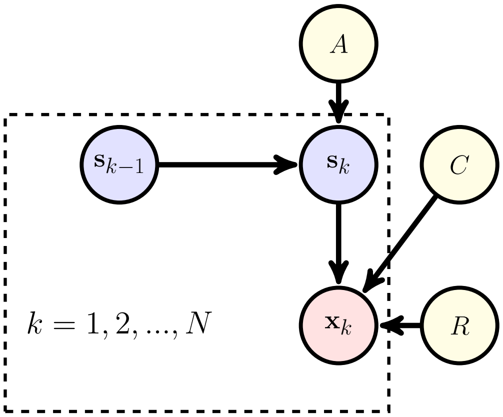


Figure 1.1: Probabilistic Graphical Model for Expectation Maximization based PSFA

- **Handling uncertainty:** PSFA accounts for uncertainty in the input data, as shown in (1.11). In many real-world scenarios, observations or measurements can be affected by noise or other factors, leading to uncertainties in the data. PSFA models this uncertainty explicitly through probabilistic distributions, allowing for more robust and reliable analysis.

- **Capturing latent variables:** PSFA provides an explicit dynamic equation (1.10) that represents the evolution of latent variables that might be hidden within the data. These latent variables can be viewed as the underlying factors or hidden causes that influence the observed data. By incorporating probabilistic modeling, PSFA can capture the uncertainty associated with these latent variables, leading to a more comprehensive understanding of the data.

- **Flexibility in modeling complex relationships**: PSFA provides a flexible framework for modeling complex relationships between variables. The probabilistic approach allows for capturing non-linear and higher-order dependencies that might exist in the data. This flexibility enables more accurate modeling of real-world phenomena.

- **Robustness to outliers [56]:** The probabilistic framework of PSFA allows a comprehensive way to estimate models robust to outliers or anomalies in the data. Outliers can significantly impact the performance of deterministic SFA, as they strive to minimize a specific objective function. In contrast, PSFA can effectively handle outliers by modeling them as low-probability events, thereby reducing their influence on the overall analysis.

- **Uncertainty quantification:** PSFA provides a means to quantify and express uncertainty in the derived features. This is particularly valuable in applications where uncertainty estimation is crucial, such as decision-making under uncertainty or in safety-critical systems. By explicitly modeling uncertainty, PSFA offers a principled approach to assess the reliability and confidence in the extracted slow features.

1.4 Motivation example:

In this section, three latent variables are generated using cosine functions with varying frequencies. Three observed variables are then obtained by combining the latent variables linearly. Through the application of DSFA, we can uniquely determine the underlying latent variables, each characterized by a distinct frequency, as shown in Fig.1.2. However, even the slightest measurement noise (SNR= 50) significantly impairs the efficiency of DSFA, as shown in Fig.1.3. To address this issue, PSFA can be used to significantly decouple the noise and the extracted features, as illustrated in the Fig. 1.4.

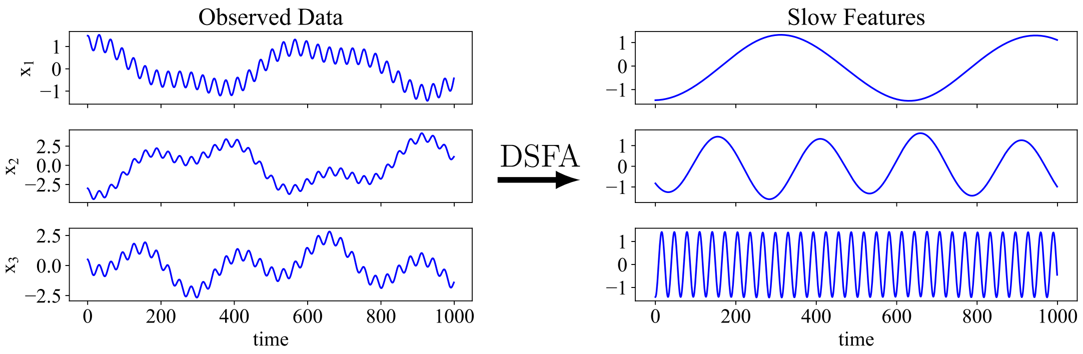


Figure 1.2: DSFA results on the noise-free input data

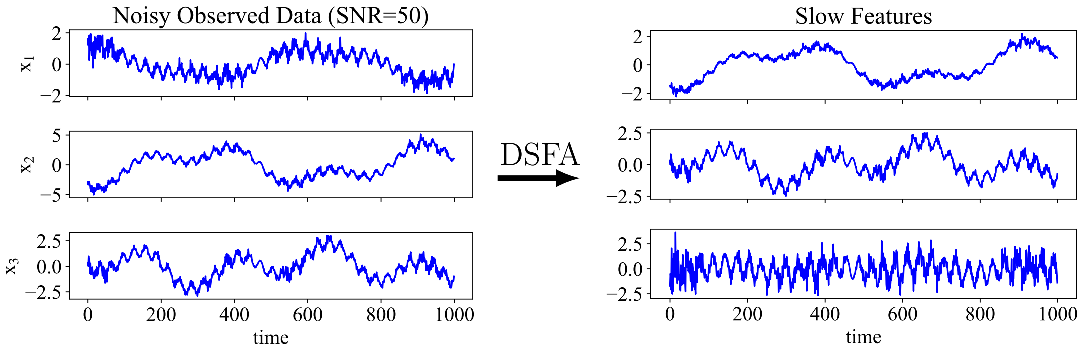


Figure 1.3: DSFA results on the noisy input data

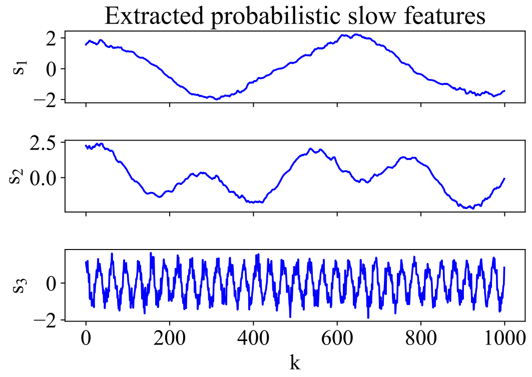


Figure 1.4: PSFA features extracted from noisy input data (SNR= 30)

While PSFA provides better separation, its effectiveness diminishes with decreasing SNR. The following (Fig. 1.5) features were extracted using PSFA from input data corrupted with measurement noise (SNR= 10). Interestingly, PSFA fails to extract the underlying cosine signals, which serves as a key motivation for the first contribution in this thesis.

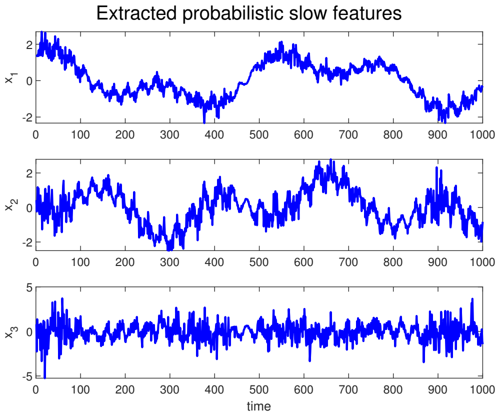


Figure 1.5: PSFA features extracted from noisy input data (SNR= 10)

1.5 Thesis Outline

The thesis begins by providing an introduction to latent variable models and probabilistic approaches. Following this background, the subsequent chapters are organized as follows.

Chapter 2 begins with an introduction to latent variable models and probabilistic approaches. The first section focuses on probability distributions, which will be used in subsequent chapters. We then present the maximum likelihood framework for parameter estimation. Furthermore, the solution methodology to approximate posterior distributions is explored through the variational inference framework. Additionally, we present the importance sampling methodology as an alternative for approximating the expectations of random variables whose priors are non-conjugate. The chapter concludes by presenting a state estimation algorithm called Kalman filter.

Chapter 3 focuses on addressing a significant limitation of the probabilistic slow feature analysis that has been discussed in the previous section. Specifically, PSFA encounters difficulties in extracting the underlying slow oscillating features when the signal-to-noise ratio is low. Given that oscillatory behavior is commonly observed in process data due to factors like inadequate control loop tuning and external disturbances (e.g., diurnal temperature variation), the extraction of these slow oscillatory patterns becomes crucial. Consequently, this chapter is dedicated to developing a technique for an effective extraction of such oscillating patterns from noisy data.

In process industries, it is common for the measured variables to exhibit non-stationary characteristics alongside oscillations. These non-stationary characteristics arise from the changing operating conditions, as well as equipment degradation and fouling. Dealing with such variations presents a significant challenge for conventional slow feature methods. To address this challenge, we introduce a probabilistic drift-type non-stationary oscillating slow feature model in Chapter 4. This model is capable of separating the oscillating patterns and non-stationary variations present in the measured data. Moreover, we recognize that not all observed variables have the same level of uncertainty, and therefore, we independently model the measurement noise for each variable. To incorporate prior information and obtain corresponding posterior distributions, we estimate the proposed model using a variational Bayesian framework. This framework allows us to account for uncertainties and leverage prior knowledge effectively.

Chapter 5 focuses on two key challenges in the process industries. First, quality-related variables are difficult to measure using sensors due to physical and financial limitations. As a result, they are less frequently available compared to other variables, often requiring time-consuming laboratory analysis. Second, process data displays non-linearity due to factors such as complex reaction kinetics, phase transitions, and mass/heat transfer limitations. To tackle these challenges, we propose a novel neural network architecture with gated recurrent units under a variational inference framework. The effectiveness of this approach is assessed on both simulated and industrial datasets.

The previous contributions made in this thesis have been based on an assumption that the measurement noise follows a Gaussian distribution, which allows for a closed-form solution. However, when dealing with industrial process data, it is common to encounter measurement problems like outliers and skewed noise. If these issues are not explicitly addressed, they can significantly hinder the performance of the extracted features. Chapter 6 addresses this by considering a Skewed t-distribution for the measurement noise in the complex slow feature model. The parameters of the model are jointly estimated using the expectation-maximization algorithm.

Chapter 7 introduces a novel approach to probabilistic latent variable models that address two crucial factors: the choice of the number of latent variables and the accuracy of the noise model for complex data. The model employs a Laplace distribution to automatically determine the dimensionality of the latent space, resulting in a sparse model. The hierarchical representation of these distributions enables feasible solutions for the latent variables and model parameters. Unlike Chapter 6, the parameters are treated as latent variables and their posterior distributions are estimated using variational Bayesian inference.

Chapter 8 of this thesis introduces the final contribution of this thesis, which focuses on integrating physics principles into the probabilistic slow feature model. Industrial processes have physical constraints, such as energy requirements, equipment limitations, conservation laws, symmetry properties, specific functional forms of relationships, and safety considerations. Previous enhancements proposed in the thesis may result in physically inconsistent or unacceptable results due to the black-box nature of the slow feature model. Although the Bayesian framework allows for the incorporation of expert information through prior distributions, the resulting model may still not align with known physics principles. In response, Chapter 8 proposes a methodology that incorporates two types of physical constraints (linear algebraic equality and inequality constraints) into the probabilistic slow feature model. The model parameters are estimated using the expectation-maximization approach.

The thesis concludes with Chapter 9, which serves as the final chapter. In this chapter, the conclusions derived from the different models and algorithms developed throughout the thesis are summarized. Additionally, the potential areas for future work are outlined.

Chapter 2

Mathematical Background

This chapter aims to provide a thorough analysis and exploration of the fundamental concepts, ideas, and methodologies that will serve as the foundation for forthcoming chapters. These aspects are aimed to equip readers with the necessary knowledge and tools to effectively engage with the subsequent contributions in a meaningful and insightful manner.

2.1 Probabilistic modelling with unknown parameters

Modeling a random variable using a probability distribution with a non-random unknown parameter is a common approach in statistical analysis [57–59]. In this type of modeling, the random variable is assumed to follow a specific probability distribution, but one or more parameters of that distribution are treated as non-random or deterministic unknowns. These parameters are fixed values that need to be estimated from the available data. Throughout the thesis, various probability distributions have been utilized for different purposes. These distributions have been chosen based on their appropriateness for specific scenarios and the characteristics of the random variables under investigation. Each chosen probability distribution provides a mathematical representation of the random variable, where the non-random unknown parameter(s) play a critical role. The following section provides descriptions of several probability distributions that have been employed for diverse purposes in the different contributions of this thesis.

2.1.1 Gaussian Distribution

The Gaussian distribution, also known as the normal distribution, is a fundamental probability distribution commonly used in statistical analysis and modeling. It is characterized by its bell-shaped curve, with the mean as its center and the standard deviation determining its spread. In many real-world phenomena, such as measurement errors or natural variations, the Gaussian distribution often provides a good approximation. It is widely utilized due to its mathematical tractability and several important properties. The Gaussian distribution is fully defined by two parameters: the mean (μ) and the standard deviation (σ). The mean represents the central tendency of the distribution, while the standard deviation measures the dispersion or variability of the data points around the mean. The probability density function (PDF) of a Gaussian distribution is given by (2.1):

$$f(x) = \mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2.1}$$

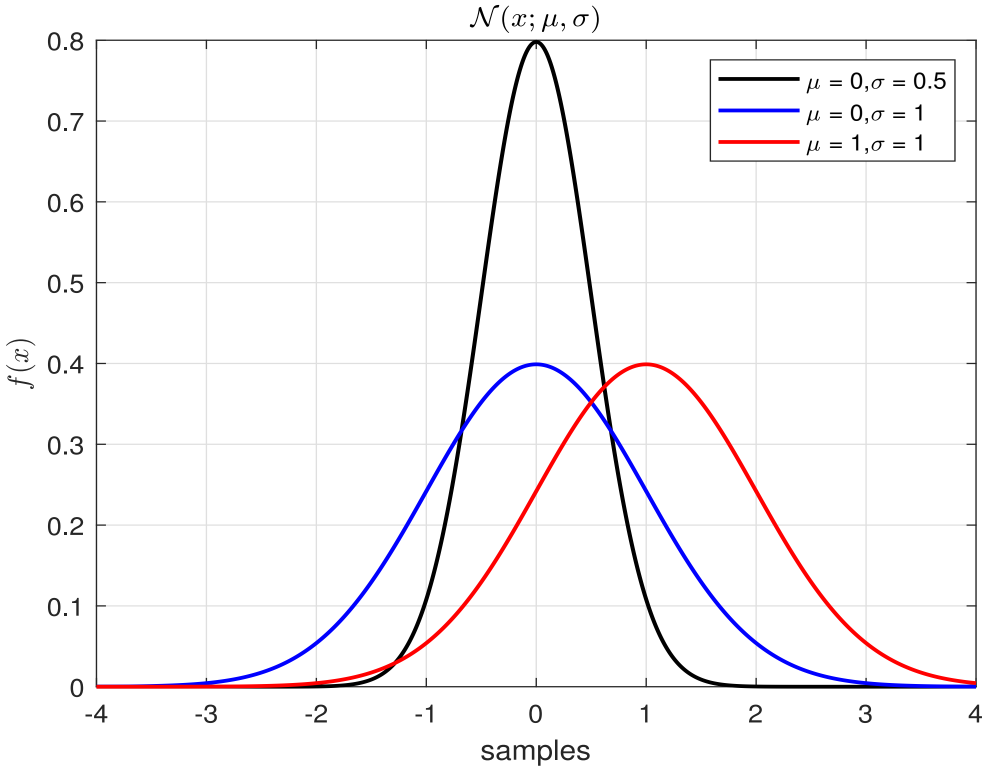


Figure 2.1: PDFs of a Gaussian distribution

As shown in Fig. 2.1, a higher value of μ moves the distribution to the right, while a lower value of μ shifts it to the left. Increasing the standard deviation σ makes the distribution wider and flatter, resulting in a broader curve with a lower peak. Conversely, decreasing σ makes the distribution narrower and taller, with a higher peak. We now present the derivation to calculate the expected value and the variance of a Gaussian distributed random variable.

- **Expected value of x:**

$$\begin{aligned}
\mathbb{E}(x) = \langle x\rangle &= \int_{-\infty}^{\infty} x f(x)dx \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} x e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}dx \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} (\sigma z+\mu)e^{-\frac{z^2}{2}}dz \ \left(\text{where } z = \frac{x-\mu}{\sigma} \implies dx = \sigma dz\right) \\
&= \frac{\sigma}{\sqrt{2\pi}}\cancelto{0}{\int_{-\infty}^{\infty} ze^{-\frac{z^2}{2}}dz} + \frac{\mu}{\sqrt{2\pi}}\cancelto{\sqrt{2\pi}}{\int_{-\infty}^{\infty} e^{-\frac{z^2}{2}}dz} \implies \mu
\end{aligned} \tag{2.2}$$

Since the integrand of the first integral is an odd function, its integral over a symmetric interval is zero. The second term relies on the property that the integral of the normal distribution evaluates to 1 i.e.,

$$\int_{-\infty}^{\infty} f(x)dx = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}dx = 1 \tag{2.3}$$

- **Variance of x:**

$$\begin{aligned}
\mathrm{Var}(x) &= \mathbb{E}(x^2) - \mathbb{E}(x)^2 \\
&= \int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}dx - \mu^2 \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} (\sigma z+\mu)^2 e^{-\frac{z^2}{2}}dz - \mu^2 \ \left(\text{where } z = \frac{x-\mu}{\sigma} \implies dx = \sigma dz\right) \\
&= \frac{\sigma^2}{\sqrt{2\pi}}\int_{-\infty}^{\infty} z^2 e^{-\frac{z^2}{2}}dz + \frac{\mu^2}{\sqrt{2\pi}}\cancelto{\sqrt{2\pi}}{\int_{-\infty}^{\infty} e^{-\frac{z^2}{2}}dz} + \frac{2\mu\sigma}{\sqrt{2\pi}}\cancelto{0}{\int_{-\infty}^{\infty} ze^{-\frac{z^2}{2}}dz} - \mu^2
\end{aligned} \tag{2.4}$$

To evaluate the integral, we substitute $\frac{1}{2}$ with α

$$= \sigma^2\sqrt{\frac{\alpha}{\pi}}\int_{-\infty}^{\infty} z^2 e^{-\alpha z^2}dz = -\sigma^2\sqrt{\frac{\alpha}{\pi}}\int_{-\infty}^{\infty}\frac{d}{d\alpha}e^{-\alpha z^2}dz$$

$$= -\sigma^2\sqrt{\frac{\alpha}{\pi}}\frac{d}{d\alpha}\cancelto{\sqrt{\frac{\pi}{\alpha}}}{\int_{-\infty}^{\infty} e^{-\alpha z^2}dz} \quad \text{(Using (2.3))} \implies \sigma^2$$

2.1.2 Gamma Distribution

The Gamma distribution is a probability distribution that is used to model continuous, positive-valued random variables. It is characterized by two parameters: shape (α) and rate parameter (β). The PDF of the Gamma distribution is given by (2.5).

$$p(x;\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1}\, e^{-\beta x} \tag{2.5}$$

where Γ represents the gamma function. Higher values of α result in a more peaked and less-skewed distribution, while lower values lead to a right-skewed distribution. Lower values of β lead to a faster decay of the distribution, and vice versa, as shown in Fig. 2.2.

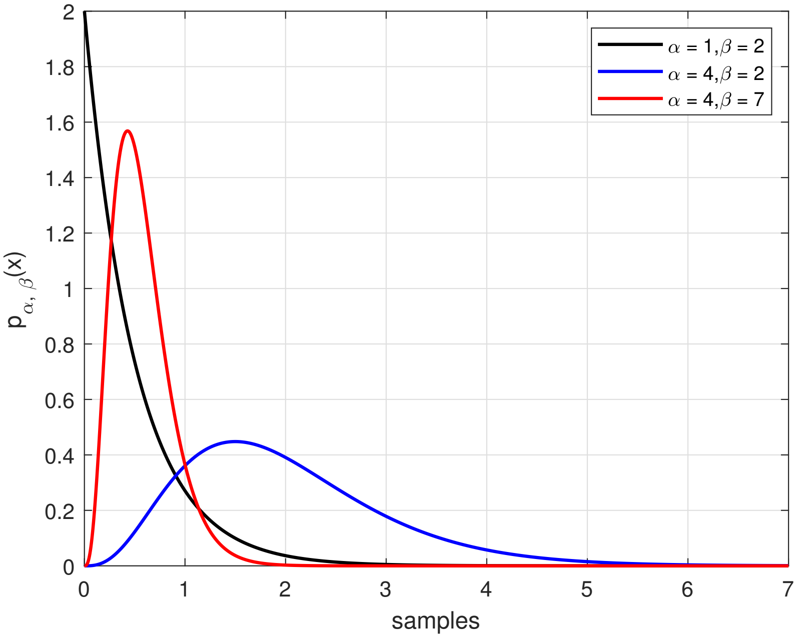


Figure 2.2: PDFs of a Gamma distribution

The important moments of the Gamma distribution can be derived using (2.2) - (2.4), but for brevity, the final equations are presented below.

$$\langle x \rangle = \frac{\alpha}{\beta} \tag{2.6}$$

$$\mathrm{Var}(x) = \frac{\alpha}{\beta^2} \tag{2.7}$$

$$\langle \ln x \rangle = \psi(\alpha) - \ln(\beta) \tag{2.8}$$

where ψ is the digamma function.

2.1.3 Beta Distribution

The Beta distribution is a continuous probability distribution defined on the interval $\begin{bmatrix}0 & 1\end{bmatrix}$. It is characterized by two shape parameters, typically denoted as alpha (α) and beta (β), which determine the shape and behavior of the distribution. The PDF of the Beta distribution is given by (2.9).

$$p(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{2.9}$$

where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha)+\Gamma(\beta)}$. The combination of α and β determines the shape, skewness, and location of the peak (mode) of the Beta distribution curve, as shown in Fig. 2.3. When $\alpha = \beta$, the distribution is symmetric around its center. As the values of α and β differ, the distribution becomes skewed. Further, larger values of α and β result in a narrower and more concentrated distribution, indicating less variability.

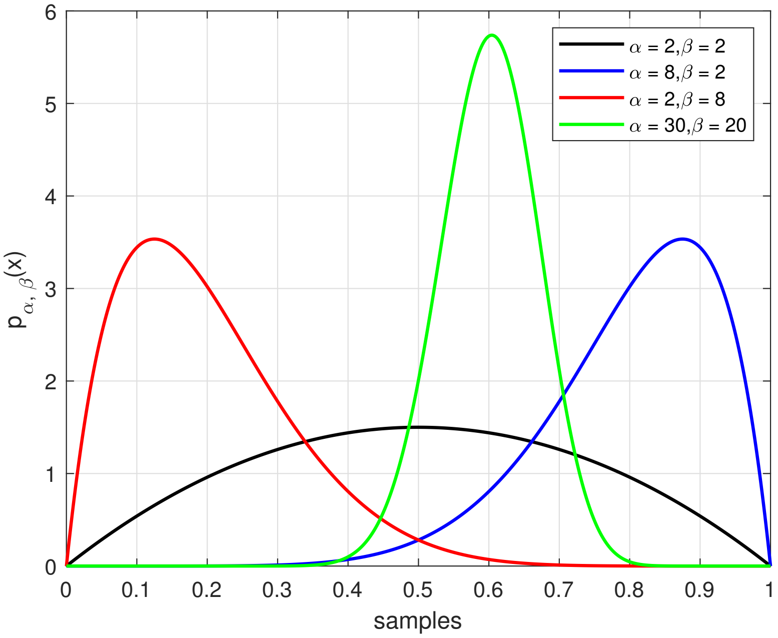


Figure 2.3: PDFs of a Beta distribution

The mean and variance of the Beta distribution are shown below.

$$\langle x \rangle = \frac{\alpha}{\alpha+\beta} \tag{2.10}$$

$$\mathrm{Var}(x) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{2.11}$$

2.1.4 Laplace Distribution

The Laplace distribution, also known as the double-exponential distribution, is characterized by its symmetric bell-shaped curve with heavy tails, similar to the normal distribution. However, unlike the normal distribution, the Laplace distribution has a sharper peak and a more rapid decay in the tails, making it suitable for modeling data with abrupt changes or outliers. The PDF of the Laplace distribution is given by (2.12)

$$p(x;\mu,b)=\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right) \tag{2.12}$$

where μ is the location parameter (representing the center of the distribution), and b is the scale parameter (controlling the spread or width of the distribution). Shifting the location parameter horizontally moves the center of the distribution. The scale parameter controls the spread of the distribution, as shown in Fig. 2.4. In the context of this thesis, the Laplace distribution is used to encourage sparse solutions. Sparsity refers to the property where a significant portion of the elements in a signal or parameter vector are exactly zero or very close to zero.

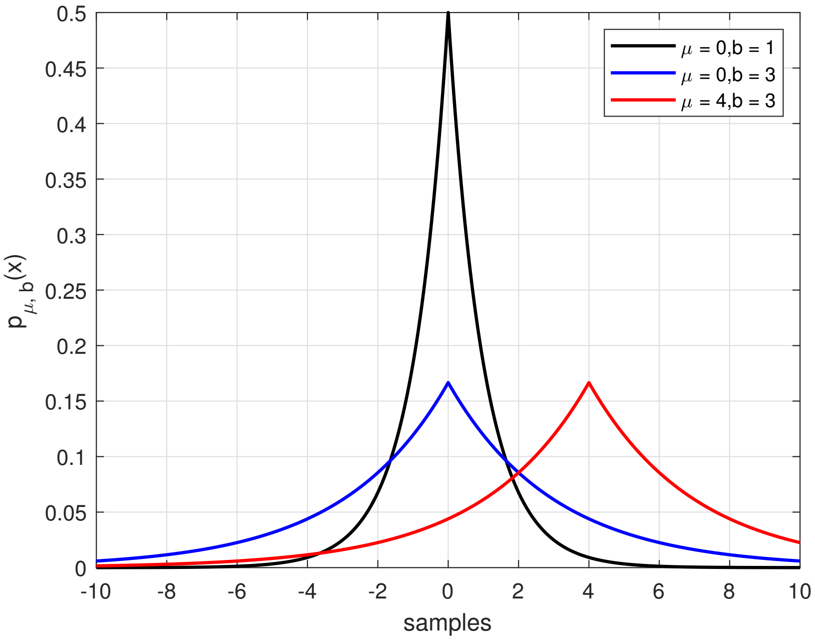


Figure 2.4: PDFs of a Laplace distribution

2.1.5 Truncated Gaussian Distribution

The truncated Gaussian distribution is a probability distribution that is derived from the Gaussian (normal) distribution but with the restriction that values outside a certain range are excluded or "truncated." This truncation is typically imposed to reflect real-world constraints or limitations in the data or application. The PDF of the truncated Gaussian distribution is given by (2.13).

$$p(x;\mu,\sigma,a,b) = \frac{1}{\sigma}\frac{\Phi\left(\frac{x-\mu}{\sigma}\right)}{\Psi\left(\frac{b-\mu}{\sigma}\right) - \Psi\left(\frac{a-\mu}{\sigma}\right)} \tag{2.13}$$

where $\Phi(\xi) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\xi^2}{2}\right)$ is the PDF of the standard normal distribution and $\Psi(z) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)\right)$ is its cumulative distribution function. Here μ and σ are the mean and standard deviation of the underlying Gaussian distribution, a and b are the lower and upper bounds. The behavior of the truncated distribution within $\begin{bmatrix}0 & 1\end{bmatrix}$ for different values of the μ and σ is shown in Fig. 2.5.

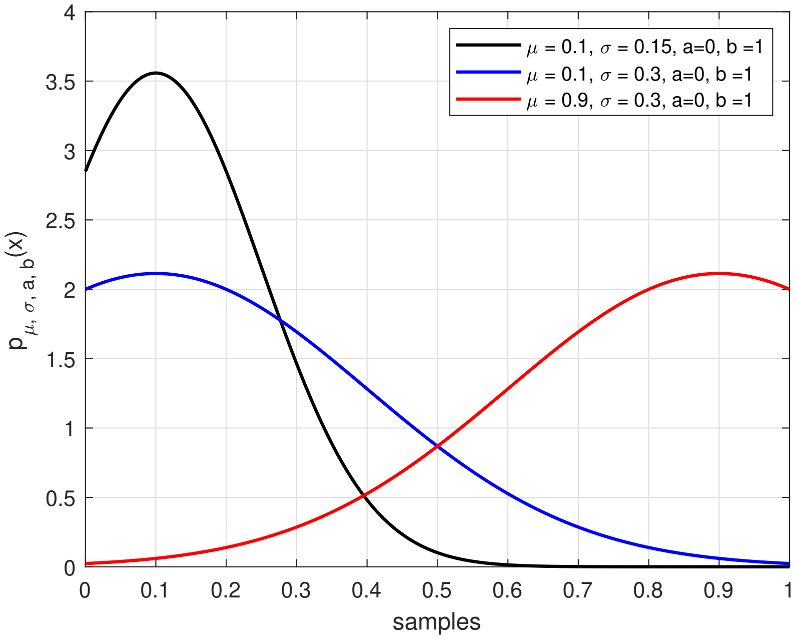


Figure 2.5: PDFs of a truncated Gaussian distribution

The mean and the variance of the truncated normal distribution can be calculated as follows:

$$\langle x\rangle = \mu - \sigma\frac{\Phi(\beta) - \Phi(\alpha)}{\Psi(\beta) - \Psi(\alpha)} \tag{2.14}$$

$$\mathrm{Var}(x) = \sigma^2 \left(1 - \frac{\beta\Phi(\beta) - \alpha\Phi(\alpha)}{\Psi(\beta) - \Psi(\alpha)} - \left(\frac{\Phi(\beta) - \Phi(\alpha)}{\Psi(\beta) - \Psi(\alpha)}\right)^2\right) \tag{2.15}$$

where $\alpha = \frac{a-\mu}{\sigma}$ and $\beta = \frac{b-\mu}{\sigma}$.

2.1.6 Skew Normal Distribution

The skew normal distribution is a probability distribution that extends the normal distribution by introducing a skewness parameter. It is commonly used to model asymmetric data in various fields. The PDF of the skew normal distribution is given by (2.16).

$$p(x; \xi, \omega, \alpha) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\psi\left(\lambda\left(\frac{x-\xi}{\omega}\right)\right) \tag{2.16}$$

where ξ, ω, and λ affect the location, scale and shape of the distribution. When $\lambda = 0$, the distribution reduces to the normal distribution. Positive values of α correspond to right-skewed distributions, while negative values indicate left-skewness, as shown in Fig. 2.6.

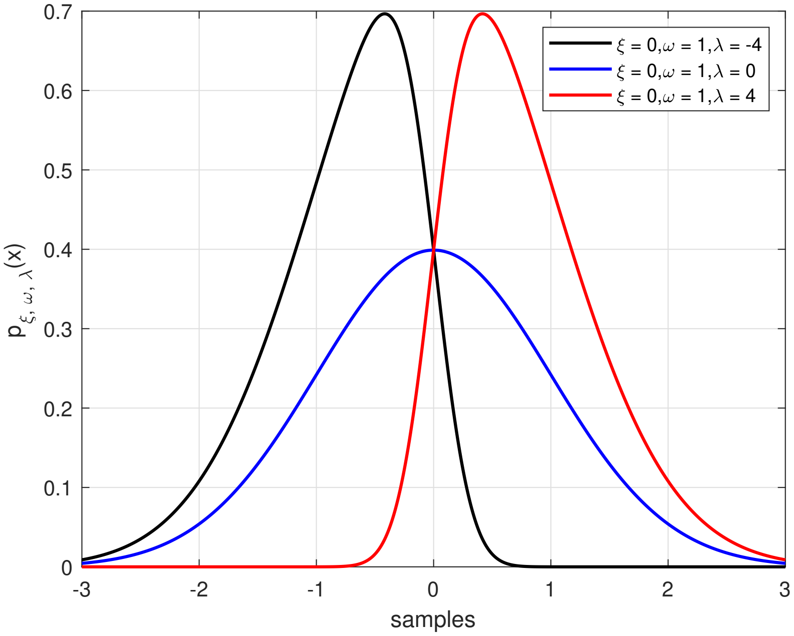


Figure 2.6: PDFs of a skew normal distribution

2.1.7 Skewed t-distribution

The Skewed-t distribution is a statistical distribution that combines the properties of the Student's t-distribution with skewness. It is commonly used in statistical modeling and data analysis to account for skewness and heavy tails in the data. The probability density function of the Skewed-t distribution is defined by four parameters: degrees of freedom (ν), skewness (λ), location (μ), and scale (σ). Similar to the t-distribution, the degrees of freedom parameter governs the tail behavior of the distribution. Higher values of ν result in thinner tails, approaching a normal distribution, while lower values lead to heavier tails, as shown in Fig.2.7. The remaining parameters μ, σ, and λ serve the same purpose as discussed in the previous subsections. The PDF of the skewed t-distribution is not included due to its complexity and will not be used in this thesis. However, we can utilize the established result that a skewed t-distribution can be expressed as a Gaussian scale mixture, as explained in Chapter 6.

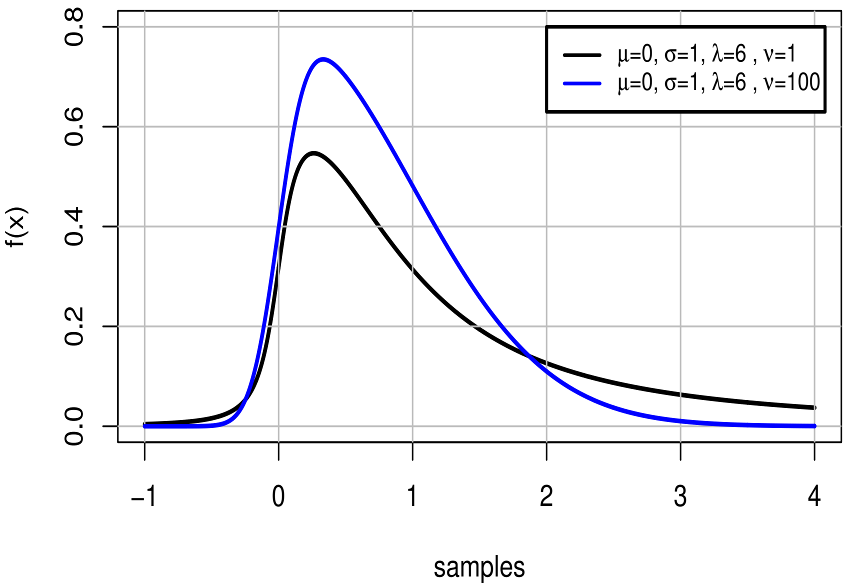


Figure 2.7: PDFs of a skewed t-distribution

2.2 Maximum likelihood Estimation

The maximum likelihood estimation (MLE) [57, 60] method is a widely used statistical technique for estimating the parameters of a probability distribution. It provides a rigorous framework for making inferences about unknown parameters based on observed data. The fundamental idea behind the MLE method is to find the set of parameter values that maximize the likelihood function, which quantifies the probability of observing the given data under different parameter settings. Intuitively, the MLE seeks to find the most likely values of the parameters that explain the observed data in the best possible way. By maximizing the likelihood function, we are effectively finding the parameter values that make the observed data most probable.

To formalize the MLE method, let's consider a parametric probability distribution with unknown parameters. Let θ represent the vector of parameters, and let $X = \big(x_1, x_2, \cdots \quad x_N\big)$ be a set of N independent and identically distributed (i.i.d.) observations from this distribution. The likelihood function $L(\theta|X)$ represents the probability of observing the data X under the parameter values θ. For a continuous distribution, it is defined as the product of the probability density function evaluated at each observation:

$$L(\theta|X) = \prod_{n=1}^{N} f(x_n; \theta)$$

where $p(x_n; \theta)$ is the PDF of the distribution. The maximum likelihood estimation problem can be formulated as follows:

$$\theta^* = \operatorname{Arg}\max_{\theta} \quad L(\theta|X)$$

where θ^* represents the estimated parameter values that maximize the likelihood. To simplify the optimization problem, it is common to take the logarithm of the likelihood function, resulting in the log-likelihood:

$$l(\theta|X) = \ln L(\theta|X) = \sum_{n=1}^{N} \ln p(x_n; \theta)$$

The log-likelihood function has the same maximum as the likelihood function, but it simplifies the calculations and avoids numerical underflow issues when dealing with small probabilities. The estimation problem can be further enhanced by introducing constraints on the parameter space. This is particularly relevant when dealing with bounded parameters or when prior knowledge about the parameters is available. In such cases, the maximum likelihood estimation problem becomes a constrained optimization problem [61–63], and additional techniques, such as Lagrange multipliers or numerical optimization algorithms, may be required.

2.3 Probabilistic modelling with a latent variable

Traditionally, probabilistic models have focused on estimating unknown parameters to capture the underlying structure of the measured data. However, in numerous real-world applications, the measured data is influenced by factors that are not directly observable. These hidden factors, or latent variables, play a crucial role in shaping the observed data distribution. The integration of latent variables [64] provides a flexible framework, as discussed below.

- **Capturing complex relationships:** Probabilistic modeling with latent variables [65–67] enables us to capture intricate relationships between observed variables and hidden factors. These hidden variables act as abstract representations that encode important but unobserved information. Latent variable modelling provides a flexible framework to model various types of dependencies, including nonlinear relationships, hierarchical structures, and context-specific patterns, leading to a more accurate and comprehensive understanding of the underlying processes.

- **Dealing with uncertainties [68–70]:** In many real-world scenarios, uncertainties are inherent and unavoidable. Probabilistic modeling with latent variables allows us to explicitly model and quantify uncertainties in our data. By incorporating uncertainty measures, such as probability distributions, we can make robust predictions and perform reliable inferences. This ability to handle uncertainties is particularly valuable in applications such as decision-making, risk assessment, and anomaly detection.

- **Enhanced interpretability [71]:** Probabilistic modeling with latent variables can enhance the interpretability of our models. By explicitly representing hidden factors, we gain insights into the driving forces behind the observed data. Latent variables can be interpreted as meaningful features or attributes that contribute to the generation or organization of the observed data.

- **Learning from incomplete data [72–75]:** In many real-world scenarios, data may be incomplete or have missing values. Probabilistic modeling with latent variables offers a principled approach for handling such incomplete data. The latent variables act as a bridge, connecting the observed variables with the missing information. Through probabilistic inference, we can infer the values of missing data points.

To formalize the problem, let's consider a set of observed variables denoted by x and a set of latent variables denoted by z. Given an observed value x^* of a random variable x, in a Bayesian approach [76], there are two goals, as discussed below.

- The first aim of a Bayesian approach is to evaluate the logarithm of the marginal likelihood (also called log model evidence) of the observed data, denoted by $\ln p(x)$. The model or hypothesis with a higher logarithm of the marginal probability is considered to be more likely or more supported by the data.

$$p(x) = \int p(x, z)dz \tag{2.17}$$

- The subsequent crucial objective is to determine the conditional distribution $p(z|x)$ of a latent variable. This conditional distribution is known as the posterior distribution. It combines our prior beliefs about z with the information provided by x^*. The posterior distribution is derived using Bayes' theorem, as shown in (2.18).

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2.18}$$

Due to the presence of numerous unobserved variables in a model, the integration of the right-hand side of (2.17) can be analytically challenging or even impossible. In order to fulfill the dual objectives of a Bayesian approach, Variational Bayes (VB) offers an alternative by substituting the integration problem with an optimization problem. The forthcoming section will elaborate on the intricacies and mechanics of this substitution.

2.4 Mean-field variational Inference

Variational Bayesian inference [77–79] is a powerful computational technique used to approximate the posterior distribution of latent variables and the log marginal likelihood of observed data in Bayesian models. In essence, the primary objective of variational Bayesian inference is to discover an approximate distribution $q(z)$ that closely resembles the true posterior distribution $p(z|x)$. The extent of this closeness is evaluated using the Kullback-Leibler (KL) divergence [80], as illustrated in the following expression.

$$\begin{aligned} \min_{q(z)} \quad & \mathrm{KL}\left(q(z)||p(z|x)\right) \\ \text{s.t.} \quad & \int q(z)dz = 1 \end{aligned} \tag{2.19}$$

The KL-divergence term is defined as

$$\mathrm{KL}\left(q(z)||p(z|x)\right) = \int q(z)\ln\left(\frac{q(z)}{p(z|x)}\right)\,dz \tag{2.20}$$

It can be understood as a non-symmetric measure of the difference between two probability distributions. It is important to note that the KL-divergence is always greater than or equal to zero. It equals zero only when $q(z)$ and $p(z|x)$ are identical, indicating perfect similarity. The KL-divergence term can be simplified as shown below.

$$\begin{aligned} \mathrm{KL}\left(q(z)||p(z|x)\right) &= \int q(z)\ln\left(\frac{q(z)}{p(z|x)}\right)\,dz \\ &= \int q(z)\ln\left(\frac{p(x)q(z)}{p(z,x)}\right)\,dz \\ &= \int q(z)\ln p(x)\,dz - \int q(z)\ln\left(\frac{p(z,x)}{q(z)}\right)\,dz \\ &= \ln p(x) - \mathrm{F}(q(z)) \end{aligned} \tag{2.21}$$

where $F(q(z))$ is called the variational free energy. The non-negativity of KL-divergence implies that the variational free energy is always less than or equal to the log model evidence. Hence, variational free energy is also referred to as evidence lower bound (ELBO).

$$\mathrm{KL}\left(q(z)||p(z|x)\right) \geq 0 \implies \mathrm{F}(q(z)) \leq \ln p(x) \tag{2.22}$$

Since the log model evidence is a fixed quantity, solving the maximization problem shown in (2.23) instead has two outcomes.

$$\begin{aligned} q^*(z) = \operatorname{Arg}\max_{q(z)} \quad & \mathrm{F}\left(q(z)\right) \\ \text{s.t.} \quad & \int q(z)dz = 1 \end{aligned} \tag{2.23}$$

- First, the lower bound on the log model evidence becomes tighter, improving the approximation of the variational free energy to the log model evidence.
- The KL-divergence between the true posterior and its variational approximation decreases, enhancing the similarity between the $q(z)$ and $p(z|x)$.

Given the absence of a closed-form solution for the true posterior, as previously elaborated in Sec. 2.3, a restricted family of distributions $q(z)$ is considered instead. The member of this family, for which the KL divergence is minimized, is sought. The mean-field approximation is a popular choice to restrict the family of distributions. It assumes that the approximate posteriors of each latent variable $z^{(i)}$ $\forall 1 \leq i \leq p$ are independent. Therefore,

$$q(z) = \prod_{j=1}^{p} q(z^{(j)}) = q(z^{(i)}) \prod_{\substack{j=1 \\ j \neq i}}^{p} q(z^{(j)}) = q(z^{(i)})q(z^{(\sim i)}) \tag{2.24}$$

where $q(z^{(\sim i)})$ refers to the joint distribution of hidden variables except $z^{(i)}$. Further, the evidence lower bound can be decomposed as shown below.

$$\begin{aligned} F(q) &= \int q(z) \ln\left(\frac{p(x,z)}{q(z)}\right) dz \qquad (2.25) \\ &= \int q(z^{(i)})q(z^{(\sim i)}) \ln p(x,z) dz^{(i)}\, dz^{(\sim i)} \\ &\qquad\qquad - \int q(z^{(i)})q(z^{(\sim i)}) \ln\left(q(z^{(i)})q(z^{(\sim i)})\right) dz^{(i)}\, dz^{(\sim i)} \\ &= \int_{z^{(i)}} q(z^{(i)}) \left[\int_{z^{(\sim i)}} q(z^{(\sim i)}) \ln p(x,z) dz^{(\sim i)}\right] dz^{(i)} \end{aligned}$$

$$
\begin{aligned}
&- \left[\int_{z^{(i)}} q(z^{(i)}) \ln q(z^{(i)}) dz^{(i)}\right] \cancelto{1}{\left[\int_{z^{(\sim i)}} q(z^{(\sim i)}) dz^{(\sim i)}\right]} \\
&\quad - \cancelto{1}{\left[\int_{z^{(i)}} q(z^{(i)}) dz^{(i)}\right]} \cancelto{c}{\left[\int_{z^{(\sim i)}} q(z^{(\sim i)}) \ln q(z^{(\sim i)}) dz^{(\sim i)}\right]} \\
&= \int_{z^{(i)}} q(z^{(i)}) \left[\int_{z^{(\sim i)}} q(z^{(\sim i)}) \ln p(x, z) dz^{(\sim i)} dz^{(\sim i)} - \ln q(z^{(i)})\right] dz^{(i)} + c \\
&= \int_{z^{(i)}} q(z^{(i)}) \left(\langle \ln p(x, z)\rangle_{q(z^{(\sim i)})} - \ln q(z^{(i)})\right) + c
\end{aligned}
\tag{2.26}
$$

where c is a constant independent of $q(z^{(i)})$. The Lagrange function is constructed, as shown in (2.27), to solve the constrained optimization in (2.23).

$$
\begin{aligned}
L(q) &= \int_{z^{(i)}} q(z^{(i)}) \left(\langle \ln p(x, z)\rangle_{q(z^{(\sim i)})} - \ln q(z^{(i)})\right) + c + \lambda \left(\int q(z^{(i)}) dz^{(i)} - 1\right) \qquad (2.27) \\
&= \int_{z^{(i)}} q(z^{(i)}) \left(\langle \ln p(x, z)\rangle_{q(z^{(\sim i)})} - \ln q(z^{(i)}) + \lambda\right)\ dz^{(i)}\ + c - \lambda
\end{aligned}
$$

As $L(q(z))$ is functional, the maximum is obtained by finding functions for which the functional derivative is equal to zero. Say $L(q) = \int f(x, q(x), q'(x))\, dx$, then $L(q)$ has a stationary value if the Euler-Lagrange differential equation is satisfied

$$
\frac{\partial f}{\partial q} - \frac{d}{dx}\left\{\frac{\partial f}{\partial q'}\right\} = 0
$$

Hence

$$
\begin{aligned}
\frac{\partial}{\partial q(z^{(\sim i)})} \left\{q(z^{(i)}) \left(\langle \ln p(x, z)\rangle_{q(z^{(\sim i)})} - \ln q(z^{(i)}) + \lambda\right)\right\} &= 0 \\
\langle \ln p(x, z)\rangle_{q(z^{(\sim i)})} - 1 - \ln q(z^{(i)}) + \lambda &= 0 \\
\ln q(z^{(i)}) &= \langle \ln\ p(x, z)\rangle_{q(z^{(\sim i)})} + \ln \exp\{\lambda - 1\} && (2.28) \\
q(z^{(i)}) &\propto \exp\left(\langle \ln\ p(x|z^{(i)}, z^{(\sim i)})\rangle_{q(z^{(\sim i)})}\right) p(z^{(i)}) && (2.29)
\end{aligned}
$$

The equation shown in (2.29) serves as a fundamental tool that is widely employed in deriving numerous posterior distributions in subsequent academic contributions. The variational inference algorithm is summarized in Fig. 2.8.

Note: Variational Bayesian inference can be closely related to the widely recognized Expectation-Maximization (EM) algorithm [81, 82], particularly under specific conditions, as elucidated in the following discussion. Assuming the model involves both

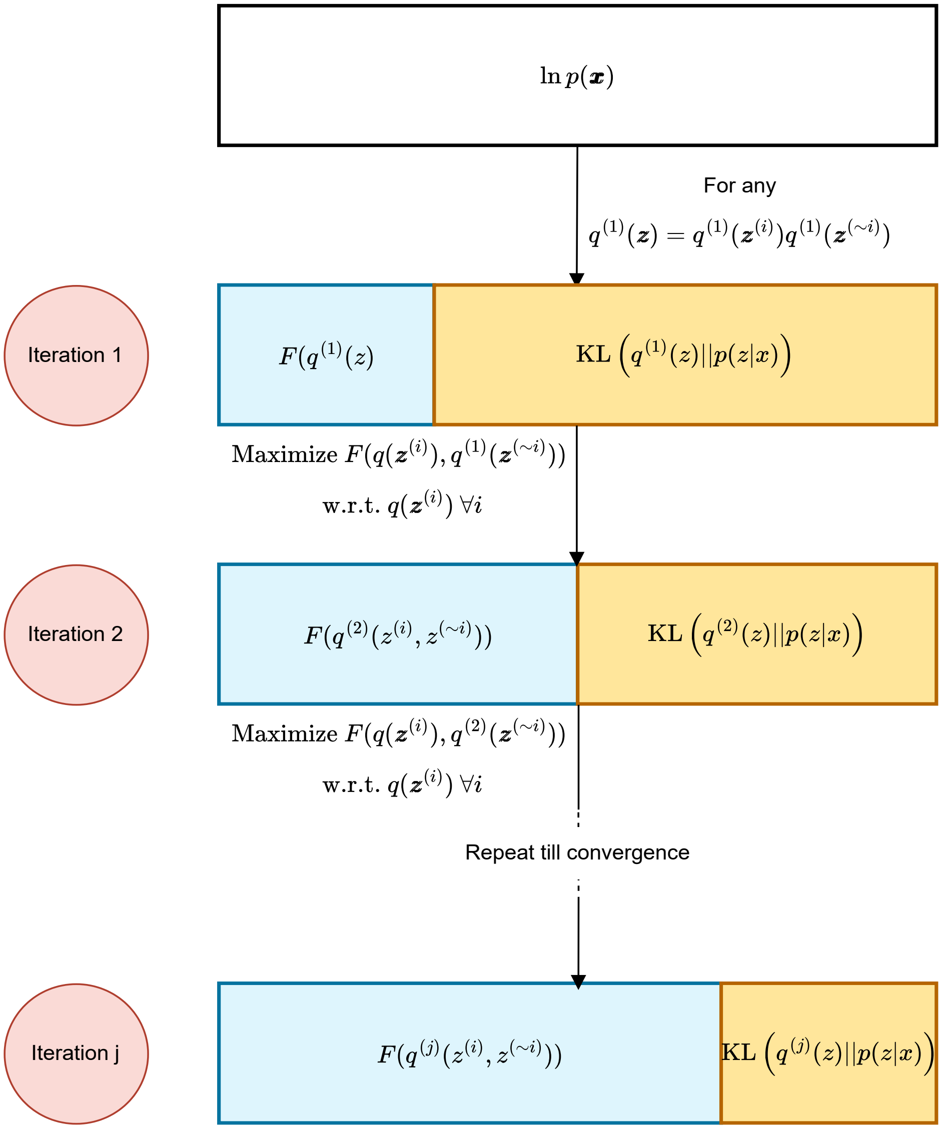

$\ln p(\boldsymbol{x})$
For any
$q^{(1)}(\boldsymbol{z}) = q^{(1)}(\boldsymbol{z}^{(i)})q^{(1)}(\boldsymbol{z}^{(\sim i)})$
Iteration 1
$F(q^{(1)}(z)$
$\mathrm{KL}\left(q^{(1)}(z)||p(z|x)\right)$
Maximize $F(q(\boldsymbol{z}^{(i)}), q^{(1)}(\boldsymbol{z}^{(\sim i)}))$
w.r.t. $q(\boldsymbol{z}^{(i)})\ \forall i$
Iteration 2
$F(q^{(2)}(z^{(i)}, z^{(\sim i)}))$
$\mathrm{KL}\left(q^{(2)}(z)||p(z|x)\right)$
Maximize $F(q(\boldsymbol{z}^{(i)}), q^{(2)}(\boldsymbol{z}^{(\sim i)}))$
w.r.t. $q(\boldsymbol{z}^{(i)})\ \forall i$
Repeat till convergence
Iteration j
$F(q^{(j)}(z^{(i)}, z^{(\sim i)}))$
$\mathrm{KL}\left(q^{(j)}(z)||p(z|x)\right)$

Figure 2.8: **Variational Inference algorithm.** The sum of $F(q(\boldsymbol{z}))$ and $KL(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x}))$ is always a constant with respect to $q(\boldsymbol{z})$. The variational free energy $F(q^{(j)}(\boldsymbol{z}))$ increases with each iteration j until convergence is achieved

latent variables and unknown parameters θ, the equation in (2.21) can be reformulated as shown below:

$$\ln p(x;\theta) = \mathrm{KL}\left(q(z)||p(z|x)\right) + \mathrm{F}(q(z),\theta) \implies \mathrm{F}(q(z),\theta) \leq \ln p(x;\theta) \tag{2.30}$$

Consequently, the optimization problem to be addressed can be expressed as (2.31).

$$\max_{q(z),\,\theta} \quad \mathrm{F}\left(q(z),\theta\right) \tag{2.31}$$

However, solving this optimization problem analytically for both $q(z)$ and θ simultaneously proves to be infeasible due to its complexity. Thus, the complex problem is decomposed, and an iterative stepwise analytical solution is pursued until convergence by fixing one entity while optimizing the other. The two steps of the expectation-maximization algorithm are elaborated on in the subsequent discussion.

- **E-step**: Given θ^{old}, the first step involves maximizing the variational free energy with respect to the approximate distribution $q(z)$. The specific problem of interest is expressed below.

  $$\operatorname{Arg}\max_{q(z)} \quad \mathrm{F}\left(q(z),\theta^{\text{old}}\right)$$

  Assuming that the posterior is perfectly tractable, the free energy is maximized when the approximate posterior is equal to the true posterior.

  $$q^{*}(z) = p(z|x;\theta^{\text{old}}) \tag{2.32}$$

- **M-step**: Once $q^{*}(z)$ is determined, the second objective is to maximize the free energy with respect to the parameter θ. The corresponding mathematical problem of interest is presented below.

  $$\operatorname{Arg}\max_{\theta} \quad \mathrm{F}\left(q^{*}(z),\theta^{\text{old}}\right)$$

  From (2.25),

  $$\begin{aligned} F(q^{*}(z),\theta) &= \int q^{*}(z)\ln\left(\frac{p(x,z;\theta)}{q^{*}(z)}\right)dz \\ &= \int q^{*}(z)\ln p(x,z;\theta)dz + \int q^{*}(z)\ln q^{*}(z)dz \end{aligned}$$

$$= \langle \ln p(x,z;\theta) \rangle_{q^*(z)} + c$$

where c represents a constant independent of θ. Consequently, the optimization problem can be reformulated as follows:

$$\theta^{\text{new}} = \operatorname{Arg}\max_{\theta} \quad \langle \ln p(x,z;\theta) \rangle_{q^*(z)} \tag{2.33}$$

2.5 Importance Sampling

Conjugate priors possess the crucial property of yielding posterior distributions that belong to the same parametric family as the prior distribution. This property facilitates analytical tractability and simplifies the variational inference process, as the optimal variational approximation can be derived in closed form. However, a fundamental challenge arises when we encounter scenarios where the use of non-conjugate priors is either desirable or inevitable. The explicit knowledge of the normalizing constant is often intractable for non-conjugate priors. In many applications, the principal reason for estimating the posterior distribution is to evaluate the expected value of some function $f(z)$ under the probability distribution $p(z)$, as shown in (2.34).

$$\langle f(z) \rangle = \int f(z)p(z)dz \tag{2.34}$$

We assume that evaluating such expectations exactly through analytical techniques is intractable. To address this, sampling methods are employed to obtain a set of independent samples, $z(l)$(where $l = 1, ..., L$), drawn from the distribution $p(z)$. This enables the approximation of the expectation (11.1) through a finite sum."

$$\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f(z(l)) \tag{2.35}$$

where L denotes the number of drawn samples and $\hat{f}$ is the basic Monte Carlo estimator of $\langle f(z) \rangle$. Suppose we seek to sample from a challenging distribution $p(z)$, which is difficult to directly sample from, under the assumption that the knowledge of $p(z)$ is limited to $\tilde{p}(z)$ with an unknown normalizing constant Z_p.

$$p(z) = \frac{1}{Z_p}\tilde{p}(z) \tag{2.36}$$

where $\tilde{p}(z)$ can readily be evaluated. By leveraging an easy-to-sample/proposal distribution $q(z)$, which can be chosen independently, the importance sampling [78] framework enables the estimation of the statistical quantities of interest. The expected value of the function $f(z)$ can be computed as shown below.

$$\hat{f} = \sum_{l=1}^{L} f(z(l))\hat{w}_l \tag{2.37}$$

where

$$\hat{w}_l = \frac{\tilde{w}_l}{\sum_{l=1}^{L} \tilde{w}_l} \text{ and } \tilde{w}_l = \frac{\tilde{p}(\nu(l))}{\tilde{q}(z(l))} \tag{2.38}$$

Here $\tilde{q}(z(l))$ is defined similarly to (2.36). The samples $z(l)\ \forall\ l \in \{1, 2, \ldots L\}$ are now drawn from the easier distribution $q(z)$, and the introduced bias from wrong distribution sampling is corrected by $\hat{w}_l$. The support distribution $q(z)$ is chosen in such a way that it should not be negligible or zero in regions where $p(z)$ may hold significant values. The benefit of importance sampling is illustrated in the Fig. 2.9.

2.6 Kalman Filtering and Smoothing

Consider a general multivariate state-space model as shown below.

$$\boldsymbol{z}_k = A\boldsymbol{z}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(0, Q) \tag{2.39}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(0, R) \tag{2.40}$$

where $A \in \mathbb{R}^{m\times m}$, $C \in \mathbb{R}^{p\times m}$, $Q \in \mathbb{R}^{m\times m}$, and $R \in \mathbb{R}^{p\times p}$ are the state-transition matrix, the emission matrix, the state-noise covariance matrix, and the measurement noise covariance matrix, respectively. Here $\boldsymbol{z}_k \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{x}_k \in \mathbb{R}^{p\times 1}$ represent the state and the measurement, respectively. State estimation refers to the process of inferring the hidden states of a dynamic system based on noisy and incomplete measurements. The Kalman filter [83] and smoother [84–86] have demonstrated remarkable success in addressing these challenges and have become essential tools in various scientific and engineering domains. The motivation behind these techniques stems from their unique ability to fuse incoming measurements with prior knowledge of system dynamics, resulting in optimal state estimation. By recursively updating

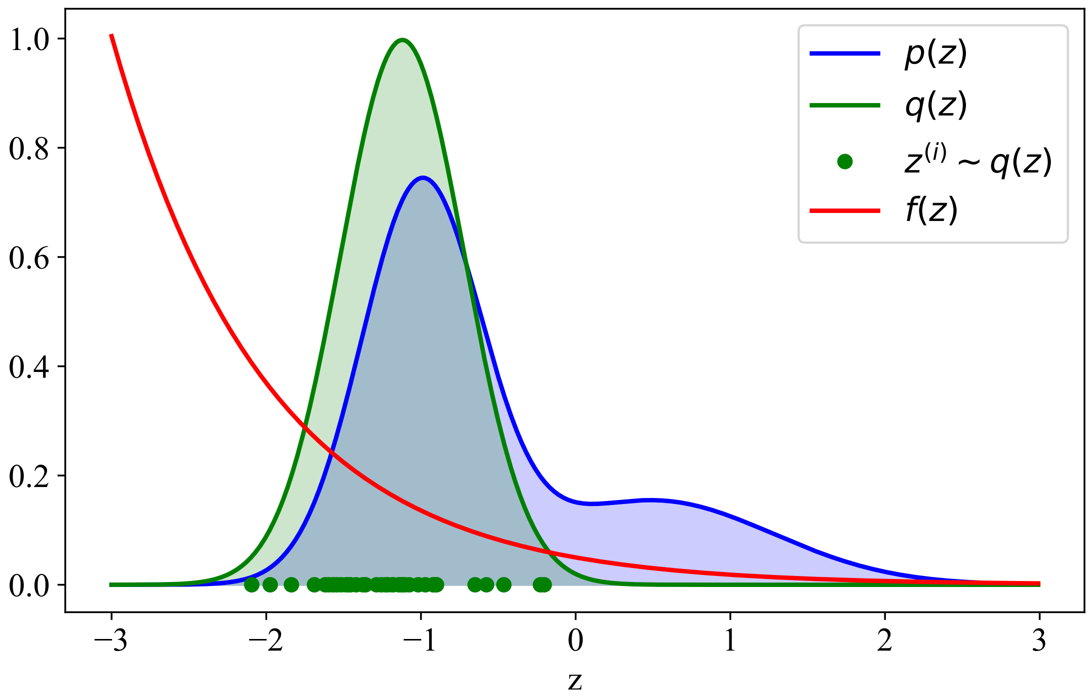


Figure 2.9: **Importance sampling demonstration:** Assuming the computation and sampling of $p(z)$ are challenging, the alternative distribution $q(z)$ serves as a surrogate for estimating the expected value of the function $f(z)$. An effective $q(z)$ should have sample points concentrated in regions where $p(z)f(z)$ exhibits high values.

state estimates, these methods provide an efficient means to handle real-time and online estimation problems.

The process disturbance $\boldsymbol{w}_k \in \mathbb{R}^{m\times 1}$ and the measurement noise $\boldsymbol{v}_k \in \mathbb{R}^{p\times 1}$ are assumed to follow Gaussian distribution. Therefore, the state transition and the emission probabilities are defined as shown in (2.41)-(2.42).

$$p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) = \mathcal{N}(\boldsymbol{z}_k; A\boldsymbol{z}_{k-1}.Q) \tag{2.41}$$

$$p(\boldsymbol{x}_k|\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{x}_k; C\boldsymbol{z}_k, R) \tag{2.42}$$

The initial-state $\boldsymbol{z}_1$ is assumed to follow Gaussian distribution with user-chosen mean $\boldsymbol{\mu}_0$ and covariance Σ_0 i.e.,

$$p(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1; \boldsymbol{\mu}_0, \Sigma_0) \tag{2.43}$$

Lemma 1 [78, 87]: Given a marginal Gaussian distribution for $\boldsymbol{z}$ and a conditional Gaussian distribution for $\boldsymbol{x}$ given $\boldsymbol{z}$ in the form

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \Sigma) \tag{2.44}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; C\boldsymbol{z} + \boldsymbol{b}, R) \tag{2.45}$$

The following equation holds

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; C\boldsymbol{\mu} + \boldsymbol{b}, C\Sigma C^T + R) \tag{2.46}$$

$$p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu} + K(\boldsymbol{x} - (C\boldsymbol{\mu} + \boldsymbol{b})), (\mathrm{I} - KC)\Sigma) \tag{2.47}$$

where

$$K = \Sigma C^T (C\Sigma C^T + R)^{-1} \tag{2.48}$$

2.6.1 Kalman Filtering

The objective is to infer the probability distribution of the state $\boldsymbol{z}_k$ given the values of all observed variables upto $\boldsymbol{x}_k$, i.e., $p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k})\ \forall k \in \begin{bmatrix}1 & N\end{bmatrix}$

1. For $k = 1$, the goal is to find $p(\boldsymbol{z}_1|\boldsymbol{x}_1)$.

$$p(\boldsymbol{z}_1|\boldsymbol{x}_1) \propto p(\boldsymbol{x}_1|\boldsymbol{z}_1)\, p(\boldsymbol{z}_1) \text{ [Bayes' theorem]}$$

$$
\begin{aligned}
&\propto \mathcal{N}(\boldsymbol{x}_1; C\boldsymbol{z}_1, R)\,\mathcal{N}(\boldsymbol{z}_1; \boldsymbol{\mu}_0, \Sigma_0) \\
&\propto \mathcal{N}(\boldsymbol{z}_1; \boldsymbol{\mu}_{\boldsymbol{z}_1|\boldsymbol{x}_1}, \Sigma_{\boldsymbol{z}_1|\boldsymbol{x}_1}) \text{ [Using (2.47)]}
\end{aligned}
$$

where

$$
\begin{aligned}
K_1 &= \Sigma_0 C^T (C\Sigma_0 C^T + R)^{-1} && (2.49) \\
\boldsymbol{\mu}_{\boldsymbol{z}_1|\boldsymbol{x}_1} &= \boldsymbol{\mu}_0 + K_1(\boldsymbol{x}_1 - A\boldsymbol{\mu}_0) && (2.50) \\
\Sigma_{\boldsymbol{z}_1|\boldsymbol{x}_1} &= (\mathrm{I} - K_1 C)\Sigma_0 && (2.51)
\end{aligned}
$$

2. **Prediction:** For $k \neq 1$, the goal for predictive step is to find $p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1})$.

$$
\begin{aligned}
p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}) &= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}}, \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}}) \\
&\propto \int p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{x}_{1:k-1}) p(\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}) d\boldsymbol{z}_{k-1} \text{[Law of total probability]} \\
&\propto \int p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) p(\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}) d\boldsymbol{z}_{k-1} \text{ [Markov assumption]} \\
&\propto \int \mathcal{N}(\boldsymbol{z}_k|A\boldsymbol{z}_{k-1}, Q)\mathcal{N}(\boldsymbol{z}_{k-1}; \boldsymbol{\mu}_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}}, \Sigma_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}})) d\boldsymbol{z}_{k-1} \\
&= \mathcal{N}\left(\boldsymbol{z}_k; A\boldsymbol{\mu}_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}}, A\Sigma_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}} A^T + Q\right) \text{ (Using (2.46))}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} &= A\boldsymbol{\mu}_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}} && (2.52) \\
\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} &= A\Sigma_{\boldsymbol{z}_{k-1}|\boldsymbol{x}_{1:k-1}} A^T + Q && (2.53)
\end{aligned}
$$

3. **Correction:** For $k \neq 1$, the goal for predictive step is to find $p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k})$

$$
\begin{aligned}
p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k}) &\propto p(\boldsymbol{x}_k|\boldsymbol{z}_k, \boldsymbol{x}_{1:k-1})\, p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}) \text{ [Bayes' theorem]} \\
&\propto p(\boldsymbol{x}_k|\boldsymbol{z}_k)\, p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}) \text{ [Markov assumption]} \\
&\propto \mathcal{N}(\boldsymbol{x}_k|C\boldsymbol{z}_k, R)\,\mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}}, \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}}) \\
&= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}, \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}})
\end{aligned}
$$

where

$$
\begin{aligned}
K_k &= \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} C^T (C\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} C^T + R)^{-1} && (2.54) \\
\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} &= \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} + K_k(\boldsymbol{x}_k - C\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}}) && (2.55) \\
\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} &= (\mathrm{I} - K_k C)\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k-1}} && (2.56)
\end{aligned}
$$

2.6.2 Kalman Smoothing

The objective is to infer the probability distribution of the state $\boldsymbol{z}_k$ given the values of all observed variables upto $\boldsymbol{x}_N$, i.e., $p(\boldsymbol{z}_k|\boldsymbol{x}_{1:N})$

1. For $k = N$, the goal is to find $p(\boldsymbol{z}_N|\boldsymbol{x}_{1:N})$, which was previously computed during the final stage of Kalman filtering.

$$p(\boldsymbol{z}_N|\boldsymbol{x}_{1:N}) = \mathcal{N}(\boldsymbol{z}_N; \boldsymbol{\mu}_{\boldsymbol{z}_N|\boldsymbol{x}_{1:N}}, \Sigma_{\boldsymbol{z}_N|\boldsymbol{x}_{1:N}}) \tag{2.57}$$

2. **Smoothing:** For $k \neq N$, the goal for smoothing step is to find $p(\boldsymbol{z}_k|\boldsymbol{x}_{1:N})$.

$$\begin{aligned} p(\boldsymbol{z}_k|\boldsymbol{x}_{1:N}) &= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:N}}, \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:N}}) \\ &\propto \int p(\boldsymbol{z}_k|\boldsymbol{z}_{k+1}, \boldsymbol{x}_{1:N}) p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:N}) d\boldsymbol{z}_{k+1} \\ &\qquad \text{[Law of total probability]} \\ &\propto \int p(\boldsymbol{z}_k|\boldsymbol{z}_{k+1}, \boldsymbol{x}_{1:N}) \mathcal{N}(\boldsymbol{z}_{k+1}; \boldsymbol{\mu}_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:N}}, \Sigma_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:N}}) d\boldsymbol{z}_{k+1} \end{aligned} \tag{2.58}$$

Now the first term is further simplified as follows

$$\begin{aligned} p(\boldsymbol{z}_k|\boldsymbol{z}_{k+1}, \boldsymbol{x}_{1:N}) &\propto p(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k, \boldsymbol{x}_{1:k-1})\, p(\boldsymbol{z}_k|\boldsymbol{x}_{1:N}) \text{ [Bayes' theorem]} \\ &\propto p(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k)\, p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k}) \text{ [Markov assumption]} \\ &\propto \mathcal{N}(\boldsymbol{z}_{k+1}|A\boldsymbol{z}_k, Q)\, \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}, \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}) \\ &= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} + J(\boldsymbol{z}_{k+1} - A\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}), (\mathrm{I} - JA)\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}) \\ &\qquad \text{[Using (2.47)]} \end{aligned} \tag{2.59}$$

where

$$J_k = \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} A^T (A \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} A^T + Q)^{-1} = \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} A^T \Sigma^{-1}_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:k}} \tag{2.60}$$

Substituting (2.59) - (2.60) into (2.58) and using (2.46) yields

$$\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:N}} = \boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} + J_k(\boldsymbol{\mu}_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:N}} - A\boldsymbol{\mu}_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}}) \tag{2.61}$$

$$\Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:N}} = \Sigma_{\boldsymbol{z}_k|\boldsymbol{x}_{1:k}} + J_k(\Sigma_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:N}} - \Sigma_{\boldsymbol{z}_{k+1}|\boldsymbol{x}_{1:k}}) J_k^T \tag{2.62}$$

The Python code presented below [2.1] is provided for research purposes, enabling the computation of the predicted, corrected, and smoothed state.

```
import pandas as pd
import numpy as np
from scipy.linalg import cholesky
from collections import OrderedDict
from numpy.linalg import inv


def kalman_filter_(x, A, Q, C, R, mu_0, Sigma_0, smoother=False):
'''Implements Kalman Filter:
   Inputs:
       1) Noisy measurements: x (p * N)
       2) State-transition matrix: A (N(optional) * m * m)
       3) State-noise covariance matrix: Q (N(optional) * m * m)
       4) Emission matrix: C (N(optional) * p * m)
       5) Measurement-noise covariance matrix: R (N(optional) *p*p)
       6) Initial state mean: mu_0 (m * 1)
       7) Initial state covariance: Sigma_0 (m * m)
       8) Kalman Smoother: Optional
   Outputs: A dictionary with
       1) Predicted state
       2) Corrected state
       3) Smoothed state'''

    p, N = x.shape
    if A.shape[0] != N:
        A = np.repeat(A[None, :, :], N, axis=0)
    if Q.shape[0] != N:
        Q = np.repeat(Q[None, :, :], N, axis=0)
    if C.shape[0] != N:
        C = np.repeat(C[None, :, :], N, axis=0)
    if R.shape[0] != N:
        R = np.repeat(R[None, :, :], N, axis=0)

    m = A.shape[1]

    # Initialization
    K = np.zeros((N, m, p))
    zhat_pred, zhat_crt = np.zeros((m, N)), np.zeros((m, N))
    P_pred, P_crt = np.zeros((N, m, m)), np.zeros((N, m, m))

    #
    K[0]=Sigma_0 @C[0].T @inv(C[0] @Sigma_0 @C[0].T +R[0])
    zhat_crt[:, [0]] = mu_0 + K[0] @ (x[:, [0]] - C[0] @ mu_0)

    P_crt[0] = (np.eye(m) - K[0] @ C[0]) @ Sigma_0
    P_pred[0] = A[0] @ P_crt[0] @ A[0].T + Q[0]

    for t in range(1, N):
        # Prediction
        zhat_pred[:, t] = A[t] @ zhat_crt[:, t-1]
        P_pred[t] = A[t] @ P_crt[t-1] @ A[t].T + Q[t]

        # Correction
        K[t]=P_pred[t]@C[t].T@inv(C[t]@P_pred[t]@C[t].T+R[t])
```

```python
        zhat_crt[:, t]=zhat_pred[:, t]+K[t]@(x[:, t]-C[t]@zhat_pred
                                                           [:, t])
        P_crt[t] = (np.eye(m) - K[t] @ C[t]) @ P_pred[t]

    kf = OrderedDict()
    kf['predict_state'], kf['correct_state'] = zhat_pred,zhat_crt

    if smoother:
        zhat_smooth = np.copy(zhat_crt)
        P_smooth = np.copy(P_crt)
        Jhat = np.ones_like(P_crt)

        for t in range(N - 2, -1, -1):
            Jhat[t] = P_crt[t]@A[t+1].T @ inv(P_pred[t+1])

            zhat_smooth[:, t] =zhat_crt[:, t]+Jhat[t]@(zhat_smooth
                                     [:,t+1] - zhat_pred[:,t+1])
            P_smooth[t] = P_crt[t] + Jhat[t] @ (P_smooth[t+1] -
                                     P_pred[t+1]) @ Jhat[t].T

        kf['smooth_state'] = zhat_smooth

    return kf
```

Listing 2.1: Kalman filter and smoother Python code

Chapter 3

Complex Probabilistic Slow Feature Extraction with Applications in Process Data Analytics *

Today, in modern industrial processes, thousands of correlated process variables are measured and stored. Dimension reduction techniques are often employed to construct informative features by discarding redundant information. Slow feature analysis is one such technique that extracts the slowly varying patterns from measured data. Oscillatory behavior is prevalent in process data due to inadequate control loop tuning and external disturbances such as diurnal temperature variation. Extracting these oscillatory patterns is vital in applications such as control loop monitoring, fault diagnosis. Slow feature analysis may not extract oscillating patterns when the signal to noise ratio is low in process data. This chapter proposes the complex probabilistic formulation that extracts slow oscillatory features. We also present the Expectation-Maximization algorithm to obtain the optimal parameter estimates. Finally, three case studies are presented to illustrate the efficacy of the proposed formulation in soft sensing and fault detection applications.

*This chapter has been published as: **V. K. Puli**, R. Raveendran, and B. Huang, in "Complex probabilistic slow feature extraction with applications in process data analytics, *Computers & Chemical Engineering*, 154, 107456.

3.1 Introduction

Data-driven models have gained enormous momentum due to the availability of a vast amount of historical data obtained with advanced measurement and data storage technologies. Significant correlations exist among the measured variables, and thus the resulting input-output model often suffers from over-fitting. Therefore, dimensionality reduction techniques are often used as a pre-processing step in process modelling to remove redundant information and extract informative variables called features present in the data. The subsequent modelling between the features and outputs is less computationally expensive as the extracted features are comparatively of a lower dimension. Hence, feature extraction or latent variable modelling [88, 89] has attracted considerable attention from various scientific disciplines such as econometrics, statistics, neuro-sciences, and industrial process modelling.

The most popular linear latent variable models include principal component analysis (PCA) [34, 35, 90], partial least squares (PLS) [37–39], independent component analysis [40], slow feature analysis (SFA) [36] and canonical correlation analysis [91], each projecting the higher dimensional data onto a lower-dimensional space based on some optimization criteria. The probabilistic versions [54, 55, 92–94] have been developed to enhance the model interpretation and handle various complexities posed by the real-world dataset. In addition, they work as generative models to obtain new data samples from the given probability distribution and provide more insights into the model characteristics.

Under healthy operating conditions, temporally related common variations govern the process variables due to the dynamic nature of the plant. Those variations are considered more important because of large process inertia, whereas quick changes are attributed to noise or/and fault. SFA can derive such temporally related representations; specifically, it extracts the slowly varying patterns from a set of correlated variables. The probabilistic slow feature analysis (PSFA) [54] assumes a naive Gaussian distribution for the observed variables, and hence, it fails to explain process data with outliers adequately. Therefore, robust models [56, 95] were proposed to deal with data contaminated with outliers. Quality-relevant models [96–98] were proposed to extract latent features from the information carried by both input and output variables. Naturally, with probabilistic extensions, Bayesian versions can be obtained, as shown in [77, 99–101], by integrating prior process knowledge with historical data. A non-stationary model [102, 103] was presented to deal with the non-stationary dynamic data as probabilistic slow feature model can only extract stationary features.

Despite the advantages, the PSFA model has shortcomings that have not been addressed previously in the literature. PSFA may extract slow oscillating features only when the observed variables are error-free or with a high signal-to-noise ratio, which is not the case in practice. Oscillations typically arise in process control loops due to external periodical disturbances, inadequate controller tuning, and control valve stiction. Faults in one process variable can propagate to various plant sections, causing plant-wide oscillations. Plant-wide oscillations result in multiple problems, such as higher energy consumption, low-quality product, and increased waste. The identification of oscillations using data visualization is impractical due to the presence of various frequency segments coupled with measurement noise. Hence, the issue of plant-wide oscillations diagnosis has been studied extensively using the nonlinearity tests [104], spectral envelope methods [10, 105], and process topology-based methods [11, 13, 14, 106–109]. In this chapter, we propose the complex probabilistic slow feature analysis (CPSFA) model to extract oscillating features in the presence of noise. Further, we present a detailed methodology to identify the possible source(s) of plant-wide oscillations. This chapter also demonstrates a simulation study that shows the efficacy of extracted oscillating features in soft-sensor applications [110–113].

The remainder of this chapter is organized as follows. Section 3.2.1 presents an overview of probabilistic SFA and discusses its advantages. In section 3.3.1, a novel methodology to extract slow features with complex roots is introduced. Parameter estimation procedure using the Expectation-Maximization algorithm and two effective initialization strategies are presented in Section 3.3.2. In section 3.4, we illustrate the applications of the proposed modelling algorithm using a simulational and two industrial datasets obtained from the SACAC repository. In section 3.5, we present the concluding remarks.

3.2 Preliminaries

SFA, an unsupervised machine learning approach proposed by [36], extracts the slowly varying lower-dimensional features from data. The linear feature extraction methods available in the literature assume the model given in Eq. (3.1).

$$\mathcal{S} = W^T X; \tag{3.1}$$

where $\mathcal{S} \in \mathbb{R}^{q \times N}$ and $X \in \mathbb{R}^{m \times N}$ are q dimensional hidden features and m dimensional observed variables with N data samples, respectively. $W \in \mathbb{R}^{m \times q}$ is the weight matrix that maps observed variables to the slow features. The feature extraction models mainly differ in the optimization objective function. For example, PCA maximizes the feature variability, whereas SFA minimizes the feature velocity. Section 1.1 details its mathematical formulation, along with the associated optimization problem.

3.2.1 Probabilistic Slow Feature Analysis

In this section, we review the related studies of slow feature analysis using the probabilistic approach [54, 55]. The probabilistic formulation is presented in Eqs. (3.2) - (3.3).

$$\mathbf{s}(k) = A\mathbf{s}(k-1) + \mathbf{w}(k); \quad \mathbf{w}(k) \sim \mathcal{N}(0, Q) \tag{3.2}$$

$$\mathbf{x}(k) = C\mathbf{s}(k) + \mathbf{v}(k); \quad \mathbf{v}(k) \sim \mathcal{N}(0, R) \tag{3.3}$$

Several constraints are employed directly from traditional slow feature analysis, as discussed in Sec. 1.3. Given the observed data $X = \{\mathbf{x}(k) \in \mathbb{R}^m, 1 \leq k \leq N\}$, the complete data joint distribution [78] is shown in Eq. (3.4).

$$p(X_{1:N}, S_{1:N}|A, C, R) = \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{s}_k, C, R) \prod_{k=2}^{N} p(\mathbf{s}_k|\mathbf{s}_{k-1}, A)p(\mathbf{s}_1) \tag{3.4}$$

where

$$p(\mathbf{s}_k|\mathbf{s}_{k-1}, A) = \mathcal{N}(A\mathbf{s}_{k-1}, I_q - AA^T)\, \forall k \in (2, \dots, N) \tag{3.5}$$

$$p(\mathbf{x}_k|\mathbf{s}_k, C, R) = \mathcal{N}(C\mathbf{s}_k, R)\, \forall k \in (1, 2, 3, \dots, N) \tag{3.6}$$

The initial states are considered to follow a standard Gaussian distribution and the conditional distributions of latent and observed variables are shown in the Eq. (3.5) and Eq. (3.6), respectively. The maximum likelihood estimation can be employed to obtain the optimal parameter estimates $\theta \triangleq \{A, C, R\}$, as shown in Eq. (3.7).

$$\theta^* = Arg \max_{\theta} p_\theta(X); \text{ where } p_\theta(X) = \sum_s p_\theta(X, S) \tag{3.7}$$

It is difficult to maximize Eq.(3.7) and obtain an analytical expression for the optimal parameters because of the presence of hidden variables. Expectation-Maximization (EM) algorithm addresses this by iteratively improving parameter estimates using the observed data.

3.3 Complex Probabilistic Slow Feature Analysis

The main limitation of the current probabilistic slow feature analysis formulation lies with the state transition matrix structure. For the slow features to be independent, it is assumed to be diagonal, and hence, the current formulation is restricted to the systems with only real poles. We propose a novel formulation to remove this limitation and encode possible oscillating features naturally, as shown in the Eqs. (3.8) - (3.9).

$$\begin{bmatrix} \mathbf{s}^c(k) \\ \mathbf{s}^r(k) \end{bmatrix} = \begin{bmatrix} A^c & 0 \\ 0 & A^r \end{bmatrix} \begin{bmatrix} \mathbf{s}^c(k-1) \\ \mathbf{s}^r(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{w}^c(k) \\ \mathbf{w}^r(k) \end{bmatrix}; \tag{3.8}$$

$$\mathbf{x}(k) = C\mathbf{s}(k) + \mathbf{v}(k); \tag{3.9}$$

where

$$\mathbf{w}(k) = \begin{bmatrix} \mathbf{w}^c(k) \\ \mathbf{w}^r(k) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} Q^c & 0 \\ 0 & Q^r \end{bmatrix}\right); \quad \mathbf{v}(k) \sim \mathcal{N}(0, R);$$

Unlike the diagonal state-transition matrix in classical formulation, we employ a block diagonal matrix that accommodates complex eigenvalues along with the real ones. Each block may either possess a single real pole or a pair of complex poles. Further, we consider a specific (weak symmetric) structure for each 2 × 2 block in A without loss of generality, as shown in Eq. (3.10).

$$\begin{aligned} A^c &= blkdiag\left\{ \begin{bmatrix} \alpha_1 & \beta_1 \\ -\beta_1 & \alpha_1 \end{bmatrix}, \dots, \begin{bmatrix} \alpha_{q_c} & \beta_{q_c} \\ -\beta_{q_c} & \alpha_{q_c} \end{bmatrix} \right\}; \\ A^r &= diag\{\lambda_1, \dots, \lambda_{q_r}\}; \end{aligned} \tag{3.10}$$

where q_r and q_c refer to the number of real and conjugate pairs of complex poles, respectively. The state-noise covariance matrix is defined as shown in Eqs. (3.11) - (3.12) to conform an identity state covariance matrix.

$$Q^c = blkdiag\left\{Q_1^c, \ldots, Q_{q_c}^c\right\}; \quad Q_j^c \ = \begin{bmatrix} 1-\alpha_j^2-\beta_j^2 & 0 \\ 0 & 1-\alpha_j^2-\beta_j^2 \end{bmatrix}; \tag{3.11}$$

$$Q^r = diag\{1-\lambda_1^2, \ldots, 1-\lambda_{q_r}^2\}; \quad R = diag\{r_1, r_2, \ldots, r_m\}; \tag{3.12}$$

The slow features, at k^{th} instant, produced using the complex pair of eigenvalues and real eigenvalues are denoted using $\mathbf{s}^c(k)$ and $\mathbf{s}^r(k)$, respectively. The corresponding state-noise sequences are indicated by $\mathbf{w}^c(k)$ and $\mathbf{w}^r(k)$, respectively.

$$\mathbf{s}^c(k) = \begin{bmatrix} \begin{pmatrix} s_1^{c_1}(k) \\ s_1^{c_2}(k) \end{pmatrix} \\ \vdots \\ \begin{pmatrix} s_{q_c}^{c_1}(k) \\ s_{q_c}^{c_2}(k) \end{pmatrix} \end{bmatrix}, \quad \mathbf{w}^c(k) = \begin{bmatrix} \begin{pmatrix} w_1^{c_1}(k) \\ w_1^{c_2}(k) \end{pmatrix} \\ \vdots \\ \begin{pmatrix} w_{q_c}^{c_1}(k) \\ w_{q_c}^{c_2}(k) \end{pmatrix} \end{bmatrix};$$
$$\mathbf{s}^r(k) = \begin{bmatrix} s_1^r(k) \\ \vdots \\ s_{q_r}^r(k) \end{bmatrix}, \quad \mathbf{w}^r(k) = \begin{bmatrix} w_1^r(k) \\ \vdots \\ w_{q_r}^r(k) \end{bmatrix};$$

where $s_j^{c_1}(k)$ and $s_j^{c_2}(k)$ represent the two hidden slow oscillating features produced by using the conjugate pair of complex eigenvalues $\alpha_j + i\beta_j$ and $\alpha_j - i\beta_j$, respectively. The two slow-feature sequences $s_j^{c_1}(k)$ and $s_j^{c_2}(k)$ differ in two ways. First, the actual signal varies because of state-noise, and the state-noise variance determines the extent to which they differ. The second difference is the temporal difference, meaning that the two sequences share the same power spectral density but different phase angle. Further, $w_j^{c_1}(k)$ and $w_j^{c_2}(k)$ represent the two state-noise variables drawn from the Gaussian distribution with mean zero and variance $1-\alpha_j^2-\beta_j^2$. The hidden slow feature, denoted by $s_l^r(k)$, is generated with the help of real pole λ_l and the state-noise variable $w_l^r(k)$. The slowness measure, velocity of the obtained complex features, can be derived and calculated using Eq. (3.13).

$$
\begin{aligned}
\nu(s_j^{c_1}) &= \langle (s_j^{c_1}(k) - s_j^{c_1}(k-1))^2 \rangle_k \\
&= \langle ((\alpha_j - 1)s_j^{c_1}(k-1) + \beta_j s_j^{c_2}(k-1) + w_j^{c_1}(k))^2 \rangle_k \\
&= (\alpha_j - 1)^2 \langle s_j^{c_1}(k-1)^2 \rangle_k + \beta_j^2 \langle s_j^{c_2}(k-1)^2 \rangle_k + \langle w_j^{c_1}(k)^2 \rangle_k \qquad (3.13) \\
&= (\alpha_j - 1)^2 + \beta_j^2 + (1 - \alpha_j^2 - \beta_j^2) \\
&= 2(1 - \alpha_j)
\end{aligned}
$$

From the Eq. (3.13), it can be seen that for any slow feature, whether derived from either real or complex pole, the velocity depends on the eigenvalue's real part. A large α_j implies strong correlation between $s_j(k)$ and $s_j(k-1)$ and the Eq. (3.13) confirms that $s_j(k)$ will have slower variations with a lower velocity. Since the proposed slow feature model is general in the sense that it can capture oscillatory dynamics and is equivalent to a state-space model with specific constraints to accommodate the slowness preference, it can be used for control design with an inclination towards controlling slow dynamics. In summary, the unknown parameters to be estimated are the state-transition matrix elements $\{(\alpha_j, \beta_j);\ 1 \leq j \leq q_c\}$, $\{\lambda_l;\ 1 \leq l \leq q_r\}$, emission matrix $C \in \mathbb{R}^{m \times q}$ and the measurement noise variances $\{r_i;\ 1 \leq i \leq m\}$.

3.3.1 Parameter estimation using the EM algorithm

In this section, we derive the EM solution using (2.32)- (2.33) to obtain the optimal parameter values of the proposed model. We obtain the Eq. (3.14) by applying the logarithm operator on both sides of the Eq. (3.4).

$$
\log p(X, S|A, C, R) = \log p(\mathbf{s}_1) + \sum_{k=2}^{N} \log p(\mathbf{s}_k|\mathbf{s}_{k-1}, A) + \sum_{k=1}^{N} \log p(\mathbf{x}_k|\mathbf{s}_k, C, R) \qquad (3.14)
$$

where

$$
\begin{aligned}
\log p(\mathbf{s}_1) &= -\frac{q}{2}\log 2\pi - \frac{1}{2}\mathbf{s}_1^T\mathbf{s}_1; \\
\sum_{k=2}^{N} \log p(\mathbf{s}_k|\mathbf{s}_{k-1}, A) &= -\frac{q(N-1)}{2}\log 2\pi - \frac{(N-1)}{2}\log |I_q - AA^T| \\
&\quad - \sum_{k=2}^{N} \left(\frac{1}{2}(\mathbf{s}_k - A\mathbf{s}_{k-1})^T (I_q - AA^T)^{-1} (\mathbf{s}_k - A\mathbf{s}_{k-1}) \right);
\end{aligned}
$$

$$\sum_{k=1}^{N} \log\, p(\mathbf{x}_k|\mathbf{s}_k, C, R) = -\frac{mN}{2}\log\, 2\pi - \frac{N}{2}\log\, |R| \\ - \sum_{k=1}^{N}\left(\frac{1}{2}(\mathbf{x}_k - C\mathbf{s}_k)^T R^{-1}(\mathbf{x}_k - C\mathbf{s}_k)\right);$$

Here $|\;|$ and log denote the determinant of matrix and the logarithmic operator, respectively. The expression $Q(\theta, \theta^{\eta-1})$ is obtained by taking the conditional expectation with respect to the observed data on both sides of Eq. (3.14). Here $\theta^{\eta-1}$ denote the parameter estimates in the previous $(\eta - 1)$ iteration. Further, $Q(\theta, \theta^{\eta-1})$ is differentiated with respect to θ and then equated to zero to obtain the update expressions for each parameter. Taking the derivative of $Q-$function with respect to λ_l and equating it to zero yields Eq. (3.15).

$$c_{l_3}\lambda_l^3 + c_{l_2}\lambda_l^2 + c_{l_1}\lambda_l + c_{l_0} = 0 \tag{3.15}$$

where

$$\begin{aligned}
c_{l_3} &= -(N-1); \\
c_{l_2} &= \sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_l^r(k)s_l^r(k-1)\right\}; \\
c_{l_1} &= (N-1) - \sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_l^r(k)s_l^r(k)\right\} - \sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_l^r(k-1)s_l^r(k-1)\right\}; \\
c_{l_0} &= \sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_l^r(k)s_l^r(k-1)\right\}
\end{aligned}$$

Taking the derivative of $Q-$function with respect to α_j and equating it to zero yields Eq. (3.16).

$$a_{j_3}\alpha_j^3 + a_{j_2}\alpha_j^2 + a_{j_1}\alpha_j + a_{j_0} = 0 \tag{3.16}$$

where

$$\begin{aligned}
a_{j_3} &= -2(N-1); \\
a_{j_2} &= \sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_1}(k-1) + s_j^{c_2}(k)s_j^{c_2}(k-1)\right\}; \\
a_{j_1} &= 2(N-1)(1-\beta_j^{\eta-1^2}) + 2\beta_j^{\eta-1}\sum_{k=2}^{N} \mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_2}(k-1) - s_j^{c_1}(k-1)s_j^{c_2}(k)\right\}
\end{aligned}$$

$$
\begin{aligned}
&-\sum_{k=2}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_1}(k)+s_j^{c_2}(k)s_j^{c_2}(k)\right.\\
&\qquad\qquad\left.+s_j^{c_1}(k-1)s_j^{c_1}(k-1)+s_j^{c_2}(k-1)s_j^{c_2}(k-1)\right\};\\
a_{j_0} &= a_{j_2}(1-\beta_j^{\eta-1^2})
\end{aligned}
$$

Taking the derivative of $Q-$function with respect to β_j and equating it to zero yields Eq. (3.17).

$$
b_{j_3}\beta_j^3+b_{j_2}\beta_j^2+b_{j_1}\beta_j+b_{j_0}=0 \tag{3.17}
$$

where

$$
\begin{aligned}
b_{j_3} &= -2(N-1);\\
b_{j_2} &= \sum_{k=2}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_2}(k-1)-s_j^{c_1}(k-1)s_j^{c_2}(k)\right\};\\
b_{j_1} &= 2(N-1)(1-\alpha_j^{\eta 2})+2\alpha_j^{\eta}\sum_{k=2}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_1}(k-1)+s_j^{c_2}(k)s_j^{c_2}(k-1)\right\}\\
&\qquad -\sum_{k=2}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{s_j^{c_1}(k)s_j^{c_1}(k)+s_j^{c_2}(k)s_j^{c_2}(k)+s_j^{c_1}(k-1)s_j^{c_1}(k-1)\right.\\
&\qquad\qquad\left.+s_j^{c_2}(k-1)s_j^{c_2}(k-1)\right\};\\
b_{j_0} &= b_{j_2}(1-\alpha_j^{\eta 2})
\end{aligned}
$$

Therefore, the estimates $\{\lambda_l^{\eta},\alpha_j^{\eta},\beta_j^{\eta}\}$ at current iteration η can be calculated as the real root of the cubic Eqs (3.15) - (3.17) such that the magnitude of the eigenvalues lies within the $(0\ \ 1)$ range. The extracted eigenvalues always satisfy this condition due to the constraint presented by Eq. (1.12), given a stationary observed data. Similarly, by setting the partial derivatives of $Q-$function to zero with respect to C and $\{r_i;\ 1\leq i\leq m\}$, the update expressions can be obtained using Eqs. (3.18) - (3.19) [78].

$$
C^{\eta}=\left[\sum_{k=1}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{x}_k\mathbf{s}_k^T\right\}\right]\left[\sum_{k=1}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k\mathbf{s}_k^T\right\}\right]^{-1} \tag{3.18}
$$

$$
r_i^{\eta}=\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{x_i^2(k)\right\}-\frac{2}{N}C^{\eta}(i,:)\sum_{k=1}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k x_i(k)\right\}
$$

$$+\frac{1}{N}C^{\eta}(i,:)\sum_{k=1}^{N}\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k\mathbf{s}_k^T\right\}C^{\eta}(i,:)^T \tag{3.19}$$

The update equations derived earlier are functions of parameters of the posterior distribution $p(S|X,\theta^{\eta-1})$. The Kalman filter and Kalman smoother algorithms are employed to infer the expectations of $p(S|X,\theta^{\eta-1})$ and the same is summarized in Algorithm 3.1 and 3.2, respectively. The expectations of the posterior distribution $p(S|X,\theta^{\eta-1})$ can be evaluated using Eq. (3.20) - (3.22) [78]. Eq. (3.23) can be used for monitoring the convergence of EM algorithm.

$$\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k\right\} = \hat{\mu}_k \tag{3.20}$$

$$\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k\mathbf{s}_{k-1}^T\right\} = \hat{V}_kJ_{k-1}^T + \hat{\mu}_k\hat{\mu}_{k-1}^T \tag{3.21}$$

$$\mathbb{E}_{X,\theta^{\eta-1}}\left\{\mathbf{s}_k\mathbf{s}_k^T\right\} = \hat{V}_k + \hat{\mu}_k\hat{\mu}_k^T \tag{3.22}$$

$$\log p(X|\theta^{\eta}) = \sum_{k=1}^{N}\log\mathcal{N}(CA\mu_{k-1}, CP_{k-1}C^T + R) \tag{3.23}$$

Algorithm 3.1 Kalman Filter

```
Input: $X$ and $\{A, C, R\}$ at iteration $(\eta - 1)$
$\mu_1 = K_1\mathbf{x}_1$   $V_1 = I_q - K_1C$   $K_1 = C^T(CC^T + R)^{-1}$   for $k = 2, 3, \ldots, N$ do
    $P_{k-1} = A(V_{k-1} - I_q)A^T + I_q$   $\mu_k = A\mu_{k-1} + K_k(\mathbf{x}_k - CA\mu_{k-1})$   $V_k = (I_q - K_kC)P_{k-1}$
     $K_k = P_{k-1}C^T(CP_{k-1}C^T + R)^{-1}$
end
Output: $\mu_k, V_k, P_k \forall k \in \{1, 2, \ldots, N\}$
```

Algorithm 3.2 Kalman Smoother

```
Input: $\mu_k, V_k, P_k \forall k \in \{1, 2, \ldots, N\}$, and $A$ at iteration $(\eta - 1)$
$\hat{\mu}_N = \mu_N$   $\hat{V}_N = V_N$   for $k = N - 1, N - 2, \ldots, 1$ do

    $J_k = V_kA^T(P_k)^{-1}$   $\hat{\mu}_k = \mu_k + J_k(\hat{\mu}_{k+1} - A\mu_n)$   $\hat{V}_k = V_k + J_k(\hat{V}_{k+1} - P_k)J_k^T$
end
Output: $\hat{\mu}_k, \hat{V}_k \forall k \in \{1, 2, \ldots, N\}$
```

3.3.2 Initialization strategy

The EM algorithm is susceptible to locally optimal solutions without appropriate initial parameter guesses. Further, it takes several iterations to reach the desired tolerance of the parameters with each random initialization, and each iteration of the algorithm is computationally expensive. Therefore, two different strategies are devised to construct efficient initial guesses as discussed below.

3.3.2.1 Using Linear slow feature analysis

The first initial parameter guess construction is based on the deterministic solution. [54] have shown that the deterministic solution is equivalent to the maximum-likelihood estimation solution in the limiting case. Hence, the deterministic solution has been used to construct initial guesses for the EM algorithm in [55]. We establish a similar relationship between the deterministic version and the proposed formulation. Given m input variables, the deterministic formulation, shown in Eqs. (1.1) - (1.5), can be solved to obtain a maximum of m slow features that satisfy Eq. (3.24).

$$X = W^{-T} S \tag{3.24}$$

where each row of S is a slow feature with increasing velocity. Eq. (3.24) can be decomposed for any q ($\leq m$), as shown in Eq. (3.25).

$$X = W_1^{-T} S_{1:q} + W_2^{-T} S_{q+1:m} \tag{3.25}$$

where W_1^{-T} and W_2^{-T} denote the first q columns and the last $(m-q)$ columns of W^{-T}, respectively, $S_{1:q}$ and $S_{q+1:m}$ denote the first q rows and the last $(m-q)$ rows of S, respectively. Since Eq. (3.3) and Eq. (3.25) are equivalent in the limiting case, the initial guesses for the emission matrix and measurement noise covariance matrix can be established as follows,

$$C_0 = W_1^{-T};\ R_0 = diag(W_2^{-T} W_2^{-1})$$

The averaged velocity of each of the slow features in $S_{1:q}$ can be computed using the Eqn. (3.26).

$$v(S_i) = \frac{1}{N-1} \sum_{k=2}^{N} (S_i(k) - S_i(k-1))^2 \tag{3.26}$$

Hence, the initial guess vector that represents the diagonal entries (or the real part of eigenvalues) of the state transition matrix can be estimated using Eq. (3.13) as follows,

$$\boldsymbol{\alpha}_0 = 1 - \frac{v(S_{1:q})}{2}$$

From Eq. (3.13), we can also infer that the slow feature's velocity depends only on the eigenvalue's real part. Hence, the velocity of the two slow features produced by a conjugate pair of eigenvalues is equivalent as they share the same real part. Conversely, we assume if the velocities of two slow features are equal, then their corresponding eigenvalues may be a complex conjugate pair. Given two deterministic slow features s_j and s_{j+1} with corresponding real part guesses α_j and α_{j+1} such that $(\alpha_j - \alpha_{j+1}) \leq \epsilon$, the initial guess for the imaginary part β_j can be estimated using Eq. (3.17) with deterministic slow features replacing conditional expectations in the coefficient expressions i.e.,

$$\begin{aligned}
b_{j_3} &= -2(N-1) \\
b_{j_2} &= \sum_{k=2}^{N} s_j(k)s_{j+1}(k-1) - s_j(k-1)s_{j+1}(k) \\
b_{j_1} &= 2(N-1)(1-\alpha_j^2) - \sum_{k=2}^{N} (s_j(k)s_j(k) + s_{j+1}(k)s_{j+1}(k) - s_j(k-1)s_j(k-1) \\
&\quad - s_{j+1}(k-1)s_{j+1}(k-1)) + 2\alpha_j \sum_{k=2}^{N} s_j(k)s_j(k-1) + 2\alpha_j \sum_{k=2}^{N} s_{j+1}(k)s_{j+1}(k-1) \\
b_{j_0} &= b_{j_2}(1-\alpha_j^2)
\end{aligned}$$

3.3.2.2 Using Subspace Identification

State-space identification methods like MOESP (Multivariable Output-Error State-Space), N4SID (N4 System Identification), and CVA (Canonical Variate Analysis) are classical techniques used in system identification to estimate linear dynamic models from input-output data. They aim to estimate the state-space representation of the system (as shown in Eqs. (3.27) - (3.28)), which includes state variables $\boldsymbol{z}_k$, input-output relationships, and noise characteristics.

$$\mathbf{z}(k) = F\mathbf{z}(k-1) + \mathbf{d}(k); \quad \mathbf{d}(k) \sim \mathcal{N}(0, P) \tag{3.27}$$

$$\mathbf{x}(k) = H\mathbf{z}(k) + \bar{\mathbf{v}}(k); \quad \bar{\mathbf{v}}(k) \sim \mathcal{N}(0, \bar{R}) \tag{3.28}$$

The key differences between subspace identification and the identification of the PSFA model using iterative algorithms like expectation-maximization or variational Bayesian inference are outlined below:

- The estimated state-transition matrix F and process noise covariance matrix P are non-diagonal. Additionally, F and P are treated as independent parameters, causing the state covariance matrix not to be equal to the identity matrix, thereby violating the basic requirements of slow features.

- While subspace identification methods possess an analytical solution, the handling of complexities in industrial process data, such as outliers and skewed noise, may not be facilitated.

- Additionally, the incorporation of process knowledge pertaining to the involved parameters into subspace identification methods may not be easily accomplished.

Nevertheless, intelligent initial guess parameters can be constructed, potentially leading to early convergence and/or higher likelihood. The procedure is summarized below.

- CVA implemented to obtain the unconstrained model, as shown in Eqs. (3.27) - (3.28).

- Since the state-transition matrix is block-diagonal in the proposed model, we use similarity transformation matrix G to achieve the same. The G matrix is constructed with the help of eigenvectors of F matrix.

  $$G = \left[\begin{array}{cccccc} \gamma_1 \mathbf{g}_1^{re} & ... & \gamma_{q_c} \mathbf{g}_{q_c}^{im} & \delta_1 \mathbf{g}_1 & ... & \delta_l \mathbf{g}_{q_r} \end{array} \right]$$

  where $\mathbf{g}_j^{re}$ and $\mathbf{g}_j^{im}$ are the real and imaginary parts of the eigenvector corresponding to j^{th} complex conjugate eigenvalue pair of F and $\mathbf{g}_l$ is the eigenvector of l^{th} real eigenvalue of F.

- Since the latent states follow unit variance constraint in the proposed formulation, the parameters γ_j, δ_l are optimized such that:

$$diag(A_0 A_0^T + Q_0) = \vec{1}$$

- The mathematical procedure is summarized below.

$$\begin{aligned}
G\hat{\mathbf{z}}(k) &= FG\hat{\mathbf{z}}(k-1) + \mathbf{d}(k); \\
\mathbf{x}(k) &= HG\hat{\mathbf{z}}(k) + \bar{\mathbf{v}}(k); \\
\implies \hat{z}(k) &= \underbrace{G^{-1}FG}_{A_0}\hat{z}(k-1) + \underbrace{G^{-1}d(k)}_{\bar{w}(k)}; \quad \bar{\mathbf{w}}(k) \sim N(0, Q_0); \\
\mathbf{z}(k) &= \underbrace{HG}_{C_0}\hat{\mathbf{z}}(k) + \bar{\mathbf{v}}(k); \quad \bar{\mathbf{v}}(k) \sim N(0, R_0);
\end{aligned}$$

where $\mathbf{z}(k) = G\hat{\mathbf{z}}(k)$ and $Q_0 = G^{-1}PG^{-T}$

- Now with $\{A_0, C_0, R_0\}$ as initial guesses for $\{A, C, R\}$, the iteration is performed until convergence.

3.4 Simulation and Applications

In this section, three case studies are presented to showcase the performance of complex probabilistic slow feature analysis formulation given noisy data.

3.4.1 Simulation

In this subsection, we present a simulation study to illustrate the efficacy of the proposed formulation in soft sensing applications. A total of six hidden oscillating sequences $\mathbf{h}^*(k) \in \mathbb{R}^{6\times 1}$ were generated using a general state-space model with the state-transition and the noise covariance matrices shown below,

$$\begin{aligned}
A^* &= \begin{bmatrix} 0.8 & 0.16 & 0.14 & 0.2 & 0.6 & 0.3 \\ -0.6 & 0.5 & 0.3 & 0.8 & 0.7 & -0.9 \\ -0.13 & 0.13 & 0.6 & 0.7 & 0.3 & -0.4 \\ -0.15 & 0.5 & -0.7 & 0.46 & 0.1 & 0.5 \\ -0.3 & -0.5 & 0.4 & -0.6 & 0.18 & 0.59 \\ 0.13 & -0.4 & 0.8 & -0.3 & -0.59 & 0.18 \end{bmatrix}; \\
Q^* &= D^* \times {D^*}^T
\end{aligned}$$

where $D^*_{6\times 6}$ is a randomly drawn matrix from the standard uniform distribution. The observation dataset is generated with the help of an emission matrix $C^*_{12\times 6}$ drawn from the standard uniform distribution and Gaussian measurement noise, such that the the ratio of true response variance to the noise variance is 0.1. Low signal-to-noise ratio often results in poor prediction performance, thereby restricting the models' end use. A regression vector $\mathbf{m}^* = \begin{bmatrix}0.56 & 0.19 & 0.69 & 0.34 & 0.42 & 0.90\end{bmatrix}^T$ was used to generate a sequence of quality variable $\mathbf{y}^*$ from the latent features using Eq. (3.29).

$$\mathbf{y}^*(k) = \mathbf{h}^*(k)^T\mathbf{m}^* + \mathbf{e}^*(k) \tag{3.29}$$

Therefore, the generated dataset consisted of 12 variables and one quality variable with 5000 data samples. The observed variables with their corresponding velocities (ν) and Pearson correlation coefficients (ρ) against the quality variable are shown in the Fig. 3.1. It can be inferred that the observed variables have low correlations with the quality variable since the measured variables are fast varying, whereas the quality variable is relatively slow.

Figure 3.2 shows the extracted slow features using deterministic formulation along with their corresponding power spectral densities (PSD). We infer that the overall signal power is shared by multiple frequency components from the PSD plots. Hence, the deterministic features lack the oscillatory nature required to explain the quality variable. The extracted complex slow features along with their corresponding PSDs and estimated eigenvalues (λ) are shown in the Fig. 3.3. The state transition matrix order was assumed to be the number of measured variables. The proposed formulation extracts some useful features that are highly correlated with the quality variable without resort to output knowledge. Besides, oscillatory behaviour is perceived effectively since fewer frequency components share the overall power in each of the first six slow features. A linear regression model was built between the quality variable and four highly correlated slow features as shown in Eq. (3.30). The order (d) was chosen to be four since the additional slow feature did not result in lower root mean square error (RMSE) on the test data.

$$\mathbf{y}^*(k) = \mathbf{s}_{1:d}(k)^T\mathbf{b} + \mathbf{e}(k) \tag{3.30}$$

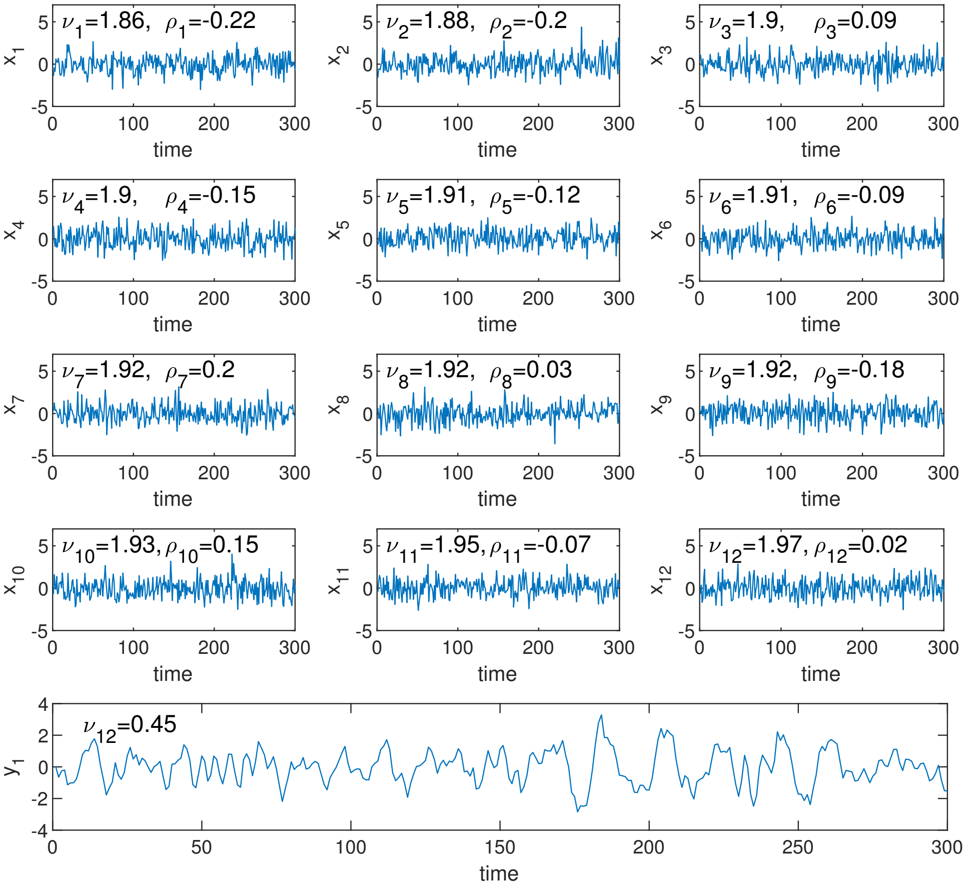


Figure 3.1: Normalized dataset with 12 variables (Only 300 data points are shown for better visualization)

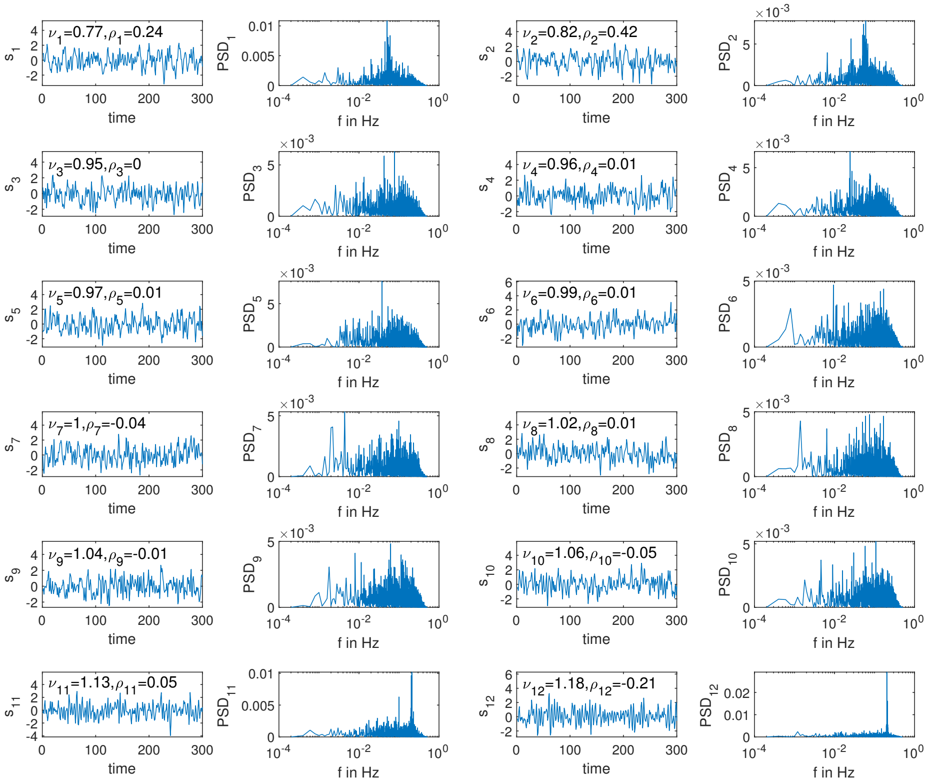


Figure 3.2: Extracted deterministic slow features

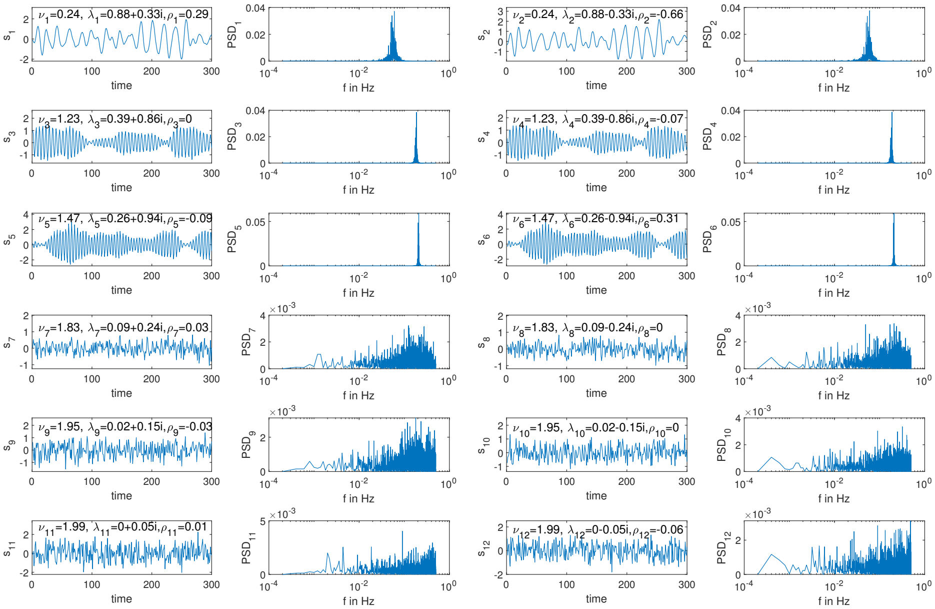


Figure 3.3: Extracted slow oscillating features

The performance of the proposed formulation was compared with the other linear latent variable models, using lagged observations, that are widely used in the literature. We choose dynamic ordinary least squares (OLS), PSFA, dynamic PCA [114], dynamic SFA, and dynamic PLS [115]. The stacking order of lagged measurements is assumed to be one since the hidden variables in CPSFA evolves according to the first-order autoregressive model. The latent features were chosen based on the correlation criteria in each of the above models, and the number of latent features was fixed to be four for comparison purposes. The performance parameters RMSE and ρ between the measured and the predicted quality variable are shown in Fig. 3.4. The significant difference in the two indices for CPSFA compared to other latent variable methods demonstrates its improved predictive abilities.

3.4.2 Industrial Case Study-1

In this section, we use the hydrogen reformer unitwide oscillatory data obtained from the South-East Asian refinery for the case study. A flow diagram of the process is shown in Fig. 3.5. The light hydrocarbons react with high-temperature (700^0C −

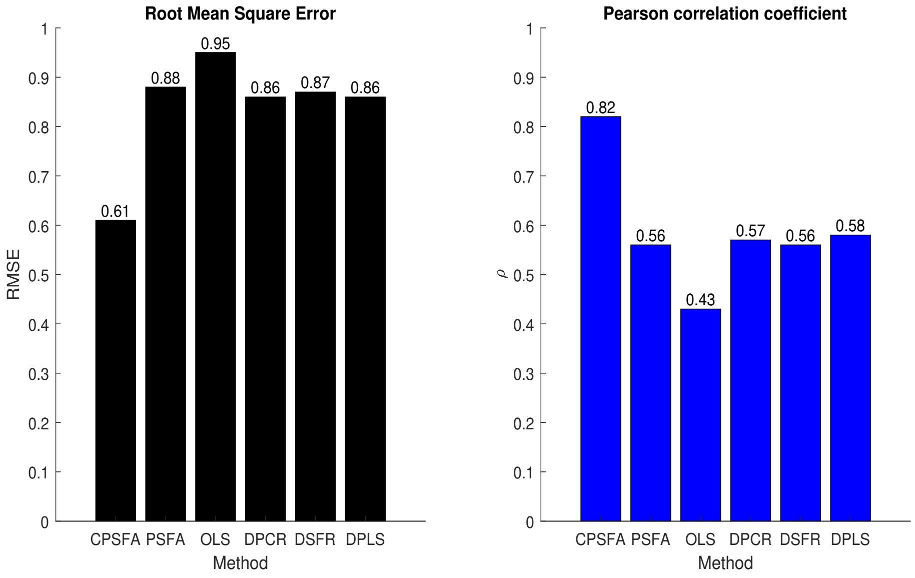


Figure 3.4: Performance comparison (four highly correlated features are chosen in each method)

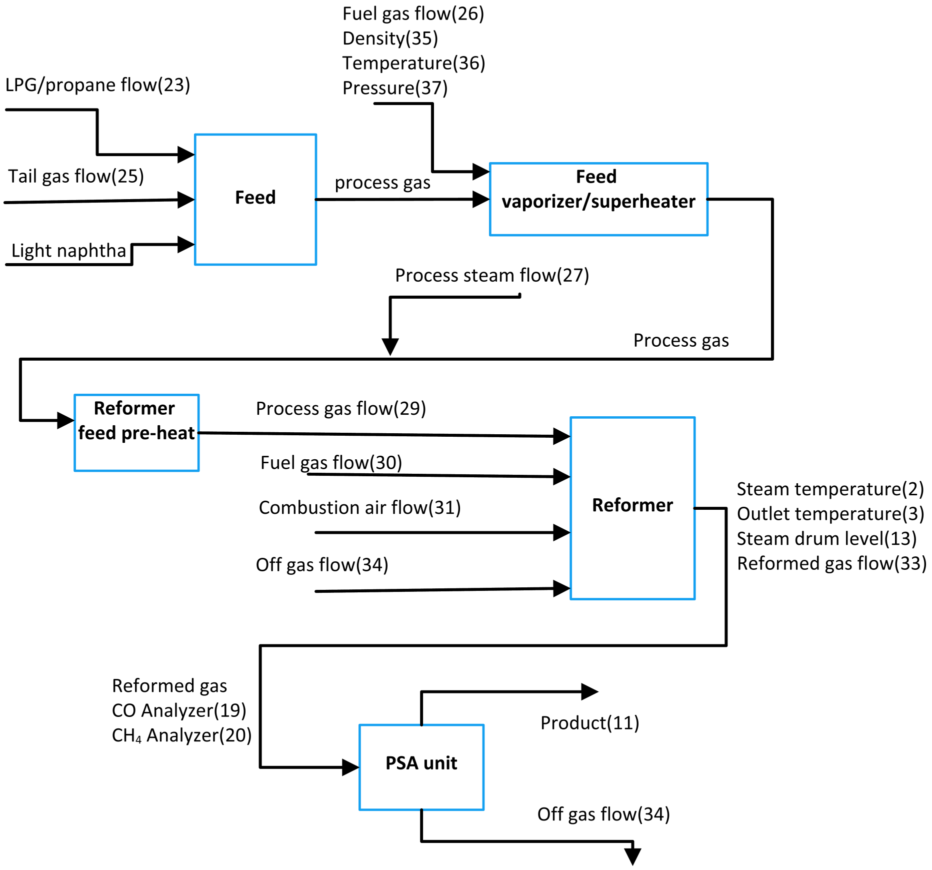


Figure 3.5: SE Asia refinery plant summarized block diagram [1]

1000^0C) steam in the presence of a nickel catalyst under $(3-25)$ bar pressure in the reformer to produce hydrogen and carbon oxides. Finally, the hydrogen is separated from the mixture stream as product flow $\{11\}$ using the Pressure Swing Adsorption (PSA) unit. The residual H_2 is fed back to the reformer as the off-gas flow $\{34\}$ to maximize H_2 recovery.

The industrial dataset[†] [116,117] includes the data of 25 controller loops. In total, 67 variables that include twenty-five controller output variables (OP), twenty-five process variables (PV), twelve indicator variables (IV), and 5 set-point (SP) variables were used for the following study. Remaining set-point variables were neglected as they were constant during the chosen time range. Further, the dataset is divided into a training set with 1000 data samples and cross-validation set with 441 data samples to compute the optimal model order.

The proposed formulation was applied to the training dataset with a pre-chosen order varying from one to ten. The optimal model order was chosen based on the log-likelihood value of the cross-validation data. Fig. 3.6 shows the cross-validation data likelihood for various model orders. For the current case study, the optimal order was chosen to be six as there was no significant improvement in the log-likelihood value with an additional feature. Finally, the extracted slow features with their corresponding PSDs are shown in Fig. 3.7.

The hidden features are interpreted as the causal variables that drive the observed variables. Since there were oscillations in the observed data, and the CPSFA captures the oscillating behaviour, the extracted feature(s) may contain the oscillatory root cause(s). The oscillatory source(s) among the extracted slow features is chosen based on the frequency band. The frequency band is defined as the set of frequency components that contribute to the overall signal power. The oscillatory slow feature with faster frequency components may be considered as a more critical oscillatory source. Low-velocity oscillating features are relatively less-problematic since they comprise low-frequency components. The larger imaginary part of the eigenvalue conforms to the oscillatory behaviour, and a smaller real part corresponds to faster frequency components. Therefore, a metric called oscillation index (ω) is defined as the logarithm of ratio of absolute values of eigenvalue's imaginary to the real part, as shown

[†] Available online at `https://sacac.org.za/resources/`

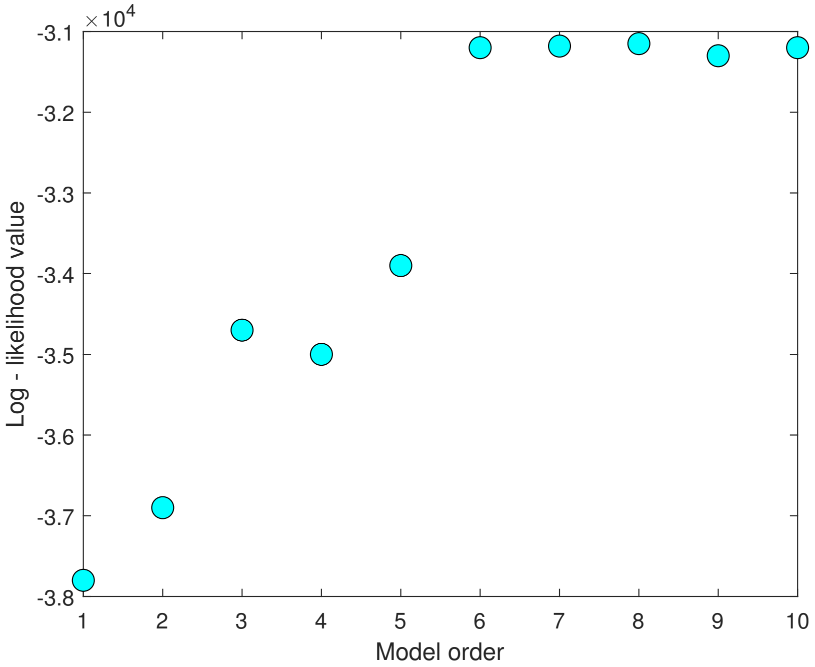


Figure 3.6: Cross validation data log-likelihood value vs. model order

in Eq. (3.31). The oscillation index is computed for each slow feature, and the slow feature with the highest ω is considered as the primary hidden source.

$$\omega_j = \ \log \left|\frac{\beta_j}{\alpha_j}\right| \ \forall\, j \in \{1, 2, ..., q\} \tag{3.31}$$

The fifth slow feature in Fig. 3.7, which offers a higher ω, was chosen to be the primary hidden source of the plant-wide oscillations. Further, a frequency match ($\digamma_{x_i,s}$) is defined in Eq. (3.32) as the correlation between the magnitude of the variables in the frequency domain. It is a measure of the amount of common frequency content between two variables. The normalized frequency match, as shown in the Eq. (3.33), was computed between the chosen hidden oscillatory source and each measured variable.

$$\begin{aligned} \digamma_{x_i,s} &= \int_{-0.5}^{0.5} |\mathbf{x}_i(f)||\mathbf{s}(f)|df \ \forall\, i \in \{1, 2, ..., m\} \\ &\approx \sum_{n=1}^{N} |\mathbf{x}_i(f_n)||\mathbf{s}(f_n)|; \quad f_n = -0.5 + \frac{n}{N}; \end{aligned} \tag{3.32}$$

$$|\digamma_{x_i,s}| = \frac{\digamma_{x_i,s}}{\sqrt{\digamma_{x_i,x_i}\digamma_{s,s}}} \tag{3.33}$$

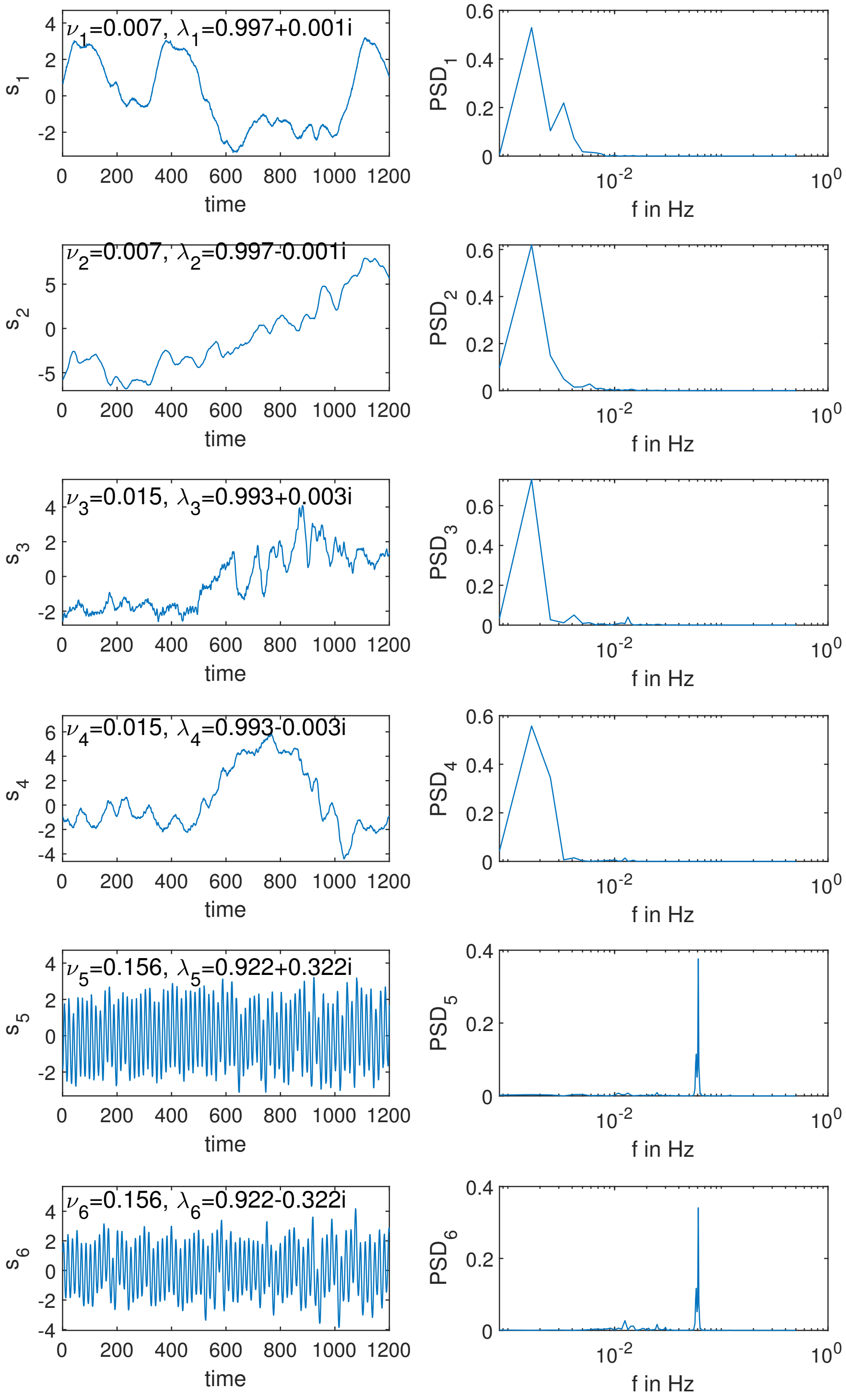


Figure 3.7: Extracted oscillatory slow features

where $\mathbf{x}_i(f)$ and $\mathbf{s}(f)$ are the i^{th} observed variable and the chosen hidden source, respectively. The variable with the highest frequency match is the possible root cause of oscillations among the observed variables. This method essentially provides a priority order set of a possible sources of oscillations. The summary of the proposed methodology to detect the root cause of the plant-wide oscillations using CPSFA is shown in Table 3.1. The algorithm is primarily designed for offline data analysis. If online application is desired, a moving window approach may be applied and the steps shown in Table 3.1 can be applied to the data within the window.

Table 3.1: Plant-wide oscillations source detection algorithm

1. Import the plant-wide oscillations dataset.
2. Form a time-series matrix X with PV, SP, OP, and IV variables.
3. Perform data standardization.
4. Extract oscillating features using CPSFA.
5. Compute oscillation index $\omega_j \; \forall \, j \in \{1, 2, ..., q\}$. The oscillatory feature with highest ω is chosen to be the primary hidden source.
6. Calculate normalized frequency match $|\digamma_{x_i,s}| \; \forall \; i \in \{1, 2, ..., m\}$ between the primary hidden source and the observed variables.
7. Obtain the priority order checklist of observed variables for possible root cause of plant-wide oscillations.

The normalized frequency match of various observed variables is shown in Fig. 3.8. We observed that seven variables have a high normalized frequency match with the selected oscillating slow feature. Further, five $\{2, 3, 10, 13, 24\}$ of them were process variables, and two $\{20, 34\}$ were indicator variables. Tag 20 indicates the methane composition in the reformer output, and Tag 34 represents the off-gas flow into the reformer. It was observed that the normalized frequency match of the five process variables was higher than their corresponding controller output variables. It means that the corresponding controllers were successful to an extent in mitigating the oscillations, and hence, the possibility of control valve stiction was ruled out. Therefore, the source of oscillations in the five process variables were either the two indicator vari-

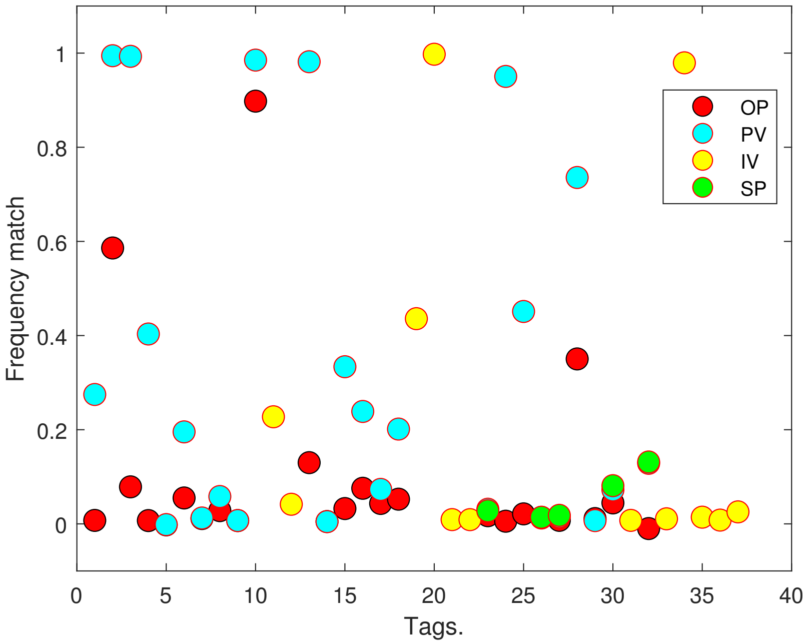


Figure 3.8: Frequency match of 67 input variables with the chosen oscillating feature

ables or faulty hardware sensors that measured the corresponding process variables. Hence, the priority order checklist for the current industrial case study is as follows. The two indicator variables must be inspected first, followed by the corresponding hardware sensors of the five process variables. The analysis in [118] conforms that the off-gas flow was indeed the origin of plant-wide disturbances.

3.4.3 Industrial Case Study-2

In this section, we adopt the Australian refinery process data[‡] provided by [119] to illustrate the ability of the proposed method to detect the source of plant-wide oscillations given noisy data where the oscillations are not apparent, unlike the first industrial case study. Figure 3.9 shows the distillation process under consideration, a separation unit that contains five control loops, namely temperature (TC1), steam flow (FC1), an analyzer (AC1), upstream (PC1) and downstream pressure (PC2) control loops.

The setpoints are constant during the chosen time range; hence, we used only the process variable and controller output data of all control loops for the current analysis.

[‡] Available online at `https://sacac.org.za/resources/`

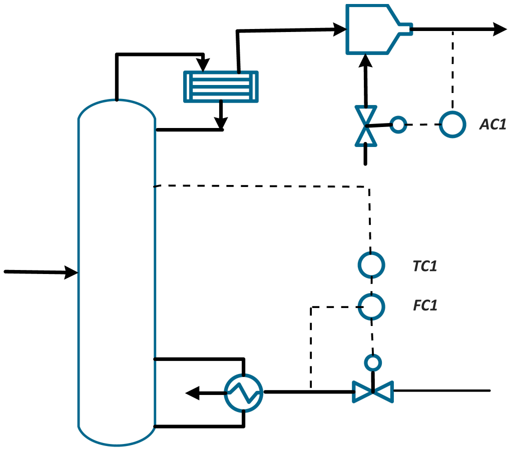


Figure 3.9: Australian refinery separation unit [1]

A measurement noise with a signal-to-noise ratio of 0.2 was added to showcase the ability of the proposed algorithm in the presence of noise. The observed data with and without measurement noise is shown in the right and left column of Fig. 3.10. Further, the dataset is divided into a training set with 700 data samples and cross-validation set with 200 data samples to compute the optimal model order.

The same idea discussed earlier was applied to the two datasets (noise-less and noisy) to obtain the optimal model order. The cross-validation noise-less and noisy data log-likelihood values reached the maximum for the CPSFA model with ten and six slow features, respectively. The normalized frequency match of various variables against the primary hidden source given noiseless and noisy data is shown in the left and right column of Fig. 3.11, respectively. We observed that the FC1 related variables have the highest normalized frequency match in both the cases, and hence, the flow controller loop was most likely to be the primary source of plant-wide oscillations. The slow oscillating feature with the next highest ω was assumed to be the second hidden source of the disturbance as there were more oscillating features, unlike the first case study. The frequency match corresponding to the second oscillating hidden source given noiseless and noisy data is shown in the left and right column of Fig. 3.12, respectively. From Fig. 3.12, it is concluded that a secondary disturbance was observed in PC1 and PC2 based on the metrics obtained using noise-less data (as shown in the left subfigure). However, the same observation was not identified using the noisy data (see the right subfigure). It was identified in [1] that the steam

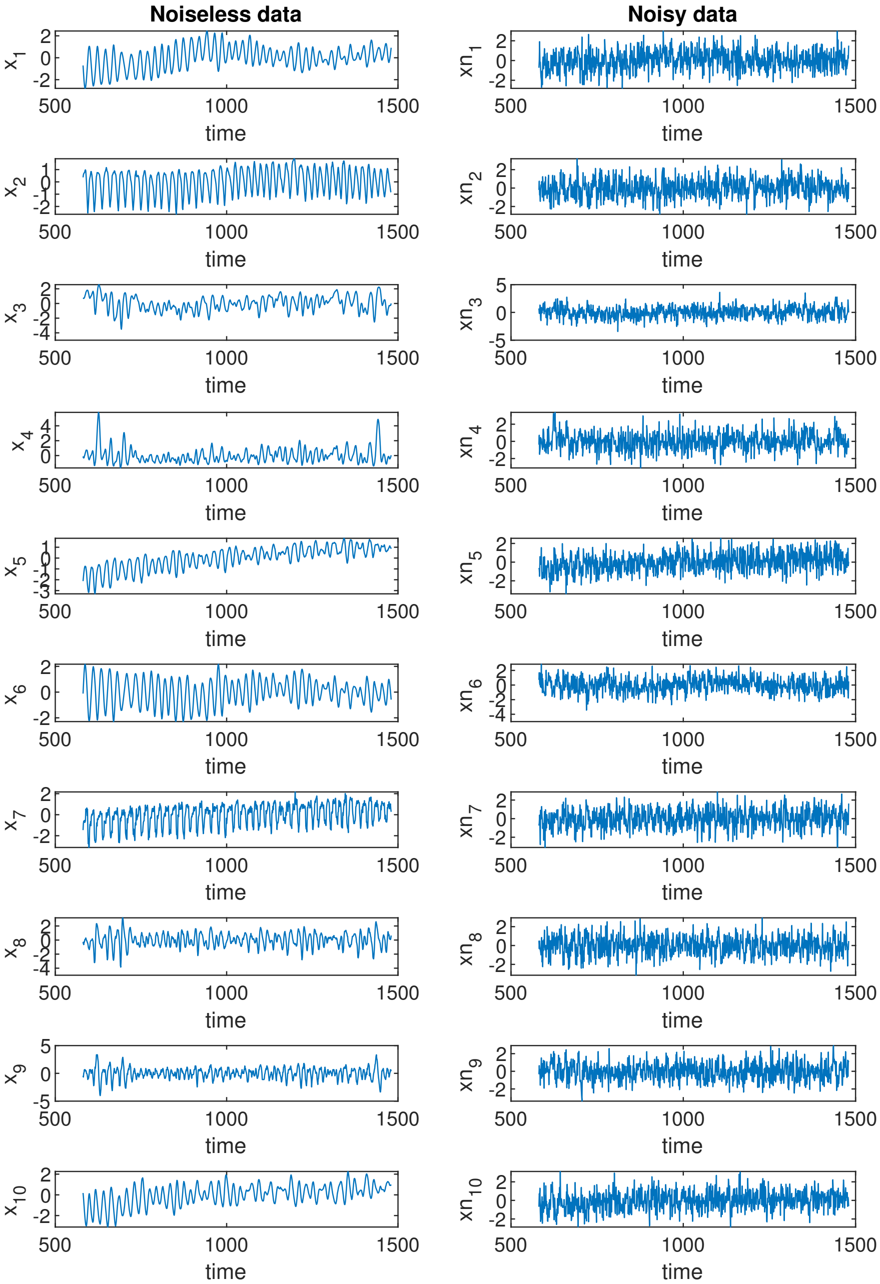


Figure 3.10: Ten observed variables

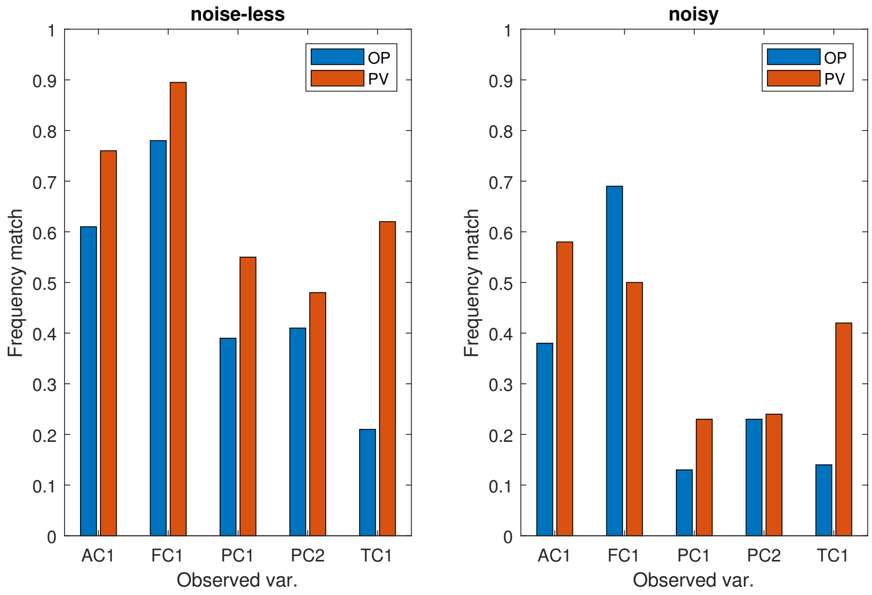


Figure 3.11: Frequency match of ten observed variables with the chosen primary oscillating feature given noiseless and noisy data

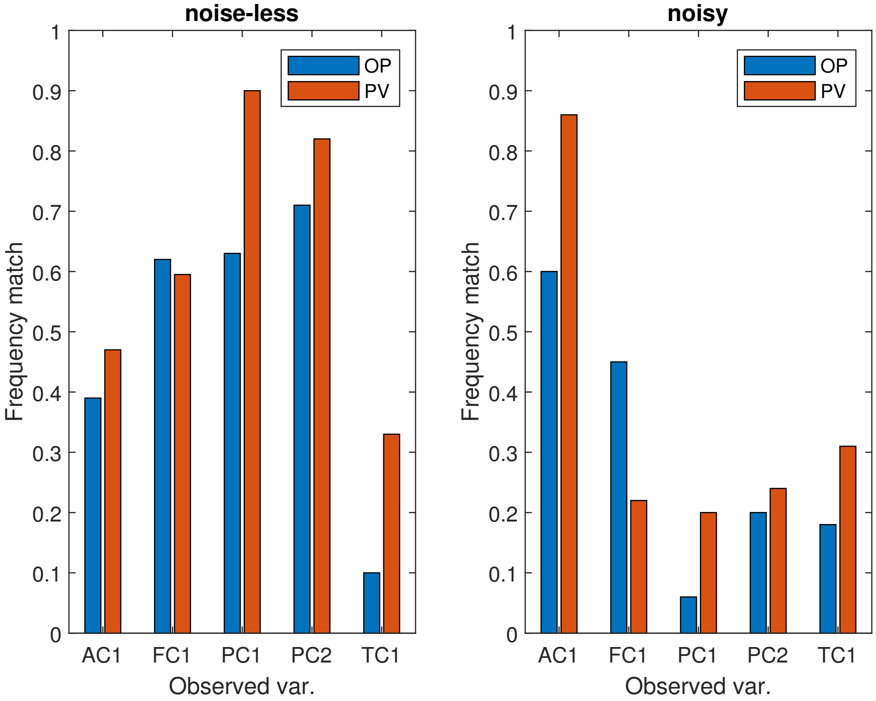


Figure 3.12: Frequency match of ten observed variables with the chosen secondary oscillating feature given noiseless and noisy data

flow loop contains a faulty orifice flow meter, which was the root cause of the primary disturbance. Thus the proposed analysis can be performed effectively to detect and diagnose the primary source of plant-wide oscillations.

3.5 Conclusion

This chapter discusses the primary shortcoming of the classical probabilistic slow feature analysis. The diagonality premise of the state-transition matrix is postulated to obtain uncorrelated hidden features, and hence, in principle, it cannot be used to extract slow oscillating features. We propose a novel data-driven model that can extract slow features with oscillating patterns. The state-transition matrix is augmented to include a pair of complex eigenvalues that provide the oscillating nature, and hence, it can no longer be a diagonal matrix. The strength of the proposed algorithm was shown using three case studies. The first one was the simulational case study that depicts the efficiency of proposed formulation over other latent variable models in soft sensing applications. Two industrial case studies were shown, one with high-dimensional dataset and the other with increased noise dataset, which uses the complex probabilistic slow feature analysis to detect and diagnose the oscillating fault. The potential problems for further study include the possible extensions to handle traditional industrial dataset complexities such as outliers, missing data, non-linear characteristics.

Chapter 4

Variational Bayesian Approach to Nonstationary and Oscillatory Slow Feature Analysis With Applications in Soft Sensing and Process Monitoring [*]

Extraction of underlying patterns from measured variables is central to various data-driven control applications, such as soft-sensor modelling, statistical process monitoring, fault detection and diagnosis. More often than not, the observed variables display non-stationary characteristics and oscillations due to the changes in operating conditions and problems in controller tuning. Such variations pose a great challenge to conventional feature extraction methods. Hence, we present a probabilistic drift-type non-stationary oscillating slow feature model that can separate oscillating patterns and non-stationary variations from measured data. Further, the measurement noise of each variable is independently modelled to account for the fact that not all the observed variables have the same level of uncertainty. Finally, the feature extractor parameters are estimated under a variational Bayesian framework to incorporate the prior information and obtain corresponding posterior distributions. The proposed methodology is applied to solve a fouling monitoring problem for an industrial oil production process.

[*] This chapter has been published as: **V. K. Puli** and B. Huang, "Variational Bayesian Approach to Nonstationary and Oscillatory Slow Feature Analysis With Applications in Soft Sensing and Process Monitoring," in IEEE Transactions on Control Systems Technology, doi: 10.1109/TCST.2023.3240980.

4.1 Introduction

Although reliable, the first principle-based approaches for complex industrial processes are seldom available and are time-consuming for domain experts to derive even if they are available. Due to superior instrumentation and sensor technologies, a massive amount of process data are already available. Therefore, process history-based data-driven strategies are favourably employed for diverse applications. Operational tasks, such as predictive modelling [5], fault diagnosis [6–8], quality monitoring, plant-wide oscillation detection [9–12], causality analysis [13–16], are greatly simplified by utilizing machine learning and deep neural network techniques. Although data-driven approaches do not require a pre-determined model, they suffer from several other issues. Primarily, the presence of highly correlated variables may result in an input-output model that often suffers from ill condition. Furthermore, the temporal relationships, non-linearity, and the underlying patterns in the data reduce the efficacy of the machine learning algorithms if not accounted for explicitly.

Latent variable models [88,89,120,121] are often employed in process modelling to overcome some of the issues mentioned earlier. Essentially, these techniques extract features from the raw data with statistical preferences. They are used for information compression as the dimension of the extracted features is typically lower than that of the original data. The most celebrated latent variable models include principal component analysis (PCA) [34, 35, 90], partial least squares [38, 39, 122], independent component analysis [40], slow feature analysis (SFA) [36], dynamic inner PCA (DiPCA) [123], and dynamic inner canonical correlation analysis (DiCCA) [124,125]. Further, the probabilistic versions [54,55,92–94] have been developed to address multiple complexities posed by the real-world dataset, such as uncertainties, outliers, and missing values.

Understanding the dynamic behaviour of the process is essential for equipment design, quality control and shutdown-startup procedure. SFA [36] is an unsupervised learning technique that can extract temporally related patterns. It has been successfully applied in statistical process monitoring [43, 45, 95], predictive modelling [126] and unit-wide fault detection [127] applications. Chao *et. al* [55] presented the probabilistic version of the slow feature model (PSFA) and provided the methodological details to estimate its parameters. The traditional PSFA was further extended to handle the outliers [56] and output time-varying time delays [128]. A few other works [96–98] were developed to incorporate the quality relevant information in slow feature extraction. The complex slow feature probabilistic model [129] was recently proposed to extract the slow oscillating patterns from noisy measured data where the oscillations are not apparent.

Due to aging equipment and feedstock changes, the chemical process data often exhibit non-stationary behaviour. A time series is said to have non-stationary behaviour if its statistical properties, more importantly, the mean and variance vary with time. The non-stationary patterns in a time series may be manifested due to the presence of unit roots, drifts, deterministic trends, and structural breaks [130]. Application of traditional PSFA, where the non-stationary behaviour is not modelled explicitly, will result in parameters with huge variances. Therefore, Zhao *et al.* [102] introduced a hybrid approach that combines deterministic SFA with cointegration analysis to extract features from non-stationary data. Further, Scott *et. al* (NSPSFA) [131] has developed a unit root-based slow feature framework to extract non-stationary features in a probabilistic framework.

Most of the extensions discussed so far assume the unknowns are non-random without any prior knowledge and hence employ the Expectation-Maximization algorithm to obtain point estimates. On the other hand, the Bayesian framework facilitates the incorporation of process knowledge and modelling preference in the form of prior distributions. Therefore Bayesian model identification is popular in machine learning [77], signal processing [132] and the engineering domain lately. In the Bayesian approach, the joint distribution over observed variables y and unobserved variables ν can be written as follows

$$p(y, \nu) = p(y|\nu)p(\nu)$$

Given an observed value y^* of y, the Bayesian inference involves the computation of posterior distribution $p(\nu|y = y^*)$. Unlike classical parameter learning, this is the primary advantage where ν is a non-random quantity. A further advantage is the computation of log model evidence $\log p(y)$ that serves as a metric to compare various models. Variational Bayesian inference (VBI) [133–136] framework was introduced to avoid the integral intractability that is inevitable in the presence of unobserved variables. VBI approaches were used to model causal relations [137], estimate time-varying time delays [138, 139], monitor multi-modal processes [140], identify models [141, 142] and develop soft sensors [143–145]. The variational Bayesian version of PSFA (VBPSFA) [99] was developed to model the uncertainties in the parameters of the slow feature model effectively. A transfer learning-based slow feature analysis technique was recently proposed using variational Bayesian learning [146] to transport information for performance enhancement.

However, the VBPSFA [99] model has several shortcomings that have not been approached earlier in the literature. It may not extract slow oscillating features due to the diagonal limitation on the state-transition matrix. The complex probabilistic slow feature analysis (CPSFA) was recently proposed [129] to address the related issue using the expectation-maximization (EM) framework, where the model parameters are treated as non-random quantities. Further, the estimated variance of the unobserved variables can be massive when VBPSFA is applied to non-stationary data since the non-stationary behaviour is not represented explicitly. Finally, a single entity was used to account for all the observed variables' noise variance, but each observed variable can have a different level of uncertainty. Therefore, we propose a novel technique called variational Bayesian complex probabilistic slow feature analysis (VBCPSFA) that can deal with the aforementioned issues. The contributions of the chapter are:

- A novel model is proposed to separate the drift-type non-stationary patterns and oscillating features naturally from the observed data.
- Each observed variable's noise is modelled independently to account for different levels of uncertainty.
- A truncated Gaussian distribution is employed to model the state-transition parameters to restrict the magnitude of the complex eigenvalue within the unit circle.
- The analytical expressions for the posterior distributions of all the unobserved variables are derived and presented using the variational Bayesian inference framework.

The remainder of this chapter proceeds as follows. Section 4.2 summarizes the essential ideas from the literature. Specifically, Sections 4.2.1, 4.2.2 and 4.2.3 discuss PSFA, CPSFA and VBPSFA, respectively. In Section 4.3, the proposed model is presented. It includes the mathematical formulation and prior distribution information, followed by parameter learning using variational inference. Section 4.4 demonstrates the efficiency of the proposed modelling algorithm using a numerical simulation and an industrial application to the once-through steam generator industrial facility. This chapter ends with closure remarks in Section 4.5. The list of notations and their corresponding descriptions are shown in Table 4.1.

4.2 Revisit

Wiskott and Sejnowski [36] coined the term "slow feature analysis", which can extract the slowly varying features from observed data. The probabilistic versions [54, 55] of slow feature analysis have been developed in recent years, and they are more flexible in dealing with real-world data.

4.2.1 Probabilistic Slow Feature Analysis

PSFA is an extension of SFA using a probabilistic framework [147]. It constitutes a linear Gaussian state-space model to describe the feature dynamics in the latent space explicitly. The PSFA formulation [55] can be presented using (4.1) - (4.2).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k^1; \quad \boldsymbol{w}_k^1 \sim \mathcal{N}(0, Q) \tag{4.1}$$

$$\boldsymbol{y}_k^s = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(0, \Gamma^{-1}) \tag{4.2}$$

where $A \in \mathbb{R}^{m\times m}$, $C \in \mathbb{R}^{p\times m}$, $Q \in \mathbb{R}^{m\times m}$, and $\Gamma \in \mathbb{R}^{p\times p}$ denote the state-transition matrix, the emission matrix, the state-noise covariance matrix, and the measurement noise precision matrix, respectively. A and Q are assumed to be diagonal to obtain uncorrelated slow features. The state variance is considered to be unity to obtain stationary slow features and avoid a trivial solution. This leads to (4.3), where a_i and q_i are the ith diagonal entries of A and Q respectively.

$$a_i^2 + q_i = 1 \tag{4.3}$$

Equation (4.4) restricts the diagonal entries of matrix A to ensure stability.

$$a_i \in \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{4.4}$$

Table 4.1: List of Notations

Symbol	Description	Symbol	Description	Symbol	Description
k	Time instant.	$\mathbb{R}$	Set of real numbers.	I_p	Identity matrix of size p.
$\boldsymbol{s}_k$	Slow feature vector at k^{th} instant.	$\boldsymbol{w}_k^1$	Slow feature noise vector at k^{th} instant.	m	Number of features.
$\boldsymbol{y}_k$	Measured vector at k^{th} instant.	$\boldsymbol{v}_k$	Measurement-noise vector at k^{th} instant.	p	Number of measured variables.
A	Slow feature transition matrix.	Q	Slow feature noise covariance matrix.	C	Matrix mapping from $\boldsymbol{s}_k$ to $\boldsymbol{y}_k^s$.
$\boldsymbol{\nu}$	Unobserved random variables set.	$\boldsymbol{\nu}_s$	Subset of $\boldsymbol{\nu}$.	$\boldsymbol{\nu}_{\setminus s}$	Complimentary set i.e., $\boldsymbol{\nu}$ - $\boldsymbol{\nu}_s$.
q_i	i$^{\text{th}}$ diagonal entry of matrix Q.	$\boldsymbol{a}$	A's eigenvalues real part vector.	$\boldsymbol{b}$	A's eigenvalues imaginary part vector.
$\boldsymbol{h}_k$	Non-stationary feature vector.	$\boldsymbol{w}_k^2$	Non-stationary feature noise vector	$\boldsymbol{y}_k^s$	Stationary observed vector.
$\boldsymbol{y}_k^{ns}$	Non-stationary observed vector.	Θ	Diagonal drift matrix.	$\boldsymbol{\theta}$	Diagonal vector of Θ.
D	Matrix mapping from $\boldsymbol{s}_k$ to $\boldsymbol{y}_k^{ns}$.	E	Matrix mapping from $\boldsymbol{h}_k$ to $\boldsymbol{y}_k^{ns}$.	m_1	No. of oscillating slow features
p_1	No. of stationary measured variables	p_2	No. of non-stationary measured variables	m_2	No. of non-stationary features
Δ	Non-stationary feature noise precision matrix.	Γ	Stationary measurement noise precision matrix.	Λ	Non-stationary measurement noise precision matrix.
$\boldsymbol{\delta}$	Diagonal vector of Δ.	$\boldsymbol{\gamma}$	Diagonal vector of Γ.	$\boldsymbol{\lambda}$	Diagonal vector of Λ.
m_a	Mean of a_i's prior distribution.	m_b	Mean of b_i's prior distribution.	$\boldsymbol{\theta}_0$	Mean of $\boldsymbol{\theta}$'s prior distribution.
v_a	Variance of a_i's prior distribution.	v_b	Variance of b_i's prior distribution.	V_0	Covariance of $\boldsymbol{\theta}$'s prior distribution.
$\boldsymbol{c}_i$	i$^{\text{th}}$ column vector of C^T.	$\boldsymbol{d}_i$	i$^{\text{th}}$ column vector of D^T.	$\boldsymbol{e}_i$	i$^{\text{th}}$ column vector of E^T.
$\boldsymbol{m}_{c_i}$	Mean of $\boldsymbol{c}_i$'s prior distribution.	$\boldsymbol{m}_{d_i}$	Mean of $\boldsymbol{d}_i$'s prior distribution.	$\boldsymbol{m}_{e_i}$	Mean of $\boldsymbol{e}_i$'s prior distribution.
V_{c_i}	Covariance of $\boldsymbol{c}_i$'s prior distribution.	V_{d_i}	Covariance of $\boldsymbol{d}_i$'s prior distribution.	V_{e_i}	Covariance of $\boldsymbol{e}_i$'s prior distribution.
α_δ	δ_i's prior distribution shape parameter.	α_λ	λ_i's prior distribution shape parameter.	α_γ	γ_i's prior distribution shape parameter.
β_δ	δ_i's prior distribution rate parameter.	β_λ	λ_i's prior distribution rate parameter.	β_γ	γ_i's prior distribution rate parameter.
$\boldsymbol{x}_k$	Slow features with eigenvalues $a+ib$	$\boldsymbol{z}_k$	Slow features with eigenvalues $a-ib$	$\boldsymbol{e}_{1:N}$	$= [\boldsymbol{s}_{1:N}; \boldsymbol{h}_{1:N}]$
π	Set of hyper-parameters.	$\mathbb{E}$	Expected value	N	No. of data points

4.2.2 Complex Probabilistic Slow Feature Analysis

The probabilistic slow feature analysis formulation assumes that the state-transition matrix is diagonal; hence, the PSFA cannot extract the oscillating patterns. Hence, Puli *et. al* [129] proposed the complex probabilistic slow feature analysis to accommodate complex poles and encode oscillating features naturally. The CPSFA can be formulated using (4.1) - (4.2), but A is assumed to be block diagonal matrix to accommodate complex eigenvalues, as shown in (4.5).

$$A = blkdiag\left\{\begin{bmatrix} a_1 & b_1 \\ -b_1 & a_1 \end{bmatrix}, \ldots, \begin{bmatrix} a_{\frac{m}{2}} & b_{\frac{m}{2}} \\ -b_{\frac{m}{2}} & a_{\frac{m}{2}} \end{bmatrix}\right\}; \tag{4.5}$$

The state-noise covariance matrix adjusts to the structure, as shown in (4.6), due to the constraint given by (4.3).

$$Q = blkdiag\{Q_1, Q_2, \ldots, Q_{\frac{m}{2}}\}; \quad Q_j = (1 - a_j^2 - b_j^2)I_2 \tag{4.6}$$

The slow feature $\boldsymbol{s}_k$ exhibits oscillating characteristics due to the presence of complex eigenvalues in the state-transition matrix. Finally, the optimal point parameter estimates were obtained readily using the EM algorithm in an iterative manner.

4.2.3 VBI for parameter learning

EM framework can only provide point parameter estimates and cannot incorporate prior process knowledge as it is a purely data-driven approach. On the other hand, Bayesian methods can integrate process information using prior distributions and compute posterior distributions. The calculation of posterior distributions and log model evidence is computationally expensive and often intractable due to unobserved random variables. Therefore, a statistical framework called Variational Bayesian Inference [77, 79, 148] was proposed to overcome such issues. The log model evidence can be decomposed as the sum of variational free energy and KL divergence using (4.7).

$$\begin{aligned}\log p(\boldsymbol{y}) &= F(q(\boldsymbol{\nu})) + \mathrm{KL}(q(\boldsymbol{\nu})||p(\boldsymbol{\nu}|\boldsymbol{y})) \\ &= \int q(\nu) \log \left(\frac{p(\boldsymbol{y}, \boldsymbol{\nu})}{q(\boldsymbol{\nu})} \right) \, d\boldsymbol{\nu} + \int q(\boldsymbol{\nu}) \log \left(\frac{q(\boldsymbol{\nu})}{p(\boldsymbol{\nu}|\boldsymbol{y})} \right) \, d\boldsymbol{\nu}\end{aligned} \tag{4.7}$$

where $q(\boldsymbol{\nu}), p(\boldsymbol{\nu}|\boldsymbol{y})$ and $p(\boldsymbol{\nu}, \boldsymbol{y})$ denote the variational distribution over unobserved random vector $\boldsymbol{\nu}$, true posterior distribution and the generative model, respectively. The variational free energy, also called the lower bound, is maximized with respect to proposal distributions, iteratively. Further, the mean-field approximation is introduced to reduce the complexity of the solution, as shown in (4.8).

$$q(\boldsymbol{\nu}) \propto q(\boldsymbol{\nu}_s) q(\boldsymbol{\nu}_{\setminus s}) \tag{4.8}$$

where $\boldsymbol{\nu}_s$ denotes a subset of unobserved variables and $\boldsymbol{\nu}_{\setminus s}$ indicates its complimentary set. Finally, the optimal variational distribution can be obtained using variational calculus as shown in (4.9).

$$\log\, q(\boldsymbol{\nu}_s) \propto \langle \log\, p(\boldsymbol{y}, \boldsymbol{\nu}) \rangle_{q(\boldsymbol{\nu}_{\setminus s})} \tag{4.9}$$

Readers are referred to Section 2.4 for a detailed explanation. The Bayesian versions of PSFA models can be obtained, as shown in [99,100]. The preference for slowness has been implemented by assuming Beta prior distribution for A. The prior distributions of the emission matrix and the noise precision variable are assumed to be Gaussian and Gamma, respectively.

$$A = diag(a_1, a_2, \ldots, a_m); \quad p(a_i) = \mathrm{Beta}(\alpha_a, \beta_a)$$

$$C = \begin{bmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \dots & \boldsymbol{c}_p\end{bmatrix}^T; \quad p(\boldsymbol{c}_i) = \mathcal{N}(\boldsymbol{0}, {\Delta_0}^{-1})$$

$$\Gamma = \gamma I_p; \quad p(\gamma) = Gamma(\alpha_\gamma, \beta_\gamma) \tag{4.10}$$

4.3 VBCPSFA for non-stationary process

Although CPSFA may be used to extract slow oscillating features, the EM-based approach has its limitations, as mentioned in subsection 4.2.3. Further, the CPSFA formulation forces the variance of hidden features to unity, and hence, it may only extract stationary features. Therefore, we propose a novel approach called variational Bayesian complex probabilistic slow feature analysis to address these shortcomings. More specifically, it can derive both non-stationary features and slow oscillating features from the data with independent noise modelling.

4.3.1 Mathematical Formulation

The non-stationary behaviour is modelled using the random-walk with drift type hidden feature. In contrast to the pure random walk model [131], a random walk with a drift [149] contains a deterministic trend, resulting from the variable Θ. Therefore, a constant mean and constant variance is not maintained in the data. Since the eigenvalues of the non-stationary feature state-transition matrix do not lie strictly within the unit circle, this segment can be used to model slow and indefinitely growing signals. The VBCPSFA can be formulated using (4.11) - (4.12).

$$\begin{bmatrix}\boldsymbol{s}_k \\ \boldsymbol{h}_k\end{bmatrix} = \begin{bmatrix}A & 0 \\ 0 & I_{m_2}\end{bmatrix} \begin{bmatrix}\boldsymbol{s}_{k-1} \\ \boldsymbol{h}_{k-1}\end{bmatrix} + \begin{bmatrix}0 \\ \Theta\end{bmatrix} \begin{bmatrix}1\end{bmatrix} + \begin{bmatrix}\boldsymbol{w}_k^1 \\ \boldsymbol{w}_k^2\end{bmatrix}; \tag{4.11}$$

$$\begin{bmatrix}\boldsymbol{y}_k^s \\ \boldsymbol{y}_k^{ns}\end{bmatrix} = \begin{bmatrix}C & 0 \\ D & E\end{bmatrix} \begin{bmatrix}\boldsymbol{s}_k \\ \boldsymbol{h}_k\end{bmatrix} + \boldsymbol{v}_k; \tag{4.12}$$

where

$$\begin{bmatrix}\boldsymbol{w}_k^1 \\ \boldsymbol{w}_k^2\end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix}I_{m_1} - AA^T & 0 \\ 0 & \Delta^{-1}\end{bmatrix}\right);$$

$$\boldsymbol{v}_k \sim \mathcal{N}\left(0, \begin{bmatrix}\Gamma^{-1} & 0 \\ 0 & \Lambda^{-1}\end{bmatrix}\right);$$

Let

$$\boldsymbol{a} = \begin{bmatrix}a_1 & a_2 & \dots & a_{\frac{m_1}{2}}\end{bmatrix}^T; \quad \boldsymbol{b} = \begin{bmatrix}b_1 & b_2 & \dots & b_{\frac{m_1}{2}}\end{bmatrix}^T$$

$$\begin{aligned}
\Theta &= \mathrm{diag}(\boldsymbol{\theta}); \quad \boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2 & \dots & \theta_{m_2}\end{bmatrix}^T; \\
\Delta &= \mathrm{diag}(\boldsymbol{\delta}); \quad \boldsymbol{\delta} = \begin{bmatrix}\delta_1 & \delta_2 & \dots & \delta_{m_2}\end{bmatrix}^T; \\
\Lambda &= \mathrm{diag}(\boldsymbol{\lambda}); \quad \boldsymbol{\lambda} = \begin{bmatrix}\lambda_1 & \lambda_2 & \dots & \lambda_{p_1}\end{bmatrix}^T; \\
\Gamma &= \mathrm{diag}(\boldsymbol{\gamma}); \quad \boldsymbol{\gamma} = \begin{bmatrix}\gamma_1 & \gamma_2 & \dots & \gamma_{p_2}\end{bmatrix}^T;
\end{aligned}$$

where $\boldsymbol{s}_k \in \mathbb{R}^{m_1 \times 1}$, $\boldsymbol{y}_k^s \in \mathbb{R}^{p_1 \times 1}$ and $\boldsymbol{y}_k^{ns} \in \mathbb{R}^{p_2 \times 1}$ denote the oscillating slow feature, stationary, and non-stationary observed variable, respectively. The drift-type non-stationary behaviour is captured by $\boldsymbol{h}_k \in \mathbb{R}^{m_2 \times 1}$. Further, $A \in \mathbb{R}^{m_1 \times m_1}$, $\Theta \in \mathbb{R}^{m_2 \times m_2}$, $\{C \in \mathbb{R}^{p_1 \times m_1}$, $D \in \mathbb{R}^{p_2 \times m_1}$, and $E \in \mathbb{R}^{p_2 \times m_2}\}$ represent block diagonal state-transition matrix, diagonal drift matrix and block-wise emission matrices, respectively. The precision matrices of non-stationary state variable noise, observed stationary variable noise, and observed non-stationary variable noise are denoted by $\Delta \in \mathbb{R}^{m_2 \times m_2}$, $\Lambda \in \mathbb{R}^{p_1 \times p_1}$ and $\Gamma \in \mathbb{R}^{p_2 \times p_2}$, respectively. Unlike a single random variable in (4.10), the precision matrices are proposed to be diagonal to accommodate different uncertainty levels for different variables.

4.3.2 Prior distribution information

Now we introduce prior distributions of various random variables using either modelling preference or conjugate distribution properties, as shown below.

1. A truncated Gaussian prior distribution between 0 and 1 is utilized similar to [146] for the real part of eigenvalue, as shown in (4.13).

$$p(\boldsymbol{a}) = \prod_{i=1}^{\frac{m_1}{2}} p(a_i|0 < a_i < 1) \tag{4.13}$$

where

$$\begin{aligned}
p(a_i|0 < a_i < 1) &= \mathcal{TN}(m_a, v_a, 0, 1) \\
&= \frac{1}{\sqrt{v_a}} \frac{\Phi\left(\frac{a_i - m_a}{\sqrt{v_a}}\right)}{\Psi\left(\frac{1-m_a}{\sqrt{v_a}}\right) - \Psi\left(\frac{-m_a}{\sqrt{v_a}}\right)} \\
\Phi(\xi) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\xi^2}{2}\right) \\
\Psi(z) &= \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right)
\end{aligned}$$

where m_a and v_a denote the mean and variance parameters of the prior distribution, respectively.

2. A truncated Gaussian prior distribution is proposed for the imaginary part of the eigenvalue between 0 and $\sqrt{1-\boldsymbol{a}^2}$ to ensure the norm of the eigenvalue is smaller than one, as shown in (4.14).

$$p(\boldsymbol{b}) = \prod_{i=1}^{\frac{m_1}{2}} p(b_i|0 < b_i < \sqrt{1-a_i^2}) \tag{4.14}$$

where

$$\begin{aligned} &p(b_i|0 < b_i < \sqrt{1-a_i^2}) \\ &\quad = \mathcal{TN}(m_b, v_b, 0, \sqrt{1-a_i^2}) \\ &\quad = \frac{1}{\sqrt{v_b}} \frac{\Phi\left(\frac{b_i - m_b}{\sqrt{v_b}}\right)}{\Psi\left(\frac{\sqrt{1-a_i^2}-m_b}{\sqrt{v_b}}\right) - \Psi\left(\frac{-m_b}{\sqrt{v_b}}\right)} \end{aligned}$$

where m_b and v_b denote the mean and variance parameters of the prior distribution, respectively.

3. A Gaussian prior distribution is assumed for the drift random variable $\boldsymbol{\theta}$ to facilitate the learning process as shown in (4.15).

$$\begin{aligned} p(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta}_0, V_0) \\ &= \frac{|V_0|^{-\frac{1}{2}}}{(2\pi)^{\frac{m_2}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T {V_0}^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\right) \end{aligned} \tag{4.15}$$

where $\boldsymbol{\theta}_0 \in \mathbb{R}^{m_2 \times 1}$ and $V_0 \in \mathbb{R}^{m_2 \times m_2}$ denote the mean and variance of the prior distribution, respectively.

4. We assume Gamma distribution for the precision variable $\boldsymbol{\delta}$ as shown in (4.16) since $\boldsymbol{\delta}$ is always greater than zero.

$$p(\boldsymbol{\delta}) = \prod_{i=1}^{m_2} p(\delta_i) \tag{4.16}$$

where

$$p(\delta_i) = \text{Gamma}(\alpha_\delta, \beta_\delta) = \frac{{\beta_\delta}^{\alpha_\delta}}{\Gamma(\alpha_\delta)} \delta_i^{\alpha_\delta - 1} \exp(-\delta_i \beta_\delta)$$

where α_δ and β_δ indicate the shape and rate parameters of the prior distribution, respectively.

5. We assume Gaussian distribution owing to conjugate distributional properties for the emission matrix C that relates the stationary observed variables and slow oscillating features as shown in (4.17).

$$p(C) = \prod_{i=1}^{p_1} p(\boldsymbol{c}_i) \tag{4.17}$$

where

$$\begin{aligned} C &= \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_{p_1} \end{bmatrix}^T; \\ p(\boldsymbol{c}_i) &= \mathcal{N}(\boldsymbol{m}_{c_i}, V_{c_i}) \\ &= \frac{|V_{c_i}|^{-\frac{1}{2}}}{(2\pi)^{\frac{m_1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{c}_i - \boldsymbol{m}_{c_i})^T {V_{c_i}}^{-1}(\boldsymbol{c}_i - \boldsymbol{m}_{c_i})\right) \end{aligned}$$

and $\boldsymbol{c}_i \in \mathbb{R}^{m_1 \times 1}$, $\boldsymbol{m}_{c_i} \in \mathbb{R}^{m_1 \times 1}$ and $V_{c_i} \in \mathbb{R}^{m_1 \times m_1}$ denote the i^{th} row vector of C, mean and covariance parameters of the prior distribution, respectively.

6. We consider a normal distribution for the emission matrix D that relates the non-stationary observed variables and slow oscillating features as shown in (4.18).

$$p(D) = \prod_{i=1}^{p_2} p(\boldsymbol{d}_i) \tag{4.18}$$

where

$$\begin{aligned} D &= \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_{p_2} \end{bmatrix}^T; \\ p(\boldsymbol{d}_i) &= \mathcal{N}(\boldsymbol{m}_{d_i}, V_{d_i}) \\ &= \frac{|V_{d_i}|^{-\frac{1}{2}}}{(2\pi)^{\frac{m_1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{d}_i - \boldsymbol{m}_{d_i})^T {V_{d_i}}^{-1}(\boldsymbol{d}_i - \boldsymbol{m}_{d_i})\right) \end{aligned}$$

where $\boldsymbol{d}_i \in \mathbb{R}^{m_1 \times 1}$, $\boldsymbol{m}_{d_i} \in \mathbb{R}^{m_1 \times 1}$ and $V_{d_i} \in \mathbb{R}^{m_1 \times m_1}$ denote the i^{th} row vector of D, mean and covariance parameters of the prior distribution, respectively.

7. A Gaussian distribution is considered due to modelling preference for the emission matrix E that associates the non-stationary observed variables with the

random-walk drift features as shown in (4.19).

$$p(E) = \prod_{i=1}^{p_2} p(\boldsymbol{e}_i) \tag{4.19}$$

where

$$\begin{aligned} E &= \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{p_2} \end{bmatrix}^T; \\ p(\boldsymbol{e}_i) &= \mathcal{N}(\boldsymbol{m}_{e_i}, V_{e_i}) \\ &= \frac{|V_{e_i}|^{-\frac{1}{2}}}{(2\pi)^{\frac{m_2}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{e}_i - \boldsymbol{m}_{e_i})^T {V_{e_i}}^{-1}(\boldsymbol{e}_i - \boldsymbol{m}_{e_i})\right) \end{aligned}$$

where $\boldsymbol{e}_i \in \mathbb{R}^{m_2 \times 1}$, $\boldsymbol{m}_{e_i} \in \mathbb{R}^{m_2 \times 1}$ and $V_{e_i} \in \mathbb{R}^{m_2 \times m_2}$ denote the i^{th} row vector of E, mean and covariance parameters of the prior distribution, respectively.

8. We assume Gamma distribution for the precision matrices $\boldsymbol{\lambda}, \boldsymbol{\gamma}$ to facilitate the parameter learning, as shown in (4.20).

$$p(\boldsymbol{\lambda}) = \prod_{i=1}^{p_1} p(\lambda_i); \quad p(\boldsymbol{\gamma}) = \prod_{i=1}^{p_2} p(\gamma_i) \tag{4.20}$$

where

$$\begin{aligned} p(\lambda_i) &= \text{Gamma}(\alpha_\lambda, \beta_\lambda) = \frac{{\beta_\lambda}^{\alpha_\lambda}}{\Gamma(\alpha_\lambda)} \lambda_i^{\alpha_\lambda - 1} \exp(-\lambda_i \beta_\lambda) \\ p(\gamma_i) &= \text{Gamma}(\alpha_\gamma, \beta_\gamma) = \frac{{\beta_\gamma}^{\alpha_\gamma}}{\Gamma(\alpha_\gamma)} \gamma_i^{\alpha_\gamma - 1} \exp(-\gamma_i \beta_\gamma) \end{aligned}$$

where $\alpha_\lambda, \alpha_\gamma$ and $\beta_\lambda, \beta_\gamma$ indicate the shape and rate parameters of the prior distribution, respectively.

9. The conditional distributions of the hidden and observed variables are shown in (4.21) considering Gaussian distribution properties.

$$\begin{aligned} p(\boldsymbol{s}_k | \boldsymbol{s}_{k-1}, \boldsymbol{a}, \boldsymbol{b}) &= \mathcal{N}(A\boldsymbol{s}_{k-1}, I_{m_1} - AA^T) \\ p(\boldsymbol{h}_k | \boldsymbol{h}_{k-1}, \boldsymbol{\theta}, \delta) &= \mathcal{N}(\boldsymbol{h}_{k-1} + \boldsymbol{\theta}, \Delta^{-1}) \\ p(\boldsymbol{y}_k^s | \boldsymbol{s}_k, C, \boldsymbol{\lambda}) &= \mathcal{N}(C\boldsymbol{s}_k, \Lambda^{-1}) \\ p(\boldsymbol{y}_k^{ns} | \boldsymbol{s}_k, \boldsymbol{h}_k, D, E, \boldsymbol{\gamma}) &= \mathcal{N}(D\boldsymbol{s}_k + E\boldsymbol{h}_k, \Gamma^{-1}) \end{aligned} \tag{4.21}$$

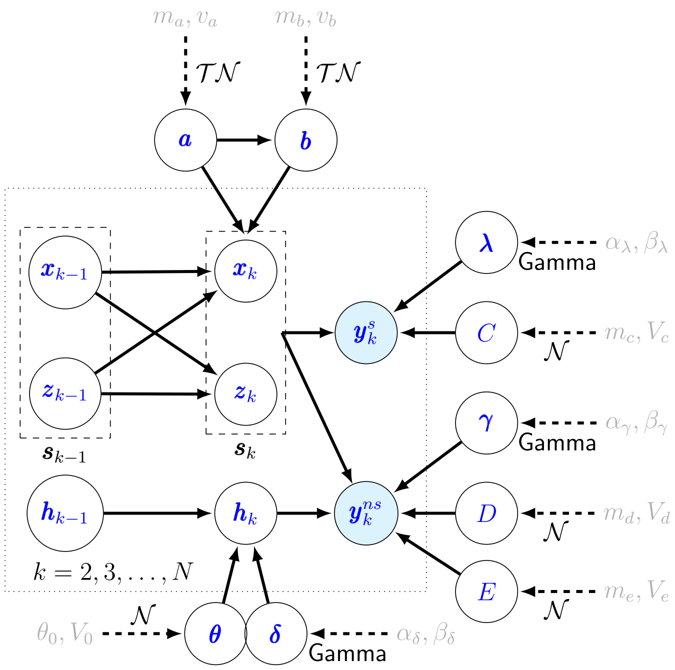


Figure 4.1: Probabilistic graphical model of VBCPSFA

The graphical depiction of VBCPSFA is shown in Fig. 4.1. Finally, the set of user-chosen prior parameters, unobserved random variables and observed variables are indicated by $\pi \in \{\boldsymbol{m}_a, \boldsymbol{v}_a, \boldsymbol{m}_b, \boldsymbol{v}_b, \boldsymbol{\theta}_0, V_0, \alpha_\delta, \beta_\delta, \boldsymbol{m}_{c_i}, V_{c_i}, \boldsymbol{m}_{d_i}, V_{d_i}, \boldsymbol{m}_{e_i}, V_{e_i}, \alpha_\lambda, \ \beta_\lambda, \alpha_\gamma, \beta_\gamma\}$, $\nu \in \{\boldsymbol{s}_{1:N}, \boldsymbol{h}_{1:N}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\delta}, C, D, E, \boldsymbol{\lambda}, \boldsymbol{\gamma}\}$, and $Y \in \{\boldsymbol{y}^s_{1:N},\ \boldsymbol{y}^{ns}_{1:N}\}$, respectively. The observed variables, unobserved random variables and variational prior parameters are denoted by shaded circle, white circle and text without circle, respectively. Here x_k and z_k represent the slow oscillating features corresponding to complex eigenvalues $a + ib$ and $a - ib$, respectively. The joint distribution over the observable and the unobservable variables is formulated to derive the update expressions for the parameters governing the variational distributions, as shown in (4.22).

$$
\begin{aligned}
&\log p(Y, S, H, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\delta}, C, D, E, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \\
&\quad = \log\ p(y_{1:N}, h_{1:N}, s_{1:N} | \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\delta}, C, D, E, \boldsymbol{\lambda}, \boldsymbol{\gamma}) + \log\ p(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\delta}, C, D, E, \boldsymbol{\lambda}, \boldsymbol{\gamma}); \\
&\quad\quad = \log p(\boldsymbol{s}_1) + \log p(\boldsymbol{h}_1) + \sum_{k=2}^{N} \log\ p(\boldsymbol{s}_k | \boldsymbol{s}_{k-1}, \boldsymbol{a}, \boldsymbol{b}) + \sum_{k=2}^{N} \log p(\boldsymbol{h}_k | \boldsymbol{h}_{k-1}, \boldsymbol{\theta}, \boldsymbol{\delta}) \\
&\quad + \sum_{k=1}^{N} \log p(\boldsymbol{y}_k^s | \boldsymbol{s}_k, C, \boldsymbol{\lambda}) + \sum_{k=1}^{N} \log\ p(\boldsymbol{y}_k^{ns} | \boldsymbol{s}_k, \boldsymbol{h}_k, D, E, \boldsymbol{\gamma}) + \sum_{i=1}^{\frac{m_1}{2}} \log\ p(a_i | m_{a_i}, v_{a_i})
\end{aligned}
$$

$$+\sum_{i=1}^{\frac{m_1}{2}} \log\, p(b_i|a_i, m_{b_i}, v_{b_i}) + \sum_{i=1}^{m_2} \log p(\delta_i|\alpha_\delta, \beta_\delta) + \log p(\boldsymbol{\theta}|\boldsymbol{\theta}_0, V_0) + \sum_{i=1}^{p_1} \log p(\boldsymbol{c}_i|\boldsymbol{m}_{c_i}, V_{c_i})$$
$$+\sum_{i=1}^{p_2} \log p(\boldsymbol{d}_i|\boldsymbol{m}_{d_i}, V_{d_i}) + \sum_{i=1}^{p_2} \log p(\boldsymbol{e}_i|\boldsymbol{m}_{e_i}, V_{e_i}) + \sum_{i=1}^{p_1} \log p(\lambda_i|\alpha_\lambda, \beta_\lambda) + \sum_{i=1}^{p_2} \log p(\gamma_i|\alpha_\gamma, \beta_\gamma); \tag{4.22}$$

4.3.3 Proposal distributions

Equation (4.9) is applied for each unobserved variable ν, and the variational parameter update equations are presented below. Further, the functional expectations of an unobserved variable are computed and subsequently used in updating other variable parameters. Here $\langle\cdot\rangle_k$ stands for the statistical expectation of a random variable and N denotes the number of data samples. **Note:** The iteration number η is omitted for all the random variables in this subsection for ease of notation.

1. The proposal distribution of the latent variables $\boldsymbol{e}_{1:N} = [\boldsymbol{s}_{1:N}; \boldsymbol{h}_{1:N}]$ can be written as shown in (4.23).

   $$\begin{aligned}\log q(\boldsymbol{e}_{1:N}) \propto &\langle \log p(\boldsymbol{y}_{1:N}, \nu)\rangle_{q(\nu_{\setminus \boldsymbol{e}_{1:N}})} \\ \propto &\langle \log p(\boldsymbol{e}_{1:N}|\boldsymbol{y}_{1:N}, \nu_{\setminus \boldsymbol{e}_{1:N}})\rangle_{q(\nu_{\setminus e_{1:N}})}\end{aligned} \tag{4.23}$$

   However, the expected value of the function $\log p(\boldsymbol{e}_{1:N}|\boldsymbol{y}_{1:N}, \nu_{\setminus \boldsymbol{e}_{1:N}})$ over the variational distribution $q(\nu_{\setminus \boldsymbol{e}_{1:N}})$ is generally not equal to the function with averaged parameters as shown below.

   $$\langle \log p(\boldsymbol{e}_{1:N}|\boldsymbol{y}_{1:N}, \nu_{\setminus \boldsymbol{e}_{1:N}})\rangle_{q(\nu \setminus \boldsymbol{e}_{1:N})} \neq \log p(\boldsymbol{e}_{1:N}|\boldsymbol{y}_{1:N}, \langle \nu_{\setminus \boldsymbol{e}_{1:N}}\rangle_{q(\nu_{\setminus \boldsymbol{e}_{1:N}})})$$

   We implement the mean and fluctuation decomposition theorem discussed by David Barber [150] to infer the variational distribution $q(e_{1:N})$ using the classical Kalman-Rauch-Tung-Striebel smoothing algorithm. Specifically, the expectations *viz.* $\langle \boldsymbol{e}_k^\eta \rangle$, $\langle \boldsymbol{e}_k^\eta \boldsymbol{e}_k^{\eta T}\rangle$, and $\langle \boldsymbol{e}_{k-1}^\eta \boldsymbol{e}_k^{\eta T}\rangle$ are computed. Readers are referred to [99, 146, 150] for more details.

2. The proposal distribution of $\boldsymbol{\theta}$ is Gaussian whose hyper-parameters are derived as

   $$V_\theta^\eta = \left\{ {V_0}^{-1} + \langle\Delta\rangle(N-1) \right\}^{-1}$$

$$\boldsymbol{m}_{\theta}^{\eta} = V_{\theta}^{\eta} \left(\langle \Delta \rangle \sum_{k=2}^{N} \langle \boldsymbol{h}_k \rangle - \langle \Delta \rangle \sum_{k=2}^{N} \langle \boldsymbol{h}_{k-1} \rangle + {V_0}^{-1} \theta_0 \right)$$

3. The derived proposal distribution of δ_i is Gamma:

$$\begin{aligned} \alpha_{\delta_i}^{\eta} &= \alpha_{\delta} + \frac{(N-1)}{2} \\ \beta_{\delta_i}^{\eta} &= \frac{1}{2} \sum_{k=2}^{N} \left(\langle {h_k^i}^2 \rangle + \langle {h_{k-1}^i}^2 \rangle \right) + \frac{N-1}{2} \langle \theta_i^2 \rangle \\ &\qquad - \sum_{k=2}^{N} \langle h_k^i h_{k-1}^i \rangle - \sum_{k=2}^{N} \langle h_k^i - h_{k-1}^i \rangle \langle \theta_i \rangle + \beta_{\delta} \end{aligned}$$

4. The derived proposal distribution of $\boldsymbol{c}_i$ is Gaussian:

$$\begin{aligned} V_{c_i}^{\eta} &= \left\{ {V_{c_i}}^{-1} + \langle \lambda_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle \right\}^{-1} \\ \boldsymbol{m}_{c_i}^{\eta} &= V_{c_i}^{\eta T} \left(\langle \lambda_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{s}_k \rangle \, y_k^{s^i} + {V_{c_i}}^{-1} \boldsymbol{m}_{c_i} \right) \end{aligned}$$

5. The derived proposal distribution of $\boldsymbol{d}_i$ is Gaussian:

$$\begin{aligned} V_{d_i}^{\eta} &= \left\{ {V_{d_i}}^{-1} + \langle \gamma_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle \right\}^{-1} \\ \boldsymbol{m}_{d_i}^{\eta} &= V_{d_i}^{\eta T} \left(\langle \gamma_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{s}_k^T \rangle \, y_k^{ns^i} + \boldsymbol{m}_{d_i}^T {V_{d_i}}^{-T} - \frac{\langle \gamma_i \rangle}{2} \sum_{k=1}^{N} \langle \boldsymbol{e}_i^T \rangle \langle \boldsymbol{h}_k \boldsymbol{s}_k^T \rangle \right)^T \end{aligned}$$

6. The derived proposal distribution of $\boldsymbol{e}_i$ is Gaussian:

$$\begin{aligned} V_{e_i}^{\eta} &= \left\{ {V_{e_i}}^{-1} + \langle \gamma_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{h}_k \boldsymbol{h}_k^T \rangle \right\}^{-1} \\ \boldsymbol{m}_{e_i}^{\eta} &= V_{e_i}^{\eta T} \left(\langle \gamma_i \rangle \sum_{k=1}^{N} \langle \boldsymbol{h}_k \rangle \, y_k^{ns^i} + {V_{e_i}}^{-1} \boldsymbol{m}_{e_i} - \frac{\langle \gamma_i \rangle}{2} \sum_{k=1}^{N} \langle \boldsymbol{h}_k \boldsymbol{s}_k^T \rangle \langle \boldsymbol{d}_i \rangle \right) \end{aligned}$$

7. The derived proposal distribution of λ_i is Gamma:

$$\begin{aligned} \alpha_{\lambda_i}^{\eta} =& \alpha_{\lambda} + \frac{N}{2} \\ \beta_{\lambda_i}^{\eta} =& \beta_{\lambda} + \frac{1}{2} \sum_{k=1}^{N} (y_k^{s^i})^2 - \sum_{k=1}^{N} y_k^{s^i} \langle \boldsymbol{c}_i \rangle^T \langle \boldsymbol{s}_k \rangle + \frac{1}{2} \sum_{k=1}^{N} tr \left(\langle \boldsymbol{c}_i \boldsymbol{c}_i^T \rangle \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle \right) \end{aligned}$$

8. The derived proposal distribution of γ_i is Gamma:

$$\alpha^{\eta}_{\gamma_i} = \alpha_{\gamma} + \frac{N}{2}$$

$$\beta^{\eta}_{\gamma_i} = \beta_{\gamma} + \frac{1}{2}\sum_{k=1}^{N}(y_k^{ns^i})^2 - \sum_{k=1}^{N} y_k^{ns^i}\langle \boldsymbol{d}_i\rangle^T\langle \boldsymbol{s}_k\rangle - \sum_{k=1}^{N} y_k^{ns^i}\langle \boldsymbol{e}_i\rangle^T\langle \boldsymbol{h}_k\rangle$$

$$+\frac{1}{2}\sum_{k=1}^{N}\ \mathrm{tr}\left(\langle \boldsymbol{d}_i\boldsymbol{d}_i^T\rangle\langle \boldsymbol{s}_k\boldsymbol{s}_k^T\rangle\right) + \frac{1}{2}\sum_{k=1}^{N} tr\left(\langle \boldsymbol{e}_i\boldsymbol{e}_i^T\rangle\langle \boldsymbol{h}_k\boldsymbol{h}_k^T\rangle\right) + \sum_{k=1}^{N} tr\left(\langle \boldsymbol{d}_i\rangle\langle \boldsymbol{e}_i^T\rangle\langle \boldsymbol{h}_k{\boldsymbol{s}_k}^T\rangle\right)$$

9. The proposal distribution of a_i, b_i up to a proportionality constant is shown in (4.24). Since the priors are chosen on the constraint that the resultant eigenvalue is within the unit circle, it is not conjugate to the Gaussian likelihood. Hence, the posterior distribution does not belong to any known families, and therefore, the update expressions of the posterior distribution parameters cannot be derived analytically.

$$q(a_i, b_i) \propto \tilde{q}(a_i, b_i)$$

where

$$\begin{aligned}\log\tilde{q}(a_i, b_i) = &-\frac{1}{2}\frac{a_i^2 - 2m_a a_i}{v_a} - \frac{1}{2}\frac{b_i^2 - 2m_b b_i}{v_b} - \log\left\{\mathrm{erf}\left(\frac{\sqrt{1-a_i^2} - m_b}{\sqrt{2v_b}}\right)\right.\\ &\left.-\mathrm{erf}\left(\frac{-m_b}{\sqrt{2v_b}}\right)\right\} - (N-1)\log|1 - a_i^2 - b_i^2| - \frac{1}{2}\sum_{k=2}^{N}\frac{\langle (x_k^i)^2 + (z_k^i)^2\rangle}{1 - a_i^2 - b_i^2}\\ &-\frac{1}{2}\sum_{k=2}^{N}\frac{(a_i^2 + b_i^2)\,\langle (x_{k-1}^i)^2 + (z_{k-1}^i)^2\rangle}{1 - a_i^2 - b_i^2} + a_i\sum_{k=2}^{N}\frac{\langle x_k^i x_{k-1}^i + z_k^i z_{k-1}^i\rangle}{1 - a_i^2 - b_i^2}\\ &+ b_i\sum_{k=2}^{N}\frac{\langle x_k^i z_{k-1}^i - z_k^i x_{k-1}^i\rangle}{1 - a_i^2 - b_i^2}\end{aligned} \tag{4.24}$$

The principal reason for writing the posterior distribution is to compute the distributional parameters and thus evaluate the functional expectations of the random variables. Hence, we employ the numerical sampling technique to obtain samples $\nu^1, \nu^2, \ldots, \nu^L$ independently from the target distribution $q(\nu)$ and approximate the expectation of some function $f(\nu)$ asymptotically as shown in (4.25)

$$\mathbb{E}_{q(\nu)}\{f(\nu)\} = \int f(\nu)\, q(\nu)\, d\nu \approx \hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(\nu^l) \tag{4.25}$$

where L denotes the number of drawn samples and $\hat{f}$ is the basic Monte Carlo estimator of $\mathbb{E}_{q(\nu)}\{f(\nu)\}$. Since the target distribution given by (4.24) is complex, it is impractical to sample directly from it. Therefore, biased importance sampling [99] can be applied to approximate the functional expectations directly using (4.26).

$$\hat{f} = \sum_{l=1}^{L} f(\nu^l)\hat{w}(\nu^l) \tag{4.26}$$

where $\hat{w}(\nu^l) = \frac{\tilde{w}(\nu^l)}{\sum_{l=1}^{L}\tilde{w}(\nu^l)}$ and $\tilde{w}(\nu^l) = \frac{\tilde{q}(\nu^l)}{\tilde{g}(\nu^l)}$. The samples $\nu^l \; \forall \; l \in \{1, 2, \dots L\}$ are drawn from an easier distribution $g(\nu)$, and the introduced bias is corrected by $\hat{w}$. The support distribution $g(\nu)$ is chosen to be the truncated Gaussian distribution for simplicity. The functional expectations of the transition variables required to update the parameters of other random variables are shown in (4.27).

$$\begin{aligned} f(a,b) \in \Bigg\{ & \frac{a^2+b^2}{1-a^2-b^2}, \frac{a}{1-a^2-b^2}, \frac{b}{1-a^2-b^2}, \frac{1}{1-a^2-b^2}, \log|1-a^2-b^2|, \\ & a^2, a, b^2, b, \; \log\left\{\Psi\left(\frac{\sqrt{1-a^2}-m_b}{\sqrt{v_b}}\right) - \Psi\left(\frac{-m_b}{\sqrt{v_b}}\right)\right\}\Bigg\} \end{aligned} \tag{4.27}$$

4.4 Simulation and Applications

In this section, the effectiveness of the proposed modelling algorithm in predictive modelling is investigated with the help of a simulation and an industrial case study. The prior parameters are selected based on the cross-validation data performance and preference for slow oscillations, as shown below:

- The mean and covariance are chosen to be zero vector and identity matrix for all the unobserved random variables whose prior is a Gaussian distribution.
- The shape and rate parameters are chosen to be 0.5 and unity for all the unobserved random variables with a Gamma distribution prior.
- The truncated Gaussian distribution prior parameters $\{m, v\}$ of the random variables a and b are selected to be $\{0.5, 0.04\}$ and $\{0.7, 0.025\}$, respectively, thus inducing a modelling preference for oscillations.

4.4.1 Numerical case study

It is shown in [129] that the velocity of the generated oscillating feature is dependent on the real part of the eigenvalue. Therefore, the block diagonal state-transition matrix shown in (4.28) is utilized to generate oscillating patterns with different velocities.

$$A^{c*} = blkdiag\left\{\begin{bmatrix} 0.96 & 0.21 \\ -0.21 & 0.96 \end{bmatrix}, \begin{bmatrix} 0.65 & 0.75 \\ -0.75 & 0.65 \end{bmatrix}\right\} \tag{4.28}$$

Further, $\boldsymbol{\theta}^*_{2\times1}, C^*_{1\times4}, D^*_{7\times4}, E^*_{7\times2}$ are randomly drawn matrices from the standard Gaussian distribution. $\boldsymbol{\delta}^*_{2\times1}$ is drawn from a standard uniform distribution to obtain positive values. The precision matrices $\Lambda^*_{1\times1}, \Gamma^*_{7\times7}$ are chosen so that the ratio of the variance of noise-free observed variables to the noise is equal to three to produce noisy observed variables. Finally, eight measured variables are generated with the help of six hidden features and corresponding parameters using (4.11)-(4.12). A quality variable $\boldsymbol{q}$ is produced, as shown in (4.29), with the help of the stationary oscillating features $\boldsymbol{s}^*$, a regression vector $\boldsymbol{m}^*$ and an additive noise $\boldsymbol{e}^*$.

$$\boldsymbol{q}_k = \boldsymbol{s}_k^{*T}\boldsymbol{m}^* + \boldsymbol{e}_k^* \tag{4.29}$$

where $\boldsymbol{m}^* \sim \mathcal{N}(0, 1)$. The observation data and their corresponding Pearson correlation coefficients (ρ) against the quality variable are shown in the Fig. 4.2. The generated dataset with 2000 data samples is divided into training and cross-validation sets. The hyper-parameters of the prior distributions are chosen with a preference for slowness and oscillating patterns. The extracted features using the VBCPSFA are shown in the Fig. 4.3. The optimal order, i.e., four stationary and two non-stationary features, in this case, is obtained based on the value of log model evidence $\log p(\boldsymbol{y})$ computed using the cross-validation data. The higher correlation coefficients indicate that the features possess superior prediction abilities.

Finally, we compare the proposed methodology with other linear unsupervised state-of-the-art models available in the literature for soft-sensing applications. The original data are used for all the dynamic models such as SFA, DiPCA, DiCCA, PSFA, NSPSFA, VBPSFA, CPSFA, and VBCPSFA. In contrast, the data are stacked with the time-delayed copies up to order two for static models like OLS and PCA, so the comparison is fair. The latent dimensions for the PCA, SFA, DiPCA, and

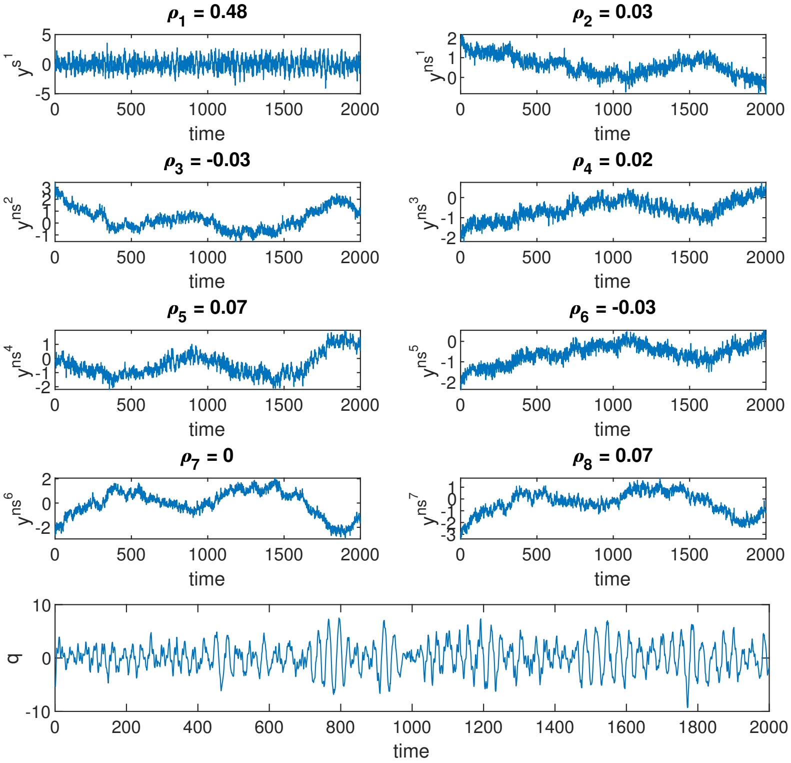


Figure 4.2: Observed dataset

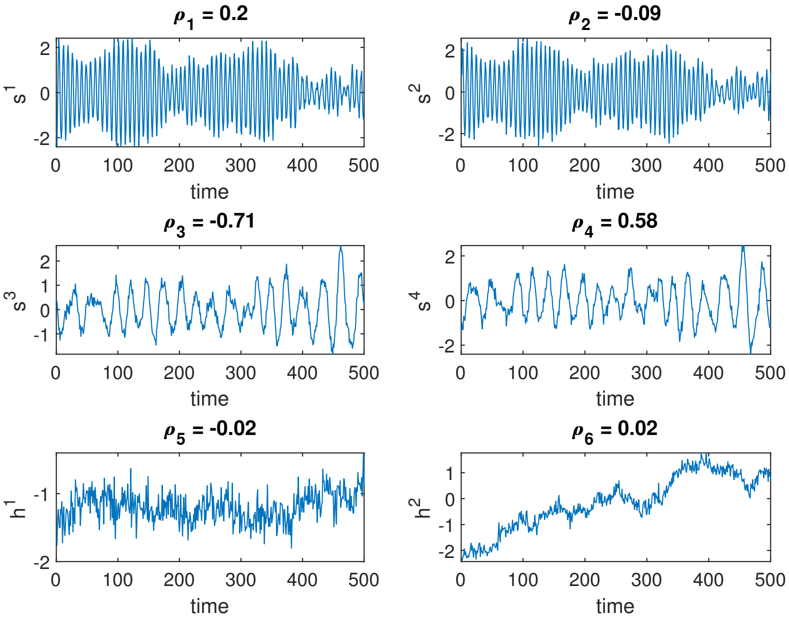


Figure 4.3: Extracted features (Only 500 data points are shown).

DiCCA are chosen based on the predictability of extracted features against the quality variable. On the other hand, the log-likelihood is utilized for the remaining iteration-based models. The performance metrics RMSE and ρ between the measured and the predicted quality variable, along with the latent dimension and CPU time, are shown in Table 4.2. We observe that the VBCPSFA features, followed by CPSFA, result in lower RMSE due to explicit modelling of the oscillations. The computation time is relatively higher since the parameters in the proposed algorithm are obtained in an iterative manner. Further, a Monte-Carlo simulation is performed in each iteration to compute the expectations of **a** and **b**, as shown in (4.25)-(4.27). It is observed that a higher dimensional process data mainly increases the dimension of C, D, E, Δ, Γ. Since the exact update expressions for all those parameters are explicitly available, an increase in the dimension does not incur additional computation time.

Table 4.2: Performance comparison

Method	latent dimension	ρ	RMSE	CPU time (s)
OLS	13	0.51	2.3	0.012
PCA	10	0.56	2.01	0.028
SFA	8	0.68	1.8	0.062
DiPCA	5	0.66	1.4	0.1
DiCCA	6	0.71	1.6	0.08
PSFA	11	0.55	2.16	13.12
VBPSFA	10	0.76	1.2	28.78
NSPSFA	8	0.71	1.5	13.16
CPSFA	8	0.82	0.99	14.69
VBCPSFA	6	0.94	0.8	21.05

4.4.2 Industrial case study

Steam-assisted gravity drainage is a process where steam is pumped into the sub-surface oil reservoir through a steam injection well to reduce the viscosity of the bitumen. The low-viscous bitumen and the condensed steam emulsion then flows downwards due to gravity and are subsequently pumped to the surface through the production well. Further, the emulsion is treated to produce boiler feed water (BFW) for new steam generation. Although several water treatment methods are being used in series, the boiler feed water still contains impurities, such as oil, grease, silica and sulphates, responsible for fouling in steam generators. The fouling material is physically removed by a process called pigging, during which the unit must shutdown. A temporary shutdown is associated with multiple problems that oil companies are trying to avoid. Therefore, developing a model that can extract the fouling pattern is extremely important to plan the pigging events.

The development of monitoring techniques [151] is challenging since fouling depends on the equipment type, feed temperature and impurities. A popular method is to monitor the pressure difference between the inlet and outlet. However, it may not be reliable because of noise. Further, several other factors may affect the pressure difference other than fouling. Different fouling mechanisms [152, 153] were developed depending upon the assumptions on fouling deposit and removal rate. The saw-tooth trend is a fouling mechanism that assumes an overall increasing trend with a periodic decrease due to the shedding of fouling deposits after reaching a threshold amount. Therefore, the start of high-amplitude oscillations marks the threshold phase. The increasing trend can be attributed to the fouling amount since the fouling material builds up over a period of time. This section presents an industrial application that leverages the proposed methodology's ability to separate oscillations from non-stationary behaviour.

Table 4.3: OTSG process variables

BFW Temperature (C)	Steam temperature (C)
BFW Pressure (Kpa)	Steam pressure (Kpa)
BFW flowrate $(\frac{Kg}{hr}))$	Field differential pressure (Kpa)
Oil and grease (ppm)	Fuel gas flow rate $(\frac{Kg}{hr})$

This study considers eight variables (listed in Table 4.3) from a once-through steam generator [154] for fouling buildup monitoring. Two pigging events were performed in May 2015 and August 2016, respectively. Hence, the process monitoring methods are expected to raise continuous alarms before those two events. We calculate False Alarm Rates (FAR) to compare the ability of different models in fouling monitoring. FAR is the fraction of the number of alarms raised during the first 90% operating time between two consecutive pigging events. A total of 4060 data points, spanning four consecutive years, are used for the current study. The pressure difference evolution between the inlet and the outlet is shown in the Fig. 4.4. We observe that the plot is quite noisy.

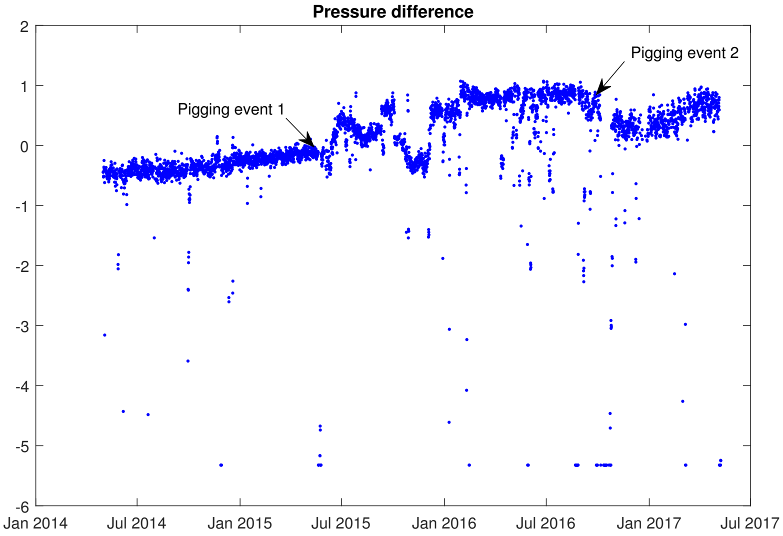


Figure 4.4: Pressure difference (For the proprietary reason, all data have been normalized)

Two statistics [131,155], namely Hotelling T^2 and squared prediction error (SPE), are constructed to detect the operating condition changes, as shown below.

$$T_k^2 = s_k^T \Sigma_s^{-1} s_k; \quad \text{SPE}_k = y_k^{rT} \Sigma_{y^r}^{-1} y_k^r \tag{4.30}$$

where $y_k = Cs_k + y_k^r$, C is the emission matrix, s_k is the latent vector and y_k^r is the residual vector. The latent and the residual vector covariance matrices are denoted by Σ_s and Σ_{y^r}, respectively. The proposed formulation is applied and compared with other extensive process monitoring techniques like NSPSFA and CPSFA.

The T^2 and SPE statistics constructed with the help of NSPSFA, CPSFA and VBCPSFA features are shown in Fig. 4.5, Fig. 4.6 and Fig. 4.7, respectively. The statistics follow a chi-square distribution to set the control limits. Table 4.4 indicates the performance comparison between different methods based on false alarm rates. Although NSPSFA-SPE statistic indicates that most of the observations match the NSPSFA model output, NSPSFA is prone to false alarms due to its inability to extract oscillatory patterns. The CPSFA-T^2 statistic produced relatively fewer false alarms. Still, the model is ineffective as it cannot separate drift-type non-stationary patterns from oscillatory behaviour. Finally, the VBCPSFA-SPE statistic demonstrates that the proposed model accommodates most of the observations. Further, the VBCPSFA-T^2 statistic continuously exceeds the 95% confidence limit around three regions, i.e., April 2015, July 2016 and February 2017. Since the VBCPSFA-T^2 statistic is constructed with the help of oscillating features, the three regions can be attributed to high-amplitude oscillations. The extracted drift-type non-stationary feature is shown in the bottom subplot of the Fig. 4.7. We observe two sudden drops in its evolution, supported by two performed cleaning events in May 2015 and August 2016. Thus, the proposed methodology can correctly extract fouling patterns and help schedule the cleaning events. A block-diagonal state-transition matrix is a more general representation to which other structures may be converted through a similarity transformation. The extracted model using the proposed methodology possesses this structure automatically. Nevertheless, some mismatch may still exist since any feature extraction is an approximate representation of the actual process to capture the main dynamics of the process. We observe that the efficiency of the proposed model is better than the other existing models due to its ability to separate oscillatory patterns from non-stationary variables.

Table 4.4: False Alarm Rates

Method	Pigging Event 1	Pigging Event 2
NSPSFA	0.41	0.53
CPSFA	0.46	0.59
VBCPSFA	0.10	0.05

4.5 Conclusion

This chapter develops a VB approach to extract slow oscillating and non-stationary hidden features. A random walk with drift-type property is utilized to model the non-stationary behaviour. Further, the measurement noise of each variable is independently modelled to handle dissimilarities in the uncertainty. An efficient algorithm is derived to estimate the distributions of underlying parameters using VBI. The primary advantage is that the proposed algorithm fuses the prior information with the observed data and thus naturally accounts for the uncertainty in the parameters. Two case studies are demonstrated to verify the proposed algorithm's efficacy in soft-sensor and fouling monitoring applications. The proposed algorithm can be extended to handle processes with varying operating conditions and non-linearities by assuming locally linear models. Further, heavy tail distributions like Student's t or Laplace can be considered for the measurement noise to construct robust models for outliers.

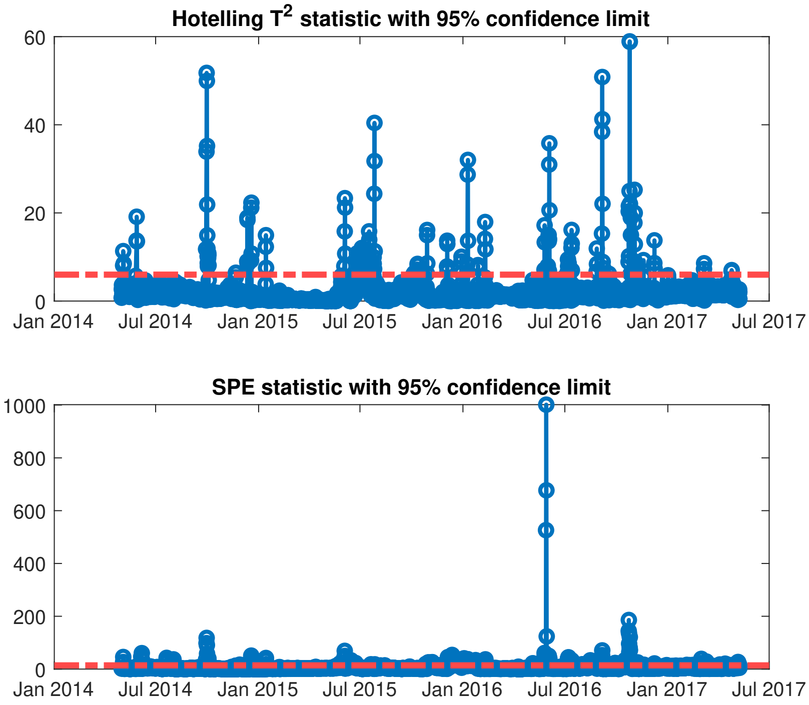


Figure 4.5: T^2 and SPE statistics using NSPSFA features

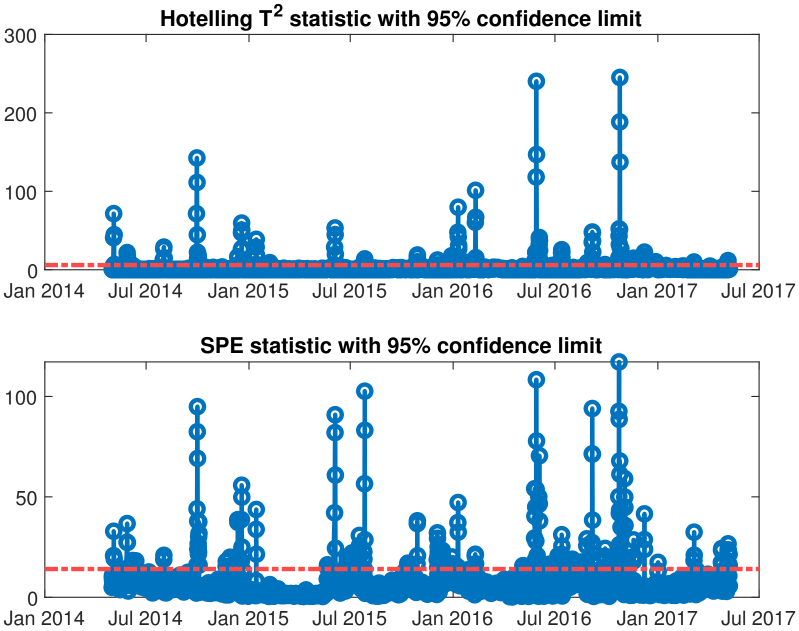


Figure 4.6: T^2 and SPE statistics using CPSFA features

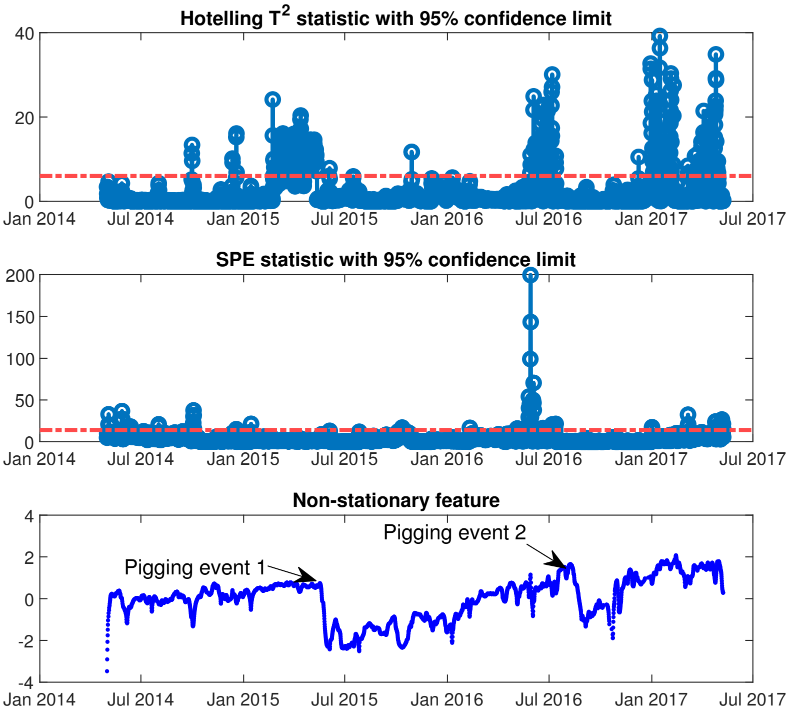


Figure 4.7: T^2 and SPE statistics using VBCPSFA features

Chapter 5

Nonlinear Slow Feature Analysis for Oscillating Characteristics under Deep Encoder-Decoder Framework [*]

Slow feature analysis aims to linearly transform measured data into uncorrelated signals that vary from slow to fast. While earlier extensions successfully extracted slow features from nonlinear sequential data, they lacked a modeling preference for non-stationary and oscillating features due to constraints on the prior distribution. To address this limitation, a semi-supervised encoder-decoder architecture is proposed in this chapter, integrating a statistical preference for such characteristics. This regularization is achieved by introducing a first-order autoregressive Gaussian prior within a regular variational auto-encoder framework, as opposed to the standard Gaussian distribution. The evidence lower bound associated with the proposed model is derived within the variational Bayesian inference framework, and the model parameters are estimated iteratively. The effectiveness of the proposed approach is evaluated on both simulated and real industrial processes.

[*] This chapter has been submitted as: **V. K. Puli** and B. Huang, "Nonlinear Slow Feature Analysis for Oscillating Characteristics under Deep Encoder-Decoder Framework," in IEEE Transactions on Industrial Informatics, 2023

5.1 Introduction

Process industries usually involve quality-related/target variables that are often difficult to measure using sensors due to physical, monetary and safety constraints. Therefore, data-based models are constructed to predict the target variables based on easy-to-measure/input variables [137,144,156,157]. The prediction performance is affected by the high dimensionality, noise and spatiotemporal correlations of the raw data. To address this, dimensionality reduction techniques are commonly applied as a pre-processing step to uncover meaningful patterns, known as features, from the measured variables.

The literature offers a range of extraction techniques tailored to the specific characteristics of the extracted features. Among these techniques are dynamic inner principal component analysis [123], dynamic inner canonical correlation analysis [124], and slow feature analysis (SFA) [36], which aim to extract intrinsic dynamic properties in a reduced dimensional subspace. SFA, in particular, focuses on extracting slowly varying patterns from the time-series data. However, it lacks a proper representation of the underlying dynamics. To address this limitation, probabilistic SFA [54,55,158] has been proposed. This approach employs a first-order autoregressive model to capture feature evolution and models the noise using probability distributions, providing better handling of outliers [56]. Additionally, various extensions, both deterministic [97], and probabilistic [98], have been proposed to extract quality-relevant slow features. More recently, complex probabilistic slow feature analysis [129] has been introduced to explicitly model oscillatory patterns in the feature space. Furthermore, a variational Bayesian approach [49] has been presented to separate drift-type non-stationary behavior and slow oscillating features.

Encoder-decoder networks [159–163] have achieved superior performance in learning representation from nonlinear data. These frameworks consist of two main components: an encoder that processes the input data and encodes it into a latent vector representation and a decoder that takes this latent vector and generates the desired output. Kingma et al. [164, 165] introduced a regularization approach by constraining the posterior distribution towards a pre-chosen prior distribution, typically a standard Gaussian. This regularization ensures that the latent variables' distribution aligns with the prior knowledge, allowing for more meaningful representations. However, this traditional regularization approach is limited to static latent variables, where the data dynamics are not considered explicitly.

Several interesting extensions [166–168] have been proposed to approximate the posterior of the dynamic latent variables using recurrent neural networks. Jiang *et al.* [169], in particular, considered the probabilistic slow feature analysis model [55] as a prior distribution for the latent variables, thus facilitating the modelling preference. Although the proposed methodology can extract slow features from nonlinear sequential data, it has several shortcomings. The generative model may not adequately separate oscillating slow features from drift-type non-stationary data due to the restricted structure of the prior distribution. Furthermore, the authors assumed that the approximate posterior distribution of the latent variable at the current time step $\boldsymbol{s}_k$ depends only on its previous time step $\boldsymbol{s}_{k-1}$ and the current observation $\boldsymbol{y}_k$, which is a major deviation from the exact posterior's variable dependency structure (details are provided in section 5.3.2). Given the assumption in [169] that $\boldsymbol{s}_k$ depends solely on $\boldsymbol{s}_{k-1}$ and $\boldsymbol{y}_k$, utilizing a gated recurrent unit (GRU) to model the mean of the posterior distribution may not offer a significant advantage, as GRUs are more advantageous for handling long-term dependencies. Finally, the proposed framework only considers a generative model for the target variable containing missing values without any inference model for their imputation, while in this work, we consider the imputation of missing target variables. We propose a novel learning algorithm called oscillating slow feature inference network (OSFIN) to deal with the aforementioned issues. The contributions of the chapter are summarized below:

- A novel learning algorithm is proposed, where the posterior distribution is influenced by a statistical preference for separating slow oscillating characteristics from drift-type non-stationary data.
- Parameter sharing is introduced to constrain the magnitude of complex eigenvalues within the unit circle.
- The structure of the posterior, given the target and input variables, is derived, and the inference model is designed to have a noncausal form, resulting in a smoothing effect.

- An additional inference model is proposed to impute missing target variables. The evidence lower bound corresponding to the proposed model is derived within the framework of variational Bayesian inference.

The remainder of this chapter is organized as follows: In Section 5.2, several probabilistic slow feature extensions from the literature are introduced. Section 5.3 presents the proposed architecture, which includes the data generating model 5.3.1, the inference structure 5.3.2, and the derivation of the evidence lower bound 5.3.3. The efficiency of the proposed learning algorithm is evaluated in Section 5.4 using both a numerical and an industrial dataset obtained from the residue hydro-conversion industrial facility. Finally, the conclusions are summarized in Section 5.5.

5.2 Background

Probabilistic slow feature analysis (PSFA) [55, 158] is a powerful framework used for learning slow and meaningful representations from high-dimensional data. Unlike conventional Slow Feature Analysis [36], which focuses on deterministic transformations, PSFA introduces a probabilistic approach to capture uncertainty in the learned representations. By incorporating probabilistic models, PSFA can effectively handle complex and noisy data, making it well-suited for real-world applications. In PSFA, the time-series dataset is represented as a sequence of observations: $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N\}$, where $\boldsymbol{y}_k$ denotes the observation at time k, and N is the total number of time steps. The learning process aims to derive a set of latent variables $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_N\}$ that capture the slow and meaningful underlying dynamics of the data. The PSFA model can be summarized using (5.1) - (5.2).

$$p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) = \mathcal{N}\left(\boldsymbol{s}_k; A\boldsymbol{s}_{k-1}, Q\right) \tag{5.1}$$

$$p(\boldsymbol{y}_k|\boldsymbol{s}_k) = \mathcal{N}\left(\boldsymbol{x}_k; \mathcal{V}\boldsymbol{s}_k, \Gamma^{-1}\right) \tag{5.2}$$

where $A \in \mathbb{R}^{m\times m}$ and $Q \in \mathbb{R}^{m\times m}$ denote the feature-transition matrix and the feature-noise covariance matrix, respectively. The Gaussian distribution is represented by $\mathcal{N}$, and m indicates the latent-space dimension. The prior distribution of the slow features $p(\boldsymbol{s}_k)$ is assumed to be standard Gaussian for all k, which facilitates the imposition of zero-mean and unit-variance constraints. This assumption leads to the constraint presented in (5.3) due to the feature-transition equation.

$$I = AA^T + Q \tag{5.3}$$

Several extensions are summarized below depending upon the characteristics of A and measurement noise $\boldsymbol{v}_k$.

- PSFA assumes A and Q to be diagonal to obtain uncorrelated slow features. Therefore, equation (5.3) boils down to (5.4), where a_i and q_i are the ith diagonal entries of A and Q respectively.

  $$a_i^2 + q_i = 1 \tag{5.4}$$

  Finally, a_i is restricted to $\begin{pmatrix}0 & 1\end{pmatrix}$ to ensure stability and avoid switching every single sample.

- In the context of state-transition matrices, the accommodation of complex poles and extraction of oscillating patterns are limited in PSFA due to its diagonal nature. However, a solution to this limitation has been proposed by Complex PSFA [129], which introduces a block-diagonal structure for the matrix. This modification allows CPSFA to accommodate complex eigenvalues and naturally encode oscillating features, as demonstrated in (5.5).

  $$A = \left[\begin{array}{c|c|c} \begin{matrix} a_1 & b_1 \\ -b_1 & a_1 \end{matrix} & 0 & 0 \\ \hline 0 & \ddots & 0 \\ \hline 0 & 0 & \begin{matrix} a_{\frac{m}{2}} & b_{\frac{m}{2}} \\ -b_{\frac{m}{2}} & a_{\frac{m}{2}} \end{matrix} \end{array}\right] \tag{5.5}$$

  At first glance, it may be inferred that the decorrelation constraint is violated by this structure. However, the two features produced by each block primarily differ in the phase angle while sharing the same power spectral density.

- The traditional PSFA is characterized by the restriction of a_i within the unit circle, resulting in the incorporation of solely stationary characteristics into the extracted features. To address this limitation and accommodate drift-type non-stationary behavior, the variational Bayesian complex probabilistic slow feature analysis is proposed by Puli *et al.* [49]. The proposed model, depicted in equations (5.6) to (5.7), incorporates a random walk with a drift-type mechanism to separate the corresponding non-stationary characteristics.

$$\begin{bmatrix} \boldsymbol{s}_k \\ \boldsymbol{h}_k \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I_{m_{ns}} \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_{k-1} \\ \boldsymbol{h}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\theta}_d \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_k^s \\ \boldsymbol{w}_k^{ns} \end{bmatrix}; \tag{5.6}$$

$$\begin{bmatrix} \boldsymbol{y}_k^s \\ \boldsymbol{y}_k^{ns} \end{bmatrix} = \begin{bmatrix} \mathcal{V}_1 & 0 \\ \mathcal{V}_2 & \mathcal{V}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_k \\ \boldsymbol{h}_k \end{bmatrix} + \boldsymbol{v}_k; \tag{5.7}$$

where

$$\begin{bmatrix} \boldsymbol{w}_k^s \\ \boldsymbol{w}_k^{ns} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} I_{m_s} - AA^T & 0 \\ 0 & \Delta^{-1} \end{bmatrix}\right);$$
$$\boldsymbol{v}_k \sim \mathcal{N}\left(0, \begin{bmatrix} \Gamma^{-1} & 0 \\ 0 & \Lambda^{-1} \end{bmatrix}\right);$$

where $\boldsymbol{s}_k \in \mathbb{R}^{m_s \times 1}$, $\boldsymbol{h}_k \in \mathbb{R}^{m_{ns} \times 1}$, $\boldsymbol{y}_k^s \in \mathbb{R}^{p_s \times 1}$ and $\boldsymbol{y}_k^{ns} \in \mathbb{R}^{p_{ns} \times 1}$ denote the oscillating slow feature, drift-type non-stationary feature, stationary, and non-stationary observed variable, respectively. Further, $A \in \mathbb{R}^{m_s \times m_s}$, $\boldsymbol{\theta}_d \in \mathbb{R}^{m_{ns} \times 1}$, $\{\mathcal{V}_1 \in \mathbb{R}^{p_s \times m_s}$, $\mathcal{V}_2 \in \mathbb{R}^{p_{ns} \times m_s}$, and $\mathcal{V}_3 \in \mathbb{R}^{p_{ns} \times m_{ns}}\}$ represent block diagonal feature-transition matrix, diagonal drift matrix and block-wise emission matrices, respectively. The precision matrices $\Delta \in \mathbb{R}^{m_{ns} \times m_{ns}}$, $\Lambda \in \mathbb{R}^{p_s \times p_s}$ and $\Gamma \in \mathbb{R}^{p_{ns} \times p_{ns}}$ are defined accordingly. In contrast to other models, consideration is given to model uncertainty by treating parameters as non-random entities. Consequently, the posterior distributions are obtained using the variational inference algorithm.

- In most state-of-the-art PSFA models, a linear mapping between the observed and latent variables is assumed, which may result in insufficient representation capabilities when dealing with data from nonlinear processes. To address this limitation, a deep Bayesian extension of probabilistic slow feature analysis was proposed by Jiang *et al.* [169]. Essentially it is a variational autoencoder framework whose prior distribution over the latent space is inherited from PSFA, as defined in (5.1). The calculation of the true posterior distribution $p_\theta(s_{1:N}|y_{1:N})$ is often difficult due to the presence of an intractable normalizing constant. To address this, an inference network $q_{\phi}(s_{1:N}|y_{1:N})$ is introduced to approximate the true posterior distribution, as demonstrated in (5.8).

$$q_{\phi}(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}, \boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{s}_k; \boldsymbol{\mu}_{\phi}(\boldsymbol{s}_{k-1}, \boldsymbol{y}_k), \boldsymbol{\sigma}^2_{\phi}(\boldsymbol{s}_{k-1}, \boldsymbol{y}_k)) \tag{5.8}$$

where mean $\boldsymbol{\mu}_{\phi}$ and the standard deviation $\boldsymbol{\sigma}^2_{\phi}$ $\forall\ k \in \{1, 2, \cdots, N\}$ are modelled using a GRU and feed-forward neural networks, respectively. However, it should be noted that the use of a GRU may not be advantageous since the mean function $\boldsymbol{\mu}_{\phi}$ exhibits only short-term dependencies, i.e., $\boldsymbol{s}_{k-1}$ and $\boldsymbol{y}_k$.

5.3 Proposed Methodology

In this chapter, a novel deep network architecture is proposed in which the prior assumption over the latent space is inherited from VBCPSFA [49] to prioritize the separation of oscillating and drift-type non-stationary features from nonlinear sequential data.

5.3.1 Data Generating Model

The generative model of the OSFIN is represented by (5.9), where the input, target, and latent variables are denoted as follows: $Y = [\boldsymbol{y}_1\ \boldsymbol{y}_2\ \ldots\ \boldsymbol{y}_N], \boldsymbol{y}_k \in \mathbb{R}^p$, $T = [\boldsymbol{t}_1\ \boldsymbol{t}_2\ \ldots\ \boldsymbol{t}_N], \boldsymbol{t}_k \in \mathbb{R}^n$, and $Z = [\boldsymbol{z}_1\ \boldsymbol{z}_2\ \ldots\ \boldsymbol{z}_N], \boldsymbol{z}_k \in \mathbb{R}^m$, respectively. The target variable is challenging to measure online and is less frequently available compared to the input variables, as it requires time-consuming laboratory analysis. Consequently, the time series is divided into two parts: $[1 : N] = \{N_o, N_m\}$, representing labelled and unlabelled time stamps, respectively. The latent variable $\boldsymbol{z}_k$ is formed by combining the oscillatory slow feature ($\boldsymbol{s}_k \in \mathbb{R}^{m_s}$) and the drift-type non-stationary feature ($\boldsymbol{h}_k \in \mathbb{R}^{m_h}$).

$$p_{\theta}(\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) = \int p_{\theta}(\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}, \boldsymbol{z}_{1:N}) d\boldsymbol{z}_{1:N} \tag{5.9}$$

where

$$p_{\theta}(\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}, \boldsymbol{z}_{1:N}) = p(\boldsymbol{z}_1) \prod_{k=2}^{N} p_{\theta}(\boldsymbol{y}_k|\boldsymbol{z}_k) p_{\theta}(\boldsymbol{t}_k|\boldsymbol{z}_k) p_{\theta}(\boldsymbol{z}_k|\boldsymbol{z}_{k-1})$$

The latent space is organized by soft constraining the approximate posterior distributions returned by the encoder to a pre-chosen prior distribution [49], as shown below.

- For $k = 1$,

$$p(\boldsymbol{z}_1) = \mathcal{N}(0, I_m)$$

- For $2 \leq k \leq N$,

$$p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) = p\left(\begin{bmatrix}\boldsymbol{s}_k \\ \boldsymbol{h}_k\end{bmatrix} \middle| \begin{bmatrix}s_{k-1} \\ h_{k-1}\end{bmatrix}\right) \tag{5.10}$$

where

$$p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) = \mathcal{N}(A\boldsymbol{s}_{k-1}, \mathrm{I} - AA^T)$$
$$p(\boldsymbol{h}_k|\boldsymbol{h}_{k-1}) = \mathcal{N}(\boldsymbol{h}_{k-1} + \boldsymbol{\theta}_d, \Delta^{-1})$$

where A is assumed to follow the structure given in (5.5). The real and imaginary parts of the eigenvalues of A are combined into two vectors for brevity, as illustrated below.

$$\boldsymbol{a} = \begin{bmatrix}a_1 & a_2 & \dots & a_{\frac{m_s}{2}}\end{bmatrix}^T; \quad \boldsymbol{b} = \begin{bmatrix}b_1 & b_2 & \dots & b_{\frac{m_s}{2}}\end{bmatrix}^T$$

The condition shown in (5.11) must be satisfied for the positive definiteness of the covariance matrix $\mathrm{I} - AA^T$.

$$\mathrm{I} - AA^T > 0 \implies 1 - a_i^2 - b_i^2 > 0 \, \forall i \tag{5.11}$$

The hard constrained optimization resulting from this condition poses a challenge for iterative algorithms such as gradient descent techniques. Consequently, a re-parameterization method is utilized to reformulate the original constrained optimization problem into an unconstrained form. This involves re-parameterizing the original parameters $\{\boldsymbol{a}, \boldsymbol{b}\}$ as $\{\boldsymbol{\theta}_a, \boldsymbol{\theta}_b\}$ in a manner that the condition in (5.11) is always satisfied.

$$\boldsymbol{a} = \frac{1}{1 + \exp(-\boldsymbol{\theta}_a)}$$
$$\boldsymbol{b} = \frac{\sqrt{\exp(-2\boldsymbol{\theta}_a) + 2\exp(-\boldsymbol{\theta}_a)}}{(1 + \exp(-\boldsymbol{\theta}_a))(1 + \exp(-\boldsymbol{\theta}_b))}$$

where $-\infty < \boldsymbol{\theta}_a, \boldsymbol{\theta}_b < \infty$. The precision matrix Δ is assumed as diagonal with its diagonal vector denoted by $\boldsymbol{\delta}$. Subsequently, $\boldsymbol{\delta}$ is constrained to the set of positive real numbers using a Softplus function.

$$\boldsymbol{\delta} = \log(1 + \exp(\boldsymbol{\theta}_\delta)) \text{ where } -\infty < \boldsymbol{\theta}_\delta < \infty$$

The input and the target variables' likelihood functions are assumed to follow a Gaussian distribution, as shown below.

$$p_{\theta_y}(\boldsymbol{y}_k|\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{\mu}_{\theta_y}(\boldsymbol{z}_k), \mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_y}(\boldsymbol{z}_k)\}) \tag{5.12}$$

$$p_{\theta_t}(\boldsymbol{t}_k|\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{\mu}_{\theta_t}(\boldsymbol{z}_k), \mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_t}(\boldsymbol{z}_k)\}) \tag{5.13}$$

where $\mathcal{D}\{\cdot\}$ represents a diagonal matrix. The non-linear functions, $\boldsymbol{\mu}_{\theta_y}, \boldsymbol{\sigma}^2_{\theta_y}$ and $\boldsymbol{\mu}_{\theta_t}, \boldsymbol{\sigma}^2_{\theta_t}$, associated with input $\boldsymbol{z}_k$, are implemented by feed-forward neural networks parametrized by $\boldsymbol{\theta}_y$ and $\boldsymbol{\theta}_t$, respectively. These functions are referred to as decoder blocks within the variational auto-encoder framework. The data-generating process is described by the probabilistic graphical model depicted in Fig. 5.1, where shaded green, yellow, and cyan circles represent the input, target, and latent variables, respectively.

5.3.2 Inference Network

The posterior distribution of the latent variable can be expanded using the Bayes' rule, as shown in (5.14).

$$p_\theta(\boldsymbol{z}_{1:N}|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) = \prod_{k=1}^{N} p_\theta(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) \tag{5.14}$$

Consider four disjoint sets of nodes $\mathbb{S}_A = \{\boldsymbol{z}_{1:k-2}, \boldsymbol{y}_{1:k-2}, \boldsymbol{t}_{1:k-2}\}$, $\mathbb{S}_B = \{\boldsymbol{z}_{k-1}\}$, $\mathbb{S}_C = \{\boldsymbol{y}_{k-1}, \boldsymbol{t}_{k-1}\}$, and $\mathbb{S}_D = \{\boldsymbol{z}_k\}$. The directed acyclic graph corresponding to the defined sets is shown in Fig. 5.2.

The joint distribution of $\mathbb{S}_A, \mathbb{S}_C, \mathbb{S}_D$ given $\mathbb{S}_B$ can be written as shown below.

$$\begin{aligned} p(\mathbb{S}_A, \mathbb{S}_C, \mathbb{S}_D|\mathbb{S}_B) &= \frac{p(\mathbb{S}_A, \mathbb{S}_B, \mathbb{S}_C, \mathbb{S}_D)}{p(\mathbb{S}_B)} \\ &= \frac{p(\mathbb{S}_D|\mathbb{S}_B)p(\mathbb{S}_C|\mathbb{S}_B)p(\mathbb{S}_B|\mathbb{S}_A)p(\mathbb{S}_A)}{p(\mathbb{S}_B)} \end{aligned}$$

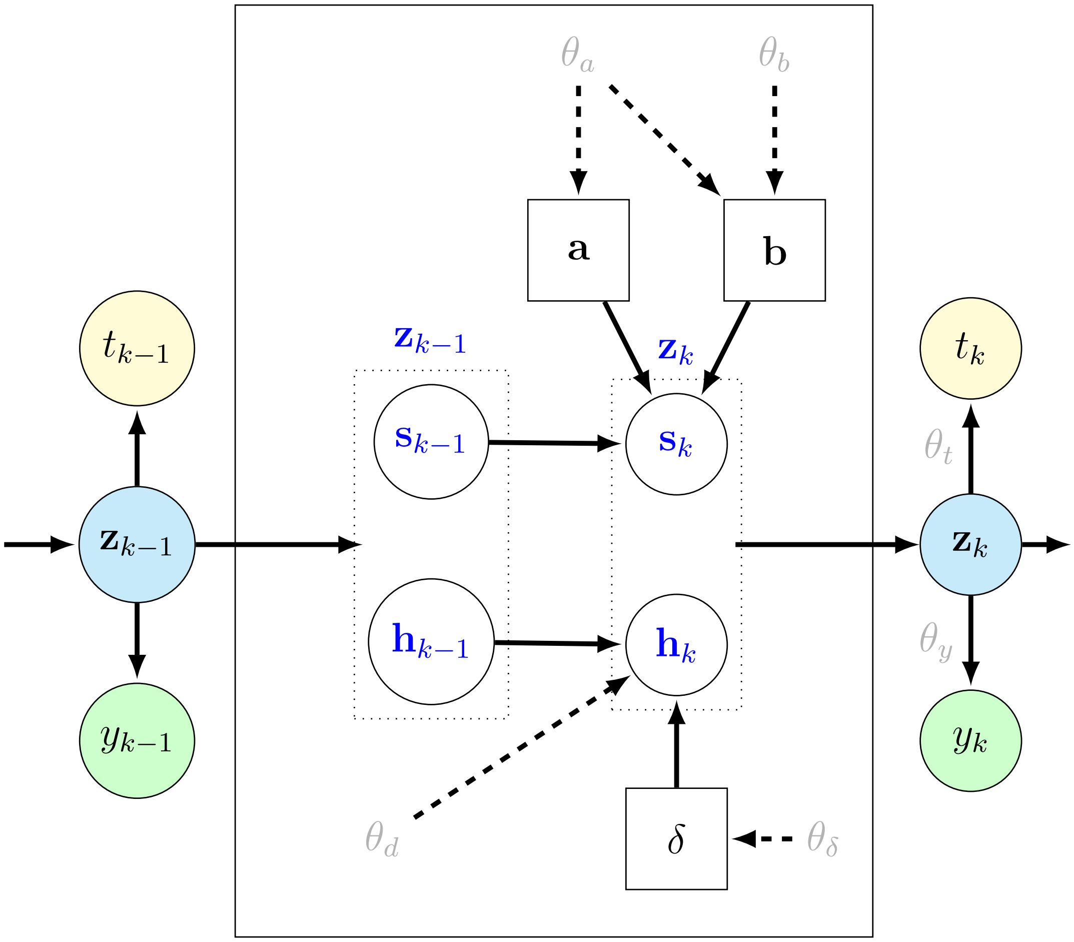


Figure 5.1: Oscillating slow feature generative model

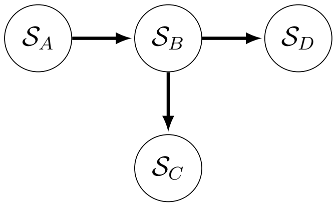


Figure 5.2: Directed acyclic graph

$$= \frac{p(\mathbb{S}_D|\mathbb{S}_B)p(\mathbb{S}_C|\mathbb{S}_B)p(\mathbb{S}_A|\mathbb{S}_B)\cancel{p(\mathbb{S}_B)}}{\cancel{p(\mathbb{S}_B)}}$$
$$= p(\mathbb{S}_D|\mathbb{S}_B)p(\mathbb{S}_C|\mathbb{S}_B)p(\mathbb{S}_A|\mathbb{S}_B)$$

The equality of the joint distribution with the product of individual distributions given $\mathbb{S}_B$ establishes the conditional independence of nodes $\mathbb{S}_A, \mathbb{S}_C$, and $\mathbb{S}_D$. As a result, the structure of the posterior distribution in (5.14) simplifies to (5.15).

$$p_\theta(\boldsymbol{z}_{1:N}|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) = \prod_{k=1}^{N} p_\theta(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}) \tag{5.15}$$

The previous latent variable $\boldsymbol{z}_{k-1}$ and the future input $\boldsymbol{y}_{k:N}$ and target $\boldsymbol{t}_{k:N}$ variables solely determine the posterior distribution of $\boldsymbol{z}_k$. Despite its simplification, the exact posterior distribution remains intractable, necessitating an approximation through an inference network $q_{\phi_z}(\cdot)$.

$$q_{\phi_z}(\boldsymbol{z}_{1:N}|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) = \prod_{k=1}^{N} q_{\phi_z}(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N})$$

The approximate posterior at time k is assumed to be Gaussian distribution, as shown below.

$$q_{\phi_z}(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}) = \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}), \mathcal{D}\{\boldsymbol{\sigma}^2_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N})\}) \tag{5.16}$$

where $\boldsymbol{\mu}_{\phi_z}$ and $\boldsymbol{\sigma}^2_{\phi_z}$ are non-linear functions parameterized by feed-forward neural networks with parameters $\boldsymbol{\phi}_z$ and inputs $\{\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}\}$. In this work, unlike the existing deep Bayesian model for PSFA [169] discussed in (5.8), the simplified functional form that solely considers $\boldsymbol{z}_{k-1}$ and $\boldsymbol{y}_k$ is not assumed. Instead, an inference model is considered with a similar structure of variable dependencies in the exact posterior [167], as illustrated in (5.16). Further, a backward gated recurrent unit (BGRU) [167] is utilized for its need to incorporate the summary of all future input and target variables at each time step, as shown in (5.17)-(5.20) [170].

$$\boldsymbol{q}_k = \sigma(W_q\boldsymbol{y}_k + U_q\boldsymbol{c}_{k+1}) \tag{5.17}$$

$$\boldsymbol{r}_k = \sigma(W_r\boldsymbol{y}_k + U_r\boldsymbol{c}_{k+1}) \tag{5.18}$$

$$\tilde{\boldsymbol{c}}_k = \tanh(W_c\boldsymbol{y}_k + U_c(\boldsymbol{r}_k \odot \boldsymbol{c}_{k+1})) \tag{5.19}$$

$$\boldsymbol{c}_k = (1 - \boldsymbol{q}_k) \odot \boldsymbol{c}_{k+1} + \boldsymbol{q}_k \odot \tilde{\boldsymbol{c}}_k \tag{5.20}$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ refer to the sigmoid and the hyperbolic tangent functions, respectively. The set of adaptable parameters, denoted by $\phi_c = \{W_q, U_q, W_r, U_r, W_c, U_c\}$, represents the first BGRU. Another BGRU, representing the second hidden variable $\boldsymbol{d}_k$ with parameters ϕ_d, is utilized to summarize future target variables information. To handle unavailable future target information, the masking method is applied to exclude corresponding missing time steps during implementation. The mean, $\boldsymbol{\mu}_{\phi_z}$, and variance, $\boldsymbol{\sigma}^2_{\phi_z}$, of the latent variable, $\boldsymbol{z}_k$, are determined by combining the two hidden variables, $\{\boldsymbol{c}_k, \boldsymbol{d}_k\}$, with the previously inferred latent variable, $\boldsymbol{z}_{k-1}$. The two hidden variables and the inferred latent variables are denoted by shaded red diamonds and cyan circles, respectively. Finally, the approximate posterior distribution of the latent variable $\forall\ k \in [1\ \ N]$ is summarized below.

$$\begin{aligned}
q_{\phi_z}(\boldsymbol{z}_1|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}) &= \mathcal{N}(\boldsymbol{z}_1; \boldsymbol{\mu}_{\phi_z}(\boldsymbol{c}_1, \boldsymbol{d}_1), \mathcal{D}\{\boldsymbol{\sigma}^2_{\phi_z}(\boldsymbol{c}_1, \boldsymbol{d}_1)\}) \\
q_{\phi_z}(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{c}_k, \boldsymbol{d}_k) &= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{\mu}_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{c}_k, \boldsymbol{d}_k), \mathcal{D}\{\boldsymbol{\sigma}^2_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{c}_k, \boldsymbol{d}_k)\})
\end{aligned}$$

5.3.3 Variational Lower Bound Maximization

The formulation of the objective function to estimate the optimal parameters is presented in this subsection. A dynamic semi-supervised method is proposed, which follows two different cases based on the static equivalent [171]. In the first case, when the target variable is observed with the input, only z is considered the latent variable (Fig. 5.4). In the second case, when the target variable is missing, both the target variable t and z are treated as latent variables (Fig. 5.5). The subsequent discussion covers the posterior inference in both cases. (**Note:** Observed variables are denoted by green nodes and latent variables by blue nodes.)

5.3.3.1 {y,t} are observed, and z is unobserved

The objective is to minimize the KL divergence between the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})$ and the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})$, as shown in (5.21).

$$\begin{aligned}
\mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})||p_\theta(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})) &= \int q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t}) \log\left(\frac{q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})}{p_\theta(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})}\right)\, d\boldsymbol{z} \qquad (5.21) \\
&= \int q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t}) \log\left(\frac{q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})}{p_\theta(\boldsymbol{z})}\right)\, d\boldsymbol{z} + \int q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t}) \log\left(\frac{p_\theta(\boldsymbol{y}, \boldsymbol{t})}{p_\theta(\boldsymbol{y}, \boldsymbol{t}|\boldsymbol{z})}\right)\, d\boldsymbol{z}
\end{aligned}$$

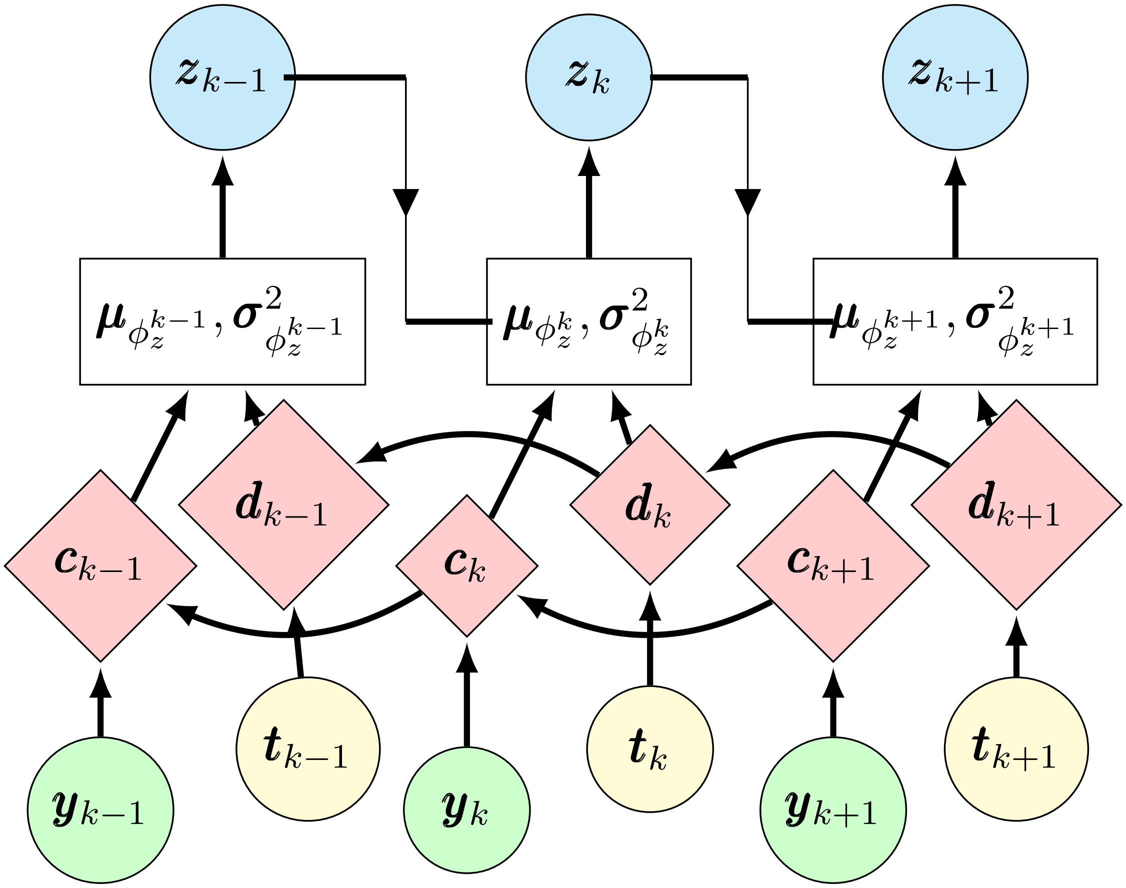

z_{k-1}
z_k
z_{k+1}
$\mu_{\phi_z^{k-1}}, \sigma^2_{\phi_z^{k-1}}$
$\mu_{\phi_z^k}, \sigma^2_{\phi_z^k}$
$\mu_{\phi_z^{k+1}}, \sigma^2_{\phi_z^{k+1}}$
d_{k-1}
d_k
d_{k+1}
c_{k-1}
c_k
c_{k+1}
t_{k-1}
t_k
t_{k+1}
y_{k-1}
y_k
y_{k+1}

Figure 5.3: Inference network

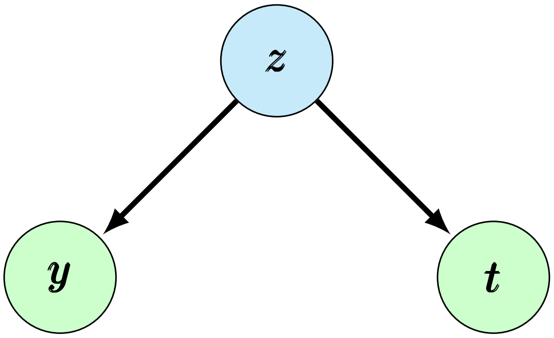


Figure 5.4: Model for the first case

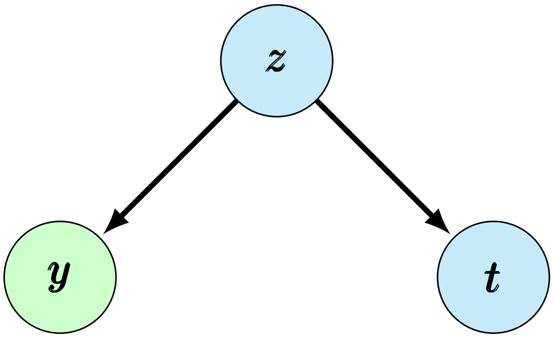


Figure 5.5: Model for the second case

$$= \log p_\theta(\boldsymbol{y}, \boldsymbol{t}) - L_{(\theta,\phi)}(\boldsymbol{y}, \boldsymbol{t})$$

where $p_\theta(\boldsymbol{y}, \boldsymbol{t})$ denotes the model evidence when $\{\boldsymbol{y}, \boldsymbol{t}\}$ are observed, and $L_{(\theta,\phi)}(\boldsymbol{y}, \boldsymbol{t})$ represents the evidence lower bound. The non-negativity property of the KL divergence ensures that $L_{(\theta,\phi)}(\boldsymbol{y}, \boldsymbol{t}) \leq \log p_\theta(\boldsymbol{y}, \boldsymbol{t})$. As the true posterior distribution is unknown, it is difficult to minimize the original objective function (5.21). Instead, the simplified lower bound $L_{(\theta,\phi)}(\boldsymbol{y}, \boldsymbol{t})$ in (5.22) is maximized.

$$L_{(\theta,\phi)}(\boldsymbol{y}, \boldsymbol{t}) \;=\; \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{t})}\left[\log p_\theta\left(\boldsymbol{y}|\boldsymbol{z}\right) + \log p_\theta\left(\boldsymbol{t}|\boldsymbol{z}\right)\right] \;-\; \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{t})||p_\theta(\boldsymbol{z})\right) \quad (5.22)$$

Substituting the previously defined distributions into (5.22), we obtain a directly implementable final lower bound. The first term in (5.22) can be approximated as

$$\mathbb{E}_{\boldsymbol{z}_k\sim q_\phi(\boldsymbol{z}_k|\boldsymbol{y}_k,\boldsymbol{t}_k)}[\log p_{\theta_y}(\boldsymbol{y}_k|\boldsymbol{z}_k)] \approx \frac{1}{R}\sum_{r=1}^{R}\log\mathcal{N}(\boldsymbol{y}_k; \boldsymbol{\mu}_{\theta_y}(\boldsymbol{z}_k^r), \mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_y}(\boldsymbol{z}_k^r)\}) \quad (5.23)$$

where the samples $\boldsymbol{z}_k^r \sim q_\phi(\boldsymbol{z}_k|\boldsymbol{y}_k, \boldsymbol{t}_k)\ \forall 1 \leq r \leq R$, and R refers to the number of Monte-Carlo samples. Similarly, the second term in (5.22) can be obtained as shown in (5.24).

$$\mathbb{E}_{\boldsymbol{z}_k\sim q_\phi(\boldsymbol{z}_k|\boldsymbol{y}_k,\boldsymbol{t}_k)}[\log p_{\theta_t}(\boldsymbol{t}_k|\boldsymbol{z}_k)] \approx \frac{1}{R}\sum_{r=1}^{R}\log\mathcal{N}(\boldsymbol{t}_k; \boldsymbol{\mu}_{\theta_t}(\boldsymbol{z}_k^r), \mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_t}(\boldsymbol{z}_k^r)\}) \quad (5.24)$$

Finally, the third term can be simplified as shown in (5.25).

$$\begin{aligned}&\mathrm{KL}(q_\phi(\boldsymbol{z}_{1:N}|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N})||p_\theta(\boldsymbol{z}_{1:N}))\\ &= \mathrm{KL}(q_\phi(\boldsymbol{z}_1|\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N})||p_\theta(\boldsymbol{z}_1)) + \sum_{k\in N_o}\mathrm{KL}(q_\phi(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N})||p_\theta(\boldsymbol{z}_k|\boldsymbol{z}_{k-1})) \quad (5.25)\end{aligned}$$

The KL divergence between the approximate posterior distribution and the prior distribution in (5.25) provides a modelling preference towards the extraction of drift-type non-stationary and oscillating slow features. Further, the KL divergence between two normal distributions can be simplified using (5.26).

$$\begin{aligned}\mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)||\mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2) = \frac{1}{2}\log\Bigg(\frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}\ \left[\Sigma_2^{-1}\Sigma_1\right] \\ + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\Sigma_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - m\Bigg)\end{aligned} \tag{5.26}$$

where Tr[·] refers to the trace of the matrix. Finally, the simplified form of the objective function in (5.22) is obtained by substituting the terms (5.23)-(5.25), as shown below.

$$-L_{(\theta,\phi)}(\boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}) = f_k(\theta) + g_{k:N}(\theta, \phi) \tag{5.27}$$

where

$$\begin{aligned}f_k(\theta) = \frac{1}{2R}\sum_{r=1}^{R}\Bigg[&\log|2\pi\mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_y}(\boldsymbol{z}^r_k)\}| + \log|2\pi\mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_t}(\boldsymbol{z}^r_k)\}| \\ &+ (\boldsymbol{y}_k - \boldsymbol{\mu}_{\theta_y}(\boldsymbol{z}^r_k))^T\mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_y}(\boldsymbol{z}^r_k)\}^{-1}(\boldsymbol{y}_k - \boldsymbol{\mu}_{\theta_y}(\boldsymbol{z}^r_k)) \\ &+ (\boldsymbol{t}_k - \boldsymbol{\mu}_{\theta_t}(\boldsymbol{z}^r_k))^T\mathcal{D}\{\boldsymbol{\sigma}^2_{\theta_t}(\boldsymbol{z}^r_k)\}^{-1}(\boldsymbol{t}_k - \boldsymbol{\mu}_{\theta_t}(\boldsymbol{z}^r_k))\Bigg]\end{aligned}$$

$$\begin{aligned}g_{k:N}(\theta, \phi) = \frac{1}{2}\Bigg[&\log\left(\frac{|I - AA^T|}{|\mathcal{D}\{\boldsymbol{\sigma}^2_{s_k}\}|}\right) + Tr\left[(I - AA^T)^{-1}\mathcal{D}\{\boldsymbol{\sigma}^2_{s_k}\}\right] \\ &+ (\boldsymbol{\mu}_{s_k} - A\boldsymbol{s}_{k-1})^T(I - AA^T)^{-1}(\boldsymbol{\mu}_{s_k} - A\boldsymbol{s}_{k-1}) - m_s \\ &- \log\left(|\Delta||\mathcal{D}\{\boldsymbol{\sigma}^2_{h_k}\}|\right) + Tr\left[\Delta\mathcal{D}\{\boldsymbol{\sigma}^2_{h_k}\}\right] \\ &+ (\boldsymbol{\mu}_{h_k} - \boldsymbol{h}_{k-1} - \boldsymbol{\theta_d})^T\Delta(\boldsymbol{\mu}_{h_k} - \boldsymbol{h}_{k-1} - \boldsymbol{\theta_d}) - m_{ns}\Bigg]\end{aligned}$$

such that

$$\begin{aligned}&\boldsymbol{\mu}_{z_1} = \boldsymbol{\mu}_{\phi_z}(\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N});\ \boldsymbol{\sigma}^2_{z_1} = \boldsymbol{\sigma}^2_{\phi_z}(\boldsymbol{y}_{1:N}, \boldsymbol{t}_{1:N}); \\ &\boldsymbol{\mu}_{z_k} = \boldsymbol{\mu}_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}) = \begin{bmatrix}\boldsymbol{\mu}_{s_k} \\ \boldsymbol{\mu}_{h_k}\end{bmatrix}; \\ &\boldsymbol{\sigma}^2_{z_k} = \boldsymbol{\sigma}^2_{\phi_z}(\boldsymbol{z}_{k-1}, \boldsymbol{y}_{k:N}, \boldsymbol{t}_{k:N}) = \begin{bmatrix}\boldsymbol{\sigma}^2_{s_k} \\ \boldsymbol{\sigma}^2_{h_k}\end{bmatrix};\end{aligned}$$

5.3.3.2 y is observed, and {t,z} are unobserved

The missing target variable is treated as a latent variable and inferred through the regressor network $q_\phi(\boldsymbol{t}|\boldsymbol{y})$. The KL divergence between the approximate posterior $q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})$ and the true posterior $p_\theta(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})$ is given by (5.28).

$$\begin{aligned}
&\mathrm{KL}(q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})||p_\theta(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})) \\
&\quad = \int q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y}) \log\left(\frac{q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})}{p_\theta(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})}\right) d\boldsymbol{t}\, d\boldsymbol{z} \qquad (5.28) \\
&\quad = \int q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y}) \log\left(\frac{q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})}{p_\theta(\boldsymbol{t},\boldsymbol{z})}\right) d\boldsymbol{t}\, d\boldsymbol{z} + \int q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y}) \log\left(\frac{p_\theta(\boldsymbol{y})}{p_\theta(\boldsymbol{y}|\boldsymbol{t},\boldsymbol{z})}\right) d\boldsymbol{t}\, d\boldsymbol{z} \\
&\quad = \log p_\theta(\boldsymbol{y}) - M_{(\theta,\phi)}(\boldsymbol{y})
\end{aligned}$$

where $\log p_\theta(\boldsymbol{y})$ denotes the log model evidence when the target variable is not observed, and its lower bound $M_{(\theta,\phi)}(\boldsymbol{y})$ can be expanded as follows.

$$\begin{aligned}
M_{(\theta,\phi)}(\boldsymbol{y}) &= \mathbb{E}_{(\boldsymbol{t},\boldsymbol{z})\sim q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})}\left[\log p_\theta\left(\boldsymbol{y}|\boldsymbol{t},\boldsymbol{z}\right)\right] - KL\left(q_\phi(\boldsymbol{t},\boldsymbol{z}|\boldsymbol{y})||p_\theta(\boldsymbol{t},\boldsymbol{z})\right) \\
&= \mathbb{E}_{\boldsymbol{t}\sim q_\phi(\boldsymbol{t}|\boldsymbol{y})}\left[L_{(\theta,\phi)}(\boldsymbol{y},\boldsymbol{t}) - \log q_\phi(\boldsymbol{t}|\boldsymbol{y})\right] \qquad (5.29)
\end{aligned}$$

Therefore, the overall objective function to maximize for the entire dataset is shown in (5.30).

$$F(\theta,\phi) = \sum_{k\in N_o} L_{(\theta,\phi)}(\boldsymbol{y}_{k:N},\boldsymbol{t}_{k:N}) + \sum_{k\in N_m} M_{(\theta,\phi)}(\boldsymbol{y}_{k:N}) + \alpha\sum_{k\in N_o}\log q_\phi(\boldsymbol{t}_k|\boldsymbol{y}_k) \qquad (5.30)$$

It should be noted that the regressor term is introduced with hyperparameter α to enable the distribution $q_{\phi_r}(\boldsymbol{t}_k|\boldsymbol{y}_k)$ to learn from the labelled dataset [171, 172].

$$q_{\phi_r}(\boldsymbol{t}_k|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{z}_k;\boldsymbol{\mu}_{\phi_r}(\boldsymbol{y}_k),\mathcal{D}\{\boldsymbol{\sigma}^2_{\phi_r}(\boldsymbol{y}_k)\}) \qquad (5.31)$$

The expression for $M_{\theta,\phi}(\boldsymbol{y}_k)$ is straightforward as it is a function of $L_{(\theta,\phi)}(\boldsymbol{y}_k,\boldsymbol{t}_k)$ and the regression network $q_{\phi_r}(\boldsymbol{t}_k|\boldsymbol{y}_k)$. The set of the inference network parameters is denoted by $\boldsymbol{\phi} = \{\boldsymbol{\phi}_r,\boldsymbol{\phi}_c,\boldsymbol{\phi}_d,\boldsymbol{\phi}_z\}$. The parameters of the generative model are represented by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_a,\boldsymbol{\theta}_b,\boldsymbol{\theta}_d,\boldsymbol{\theta}_\delta,\boldsymbol{\theta}_y,\boldsymbol{\theta}_t\}$. The optimal parameter estimates $\{\boldsymbol{\theta},\boldsymbol{\phi}\}$ can be obtained by minimizing the objective function defined in (5.30). The overall architecture of the proposed encoder-decoder network is depicted in Fig. 5.6. Finally, a mask metric, denoted as $\boldsymbol{e}_{1:N}$, is defined based on the availability of the target variable. The problem of vanishing gradients due to the length of the data N being typically long is encountered in recurrent neural networks, which are trained using the backpropagation-through-time algorithm. To address this, we construct a dataset $\begin{bmatrix}\boldsymbol{y}_k & \boldsymbol{y}_{k+1} & \dots & \boldsymbol{y}_{k+l-1}\end{bmatrix} \forall : 1 \leq k \leq N-l+1$ comprising overlapping time steps. Each sample in the dataset consists of a single time series with l time steps. Algorithm 5.1 provides an overview of the learning methodology.

Algorithm 5.1 OSFIN learning methodology

```
Input: $\boldsymbol{y}_{1:N}$, $\boldsymbol{t}_{1:N}$.
    Hyper-parameters: $m_s$, $m_{ns}$, $R$ and $l$
    Inference parameters: $\boldsymbol{\phi}_r, \boldsymbol{\phi}_c, \boldsymbol{\phi}_d, \boldsymbol{\phi}_z$
    Generative parameters: $\boldsymbol{\theta}_a, \boldsymbol{\theta}_b, \boldsymbol{\theta}_d, \boldsymbol{\theta}_\delta, \boldsymbol{\theta}_y, \boldsymbol{\theta}_t$
    Construct the mask metric $\boldsymbol{e}_{1:N}$
while notConverged() do
    $F(\theta, \phi) = 0$;
    for $k = 1, 2, \cdots, N-l+1$ do
        $\overline{N} = k + l - 1$;
        if $e_k$ is False then
            $F(\theta, \phi) = F(\theta, \phi) + M_{(\theta,\phi)}(\boldsymbol{y}_{k:\overline{N}})$
        else
            $F(\theta, \phi) = F(\theta, \phi) + L_{(\theta,\phi)}(\boldsymbol{y}_{k:\overline{N}}, \boldsymbol{t}_{k:\overline{N}}) + \alpha \log q_\phi(\boldsymbol{t}_k|\boldsymbol{y}_k)$
        end
    end
    loss = $-\frac{1}{N-l+1}F(\theta, \phi)$   Compute $\frac{\partial}{\partial\theta}$(loss), $\frac{\partial}{\partial\phi}$(loss)  Update $(\theta, \phi)$
end
Output: $\theta, \phi$
```

5.4 Simulation and Industrial Application

5.4.1 Simulation Case Study

In this subsection, the simulated dataset is utilized to test the proposed methodology. Two oscillating features are generated via eigenvalues $0.7457 \pm 0.6482i$ in (5.6). Additionally, the drift-parameter θ_d and variance δ^{-1} are drawn from a standard uniform distribution to create a drift-type non-stationary feature. Four observed variables and a target variable are subsequently formed by non-linearly transforming the generated features, as shown in Fig. 5.7.

$$t[1] = z[1] \tanh z[1] + z[2] + 0.01(z[1] + z[2])^2 + z[1]z[2]$$

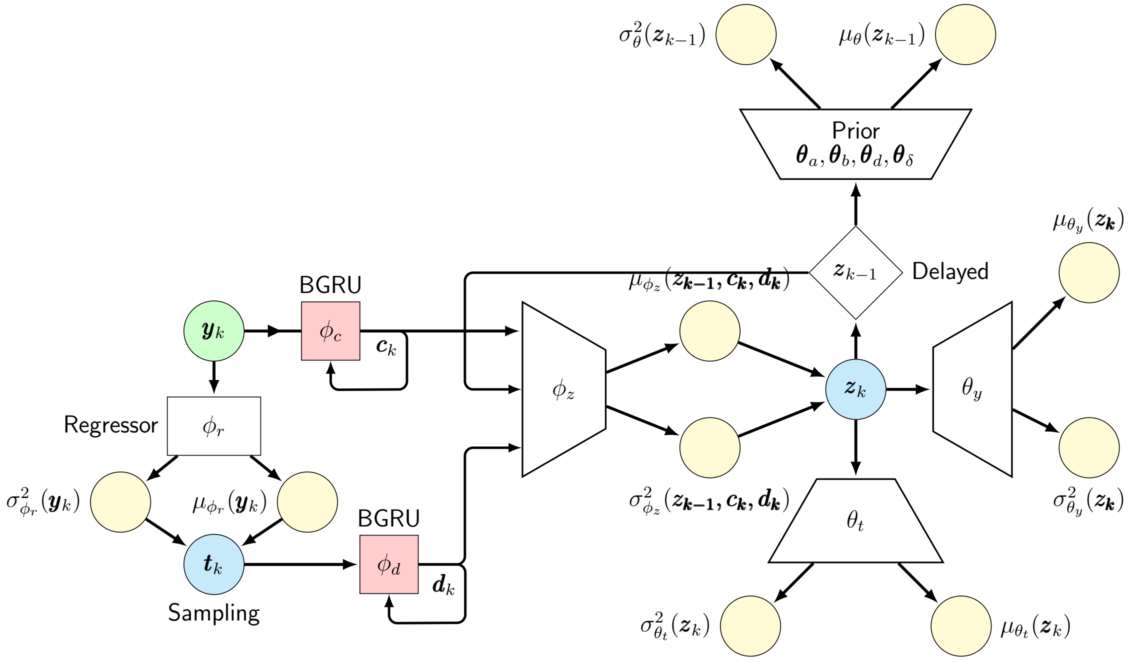

$\sigma^2_\theta(\boldsymbol{z}_{k-1})$
$\mu_\theta(\boldsymbol{z}_{k-1})$
Prior
$\boldsymbol{\theta}_a, \boldsymbol{\theta}_b, \boldsymbol{\theta}_d, \boldsymbol{\theta}_\delta$
$\mu_{\theta_y}(\boldsymbol{z_k})$
$\boldsymbol{z}_{k-1}$
Delayed
$\mu_{\phi_z}(\boldsymbol{z_{k-1}, c_k, d_k})$
BGRU
$\boldsymbol{y}_k$
ϕ_c
$\boldsymbol{c}_k$
ϕ_z
$\boldsymbol{z}_k$
θ_y
Regressor
ϕ_r
$\sigma^2_{\phi_r}(\boldsymbol{y}_k)$
$\mu_{\phi_r}(\boldsymbol{y}_k)$
BGRU
$\sigma^2_{\phi_z}(\boldsymbol{z_{k-1}, c_k, d_k})$
θ_t
$\sigma^2_{\theta_y}(\boldsymbol{z_k})$
$\boldsymbol{t}_k$
ϕ_d
$\boldsymbol{d}_k$
Sampling
$\sigma^2_{\theta_t}(\boldsymbol{z}_k)$
$\mu_{\theta_t}(\boldsymbol{z}_k)$

Figure 5.6: Overall proposed variational autoencoder architecture

$$\begin{aligned}
y[1] &= \frac{3}{(z[1]+10)^2} + 0.6\, z[2]^2 \\
y[2] &= \frac{z[2] - 2\, \tanh z[3]}{1 + \exp\{-0.4 z[1]\}} \\
y[3] &= 5\, \exp\left\{-0.1 z[1]^2 - 0.1 z[2]^2\right\} \\
y[4] &= 5\, \log\left(1 + (2z[1] - z[3])^2\right)
\end{aligned}$$

Finally, the observed variables are corrupted with additive Gaussian noise with a signal-to-noise ratio of ten. The original dataset is divided into training, validation and testing datasets with 1500, 750 and 750 samples, respectively. The network is trained using the training dataset, while the validation dataset is employed to determine the hyperparameters. The optimal latent dimensions (m_s and m_{ns}) are computed based on the validation data, where the loss function values at convergence for each pair of m_s and m_{ns} are plotted as shown in Fig. 5.8. It is observed that the training data loss function continued to decrease due to network overfitting, whereas the validation data loss provides a clearer insight into the optimal dimension pair. The minimum validation data loss occurs at ($m_s = 2, m_{ns} = 1$), indicating it as

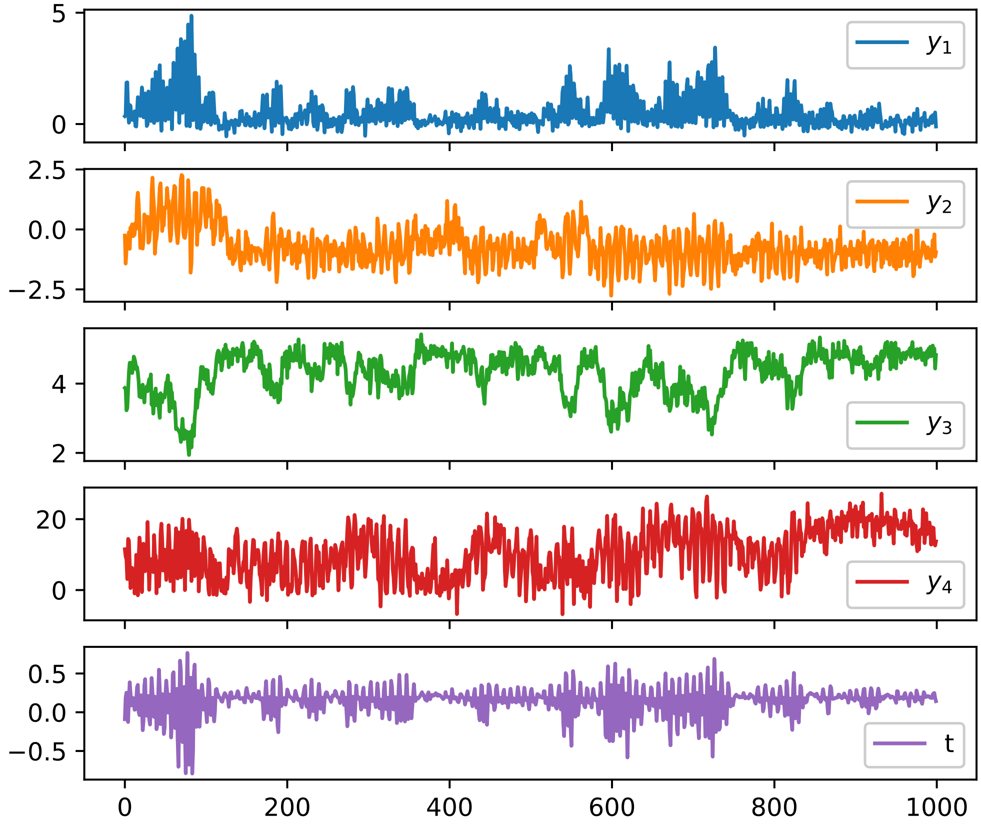


Figure 5.7: Four observed variables and one target variable. Only 1000 data points are shown to visualize the oscillatory patterns

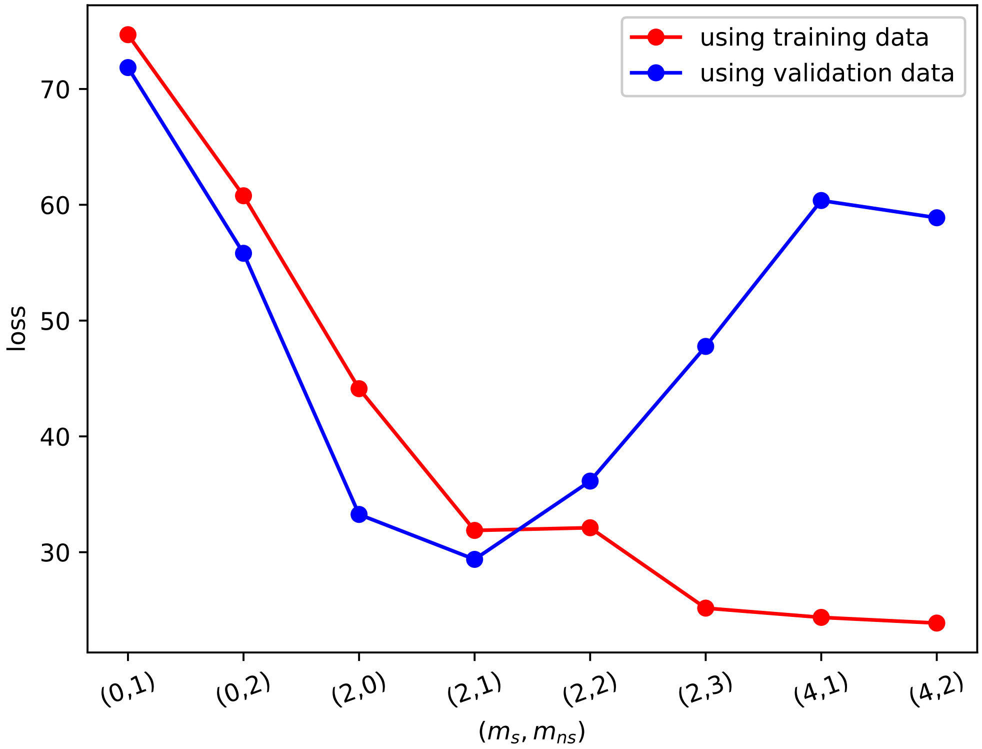


Figure 5.8: Dimension of latent variable

the optimal latent dimension pair. This process can be similarly repeated for other hyperparameters.

The performance of the proposed method on the test data is compared with the other state-of-the-art techniques, including a regular quality-relevant slow feature analysis (QRSFA) [98], GRU-based auto-encoder (GRU-AE), variational Bayesian complex PSFA (VBCPSFA) [49], and variable-wise deep Bayesian PSFA (VW-DBPSFA) [169]. Table 5.1 presents the latent variable dimension, observed variables reconstruction root mean square error (R-RMSE), target variable prediction root mean square error (P-RMSE), and the correlation between the prediction and the actual target variable (ρ) of different methods for two scenarios. We observe an increase in P-RMSE of $35.6\%, 37.7\%, 56.71\%, 22.22\%$ and 18.75% for the QRSFA, GRU-AE, VBCPSFA, VW-DBPSFA, and OSFIN models, respectively. VBCPSFA performs poorly due to its inability to model non-linear relations. Although GRU-AE and VW-DBPSFA perform better than VBCPSFA, the performance indices in Table 5.1 indicate that the proposed OSFIN model performs better due to the explicit representation of the oscillating patterns using a complex slow feature model prior.

Table 5.1: Performance comparison on simulated dataset

Method	dim(z)	R-RMSE	P-RMSE	ρ
No missing label				
QRSFA	7	0.95	0.73	0.69
GRU-AE	6	0.72	0.53	0.81
VBCPSFA	6	0.89	0.67	0.77
VW-DBPSFA	4	0.68	0.45	0.87
OSFIN	3	0.54	0.32	0.94
30% missing labels				
QRSFA	7	1.22	0.99	0.47
GRU-AE	7	0.94	0.73	0.73
VBCPSFA	8	1.14	1.05	0.51
VW-DBPSFA	7	0.87	0.59	0.79
OSFIN	5	0.63	0.35	0.88

5.4.2 Industrial Case Study

In this subsection, the efficiency of the proposed modelling algorithm is tested using a Residue Hydroconversion industrial facility dataset. The residue of the vacuum distillation unit usually consists of long-chain hydrocarbons with higher boiling points. These hydrocarbons are less flammable and must be converted into higher-value hydrocarbons. The Residue Hydroconversion process is a pivotal operation in petroleum refining for such an upgradation. The process involves three essential steps: Stabilizer, Depropanizer and Amine absorber, as shown in Fig. Fig. 5.9. In the stabilizer unit, where initial distillation occurs, volatile components are separated from the heavy feedstock. The separated vapor then advances to the depropanizer unit to further fractionate the vapor, primarily focusing on isolating propane and other heavier components from the desired product stream. The final stage of the process involves the amine absorber, where any remaining traces of undesirable components, such as hydrogen sulphide and other acidic gases, are removed through selective absorption.

Given the high temperature associated with the bottoms section and the relative rarity of level sensors capable of withstanding such extreme conditions, the prediction of the depropanizer's bottoms level emerges as a focal point in this analytical study. This prediction aids operators in preventing damage to trays and pumps while efficiently managing downstream separation processes. This investigation encompasses ten variables across 1438 samples, partitioned into training, validation, and testing sets, comprising 1100, 150, and 188 samples, respectively. All the variables are normalized for proprietary reasons. The activation function employed in all hidden layers is the hyperbolic tangent form.

The oscillating slow features (Fig. 5.10) are extracted using the proposed methodology, and the trained generative model $p_\theta(\boldsymbol{t}_k|\boldsymbol{z}_k)$ is utilized for predicting the bottoms level. The yellow dashed curve in Fig. 5.11 represents the predictive pattern on the test-dataset. It is observed that the predictive pattern closely follows the reference (black solid curve), and this observation is further reinforced by the best performance index of 0.321 P-RMSE, surpassing the performance of other state-of-the-art methods. The performance of the proposed model under multiple missing label scenarios is also analyzed, as depicted in Table 5.2. The explicit modeling preference for oscillatory

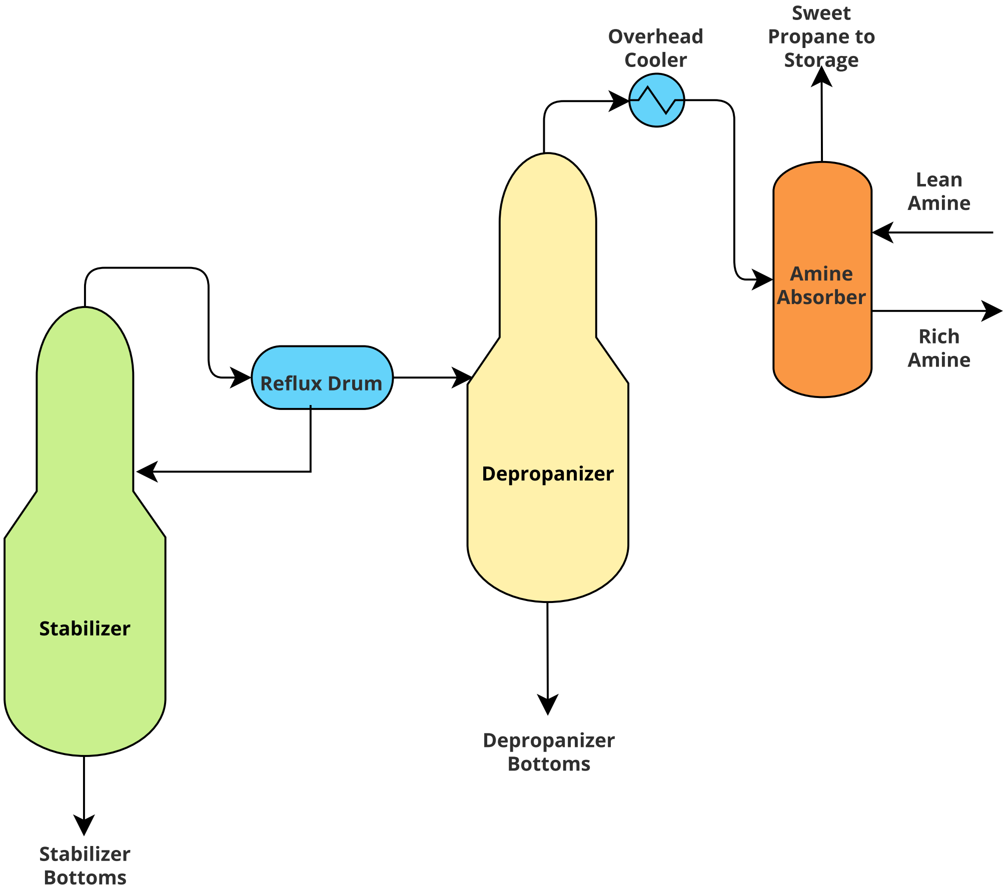


Figure 5.9: Residue Hydroconversion with Integrated Amine Absorber Unit

and drift-type non-stationary features enables the proposed model to achieve significantly improved performance, even when dealing with missing labels. For instance, under a 50% missing label scenario, the proposed model exhibits a P-RMSE of 0.371, whereas the next best model shows a P-RMSE of 0.724, representing an improvement of 95.14%. The results clearly demonstrate the efficacy of the proposed methodology in capturing oscillating slow features and utilizing them for accurate predictions of the bottoms level. Moreover, its robustness under missing label scenarios showcases the model's ability to handle real-world data with incomplete information, making it a promising approach for various practical applications.

Table 5.2: P-RMSE comparison on industrial dataset

Missing label %	GRU-NN	VBPSFA	VW-DBPSFA	OSFIN
0	0.832	0.975	0.623	0.321
10	0.851	0.901	0.691	0.345
25	0.812	0.951	0.675	0.338
40	0.887	0.997	0.715	0.355
50	0.815	1.121	0.724	0.371

5.5 Conclusion

A new deep learning algorithm with a preference for modeling non-stationary and oscillatory representations is presented in this work. Parameter sharing during back-propagation is utilized to impose a hard constraint on the magnitude of the complex eigenvalue within the unit circle. The structure of the true posterior distribution, given the target and input variables, is employed for the inference model. The latent variable is inferred based on past and future observations according to this posterior structure. Accordingly, a Gaussian regression network is introduced naturally to estimate missing target variables from observed variables. The results are validated through simulation and an industrial case study. The superior performance of the proposed model compared to state-of-the-art methods highlights its potential as a valuable tool in the field of predictive modeling and time-series analysis.

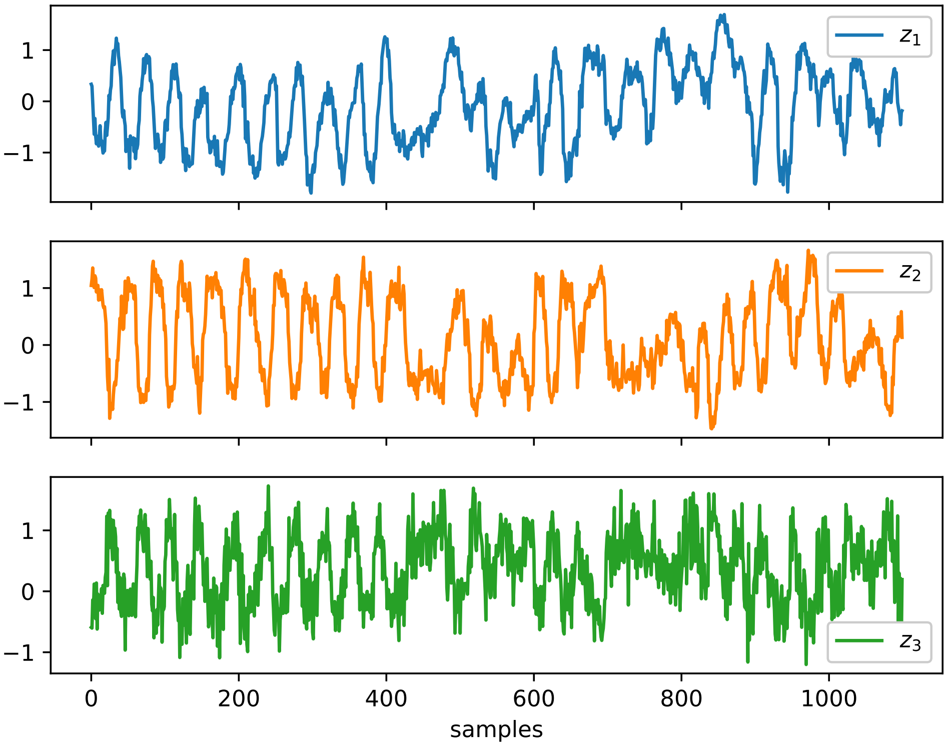


Figure 5.10: Extracted OSFIN features

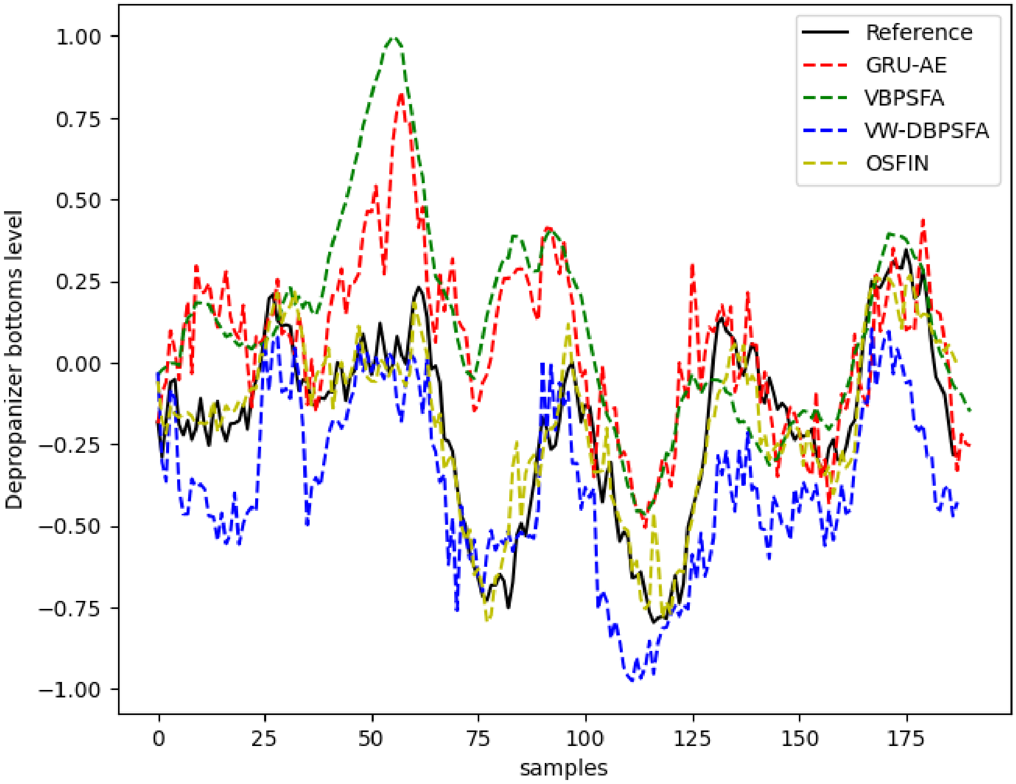


Figure 5.11: Time trend comparison

Chapter 6

Robust Complex Probabilistic Slow Feature Analysis in the Presence of Skewed Measurement Noise [*]

Complex slow feature analysis is a feature extraction technique that extracts slow oscillating patterns from the measured data. The measurement noise is usually assumed to follow a Gaussian distribution to obtain a closed-form solution. However, industrial process data is often characterized by measurement issues such as outliers, including asymmetric measurement noise. Such issues reduce the performance of the extracted features if not accounted for explicitly. Therefore, this chapter proposes a novel robust complex slow feature model to tackle the mentioned issues. In particular, this work considers a Skewed t-distribution for the measurement noise of the complex slow feature model. The parameters of the Skewed t-distribution, especially the degree of freedom and the shape parameters, account for the outliers and the asymmetric nature of the measurement noise. The parameters of the proposed model are jointly estimated using the expectation-maximization algorithm. The efficiency of the approach is demonstrated using simulated and industrial data.

6.1 Introduction

Slow feature analysis (SFA) is an unsupervised latent variable (LV) extraction method that extracts temporally correlated or slow features from a time-series dataset [173].

[*] This chapter has been published as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Robust Complex Probabilistic Slow Feature Analysis in the Presence of Skewed Measurement Noise," *IFAC-PapersOnLine*, 56(2), 10947-10952, 2023

This modelling technique is especially suited for data obtained from process systems with slowly varying dynamics. Thus, SFA has found increasing applications, such as soft sensing and fault detection, in recent times [174].

Although effective for slow processes, the deterministic nature of SFA makes it incapable of handling data complexities such as non-linearity, outliers, missing data, and asymmetric noise. Probabilistic slow feature analysis (PSFA) is the stochastic extension of the SFA that is capable of handling many such issues since the measurements and LVs are modelled as random variables [175]. The PSFA model is a linear hidden Markov model (HMM) with a unique structure for the evolution of the LVs, which ensures slowness during the feature extraction. The PSFA model thus has been used in a variety of applications such as soft sensing [55], process monitoring [176], and gross error detection [177]. The HMM formulation of PSFA also allows for more flexibility in a model to tailor it to various scenarios. Hence, various extensions of the PSFA model have been proposed in the literature, such as the robust PSFA model fan2018identification to address the issue of measurement outliers, PSFA with the additional random-walk model [49, 178] for non-stationary processes, and non-linear PSFA [179, 180].

The complex probabilistic slow feature analysis (CPSFA) [129] model is recently proposed to extract slow LVs with oscillatory patterns. Oscillations are common in many industrial datasets caused by valve stiction, aggressive control tuning, and external oscillatory disturbances [181]. Hence, models for such systems need to consider the oscillatory behaviour explicitly. Given data with a low noise, a deterministic SFA is generally suited to extract oscillatory LVs as oscillating signals exhibit temporal correlations. However, a probabilistic version of the SFA needs to be considered in the presence of significant noise. The vanilla PSFA model structure does not account for the oscillatory behaviour explicitly. Thus, the CPSFA model is more apt for such a scenario as it contains complex poles in the transition matrix that represents the dynamics of the slow features.

The performance of any machine learning model depends on how well it represents the underlying patterns and complexities of the real-world data. Industrial datasets are usually characterized by measurement outliers. Moreover, the sensory measurements of variables such as pressure and flow rate are often corrupted by skewed noise [182]. The basic version of PSFA (and CPSFA) assumes a Gaussian measurement noise, which fails to account for these complexities because of the thin tails and symmetric nature of the Gaussian density function. This inaccurate description of the measurement noise makes the estimated parameters unreliable and manifests outliers with asymmetry in the extracted features. Such a manifestation is not desirable because the extracted features, rather than measured variables, are often used to predict the target variable that is usually free of complexities. Therefore, this work proposes a novel approach to address the case where oscillatory datasets have the measurement issues mentioned earlier. In this approach, a Skewed t-distribution, which has both asymmetry and fat-tails, is used to represent the noise in the measured data. However, estimating the distribution of features with a Skewed t-noise leads to intractability. The Skewed t-distribution can be considered as a Gaussian scale mixture (GSM) distribution and thus can be represented in a hierarchical form [183,184]. This hierarchical representation allows one to obtain a closed-form analytical expression of the features and model parameters. The effectiveness of the proposed approach is demonstrated through two case studies: a simulated and an industrial case study. The main contributions of this chapter are as follows:

1. Extracting the oscillatory LVs from data corrupted by both outliers and asymmetric measurement noise under the complex PSFA framework.

2. Utilizing the hierarchical representation of the Skewed t-distribution to represent the complex measurement noise. A detailed derivation of the expressions for the oscillatory LVs and model parameters is provided through the expectation-maximization (EM) and variational Bayesian inference (VBI) framework.

The remainder of the chapter is organized as follows. Section 2 revisits the probabilistic SFA. The proposed model and the detailed derivation of the LVs and parameters are given in section 3. The results from the case studies are presented in section 4. Section 5 summarizes the obtained results.

6.2 Revisit of SFA

This section summarizes the relevant literature and highlights the limitations of existing methods.

6.2.1 Probabilistic Slow Feature Analysis (PSFA)

Shang et al. [55] proposed the probabilistic slow feature analysis to explicitly represent the dynamics (and hence slowness) in the feature space with a probabilistic description, as shown in (6.1) - (6.2).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; 0, I - AA^T) \tag{6.1}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{v}_k; 0, R) \tag{6.2}$$

where $\boldsymbol{s}_k \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{x}_k \in \mathbb{R}^{p\times 1}$ denote the slow feature and the observation, respectively. The state-transition and the emission matrices are denoted by $A \in \mathbb{R}^{m\times m}$ and $C \in \mathbb{R}^{p\times m}$, respectively. Further, the feature noise $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{v}_k$ are assumed to follow Gaussian distribution with mean 0 and covariance $(I_m - AA^T)$ and R, respectively. Here I_m represents an identity matrix of size m. Finally, m and p represent the dimension of the feature and the measurement space, respectively. The authors proposed Expectation-Maximization based algorithm to estimate the model parameters in an iterative manner.

6.2.2 Robust Probabilistic Slow Feature Analysis (RPSFA)

Fan et al. [56] proposed robust probabilistic slow feature analysis to deal with the data contaminated by outliers. The author used Student's t-distribution denoted by $\mathcal{S}t$ to describe the measurement noise as it has fat tails to accommodate for the outliers, as shown in (6.1) and (6.3).

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{S}t(\boldsymbol{v}_k; 0, R, \nu) \tag{6.3}$$

Since the feature estimation requires a Kalman filter, which is only optimal when the model is linear, and both the feature and measurement noises are described by a normal distribution, the estimation problem is therefore recast as shown in (6.4) and

(6.5).

$$\boldsymbol{v}_k|\lambda_k \sim \mathcal{N}(\boldsymbol{v}_k; 0, R/\lambda_k) \tag{6.4}$$

$$\lambda_k \sim \mathcal{G}(\lambda_k; \frac{\nu}{2}, \frac{\nu}{2}) \tag{6.5}$$

where

$$\mathcal{S}t(\boldsymbol{v}_k; 0, R, \nu) = \int_0^\infty \mathcal{N}(\boldsymbol{v}_k; 0, R/\lambda_k)\, \mathcal{G}(\lambda_k; \frac{\nu}{2}, \frac{\nu}{2})\, d\lambda_k$$

The Student's t-distribution can be viewed as an infinite mixture of Gaussian distributions with an introduction of an additional LV λ_k that follows the Gamma distribution $\mathcal{G}$. However, the authors approximate solutions by using a weighted gain in the Kalman filter equations to provide heavier weights to normal observations than to outliers. Further, Student's t-distribution is not appropriate to cope with data that has a noise skewed towards one side, as shown in [185], [184] and [183].

6.2.3 Complex Probabilistic Slow Feature Analysis (CPSFA)

The probabilistic slow feature model assumes a diagonal structure in the state transition matrix of feature space to obtain uncorrelated features. Therefore, the oscillating patterns cannot be extracted since a diagonal matrix cannot accommodate complex poles. Hence, the complex probabilistic slow feature analysis as proposed by [129] can extract slow features with oscillating patterns, as shown in the (6.6) and (6.7).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; 0, Q) \tag{6.6}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{v}_k; 0, R) \tag{6.7}$$

where

$$A = blkdiag\left\{\begin{bmatrix} a_1 & b_1 \\ -b_1 & a_1 \end{bmatrix}, \dots, \begin{bmatrix} a_{\frac{m}{2}} & b_{\frac{m}{2}} \\ -b_{\frac{m}{2}} & a_{\frac{m}{2}} \end{bmatrix}\right\};$$

$$Q = blkdiag\left\{Q_1, \dots, Q_{\frac{m}{2}}\right\}; Q_j = (1 - a_j^2 - b_j^2)I_2;$$

Essentially, the author relaxed the diagonal assumption of the state-transition matrix and assumed a block diagonal structure to accommodate complex poles. In particular, the structure shown in (6.3) was chosen to satisfy the constraint that ensures the extracted features are uncorrelated. Although CPSFA can extract slow features with oscillating patterns, it is not robust to outliers and cannot deal with asymmetric noise.

6.3 Robust CPSFA for outliers and asymmetric noise

6.3.1 Proposed methodology

Nurminen et al. [185] introduced the Skewed t-distribution $\mathcal{ST}$, which is robust to outliers and can describe skewed noise distribution, and presented an algorithm to estimate the states for a general state-space model with known parameters. In this section, we propose an extension to the CPSFA model but with unknown parameters in the Skewed t-distribution and derive an algorithm that can jointly estimate both the hidden variables and unknown parameters. The proposed formulation is shown in (6.8)-(6.9).

$$\boldsymbol{s}_k^{i:i+1} = \begin{bmatrix} a_i & b_i \\ -b_i & a_i \end{bmatrix} \boldsymbol{s}_{k-1}^{i:i+1} + \boldsymbol{w}_k^{i:i+1}; \tag{6.8}$$

$$\boldsymbol{w}_k^{i:i+1} \sim \mathcal{N}(\overrightarrow{0}, (1 - a_i^2 - b_i^2) I_2); \quad i = 2j-1; \forall 1 \leq j \leq \frac{m}{2}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \prod_{i=1}^{p} \mathcal{ST}(\boldsymbol{v}_k^i; \mu_i, R_{ii}, \Delta_{ii}, \nu_i) \tag{6.9}$$

where the location parameter of the Skewed t-distribution is represented by μ_i. Further, the diagonal entries of the scale and the shape matrix are denoted by R_{ii} and Δ_{ii}, respectively. The i^{th} entry of the degrees of freedom vector is represented by ν_i. We employ the Gaussian scale mixture representation to obtain the closed-form solution, as shown in (6.10)-(6.12).

$$\boldsymbol{v}_k | \boldsymbol{u}_k, \Lambda_k \sim \mathcal{N}(\boldsymbol{v}_k; \boldsymbol{\mu} + \Delta \boldsymbol{u}_k, {\Lambda_k}^{-1} R) \tag{6.10}$$

$$\boldsymbol{u}_k | \Lambda_k \sim \mathcal{N}_+(\boldsymbol{u}_k; \overrightarrow{0}, {\Lambda_k}^{-1}) \tag{6.11}$$

$$\Lambda_k \sim \mathcal{G}(\Lambda_k; \frac{\boldsymbol{\nu}}{2}, \frac{\boldsymbol{\nu}}{2}) = \prod_{i=1}^{p} \mathcal{G}(\lambda_k^{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}) \tag{6.12}$$

where

$$\mathcal{ST}(\boldsymbol{v}_k; 0, R, \Delta, \nu)$$
$$= \int_0^\infty \int_0^\infty \mathcal{N}(\boldsymbol{v}_k; \boldsymbol{\mu} + \Delta \boldsymbol{u}_k, {\Lambda_k}^{-1} R)\, \mathcal{N}_+(\boldsymbol{u}_k; \overrightarrow{0}, {\Lambda_k}^{-1}) \mathcal{G}(\Lambda_k; \frac{\boldsymbol{\nu}}{2}, \frac{\boldsymbol{\nu}}{2})\, d\boldsymbol{u}_k\, d\Lambda_k$$

where $\mathcal{N}_+$ denotes a multivariate truncated Gaussian distribution with closed positive orthant as support. The observed variables, LVs and parameters are denoted by $X := \boldsymbol{x}_{1:N}$, $Z := \{\boldsymbol{s}_{1:N}, \boldsymbol{u}_{1:N}, \Lambda_{1:N}\}$, and $\theta := \{\boldsymbol{a}, \boldsymbol{b}, C, \boldsymbol{\mu}, R, \Delta, \boldsymbol{\nu}\}$, respectively.

6.3.2 Parameter Estimation

Given the observed variables, the parameters are estimated with the help of the Expectation-Maximization algorithm in an iterative manner. The joint log-likelihood $p(X, Z|\theta)$ of the proposed model is shown below.

$$\log p(X, Z|\theta) = \log p(\boldsymbol{s}_1) + \sum_{k=2}^{N} \log p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}; \theta) +$$
$$\sum_{k=1}^{N} \left[\log p(\boldsymbol{x}_k|\boldsymbol{s}_k, \boldsymbol{u}_k, \boldsymbol{\Lambda}_k; \theta) + \log p(\boldsymbol{u}_k|\Lambda_k; \theta) + \log p(\Lambda_k; \theta)\right];$$

where

$$\begin{aligned}
\log\, p(\boldsymbol{s}_1) &= -\frac{m}{2}\log 2\pi - \frac{1}{2}\boldsymbol{s}_1^T\boldsymbol{s}_1;\\
\log p(\mathbf{s}_k|\mathbf{s}_{k-1}; \theta) &= -\frac{m}{2}\log 2\pi - \sum_{i=1}^{\frac{m}{2}} \log|1 - a_i^2 - b_i^2|\\
&\qquad - \frac{1}{2}(\boldsymbol{s}_k - A\boldsymbol{s}_{k-1})^T(I_m - AA^T)^{-1}(\boldsymbol{s}_k - A\boldsymbol{s}_{k-1});\\
\log p(\boldsymbol{x}_k|\boldsymbol{s}_k, \boldsymbol{u}_k, \Lambda_k; \theta) &= -\frac{p}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{p}\log\left|\frac{R^{ii}}{\Lambda_k^{ii}}\right|\\
&\qquad - \frac{1}{2}\sum_{i=1}^{p}\frac{\Lambda_k^{ii}}{R^{ii}}(x_k^i - C^i\boldsymbol{s}_k - \mu_i - \Delta^i\boldsymbol{u}_k)^2;\\
\log p(\boldsymbol{u}_k|\Lambda_k) &= -\frac{p}{2}\log 2\pi + \frac{1}{2}\sum_{i=1}^{p}\left[\log\Lambda_k^{ii} - {u_k^i}^2\Lambda_k^{ii}\right];\\
\log p(\Lambda_k|\boldsymbol{\nu}) &= \sum_{i=1}^{p}\left[\frac{\nu_i}{2}\log\frac{\nu_i}{2} - \log\Gamma\left(\frac{\nu_i}{2}\right) + \left(\frac{\nu_i}{2} - 1\right)\log\Lambda_k^{ii} - \frac{\nu_i}{2}\Lambda_k^{ii}\right];
\end{aligned}$$

Here C^i and Δ^i denote the i^{th} rows of C and Δ, respectively. Further, the Q-function is defined as follows.

$$\begin{aligned}
Q(\theta|\theta^{\eta-1}) &= \mathbb{E}_{Z\sim p(Z|X;\theta^{\eta-1})}\left\{\log p(X, Z; \theta)\right\}\\
&= \langle\log p(X, Z; \theta)\rangle
\end{aligned}$$

where $\theta^{\eta-1}$ refers to the parameters from previous iteration $\eta - 1$. Finally, the update expressions for each parameter can be obtained by equating the derivative of the Q-function, with respect to each parameter, to zero.

6.3.3 Update expressions:

The update expression for a_i can be derived as shown in [129].

$$\alpha_{i_3} a_i^3 + \alpha_{i_2} a_i^2 + \alpha_{i_1} a_i + \alpha_{i_0} = 0 \tag{6.13}$$

where

$$\begin{aligned}
\alpha_{i_3} &= -2(N-1); \\
\alpha_{i_2} &= \sum_{k=2}^{N} \left\langle s_k^i s_{k-1}^i + s_k^{i+1} s_{k-1}^{i+1} \right\rangle; \\
\alpha_{i_1} &= -\alpha_{i_3}(1 - {b_i}^2) + 2b_i \sum_{k=2}^{N} \left\langle s_k^i s_{k-1}^{i+1} - s_{k-1}^i s_k^{i+1} \right\rangle \\
&\quad - \sum_{k=2}^{N} \left\langle s_k^i s_k^i + s_k^{i+1} s_k^{i+1} + s_{k-1}^i s_{k-1}^i + s_{k-1}^{i+1} s_{k-1}^{i+1} \right\rangle; \\
\alpha_{i_0} &= \alpha_{i_2}(1 - {b_i}^2);
\end{aligned}$$

Similarly, the equation for b_i is shown in (6.14).

$$\beta_{i_3} b_i^3 + \beta_{i_2} b_i^2 + \beta_{i_1} b_i + \beta_{i_0} = 0 \tag{6.14}$$

where

$$\begin{aligned}
\beta_{i_3} &= -2(N-1); \\
\beta_{i_2} &= \sum_{k=2}^{N} \left\langle s_k^i s_{k-1}^{i+1} - s_{k-1}^i s_k^{i+1} \right\rangle; \\
\beta_{i_1} &= -\beta_{i_3}(1 - {a_i}^2) + 2a_i \sum_{k=2}^{N} \left\langle s_k^i s_{k-1}^i + s_k^{i+1} s_{k-1}^{i+1} \right\rangle \\
&\quad - \sum_{k=2}^{N} \left\langle s_k^i s_k^i + s_k^{i+1} s_k^{i+1} + s_{k-1}^i s_{k-1}^i + s_{k-1}^{i+1} s_{k-1}^{i+1} \right\rangle; \\
\beta_{i_0} &= \beta_{i_2}(1 - {a_i}^2);
\end{aligned}$$

The estimates a_i and b_i, $\{i = 2j - 1; \forall 1 \leq j \leq \frac{m}{2}\}$, for the current iteration can be obtained by solving the cubic equations shown in (6.13) and (6.14). Similarly by setting the partial derivatives of $Q-$function to zero with respect to C, μ, R^{ii}, Δ^{ii} and ν_i $\forall 1 \leq i \leq p$, the update expressions can be obtained as shown in (6.15) - (6.19).

$$C = \left[\sum_{k=1}^{N} (\boldsymbol{x}_k - \boldsymbol{\mu} - \Delta \langle \boldsymbol{u}_k \rangle) \left\langle \boldsymbol{s}_k^T \right\rangle \right] \left[\sum_{k=1}^{N} \left\langle \boldsymbol{s}_k \boldsymbol{s}_k^T \right\rangle \right]^{-1} \tag{6.15}$$

$$\mu_i = \frac{\sum_{k=1}^{N} \langle \Lambda_k^{ii}(x_k^i - C^i \boldsymbol{s}_k - \Delta^i \boldsymbol{u}_k) \rangle}{\sum_{k=1}^{N} \langle \Lambda_k^{ii} \rangle} \tag{6.16}$$

$$R^{ii} = \frac{1}{N} \sum_{k=1}^{N} \left\langle \Lambda_k^{ii}(x_k^i - C^i \boldsymbol{s}_k - \mu_i - \Delta^i \boldsymbol{u}_k)^2 \right\rangle \tag{6.17}$$

$$\Delta^{ii} = \frac{\sum_{k=1}^{N} \langle \Lambda_k^{ii}(x_k^i - C^i \boldsymbol{s}_k - \mu_i) u_k^i \rangle}{\sum_{k=1}^{N} \left\langle \Lambda_k^{ii} {u_k^i}^2 \right\rangle} \tag{6.18}$$

$$\log \frac{\nu_i}{2} - \psi\left(\frac{\nu_i}{2}\right) + \frac{1}{N} \sum_{k=1}^{N} \left[\langle \log \Lambda_k^{ii} \rangle - \langle \Lambda_k^{ii} \rangle \right] + 1 = 0 \tag{6.19}$$

where ψ is a digamma function. The update expressions in (6.13) - (6.18) involve terms that require the expectations of coupled terms, such as $\boldsymbol{s}_k, \boldsymbol{u}_k$ and Λ_k, with respect to the joint posterior $p(\boldsymbol{s}_k, \boldsymbol{u}_k, \Lambda_k | \boldsymbol{x}_{1:N}; \theta)$, which is not analytically tractable. Therefore, we use the variational Bayesian inference algorithm to approximate the joint posterior.

$$p(\boldsymbol{s}_k, \boldsymbol{u}_k, \Lambda_k | \boldsymbol{x}_{1:N}; \theta) \approx q(\boldsymbol{s}_k | \boldsymbol{x}_{1:N}; \theta) q(\boldsymbol{u}_k | \boldsymbol{x}_{1:N}; \theta) q(\Lambda_k | \boldsymbol{x}_{1:N}; \theta)$$

Since conjugate priors are chosen for the LVs, as shown in (6.10)-(6.12), the posterior distributions belong to the same family as shown in (6.20) - (6.22).

$$q(\boldsymbol{s}_k | \boldsymbol{x}_{1:N}; \theta) = \mathcal{N}(\boldsymbol{s}_k; \boldsymbol{s}_{k|N}, P_{k|N}) \tag{6.20}$$

where

$$\begin{aligned}
P_{k|k-1} &= A P_{k-1|k-1} A^T + I_m - A A^T \\
K_x &= P_{k|k-1} C^T (C P_{k|k-1} C^T + \langle \Lambda_k \rangle^{-1} R)^{-1} \\
\boldsymbol{s}_{k|k} &= A \boldsymbol{s}_{k-1|k-1} + K_x (\boldsymbol{x}_k - C A \boldsymbol{s}_{k-1|k-1} - \boldsymbol{\mu} - \Delta \langle \boldsymbol{u}_k \rangle) \\
P_{k|k} &= (I_m - K_x C) P_{k|k-1}
\end{aligned}$$

The smoothing is applied using the following equations

$$J_k = P_{k|k} A^T P_{k+1|k}^{-1}$$

$$\begin{aligned}\boldsymbol{s}_{k|N} &= \boldsymbol{s}_{k|k} + J_k(\boldsymbol{s}_{k+1|N} - A\boldsymbol{s}_{k|k}) \\ P_{k|N} &= P_{k|k} + J_k(P_{k+1|N} - P_{k+1|k})J_k^T\end{aligned}$$

$$q(\boldsymbol{u}_k|\boldsymbol{x}_{1:k};\theta) = \mathcal{N}_+(\boldsymbol{u}_k;\boldsymbol{u}_{k|N}, U_{k|N}) \tag{6.21}$$

where

$$\begin{aligned}\epsilon_k &= \boldsymbol{x}_k - C\boldsymbol{s}_{k|N} - \mu \\ K_u &= \Delta(\Delta^T\Delta + R)^{-1} \\ \boldsymbol{u}_{k|N} &= K_u\epsilon_k \\ U_{k|N} &= (I_p - K_u\Delta)\langle\Lambda_k\rangle^{-1}\end{aligned}$$

The following expressions are computed $\forall\, 1 \leq i \leq p$ to estimate the mean and covariance of $\boldsymbol{u}_k$.

$$\begin{aligned}\chi_k^i &= \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{{u_{k|N}^i}^2}{2U_{k|N}^{ii}}\right\}; \\ \omega_k^i &= \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{-u_{k|N}^i}{\sqrt{2U_{k|N}^{ii}}}\right)\right); \\ \gamma_k^i &= u_{k|N}^i + \sqrt{U_{k|N}^{ii}}\left(\frac{\chi_k^i}{1-\omega_k^i}\right); \\ \Sigma_k^{ii} &= U_{k|N}^{ii}\left[1 - \left(\frac{u_{k|N}^i}{\sqrt{U_{k|N}^{ii}}}\right)\left(\frac{\chi_k^i}{1-\omega_k^i}\right) - \left(\frac{\chi_k^i}{1-\omega_k^i}\right)^2\right] + {\gamma_k^i}^2;\end{aligned}$$

$$q(\Lambda_k^{ii}|\boldsymbol{x}_{1:N};\theta) = \mathcal{G}\left(\Lambda_k^{ii}; \frac{\nu_i}{2}+1, \frac{\nu_i + \phi_k^{ii}}{2}\right) \tag{6.22}$$

where

$$\phi_k = R^{-1}(\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^T + CP_{k|N}C^T) + (\Delta R^{-1}\Delta + I)\langle\boldsymbol{u}_k\boldsymbol{u}_k^T\rangle - R^{-1}\Delta\langle\boldsymbol{u}_k\rangle\boldsymbol{\epsilon}_k^T - \Delta R^{-1}\boldsymbol{\epsilon}_k\langle\boldsymbol{u}_k\rangle^T$$

Finally, the expectations that are required to compute the parameter update expressions are shown below.

$$\langle\boldsymbol{s}_k\rangle = \boldsymbol{s}_{k|N};\ \left\langle\boldsymbol{s}_k\boldsymbol{s}_{k-1}^T\right\rangle = P_{k|N}J_{k-1}^T + \boldsymbol{s}_{k|N}\boldsymbol{s}_{k-1|N}^T;$$

$$
\begin{aligned}
\left\langle \boldsymbol{s}_k\boldsymbol{s}_k^T \right\rangle &= P_{k|k} + \boldsymbol{s}_{k|N}\boldsymbol{s}_{k|N}^T;\ \langle \boldsymbol{u}_k \rangle = \boldsymbol{\gamma}_k;\ \left\langle \boldsymbol{u}_k\boldsymbol{u}_k^T \right\rangle = \Sigma_k;\\
\left\langle \Lambda_k^{ii} \right\rangle &= \frac{\nu_i + 2}{\nu_i + \phi_k^{ii}};\\
\left\langle \log \Lambda_k^{ii} \right\rangle &= \psi\left(\frac{\nu_i}{2} + 1\right) - \log\left(\frac{\nu_i + \phi_k^{ii}}{2}\right)
\end{aligned}
$$

In the proposed methodology, explicit update equations are presented for each of the involved parameters. So the computational load is similar to other EM-based probabilistic models.

6.4 Case studies

This section demonstrates the effectiveness of the proposed algorithm using a simulated and an industrial data set for soft-sensor applications. In both the case studies, the measurement noise follows a skewed t-distribution with location parameter $\mu^* = 0$, shape parameter $\Delta^* = 2$ for positive skewness, degree of freedom parameter $\nu^* = 5$ for outliers, and a scale parameter R^* such that the signal-to-noise ratio is 0.2.

6.4.1 Simulation case study

We generated four slow oscillating features using (6.8) by using the state transition matrix shown in (6.23). A target variable y is generated using these features with the help of a linear model whose coefficients are drawn from a standard Gaussian distribution and additive Gaussian noise.

$$
A^* = \begin{bmatrix} 0.49 & 0.86 & 0 & 0 \\ -0.86 & 0.49 & 0 & 0 \\ 0 & 0 & 0.68 & 0.7 \\ 0 & 0 & -0.7 & 0.68 \end{bmatrix};\ Q^* = I_4 - AA^T; \tag{6.23}
$$

We considered an emission matrix C^* drawn randomly from a standard Gaussian distribution to construct six measured variables, as shown in Fig. 6.1. The measured variables are positively skewed with outliers. A total of 2000 data samples are generated for the analysis.

The data are divided into training (1000 samples), validation (500 samples), and testing sets (500 samples) to train the model, select the hyper-parameters, and compare the performance. LVs are extracted using the proposed methodology and other state-of-the-art feature extraction methods, such as SFA, dynamic partial least

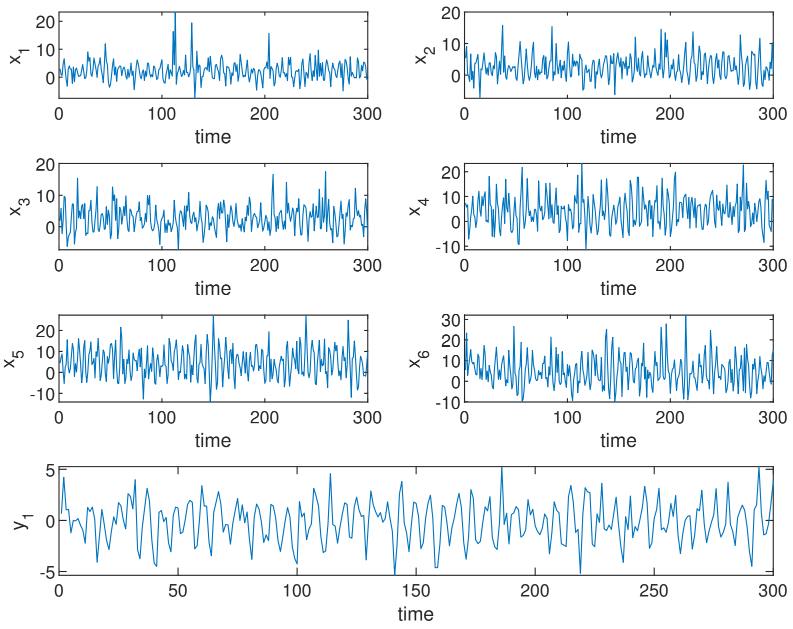


Figure 6.1: Six observed variables and one target variable (Only 300 data points are shown for better visualization)

squares (DPLS) kaspar1993dynamic, RPSFA, and CPSFA. Finally, regression models are built between the target training data and the extracted LVs from each feature extraction method. The scatter plot between the noise-free target variable and the predictions based on various feature extraction methods is shown in Fig. 6.2. It is observed that the predictions based on the RCPSFA features are closer to the 45^0 line, indicating a better prediction. Table 6.1 shows the performance indices, namely the root mean square error (RMSE) and the coefficient of determination (R^2), calculated with the help of predicted and the actual target testing dataset. It is observed that the proposed method results in the highest R^2 due to the explicit representation of the oscillations and the consideration of data complexities.

Table 6.1: Performance comparison on simulated dataset

Method	SFA	DPLS	RPSFA	CPSFA	RCPSFA
RMSE	0.94	0.88	0.75	0.71	0.56
R^2	0.38	0.43	0.55	0.60	0.77

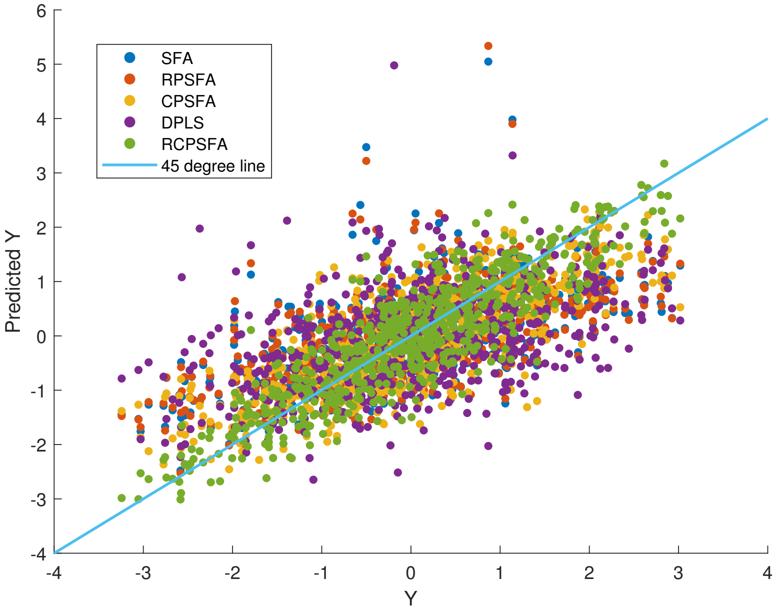


Figure 6.2: Scatter plot of the simulated case study

6.4.2 Industrial case study

The depropanizer bottoms level is essential to the design of the operating conditions in subsequent downstream cracking processes. However, a level sensor that can sustain extreme bottoms temperature involves expensive installation and maintenance costs. Therefore, we develop a soft sensor for the depropanizer bottoms level using the dataset obtained from a residue hydro-conversion unit. The eighteen measured variables are artificially corrupted with an additive Skewed t-distribution noise to demonstrate the efficacy of the proposed algorithm. The measured and the target variables are shown in their normalized form in Fig. 6.4 for proprietary reasons. The dataset is partitioned into training, validation, and testing sets with 800, 160, and 480 samples, respectively.

We performed a similar analysis as discussed in the simulation case study. The proposed algorithm is iterated until the Q-function, computed using the validation data, no longer improves. The extracted slow oscillating features using the proposed methodology are shown in Fig. 6.3. The extracted LVs are used to build a predictive model for the target variable. The predictions and the performance indices constructed using the predictions obtained from various feature extraction methods on the test dataset are shown in Fig. 6.5 and Table 6.2 respectively. It is shown that the proposed model exhibits better performance than the other slow feature extraction methods when the observed variables contain outliers with skewed noise.

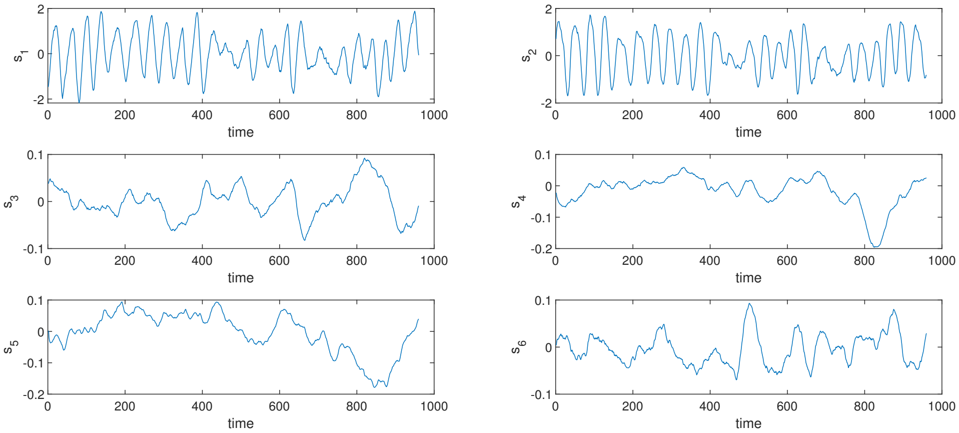


Figure 6.3: Extracted features using the RCPSFA model in the industrial case study

Table 6.2: Performance comparison on the industrial dataset

Method	SFA	DPLS	RPSFA	CPSFA	RCPSFA
RMSE	0.99	1.13	0.88	0.75	0.61
R^2	0.43	0.36	0.48	0.55	0.66

6.5 Conclusion

This chapter proposes a novel feature extraction model that extracts slow oscillating features from data that is corrupted with additive noise characterized by outliers and asymmetric distributions. Using the expectation-maximization algorithm, we have derived explicit update equations for the involved parameters, especially the skewed-t distribution parameters. The effectiveness of the proposed algorithm is verified using a simulation and an industrial case study. The future work is to extend the solution to the case where both the measured and the target variables are corrupted with Skewed t-noise.

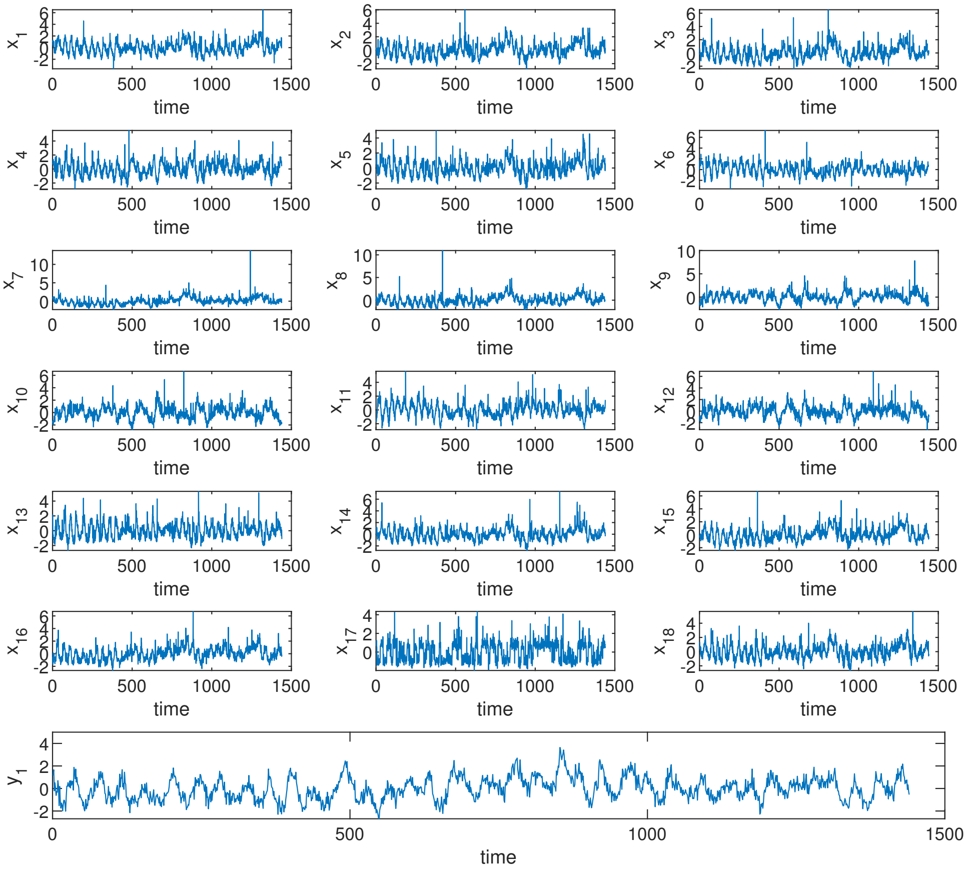


Figure 6.4: Observed and target variables

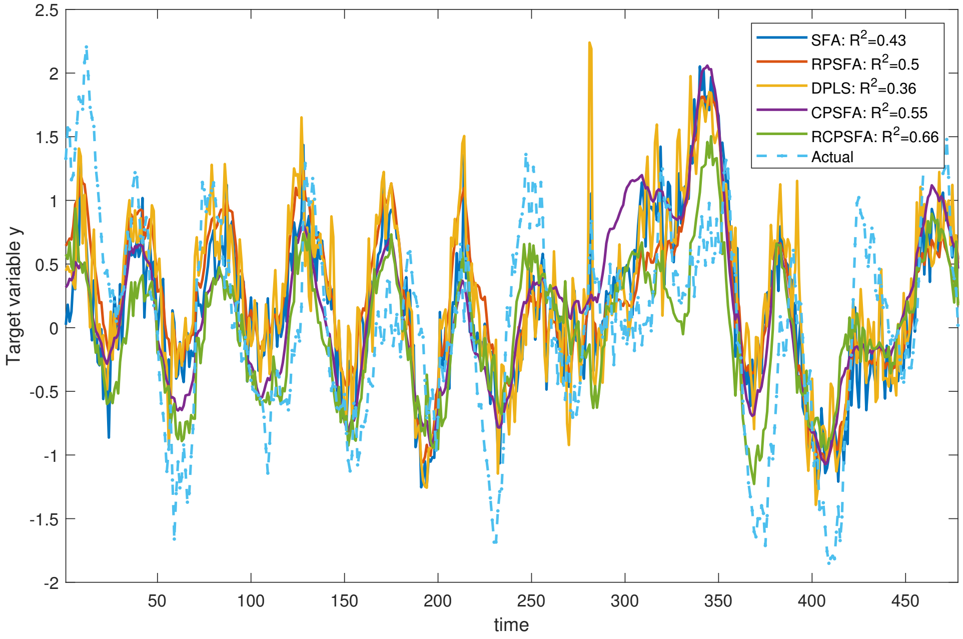


Figure 6.5: Prediction on test data in the industrial case study

Chapter 7

Sparse Robust Dynamic Feature Extraction using Bayesian Inference [*]

Data sets of large-scale industrial processes are often high-dimensional and are characterized by outliers. Probabilistic latent variable models are effective for modeling such data complexities. However, the performance of such models is influenced by the number of latent variables and the adequacy of the noise model that describes the data complexities, such as outliers and skewness. This chapter presents a probabilistic slow feature model that considers these two issues simultaneously. The latent space dimensionality is automatically obtained by modeling the emission matrix with a Laplace distribution, resulting in a sparse model. Further, the measurement noise is modeled with a skewed-t distribution to account for the outliers and asymmetry of the noise. The hierarchical representation of these two distributions is considered to obtain tractable solutions for the distributions of the latent variables and the model parameters. The resulting model is estimated through variational Bayesian inference. This chapter is a further extension of Chapter 6.

7.1 Introduction

Data-based modeling has been increasingly preferred to model complex processes for industrial applications, such as soft sensing, process monitoring and prognosis. A few

[*] This chapter has been published as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Sparse Robust Dynamic Feature Extraction using Bayesian Inference," in IEEE Transactions on Industrial Electronics, 2023

of the main challenges of industrial datasets are measurement-related issues, such as collinearity, outliers, missing data, and asymmetric noise. Probabilistic latent variable modeling is an effective tool to handle such issues [186]. This is a flexible framework that allows one to represent a variety of aspects of a dataset based on the choice of model structure and assumed distributions [187].

Often, high-dimensional datasets are generated from a very low-dimensional latent space. Further, not all latent features affect all the input variables. Therefore, extracting the correct number of latent variables corresponding to each input variable is crucial. It can be achieved by a sparse representation of the emission matrix using an L1 regularizer. A natural extension to probabilistic sparse latent models [188] was introduced by assuming a generalized hyperbolic prior to the emission matrix. As a particular case, Guan *et al.* [189] proposed the sparse probabilistic principal component analysis model by incorporating a Laplace prior. Laplace distribution, as such, does not yield tractable posterior estimates. However, the Laplace distribution can be expressed as a Gaussian scale mixture distribution, resulting in an analytical expression for the posteriors. The sparse principal component analysis model has not been well explored for industrial process applications, and the research in this field is limited [190].

Another important aspect of industrial datasets is the asymmetric nature of measurement noise along with outliers. This is a common occurrence in many fields, such as target-tracking and robotics [184]. In process industries, the measurement noise of variables in the ranges of their limiting values will be skewed (e.g., the liquid level close to zero). For such cases, an asymmetric and heavy-tailed distribution such as the skewed-t distribution is the ideal choice to model the measurement noise [185, 191]. Various skewed-t distribution-based filtering schemes have been proposed in recent years, such as the normal-skew mixture distribution-based filtering [192], skewed-t mixture distribution-based filtering [193], and robust filtering with skewed-t noise for state transition model mismatch [194]. It must be noted that using the skewed-t distribution as such also leads to intractability issues for the state estimation procedure. Hence, the hierarchical representation of the skewed-t distribution is adopted, which uses the fact that the skewed-t distribution can also be expressed as a Gaussian scale mixture distribution [184].

In this work, these two aspects are implemented in a dynamic latent variable model. In particular, the probabilistic slow feature analysis (PSFA) is selected as the latent variable model. The PSFA model is an extension of the slow feature analysis (SFA) method, which is an unsupervised learning method that projects the input data onto a latent space such that the latent variables are characterized by slow variations [173]. The main rationale behind the technique is that although the input data might have faster variations, the primary underlying phenomena are characterized by temporally slowly changing sources. This technique is thus relevant to many industrial processes that have slower dynamics and thus is increasingly being used in process industries [174, 195, 196]. The PSFA model is a linear dynamic model that explicitly characterizes the slowness of a process through a constrained state transition model [55]. The PSFA model has been demonstrated to be effective for applications, such as soft sensing [55], and process monitoring [176]. Numerous extensions to the basic PSFA model have been explored to account for various aspects of data, such as oscillatory behavior [129], measurement outliers [56], and multi-modality [197].

In PSFA, the slowest latent variables are usually selected for modeling because they are regarded to be capable of presenting the primary dynamics of the process. Their dimensionality is generally determined based on a trial-and-error procedure [49, 99]. This procedure assumes that the order of importance is strictly based on velocities, which may not be always accurate. A more efficient way is to be able to select the latent variables that generate the input data automatically. This work proposes a method to achieve it by having a sparse representation of the emission matrix. This is achieved by considering the emission matrix rows as Laplace-distributed random vectors. The method is also designed to be robust to asymmetric noise and outliers by using a skewed-t distribution to model the measurement noise. Although the issue of such a noise in the PSFA model is considered in a recent work [198], it assumes the parameters as deterministic entities. The proposed approach has model parameters modeled by random variables to account for model uncertainties and include prior knowledge. The proposed model is estimated using the variational Bayesian inference approach [77,199]. Finally, the efficiency of the proposed algorithm is evaluated using two soft-sensor case studies: a numerical and an experimental case study. The main contributions of the chapter are as follows.

1. Automatic selection of the slow features relevant to the input-output data through a sparse representation of the PSFA model.

2. Robust PSFA model in the presence of asymmetric measurement noise and outliers.

3. Estimation of the posterior distribution of the model random variables using both the input-output data for soft-sensor applications.

The remainder of the chapter is organized as follows. The background and the research gap are presented in section II. The proposed model formulation is presented in section III. The detailed derivation of the posterior distributions of the latent variables and parameters is given in section IV. Section V presents the results from the case studies and section VI outlines the conclusions.

7.2 Background and shortcomings

The PSFA model [55] was proposed to extract slowly varying dynamics from the input data with a probabilistic interpretation, as shown in (7.1) - (7.2).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; 0, I - AA^T) \tag{7.1}$$

$$\boldsymbol{z}_k = C_z\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{v}_k; 0, \gamma^{-1}\mathrm{I}_p) \tag{7.2}$$

where $\boldsymbol{s}_k \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{z}_k \in \mathbb{R}^{p\times 1}$ denote the slow feature and the input observation, respectively. Further, $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ represent the process and measurement noise, respectively. The point estimates of the parameters were determined using the Expectation-Maximization algorithm. Further, an extension to this work that estimates the posterior distribution of the parameters has been proposed [99]. Essentially, the prior information of the parameters A, C_z, and γ was introduced as shown below.

$$\begin{aligned}
A &= \mathrm{diag}\{a_1, \cdots, a_m\};\ p(A) = \prod_{i=1}^{m} \mathcal{B}eta(a_i|\alpha_a, \beta_a);\\
C_z &= \begin{bmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_p\end{bmatrix}^T;\ p(C_z) = \prod_{i=1}^{p} \mathcal{N}(\boldsymbol{c}_i|\vec{0}, \Lambda_0^{-1});\\
R &= \gamma^{-1}\mathrm{I}_p;\ p(\gamma) = \mathcal{G}(\gamma|\alpha_\gamma, \beta_\gamma);
\end{aligned}$$

A $\mathcal{B}eta$ distribution was selected for $a_i \; \forall 1 \leq i \leq m$ to pass the prior information in which the magnitude is within the unit circle. Further, the prior distributions of the parameters C_z and γ were assumed to be Gaussian $\mathcal{N}$ and Gamma $\mathcal{G}$, respectively, owing to conjugate distributional properties. Although the method proposed in [99] was successful in extracting slow features, as discussed in the introduction, it has three primary shortcomings.

- Typically, the optimal number of slow features m is not known a priori. Therefore, it is usually assumed to be the number of input variables p. However, such a practice may result in the extraction of some irrelevant features.
- A Gaussian distribution is symmetric around its mean and possesses short tails. Hence, it does not accommodate common industrial complexities like outliers and skewed noise.
- Soft sensing is one of the important applications of the extracted features. However, the proposed slow feature model [99] does not consider the output information for the estimation of the posterior distributions.

Recently, an algorithm called robust complex probabilistic slow feature analysis [198] has been proposed to deal with the second shortcoming. Essentially, a skewed t-distribution $\mathcal{ST}(\boldsymbol{v}_k|0, R, \Delta, \nu)$ was considered for the measurement noise. Here R, Δ, and ν represent the scale, shape and degree of freedom parameters, respectively. However, the algorithm assumes the parameters as non-random quantities, and hence, the prior knowledge of these parameters cannot be integrated. Therefore, in the next section, a new slow feature algorithm is proposed that can extract slowly varying patterns from the input-output data corrupted with outliers and skewed noise, and automatically select relevant features.

7.3 Mathematical Formulation

In this section, we propose a robust sparse probabilistic slow feature model that can extract slowly varying patterns using input-output data in the presence of noise with outliers and skewness, as shown in (7.3) - (7.4).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; 0, I - AA^T) \tag{7.3}$$

Table 7.1: Prior distribution information

s.no.	**Variable**	**Prior/likelihood**	**Hyper parameters**
1	A	$p(A) = \prod_{i=1}^{m} \mathcal{B}eta(a_i\|\alpha_a, \beta_a)$	α_a, β_a
2	C	$p(C\|H) = \prod_{i=1}^{p} \mathcal{N}(\boldsymbol{c}_i\|\vec{0}, \text{diag}(h_i))$	$-$
3	H	$p(H) = \prod_{i=1}^{p} \prod_{j=1}^{m} \mathcal{E}xp(h_i^j\|\lambda)$	λ
4	R	$p(R) = \prod_{i=1}^{p} \mathcal{G}^{-1}(R^{ii}\|\alpha_R, \beta_R)$	α_R, β_R
5	Δ	$p(\text{diag}(\Delta)) = \mathcal{N}(\text{diag}(\Delta)\|\mu_\Delta, \Sigma_\Delta)$	$\mu_\Delta, \Sigma_\Delta$
6	ν	$p(\nu) = \prod_{i=1}^{p} \mathcal{G}(\nu_i\|\alpha_\nu, \beta_\nu)$	α_ν, β_ν
7	$\boldsymbol{u}_k$	$p(\boldsymbol{u}_k\|\Lambda_k) = \mathcal{N}_+(\boldsymbol{u}_k; \overrightarrow{0}, {\Lambda_k}^{-1})$	$-$
8	Λ_k	$p(\Lambda_k\|\nu) = \prod_{i=1}^{p} \mathcal{G}(\Lambda_k^{ii}\|\frac{\nu_i}{2}, \frac{\nu_i}{2})$	$-$
9	$\boldsymbol{s}_1$	$p(\boldsymbol{s}_1) = \mathcal{N}(\boldsymbol{s}_1\|\vec{0}, \text{I}_m)$	$-$
10	$\boldsymbol{s}_k$	$p(\boldsymbol{s}_k\|\boldsymbol{s}_{k-1}, A) = \mathcal{N}(\boldsymbol{s}_k\|A\boldsymbol{s}_{k-1}, \text{I}_m - AA^T)$	$-$
11	$\boldsymbol{x}_k$	$p(\boldsymbol{x}_k\|\boldsymbol{s}_k, \boldsymbol{u}_k, \Lambda_k, C, R, \Delta) = \mathcal{N}(\boldsymbol{x}_k\|C\boldsymbol{s}_k + \Delta\boldsymbol{u}_k, {\Lambda_k}^{-1}R)$	$-$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k;\ \boldsymbol{v}_k \sim \prod_{i=1}^{p} \mathcal{ST}(\boldsymbol{v}_k^i; 0, R_{ii}, \Delta_{ii}, \nu_i) \tag{7.4}$$

where the augmented data $\boldsymbol{x}_k$ and the emission matrix C is defined as follows.

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{z}_k \\ \boldsymbol{y}_k \end{bmatrix}; \quad C = \begin{bmatrix} C_z \\ C_y \end{bmatrix}$$

Here $\boldsymbol{z}_k$ and $\boldsymbol{y}_k$ represent input and output vectors with their corresponding emission matrices C_z and C_y, respectively. Since the non-Gaussian representation of the measurement noise hinders the regular usage of the Kalman state estimation algorithm, we employ the hierarchical representation [183, 185, 200], as shown in (7.5) - (7.7).

$$p(\boldsymbol{v}_k|\boldsymbol{u}_k, \Lambda_k) = \mathcal{N}(\boldsymbol{v}_k; \Delta\boldsymbol{u}_k, {\Lambda_k}^{-1}R) \tag{7.5}$$

$$p(\boldsymbol{u}_k|\Lambda_k) = \mathcal{N}_+(\boldsymbol{u}_k; \overrightarrow{0}, {\Lambda_k}^{-1}) \tag{7.6}$$

$$p(\Lambda_k) = \prod_{i=1}^{p} \mathcal{G}\left(\lambda_k^{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right) \tag{7.7}$$

where

$$\mathcal{ST}(\boldsymbol{v}_k; 0, R, \Delta, \nu) = \int_0^\infty \int_0^\infty \mathcal{N}(\boldsymbol{v}_k; \Delta\boldsymbol{u}_k, {\Lambda_k}^{-1}R)\, \mathcal{N}_+(\boldsymbol{u}_k; \overrightarrow{0}, {\Lambda_k}^{-1})\mathcal{G}(\Lambda_k; \frac{\boldsymbol{\nu}}{2}, \frac{\boldsymbol{\nu}}{2})\, d\boldsymbol{u}_k\, d\Lambda_k$$

where $\mathcal{N}_+$ denotes a multivariate truncated Gaussian distribution with closed positive orthant as support.

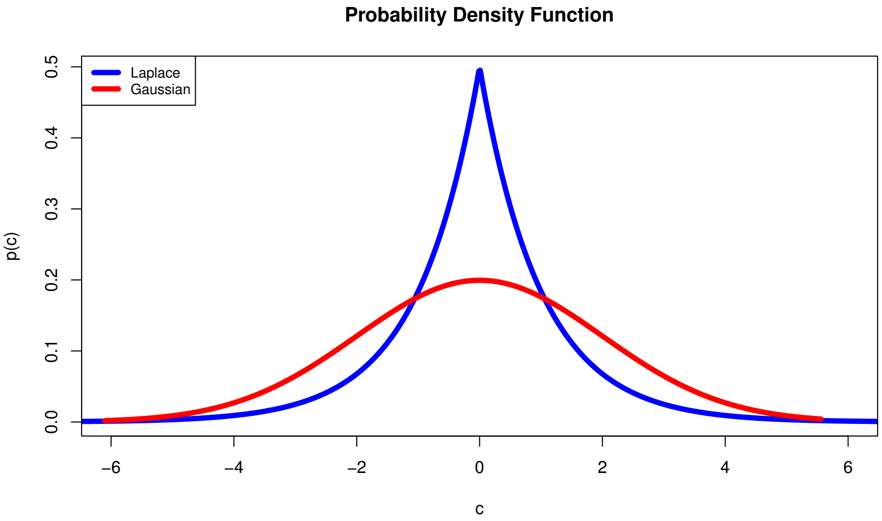


Figure 7.1: Laplace vs. Gaussian distribution

Since the number of extracted slow features is assumed to be p, it is essential to obtain a sparse emission matrix, and thus, extract only relevant features. Introducing the L_1 norm to the objective function is one of the popular ways to achieve sparsity. In a probabilistic formulation, an L_1 norm can be indirectly incorporated in the objection function by assuming a Laplace prior to each of the entries of the emission matrix [188, 189]. It is observed that the probability of obtaining a value closer to zero is very high compared to the Gaussian distribution with the same mean and the standard deviation, as shown in Fig. 7.1. Since the Laplace prior is not conjugate for the Gaussian likelihood, we again employ a hierarchical representation [201]. Essentially, the Laplace distribution can be written as a combination of a Gaussian and an Exponential distribution with the introduction of an additional variable h_i^j, as shown in (7.8) - (7.9).

$$p(\boldsymbol{c}_i|\boldsymbol{h}_i) = \mathcal{N}(\boldsymbol{c}_i; \vec{0}, \mathrm{diag}(\boldsymbol{h}_i)) \tag{7.8}$$

$$p(h_i^j) = \frac{1}{\lambda} \exp\left\{-\frac{h_i^j}{\lambda}\right\} \tag{7.9}$$

where

$$p(c_i^j|\lambda) = \int_0^\infty p(c_i^j|h_i^j)\, p(h_i^j)\, dh_i^j$$

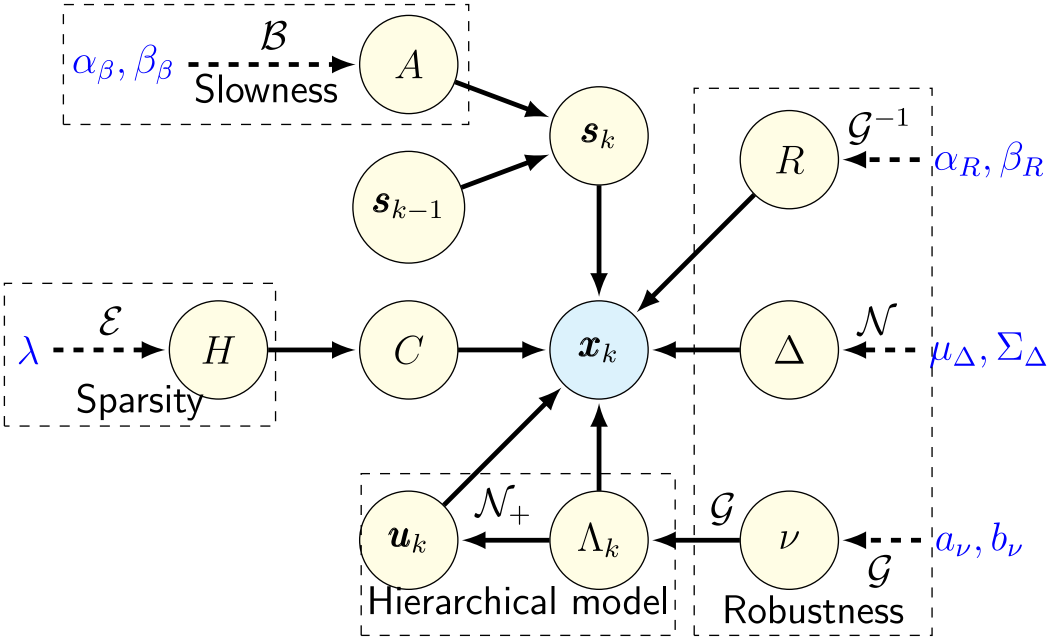


Figure 7.2: Probabilistic graphical model of Robust Sparse PSFA

Due to modelling preference, the likelihood and prior distributions of various other variables are introduced in Table 7.1. The hierarchical probabilistic graphical model corresponding to the (7.3)-(7.9) is shown in the Fig. 7.2. The hyperparameters $\theta \in \{\alpha_a, \beta_a, \lambda, \alpha_R, \beta_R, \mu_\Delta, \sigma_\Delta, \alpha_\nu, \beta_\nu\}$ and latent variables $d \in \{A, C, H, R, \Delta, \nu, \Lambda_{1:N}, \boldsymbol{u}_{1:N}, \boldsymbol{s}_{1:N}\}$ are denoted by the yellow circle, and text without the circle, respectively. Finally, the complete data likelihood is computed to obtain the posterior distributions of all the latent variables.

$$\log p(X, d|\theta) = \sum_{k=1}^{N} \log p(\boldsymbol{x}_k|C, \boldsymbol{s}_k, \Delta, \boldsymbol{u}_k, \boldsymbol{\Lambda}_k, R) + \log p(\boldsymbol{s}_1) + \sum_{k=2}^{N} \log p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}, A)$$
$$+ \sum_{k=1}^{N} \log p(\boldsymbol{u}_k|\Lambda_k) + \sum_{k=1}^{N} \log p(\Lambda_k|\nu) + \sum_{i=1}^{m} \log p(a_i) + \sum_{i=1}^{p} \log p(\boldsymbol{c}_i|\boldsymbol{h}_i)$$
$$+ \sum_{i=1}^{p}\sum_{j=1}^{m} \log p(h_i^j) + \sum_{i=1}^{p} \log p(R^{ii}) + \log p(\mathrm{diag}(\Delta)) + \sum_{i=1}^{p} \log p(\nu_i);$$

7.4 Posterior Distributions

Variational Inference can be used to obtain the expression for the approximate posterior distribution $q(d_s)$ of any latent variable $d_s \in d$ as

$$\log q(d_s) \propto \langle \log p(X, d|\theta) \rangle_{q(d_s^{'})}$$

where $d_s^{'}$ denotes its complementary set. The approximate posterior distribution of various latent variables is derived as shown below.

1. $q(\boldsymbol{c}_i) = \mathcal{N}(\boldsymbol{c}_i | \langle c_i \rangle, \Sigma_{c_i})$ where

$$\Sigma_{c_i} = \left(\langle \mathrm{diag}(\boldsymbol{h}_i)^{-1} \rangle + \sum_{k=1}^{N} \left\langle \frac{\Lambda_k^{ii}}{R^{ii}} \boldsymbol{s}_k \boldsymbol{s}_k^T \right\rangle \right)^{-1};$$

$$\langle c_i \rangle = \Sigma_{c_i}^T \sum_{k=1}^{N} \left\langle \frac{\Lambda_k^{ii}(x_k^{(i)} - \Delta^{ii} u_k^i)}{R^{ii}} \boldsymbol{s}_k \right\rangle;$$

2. The distribution of h_i^j belongs to an exponential family, as shown below.

$$q(h_i^j) \propto \frac{1}{\sqrt{h_i^j}} \exp\left\{ -\frac{(c_i^j)^2}{2h_i^j} - \frac{h_i^j}{\lambda} \right\}$$

3. $q(\mathrm{diag}(\Delta)) = \mathcal{N}(\mathrm{diag}(\Delta) | \mu_\Delta, \Sigma_\Delta)$

$$\Sigma_\Delta = \left\{ \Sigma_\Delta^{-1} + \sum_{k=1}^{N} \langle \boldsymbol{u}_k^T R^{-1} \Lambda_k \boldsymbol{u}_k \rangle \right\}^{-1};$$

$$\mu_\Delta = \Sigma_\Delta \left(\sum_{k=1}^{N} \langle \boldsymbol{u}_k^T R^{-1} \Lambda_k (\boldsymbol{x}_k - C\boldsymbol{s}_k) \rangle + \Sigma_\Delta^{-1} \mu_\Delta \right);$$

4. $q(R^{ii}) = \mathcal{G}^{-1}(R^{ii} | \alpha_{R^{ii}}, \beta_{R^{ii}})$

$$\alpha_{R^{ii}} = \alpha_R + \frac{N}{2};$$

$$\beta_{R^{ii}} = \beta_R + \frac{1}{2} \sum_{k=1}^{N} \Lambda_k^{ii} \Big((x_k^{(i)})^2 + tr\left(\langle \boldsymbol{c}_i \boldsymbol{c}_i^T \rangle \langle \boldsymbol{s}_k \boldsymbol{s}_k^T \rangle \right) +$$
$$\langle {\Delta^{ii}}^2 {u_k^i}^2 \rangle - 2x_k^{(i)} \langle \boldsymbol{c}_i^T \boldsymbol{s}_k \rangle + 2\langle \boldsymbol{c}_i^T \boldsymbol{s}_k \Delta^{ii} u_k^i \rangle - 2x_k^{(i)} \langle \Delta^{ii} u_k^i \rangle \Big);$$

5. The posterior distribution of a_i upto some normalizing constant is given by

$$\tilde{q}(a_i) = \exp\left\{\sum_{k=2}^{N}\langle\log p(\boldsymbol{s}_k^i|\boldsymbol{s}_{k-1}^i, a_i)\rangle\right\} p(a_i|\alpha_a, \beta_a)$$

   Since the $\mathcal{B}eta$ distribution is not conjugate to the Gaussian likelihood, the posterior distribution does not belong to a known family. Hence, the importance sampling is performed to calculate the expectations with respect to the distribution $q(a_i)$, as shown below.

$$\langle f\rangle = \sum_{l=1}^{L} f(a_i^l)\hat{w}(a_i^l) \tag{7.10}$$

   where $\hat{w}(a_i^l) = \frac{\tilde{w}(a_i^l)}{\sum_{l=1}^{L}\tilde{w}(a_i^l)}$ and $\tilde{w}(a_i^l) = \frac{\tilde{q}(a_i^l)}{\tilde{g}(a_i^l)}$. Here $f(a_i)$ is some function of a_i whose expectations are of primary interest. The samples $a_i^l\ \forall\ l \in \{1, 2, \ldots L\}$ are drawn from an easier distribution $\tilde{g}(a_i)$, for example, $p(a_i|\alpha_a, \beta_a)$, and the introduced bias is corrected by $\hat{w}$. Therefore

$$\hat{w}(a_i^l) = \frac{\exp\left\{\sum_{k=2}^{N}\langle\log p(s_k^{(i)}|s_{k-1}^{(i)}, a_i^l)\rangle\right\}}{\sum_{l=1}^{L}\exp\left\{\sum_{k=2}^{N}\langle\log p(s_k^{(i)}|s_{k-1}^{(i)}, a_i^l)\rangle\right\}}$$

   where

$$\begin{aligned}\sum_{k=2}^{N}\langle\log p(s_k^{(i)}|s_{k-1}^{(i)}, a_i^l)\rangle &= -\frac{N-1}{2}\log(1-a_i^2) - \frac{1}{2}\left(\sum_{k=2}^{N}\langle s_{k-1}^{(i)}s_{k-1}^{(i)}\rangle\right)\frac{a_i^2}{1-a_i^2}\\ &+ \left(\sum_{k=2}^{N}\langle s_k^{(i)}s_{k-1}^{(i)}\rangle\right)\frac{a_i}{1-a_i^2} - \frac{1}{2}\left(\sum_{k=2}^{N}\langle s_k^{(i)}s_k^{(i)}\rangle\right)\frac{1}{1-a_i^2}\end{aligned}$$

6. The derived proposal distribution for ν_i up to some normalizing constant is given by

$$\tilde{q}(\nu_i) = \exp\left\{\sum_{k=1}^{N}\langle\log p(\Lambda_k^{ii}|\nu_i)\rangle\right\} p(\nu_i|\alpha_\nu, \beta_\nu)$$

   Since it does not belong to any known distribution, we again use important sampling, as shown below, where samples $\nu_i^l\ \forall\ l \in \{1, 2, \ldots L\}$ are drawn from the distribution $p(\nu_i|\alpha_\nu, \beta_\nu)$.

$$\hat{w}(\nu_i^l) = \frac{\exp\left\{\sum_{k=1}^{N}\langle\log p(\Lambda_k^{ii}|\nu_i^l)\rangle\right\}}{\sum_{l=1}^{L}\exp\left\{\sum_{k=1}^{N}\langle\log p(\Lambda_k^{ii}|\nu_i^l)\rangle\right\}}$$

where

$$\sum_{k=1}^{N}\langle \log p(\Lambda_k^{ii}|\nu_i^l)\rangle = \frac{\nu_i^l N}{2}\log\frac{\nu_i^l}{2} - N\log\Gamma\left(\frac{\nu_i^l}{2}\right) + \left(\frac{\nu_i^l}{2}-1\right)\sum_{k=1}^{N}\log\Lambda_k - \frac{\nu_i^l}{2}\sum_{k=1}^{N}\Lambda_k$$

7. The derived proposal distribution $q(s_{1:N})$ of slow features

$$\begin{aligned}\log q(\boldsymbol{s}_{1:N}) &\propto \langle \log p(\boldsymbol{x}_{1:N}, \boldsymbol{s}_{1:N}, \boldsymbol{u}_{1:N}, \Lambda_{1:N}, A, C, R, \Delta)\rangle \\ &\propto \langle \log p(\boldsymbol{s}_{1:N}|\boldsymbol{x}_{1:N}, \boldsymbol{u}_{1:N}, \Lambda_{1:N}, A, C, R, \Delta)\rangle \\ &\not\propto \;\log p(\boldsymbol{s}_{1:N}|\boldsymbol{x}_{1:N}, \langle\boldsymbol{u}_{1:N}\rangle, \langle\Lambda_{1:N}\rangle, \langle A\rangle, \langle C\rangle, \langle R\rangle, \langle\Delta\rangle)\end{aligned}$$

To infer the distribution $q(\boldsymbol{s}_{1:N})$ using the classical Kalman filter algorithm, we define the following:

$$\begin{aligned}\tilde{x}_k &= \begin{bmatrix} x_k - \langle\Delta\rangle\langle\boldsymbol{u}_k\rangle \\ 0_m \\ 0_m \end{bmatrix} \forall\, k = 1, 2, \cdots N; \\ \tilde{C}_k &= \begin{bmatrix} \langle C\rangle \\ U^A \\ U_k^C \end{bmatrix} \forall\, k = 1, 2, \cdots N-1 \text{ and } \tilde{C}_N = \begin{bmatrix} \langle C\rangle \\ 0_{m\times m} \\ U_N^C \end{bmatrix}; \\ \tilde{R}_k &= \mathrm{diag}\left[\langle\Lambda_k^{-1}\rangle\langle R^{-1}\rangle^{-1}, \mathrm{I}_m, \mathrm{I}_m\right]; \quad \tilde{A} = \langle A\rangle;\end{aligned}$$

with U^A and U_k^C defined by the Cholesky decompositions of $\langle A^T(I_m - AA^T)^{-1}A\rangle - \langle A^T\rangle\langle(\mathrm{I}_m - AA^T)^{-1}\rangle\langle A\rangle$ and $\langle C^T R^{-1}\Lambda_k C\rangle - \langle C^T\rangle\langle R^{-1}\Lambda_k\rangle\langle C\rangle$, respectively. Finally, the following equation is obtained from the unified inference theorem [150]

$$\begin{aligned}q(\boldsymbol{s}_{1:N}) &= \tilde{p}(\boldsymbol{s}_{1:N}|\tilde{\boldsymbol{x}}_{1:N}, \langle\tilde{A}\rangle, \langle\tilde{C}\rangle_{1:N}, \langle\tilde{R}\rangle_{1:N}) \\ &= \prod_{k=1}^{N}\mathcal{N}(\boldsymbol{s}_k; \boldsymbol{s}_{k|N}, P_{k|N})\end{aligned}$$

where

$$\begin{aligned}P_{k|k-1} &= \tilde{A}P_{k-1|k-1}\tilde{A}^T + I_m - \tilde{A}\tilde{A}^T \\ K_x &= P_{k|k-1}\tilde{C}_k^T(\tilde{C}_k P_{k|k-1}\tilde{C}_k^T + \tilde{R}_k)^{-1} \\ \boldsymbol{s}_{k|k} &= \tilde{A}\boldsymbol{s}_{k-1|k-1} + K_x(\tilde{\boldsymbol{x}}_k - \tilde{C}_k\tilde{A}\boldsymbol{s}_{k-1|k-1}) \\ P_{k|k} &= (I_m - K_x\tilde{C}_k)P_{k|k-1}\end{aligned}$$

The smoothing is applied using the following equations

$$J_k = P_{k|k}\tilde{A}^T P_{k+1|k}^{-1}$$

$$\begin{aligned} \boldsymbol{s}_{k|N} &= \boldsymbol{s}_{k|k} + J_k(\boldsymbol{s}_{k+1|N} - \tilde{A}\boldsymbol{s}_{k|k}) \\ P_{k|N} &= P_{k|k} + J_k(P_{k+1|N} - P_{k+1|k})J_k^T \end{aligned}$$

8. $q(\boldsymbol{u}_k|\boldsymbol{x}_{1:k};\theta) = \mathcal{N}_+(\boldsymbol{u}_k;\boldsymbol{u}_{k|N}, U_{k|N})$ where

$$\begin{aligned} \epsilon_k &= \boldsymbol{x}_k - \langle C\rangle \boldsymbol{s}_{k|N} \\ K_u &= \langle \Delta\rangle(\langle \Delta^T\Delta\rangle + \langle R^{-1}\rangle^{-1})^{-1} \\ \boldsymbol{u}_{k|N} &= K_u\epsilon_k \\ U_{k|N} &= (I_p - K_u\langle\Delta\rangle)\langle\Lambda_k\rangle^{-1} \end{aligned}$$

The following expressions are computed $\forall\, 1 \leq i \leq p$ to estimate the mean and covariance of $\boldsymbol{u}_k$.

$$\begin{aligned} \chi_k^i &= \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{{u_{k|N}^i}^2}{2U_{k|N}^{ii}}\right\}; \\ \omega_k^i &= \frac{1}{2}\left(1 + \text{ erf }\left(\frac{-u_{k|N}^i}{\sqrt{2U_{k|N}^{ii}}}\right)\right); \\ \gamma_k^i &= u_{k|N}^i + \sqrt{U_{k|N}^{ii}}\left(\frac{\chi_k^i}{1-\omega_k^i}\right); \\ \Sigma_k^{ii} &= U_{k|N}^{ii}\left[1 - \left(\frac{u_{k|N}^i}{\sqrt{U_{k|N}^{ii}}}\right)\left(\frac{\chi_k^i}{1-\omega_k^i}\right) - \left(\frac{\chi_k^i}{1-\omega_k^i}\right)^2\right] + {\gamma_k^i}^2; \end{aligned}$$

9. $q(\Lambda_k^{ii}|\boldsymbol{x}_{1:N};\theta) = \mathcal{G}\left(\Lambda_k^{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + \phi_k^{ii}}{2}\right)$ where

$$\phi_k = \left\langle R^{-1}(\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^T + CP_{k|N}C^T) + (\Delta R^{-1}\Delta + I)\boldsymbol{u}_k\boldsymbol{u}_k^T - R^{-1}\Delta\boldsymbol{u}_k\boldsymbol{\epsilon}_k^T - \Delta R^{-1}\boldsymbol{\epsilon}_k\boldsymbol{u}_k^T\right\rangle$$

Finally, the expectations that are required to compute the parameter update expressions are shown below.

$$\begin{aligned} \langle\Delta\rangle &= \text{diag}(\mu_\Delta); \\ \left\langle R^{ii^{-1}}\right\rangle &= \frac{\alpha_{R^{ii}}}{\beta_{R^{ii}}}; \\ \left\langle h_i^{j^{-1}}\right\rangle &= \frac{2}{\sqrt{2\lambda}|\langle c_i^j\rangle|}; \\ \langle c_i c_i^T\rangle &= \Sigma_{c_i} + \langle c_i\rangle\langle c_i\rangle^T; \end{aligned}$$

$$
\begin{aligned}
\langle \boldsymbol{s}_k \rangle &= \boldsymbol{s}_{k|N};\\
\left\langle \boldsymbol{s}_k \boldsymbol{s}_{k-1}^T \right\rangle &= P_{k|N} J_{k-1}^T + \boldsymbol{s}_{k|N} \boldsymbol{s}_{k-1|N}^T;\\
\left\langle \boldsymbol{s}_k \boldsymbol{s}_k^T \right\rangle &= P_{k|k} + \boldsymbol{s}_{k|N} \boldsymbol{s}_{k|N}^T;\\
\langle \boldsymbol{u}_k \rangle &= \boldsymbol{\gamma}_k;\\
\left\langle \boldsymbol{u}_k \boldsymbol{u}_k^T \right\rangle &= \Sigma_k;\\
\left\langle \Lambda_k^{ii} \right\rangle &= \frac{\nu_i + 2}{\nu_i + \phi_k^{ii}};\\
\left\langle \log \Lambda_k^{ii} \right\rangle &= \psi\left(\frac{\nu_i}{2} + 1\right) - \log\left(\frac{\nu_i + \phi_k^{ii}}{2}\right);
\end{aligned}
$$

Given a hyperparameter set, the proposed algorithm is run iteratively until the convergence of the variational free energy is achieved. The variational free energy is defined as follows.

$$
L(q(d)) = \langle \log p(X, d|\theta) \rangle_{q(d)} - \langle \log q(d) \rangle_{q(d)}
$$

Different initial hyperparameter guess results in a different variational free energy at the end of iterations. Therefore, the hyperparameters are selected based on the validation data's variational free energy with a preference for slowness and sparsity. Further, we assume that outputs are not available during the online implementation, and hence, only input information is being used to evaluate the states of the Kalman filter [98]. In particular, only C_z is utilized to calculate the Kalman gain K_x, slow feature $s_{k|k}$, and the covariance matrix $P_{k|k}$. Even though the output information is not used during the state computation, the dimension of $s_{k|k}$ does not alter. It is to be noted that the Kalman filter is used to reconstruct only the input data in the online implementation. Since the dimension of retained states (slow features) is much lower than the dimension of the input due to sparsity constraint, the observability is not a problem.

7.5 Soft Sensor Case Studies

This section demonstrates the efficiency of the proposed robust sparse probabilistic slow feature model using a simulated and experimental data set for soft-sensor applications with hyperparameters $\{\alpha_a, \beta_a, \lambda, \alpha_R, \beta_R, \mu_\Delta, \Sigma_\Delta, \alpha_\nu, \beta_\nu\} = \{5, 1, 0.005, 3, 0.5, 0,$

$1, 50, 1\}$. The measurement noise is drawn from a skewed t-distribution with $\Delta = 2$, $\nu = 7$, and a scale parameter R such that the signal-to-noise ratio is 1.

7.5.1 Simulation case study

The state-transition matrix for this constructed simulation is chosen to be $A = \text{diag}[0.99, 0.75]$. Two slow features and four observed variables with 1500 data samples (800, 200 and 500 for training, validation and testing, respectively) are generated using (7.3)-(7.4). The fourth observed variable serves as an output, whereas the remaining three are used as inputs. We do not make any differentiation between these variables during the training phase. Due to the assumption that the output is not available during testing, only the inputs and the input emission matrix C_z are used to construct the Kalman gain and, thus, the slow features. The emission matrix C used for this study is shown in (7.11). Finally, a skewed t-measurement noise is added to the observed variables. The simulated input-output variables are shown in the Fig. 7.3.

$$C = \begin{bmatrix} 0.54 & 0.32 \\ 1.83 & -1.31 \\ 0 & -0.43 \\ 0.86 & 0 \end{bmatrix}; \tag{7.11}$$

Latent variables are extracted using the proposed methodology and other state-of-the-art feature extraction methods, such as variational Bayesian probabilistic slow feature analysis (VBPSFA) [99], dynamic partial least squares (DPLS), slow feature analysis (SFA) [173], robust PSFA [56], and least squares. The estimated emission matrices using the VBPSFA and the proposed methodology are shown in the Fig. 7.4, and Fig. 7.5, respectively. It is observed that the estimated emission matrix using the proposed method is sparse and thus, it can be inferred that the effective number of latent variables required for the reconstruction of the input variables is two. Further, only one feature is required for the prediction of the output variable since the second element of the last row is statistically zero. Table 7.2 shows the performance indices, namely the root mean square error (RMSE) and the concordance correlation coefficient (ρ_c), calculated with the help of predicted and the actual output testing

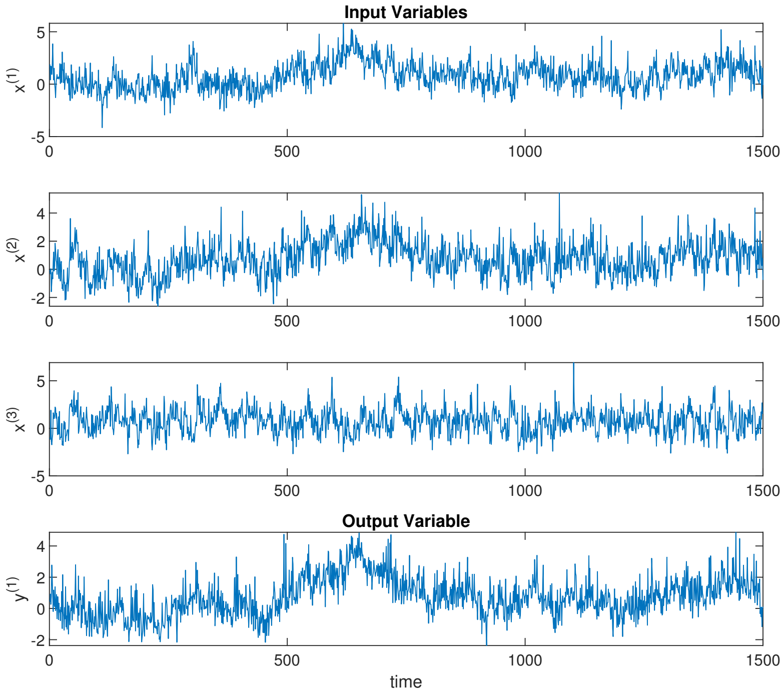


Figure 7.3: Three input variables and one output variable

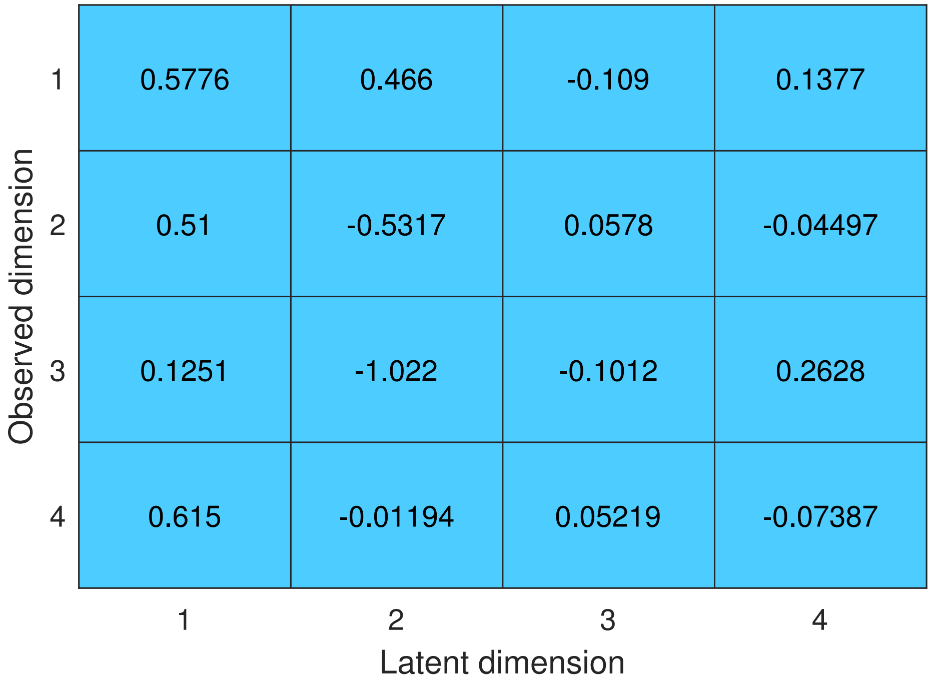
dataset. It is observed that the proposed method results in the lowest RMSE due to the explicit representation of the outliers and skewness.

Table 7.2: Performance comparison on the simulation dataset

Method	RSPSFA	VBPSFA	DPLS	SFA	RPSFA	OLS
RMSE	0.34	0.68	0.86	0.88	0.52	0.90
ρ_c	0.83	0.53	0.46	0.42	0.60	0.41
Computational time (in sec.)	16.44	10.12	0.02	0.01	11.60	0.01

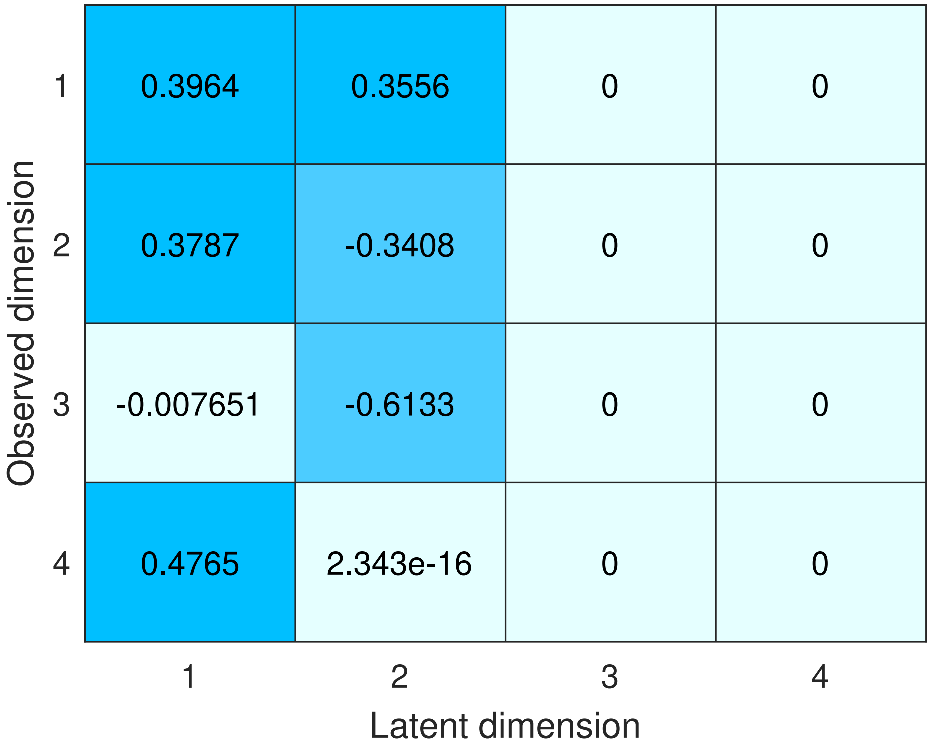
7.5.2 Experimental case study

In this subsection, the efficiency of the proposed method is further validated using the data generated from a pilot-scale hybrid tank system. Three cylindrical tanks are connected in series through six valves, and water can be pumped into the left and the right tanks separately using two pumps, as shown in Fig. 7.9. Three exit valves are situated at the bottom of each of the tanks. The water level in the middle tank is the


Observed dimension
Latent dimension

	1	2	3	4
1	0.5776	0.466	-0.109	0.1377
2	0.51	-0.5317	0.0578	-0.04497
3	0.1251	-1.022	-0.1012	0.2628
4	0.615	-0.01194	0.05219	-0.07387

Figure 7.4: Estimated C using VBPSFA. It is observed that all the entries are statistically significant; therefore, they are coloured dark blue.


Observed dimension
Latent dimension

	1	2	3	4
1	0.3964	0.3556	0	0
2	0.3787	-0.3408	0	0
3	-0.007651	-0.6133	0	0
4	0.4765	2.343e-16	0	0

Figure 7.5: Estimated C using RSPSFA. All the entries that are statistically close to zero are coloured light blue.

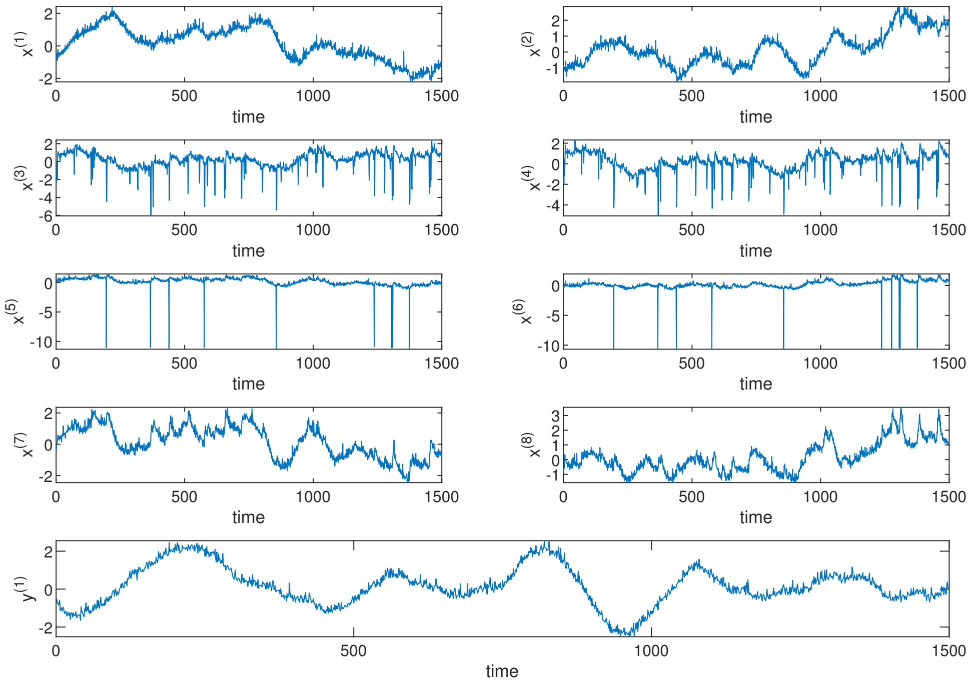
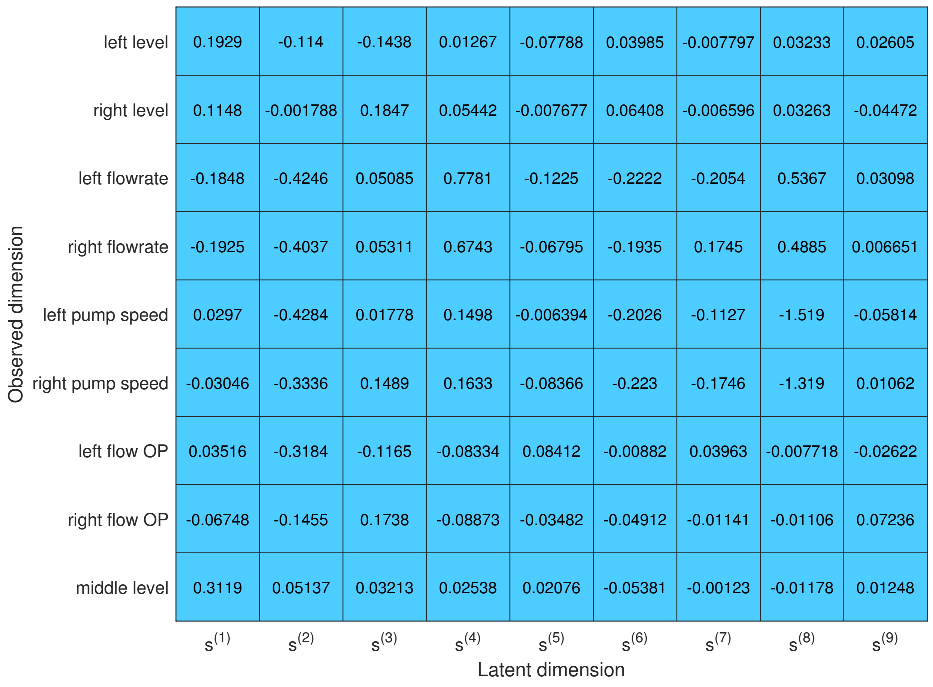


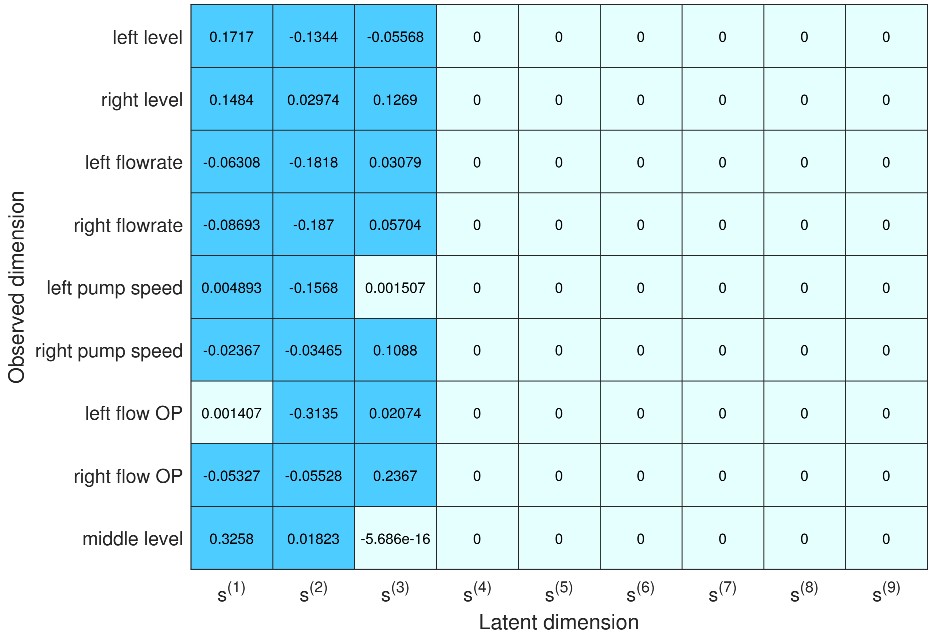
Figure 7.6: Input-Output variables of the experimental case study



Observed dimension	$s^{(1)}$	$s^{(2)}$	$s^{(3)}$	$s^{(4)}$	$s^{(5)}$	$s^{(6)}$	$s^{(7)}$	$s^{(8)}$	$s^{(9)}$
left level	0.1929	-0.114	-0.1438	0.01267	-0.07788	0.03985	-0.007797	0.03233	0.02605
right level	0.1148	-0.001788	0.1847	0.05442	-0.007677	0.06408	-0.006596	0.03263	-0.04472
left flowrate	-0.1848	-0.4246	0.05085	0.7781	-0.1225	-0.2222	-0.2054	0.5367	0.03098
right flowrate	-0.1925	-0.4037	0.05311	0.6743	-0.06795	-0.1935	0.1745	0.4885	0.006651
left pump speed	0.0297	-0.4284	0.01778	0.1498	-0.006394	-0.2026	-0.1127	-1.519	-0.05814
right pump speed	-0.03046	-0.3336	0.1489	0.1633	-0.08366	-0.223	-0.1746	-1.319	0.01062
left flow OP	0.03516	-0.3184	-0.1165	-0.08334	0.08412	-0.00882	0.03963	-0.007718	-0.02622
right flow OP	-0.06748	-0.1455	0.1738	-0.08873	-0.03482	-0.04912	-0.01141	-0.01106	0.07236
middle level	0.3119	0.05137	0.03213	0.02538	0.02076	-0.05381	-0.00123	-0.01178	0.01248

Latent dimension

Figure 7.7: Estimated C using VBPSFA.



Observed dimension	$s^{(1)}$	$s^{(2)}$	$s^{(3)}$	$s^{(4)}$	$s^{(5)}$	$s^{(6)}$	$s^{(7)}$	$s^{(8)}$	$s^{(9)}$
left level	0.1717	-0.1344	-0.05568	0	0	0	0	0	0
right level	0.1484	0.02974	0.1269	0	0	0	0	0	0
left flowrate	-0.06308	-0.1818	0.03079	0	0	0	0	0	0
right flowrate	-0.08693	-0.187	0.05704	0	0	0	0	0	0
left pump speed	0.004893	-0.1568	0.001507	0	0	0	0	0	0
right pump speed	-0.02367	-0.03465	0.1088	0	0	0	0	0	0
left flow OP	0.001407	-0.3135	0.02074	0	0	0	0	0	0
right flow OP	-0.05327	-0.05528	0.2367	0	0	0	0	0	0
middle level	0.3258	0.01823	-5.686e-16	0	0	0	0	0	0

Figure 7.8: Estimated C using RSPSFA.

interest in the current experimental case study. We consider a total of eight inputs: left and right tank level, left and right flow rate, left and right pump speed, and left and right flow controller output. Readers are referred to [56] for further details of the experimental setup. The measured and the target variables are shown in their normalized form in Fig. 7.6. The dataset is partitioned into training, validation, and testing sets with 800, 200, and 500 samples, respectively.

Table 7.3: Performance comparison on the experimental dataset

Method	RSPSFA	VBSFA	DPLS	SFA	RPSFA	OLS
RMSE	0.23	0.32	0.39	0.48	0.30	0.42
ρ_c	0.90	0.76	0.73	0.65	0.79	0.70
R^2	0.77	0.60	0.37	0.08	0.67	0.27

We perform a similar analysis as discussed in the simulation case study. The proposed algorithm is iterated until the Q-function, computed using the validation data, no longer improves. It is difficult to interpret the importance of the extracted features using the estimated emission matrix obtained from the VBPSFA method, as shown in Fig. 7.7. However, it is observed that only three features are needed to sufficiently account for all the input-output variables, as shown in Fig. 7.8. Several

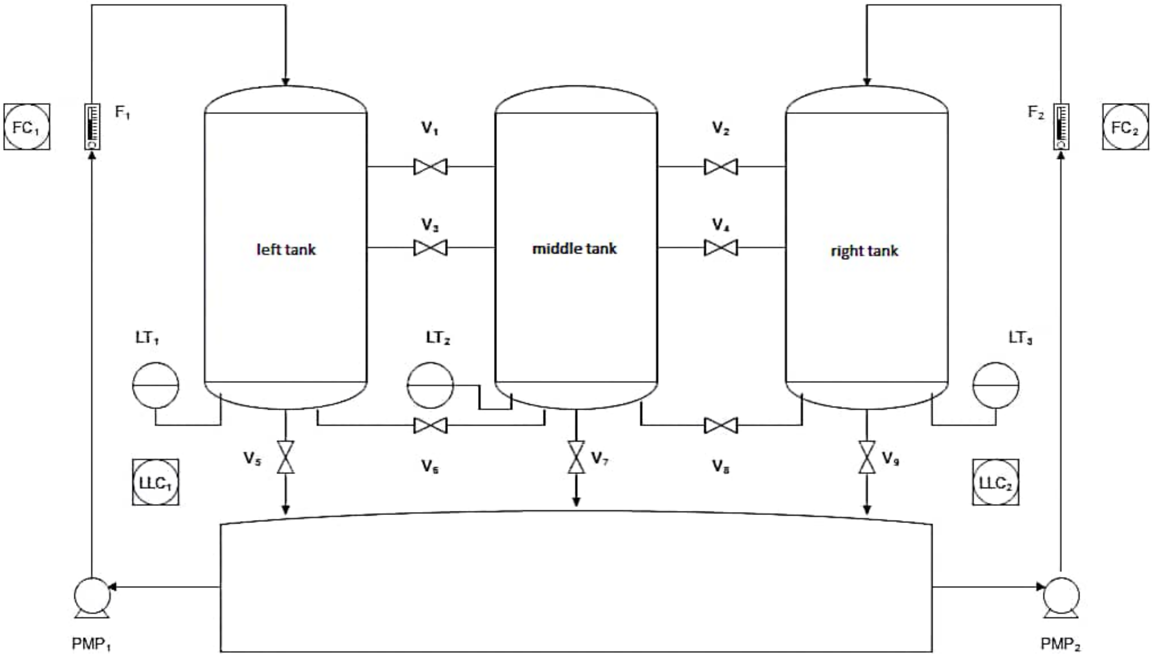


Figure 7.9: Experimental setup

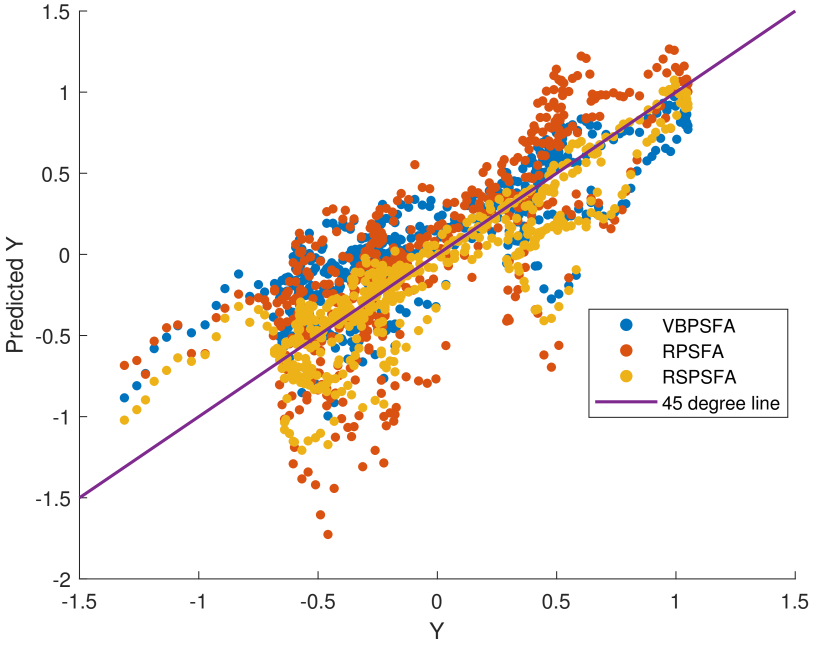


Figure 7.10: Scatter plot between the predicted and actual output

performance indices are constructed utilizing the predictions obtained from different feature extraction methods on the test dataset, and their results are shown in Table 7.3. It is observed that the proposed method has resulted in higher R^2 compared to the other models due to the efficient modelling of the emission matrix and the measurement noise. Fig. 7.10 indicates the plot between the noise-free output variable and the predictions based on three latent variable models. It is observed that the predictions based on the RSPSFA features are closer to the 45^0 line, indicating a better prediction.

Apart from the increased accuracy, the proposed algorithm also has advantages in terms of physical interpretability. It can be observed from Fig. 7.8 that the most dominant latent variable in the level of the middle tank is $s^{(1)}$, which is also the dominant latent variable in the levels of the left and right tanks. Thus, it can be inferred that these two variables are most related to the level in the middle tank. This corroborates with the process knowledge as evident in Fig. 7.9, where it can be seen that the water level in the middle tank is most influenced by the levels in the right and left tanks. Additionally, it can be observed that $s^{(2)}$ is predominantly observed in the variables to the left side of the middle tank and $s^{(3)}$ in the variables to the right side of the middle tank. Although this rule is broken for the flow rate, one may conclude that $s^{(2)}$ and $s^{(3)}$ primarily represent the variations observed in the left and right-hand side variables of the experimental setup. Such an interpretation is not possible for the emission matrix of the VBPSFA model (Fig. 7.7)

7.6 Conclusion

This chapter introduces a new feature extraction model that extracts only the essential slow features from data corrupted with outliers and skewed noise. A Laplace and skewed t-distribution are introduced for the emission matrix and measurement noise to achieve a robust sparse slow feature model. Further, the posterior distributions of all the latent variables are derived under the variational inference framework. We deduce that the proposed algorithm performs better than the state-of-the-art models, especially when the noise is skewed with outliers, as supported by the simulation and experimental case study results. Since the proposed method results in a sparse representation of the latent space that generates the data, the latent variables can be helpful for process monitoring applications. The limitation of the proposed algorithm is that it requires a slightly higher training computation time than the other state-of-the-art methods, as shown in Table 7.2. However, the online implementation does not require much time since the Kalman gain and, thus, prediction is computed using standard filtering equations. The proposed model can be further extended to accommodate multi-modal process data, where the operating conditions change over a period of time.

Chapter 8

Physics-Informed Probabilistic Slow Feature Analysis *

This chapter presents a novel approach called the physics-informed slow feature analysis. Slow feature analysis, a probabilistic method, is employed to extract slowly varying latent patterns from high-dimensional measured data. The extracted slow features have proven effective in industrial applications such as soft sensing and process monitoring. However, industrial processes come with various physical constraints that must be taken into account, such as energy requirements, equipment limitations, and safety considerations. The conventional black-box nature of the slow feature model often leads to physically inconsistent or unacceptable results. To address this issue, we propose integrating physics principles into the probabilistic slow feature model, ensuring that the extracted features adhere to physics laws. Our formulation incorporates two types of physical constraints: linear algebraic equality and inequality constraints. The model parameters are estimated using the expectation-maximization approach. Through an experimental case study, we demonstrate the effectiveness of our methodology, showcasing the advantages of incorporating physics in feature extraction. These advantages include improved interpretability, reduced data dimensionality, and enhanced generalization performance.

*This chapter has been submitted as: **V. K. Puli**, R. Chiplunkar and B. Huang, "Physics-Informed Probabilistic Slow Feature Analysis", in Automatica, 2023

8.1 Introduction

In recent years, there has been a growing prominence in the use of data-based modeling to represent dynamic systems. This trend has been driven by a confluence of multiple factors, including enhanced data storage and handling capabilities, the emergence of sophisticated data analytics tools, and ambitious manufacturing goals [202]. Although data analytics has been maturing over the years, developing dynamic models that yield reliable long-term predictions remains a challenging problem. This challenge arises due to issues such as sensory errors and the lack of comprehensive dynamic information in historical data. To overcome these challenges, researchers have turned to mechanistic models, which are built on a deep understanding of the underlying physics of the processes. Integrating these models with observed data can lead to enhanced models, a concept commonly referred to as ”physics-informed machine learning” [203]. As a result, this approach has garnered significant interest and application in diverse fields, such as climatology [204], fluid mechanics [205], power-systems [206], and process systems engineering [207]. Although there is substantial research in the domain of physics-informed neural networks, the fusion of physics-based knowledge with other data analysis methods remains relatively unexplored territory.

Latent variable models [186] are a powerful class of statistical tools that enable the analysis of complex data by capturing underlying structures and relationships among observed variables. Among these models, Principal Component Analysis (PCA) [34] stands out as a widely-used technique for dimensionality reduction and feature extraction. However, in dynamic systems with temporal dependencies, traditional PCA may not fully capture the evolving patterns. This limitation has led to the development of Dynamic Latent Variable Models (DLVM) [208–210], which encompass a range of methods tailored to model time-varying data. One prominent approach in this domain is Slow Feature Analysis (SFA) [173], which focuses on detecting and characterizing slow variations in dynamic systems. SFA has shown promise in various domains, particularly in process industries [174] where slow dynamics are prevalent.

SFA is designed to extract latent variables called ”slow features” by minimizing their velocity [173]. These slow features are utilized in modeling, as they capture significant variations observed in systems primarily driven by slower changes. SFA has been widely adopted in applications such as soft sensing and process monitoring [174]. Probabilistic Slow Feature Analysis (PSFA), an alternative interpretation of SFA, assumes the latent variables to be dynamic random variables [54,55]. The probabilistic nature of PSFA equips it with enhanced capabilities in handling noise, missing data, and model uncertainties. Beyond basic PSFA versions, various extensions have been proposed to address specific aspects of the data. For instance, Fan et al. introduced a robust PSFA framework to handle measurement outliers [56], while Ma and Huang extended the PSFA model to incorporate model uncertainties [99]. Furthermore, Puli et al. proposed the complex PSFA model tailored for datasets characterized by oscillations [129]. Recent literature has presented numerous other variants of the PSFA model that consider aspects like nonstationary data [49], multimodal processes [197], sparsity of the latent space [211], and irregular sampling rates [212].

The importance of physics information in enhancing the reliability of data-based models has been highlighted previously. However, existing research on physics-informed latent variable modeling has been confined to static models [213, 214] and has not explored dynamic latent variable models. To bridge this gap, this study introduces a novel approach that integrates physics knowledge with data in a DLVM framework, specifically using the PSFA model due to its suitability for process systems modeling. The incorporation of physics information occurs through linear equality and inequality constraints. Equality constraints arise from physical realities or equilibrium relations among variables (e.g., constant sum of concentrations or steady-state mass/energy balance equations). Inequality constraints, on the other hand, define the physical limits or safety-related constraints for variables. To account for the probabilistic nature of PSFA, this study proposes formulating these constraints in a probabilistic manner. Both constraints are expressed similarly to the emission equation of the PSFA model, with the key difference lying in the choice of distributions. Specifically, a Gaussian distribution is utilized to represent the uncertainty in the equality constraints, while a truncated Gaussian distribution is employed for the inequality constraints. The model estimation is performed using the expectation-maximization (EM) algorithm [215]. The effectiveness of the proposed framework is demonstrated through a case study involving a pilot-scale hybrid tank system. The results show that the incorporation of physics information leads to physically consistent predictions. The main contributions of this chapter can be outlined as follows:

1. PSFA model is utilized to incorporate the physical constraints of the system in a probabilistic fashion, encompassing both equality and inequality aspects.
2. The model estimation procedure is subsequently elaborated upon, with a thorough derivation provided using the EM algorithm.

The chapter is structured as follows: In Section 8.2, the SFA and PSFA methods are introduced. Section 8.3 presents the physics-informed PSFA method proposed in this study. Subsequently, Section 8.4 discusses the results obtained from the case study, while the conclusions are summarized in Section 8.5.

8.2 Literature

Given an input sequence $X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_{\mathcal{T}} \end{bmatrix}$, $\boldsymbol{x}_k \in \mathbb{R}^p$, the optimization problem shown in (8.1)-(8.5) can be solved to obtain slow features in the order of increasing velocities.

$$\begin{aligned} \min_{W} \quad & \langle \dot{s}_k^{(i)^2} \rangle && (8.1) \\ \text{s.t} \quad & \boldsymbol{s}_k = W^T \boldsymbol{x}_k && (8.2) \\ & \langle s_k^{(i)} \rangle = 0 && (8.3) \\ & \langle s_k^{(i)^2} \rangle = 1 && (8.4) \\ \forall i \neq j, \, & \langle s_k^{(i)} \cdot s_k^{(j)} \rangle = 0 && (8.5) \end{aligned}$$

where $\langle \dot{s}_k^{(i)^2} \rangle = \frac{1}{\mathcal{T}-1} \sum_{k=2}^{\mathcal{T}} (s_k^{(i)} - s_{k-1}^{(i)})^2$ denotes the squared average velocity, $W \in \mathbb{R}^{p \times m}$ indicates the projection matrix and $\langle \cdot \rangle$ stands for the average over data samples. Equation (8.3) - (8.4) are applied to each slow feature to avoid trivial solutions, whereas (8.5) ensures zero correlation among the extracted features.

Three observed variables are depicted in Fig. 8.1, originating from two cosines with varying frequencies and a third variable representing a brief, high-frequency cosine signal. By employing DSFA, we can accurately identify the latent variables associated with different frequencies, as illustrated in Fig. 8.3. In contrast to PCA, which captures the latent variable with the highest variability (Fig. 8.2), SFA can effectively isolate and eliminate the impact of the short-lived disturbance on other latent variables.

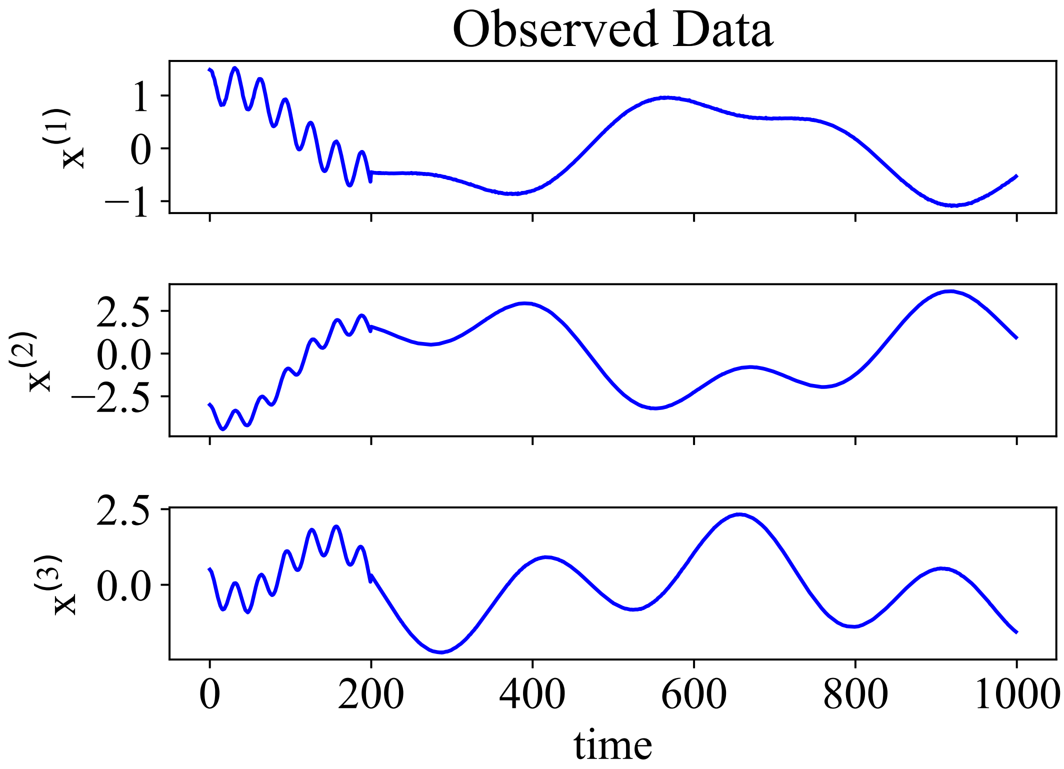


Figure 8.1: Measured variables

However, SFA assumes that the underlying factors causing the data variations are deterministic, which may not always hold true in real-world scenarios. Therefore, the probabilistic formulation [54, 55] is introduced, as shown in (8.6) - (8.7).

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \quad \boldsymbol{w}_k \sim \mathcal{N}(0, \Gamma) \tag{8.6}$$

$$\boldsymbol{x}_k = C\boldsymbol{s}_k + \boldsymbol{v}_k; \quad \boldsymbol{v}_k \sim \mathcal{N}(0, R) \tag{8.7}$$

where $A \in \mathbb{R}^{m\times m}$, $C \in \mathbb{R}^{p\times m}$, $\Gamma \in \mathbb{R}^{m\times m}$, and $R \in \mathbb{R}^{p\times p}$ are the state-transition matrix, the emission matrix, the state-noise covariance matrix, and the measurement noise covariance matrix, respectively. Assumptions (8.8)-(8.10) were made regarding the parameters involved to ensure that, in the limiting case, the solution of proba-

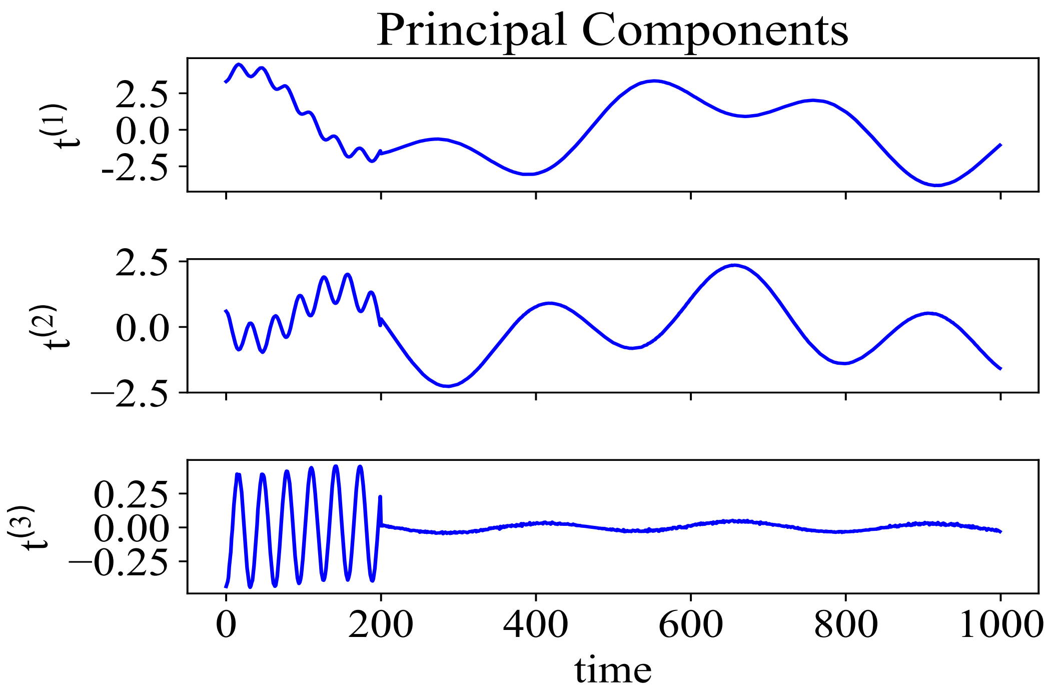


Figure 8.2: Princpal components

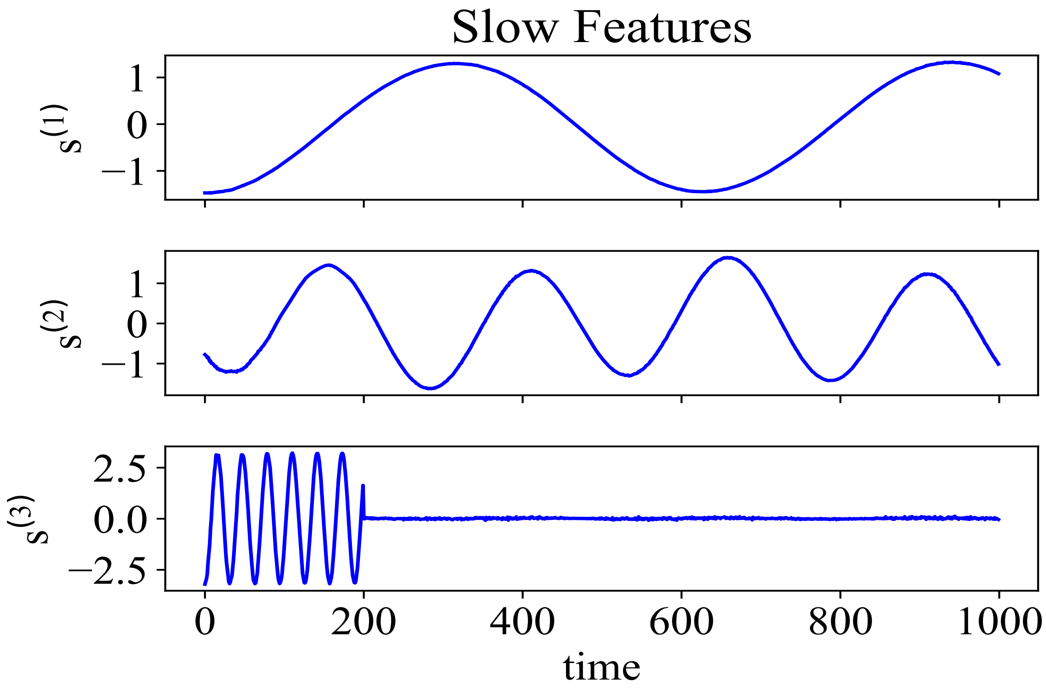


Figure 8.3: Slow features

bilistic slow feature analysis converges to its deterministic counterpart.

$$A = \text{diag}(a_1, a_2, \cdots, a_m) \tag{8.8}$$

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \cdots, \gamma_m) \tag{8.9}$$

$$\gamma_i = 1 - a_i^2 \ \forall 1 \leq i \leq m \tag{8.10}$$

The expectation-maximization algorithm is employed to estimate the distribution of slow features and associated parameters solely from measured data. However, heavy reliance on measured data may render decision-making unreliable and unrealistic, especially when the data length is small or contains faulty regions. To address this issue, we propose a novel algorithm that incorporates process knowledge, enabling intelligent and reliable decision-making, as demonstrated in section 8.3.

8.3 Physics-informed slow feature model

In this section, it is assumed that the observed variables adhere to the physics laws, manifesting in the form of two constraints as described below.

1. The first constraint is expressed as a linear equality relation among the non-mean-centered observed variables without the loss of generality.

$$\begin{aligned} M(\boldsymbol{x}_k + \bar{\boldsymbol{x}}) &= b \\ \implies b - MC\boldsymbol{s}_k - M\bar{\boldsymbol{x}} &= 0 \end{aligned}$$

   where $\bar{\boldsymbol{x}}$ is the mean of the observed data. Uncertainty is introduced to account for the reliability of expert information, as shown in (8.11).

$$b - MC\boldsymbol{s}_k - M\bar{\boldsymbol{x}} = \boldsymbol{u}_k; \quad \boldsymbol{u}_k \sim \mathcal{N}(0, Q) \tag{8.11}$$

   Here, the confidence associated with the physics-based formulation is denoted as Q, while the discrepancy between the linear expert physics and the actual reality is represented as $\boldsymbol{u}_k$.

2. The second form of physics manifests as a linear inequality relation among the observed variables.

$$N(\boldsymbol{x}_k + \bar{\boldsymbol{x}}) < f$$

$$\implies f - NC\boldsymbol{s}_k - N\bar{\boldsymbol{x}} > 0$$

> Similarly, the reliability of this inequality information can be modelled as shown in (8.12).
>
> $$f - NC\boldsymbol{s}_k - N\bar{\boldsymbol{x}} = \boldsymbol{e}_k \tag{8.12}$$
>
> Due to the fact that $\boldsymbol{e}_k > 0$ is greater than zero, a simple Gaussian distribution assumption is inadequate in this case. A distribution that has positive support may be utilized instead. Specifically, a skew-normal distribution is selected due to its ability to be expressed as a Gaussian scale mixture, effectively resolving state estimation concerns with a non-Gaussian distribution.

Thus, by combining all the relevant information, the physics-informed slow feature model is presented below.

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \tag{8.13}$$

$$\begin{bmatrix} \boldsymbol{x}_k \\ b \\ f \end{bmatrix} = \begin{bmatrix} C \\ MC \\ NC \end{bmatrix} \boldsymbol{s}_k + \begin{bmatrix} 0 \\ M \\ N \end{bmatrix} \bar{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{v}_k \\ \boldsymbol{u}_k \\ \boldsymbol{e}_k \end{bmatrix}; \tag{8.14}$$

where

$$\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k; 0, I - AA^T); \;\; \boldsymbol{v}_k \sim \mathcal{N}\left(\boldsymbol{v}_k; 0, R\right); \tag{8.15}$$

$$\boldsymbol{u}_k \sim \mathcal{N}\left(\boldsymbol{v}_k; 0, Q\right); \;\; \boldsymbol{e}_k \sim \prod_{i=1}^{p} \mathcal{SN}(\boldsymbol{e}_k^i; \mu_i, \sigma_i, \delta_i) \tag{8.16}$$

Here the location, scale, and shape of the skew-normal distribution are represented by μ_i, σ_i and δ_i, respectively. The use of the skew-normal distribution complicates the filtering process due to the assumption made by the Kalman filter that measurement noise follows a Gaussian distribution. While it is possible to derive analytical expressions for the filtering and smoothing of skew-normal state space models, they are computationally infeasible to implement [216, 217].

To address this challenge, a hierarchical representation of the skew-normal distribution can be employed. Assuming that variable y follows a skew-normal distribution, it can be expressed as a linear combination of a Gaussian distribution e and a truncated Gaussian distributed variable z, as shown in (8.17). This is similar to the form given in [218] with an additional scaling factor σ. This approach allows for more effective handling of the skew-normal distribution within the filtering process.

$$y = \sigma\delta z + \sigma\sqrt{(1-\delta^2)}e \tag{8.17}$$

where $z \sim \mathcal{N}_+(\mu, 1)$ and $e \sim \mathcal{N}(0,1)$. Therefore,

$$\begin{aligned} p(y) &= \int_0^\infty p(y,z)\,dz \\ &= \int_0^\infty p(y|z)p(z)\,dz \\ &= \int_0^\infty \mathcal{N}\left(y;\sigma\delta z, \sigma^2(1-\delta^2)\right)\mathcal{N}_+\left(z;\mu,1\right)\,dz \\ &= \int_0^\infty \mathcal{N}\left(y;\sigma\delta z, \sigma^2(1-\delta^2)\right)\frac{\frac{1}{2\pi}\exp\left\{-\frac{1}{2}(z-\mu)^2\right\}}{(\phi(\infty)-\phi\left(-\mu\right))}\,dz \\ &= \frac{1}{\phi(\mu)}\int_0^\infty \mathcal{N}\left(y;\sigma\delta z, \sigma^2(1-\delta^2)\right)\mathcal{N}\left(z;\mu,1\right)\,dz \end{aligned}$$

Given the marginal Gaussian distribution for z and a conditional Gaussian distribution for y given z,

$$\begin{aligned} p(z) &= \mathcal{N}(z;m,v) \\ p(y|z) &= \mathcal{N}(y;az+b,c) \end{aligned}$$

the marginal distribution of y and the conditional distribution of z given y are determined by using Bayes' theorem [219].

$$\begin{aligned} p(y) &= \mathcal{N}(y;am+b,a^2v+c) \\ p(z|y) &= \mathcal{N}(z;m+k(y-am-b),(1-ka)v) \text{ where } k = \frac{av}{a^2v+c} \end{aligned}$$

Hence

$$\begin{aligned} p(y) &= \frac{1}{\phi(\mu)}\int_0^\infty \mathcal{N}\left(y;\sigma\delta\mu,\sigma^2\right)\mathcal{N}\left(z;\frac{\mu\sigma(1-\delta^2)+\delta y}{\sigma},(1-\delta^2)\right)\,dz \\ &= \frac{\mathcal{N}\left(y;\sigma\delta\mu,\sigma^2\right)}{\phi(\mu)}\int_0^\infty \mathcal{N}\left(z;\frac{\mu\sigma(1-\delta^2)+\delta y}{\sigma},(1-\delta^2)\right)\,dz \\ &= \frac{\mathcal{N}\left(y;\sigma\delta\mu,\sigma^2\right)}{\phi(\mu)}\,\phi\left(\frac{\mu\sigma(1-\delta^2)+\delta y}{\sigma\sqrt{(1-\delta^2)}}\right) \end{aligned}$$

The overall distribution is a product of a normal distribution multiplied by a CDF of a normal distribution. For $\delta \to 1$,

$$\lim_{\delta\to 1}\phi\left(\frac{\mu\sigma(1-\delta^2)+\delta y}{\sigma\sqrt{(1-\delta^2)}}\right) = \begin{cases} 1 & y > 0 \\ 0 & y < 0 \end{cases}$$

Thus, we can obtain a skew-normal distribution with only positive support in the limiting case as $\delta \to 1$, as shown in Fig. 8.4.

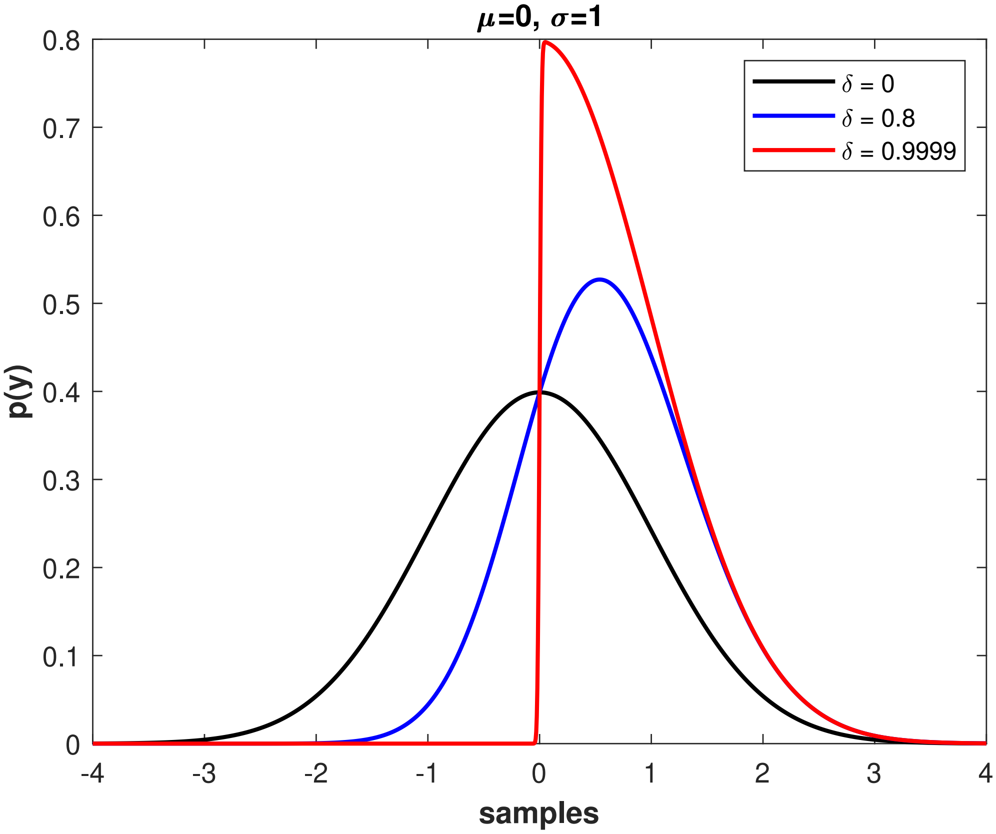


Figure 8.4: Density functions as $\delta \to 1$

Note: This formulation of constraints shares similarities with the concept of chance constraints as discussed in the literature on chance-constrained optimization [220]. In general, a linear chance constraint is represented as follows:

$$P[Nx_k <= f] >= 1 - \epsilon \tag{8.18}$$

In this equation, the parameter ϵ represents the probability to which the constraint is allowed to deviate from its desired condition. Notably, a resemblance emerges between the proposed approach and the chance constraints framework. In the proposed framework, Any value of δ that deviates from its limiting case introduces a finite probability less than zero, thus permitting the constraint to be violated with a probability equivalent to the area under the probability density function below zero (shaded red region in Fig. 8.5). While (8.18) specifies that Nx must be less than f, defining the constraint using a skew-normal distribution allows for the selection of the mode's position, resulting in a more adaptable framework for constraint definition, as shown in Fig. 8.5.

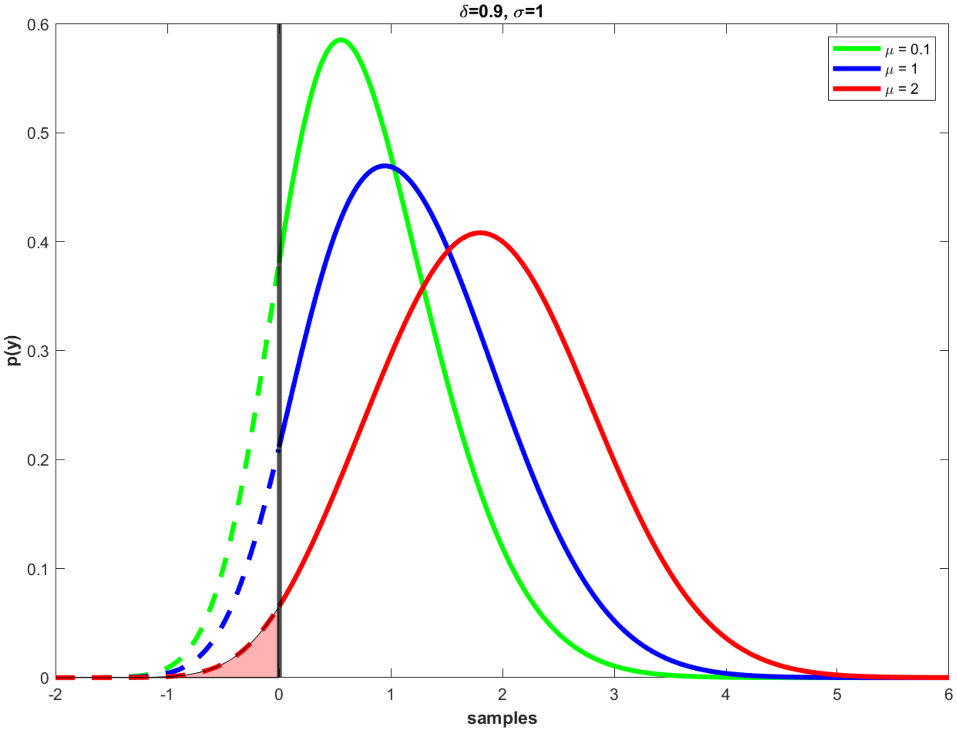


Figure 8.5: Locations of the distribution modes for various μ

Finally, the physics-informed slow feature model can be written as follows.

$$\boldsymbol{s}_k = A\boldsymbol{s}_{k-1} + \boldsymbol{w}_k; \tag{8.19}$$

$$\begin{bmatrix}\boldsymbol{x}_k \\ b \\ f\end{bmatrix} = \begin{bmatrix} C \\ MC \\ NC\end{bmatrix}\boldsymbol{s}_k + \begin{bmatrix}0 \\ M \\ N\end{bmatrix}\bar{\boldsymbol{x}} + \boldsymbol{v}_k; \tag{8.20}$$

where

$$\boldsymbol{v}_k \sim \mathcal{N}\left(\begin{bmatrix}0 \\ 0 \\ \boldsymbol{\sigma}\odot\boldsymbol{\delta}\odot\boldsymbol{z}_k\end{bmatrix}, \begin{bmatrix}R & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & \mathrm{diag}(\boldsymbol{\sigma}^2\odot(1-\boldsymbol{\delta}^2))\end{bmatrix}\right)$$

where $z_k^{(i)} \sim \mathcal{N}_+\left(z_k^{(i)};\mu_i,1\right)$, $R = \mathrm{diag}(r_1, r_2, \cdots, r_p)$ and $Q = \mathrm{diag}(q_1, q_2, \cdots, q_{n_1})$. Here $\odot$ represents element-wise multiplication. Additionally, the description of the various entities involved can be described as follows.

- **Measured Variables:** $\boldsymbol{x}_k \in \mathbb{R}^{p\times 1}\ \forall 1 \leq k \leq \mathcal{T}$
- **Latent Variables:** $\boldsymbol{s}_k \in \mathbb{R}^{m\times 1}$, $\boldsymbol{z}_k \in \mathbb{R}^{n_2\times 1}$
- **Parameters** (θ)**:**

$A \in \mathbb{R}^{m\times m}$	$C \in \mathbb{R}^{p\times m}$	$M \in \mathbb{R}^{n_1\times p}$
$N \in \mathbb{R}^{n_2\times p}$	$R \in \mathbb{R}^{p\times p}$	$Q \in \mathbb{R}^{n_1\times n_1}$
$\boldsymbol{\mu} \in \mathbb{R}^{n_2\times 1}$	$\boldsymbol{\sigma} \in \mathbb{R}^{n_2\times 1}$	$\boldsymbol{\delta} \in \mathbb{R}^{n_2\times 1}$

- **Physics information:** $\boldsymbol{b} \in \mathbb{R}^{n_1\times 1}$, $\boldsymbol{f} \in \mathbb{R}^{n_2\times 1}$

where m represents the latent variable dimension, p represents the observed variable dimension, and n_1 and n_2 represent the number of equality and inequality constraints, respectively. The sample size is denoted by $\mathcal{T}$. Finally, the complete data-likelihood can be expressed as follows.

$$\log p(X,S,Z|\theta) = \log p(\boldsymbol{s}_1) + \sum_{k=2}^{\mathcal{T}} \log p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1};A) + \sum_{k=1}^{\mathcal{T}} \log p(\boldsymbol{x}_k|\boldsymbol{s}_k;C,R)$$
$$+ \sum_{k=1}^{\mathcal{T}} \log p(b|\boldsymbol{s}_k;M,C,Q) + \sum_{k=1}^{\mathcal{T}} \log p(f|\boldsymbol{s}_k,\boldsymbol{z}_k;N,C,\boldsymbol{\sigma},\boldsymbol{\delta}) + \sum_{k=1}^{\mathcal{T}}\sum_{i=1}^{p} \log p(z_k^{(i)}|\mu_i)$$

The individual terms can be expanded as follows

$$\begin{aligned}
\log p(\boldsymbol{s}_1) &= -\frac{m}{2}\log 2\pi - \frac{1}{2}\boldsymbol{s}_1^T\boldsymbol{s}_1;\\
\log p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1},A) &= -\frac{m}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{m}\log|1-a_i^2| - \frac{1}{2}\sum_{i=1}^{m}\frac{(\boldsymbol{s}_k^i - a_i\boldsymbol{s}_{k-1}^i)^2}{1-a_i^2};\\
\log p(\boldsymbol{x}_k|\boldsymbol{s}_k;C,R) &= -\frac{p}{2}\log 2\pi - \frac{1}{2}\log|R| - \frac{1}{2}(\boldsymbol{x}_k - C\boldsymbol{s}_k)^T R^{-1}(\boldsymbol{x}_k - C\boldsymbol{s}_k);\\
\log p(\boldsymbol{b}|\boldsymbol{s}_k;C,M,Q) &= -\frac{n_1}{2}\log 2\pi - \frac{1}{2}\log|Q|\\
&\quad - \frac{1}{2}(\boldsymbol{b} - MC\boldsymbol{s}_k - M\bar{\boldsymbol{x}})^T Q^{-1}(\boldsymbol{b} - MC\boldsymbol{s}_k - M\bar{\boldsymbol{x}});
\end{aligned}$$

$$\log p(\boldsymbol{f}|\boldsymbol{s}_k, \boldsymbol{z}_k; N, C, \boldsymbol{\sigma}, \boldsymbol{\delta}) = -\frac{1}{2}\sum_{i=1}^{n_2} \log \left|\sigma_i^2(1-\delta_i^2)\right| \\ - \frac{n_2}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n_2} \frac{(\boldsymbol{f}_i - \boldsymbol{n}_i^T C\boldsymbol{s}_k - \boldsymbol{n}_i^T \bar{\boldsymbol{x}} - \sigma_i\delta_i z_k^{(i)})^2}{\sigma_i^2(1-\delta_i^2)};$$

$$\log p(\boldsymbol{z}_k|\boldsymbol{\mu}) = -\frac{n_2}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n_2}(z_k^{(i)} - \mu_i)^2 - \sum_{i=1}^{n_2}\log\phi(\mu_i);$$

where ϕ is the cumulative distribution function of standard normal distribution. All the matrices have been defined with suitable dimensions in the following manner for the purpose of facilitating representation.

$$C = \begin{bmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_p\end{bmatrix}^T;$$

The parameters are estimated using the expectation-maximization algorithm. Essentially, the update equations are obtained by taking the derivative of $\langle \log p(X, S, Z|\theta)\rangle$ with respect to θ and setting it equal to zero.

$$\frac{\partial \overbrace{\langle \log p(X, S, Z|\theta)\rangle}^{J(\theta)}}{\partial\theta} = 0 \tag{8.21}$$

where $\langle . \rangle$ represents the expectation operator with respect to the joint posterior distribution $p(S, Z|X; \theta^{\text{old}})$, and θ^{old} denotes the parameters from the previous iteration. However, it is to be noted that the true joint posterior distribution does not have a closed-form solution.

1. With respect to a_i

$$-(\mathcal{T}-1)a_i^3 + \left(\sum_{k=2}^{\mathcal{T}} \left\langle s_k^i s_{k-1}^i \right\rangle\right) a_i^2 + \left((\mathcal{T}-1) - \sum_{k=2}^{\mathcal{T}} \left\langle {s_k^i}^2 + {s_{k-1}^i}^2 \right\rangle\right) a_i \\ + \left(\sum_{k=2}^{\mathcal{T}} \left\langle s_k^i s_{k-1}^i \right\rangle\right) = 0$$

2. With respect to r_i

$$r_i = \frac{1}{N}\sum_{k=1}^{\mathcal{T}} \left({x_k^{(i)}}^2 + \boldsymbol{c}_i^T \langle \boldsymbol{s}_k \boldsymbol{s}_k^T\rangle \boldsymbol{c}_i - 2\boldsymbol{c}_i^T \langle \boldsymbol{s}_k\rangle \boldsymbol{x}_k^{(i)}\right)$$

3. With respect to Q

$$q_i = \frac{1}{N}\sum_{k=1}^{\mathcal{T}} \big(b_i^2 + \boldsymbol{m}_i^T C\langle \boldsymbol{s}_k\boldsymbol{s}_k^T\rangle C^T\boldsymbol{m}_i + \boldsymbol{m}_i^T\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T\boldsymbol{m}_i$$
$$-2\boldsymbol{m}_i^T C\langle\boldsymbol{s}_k\rangle b_i - 2\boldsymbol{m}_i^T\bar{\boldsymbol{x}} b_i + 2\boldsymbol{m}_i^T C\langle \boldsymbol{s}_k\rangle\bar{\boldsymbol{x}}^T\boldsymbol{m}_i\big)$$

4. With respect to C

$$C = \left(R^{-1} + M^TQ^{-1}M + \sum_{i=1}^{n_2}\frac{\boldsymbol{n}_i\boldsymbol{n}_i^T}{\sigma_i^2(1-\delta_i^2)}\right)^{-1}$$
$$\left(R^{-1}\sum_{k=1}^{\mathcal{T}}\boldsymbol{x}_k\langle\boldsymbol{s}_k^T\rangle + M^TQ^{-1}(\boldsymbol{b} - M\bar{\boldsymbol{x}})\sum_{k=1}^{\mathcal{T}}\langle\boldsymbol{s}_k^T\rangle + \sum_{i=1}^{n_2}\right.$$
$$\left.\sum_{k=1}^{\mathcal{T}}\frac{(f_i - \sigma_i\delta_i\langle z_k^{(i)}\rangle - n_i^T\bar{x})}{\sigma_i^2(1-\delta_i^2)}\boldsymbol{n}_i\langle\boldsymbol{s}_k^T\rangle\right)\left(\sum_{k=1}^{\mathcal{T}}\langle\boldsymbol{s}_k\boldsymbol{s}_k^T\rangle\right)^{-1}$$

5. The derivation of the update equation for M_{ij} is more complicated that will now be presented in a concise and coherent manner. We will utilize the chain rule, which will allow us to break down the derivation into manageable steps. From Matrix Cookbook [221]:

$$\frac{\partial J(\theta)}{\partial M_{ij}} = \mathrm{Tr}\left[\left(\frac{\partial J(\theta)}{\partial M}\right)^T\frac{\partial M}{\partial M_{ij}}\right]$$

First,

$$\frac{\partial J(\theta)}{\partial M} \;=\; 2\sum_{k=1}^{\mathcal{T}} Q^{-1}M\left\langle(C\boldsymbol{s}_k+\bar{\boldsymbol{x}})(C\boldsymbol{s}_k+\bar{\boldsymbol{x}})^T\right\rangle \;-\; 2\sum_{k=1}^{\mathcal{T}} Q^{-1}\boldsymbol{b}(C\langle\boldsymbol{s}_k\rangle \;+\; \bar{\boldsymbol{x}})^T$$

Therefore,

$$\frac{\partial J(\theta)}{\partial M_{ij}} = 0 \implies$$
$$\sum_{k=1}^{\mathcal{T}}\mathrm{Tr}\left[\left\langle(C\boldsymbol{s}_k+\bar{\boldsymbol{x}})(C\boldsymbol{s}_k+\bar{\boldsymbol{x}})^T\right\rangle M^TQ^{-1}J_{ij}\right] = \sum_{k=1}^{\mathcal{T}}\mathrm{Tr}\left[(C\langle\boldsymbol{s}_k\rangle+\bar{\boldsymbol{x}})\boldsymbol{b}^TQ^{-1}J_{ij}\right]$$

where J_{ij} is the single-entry matrix, 1 at (i, j) and zero elsewhere. The following two properties are used repeatedly for further simplifications.

$$\mathrm{Tr}[AJ_{ij}] = A_{ji} \tag{8.22}$$

$$(BCD)_{ji} = B_{j,:}CD_{:,i} \tag{8.23}$$

where $B_{j,:}$ and $D_{:,i}$ represent the j^{th} row and i^{th} column of matrices B and D, respectively. Therefore

$$\begin{aligned}
&\mathrm{Tr}[\langle(C\boldsymbol{s}_k + \bar{\boldsymbol{x}})(C\boldsymbol{s}_k + \bar{\boldsymbol{x}})^T\rangle M^T Q^{-1} J_{ij}] \\
&\qquad = \left(\left\langle(C\boldsymbol{s}_k + \bar{\boldsymbol{x}})(\boldsymbol{s}_k^T C^T + \bar{\boldsymbol{x}}^T)\right\rangle M^T Q^{-1}\right)_{ji} \\
&\qquad = \left\langle(C\boldsymbol{s}_k + \bar{\boldsymbol{x}})_{j,:}\ (\boldsymbol{s}_k^T C^T + \bar{\boldsymbol{x}}^T)\right\rangle M^T Q^{-1}_{:,i} \\
&\qquad = \left\langle(\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j)\ (\boldsymbol{s}_k^T C^T + \bar{\boldsymbol{x}}^T)\right\rangle \boldsymbol{m}_i Q^{-1}_{ii} \\
&\qquad = \left\langle (\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j) \left(\boldsymbol{s}_k^T \sum_{l=1}^{p} (\boldsymbol{c}_l\, M_{il}) + \sum_{l=1}^{p} (\bar{x}_l\, M_{il})\right)\right\rangle Q^{-1}_{ii} \\
&\qquad = \left\langle (\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j) \left(\sum_{\substack{l=1\\ l\neq j}}^{p} (\boldsymbol{s}_k^T \boldsymbol{c}_l + \bar{x}_l) M_{il} + (\boldsymbol{s}_k^T \boldsymbol{c}_j + \bar{x}_j) M_{ij}\right)\right\rangle Q^{-1}_{ii} \\
&\qquad = \left\langle (\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j) \left(\sum_{\substack{l=1\\ l\neq j}}^{p} (\boldsymbol{s}_k^T \boldsymbol{c}_l + \bar{x}_l) M_{il}\right)\right\rangle Q^{-1}_{ii} \\
&\qquad\qquad + \left\langle(\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j)(\boldsymbol{s}_k^T \boldsymbol{c}_j + \bar{x}_j)\right\rangle M_{ij} Q^{-1}_{ii}
\end{aligned} \tag{8.24}$$

Similarly

$$\mathrm{Tr}\left[(C\langle\boldsymbol{s}_k\rangle + \bar{\boldsymbol{x}})\,\boldsymbol{b}^T Q^{-1} J_{ij}\right] = \left(\boldsymbol{c}_j^T \langle\boldsymbol{s}_k\rangle + \bar{x}_j\right) b_i Q^{-1}_{ii} \tag{8.25}$$

Finally, the explicit equation for M_{ij} is presented below after combining all the terms and simplifying it further.

$$M_{ij} = \frac{\sum_{k=1}^{\mathcal{T}} \left\langle (\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j) \left(b_i - \sum_{\substack{l=1\\ l\neq j}}^{p} (\boldsymbol{s}_k^T \boldsymbol{c}_l + \bar{x}_l) M_{il}\right)\right\rangle}{\sum_{k=1}^{\mathcal{T}} \left\langle(\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j)(\boldsymbol{s}_k^T \boldsymbol{c}_j + \bar{x}_j)\right\rangle} \tag{8.26}$$

6. With respect to N_{ij}

$$N_{ij} = \frac{\mathrm{Term}_1 - \mathrm{Term}_2}{\sum_{k=1}^{\mathcal{T}} (\boldsymbol{c}_j^T \langle\boldsymbol{s}_k\rangle + \bar{x}_j)(\boldsymbol{s}_k^T \boldsymbol{c}_j + \bar{x}_j)}$$

where

$$\begin{aligned}
\mathrm{Term}_1 &= \sum_{k=1}^{\mathcal{T}} (\boldsymbol{c}_j^T \langle\boldsymbol{s}_k\rangle + \bar{x}_j)(f_i - \sigma_i \delta_i \langle z_k^{(i)}\rangle) \\
\mathrm{Term}_2 &= \sum_{k=1}^{\mathcal{T}} \left\langle (\boldsymbol{c}_j^T \boldsymbol{s}_k + \bar{x}_j) \sum_{\substack{l=1\\ l\neq j}}^{p} (\boldsymbol{s}_k^T \boldsymbol{c}_l + \bar{x}_l)\right\rangle N_{il}
\end{aligned}$$

7. With respect to σ_i

$$\frac{\partial}{\partial \sigma_i}\left(-\frac{1}{2}\sum_{k=1}^{\mathcal{T}}\sum_{i=1}^{n_2}\frac{\langle (f_i - \boldsymbol{n}_i^T C\boldsymbol{s}_k - \boldsymbol{n}_i^T\bar{\boldsymbol{x}} - \sigma_i\delta_i z_k^{(i)})^2\rangle}{\sigma_i^2(1-\delta_i^2)}\right) \\ -\frac{\partial}{\partial \sigma_i}\left(\frac{\mathcal{T}}{2}\sum_{i=1}^{n_2}\log\left|\sigma_i^2(1-\delta_i^2)\right|\right) = 0$$

$$\mathcal{T}(1-\delta_i^2)\sigma_i^2 + \left(\sum_{k=1}^{\mathcal{T}}\left(f_i - \boldsymbol{n}_i^T C\langle \boldsymbol{s}_k\rangle - \boldsymbol{n}_i^T\bar{\boldsymbol{x}}\right)\delta_i\langle z_k^{(i)}\rangle\right)\sigma_i \\ -\sum_{k=1}^{\mathcal{T}}\left\langle\left(f_i - \boldsymbol{n}_i^T C\boldsymbol{s}_k - \boldsymbol{n}_i^T\bar{\boldsymbol{x}}\right)^2\right\rangle = 0$$

8. With respect to μ_i

$$\sum_{k=1}^{\mathcal{T}}\left(\langle z_k^{(i)}\rangle - \mu_i\right) - \frac{\mathcal{T}}{\phi(\mu_i)\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\mu_i^2\right\} = 0$$

The solution to the above equation is obtained using a numerical solver because an explicit equation for μ_i is not obtained.

9. Since the true joint posterior is not tractable, a mean-field approximation is assumed as follows.

$$p\left(S, Z|X;\theta^{\text{old}}\right) \approx q\left(S|X;\theta^{\text{old}}\right) q\left(Z|X;\theta^{\text{old}}\right)$$

The approximate posterior $q\left(z_k^{(i)}|X;\theta^{\text{old}}\right)$ can be calculated using the variational Bayesian inference, as shown below.

$$\log q\left(z_k^{(i)}|X;\theta^{\text{old}}\right) \propto \langle \log p(X, S, Z|\theta^{\text{old}})\rangle_{q\left(S|X;\theta^{\text{old}}\right)} \\ \propto -\frac{1}{2}\frac{\left\langle\left(f_i - \boldsymbol{n}_i^T C\boldsymbol{s}_k - \boldsymbol{n}_i^T\bar{\boldsymbol{x}} - \sigma_i\delta_i z_k^{(i)}\right)^2\right\rangle}{\sigma_i^2(1-\delta_i^2)} - \frac{1}{2}(z_k^{(i)} - \mu_i)^2$$

Upon simplification, the following result is obtained.

$$q\left(z_k^{(i)}|X;\theta^{\text{old}}\right) = \mathcal{N}_+\left(z_k^{(i)};\mu_{z_k^{(i)}}, (1-\delta_i^2)\right)$$

where

$$\mu_{z_k^{(i)}} = (f_i - \boldsymbol{n}_i^T C\langle \boldsymbol{s}_k\rangle - \boldsymbol{n}_i^T\bar{\boldsymbol{x}})\frac{\delta_i}{\sigma_i} + \mu_i(1-\delta_i^2)$$

10. Similarly, the posterior distribution $q\left(\boldsymbol{s}_k|X;\theta^{\text{old}}\right)$ can be obtained as shown below.

$$\log q\left(\boldsymbol{s}_k|X;\theta^{\text{old}}\right) \propto \langle \log p(X,S,Z|\theta^{\text{old}})\rangle_{q\left(Z|X;\theta^{\text{old}}\right)}$$

A typical state estimation problem is represented by this scenario, where the mean and covariance of an approximate posterior distribution, which follows a Gaussian distribution, are estimated using the widely recognized equations of the Kalman filter and smoother. The equations [222] are avoided for brevity.

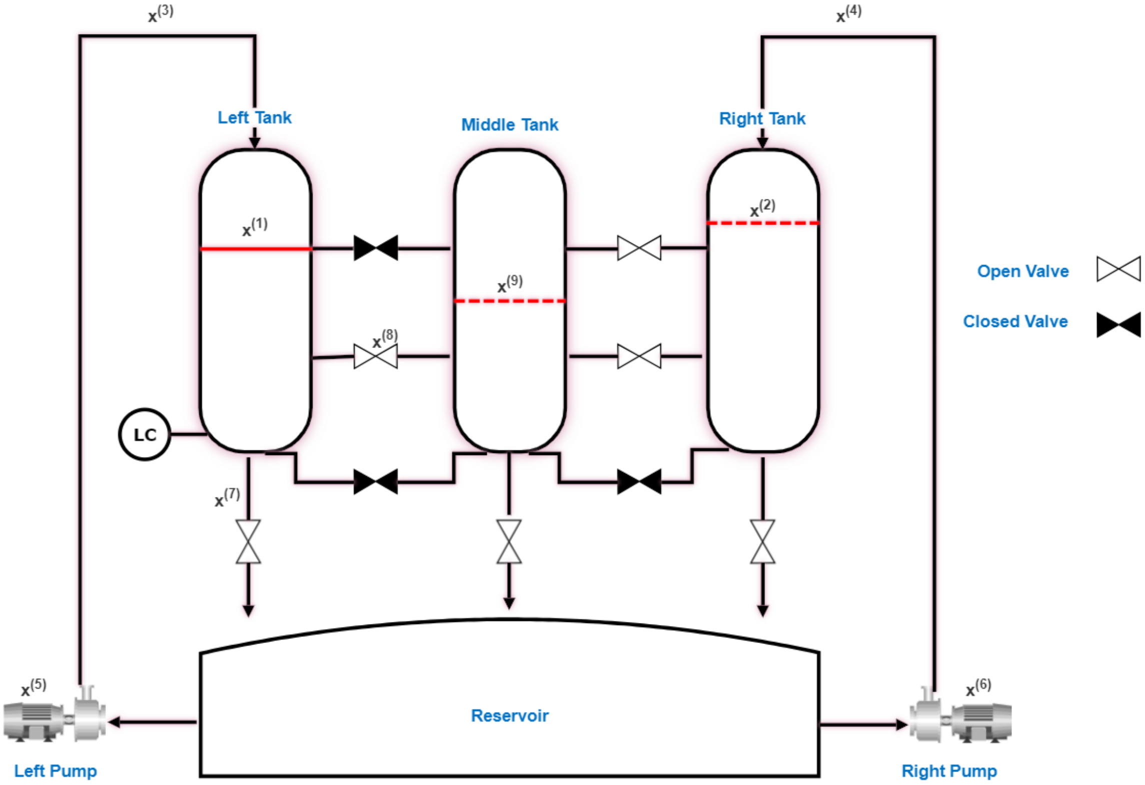


Figure 8.6: A Schematic of Hybrid Tank Pilot Plant

8.4 Experimental Validation

The efficiency of the proposed physics-informed probabilistic slow feature model is demonstrated in this section through the utilization of a pilot-scale experimental dataset for soft-sensor applications.

Apparatus: The Hybrid Tank Pilot Plant's schematic, depicted in Fig. 8.6, consists of three equally sized transparent tanks installed at the same level. The middle tank connects to both the right and left tanks through three connections at different levels. Each tank is equipped with a discharge pipe at the bottom, which empties into a reservoir on the lower level. Furthermore, overflow discharge pipes are installed within each tank. The pumps enable water flow into either the left or right tank, and various valves can be opened or closed, allowing for multiple flow paths, including into the middle tank. The flow rates are measured using flow sensors, while the levels of all three tanks are monitored using differential pressure (DP) sensors. Table 8.1 presents the description of various variables used in the subsequent case study.

Table 8.1: Process Variables description

s.no.	Variable	Symbol
1	Left tank level	$x^{(1)}$
2	Right tank level	$x^{(2)}$
3	Left flow-rate in	$x^{(3)}$
4	Right flow-rate in	$x^{(4)}$
5	Left pump speed	$x^{(5)}$
6	Right pump speed	$x^{(6)}$
7	Left flow-rate out	$x^{(7)}$
8	Intermediate flow-rate	$x^{(8)}$
9	Middle tank level	$x^{(9)}$

Process Design: The process has been designed to demonstrate the efficiency of the proposed algorithm. The equality constraint $x^{(3)} = x^{(7)} + x^{(8)}$ is achieved by maintaining the level in the left tank at the red solid line (set point) indicated in Fig. 5.1 using a PID controller. This equality information is integrated through the utilization of the M matrix, which is presented below.

$$M = \begin{bmatrix} 0 & NaN & 1 & 0 & 0 & 0 & NaN & -1 & 0 \end{bmatrix}; \;\; b = 0$$

Importantly, it should be noted that the knowledge at hand is only partial, and any missing information is denoted by NaN, reflecting a more realistic representation of the scenario. The proposed algorithm exhibits a distinct advantage in accurately estimating the missing physics using (8.26). Simultaneously, a pseudo-random binary input sequence is utilized for the right pump speed $x^{(6)}$ to enable fluctuations in the level of the middle tank between the top and middle connections. This deliberate design allows the middle tank level to influence the intermediate flow rate $x^{(8)}$, prompting the PID controller to adjust $x^{(3)}$ and maintain the left tank level $x^{(1)}$ at the specified set point. The collected dataset has been corrupted with additive Gaussian distributed noise. The focus of the current experimental case study is on the water level in the middle tank. It is essential for the middle tank level to remain below 25 units, as indicated by expert knowledge. To demonstrate the effectiveness of the proposed methodology, two regions of interest in the target variable, representing continuous sensory failures beyond the physical limit, are intentionally created, as illustrated in Fig. 8.7. The incorporation of expert information is achieved using the N matrix, as provided below.

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \;\; f = 25$$

The original dataset is divided into three distinct sets, with 1700 samples allocated for model training, 1000 samples for testing, and 300 samples for validation.

The algorithm proposed in equations (1)-(10) is iterated until the objective function $J(\theta)$ converges. The latent variable dimension is selected to maximize the corresponding $J(\theta)$. A comparison with state-of-the-art methods is conducted to demonstrate the superiority of the proposed algorithm. Notably, the predictions of the proposed model remain strictly within the specified threshold, unlike other models that do not integrate expert information. Fig. 8.8 depicts the extracted slow features that adhere to the equality constraint, while the corresponding recovered physics is presented below.

$$\hat{M} = \begin{bmatrix} 0 & 0.013 & 1 & 0 & 0 & 0 & -8.130 & -1 & 0 \end{bmatrix};$$

The estimated missing values of M are found to be closer to the true values, showcasing the effectiveness of the proposed algorithm. The proposed algorithm is compared with several competing algorithms, including ordinary least squares (OLS), PLS [122], SFR [223], and PSFA [55]. Results show that predictions based on the PI-PSFA features more closely align with the underlying truth (represented by a black curve), demonstrating superior predictive capability. Moreover, these predictions remain within the physical limit (indicated by the red line), highlighting the physics-informed model as superior to other approaches. Performance indices in Table 8.2,

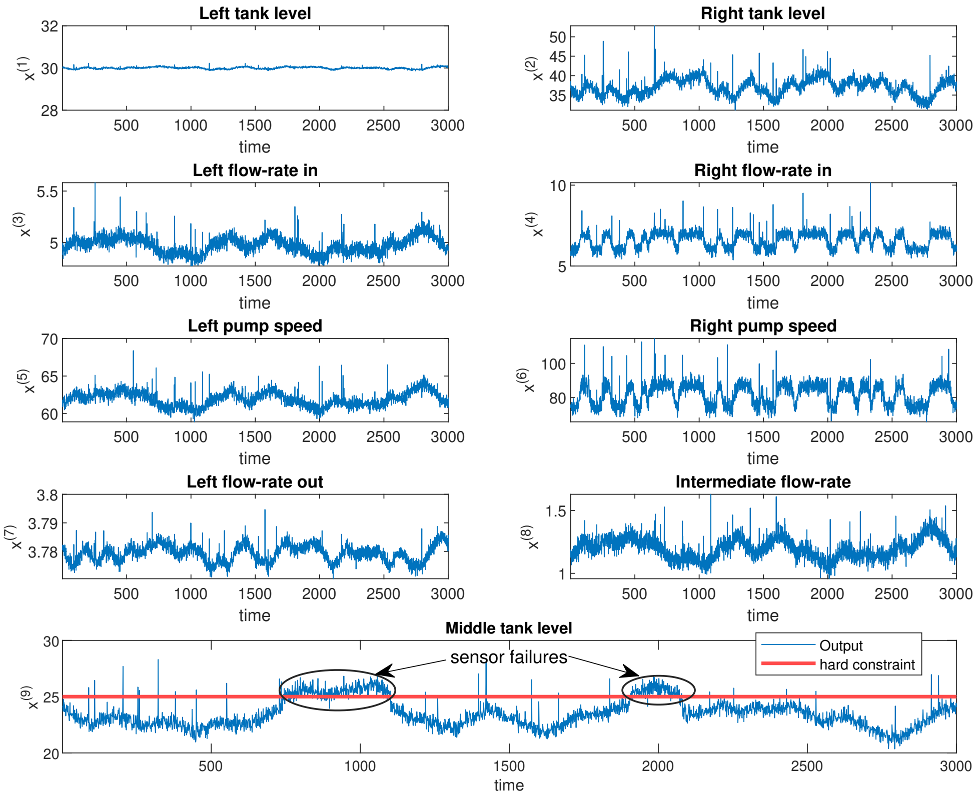


Figure 8.7: Input-Output variables

such as root mean square error (RMSE) and concordance correlation coefficient (ρ_c), are calculated using the test dataset predictions from each model. The proposed method achieves a significantly lower RMSE compared to other models, attributed to the integration of process knowledge.

Table 8.2: Performance metrics comparison

Method	OLS	PLS	SFR	PSFA	PI-PSFA
RMSE	0.635	0.607	0.619	0.485	0.254
ρ_c	0.671	0.667	0.676	0.783	0.874

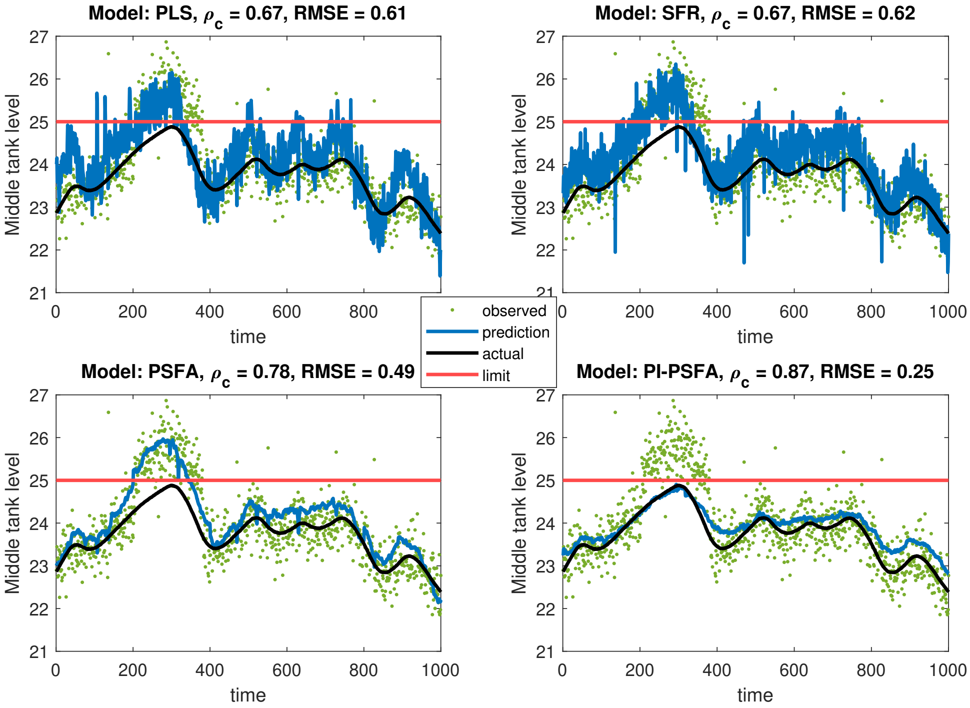


Figure 8.8: Time trend comparison

8.5 Conclusion

The current state of probabilistic feature extraction methods relies solely on data-based approaches. This chapter aims to enhance the reliability and consistency of inferences by introducing a novel probabilistic slow feature model that incorporates process knowledge. The model considers two types of process knowledge: linear static equality and linear static inequality relationships among the observed variables. By efficiently utilizing the available data, the proposed algorithm can estimate missing physics information effectively. This chapter provides the derivation and presentation of updated equations for all relevant parameters. To demonstrate its efficacy, the proposed model is applied to an experimental pilot plant case study. Furthermore, this model can be extended to incorporate other forms of expert information encompassing non-linearity and dynamics.

Chapter 9

Conclusions and Recommendations for Future Work

This chapter presents the conclusions drawn from our investigation of slow feature analysis. Moreover, based on the insights gained, we offer recommendations for future work in this domain. These suggestions aim to further advance the understanding of PSFA and explore its untapped potential.

9.1 Concluding Remarks

The central theme of this thesis has focused on probabilistic slow feature analysis, a technique employed for low-velocity feature extraction from a given set of measured variables that represent a specific process. Nevertheless, the existing framework may not fully address broader industrial challenges, including plant-wide oscillations, non-stationarities, non-linearities, strict physical limits, and skewed noise. This thesis approaches each challenge individually in separate chapters, presenting novel and intriguing models of PSFA tailored to effectively address these specific issues. The efficiency of each modified model is thoroughly assessed through real case studies, enabling a thorough evaluation of their practical applicability and performance.

1. Chapter 3 presents the first contribution of this thesis, which addresses the limitation of PSFA in extracting oscillatory features from noisy data due to the diagonality assumption of the state-transition matrix for uncorrelated features. This assumption makes it challenging to handle complex poles, which are required for such feature extraction. To overcome this, a block-diagonal structure

is proposed to accommodate complex poles effectively. An iterative algorithm is developed, leveraging the expectation maximization framework, to simultaneously estimate the states and parameters. Moreover, initial guesses for all involved parameters are provided using the deterministic SFA framework. The proposed model has been applied to identify the source of plant-wide oscillations.

2. In process data analysis, drift represents a significant non-stationary characteristic wherein the statistical properties of the data evolve over time. Factors such as machinery and equipment wear, accumulation of contaminants, and calibration drift contribute to the manifestation of non-stationary behaviors in process variables. However, the existing PSFA method assumes that the underlying patterns are solely slow features, characterized by a constant mean of 0 and variance of 1. Chapter 4 addresses the limitations of PSFA when dealing with drift-type non-stationary data. To effectively segregate the underlying trends, we propose the incorporation of an additional latent variable equation based on a drift-type random walk model. Moreover, we account for model uncertainty by introducing priors on the model parameters and subsequently derived their posterior distributions using the variational Bayesian framework. The effectiveness of the proposed model is demonstrated in its successful application to monitor fouling, which also exhibits non-stationary characteristics.

3. The constrained linear state-space model, PSFA, examined in prior chapters, lacks the ability to represent non-linear processes. To address this limitation, we turn our attention to this problem in Chapter 5, where we tackle the issue of non-linearity by incorporating neural networks. Specifically, we develop a novel neural network architecture capable of extracting slow oscillating patterns from non-linear data. This architecture utilizes a gated recurrent unit, which facilitates the flow of information across different time steps. By employing this approach, we enhance the model's capacity to capture dynamics in the nonlinear data.

4. In regular PSFA, the Gaussian distribution is commonly adopted to model the

measurement noise due to its compatibility with the standard Kalman filter for state estimation. Nevertheless, the limitations of this assumption arise when dealing with broader applications characterized by the presence of outliers and skewed values in the measurement noise. To address this concern, in Chapter 6, we propose the adoption of a skewed-t distribution as an alternative approach. However, this change introduces challenges in the state estimation process. To mitigate these challenges, we opt for the implementation of a Gaussian scale mixture representation of the skewed t-distribution. By employing this approach, the measurement noise continues to follow the form of Gaussian distribution, albeit with varying mean and covariance, accommodating the complexities associated with the presence of outliers and skewed values.

5. Chapter 6 introduces an advanced PSFA model capable of addressing asymmetric noise. Nevertheless, this model still exhibits two main drawbacks. First, it considers the model parameters as fixed unknown entities, thus rendering it incapable of handling model uncertainty. Additionally, the latent variable dimension is taken as the hyper-parameter, which is to be tuned. To deal with these issues in Chapter 7, each parameter is assigned its own prior distribution. As new data is acquired, the model updates its beliefs regarding these parameters using Bayes' theorem, which combines prior knowledge with the observed data's likelihood to obtain posterior probability distributions. Specifically, the emission matrix is assumed to follow a Laplace distribution, allowing for a sparse representation and automatic determination of the optimal latent dimension.

6. The physics-informed slow feature model, introduced in Chapter 8, enables the integration of expert information in a probabilistic manner. This chapter focuses on two types of physics laws: linear equality and inequality constraints. Unlike the data-based regular PSFA model, the integration of physics allows for more reliable inferences, particularly in cases of limited data length and calibration drifts beyond the physical limits. The efficacy of the proposed methodology is demonstrated through pilot plant hybrid tank experimental data.

9.2 Future Scope

1. **Physics-Informed Dynamic and Non-linear Constraints:** An extension of probabilistic slow feature analysis is considered, where the incorporation of physics-informed dynamic and non-linear constraints is explored. By extending the existing method to handle more complex and realistic physics laws, the model's applicability can be broadened to encompass a wider range of real-world problems. This entails formulating the slow feature model to be adaptive to dynamic changes and capable of accounting for non-linear relationships among variables. The implementation involves integrating expert knowledge of the underlying physics and their corresponding temporal dynamics into the probabilistic framework. The inclusion of dynamic and non-linear constraints can significantly enhance the interpretability and predictive capabilities of the model.

2. **Gaussian Process Regression for the Emission Equation as a Non-parametric Approach:** Another extension involves leveraging Gaussian process regression as a non-parametric approach for modeling the emission equation within the probabilistic slow feature analysis framework. This extension is motivated by the need to address scenarios where traditional parametric models may be inadequate to capture the underlying complexity of the data. By adopting Gaussian process regression, the emission equation gains the flexibility to adapt to different data distributions and non-linear relationships, inherently accommodating uncertainty in the modeling process. The application of a non-parametric approach also offers the advantage of capturing complex patterns and correlations present in the data, thereby enabling the extraction of more informative slow features.

3. **PSFA for Transfer Learning:** Develop adaptive probabilistic weights for the transfer of slow features from the source to the target domain. These weights can dynamically adjust based on the similarity between domains or the confidence in the transfer process, allowing for more accurate and efficient knowledge transfer. One can further generalize this extension to support multi-source transfer

learning, where information is transferred from multiple source domains to a single target domain. This can be valuable in scenarios where multiple related domains can collectively improve the target domain's performance.

4. **Slow feature-based reward shaping:** Reward shaping is a technique used in reinforcement learning to modify the reward signal to encourage desirable behavior. Slow features can be used to shape the reward signal by providing a more stable and informative representation of the state of the environment. This can help the agent learn more efficiently and robustly.

References

[1] N. F. Thornhill, “Finding the source of nonlinearity in a process with plant-wide oscillation,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 434–443, 2005.

[2] D. P. Solomatine and A. Ostfeld, “Data-driven modelling: some past experiences and new approaches,” *Journal of hydroinformatics*, vol. 10, no. 1, pp. 3–22, 2008.

[3] F. J. Montáns, F. Chinesta, R. Gómez-Bombarelli, and J. N. Kutz, “Data-driven modeling and learning in science and engineering,” *Comptes Rendus Mécanique*, vol. 347, no. 11, pp. 845–855, 2019.

[4] E. Kuhl, “Data-driven modeling of covid-19—lessons learned,” *Extreme Mechanics Letters*, vol. 40, p. 100921, 2020.

[5] Q. Sun and Z. Ge, “Gated Stacked Target-Related Autoencoder: A Novel Deep Feature Extraction and Layerwise Ensemble Method for Industrial Soft Sensor Application,” *IEEE Transactions on Cybernetics*, 2020.

[6] H. Chen, B. Jiang, S. X. Ding, and B. Huang, “Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] Q. Jiang, X. Yan, and B. Huang, “Deep discriminative representation learning for nonlinear process fault detection,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1410–1419, 2019.

[8] C. Zhao and B. Huang, “Incipient fault detection for complex industrial processes with stationary and nonstationary hybrid characteristics,” *Industrial & Engineering Chemistry Research*, vol. 57, no. 14, pp. 5045–5057, 2018.

[9] E. Naghoosi and B. Huang, “Wavelet transform based methodology for detection and characterization of multiple oscillations in nonstationary variables,” *Industrial & Engineering Chemistry Research*, vol. 56, no. 8, pp. 2083–2093, 2017.

[10] A. Tangirala, S. Shah, and N. Thornhill, “PSCMAP: A new tool for plant-wide oscillation detection,” *Journal of Process Control*, vol. 15, no. 8, pp. 931–941, 2005.

[11] A. K. Tangirala, J. Kanodia, and S. L. Shah, “Non-negative matrix factorization for detection and diagnosis of plantwide oscillations,” *Industrial & Engineering Chemistry Research*, vol. 46, no. 3, pp. 801–817, 2007.

[12] S. Selvanathan and A. Tangirala, “Diagnosis of Oscillations Due to Multiple Sources in Model-Based Control Loops Using Wavelet Transforms,” *The IUP Journal of Chemical Engineering*, vol. 1, no. 1, pp. 7–21, 2009.

[13] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch, “Detecting causality in complex ecosystems,” *science*, vol. 338, no. 6106, pp. 496–500, 2012.

[14] V. K. Puli and A. K. Tangirala, “Inferring Direct Causality from Noisy Data using Convergent Cross Mapping,” in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1523–1528, IEEE, 2019.

[15] S. Kathari and A. K. Tangirala, “A Novel Causality Method for Reconstruction of Process Topology in Multivariable LTI Dynamical Systems,” in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 199–204, IEEE, 2019.

[16] S. Kathari and A. K. Tangirala, "Efficient reconstruction of granger-causal networks in linear multivariable dynamical processes," *Industrial & Engineering Chemistry Research*, vol. 58, no. 26, pp. 11275–11294, 2019.

[17] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[18] M. A. Babyak, "What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models," *Psychosomatic medicine*, vol. 66, no. 3, pp. 411–421, 2004.

[19] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207. Springer, 2008.

[20] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, vol. 1168, p. 022022, IOP Publishing, 2019.

[21] L. Van Der Maaten, E. Postma, J. Van den Herik, *et al.*, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

[22] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *Ieee Access*, vol. 8, pp. 54776–54788, 2020.

[23] I. K. Fodor, "A survey of dimension reduction techniques," tech. rep., Lawrence Livermore National Lab., CA (US), 2002.

[24] A. Algaba, D. Ardia, K. Bluteau, S. Borms, and K. Boudt, "Econometrics meets sentiment: An overview of methodology and applications," *Journal of Economic Surveys*, vol. 34, no. 3, pp. 512–547, 2020.

[25] A. Ntakaris, G. Mirone, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Feature engineering for mid-price prediction with deep learning," *Ieee Access*, vol. 7, pp. 82390–82412, 2019.

[26] R. Bose, K. Samanta, and S. Chatterjee, "Cross-correlation based feature extraction from emg signals for classification of neuro-muscular diseases," in *2016*

International Conference on Intelligent Control Power and Instrumentation (ICICPI), pp. 241–245, IEEE, 2016.

[27] L. Hu and Z. Zhang, *EEG signal processing and feature extraction.* Springer, 2019.

[28] H. Das, B. Naik, and H. Behera, “Medical disease analysis using neuro-fuzzy with feature extraction model for classification,” *Informatics in Medicine Unlocked*, vol. 18, p. 100288, 2020.

[29] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, “Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,” *CIRP annals*, vol. 65, no. 1, pp. 417–420, 2016.

[30] Z. Ge, Z. Song, S. X. Ding, and B. Huang, “Data mining and analytics in the process industry: The role of machine learning,” *Ieee Access*, vol. 5, pp. 20590–20616, 2017.

[31] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, “Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted sae,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3235–3243, 2018.

[32] L. Guo, P. Wu, S. Lou, J. Gao, and Y. Liu, “A multi-feature extraction technique based on principal component analysis for nonlinear dynamic process monitoring,” *Journal of Process Control*, vol. 85, pp. 159–172, 2020.

[33] I. D. Apostolopoulos and M. A. Tzani, “Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with deep learning approach,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2022.

[34] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[35] Z. Ge, B. Huang, and Z. Song, “Mixture semisupervised principal component regression model and soft sensor application,” *AIChE Journal*, vol. 60, no. 2, pp. 533–545, 2014.

[36] L. Wiskott and T. J. Sejnowski, “Slow feature analysis: Unsupervised learning of invariances,” *Neural Computation*, vol. 14, no. 4, pp. 715–770, 2002.

[37] S. De Jong, “SIMPLS: an alternative approach to partial least squares regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, 1993.

[38] B. S. Dayal and J. F. MacGregor, “Recursive exponentially weighted PLS and its applications to adaptive control and prediction,” *Journal of Process Control*, vol. 7, no. 3, pp. 169–179, 1997.

[39] E. Zamprogna, M. Barolo, and D. E. Seborg, “Estimating product composition profiles in batch distillation via partial least squares regression,” *Control Engineering Practice*, vol. 12, no. 7, pp. 917–929, 2004.

[40] A. Tharwat, “Independent component analysis: An introduction,” *Applied Computing and Informatics*, 2020.

[41] C. Shang, F. Yang, X. Gao, X. Huang, J. A. Suykens, and D. Huang, “Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis,” *AIChE Journal*, vol. 61, no. 11, pp. 3666–3682, 2015.

[42] C. Shang, B. Huang, F. Yang, and D. Huang, “Slow feature analysis for monitoring and diagnosis of control performance,” *Journal of Process Control*, vol. 39, pp. 21–34, 2016.

[43] S. Zhang and C. Zhao, “Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3773–3783, 2018.

[44] J. Wang, Z. Zhao, and F. Liu, “Robust slow feature analysis for statistical process monitoring,” *Industrial & Engineering Chemistry Research*, vol. 59, no. 27, pp. 12504–12513, 2020.

[45] X. Ma, Y. Si, Z. Yuan, Y. Qin, and Y. Wang, “Multistep dynamic slow feature analysis for industrial process monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9535–9548, 2020.

[46] Y. Xu, M. Jia, and Z. Mao, “A novel auto-regressive dynamic slow feature analysis method for dynamic chemical process monitoring,” *Chemical Engineering Science*, vol. 248, p. 117236, 2022.

[47] P. Song and C. Zhao, “Slow down to go better: A survey on slow feature analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[48] N. Zhang, X. Tian, L. Cai, and X. Deng, “Process fault detection based on dynamic kernel slow feature analysis,” *Computers & Electrical Engineering*, vol. 41, pp. 9–17, 2015.

[49] V. K. Puli and B. Huang, “Variational bayesian approach to nonstationary and oscillatory slow feature analysis with applications in soft sensing and process monitoring,” *IEEE Transactions on Control Systems Technology*, vol. 31, no. 4, pp. 1708–1719, 2023.

[50] P. Jia and G. Chen, “Wind power icing fault diagnosis based on slow feature analysis and support vector machines,” in *2020 10th International Conference on Power and Energy Systems (ICPES)*, pp. 398–403, 2020.

[51] H. Zhang, C. Li, D. Li, Y. Zhang, and W. Peng, “Fault detection and diagnosis of the air handling unit via an enhanced kernel slow feature analysis approach considering the time-wise and batch-wise dynamics,” *Energy and Buildings*, vol. 253, p. 111467, 2021.

[52] Z. Zhang and D. Tao, “Slow feature analysis for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 436–450, 2012.

[53] S. Buchheit, A. C. Dzuranin, C. Hux, and M. E. Riley, "Data visualization in local accounting firms: is slow technology adoption rational?," *Current Issues in Auditing*, vol. 14, no. 2, pp. A15–A24, 2020.

[54] R. Turner and M. Sahani, "A maximum-likelihood interpretation for slow feature analysis," *Neural Computation*, vol. 19, no. 4, pp. 1022–1038, 2007.

[55] C. Shang, B. Huang, F. Yang, and D. Huang, "Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling," *AIChE Journal*, vol. 61, no. 12, pp. 4126–4139, 2015.

[56] L. Fan, H. Kodamana, and B. Huang, "Identification of robust probabilistic slow feature regression model for process data contaminated with outliers," *Chemometrics and Intelligent Laboratory Systems*, vol. 173, pp. 1–13, 2018.

[57] G. Casella and R. L. Berger, *Statistical inference. 2nd Edition.* Duxbury Press, Pacific Grove, 2022.

[58] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions.* Wiley, 1994.

[59] L. Wasserman, *All of statistics: a concise course in statistical inference*, vol. 26. Springer, 2004.

[60] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.

[61] R. J. Hathaway, "A constrained formulation of maximum-likelihood estimation for normal mixture distributions," *The Annals of Statistics*, vol. 13, no. 2, pp. 795–800, 1985.

[62] N. Matthews, GB* & Crowther, "A maximum likelihood estimation procedure when modelling in terms of constraints," *South African Statistical Journal*, vol. 29, no. 1, pp. 29–50, 1995.

[63] T. J. Moore, B. M. Sadler, and R. J. Kozick, "Maximum-likelihood estimation, the cramér–rao bound, and the method of scoring with parameter constraints," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 895–908, 2008.

[64] C. M. Bishop, “Latent variable models,” in *Learning in graphical models*, pp. 371–403, Springer, 1998.

[65] Z. Ge and X. Chen, “Dynamic probabilistic latent variable model for process data modeling and regression application,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 323–331, 2017.

[66] Z. Ge, “Process data analytics via probabilistic latent variable models: A tutorial review,” *Industrial & Engineering Chemistry Research*, vol. 57, no. 38, pp. 12646–12661, 2018.

[67] B. Shen, L. Yao, and Z. Ge, “Nonlinear probabilistic latent variable regression models for soft sensor application: From shallow to deep structure,” *Control Engineering Practice*, vol. 94, p. 104198, 2020.

[68] C. Soize, “A comprehensive overview of a non-parametric probabilistic approach of model uncertainties for predictive models in structural dynamics,” *Journal of sound and vibration*, vol. 288, no. 3, pp. 623–652, 2005.

[69] K. Miki, M. Panesi, E. E. Prudencio, and S. Prudhomme, “Probabilistic models and uncertainty quantification for the ionization reaction rate of atomic nitrogen,” *Journal of Computational Physics*, vol. 231, no. 9, pp. 3871–3886, 2012.

[70] I. Meedeniya, I. Moser, A. Aleti, and L. Grunske, “Evaluating probabilistic models with uncertain model parameters,” *Software & Systems Modeling*, vol. 13, pp. 1395–1415, 2014.

[71] G. W. Taylor, G. E. Hinton, and S. Roweis, “Modeling human motion using binary latent variables,” *Advances in neural information processing systems*, vol. 19, 2006.

[72] T. J. Green and V. Tannen, “Models for incomplete and probabilistic information,” in *International Conference on Extending Database Technology*, pp. 278–296, Springer, 2006.

[73] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West, *et al.*, “The variational bayesian em algorithm for incomplete data:

with application to scoring graphical model structures," *Bayesian statistics*, vol. 7, no. 453-464, p. 210, 2003.

[74] S. Polasky, J. D. Camm, A. R. Solow, B. Csuti, D. White, and R. Ding, "Choosing reserve networks with incomplete species information," *Biological Conservation*, vol. 94, no. 1, pp. 1–10, 2000.

[75] C. Luo, T. Li, and Y. Yao, "Dynamic probabilistic rough sets with incomplete data," *Information Sciences*, vol. 417, pp. 39–54, 2017.

[76] D. Barber, *Bayesian reasoning and machine learning.* Cambridge University Press, 2012.

[77] M. J. Beal, *Variational algorithms for approximate Bayesian inference.* University of London, University College London (United Kingdom), 2003.

[78] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

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

[80] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[81] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[82] C. J. Wu, "On the convergence properties of the em algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[83] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[84] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[85] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state–space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, pp. 61–89, 2010.

[86] D. Barber, A. T. Cemgil, and S. Chiappa, *Bayesian time series models.* Cambridge University Press, 2011.

[87] W. J. Miller, "Lecture notes on advanced stochastic modeling," *Duke University, Durham, NC*, 2016.

[88] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[89] R. Raveendran, H. Kodamana, and B. Huang, "Process monitoring using a generalized probabilistic linear latent variable model," *Automatica*, vol. 96, pp. 73–83, 2018.

[90] Z. Ge, B. Huang, and Z. Song, "Nonlinear semisupervised principal component regression for soft sensor modeling and its mixture form," *Journal of Chemometrics*, vol. 28, no. 11, pp. 793–804, 2014.

[91] T. Michaeli, W. Wang, and K. Livescu, "Nonparametric canonical correlation analysis," in *International conference on machine learning*, pp. 1967–1976, PMLR, 2016.

[92] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[93] M. G. Gustafsson, "A probabilistic derivation of the partial least-squares algorithm," *Journal of Chemical Information and Computer Sciences*, vol. 41, no. 2, pp. 288–294, 2001.

[94] C. F. Beckmann and S. M. Smith, "Probabilistic independent component analysis for functional magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 137–152, 2004.

[95] F. Guo, C. Shang, B. Huang, K. Wang, F. Yang, and D. Huang, “Monitoring of operating point and process dynamics via probabilistic slow feature analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 115–125, 2016.

[96] Y. Qin, C. Zhao, and B. Huang, “A new soft-sensor algorithm with concurrent consideration of slowness and quality interpretation for dynamic chemical process,” *Chemical Engineering Science*, vol. 199, pp. 28–39, 2019.

[97] R. Chiplunkar and B. Huang, “Output relevant slow feature extraction using partial least squares,” *Chemometrics and Intelligent Laboratory Systems*, vol. 191, pp. 148–157, 2019.

[98] L. Fan, H. Kodamana, and B. Huang, “Semi-supervised dynamic latent variable modeling: I/O probabilistic slow feature analysis approach,” *AIChE Journal*, vol. 65, no. 3, pp. 964–979, 2019.

[99] Y. Ma and B. Huang, “Bayesian learning for dynamic feature extraction with application in soft sensing,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7171–7180, 2017.

[100] Y. Ma and B. Huang, “Extracting dynamic features with switching models for process data analytics and application in soft sensing,” *AIChE Journal*, vol. 64, no. 6, pp. 2037–2051, 2018.

[101] K. Zhong, D. Ma, and M. Han, “Distributed dynamic process monitoring based on dynamic slow feature analysis with minimal redundancy maximal relevance,” *Control Engineering Practice*, vol. 104, p. 104627, 2020.

[102] C. Zhao and B. Huang, “A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis,” *AIChE Journal*, vol. 64, no. 5, pp. 1662–1681, 2018.

[103] D. Scott, C. Shang, B. Huang, and D. Huang, “A holistic probabilistic framework for monitoring nonstationary dynamic industrial processes,” *IEEE Transactions on Control Systems Technology*, 2020.

[104] M. S. Choudhury, S. L. Shah, N. F. Thornhill, and D. S. Shook, "Automatic detection and quantification of stiction in control valves," *Control Engineering Practice*, vol. 14, no. 12, pp. 1395–1412, 2006.

[105] H. Jiang, M. S. Choudhury, and S. L. Shah, "Detection and diagnosis of plant-wide oscillations from industrial data using the spectral envelope method," *Journal of Process Control*, vol. 17, no. 2, pp. 143–155, 2007.

[106] M. J. Kaminski and K. J. Blinowska, "A new method of the description of the information flow in the brain structures," *Biological Cybernetics*, vol. 65, no. 3, pp. 203–210, 1991.

[107] L. A. Baccalá and K. Sameshima, "Partial directed coherence: a new concept in neural structure determination," *Biological Cybernetics*, vol. 84, no. 6, pp. 463–474, 2001.

[108] D. Marinazzo, M. Pellicoro, and S. Stramaglia, "Kernel-Granger causality and the analysis of dynamical networks," *Physical Review E*, vol. 77, no. 5, p. 056215, 2008.

[109] R. Kannan and A. K. Tangirala, "Correntropy-based partial directed coherence for testing multivariate Granger causality in nonlinear processes," *Physical Review E*, vol. 89, no. 6, p. 062144, 2014.

[110] L. Fortuna, S. Graziani, A. Rizzo, M. G. Xibilia, *et al.*, *Soft sensors for monitoring and control of industrial processes*, vol. 22. Springer, 2007.

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

[112] S. Khatibisepehr, B. Huang, and S. Khare, "Design of inferential sensors in the process industry: A review of Bayesian methods," *Journal of Process Control*, vol. 23, no. 10, pp. 1575–1596, 2013.

[113] F. A. Souza, R. Araújo, T. Matias, and J. Mendes, "A multilayer-perceptron based method for variable selection in soft sensor design," *Journal of Process Control*, vol. 23, no. 10, pp. 1371–1378, 2013.

[114] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 30, no. 1, pp. 179–196, 1995.

[115] M. Kano, K. Miyazaki, S. Hasebe, and I. Hashimoto, "Inferential control system of distillation compositions using dynamic partial least squares regression," *Journal of Process Control*, vol. 10, no. 2-3, pp. 157–166, 2000.

[116] M. Bauer, A. Horch, L. Xie, M. Jelali, and N. Thornhill, "The current state of control loop performance monitoring–a survey of application in industry," *Journal of Process Control*, vol. 38, pp. 1–10, 2016.

[117] M. Bauer, L. Auret, D. Le Roux, and V. Aharonson, "An industrial PID data repository for control loop performance monitoring (CPM)," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 823–828, 2018.

[118] N. F. Thornhill, S. L. Shah, B. Huang, and A. Vishnubhotla, "Spectral principal component analysis of dynamic process data," *Control Engineering Practice*, vol. 10, no. 8, pp. 833–846, 2002.

[119] N. Thornhill, "Locating the source of a disturbance," in *Process control performance assessment*, pp. 199–225, Springer, 2007.

[120] X. Gao, X. Wang, D. Tao, and X. Li, "Supervised Gaussian process latent variable model for dimensionality reduction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 425–434, 2010.

[121] X. Jiang, J. Gao, T. Wang, and L. Zheng, "Supervised latent linear Gaussian process latent variable model for dimensionality reduction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 6, pp. 1620–1632, 2012.

[122] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica chimica acta*, vol. 185, pp. 1–17, 1986.

[123] Y. Dong and S. J. Qin, "A novel dynamic PCA algorithm for dynamic data modeling and process monitoring," *Journal of Process Control*, vol. 67, pp. 1–11, 2018.

[124] Y. Dong and S. J. Qin, "Dynamic latent variable analytics for process operations and control," *Computers & Chemical Engineering*, vol. 114, pp. 69–80, 2018.

[125] Y. Dong, Y. Liu, and S. J. Qin, "Efficient dynamic latent variable analysis for high-dimensional time series data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4068–4076, 2019.

[126] C. Jiang, W. Zhong, Z. Li, X. Peng, and M. Yang, "Real-time semisupervised predictive modeling strategy for industrial continuous catalytic reforming process with incomplete data using slow feature analysis," *Industrial & Engineering Chemistry Research*, vol. 58, no. 37, pp. 17406–17423, 2019.

[127] X. Gao, C. Shang, F. Yang, and D. Huang, "Detecting and isolating plant-wide oscillations via slow feature analysis," in *2015 American Control Conference (ACC)*, pp. 906–911, IEEE, 2015.

[128] L. Fan, *Robust Latent Variable Modeling Using Probabilistic Slow Feature Analysis*. PhD thesis, University of Alberta, 2020.

[129] V. K. Puli, R. Raveendran, and B. Huang, "Complex probabilistic slow feature extraction with applications in process data analytics," *Computers & Chemical Engineering*, vol. 154, p. 107456, 2021.

[130] R. Salles, K. Belloze, F. Porto, P. H. Gonzalez, and E. Ogasawara, "Nonstationary time series transformation methods: An experimental review," *Knowledge-Based Systems*, vol. 164, pp. 274–291, 2019.

[131] D. Scott, C. Shang, B. Huang, and D. Huang, "A Holistic Probabilistic Framework for Monitoring Nonstationary Dynamic Industrial Processes," *IEEE*

Transactions on Control Systems Technology, vol. 29, no. 5, pp. 2239–2246, 2021.

[132] S. J. Roberts and W. D. Penny, "Variational Bayes for generalized autoregressive models," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2245–2257, 2002.

[133] N. Nasios and A. G. Bors, "Variational learning for Gaussian mixture models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 4, pp. 849–862, 2006.

[134] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, "A novel semi-supervised sparse Bayesian regression based on variational inference for industrial datasets with incomplete outputs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4773–4786, 2018.

[135] C. W. Fox and S. J. Roberts, "A tutorial on variational Bayesian inference," *Artificial Intelligence Review*, vol. 38, no. 2, pp. 85–95, 2012.

[136] J. Paisley, D. Blei, and M. Jordan, "Variational Bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*, 2012.

[137] R. Raveendran and B. Huang, "Variational Bayesian approach for causality and contemporaneous correlation features inference in industrial process data," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2580–2590, 2018.

[138] Y. Ma, S. Kwak, L. Fan, and B. Huang, "A variational Bayesian approach to modelling with random time-varying time delays," in *2018 Annual American Control Conference (ACC)*, pp. 5914–5919, IEEE, 2018.

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

[140] Y. Cao, N. M. Jan, B. Huang, M. Fang, Y. Wang, and W. Gui, "Multimodal process monitoring based on variational Bayesian PCA and Kullback-Leibler

divergence between mixture models," *Chemometrics and Intelligent Laboratory Systems*, vol. 210, p. 104230, 2021.

[141] Y. Lu, B. Huang, and S. Khatibisepehr, "A variational Bayesian approach to robust identification of switched ARX models," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3195–3208, 2015.

[142] L. Zuo, Y. Shi, and W. Yan, "Dynamic coverage control in a time-varying environment using Bayesian prediction," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 354–362, 2017.

[143] Z. Yang, L. Yao, and Z. Ge, "Streaming parallel variational Bayesian supervised factor analysis for adaptive soft sensor modeling with big process data," *Journal of Process Control*, vol. 85, pp. 52–64, 2020.

[144] W. Shao, Z. Ge, and Z. Song, "Semisupervised Bayesian Gaussian mixture models for non-Gaussian soft sensor," *IEEE Transactions on Cybernetics*, 2019.

[145] J. Li, F. Deng, and J. Chen, "A fast distributed variational Bayesian filtering for multisensor LTV system with non-Gaussian noise," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2431–2443, 2018.

[146] J. Xie, B. Huang, and S. Dubljevic, "Transfer Learning for Dynamic Feature Extraction Using Variational Bayesian Inference," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[147] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

[148] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[149] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

[150] D. Barber and S. Chiappa, "Unified inference for variational Bayesian linear Gaussian state-space models," *Advances in Neural Information Processing Systems*, vol. 19, 2006.

[151] M. Kreider and G. White, "Thermal performance degradation and heat-transfer fouling," in *Steam Generators for Nuclear Power Plants*, pp. 365–404, Elsevier, 2017.

[152] J. Taborek, "Predictive methods for fouling behavior," *Chem. Eng. Prog. Chem. Eng. Prog.*, vol. 68, no. 7, pp. 69–78, 1972.

[153] S. M. Alsadaie and I. M. Mujtaba, "Dynamic modelling of Heat Exchanger fouling in multistage flash (MSF) desalination," *Desalination*, vol. 409, pp. 47–65, 2017.

[154] S. Kwak, Y. Ma, and B. Huang, "Extracting nonstationary features for process data analytics and application in fouling detection," *Computers & Chemical Engineering*, vol. 135, p. 106762, 2020.

[155] R. A. Johnson, D. W. Wichern, *et al.*, *Applied multivariate statistical analysis*, vol. 6. Pearson London, UK:, 2014.

[156] N. Nasios and A. Bors, "Variational learning for Gaussian mixture models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 4, pp. 849–862, 2006.

[157] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, "A Novel Semi-Supervised Sparse Bayesian Regression Based on Variational Inference for Industrial Datasets With Incomplete Outputs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4773–4786, 2020.

[158] L. Zafeiriou, M. A. Nicolaou, S. Zafeiriou, S. Nikitidis, and M. Pantic, "Probabilistic Slow Features for Behavior Analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 1034–1048, 2015.

[159] J.-T. Chien and H.-L. Hsieh, “Nonstationary Source Separation Using Sequential and Variational Bayesian Learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 681–694, 2013.

[160] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, “Stacked Convolutional Denoising Auto-Encoders for Feature Representation,” *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1017–1027, 2016.

[161] Śmieja, Marek and Wołczyk, Maciej and Tabor, Jacek and Geiger, Bernhard C, “Segma: Semi-supervised gaussian mixture autoencoder,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 3930–3941, 2020.

[162] R. Chiplunkar and B. Huang, “Siamese Neural Network-Based Supervised Slow Feature Extraction for Soft Sensor Application,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8953–8962, 2020.

[163] Q. Xie, P. Zhang, B. Yu, and J. Choi, “Semisupervised Training of Deep Generative Models for High-Dimensional Anomaly Detection,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2444–2453, 2021.

[164] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[165] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *arXiv preprint arXiv:1906.02691*, 2019.

[166] R. G. Krishnan, U. Shalit, and D. Sontag, “Deep Kalman Filters,” *arXiv preprint arXiv:1511.05121*, 2015.

[167] R. Krishnan, U. Shalit, and D. Sontag, “Structured Inference Networks for Nonlinear State Space Models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[168] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, “Dynamical Variational Autoencoders: A Comprehensive Review,” *arXiv preprint arXiv:2008.12595*, 2020.

[169] C. Jiang, Y. Lu, W. Zhong, B. Huang, W. Song, D. Tan, and F. Qian, "Deep Bayesian Slow Feature Extraction with Application to Industrial Inferential Modeling," *IEEE Transactions on Industrial Informatics*, 2021.

[170] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[171] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-Supervised Learning with Deep Generative Models," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[172] C. Wu, F. Wu, S. Wu, Z. Yuan, J. Liu, and Y. Huang, "Semi-Supervised Dimensional Sentiment Analysis with Variational Autoencoder," *Knowledge-Based Systems*, vol. 165, pp. 30–39, 2019.

[173] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[174] P. Song and C. Zhao, "Slow down to go better: A survey on slow feature analysis," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022.

[175] R. Turner and M. Sahani, "A maximum-likelihood interpretation for slow feature analysis," *Neural computation*, vol. 19, no. 4, pp. 1022–1038, 2007.

[176] F. Guo, C. Shang, B. Huang, K. Wang, F. Yang, and D. Huang, "Monitoring of operating point and process dynamics via probabilistic slow feature analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 115–125, 2016.

[177] C. Shang, B. Huang, Y. Lu, F. Yang, and D. Huang, "Dynamic modeling of gross errors via probabilistic slow feature analysis applied to a mining slurry preparation process," in *IFAC-PapersOnLine*, vol. 49, pp. 25–30, 2016.

[178] D. Scott, C. Shang, B. Huang, and D. Huang, "A holistic probabilistic framework for monitoring nonstationary dynamic industrial processes," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2239–2246, 2021.

[179] C. Jiang, Y. Lu, W. Zhong, B. Huang, W. Song, D. Tan, and F. Qian, "Deep Bayesian Slow Feature Extraction with Application to Industrial Inferential Modeling," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[180] V. K. Puli and B. Huang, "Nonlinear Semi-supervised Inference Networks for the Extraction of Slow Oscillating Features," *Techrxiv*, 10 2022.

[181] M. Lucke, M. Chioua, and N. F. Thornhill, "From oscillatory to non-oscillatory disturbances: A comparative review of root cause analysis methods," *Journal of Process Control*, vol. 113, pp. 42–67, 2022.

[182] B. Bahrami, S. Mohsenpour, H. R. Shamshiri Noghabi, N. Hemmati, and A. Tabzar, "Estimation of flow rates of individual phases in an oil-gas-water multiphase flow system using neural network approach and pressure signal analysis," *Flow Measurement and Instrumentation*, vol. 66, pp. 28–36, 2019.

[183] C. Xu, S. Zhao, Y. Ma, B. Huang, and F. Liu, "Robust filter design for asymmetric measurement noise using variational bayesian inference," *IET Control Theory & Applications*, vol. 13, no. 11, pp. 1656–1664, 2019.

[184] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, and J. A. Chambers, "Robust kalman filters based on gaussian scale mixture distributions with application to target tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 10, pp. 2082–2096, 2017.

[185] H. Nurminen, T. Ardeshiri, R. Piche, and F. Gustafsson, "Robust inference for state-space models with skewed measurement noise," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, 2015.

[186] X. Kong, X. Jiang, B. Zhang, J. Yuan, and Z. Ge, "Latent Variable Models in the Era of Industrial Big Data: Extension and Beyond," *Annual Reviews in Control*, vol. 54, pp. 167–199, 2022.

[187] W. Yu, M. Wu, B. Huang, and C. Lu, "A Generalized Probabilistic Monitoring Model with both Random and Sequential Data," *Automatica*, vol. 144, 2022.

[188] C. Archambeau and F. Bach, “Sparse Probabilistic Projections,” *Advances in neural information processing systems*, vol. 21, 2008.

[189] Y. Guan and J. Dy, “Sparse Probabilistic Principal Component Analysis,” in *Artificial Intelligence and Statistics*, pp. 185–192, PMLR, 2009.

[190] J. Zeng, K. Liu, W. Huang, and J. Liang, “Sparse Probabilistic Principal Component Analysis Model for Plant-wide Process Monitoring,” *Korean Journal of Chemical Engineering*, vol. 34, no. 8, pp. 2135–2146, 2017.

[191] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, “Skew-t Filter and Smoother With Improved Covariance Matrix Approximation,” *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5618–5633, 2018.

[192] M. Bai, Y. Huang, B. Chen, and Y. Zhang, “A Novel Robust Kalman Filtering Framework Based on Normal-Skew Mixture Distribution,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 11, pp. 6789–6805, 2022.

[193] S. X. Lee, K. L. Leemaqz, and G. J. McLachlan, “A Block EM Algorithm for Multivariate Skew Normal and Skew t -Mixture Models,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5581–5591, 2018.

[194] P. Yun, P. Wu, S. He, and X. Li, “Robust Kalman Filter with Fading Factor Under State Transition Model Mismatch and Outliers Interference,” *Circuits, Systems, and Signal Processing*, vol. 40, no. 5, pp. 2443–2463, 2021.

[195] C. Zhao, W. Wang, C. Tian, and Y. Sun, “Fine-Scale Modeling and Monitoring of Wide-Range Nonstationary Batch Processes With Dynamic Analytics,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8808–8818, 2021.

[196] C. Shang, F. Yang, B. Huang, and D. Huang, “Recursive Slow Feature Analysis for Adaptive Monitoring of Industrial Processes,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8895–8905, 2018.

[197] J. Zhang, D. Zhou, M. Chen, and X. Hong, “Continual Learning-Based Probabilistic Slow Feature Analysis for Monitoring Multimode Nonstationary Processes,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2022.

[198] V. K. Puli, R. Chiplunkar, and B. Huang, “Robust complex probabilistic slow feature analysis in the presence of skewed measurement noise,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10947–10952, 2023. 22nd IFAC World Congress.

[199] D. Ostwald, E. Kirilina, L. Starke, and F. Blankenburg, “A Tutorial on Variational Bayes for Latent Linear Stochastic Time-Series Models,” *Journal of mathematical psychology*, vol. 60, pp. 1–19, 2014.

[200] T.-I. Lin, “Robust Mixture Modeling using Multivariate Skew t-Distributions,” *Statistics and Computing*, vol. 20, no. 3, pp. 343–356, 2010.

[201] K. Lange and J. S. Sinsheimer, “Normal/Independent Distributions and their Applications in Robust Regression,” *Journal of Computational and Graphical Statistics*, vol. 2, no. 2, pp. 175–198, 1993.

[202] Z. Ge, Z. Song, S. X. Ding, and B. Huang, “Data Mining and Analytics in the Process Industry: The Role of Machine Learning,” *IEEE Access*, vol. 5, pp. 20590–20616, 2017.

[203] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-Informed Machine Learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[204] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, *et al.*, “Physics-Informed Machine Learning: Case Studies for Weather and Climate Modelling,” *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200093, 2021.

[205] P. Sharma, W. T. Chung, B. Akoush, and M. Ihme, "A Review of Physics-Informed Machine Learning in Fluid Mechanics," *Energies*, vol. 16, no. 5, p. 2343, 2023.

[206] B. Huang and J. Wang, "Applications of Physics-Informed Neural Networks in Power Systems-A Review," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 572–588, 2022.

[207] T. Xiao and F. You, "Building Thermal Modeling and Model Predictive Control with Physically Consistent Deep Learning for Decarbonization and Energy Optimization," *Applied Energy*, vol. 342, p. 121165, 2023.

[208] Y. Dong and S. J. Qin, "Dynamic-Inner Partial Least Squares for Dynamic Data Modeling," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 117–122, 2015.

[209] W. Yu, M. Wu, B. Huang, and C. Lu, "A Generalized Probabilistic Monitoring Model with both Random and Sequential Data," *Automatica*, vol. 144, p. 110468, 2022.

[210] S. J. Qin, Y. Dong, Q. Zhu, J. Wang, and Q. Liu, "Bridging Systems Theory and Data Science: A Unifying Review of Dynamic Latent Variable Analytics and Process Monitoring," *Annual Reviews in Control*, vol. 50, pp. 29–48, 2020.

[211] V. K. Puli, R. Chiplunkar, and B. Huang, "Sparse Robust Dynamic Feature Extraction using Bayesian Inference," *IEEE Transactions on Industrial Electronics*, pp. 1–9, 2023.

[212] J. Zheng, X. Chen, and C. Zhao, "Interval-Aware Probabilistic Slow Feature Analysis for Irregular Dynamic Process Monitoring With Missing Data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2023.

[213] P. K. Huynh, A. A. Alqarni, O. P. Yadav, and T. Q. Le, "A Physics-informed Latent Variables of Corrosion Growth in Oil and Gas Pipelines," in *2023 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7, 2023.

[214] Z. Zhao, Y. Li, C. Liu, and X. Liu, "Predicting Part Deformation Based on Deformation Force Data Using Physics-Informed Latent Variable Model," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102204, 2021.

[215] N. Sammaknejad, Y. Zhao, and B. Huang, "A Review of the Expectation Maximization Algorithm in Data-Driven Process Identification," *Journal of process control*, vol. 73, pp. 123–136, 2019.

[216] J. Rezaie and J. Eidsvik, "Kalman Filter Variants in the Closed Skew Normal Setting," *Computational Statistics & Data Analysis*, vol. 75, pp. 1–14, 2014.

[217] R. Chiplunkar and B. Huang, "Latent Variable Modeling and State Estimation of Non-Stationary Processes Driven by Monotonic Trends," *Journal of Process Control*, vol. 108, pp. 40–54, 2021.

[218] A. Azzalini, "The Skew-Normal Distribution and Related Multivariate Families," *Scandinavian journal of statistics*, vol. 32, no. 2, pp. 159–188, 2005.

[219] C. M. Bishop, "Probability distributions," in *Pattern Recognition and Machine Learning*, ch. 2, p. 93, Springer, 2006.

[220] X. Geng and L. Xie, "Data-driven decision making in power systems with probabilistic guarantees: Theory and applications of chance-constrained optimization," *Annual Reviews in Control*, vol. 47, pp. 341–363, 2019.

[221] K. B. Petersen and M. S. Pedersen, "The Matrix Cookbook," nov 2012. Version 20121115.

[222] C. M. Bishop, "Sequential data," in *Pattern Recognition and Machine Learning*, ch. 13, pp. 639–642, Springer, 2006.

[223] C. Shang, F. Yang, X. Gao, and D. Huang, "Extracting Latent Dynamics from Process Data for Quality Prediction and Performance Assessment via Slow Feature Regression," in *2015 American Control Conference (ACC)*, pp. 912–917, 2015.