Extracting Dynamic Latent Feature with Bayesian Approaches for Process Data Analysis

by

Yanjun Ma

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

Doctor of Philosophy

in

PROCESS CONTROL

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Data-driven approaches have been profoundly studied and successfully applied for process industries, such as in the development of inferential sensors. Among a variety of modelling techniques, the latent variable modelling approaches are widely preferred, which can learn informative features from massive industrial data. In order to make latent variable models more practical for process data analytics, the temporal correlations should be considered in feature extraction. Following the probabilistic modelling procedure, dynamic models are developed in this thesis to describe the latent feature. Besides several probability models, novel inferencing algorithms are elaborated for different application scenarios.

In most chemical processes, features with large inertia and small varying velocity are believed to be more informative. By imposing this modelling preferences as prior distributions of model parameters, the first contribution of this thesis builds the dynamic latent features under a fully Bayesian framework. The preference for large inertia is implemented through a constraint and a prior distribution for the dynamic model of latent features, namely the transition function. The consideration of regularization is implemented through the generative model of raw process data, namely the observation function. Based on the variational Bayesian inference, a novel learning method is developed to extract the slowly varying features and learn model parameters.

The second contribution of this thesis forms a transition function for the constrained latent features. As a hierarchical extension of the hidden Markov model, it describes a dynamic model for the probabilities of discrete variables. By using the Beta distribution to replace the Gaussian distribution, the novel transition function retained similar dynamic characteristics in the constrained domain. The preferred region of transition parameters can be determined for Bayesian inference. In this feature extraction model, a non-linear observation function is used to learn the constrained feature from unconstrained observations, where novel smoothing and marginalizing algorithms are created.

In the third contribution of this thesis, a more practical observation function is proposed to extract dynamic features from multiple operating regions and outlier contaminated data.

Specifically, multiple linear models are utilized to accommodate switching operation regions, and a heavy-tailed noise distribution is used to improve robustness. In order to integrate multiple observation models into the unified dynamic latent feature, a novel Bayesian state estimation algorithm is developed. In its online application, the proposed method is also extended to general multiple model state estimation.

In the fourth contribution of this thesis, another observation function is proposed, which generalizes the ARMAX identification problem under the probabilistic framework. In this work, the dynamic latent feature is used to represent the random (time-variant) time delay, and the proposed Bayesian algorithm can solve the problem of parameter estimation and time delay estimation jointly. In particular, the random time delay is studied for three scenarios, where a static model, a hierarchical model, and a Markov model are developed. With the consideration of temporal correlations, the Markov model provides better performance for system identification. Besides, the hierarchical model also demonstrates its effectiveness of modelling sequentially independent time delay.

The practicality of these proposed feature extraction models and inferencing algorithms are verified using numerical examples, benchmark simulations, and case studies on industrial data. Specifically, the application includes modelling the emulsion quality from the subsurface recovery process, modelling the steam quality from the steam generation process, and a target tracking problem with multiple models.

Acknowledgements

It is with immense gratitude that I would like to thank my supervisor Dr. Biao Huang for all the inspiration and guidance, not only as an unparalleled research pioneer but also as a great mentor. I would like to appreciate significantly his patience and time invested during this fruitful program for five years. Without his mentoring and supervision, a new graduate could not complete academic research work. Besides the advanced knowledge, his positive attitude towards difficulties is what I have mostly benefited from this journey.

I consider it an honour to work with Dr. Shunyi Zhao, Dr. Chao Shang, and Dr. Chunhui Zhao. They inspired my research interest, elevated my studies with genius advices, and showed me the faithfulness towards theoretical research. These academic role models encouraged me to become a diligent and focused professional.

A significant part of this thesis and other achievements in this program come from close collaborations with the experts in oil sand industry. I much appreciate the help of Yamei Liu and Sandeep Sheth, who have taught me to take significant initiatives as a responsible engineer. I also would like to thank Dr. Fei Qi and Eliyya Shukeir, for the guidance on project management and solving complex industrial problems.

I am indebted to my many colleagues in the Computer Process Control group, including Seraphina Kwak, Rahul Raveendran, Ruomu Tan, Yaojie Lu, Ming Ma, Nima Sammaknejad and many others whom I have truly enjoyed their friendship. I want to acknowledge National Science and Engineering Research Council (NSERC) of Canada for their financial support, and the Department of Chemical and Materials Engineering at the University of Alberta for providing a pleasant environment to pursue my Ph.D. degree.

I wish to thank my parents and my friends for supporting my decisions. Without their encouragement, this journey cannot be completed.

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Literature Review . . . 5

1.2.1 Latent Variable Modelling for Process Data . . . 5

1.2.2 Probabilistic Modelling for Process Data . . . 8

1.3 Thesis Outline . . . 11

1.4 Published, Submitted and Under Preparation Materials . . . 14

1.5 Main Contributions . . . 15

2 Mathematical Fundamentals **16**

2.1 Dynamic Linear Models . . . 16

2.1.1 State Space Model . . . 18

2.1.2 Hidden Markov Model . . . 22

2.1.3 Practical Deficiency . . . 24

2.2 Variational Bayesian Inference . . . 25

2.2.1 Comparison with Expectation-Maximization Algorithm . . . 28

2.2.2 Limitations of Variational Bayesian Inference . . . 31

2.3 Particle Based Algorithms . . . 31

2.3.1 Sampling Algorithms . . . 32

2.3.2 Particle Filter and Smoother . . . 34

2.3.3 Stochastic Variational Inference . . . 35

3 Bayesian Solution to Slow Feature Analysis **37**

3.1 Introduction . . . 37

3.2 Problem Statement . . . 39

3.3 Probabilistic Formulation . . . 40

3.3.1 Statistical Dependency . . . 40

3.3.2 Prior Distributions 43
3.4 Variational Inference Methods 45
3.4.1 Estimation of Dynamic Latent Features 47
3.4.2 Estimation of Transition Parameters 49
3.4.3 Variational Lower Bound 50
3.5 Simulations 51
3.5.1 Numerical Examples 51
3.5.2 Process Data Examples 52
3.5.3 Industrial Case Study 54
3.6 Conclusion 56

4 Bayesian Generalization of Markov Transition Model 58
4.1 Introduction 58
4.2 Problem Statement 60
4.3 Probabilistic Formulation 61
4.3.1 Apply Dirichlet Distribution for Hierarchical Hidden Markov Model 61
4.3.2 Transition Model Design with Beta distribution 63
4.3.3 Emission Model Design 67
4.3.4 Prior Distributions 67
4.4 Variational Inference Methods 69
4.4.1 Estimate Latent Features with Revised Particle Smoother 71
4.4.2 Estimate Transition Parameters with Revised Importance Sampling 74
4.5 Simulations 77
4.5.1 Numerical Examples 77
4.5.2 Process Data Examples 83
4.5.3 Industrial Case Study 84
4.6 Conclusions 86

5 Dynamic Latent Feature for Multiple Model Problems 87
5.1 Introduction 87
5.2 Problem Statement 91
5.3 Probabilistic Formulation 91
5.3.1 Multiple Emission Models 91
5.3.2 Robust Realization of Noise Model 93
5.3.3 Use Identity Vector for Switching Mechanism 95

5.3.4 Prior Distributions 96
5.4 Off-line Variational Inference Methods 99
5.4.1 Learning Multiple Robust Emission Models 100
5.4.2 Smoothing States with Multiple Emission Models 102
5.4.3 Selecting Model Structures with Variational Lower Bound 105
5.4.4 Numerical Simulations 107
5.5 On-line Application with Variational Inference Results 112
5.5.1 Filtering States with Results of Variational Inference 113
5.5.2 Process Description of Steam Quality Inferential Sensor 114
5.5.3 Apply the Bayesian Feature Extractions 115
5.6 Extending On-line Feature Learning Algorithm States to General Multiple Models 118
5.6.1 General Linear Multiple State-Space Models 118
5.6.2 State Estimation with Variational Methods 119
5.6.3 Approximate the Filter with Proposal Distributions 120
5.6.4 Extension for Other Latent Variables 123
5.6.5 Application Examples of Filtering Algorithm 124
5.7 Conclusions 128

6 Dynamic Latent Feature for Time Delay Problems 130
6.1 Introduction 130
6.1.1 Random Time Delays 131
6.1.2 Probabilistic Identification 132
6.2 Problem Statement 133
6.3 Bayesian Interpretation of ARMAX Identification 134
6.3.1 Probability Density of Coloured Noise 134
6.3.2 Log-Likelihood and Bayesian formulation of ARMAX Model 136
6.3.3 Identification with Variational Bayesian Inference 138
6.4 Probabilistic Formulation of Time-Invariant Time Delays 144
6.4.1 Modelling Assumption 145
6.4.2 Variational Inference 146
6.5 Probabilistic Formulation of Independent Time-Variant Time Delays 149
6.5.1 Modelling Assumption 149
6.5.2 Variational Inference 150

6.6 Probabilistic Formulation of Dependent Time-Variant Time Delays 152

6.6.1 Modelling Assumption 153

6.6.2 Variational Inference 153

6.7 Practical Consideration for Time-Variant Time Delays 157

6.8 Simulation 159

6.8.1 Validation for Time Delay Estimation 159

6.8.2 Validation for System Identification 160

6.8.3 Validation for Inferential Sensor Modelling 163

6.9 Conclusions 166

7 Concluding Remarks and Future Directions 167

7.1 Concluding Remarks 167

7.2 Future Directions 170

7.2.1 Bayesian Co-integration Analysis 170

7.2.2 Transition Noises in Recurrent Neural Networks 172

Bibliography 174

List of Tables

3.1 Comparison in the Tennessee Eastman Process 53

3.2 Comparison in the SAGD Data 1 55

3.3 Comparison in the SAGD Data 2 55

4.1 Validation Results of Estimating Precision Parameter 79

4.2 Validation Results of Estimating SAGD Water Content 85

5.1 Comparison of the Variational Lower Bounds 109

5.2 List of available process variables 115

5.3 Validation Results of Steam Quality Inferential Sensing 117

5.4 Error from Monte Carlo Tests for the Tracking Problem 126

5.5 Error from Monte Carlo Tests for Different Outliers 127

6.1 Estimate C Parameters with Different Orders 142

6.2 Validation Results for Constrained Transition Matrix 158

List of Figures

1.1 Sketch of SAGD Subsurface Recovery Process 3

1.2 Sketch of Once Through Steam Generator Process 4

1.3 Validating Latent Variable Models with Regression Tasks in TEP 6

1.4 Feature Extraction with Encoding and Decoding Format 9

1.5 Extracting Dynamic Latent Features for Process Data Modelling 12

2.1 Advantage of Using Parameter Distributions 29

2.2 Kullback-Leibler Divergences between Normal Distributions 30

3.1 Latent Features with Different Varying Velocities 42

3.2 Different Choices of Hyper-Parameters in the Beta Distribution 43

3.3 Proposed Bayesian design for slow feature extraction 44

3.4 Unsupervised indices to determine the latent dimension 52

3.5 Supervised indices to determine the latent dimension 53

3.6 Consistency of the proposed variation methods 54

3.7 Validation Results for Modelling SAGD Data 1 56

3.8 Validation Results for Modelling SAGD Data 1 57

4.1 Extending Hidden Markov Model with Constrained Dynamic Latent Feature 62

4.2 Transition Properties of Constrained Dynamic Features 65

4.3 Stationary Properties of Constrained Dynamic Features 66

4.4 Proposed Bayesian design for constrained dynamic feature extraction 68

4.5 Learning Algorithm for the Proposed Feature Extraction Model 70

4.6 Updating Latent Feature with Particle Based State Estimation 74

4.7 Log-Likelihood Function for a and ρ 76

4.8 Estimation of Constrained Dynamic Latent Features with Model 78

4.9 Estimation of Transition Parameters for Constrained Dynamic Feature . . . 79

4.10 Estimation of Constrained Dynamic Latent Features without Model 80

4.11 Estimation of Emission Matrix for Constrained Dynamic Feature 81
4.12 Example-1 Estimation of Emission Matrix for Constrained Dynamic Feature 82
4.13 Example-2 Estimation of Emission Matrix for Constrained Dynamic Feature 83
4.14 Prediction Performance on Tennessee Eastman Process Simulations 84
4.15 Estimation of Water Content in SAGD Process 86

5.1 Multiple Emission Model for Dynamic Latent Feature 92
5.2 Handling Outliers with Student-t distribution 94
5.3 Selecting model identity from Dirichlet distribution 97
5.4 Proposed Bayesian design for Multi-Model Dynamic Feature 98
5.5 The Spiral Data in 3-Dimension Space with a 1-Dimension Non-Linearly Embedded 108
5.6 Evolutions of the variational Lower Bound with Randomized Initializations 109
5.7 Comparisons of Features and Clusters for Different Identified Models 110
5.8 Selecting the Number of Latent Dimension and the Number of Emission Models 111
5.9 Comparison of Estimated Model Identities 112
5.10 Comparison between Off-line and On-line Model Identity Estimations . . . 114
5.11 Flow Diagram of Once Through Steam Generator 115
5.12 Typical Measurements from an Operating Steam Generator 116
5.13 Comparison of Prediction from Different Latent Features 117
5.14 Root Mean Squared Error for 19 Examples 124
5.15 Root Mean Squared Error for Example-4 125
5.16 Example-1 of Tracking Results 127
5.17 Example-2 of Tracking Results 127
5.18 Step-wise RMSE of Estimation with Outliers 128

6.1 Random Time Delays 132
6.2 Modelling Time Delay with Latent Variable 133
6.3 Log-Likelihood Function of ARMAX Covariance Parameters 137
6.4 Estimating Covariance Parameters through Re-Sampling Methods 141
6.5 Validating Identification Performance of Bayesian ARMAX 143
6.6 Modelling Assumptions for Time-Invariant Time Delays 146
6.7 Learning Time Invariant Delay with Trivial Initial 148
6.8 Learning Time Invariant Delay with Wrong Initial 148

6.9 Modelling Assumptions for Independent Time-Variant Time Delays 150
6.10 Modelling Assumptions for Independent Time-Variant Time Delays 154
6.11 Estimate Hidden Markov Sequence with High Order Observations 156
6.12 Effects of Two Assumptions for Time Variant Time Delays 159
6.13 Describe Random Delays with Dirichlet Distribution 160
6.14 Evolutions in the Distribution of π in VB Inference 161
6.15 Comparison of Delay Estimations in ARMAX Identification 162
6.16 Learning Procedure with Independent Time Delays 162
6.17 Learning Procedure with Dependent Time Delays 163
6.18 Demonstration of Hierarchical Model in Steam Quality Soft Sensor 164
6.19 Demonstration of Hierarchical Model in Water Content Soft Sensor 165

7.1 Scaling Monitoring in Steam Generation Process 171
7.2 Possible Structure for Recurrent Variational Auto-Encoder 173

Chapter 1

Introduction

In this chapter, the motivations of this thesis are introduced at first. Literature related to process data analytics is reviewed for latent variable modelling and probabilistic modelling. After that, the outline and the contributions of this thesis are provided.

1.1 Motivation

In the process industry, abundant sensors and transmitters are installed for control and monitoring applications. In the corresponding Process History Database, both steady and dynamic behaviours can be recorded for the target process. In recent years, with the development of data analytics technologies, historical data becomes more valuable for process control and optimization. For example, in the robust optimization problem, historical data can be used to learn statistical moments and confidence regions to describe the uncertain constraints [1]; in the root cause analysis problem, historical data can be used to extract logical connections between process variables [2]. Besides its use in preliminary data analysis, availability of historical data makes the data-driven approaches viable for model development. For example, in the statistical process monitoring approaches, the model can be built by using the data from normal operating regions and then be used to detect and diagnose conditions that are deviated from the normal operations [3]. The essence of historical data can be learned as a model, upon which the real-time decision can be made once new data samples have arrived.

In this thesis, the objective of analyzing process data is to build a model for the online estimation of key process variables. Monitoring of key process variables, typically the quality variables in chemical processes, is important for process operations. Real-time availability of these variables is crucial for control and optimization. However, hardware instruments do not always work satisfactorily. Advanced analyzers, such as those based on nuclear technologies, may be useful in pilot projects. However, they usually suffer from reliability issues and have a high maintenance cost in practical applications. For example, in a bitumen recovery process of the oil-sand industry, maintaining multiphase flow meters for produced emulsions can be prohibitively expensive. As a conventional solution, sparse accurate measurements can be obtained from the sampled product. Usually, these results can be used for certain regulatory and performance reports. However, because of the slow-rate and time-consuming analyzing procedure, they are not sufficient for Advanced Process Control. As an alternative solution, estimating key process variables with inferential sensors has become a more favourable solution for the on-line measurement of key process variables [4, 5, 6].

There are several noticeable features and advantages of inferential sensors in comparison with traditional instrumentation.

- Performance satisfaction. The sampling rate of input variables is normally fast, which makes the outputs of the inferential sensor granular enough for the online application. More importantly, the customized regression model can be established and optimized for certain real-time application scenarios.

- Insight of processes. Since the regression model of an inferential sensor is built from historical data of the designated unit, the parameters can reveal the connection between regularly measured process variables and the quality variables. Besides, the prediction performance of soft sensors will also be improved by taking both process knowledge and data analytics into consideration.

- Robustness. The inferential sensors are more than a regression model. As an integrated algorithm package, the inferential sensor protects the model by using data preprocessing and filtering techniques, which can improve the robustness in the presence of abnormal situations.

- Adaptation. The inferential sensors are often implemented with an adaptation mechanism, which conducts the on-line updating by using the latest reference samples. With the inherited parameter learning methods, further improvement of inferential sensors can be realized automatically with the addition of new process data.

It can be observed that the essential part of an inferential sensor is the predictive model, which predicts the quality variables from the related and reliable process measurements. In ideal cases, this model can be determined based on physical understandings and conservation laws, such as establishing the mass balance and the energy balance equations. However, similar to the hardware instruments, the knowledge-based models can frequently fail due to unattainable physical assumptions, biased process measurements, and harsh operating conditions. On the other hand, with the increasing size of process data and the well-developed data analytics algorithms, data-driven models are proved to be useful in those situations [7, 8, 9]. The main advantage of data-driven methods is their flexibility for various applications. Typically, for the chemical processes with a variety of operating conditions, the historical data becomes more important in dealing with the modelling challenges. In what follows, two industrial processes will be briefed to introduce the primary interest of this thesis.

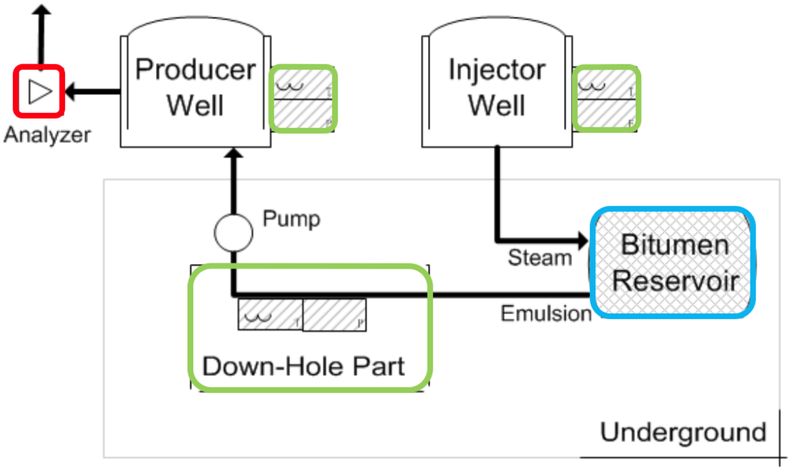


Figure 1.1: Sketch of SAGD Subsurface Recovery Process

The first process is about the subsurface portion of the Steam Assisted Gravity Drainage (SAGD) process. In this subsurface recovery process, two horizontal wells are drilled into underground bitumen reservoir, where the high-pressure steam is injected through the top injector well to heat the heavy crude oil, and a mixture of oil and water (emulsion) is pumped out through the bottom producer well. The injected steam is used to form a steam chamber in the bitumen reservoir. The heated bitumen can flow down along the edges of this steam chamber. In order to maximize water usage and monitor the production efficiency, the composition of produced emulsion should be measured. Specifically, an online estimation of the total flow rate of water and oil ($m_{water} + m_{oil}$) and an associated composition ratio ($\frac{m_{water}}{m_{water}+m_{oil}} \times 100\%$) are desired for subsequent control applications. Based on the physical understanding, the production of oil (m_{oil}) is known to be highly dependent on the chamber's condition, such as the chamber temperature, the chamber pressure, and the porosities of bitumen reservoir. Unfortunately, such properties are difficult to measure or only sampled sparsely in reality. On the other hand, many other process measurements can be used to infer those hard-to-measure ones. For example, Figure 1.1 illustrates a typical topology for one well-pair. Available input variables are shown with green boxes, and the most prominent features are contained within the blue box. In some pilot project, the advanced analyzers, as denoted in the red box, may be available to provide the reference outputs in real-time. In more general cases, only a limited number of certain references can be obtained from a downstream test separator. To build a "predictive" model from available input variables to the quality variable, the latent features in the blue section will play an important role.

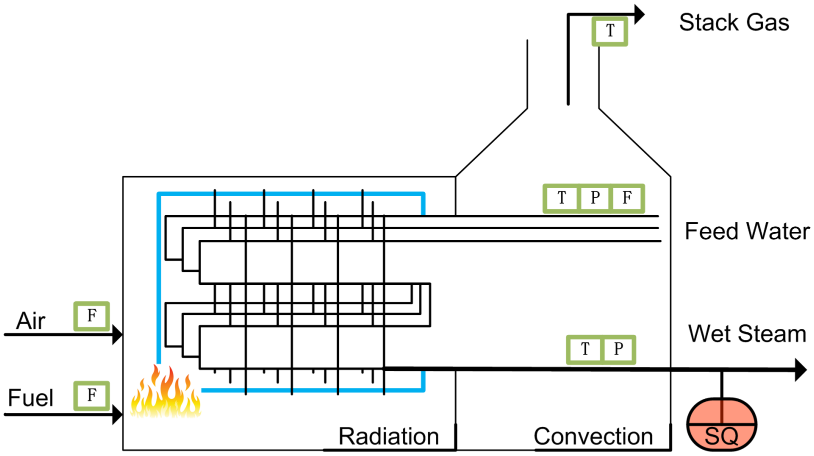


Figure 1.2: Sketch of Once Through Steam Generator Process

The second process is about generating the high-pressure steam for the injector well. The most widely used methodology is based on Once-Through Steam Generators (OTSGs). In this process, the fuel gas is burned with air flow to boil the feed water into the high-pressure wet steam. Since the produced fluid is a mixture of steam and water, the steam portion needs to be further separated before being used in the subsurface recovery process. The efficiency of this steam generation process is closely related to the composition of wet steam, for which a steam quality measure ($\frac{m_{steam}}{m_{water}+m_{steam}} \times 100\%$) is used as an indicator. If this steam quality can be accurately measured in real-time, its operational variations can be reduced, and the efficiency of water usage can be improved. In Figure 1.2, a brief flow diagram summarises the process and available measurements of a typical OTSG. In this process, the inlet flow of boiler feed water is divided into several individual passes in order to improve heat transfer. As a common practice, fuel gas flow-rate, feed water flow-rate, feed water temperature and pressure, and steam temperature and pressure are available. Based on process knowledge, the crucial part of heat transfer occurs in the radiation section, which affects the steam quality dominantly. Unfortunately, in most application cases, no reliable sensors can be implemented there. Thus, to build an interpretable model from the input variables (green) to the quality variable (red), it is necessary to extract the latent variations/features in the blue section.

It can be observed that in both of these two industrial processes, the quality variables of interest are not directly related to the available process measurements. In Figure 1.1, the water content depends on the underlying condition of bitumen reservoir and the steam chamber, while this condition can be indirectly determined by the injected steam and be reflected by the "down-hole" and producer measurements. In Figure 1.2, the steam quality depends on the operation condition of the radiation section, but available measurements of input variables do not directly reveal this condition. To correctly model the quality variable in such scenarios, latent variable models are usually favoured comparing to other data-driven models. Besides, the probabilistic approaches are demonstrated to be effective in dealing with real-world process data. In the following section, related studies and algorithms will be reviewed for latent variable modelling and probabilistic approaches.

1.2 Literature Review

1.2.1 Latent Variable Modelling for Process Data

The historical data of most processes is rich, but it usually suffers from information redundancy [8]. For example, concurrent variations from numerous process variables can be originated from the same source, such as the correlated pressure and temperature shown in Figure 1.2. Without an effective dimension reduction method, the regression model will be affected by the co-linearity of process variables, which results in an inaccurate and misleading estimation of model parameters. On the other hand, within a specified operating range, most process variables can be determined by a small set of manipulated variables. For the example shown in Figure 1.1, the pump frequency affects both "down-hole" pressure and temperature. Usually, the significant variations can be captured by the pump speed. However, for specific prediction objectives, the residual variation or the insignificant trends could be more informative for predicting the output. For example, in one application case, the steam quality is found to be highly correlated with the residual variation after correlating the steam temperature and the steam pressure. Based on the above examples and modelling challenges, the latent variable modelling techniques, or representation learning methods [10], are widely applied in learning the shared causes. Instead of building an input-output model directly, this approach formulates a latent space to represent useful information in the input variables.

Formulating the latent variables is a step to extract informative features from raw data, which is critical for multivariate data analysis [10, 11]. It is expected that latent features can provide higher effectiveness and more robustness than the raw input data itself. Originally, feature extraction is used in image processing [12, 13] to identify specific objects. For example, in geographical image processing, analyzers are designed to detect linear objects such as roads and airport runways, or non-linear objects such as rivers and trails. While applied for general latent variable models, the scope of feature extraction has been extended, and the features are not necessarily related to the real objects. In the most important latent variable modelling approach, principal component analysis (PCA) [14], the features (or the latent scores) are extracted to represent the variability of multivariate observations. In the regression-oriented latent variable models such as partial least squares (PLS) [15], the features are extracted to capture common variations between inputs and outputs. In these methods, specified preferences are used to formulate the latent features, for example, emphasizing the large variance in PCA and emphasizing the large mutual correlation in PLS.

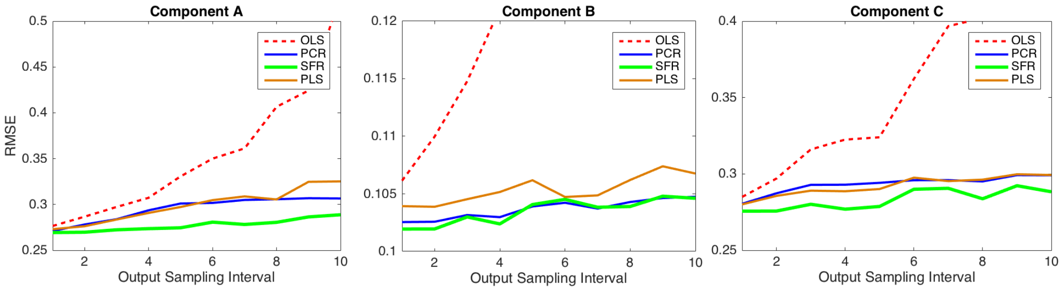


Figure 1.3: Validating Latent Variable Models with Regression Tasks in TEP

As a demonstration for latent variable modelling, an initial numerical experiment is performed to test the prediction performance on the Tennessee Eastman simulation [2]. The result is presented in Figure 1.3, where several popular regression methods are compared. Root mean square error (RMSE) is evaluated by varying the sampling interval of the training output samples. In practice, these large sampling intervals correspond to the slow sampling rate of accurate output measurements. As shown in Figure 1.1 and Figure 1.2, the accurate measurements of water content come from a shared test separator [16], and the accurate measurements of steam quality are obtained from manual sampling and laboratory analysis. Thus, a slower sampling rate is commonly encountered in the measurements of quality variables. In this simulation, the ordinary least squares (OLS) method has shown degraded performance for large output sampling intervals, but the other latent variable models provide more consistent performance when the sampling interval increases. The best performance in this simulation is achieved by the slow feature based regression model (SFR), which has a more favourable modelling preference in process data analysis.

As the basis for this best regression model shown in Figure 1.3, slow feature analysis (SFA) is an unsupervised learning approach for extracting dynamic latent features [17]. Comparing to PCA, the learning objective of SFA focuses on the slowness of latent features rather than the magnitude of variations. To describe the slowness, an averaged varying velocity [17] for the stochastic sequence $s_{1:T} = \{s_1, s_2, ..., s_T\}$ can be defined as

$$v(s) = \mathbb{E}\left[\|s_{t+1} - s_t\|^2\right] \approx \frac{1}{T-1}\sum_{t=1}^{T-1}\left(s_{t+1} - s_t\right)^2, \tag{1.1}$$

where $\mathbb{E}[\cdot]$ stands for the average over whole sequence of features. In SFA, the latent feature with the smallest varying velocity is favoured. Based on the above definition, an optimization problem can be formulated to extract the slowest latent feature:

$$\begin{aligned} \min_{g} \quad & v(s), \\ s.t. \quad & s_t = g(X_t), \quad \forall\, t \in \{1, 2, ..., T\}, \\ & \mathbb{E}\left[s_t\right] = 0, && (1.2) \\ & \mathbb{E}\left[s_t^2\right] = 1, && (1.3) \end{aligned}$$

where g is the encoding function, which maps each observed data sample X_t to each latent state s_t. Combined with the constraints of zero-mean (1.2) and unit-variance (1.3), an optimal solution can be attained deterministically. In this optimization procedure, these two constraints prevent the trivial solution of a straight line. When applied to extract multiple latent features, another constraint on mutual independence is enforced as

$$s^{(i)} \perp s^{(j)}, \quad \forall\, i \neq j. \tag{1.4}$$

Based on the above three constraints, a complete slow feature extraction is amenable. Specially, by using a linear encoding function $g^{(j)}(X_t) = p_j' \cdot X_t$, the projection can be formulated similar to the PCA modelling, resulting in a matrix of projection coefficients $P = [p_1, p_2, ..., p_m]$.

The effectiveness of SFA comes from its preference for slowness. Typically, in a chemical process, slower variation is usually believed to be more informative than the fast variation because of the comparatively large inertia of chemical processes. Therefore, the consideration of temporal correlation between consecutive latent samples can improve modelling performance. Based on this consideration, several variants and applications have been developed [18, 19].

In addition to the extraction methods mentioned above, latent features can also be formulated according to other modelling preferences, such as through the canonical correlation analysis [20] and the co-integration analysis [21]. Enrichment of modelling preferences and more detailed description of latent space will improve the efficiency of feature extraction procedure, as well as extend the application scope. Motivated by the improved result from the consideration of temporal correlation, this thesis considers building the dynamic model to describe latent features, which can be presented in an auto-regressive form: $s_t = f(s_{t-1})$. Consequently, the observed data is expressed as the output of a decoding function, for example $X_t = g(s_{t-1})$. In what follows, this decoding format of feature extraction and related probabilistic modelling approaches will be introduced.

1.2.2 Probabilistic Modelling for Process Data

Generally, a feature extraction model can be formulated with two complementary formats: one models the latent feature as a function of observations, and the other treats the observed data as a projection from latent features. A comparison is presented in Figure 1.4. With a set of invertible functions, these two formats are convertible. As an example, for linear PCA, after having the encoding matrix P learned from the eigenvectors of $\sum_{t=1}^{T} X_t X_t'$, the corresponding decoding matrix can be modelled as P' (the transpose of P). However, in more general cases, these two formats have significant differences. In the encoding formulation, the learning objective is usually formulated as a deterministic function of the latent features, such as the averaged varying velocity in slow feature analysis. On the opposite, the decoding format is commonly formulated as a maximum likelihood estimation problem, which uses a conditional distribution to describe the observed data. Based on the specified probabilistic formation, the features and parameters with the best performance in explaining the observed data will be the optimal solution.

For example, as the decoding counterpart of the linear PCA, probabilistic principal component analysis [22] has been widely acknowledged. In this model, the formulation of $X_t = [h_1, ..., h_m]' \cdot S_t + e_t$ is considered, where the latent states are denoted with a vector

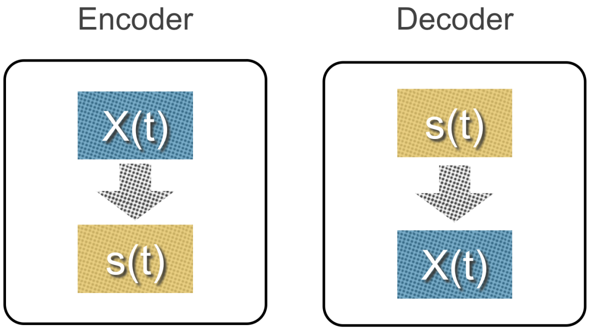


Figure 1.4: Feature Extraction with Encoding and Decoding Format

S_t. Based on the assumed observation noise e_t, the conditional distribution of observed data can be written as $p(X \mid H)$. The modelling objective thus becomes maximizing this likelihood function $p(X \mid H)$ in terms of the parameter matrix $H = [h_1, ..., h_m]$. Unlike the deterministic optimization techniques from the encoding formulation, the probabilistic inference approaches can be introduced for the decoding formulation, where a variety of probability distributions are available to reinforce the learning procedure.

As the mainstay of engineering and computer science, the probability theory is viewed as a preferred tool in formulating models and learning parameters from data [23]. In probabilistic approaches, the model can be built under an explicit prior preference, the structure can be selected based on certain Bayesian evaluation indices, and the model predictions can be assigned with specific beliefs for on-line usage [8]. In addition to the aforementioned probabilistic PCA, most deterministic data analysis approaches have their probabilistic extensions as well [24, 25]. In the following discussion, it will be illustrated that the decoding format and the associated probabilistic algorithms are capable of providing a better interpretation for the latent features, and improving the learning efficiency.

First, better modelling performance is achieved by adopting a better and more robust description of observation noises. In engineering problems, particularly for industrial process data, the raw inputs are often contaminated by different types of noises, including outliers. Usually, data pre-processing methods, such as wavelet analysis, can be applied individually for a filtering purpose. However, the resultant signals could be distorted inevitably, and a strong filter may mask informative variations of the raw data. By using the decoding format and the probabilistic framework, the measurement uncertainties will be modelled together with the latent features. In particular, an explicit noise term e_t is used to describe the disturbances, and the remaining part (the informative/relevant variations) can be reflected in the latent space. By assigning an appropriate probabilistic description to e_t, such as requirements on statistical moments or certain forms of probability distributions, the noise of observations can be well described by e_t, and thus be screened out of the latent space. For example, the robustness against outliers and sparse measurements can be improved by using appropriate probability distributions for observed data [24, 26].

Second, the better interpretation of latent features comes from the explicit model for latent feature $s_{1:T}$. In the probabilistic PCA case, each latent state s_t is assumed to be drawn from the standard Normal/Gaussian distribution $\mathcal{N}(0, 1)$. This description of latent feature materializes the normalization requirements of original PCA. In the mixture models, the discrete latent states are usually formulated with a hierarchical model, where the parsimony principle can be materialized [27]. In the emerging deep learning approach [28], the distribution of latent state can be described by a multi-layer neural network. In the above decoding formats, the specification of latent features is summarised by a probability distribution of $p(s_{1:T})$. In this thesis, by considering a dynamic model for the latent features, the formulation of $p(s_{1:T})$ will be realized for the sequentially dependent latent states, namely $s_{t-1} \not\perp s_t$. Comparing to those dynamic statistics of $s_{1:T}$ enlisted in the encoding format, such as the velocity form in (1.1), establishing an explicit dynamic model for $s_{1:T}$ can provide more information and obtain more detailed descriptions for the latent feature.

With broad modelling scope comes a large solution space. Regarding the decoding format, the maximum likelihood estimation could yield multiple solutions and might be stuck in local optima. For example, the probabilistic PCA needs a well-established initial point, which is usually obtained by applying the original PCA method first [22]. In order to avoid the difficulties in parameter estimation, some assumptions about available data can be considered in the theoretical derivation, which helps to achieve a consistent parameter estimation [22, 25]. As for the applications on industrial data, practical challenges, such as the time-varying parameters and the lack of distributional normality [16], can introduce more difficulties to parameter estimation. To overcome these difficulties, from a probabilistic modelling framework, the Bayesian inference approach [8, 26] is used throughout this thesis.

When the Bayes rule is applied to estimate model parameter, the available process knowledge and modelling preferences can be implemented through the prior distribution of model parameters. During the inference procedure, instead of point estimation, model parameters can be learned along with their probability distributions. Regarding the objective of feature extraction, the Bayesian framework can also provide an explicit indicator for the training performance [29], leading to an automatic selection of the optimal model structure.

In this thesis, several Bayesian models and their inferencing algorithms are developed for various modelling problems in process data analytics.

1.3 Thesis Outline

Having introduced the background of latent variable models and probabilistic approaches, the following chapters of this thesis are organized as follows.

In Chapter 2, a review of related models and inference algorithms is provided as the mathematical background. It focuses on two main aspects, dynamic linear models and the variational Bayesian inference. The dynamic linear model, describing the temporal connection of latent states, is used as a basis of the decoding format for extracting dynamic latent feature. Depending on different supporting domains, the dynamic models can be distinguished between the state space model and the hidden Markov model. In the first section of Chapter 2, these two fundamental structures are reviewed under the context of feature extraction. For each structure, a sample method is introduced for parameter estimation. In the second section of Chapter 2, the variational Bayesian inference is introduced as an advanced probabilistic learning approach. Its concept and usage are presented using the general structure for dynamic feature extraction. Under this Bayesian framework, model parameters and latent features can have flexible representations. Notably, the particle-based representation is introduced as a preferred option. In the last section of Chapter 2, the related particle-based algorithms are reviewed.

The novel contribution of this thesis is organized into four chapters, from Chapter 3 to Chapter 6. The first two chapters focus on designing the novel transition function for underlying dynamics, and the latter two chapters present more practical observation functions for the application of dynamic latent features. Figure 1.5 shows their relation in the dynamic feature extraction process.

In Chapter 3, the aforementioned slow feature analysis model is interpreted under a Bayesian framework. Based on a state space formulation of the probabilistic slow feature analysis, prior distributions are developed for the transition parameters and the observation parameters. In the inference process for those constrained parameters, a novel algorithm is designed by integrating the importance sampling into the variational inference framework. Based on the validation results on simulated datasets, the established Bayesian model shows its strength for feature extraction, and the learning results are robust to randomized initial guesses. Therefore, the problem of inconsistent parameter estimation is prevented by developing the fully Bayesian framework and using the variational inference algorithm for

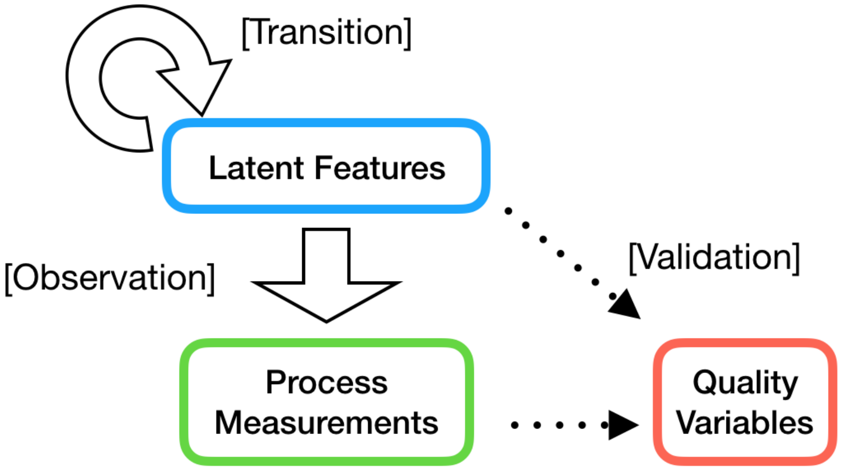


Figure 1.5: Extracting Dynamic Latent Features for Process Data Modelling

this decoding format. Based on the validation with industrial datasets, the proposed model is demonstrated to have advantages over other similar feature extraction models. For this inferential sensing application, the extracted dynamic features provided more accurate and more robust predictions.

In Chapter 4, the transition function of hidden Markov model is generalized for constrained continuous variables. Rather than modelling the transition between discrete states, this chapter considers the transition between their parameters. As the probabilities for the original discrete states, they have a constrained supporting domain. In the two-dimensional case, this latent state describes a possibility in $(0, 1)$. For such constrained states, a transition function is developed by generalizing the hidden Markov model with the Beta distribution for the two-dimensional case and generalizing with the Dirichlet distribution for the multi-dimensional case. In this study, it is found that the transition function with Beta distribution yields more consistent learning results for the feature extraction. The properties of associated transition parameters are shown to be comparable with the general state space formation in Chapter 2. The connection between the transition parameters and the averaged characteristics of latent features is studied through numerical simulations, where the preferred regions of model parameters are provided. Thus, a guideline for selecting the prior distribution is available for probabilistic inference. In order to project constrained dynamic latent features onto the general and unconstrained observations, a non-linear observation function is used, which is similar to the reversed sigmoid function. In order to learn these constrained dynamic latent features, the inference algorithm is developed under the variational Bayesian framework, and a particle-based representation is adopted following the Beta distribution. As a result, the constrained and non-linear state estimation problem is solved with higher efficiency. Besides, a novel marginalization method is proposed for estimating the transition parameters, which provides more informative estimates from their joint distribution. Based on numerical simulations, this marginalization algorithm has significantly improved the estimation of variance parameters. The overall performance of the proposed model is tested with the case study on a SAGD subsurface recovery process, where the underground variables have a constrained operating range.

In Chapter 5, the observation function is considered to have multiple linear models. In order to model a time-varying process where the quality variable is controlled under several operating conditions, a combinatorial feature extraction model is developed. By assuming consistent dynamics for the quality variable under different operating conditions, the transition function is kept the same as that in Chapter 3. However, the observation function is extended to have multiple linear models for different operating conditions. In order to avoid false modelling caused by outliers, the observation noise is described by a heavy-tailed distribution. In off-line learning, the number of emission models tends to be smaller when adopting the student-t distribution. Similarly, in the online application, the switching actions become less sensitive to abnormal samples or outliers. For the inferencing algorithm, a novel Bayesian method is developed to estimate the latent states by merging multiple models. To determine the model structures, the variational lower bound is evaluated as a performance index for selecting the dimension of latent space, as well as for selecting the number of emission models. Unlike the conventional cross-validation approach, this index has incorporated the parsimony principle through the prior distributions. Its effectiveness is demonstrated through simulated datasets. Furthermore, a case study on steam quality inferential sensors is used to demonstrate the strength of extracted dynamic latent features.

A study of variational Bayesian methods for multiple model state estimation problem is also included in Chapter 5. For the general multiple model problem, both the observation function and the transition function are formulated with multiple linear models. By solving this state estimation problem under the variational Bayesian framework, the original estimation objective becomes a minimization problem of a Kullback-Leibler divergence. Under this framework, the distribution of the state variable and the distribution of the model identity variable can be optimized simultaneously. Based on an integration-based approximation strategy, the resultant algorithm improves the estimation accuracy, as well as reduces the computational load when comparing to other state-of-art algorithms.

In Chapter 6, the observation function is developed to represent a likelihood function of the autoregressive-moving-average model with exogenous inputs (ARMAX) system. In this case, the latent feature itself represents a time-variant time delay variable. Based on the transition function of the hidden Markov model, which is introduced in Chapter 2, the time delay variable can be considered to possess first-order Markov properties. Usually, the prediction error method in system identification optimizes the parameters with a deterministic optimization algorithm. Through the proposed probabilistic interpretation, a novel representation of the coloured noise is used to formulate ARMAX identification as a maximum likelihood problem. After the corresponding validations, this maximum likelihood formulation is extended to integrate the probability models of time delay. In this work, the variational inference method is developed to determine the random time delay along with identifying the parameters for the ARMAX model. The challenge in learning this dynamic feature, namely time-variant time delay, is that the "observations" are correlated because of the coloured noise in the ARMAX model. In other words, one instant of time delay can affect multiple observation data samples. In order to solve this problem under the variational Bayesian framework, an augmentation of the original hidden states is proposed, and a novel state estimation algorithm is developed. Through numerical simulations and industrial case studies, the proposed probability model of time delay is validated and proved to have advantages in system identification.

In Chapter 7, concluding remarks are presented, and the possible future work and further improvements are discussed.

1.4 Published, Submitted and Under Preparation Materials

Most materials of this thesis have appeared in the following publications and patent.

1. Ma, Yanjun, and Biao Huang. "Bayesian learning for dynamic feature extraction with application in soft sensing." IEEE Transactions on Industrial Electronics 64, no. 9 (2017): 7171-7180. (***Chapter 3***)

2. [Accepted] Ma, Yanjun, Shunyi Zhao, and Biao Huang. "Feature Extraction of Constrained Dynamic Latent Variables." IEEE Transactions on Industrial Informatics. (***Chapter 4***)

3. Ma, Yanjun, and Biao Huang. "Extracting dynamic features with switching models for process data analytics and application in soft sensing." AIChE Journal 64, no. 6 (2018): 2037-2051. (***Chapter 5 - except Section 5.6***)

4. Ma, Yanjun, Shunyi Zhao, and Biao Huang. "Multiple-Model State Estimation Based on Variational Bayesian Inference." IEEE Transactions on Automatic Control (2018) (***Section 5.6***)

5. Ma, Yanjun, Seraphina Kwak, Lei Fan, and Biao 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. (***Chapter 6 - short version***)

6. [To be submitted] Ma, Yanjun, Shunyi Zhao, and Biao Huang. "Probabilistic Identi cation of ARMAX Model with Random Time Delays." (***Chapter 6 - complete version***)

7. Biao, Huang, Yanjun Ma, and Seraphina Kwak ."System and Methods for Real-Time Steam Quality Estimation." (***United States Provisional Patent Application*** filed on March 5, 2018, ***Canadian Patent Application*** filed on March 5, 2019)

1.5 Main Contributions

The main contributions of this thesis can be summarized as follows.

1. Development of four probabilistic feature extraction models to extract dynamic latent features based on historical process data and modelling preferences.

2. Development of an integration procedure between the variational Bayesian inference and the particle-based algorithms to deal with constraints and non-conjugate prior.

3. Development of constrained continuous latent states with a transition function to investigate the dynamic property of probability vectors.

4. Development of Bayesian inference for ARMAX model identification to include more flexible and more detailed modelling structures.

5. Formulating process knowledge as the specific prior distributions of latent feature and parameters towards better modelling performance for quality variables in process industries, particularly with application to the steam generation process and the SAGD subsurface recovery process.

Chapter 2

Mathematical Fundamentals

In this chapter, a review is provided for the mathematical models and the inference algorithms that have been adopted in this thesis. As aforementioned, the transition behaviour of latent features can be described explicitly through dynamic models. Here, two fundamental structures are introduced, namely the state space model and the hidden Markov model. As a selected inferencing framework, the variational Bayesian approach is introduced as well, which supports the algorithms that will be developed in the later chapters. Besides, particle-based inferencing methods are also introduced as a preliminary for dealing with practical challenges.

2.1 Dynamic Linear Models

Dynamic linear models are capable of describing a general class of time series models. By formulating a transition function for the consecutive states, the conditional distribution of state s_t can be modelled as a function of the preceding state s_{t-1}. Thus, the entire state sequence can be described by a unique probability distribution $p(s_{1:T})$. Usually, it only uses a first-order transition function, and if necessary, the order can be increased by using a multi-dimensional latent space. For the application on industrial datasets, this first-order transition function also avoids additional inferencing and maintenance difficulties.

In this thesis, a general form of the first-order transition model is studied in the form of

$$s_t \sim f(s;\ s_{t-1}, \theta_{tr}), \tag{2.1}$$

where θ_{tr} represents the transition parameters. The function $f(\cdot)$ is modelled as a probability distribution for the state s_t, where s_{t-1} and θ_{tr} are regarded as the associated parameters. Unlike the deterministic transition functions, such as the Long Short-Term Memory in recurrent neural networks, the transition function (2.1) defines the stochastic behaviour of $s_{1:T}$. Although the consecutive state s_t still depends on s_{t-1}, it has a certain level of variability through the transition function (2.1), which is used as a conditional distribution. This consideration of stochastic behaviour is based on the inevitable uncertainties of the assumed transition function (2.1). When adopted for process data analysis, this consideration matches the common understanding about the stochastic behaviour of physical systems. As a result, the possibilities of inaccurate transition parameters and low-quality observations can be accommodated in modelling.

In the observation function, the observation sample x_t is connected with the latent state s_t through a general form of

$$x_t \sim g(x;\ s_t, \theta_{ob}). \tag{2.2}$$

Similarly, the observation function $g(\cdot)$ is used to describe a probability distribution of x_t, where the latent state s_t and the observation parameter θ_{ob} are regarded as the parameters. By assuming that all the dynamic behaviour has been included in the function (2.1), this observation function is usually used as a static portion of the model, only describing the static connections. Although other forms of non-linear functions can be utilized [30], this basic one is preferred in many practical applications. It has the demonstrated modelling advantage [31], as well as the associated practical attractiveness for on-line implementation [32]. By doing so, the variation of $x_{1:T}$ can be split into the variation with temporal connections and the variation without temporal connections. Usually, the first portion that is captured by the dynamic latent features will be treated as the informative trends of process data.

Unlike the system identification problem, there is no “input” signal in the transition function (2.1). Based on the objective of feature extraction, it is considered unnecessary to have the “input” signal in the dynamic model of latent features. The main reason is that there may not exist an explicit input in the historical process data, especially for the process with multiple control loops and a complex operational mechanism. Alternatively speaking, the available measurements may not necessarily be the “system input” to affect the transition of latent states. Thus, for the objective of feature extraction, available process measurements can either be treated as the observation x_t in (2.2), or become the regressors for the quality variable y_t:

$$y_t \sim g(y;\ x_t, s_t, \theta_{ob}). \tag{2.3}$$

In the first case of (2.2), the quality variable y_t will be treated as another projection from the latent feature. Theoretically, this strategy can also integrate y_t into the observation function (2.2). However, in reality, there could be differences between x_t and y_t, such as their sampling intervals. In the second case of (2.3), the latent feature is built to assist the regression analysis between the input $x_{1:T}$ and the output $y_{1:T}$. As the example in Chapter 6, the latent feature is used as time-variant parameters, and the corresponding dynamic model is learned for the possible switching behaviour.

In the remaining part of this section, two typical structures of (2.1) and (2.2) for feature extraction are introduced. Along with them, the Expectation-Maximization algorithm will be reviewed as a basic solution.

2.1.1 State Space Model

For latent states with the continuous supporting domain, $s_t \in (-\infty, +\infty)$, the state space model can be used by adopting a linear transition function:

$$s_t = a \cdot s_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, \sigma^2), \tag{2.4}$$

where the additive noise w_t is assumed to follow a Normal/Gaussian distribution "$\mathcal{N}$" with the zero-mean and the variance σ^2. The transition parameter is formed as $\theta_{tr} = \{a, \sigma^2\}$, and the probabilistic formation becomes

$$s_t \sim f(s;\ s_{t-1}, \theta_{tr}) = \mathcal{N}(s;\ a \cdot s_{t-1}, \sigma^2). \tag{2.5}$$

The coefficient parameter "a" determines the strength of this auto-correlation. For a system with inertia, this coefficient "a" is believed to be positive, which can propagate the similarities from s_{t-1} to s_t. For a stable system, this coefficient "a" is constrained inside the unit circle. The variance parameter σ^2 (also the precision parameter $\rho = \sigma^{-2}$) is always positive, reflecting the accuracy of this connection. If the mapping linearity between s_{t-1} and s_t is not so assured or the parameter "a" is inaccurate, a larger variance (or a smaller precision) can be used to reflect a higher uncertainty level.

By extending this transition model to the multivariate case, a basic dynamic feature extraction model can be formulated for continuous variables:

$$S_t = A \cdot S_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{2.6}$$

$$X_t = H \cdot S_t + v_t, \qquad v_t \sim \mathcal{N}(0, R). \tag{2.7}$$

As an extension of the coefficient parameter "a", the coefficient matrix A is usually formed as a diagonal matrix. The observation function is built on another linear mapping function with the projection matrix H and the noise term v_k. The extended noise terms w_k and v_k are modelled by the multivariate Normal distribution, which are determined by the covariance matrices Q and R, respectively.

Comparing to the SFA method mentioned above, rather than using the whole observation sample X_t, this decoding format allows X_t to have a certain "useless" part v_k. After describing v_k with a zero-mean distribution, it makes the latent sample S_t concentrate more on the remaining "useful" part. More importantly, the dynamic linear model can describe the latent feature more explicitly. In the deterministic SFA method, only required statistics are calculated to describe the latent feature, such as through the equations of (1.1), (1.2), and (1.3). In comparison, the explicit dynamic model (2.6) allows more detailed descriptions of the latent feature. For example, if the student-t distribution is selected to describe w_k, excess kurtosis can also be considered in the learning procedure. As another special advantage in Bayesian inference, this decoding format can implement the prior knowledge about the latent feature onto the transition parameter. In other words, by manipulating A and Q, a preferred set of $S_{1:T}$ can be defined before using the observation data.

Comparing to the probabilistic PCA model [22], rather than describing the latent feature as a collection of independent random variables, the above dynamic linear model uses A and Q to describe itself as an auto-correlated stochastic process. For the application of process data analysis, where the meaningful driving forces are often believed to have the temporal correlation, this description can provide more physical significance. From the statistical perspective, the independence assumption in probabilistic PCA restricts the distribution of latent feature $s_{1:T}$ to have the mutual independence:

$$p(s_{1:T};\ \theta_s) = \prod_{t=1}^{T} p(s_t). \tag{2.8}$$

Meanwhile, a transition function of latent states can relax this restriction. In the case of first order Markov model (2.6), the distribution of $s_{1:T}$ becomes

$$\begin{aligned} p(s_{1:T};\ \theta_s) &= p(s_1) \cdot \prod_{t=2}^{T} p(s_t \mid s_{1:t-1}) = p(s_1) \cdot \prod_{t=2}^{T} p(s_t \mid s_{t-1}) \\ &= p(s_1) \cdot \prod_{t=2}^{T} \frac{p(s_t, s_{t-1})}{p(s_{t-1})} = \frac{\prod_{t=2}^{T} p(s_t, s_{t-1})}{\prod_{t=2}^{T-1} p(s_t)}, \end{aligned} \tag{2.9}$$

which actually covers the case of (2.8) as one specific realization.

After increasing the modelling scope, the parameter estimation problem has naturally become more challenging than those for static modelling. That is, the parameters in (2.6) and (2.7) should be determined through their interaction with the estimation of $S_{1:T}$.

At this point, one possible approach for the maximum likelihood estimation, namely the Expectation-Maximization (EM) algorithm, can be adopted [33]. A brief review of the EM algorithm is presented here. In the following derivation, all the parameters in (2.6) and (2.7) are denoted with $\Theta = \{\theta_{tr}, \theta_{ob}\}$.

First, the objective function of EM algorithm, $\mathcal{L}\{q(S_{1:T}), \Theta\}$, can be formulated as a function of parameter Θ and a distribution of latent feature $q(S_{1:T})$:

$$\begin{aligned}
\mathcal{L}\{q(S_{1:T}), \Theta\} \equiv& \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta) + \mathcal{H}\{q(S_{1:T})\} \\
=& \int q(S_{1:T}) \ln p(X_{1:T}, S_{1:T} \mid \Theta) dS_{1:T} - \int q(S_{1:T}) \ln q(S_{1:T}) dS_{1:T} \\
=& \int q(S_{1:T}) \ln p(X_{1:T} \mid \Theta) dS_{1:T} - \int q(S_{1:T}) \ln \frac{q(S_{1:T})}{p(S_{1:T} \mid X_{1:T}, \Theta)} dS_{1:T} \\
=& \ln p(X_{1:T} \mid \Theta) - D_{KL}\{q(S_{1:T}) \parallel p(S_{1:T} \mid X_{1:T}, \Theta)\}, \qquad (2.10)
\end{aligned}$$

where $\mathbb{E}_{q(\cdot)}$ stands for the probability expectation with respect to distribution $q(\cdot)$, and $\mathcal{H}\{q(\cdot)\}$ stands for the entropy of distribution $q(\cdot)$. The Kullback-Leibler (K-L) divergence $D_{KL}\{q||p\}$ is used as a similarity measure between probability distributions. It has been proven that if $p(S_{1:T} \mid X_{1:T}, \Theta)$ varies continuously with Θ, this constructed objective function shares the same maxima Θ^* as the marginalized likelihood function [34]:

$$\arg\max_{\Theta} p(X_{1:T} \mid \Theta) \in \arg \max_{\{\Theta,\ q(S_{1:T})\}} \mathcal{L}\{q(S_{1:T}), \Theta\}. \qquad (2.11)$$

The EM algorithm, as a realization of coordinate ascent strategy for the objective function $\mathcal{L}\{q(S_{1:T}), \Theta\}$, consists of iterative updating steps for Θ and $q(S_{1:T})$ respectively:

$$\begin{aligned}
(E-step) \quad q(S_{1:T})^{(z)} &= \arg\max_{q(S_{1:T})} \mathcal{L}\{q(S_{1:T}), \Theta^{(z)}\} = p(S_{1:T} \mid X_{1:T}, \Theta^{(z)}), \qquad (2.12) \\
(M-step) \quad \Theta^{(z+1)} &= \arg\max_{\Theta} \mathcal{L}\{q(S_{1:T})^{(z)}, \Theta\} \\
&= \arg\max_{\Theta} \mathbb{E}_{q(S_{1:T})^{(z)}} \ln p(X_{1:T}, S_{1:T} \mid \Theta), \qquad (2.13)
\end{aligned}$$

where the superscript (z) stands for the index of each updating step. It should be noted that the E-step is based on the fact that $H\{q(S_{1:T})\}$ is irrelevant to parameter Θ, and the M-step is based on the fact that the minimum of $D_{KL}\{q||p\}$ is only achieved when $q(\cdot) = p(\cdot)$. In this sense, the latent feature will be learned jointly with the model parameters.

In the updating step (2.12) for the latent feature, given the observation $X_{1:T}$ and parameter $\Theta^{(z)}$, it yields the optimal solution for the posterior distribution of latent features. Based on its formulation in (2.9), this estimation can be simplified as estimating each joint distribution for consecutive latent states $p(s_t, s_{t-1} \mid X_{1:T}, \Theta^{(z)})$. For this state estimation problem, the algorithms for fixed interval smoother are applicable. Specifically, for the model shown in (2.6) and (2.7), the Rauch-Tung-Striebel (RTS) smoother can be applied. As a quick review, a nominal solution for $p(s_t, s_{t-1} \mid X_{1:T}, \Theta)$ is provided here through the sequential forward-backward algorithm.

- Forward path, starting with $p(s_0 \mid \Theta)$:

$$\begin{aligned} p(s_t \mid x_{1:t-1}, \Theta) &= \int p(s_t, s_{t-1} \mid x_{1:t-1}, \Theta)\ ds_{t-1} \\ &= \int p(s_t \mid s_{t-1}, \Theta)\ p(s_{t-1} \mid x_{1:t-1}, \Theta)\ ds_{t-1}, \end{aligned} \tag{2.14}$$

$$\begin{aligned} p(s_t \mid x_{1:t}, \Theta) &= \frac{p(s_t, x_t \mid x_{1:t-1}, \Theta)}{p(x_t \mid x_{1:t-1}, \Theta)} \\ &= \frac{p(x_t \mid s_t, \Theta)\ p(s_t \mid x_{1:t-1}, \Theta)}{\int p(x_t \mid s_t, \Theta)\ p(s_t \mid x_{1:t-1}, \Theta)\ ds_t}. \end{aligned} \tag{2.15}$$

- Backward path, starting with $p(s_T \mid x_{1:T}, \Theta)$ from the final step of forward path:

$$\begin{aligned} p(s_t, s_{t-1} \mid x_{1:T}, \Theta) &= p(s_t \mid x_{1:T}, \Theta)\ p(s_{t-1} \mid s_t, x_{1:T}, \Theta) \\ &= p(s_t \mid x_{1:T}, \Theta)\ p(s_{t-1} \mid s_t, x_{1:t-1}, \Theta) \\ &= p(s_t \mid x_{1:T}, \Theta)\ \frac{p(s_t, s_{t-1} \mid x_{1:t-1}, \Theta)}{p(s_t \mid x_{1:t-1}, \Theta)} \\ &= p(s_t \mid x_{1:T}, \Theta)\ \frac{p(s_t \mid s_{t-1}, \Theta)\ p(s_{t-1} \mid x_{1:t-1}, \Theta)}{p(s_t \mid x_{1:t-1}, \Theta)}, \end{aligned} \tag{2.16}$$

$$p(s_{t-1} \mid x_{1:T}, \Theta) = \int p(s_t, s_{t-1} \mid x_{1:T}, \Theta)\ ds_t. \tag{2.17}$$

On the basis of above derivations, the E-step algorithm can be formulated not only for the latent feature in (2.6) and (2.7), but also for other dynamic features with more sophisticated realizations of (2.1) and (2.2).

In the updating step (2.13) of model parameters, namely the M-step, detailed methods should be subjective to the problem at hand. For example, based on the formulation of (2.6) and (2.7), the transition matrix A, the emission matrix H, and the covariance matrices Q and R can all be optimized based on the quadratic function $\ln p(X_{1:T}, S_{1:T} \mid \Theta)$:

$$\begin{aligned} &\max_{\Theta}\ \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta) \\ \Leftrightarrow\ &\max_{\{A,Q,H,R\}}\ \mathbb{E}_{q(S_{1:T})}\{-(T-1)\cdot \ln|Q| - \sum_{t=2}^{T}[S_t - A\cdot S_{t-1}]'\ Q^{-1}\ [S_t - A\cdot S_{t-1}] \\ &\qquad - T\cdot \ln|R| - \sum_{t=2}^{T}[X_t - H\cdot S_t]'\ R^{-1}\ [X_t - H\cdot S_t]\}. \end{aligned} \tag{2.18}$$

In this formulation, the expectation $\mathbb{E}_{q(S_{1:T})}$ can be achieved from (2.17) and (2.16). It should be noted that given $q(S_{1:T})$, the optimization step of each model parameter is independent from each other.

2.1.2 Hidden Markov Model

For latent states with a finite set of possible values, $s_t \in \{1, ..., k\}$, the hidden Markov model is usually applied via a transition model:

$$\Pr(s_t = i \mid s_{t-1} = j) = a_{i,j}, \quad \forall \; i, j \in \{1, ..., k\}. \tag{2.19}$$

Different from the probability density function $p(\cdot)$, here $\Pr(\cdot)$ is used for the probability mass function. By enumerating every combination of two consecutive states, the transition parameter $\theta_{tr} = A_{k \times k} = [a_{i,j}]_{i,j=1,\cdots,k}$ can provide a comprehensive description for the dynamic behaviour of $s_{1:T}$, such as calculating its stationary distribution $p(s_\infty)$ and return time [35]. In the corresponding probabilistic formation, the categorical distribution "*Cat*" can be used [23]:

$$s_t \sim f(s;\; s_{t-1}, \theta_{tr}) = Cat(s;\; A_{:,s_{t-1}}) = \prod_{i=1}^{k} a_{i,s_{t-1}}{}^{[s=i]}, \tag{2.20}$$

where $[\cdot]$ is the Iverson bracket. Given s_{t-1}, a column vector of the transition matrix, $A_{:,s_{t-1}} = [a_{i,s_{t-1}}]_{i=1,\cdots,k}$, can be selected as the k-dimensional parameter vector for this categorical distribution. Based on its definition in (2.19), each column of this transition parameter $A_{k \times k}$ must regulate its elements to have the unit-sum. Comparing to the state space model that has $\theta_{tr} = \{A, Q\}$, this transition function only uses $\theta_{tr} = A_{k \times k}$ to determine $p(s_{1:T})$.

In the observation function, each possible value of s_t should be assigned with an observation probability, which is based on its interaction with the observation sample x_t. If x_t also has a finite set of possible values $x_t \in \{1, ..., p\}$, the observation function can be formed as

$$\Pr(x_t = i \mid s_t = j) = h_{i,j}, \quad \forall \; i \in \{1, ..., p\}, \; j \in \{1, ..., k\}. \tag{2.21}$$

Similar to matrix $A_{k \times k}$, the observation matrix $\theta_{ob} = H_{p \times k} = [h_{i,j}]_{i=1,\cdots,p;j=1,\cdots,k}$ has the constraint of unit-sum for each column vector. However, process observations are normally continuous-valued. In this case, s_t can be used as a mode identity variable to determine the distribution of observed samples. Generally, two kinds of observation functions can be proposed to relate between the discrete data and the continuous data. In the first case, the observation sample X_t is assumed to have a Normal distribution, where the mean and the variance are determined according to s_t:

$$X_t \sim \mathcal{N}(X;\; m^{(j)}, R^{(j)}), \qquad \textit{if} \;\; s_t = j. \tag{2.22}$$

In the other case, the connection between the observation sample X_t and the quality variable y_t can be determined by the latent state s_t:

$$y_t \sim \mathcal{N}(y;\ H^{(j)} \cdot X_t, R^{(j)}), \qquad \textit{if}\ \ s_t = j. \tag{2.23}$$

In these realizations, the observation function parameters usually contain k parallel sets, such as k vectors of $[m^{(j)}]_{j=1,\cdots,k}$ to describe the possible mean vector and k matrices of $[R^{(j)}]_{j=1,\cdots,k}$ to describe the possible covariance matrix. Based on the selecting criterion $s_t = j$, the probability connection between the continuous X_t and the discrete s_t is built on these parallel sets of parameters. To proceed with the probabilistic inference, a unified description of the observation function is provided here as a summary of the above discussion:

$$p(X_t \mid s_t) = \prod_{j=1}^{k} p(X_t \mid s_t = j)^{[s_t=j]}, \tag{2.24}$$

where the Iverson bracket $[\cdot]$ is used.

The parameter estimation for $s_{1:T}$ with discrete value is similar to that for the continuous case. According to the EM algorithm, the objective function can be transformed from (2.10), which converts $\mathbb{E}_{q(\cdot)}$, $\mathcal{H}\{q(\cdot)\}$, and $D_{KL}\{q||p\}$ to the objective function for the discrete variable. For example, the K-L divergence $D_{KL}\{q||p\}$ is converted to

$$D_{KL}\{q(s_t)\ ||\ p(s_t)\} = \sum_{i=1}^{k} q(s_t = i) \ln \frac{q(s_t = i)}{p(s_t = i)}. \tag{2.25}$$

Thus, the iterative updating steps can be generalized from (2.12) and (2.13):

$$(E - step) \quad q(s_{1:T})^{(z)} = p(s_{1:T} \mid X_{1:T}, \Theta^{(z)}), \tag{2.26}$$

$$(M - step) \qquad \Theta^{(z+1)} = \arg\max_{\Theta}\ \mathbb{E}_{p(s_{1:T}|X_{1:T},\Theta^{(z)})} \ln p(X_{1:T}, S_{1:T} \mid \Theta), \tag{2.27}$$

where the superscript (z) stands for the index of each learning step. Similarly, the E-step (2.26) estimates the posterior distribution for the sequenced latent variables, and the M-step (2.27) optimizes the model parameters based on a given distribution of latent variables.

In the E-step, the algorithm (2.14) $-$ (2.16) is still applicable when learning $q(s_{1:T})^{(z)}$. Besides this sequential algorithm, a parallel forward-backward algorithm is reviewed here, which is more efficient in learning the discrete state s_t. To estimate the posterior $p(s_{1:T} \mid X_{1:T}, \Theta^{(z)})$ with the form of (2.9), each elementary distribution can be factorized as

$$p(s_t \mid X_{1:T}, \Theta) = \frac{p(X_{t+1:T}, s_t \mid X_{1:t}, \Theta)}{p(X_{t+1:T} \mid X_{1:t}, \Theta)} \propto p(X_{t+1:T} \mid s_t, \Theta)\ p(s_t \mid X_{1:t}, \Theta), \tag{2.28}$$

$$p(s_t, s_{t-1} \mid X_{1:T}, \Theta) = \frac{p(X_{t:T}, s_t, s_{t-1} \mid X_{1:t-1}, \Theta)}{p(X_{t:T} \mid X_{1:t-1}, \Theta)}$$

$$\propto p(X_{t+1:T} \mid s_t, \Theta)\ p(X_t \mid s_t, \Theta)\ p(s_t \mid s_{t-1}, \Theta)\ p(s_{t-1} \mid X_{1:t-1}, \Theta), \tag{2.29}$$

where the Markov property and the chain rule are used. On the basis of above factorization, the parallel forward-backward algorithm is presented as follows.

- Forward path, starting from $p(s_0 \mid \Theta) = 1$:

$$p(s_t \mid X_{1:t}, \Theta) \propto p(y_t \mid s_t, \Theta) \sum_{s_t} p(s_t \mid s_{t-1}, \Theta)\ p(s_{t-1} \mid X_{1:t-1}, \Theta). \tag{2.30}$$

- Backward path, starting from $p(X_{T+1:T} \mid s_T, \Theta) = 1$:

$$p(X_{t+1:T} \mid s_t, \Theta) = \sum_{s_{t+1}} p(X_{t+2:T} \mid s_{t+1}, \Theta)\ p(X_{t+1} \mid s_{t+1}, \Theta)\ p(s_{t+1} \mid s_t, \Theta). \tag{2.31}$$

Here, the summation $\sum_{s_t}$ enumerates all possible values of s_t. Because of its finite supporting domain, the discrete s_t here has a simplified normalization procedure in comparison with the continuous case. As a consequence, the backward path becomes independent of the forward path. In practice, this property also facilitates a parallel estimation algorithm for latent features, which increases the computational efficiency.

In the optimization step for $\Theta^{(z+1)}$, namely the M-step, detailed algorithms depend on specified models [36]. Typically, for the observation function parameters, the learning algorithm is determined by the selected observation function, such as that introduced in (2.21) – (2.23). As for the transition function parameters $\theta_{tr} = [a_{i,j}]_{i,j=1,\cdots,k}$, because of their independence from θ_{ob} in the M-step, it is usually optimized individually. For example, an optimization can be performed as

$$a_{i,j}{}^{(z+1)} = \frac{\sum_{t=2}^{T} \Pr(s_t = i,\ s_{t-1} = j)}{\sum_{t=2}^{T} \Pr(s_{t-1} = j)}, \tag{2.32}$$

where the numerators come from (2.29) and the denominators come from (2.28).

2.1.3 Practical Deficiency

Two structures of dynamic linear models and their inference algorithms have been introduced so far. However, from both modelling and inference perspective, the above solutions are not yet sufficient for complete process data analysis.

From the modelling point of view, these structures are too general to be useful for extracting representative latent features. In Chapter 1, it has been discussed that the latent features should be formulated to represent certain process variations. For example, Figure 1.3 shows that the "meaningful" variations should have large inertia for certain applications. Besides this slowness preference, the informative features may also contain either stationary or non-stationary behaviours. In the structures mentioned above, however, the latent features are built for almost all types of variations. In order to learn certain variations with these general structures, a large number of observations, with repeated dynamic behaviours, are required for parameter estimation. Unfortunately, in practice, not all dynamic behaviour is repeated frequently. Thus, the model structure should be more specific for specific trends or modelling objectives, where some constraints and preferences are desired.

From the inference point of view, the parameter may not be optimally identifiable from the likelihood function directly. To have the shared optimal point with the maximized likelihood function $p(X_{1:T} \mid \Theta)$, the EM algorithm requires a condition [34]: $p(S_{1:T} \mid X_{1:T}, \Theta)$ must vary continuously with Θ. In the structures mentioned above, this condition may not be satisfied. An intuitive example can be found in the case of multiple realizations of a state space model [37]. Through the similarity transformation, multiple sets of model parameters can yield the same marginalized likelihood: $p(X_{1:T} \mid \Theta^{(1)}) = p(X_{1:T} \mid \Theta^{(2)})$. Thus, a structural regulation is essential for the uniqueness of the estimation. As a well-known example of the structural regulation, the probabilistic PCA [22] forces the covariance matrix R in (2.7) to be a diagonal matrix, as well as the latent features are restricted to be mutually independent. In the following chapters, several structural constraints will be illustrated based on specific application scenarios. Besides that, a Bayesian inference framework is introduced in the remainder of this chapter, which improves the uniqueness property of parameter estimation through the prior distribution $p(\Theta)$.

2.2 Variational Bayesian Inference

After introducing model structures for dynamic feature extraction, a general probability formulation can be written through the following conditional distributions:

$$p(S_{1:T} \mid \theta_{tr}) \quad and \quad p(X_{1:T} \mid S_{1:T}, \theta_{ob}). \tag{2.33}$$

It should be noted that the case with output variable related observation function is also included, since

$$p(y_{1:T} \mid X_{1:T}, S_{1:T}, \theta_{ob}) = p((y_{1:T} \mid X_{1:T}) \mid S_{1:T}, \theta_{ob}) = p(\hat{X}_{1:T} \mid S_{1:T}, \theta_{ob}).$$

Thus, either the observation $X_{1:T}$ or the observation pairs $\hat{X}_{1:T} = y_{1:T} \mid X_{1:T}$ can be determined according to the latent feature $S_{1:T}$. In this section, the model (2.33) is used as an introduction to the variational Bayesian inference.

As a Bayesian approach, it first assigns the model parameters $\Theta = \{\theta_{tr}, \theta_{ob}\}$ with a prior distribution:

$$p(\Theta \mid \eta) = p(\theta_{tr} \mid \eta_{tr}) \cdot p(\theta_{ob} \mid \eta_{ob}). \tag{2.34}$$

Here, $\eta = \{\eta_{tr}, \eta_{ob}\}$ is the set of hyper-parameters, representing the prior belief about the transition function and the observation function. According to this prior distribution and the marginalized likelihood:

$$p(X_{1:T} \mid \Theta) = \int p(X_{1:T} \mid S_{1:T}, \theta_{ob}) \cdot p(S_{1:T} \mid \theta_{tr}) \ dS_{1:T}, \tag{2.35}$$

the posterior distribution of model parameters can be formulated through

$$p(\Theta \mid X_{1:T}, \eta) = \frac{p(X_{1:T} \mid \Theta) \cdot p(\Theta \mid \eta)}{p(X_{1:T} \mid \eta)} = \frac{p(X_{1:T} \mid \Theta) \cdot p(\Theta \mid \eta)}{\int p(X_{1:T} \mid \Theta) \cdot p(\Theta \mid \eta) \ d\Theta}. \tag{2.36}$$

With an explicit expression of this distribution, the optimal estimate of the parameters, as well as the estimate of uncertainties can be obtained. Unfortunately, the derivation for this exact posterior is usually computationally prohibited. Especially for the case with latent feature $S_{1:T}$, the calculation of likelihood term in (2.35) and the denominator in (2.36) will contain the inseparable integration between $S_{1:T}$ and Θ. Without further simplification, such as letting $R \to 0$ in model (2.7) [25], calculating this integration is generally an NP-hard problem [23]. Furthermore, if the prior distribution (2.34) is not a conjugate prior to the likelihood (2.35), the exact posterior distribution may not belong to a known distribution family. As a conventional solution, the maximum-a-posterior (MAP) method can provide the point-estimation for Θ. However, on the other hand, it fails to evaluate the estimation uncertainties and cannot use them in the inference procedure. In this thesis, the variational Bayesian inference is introduced as a more advanced learning strategy.

According to the definition of "variational", this method optimizes a proposal distribution of unknown (to-be-estimated) variables to approximate the true posterior distribution:

$$q(S_{1:T}, \Theta) \to p(S_{1:T}, \Theta \mid X_{1:T}, \eta). \tag{2.37}$$

The proposal distribution is usually selected to have an explicit form, which makes it easier to estimate this proposal distribution than to calculate the integrations. Thus, the inferencing objective can be transformed to the minimization of the distance between the two distributions in (2.37). From the probability point of view, several divergence functions are available to describe this distance. Here, the K-L divergence $D_{KL}\{q||p\}$ is used for the variational Bayesian inference:

$$D_{KL}\{q(S_{1:T}, \Theta) \ || \ p(S_{1:T}, \Theta \mid X_{1:T}, \eta)\}$$

$$= \int q(S_{1:T}, \Theta)\ \ln \frac{q(S_{1:T}, \Theta)}{p(S_{1:T}, \Theta \mid X_{1:T}, \eta)}\ dS_{1:T}\ d\Theta. \tag{2.38}$$

As an asymmetric function, $D_{KL}\{q||p\}$ is selected instead of $D_{KL}\{p||q\}$ because of the advantages from using the explicit properties of $q(S_{1:T}, \Theta)$. In most of other divergence functions, including $D_{KL}\{p||q\}$, there are additional challenges owing to the integration operation of (2.38).

According to the chain rule, the marginal distribution of observed data $p(X_{1:T} \mid \eta)$ only depends on the selected model structure and the hyper-parameter η, which are irrelevant to the proposal distribution $q(S_{1:T}, \Theta)$. The variational Bayesian inference uses this concept to convert the minimization problem of (2.38) into a maximization problem:

$$\begin{aligned}
\ln p(X_{1:T} \mid \eta) &= \int q(S_{1:T}, \Theta)\ \ln p(X_{1:T} \mid \eta)\ dS_{1:T}\ d\Theta \\
&= \int q(S_{1:T}, \Theta)\ \ln \frac{p(X_{1:T}, S_{1:T}, \Theta \mid \eta)}{p(S_{1:T}, \Theta \mid X_{1:T}, \eta)}\ dS_{1:T}\ d\Theta \\
&= \int q(S_{1:T}, \Theta)\ \ln \frac{p(X_{1:T}, S_{1:T}, \Theta \mid \eta)}{q(S_{1:T}, \Theta)}\ dS_{1:T}\ d\Theta \\
&\qquad + \int q(S_{1:T}, \Theta)\ \ln \frac{q(S_{1:T}, \Theta)}{p(S_{1:T}, \Theta \mid X_{1:T}, \eta)}\ dS_{1:T}\ d\Theta \\
&= \mathcal{L}_{q(S_{1:T},\Theta)} + D_{KL}\{q(S_{1:T}, \Theta)\ ||\ p(S_{1:T}, \Theta \mid X_{1:T}, \eta)\}.
\end{aligned} \tag{2.39}$$

Because of the constant value taken by $\ln p(X_{1:T} \mid \eta)$, minimizing the K-L divergence in (2.38) is equivalent to maximizing $\mathcal{L}_{q(S_{1:T},\Theta)}$. Since the K-L divergence is always non-negative, $\mathcal{L}_{q(S_{1:T},\Theta)}$ becomes a lower bound measure for $\ln p(X_{1:T} \mid \eta)$. It is usually called the variational lower bound, which is a function of the proposal distribution:

$$\begin{aligned}
\mathcal{L}_{q(S_{1:T},\Theta)} &= \int q(S_{1:T}, \Theta)\ \ln \frac{p(X_{1:T}, S_{1:T}, \Theta \mid \eta)}{q(S_{1:T}, \Theta)}\ dS_{1:T}\ d\Theta \\
&= \int q(S_{1:T}, \Theta)\ \ln p(X_{1:T}, S_{1:T}, \Theta \mid \eta)\ dS_{1:T}\ d\Theta + \mathcal{H}\{q(S_{1:T}, \Theta)\}.
\end{aligned} \tag{2.40}$$

In this expression, $\mathcal{H}\{q(\cdot)\}$ is used in the same way as that in (2.10), and the joint likelihood of observations, latent features, and model parameters can be established according to model (2.33):

$$p(X_{1:T}, S_{1:T}, \Theta \mid \eta) = p(X_{1:T} \mid S_{1:T}, \theta_{ob}) \cdot p(S_{1:T} \mid \theta_{tr}) \cdot p(\theta_{ob} \mid \eta_{ob}) \cdot p(\theta_{tr} \mid \eta_{tr}). \tag{2.41}$$

To maximize the variational lower bound $\mathcal{L}_{q(S_{1:T},\Theta)}$, multiple optimization methods can be adopted from the variational calculus. In order to obtain an efficient solution, mean field assumption [38] is generally applied to reduce the computational load in the variational Bayesian inference process. This assumption decomposes the proposal distribution to a product of independent proposal distributions. Usually, this factorization procedure is determined by the probabilistic dependencies in the likelihood function. Based on the likelihood in (2.41), the mean field assumption gives

$$q(S_{1:T}, \Theta) = q(S_{1:T}) \cdot q(\theta_{ob}) \cdot q(\theta_{tr}). \tag{2.42}$$

On the basis of this introduced probabilistic independence, the coordinate ascent strategy can be applied to update the individual proposal distributions. For example, by keeping $q(\theta_{ob})$ and $q(\theta_{tr})$ unchanged, the variational lower bound $\mathcal{L}_{q(S_{1:T},\Theta)}$ can be simplified as another K-L divergence, which only has one unknown part $q(S_{1:T})$:

$$\begin{aligned}
&\mathcal{L}_{q(S_{1:T})} = \mathcal{L}_{q(S_{1:T},\Theta)} \mid q(\theta_{ob}),\ q(\theta_{tr}) \\
=& \int q(S_{1:T})\ q(\theta_{ob})\ q(\theta_{tr}) \ln p(X_{1:T}, S_{1:T}, \Theta \mid \eta)\ dS_{1:T}\ d\theta_{ob}\ d\theta_{tr} + \mathcal{H}\{q(S_{1:T})\} + const. \\
=& \int q(S_{1:T}) \left[\int q(\theta_{ob}) \ln p(X_{1:T} \mid S_{1:T}, \theta_{ob})\ d\theta_{ob} + \int q(\theta_{tr}) \ln p(S_{1:T} \mid \theta_{tr})\ d\theta_{tr} \right] dS_{1:T} \\
&\qquad + \mathcal{H}\{q(S_{1:T})\} + const. \\
=& - D_{KL}\{q(S_{1:T}) \mid\mid c \cdot e^{\mathbb{E}_{q(\theta_{ob})} \ln p(X_{1:T}|S_{1:T},\theta_{ob}) + \mathbb{E}_{q(\theta_{tr})} \ln p(S_{1:T}|\theta_{tr})}\} + const..
\end{aligned} \tag{2.43}$$

In this derivation, *const.* stands for the constant value, and the normalizing factor c is used to transfer the exponential term to a probability density function. As a result, the optimal $q(S_{1:T})$ can be derived given $q(\theta_{ob})$ and $q(\theta_{tr})$:

$$\ln q^*(S_{1:T}) = \mathbb{E}_{q(\theta_{ob})} \ln p(X_{1:T} \mid S_{1:T}, \theta_{ob}) + \mathbb{E}_{q(\theta_{tr})} \ln p(S_{1:T} \mid \theta_{tr}) + const.. \tag{2.44}$$

Similarly, the updating equation for $q(\theta_{tr})$ and $q(\theta_{ob})$ can be derived as

$$\ln q^*(\theta_{tr}) = \mathbb{E}_{q(S_{1:T})} \ln p(S_{1:T} \mid \theta_{tr}) + \ln p(\theta_{tr} \mid \eta_{tr}) + const., \tag{2.45}$$

$$\ln q^*(\theta_{ob}) = \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T} \mid S_{1:T}, \theta_{ob}) + \ln p(\theta_{ob} \mid \eta_{ob}) + const.. \tag{2.46}$$

2.2.1 Comparison with Expectation-Maximization Algorithm

Based on the equations (2.44) – (2.46), an iterative algorithm, namely variational Bayesian Expectation-Maximization (VBEM) algorithm, can be developed. It can be observed that this algorithm has a similar structure as that of the EM algorithm. For the dynamic feature extraction problem, a comparison between these two iterative algorithms is presented in this section.

First, the objective functions of EM algorithm (2.10) and VBEM algorithm (2.40) can be compared through

$$\mathcal{L}^{(EM)}_{q(S_{1:T}),\Theta} = \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta) + \mathcal{H}\{q(S_{1:T})\},$$

$$
\begin{aligned}
\mathcal{L}^{(VBEM)}_{q(S_{1:T},\Theta)} &= \mathbb{E}_{q(S_{1:T},\Theta)} \ln p(X_{1:T}, S_{1:T}, \Theta \mid \eta) + \mathcal{H}\{q(S_{1:T}, \Theta)\} \\
&= \mathbb{E}_{q(S_{1:T},\Theta)} \ln p(X_{1:T}, S_{1:T} \mid \Theta) + \mathcal{H}\{q(S_{1:T})\} - D_{KL}\{q(\Theta) \mid\mid p(\Theta \mid \eta)\}.
\end{aligned}
$$

It should be noted that $\mathcal{L}^{(VBEM)}_{q(S_{1:T},\Theta)}$ has been further simplified based on the independence assumption in (2.42). There are two differences that can be observed: (1) both $q(S_{1:T})$ and $q(\Theta)$ are used in the VBEM algorithm to calculate the expectation of joint log-likelihood term $\ln p(X_{1:T}, S_{1:T} \mid \Theta)$, instead of only using $q(S_{1:T})$ as in the EM algorithm; (2) the VBEM algorithm adds an additional K-L divergence between $q(\Theta)$ and its prior distribution $p(\Theta \mid \eta)$. In the following discussion, the advantages of VBEM are illustrated based on these two differences.

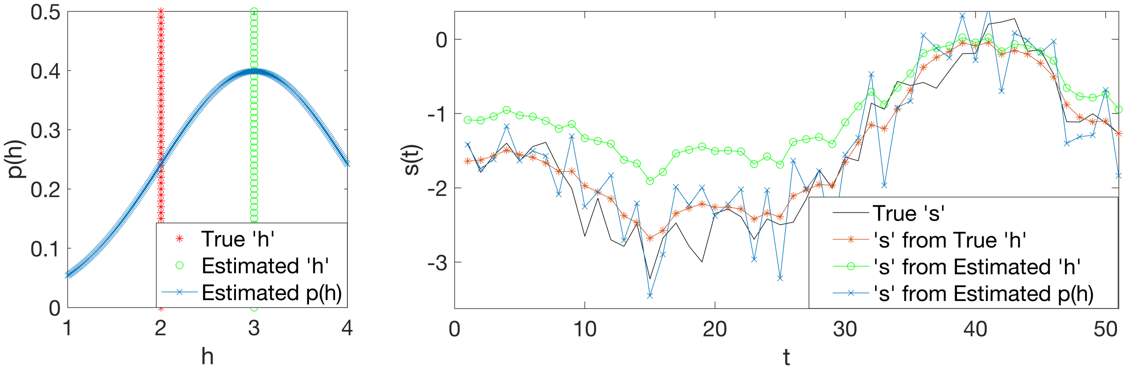


Figure 2.1: Advantage of Using Parameter Distributions

The first extension of VBEM can improve the learning performance of $q(S_{1:T})$ in the iterative steps. In the VBEM method, the model parameter Θ is described by a probability distribution, which contains much more information than a single point estimation. By using this distributional description of Θ, the E-step can be generally improved. As an example, Figure 2.1 compares the updated $q(S_{1:T})$ between EM and VBEM. In this example, one-dimensional latent feature and one-dimensional observation are generated from (2.6) and (2.7) accordingly. Specifically, the coefficient parameter in the observation function is set as $h = 2$, where "h" is used for one-dimensional H. As an intermediate result in the iterative algorithm, the estimated "h" can be assumed to have a distance from the true "h". Here in the left plot of Figure 2.1, the intermediate result of EM algorithm (2.13) is assumed to be "h" $= 3$, and the intermediate result of VBEM algorithm (2.46) is assumed to be $q(h) = \mathcal{N}(h; 3, 1)$. In the right plot of Figure 2.1, the estimation results from the following E-step are compared, where the mean values of $q(S_{1:T})$ are plotted. It can be observed that the estimate from VBEM (the blue curve with crosses) is closer to the optimal estimation (the red curve with dots) than the estimate from EM (the green curve with circles). Intuitively, it is because that the distribution $q(h)$ has included the true "h" with a certain possibility, which can contribute to learning the correct $s_{1:T}$. Technically speaking, the variance in $q(h)$ has enlarged the uncertainty of the observation function so that the state estimation algorithm has less confidence in the observation samples. Therefore, $q(s_{1:T})$ becomes more robust to the inaccurate observation parameter.

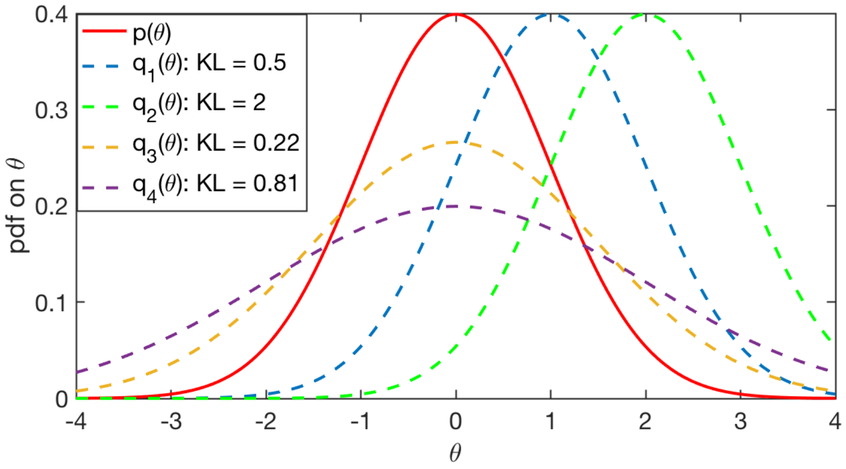


Figure 2.2: Kullback-Leibler Divergences between Normal Distributions

The second extension of VBEM is an implementation of the Bayes rule, which uses $p(\Theta \mid \eta)$ to regulate the learning of Θ. It is materialised through the subtracted K-L divergence $D_{KL}\{q(\Theta) \mid\mid p(\Theta \mid \eta)\}$. Conceptually, the proposal distribution that is far from the prior distribution will be penalized more in $\mathcal{L}^{(VBEM)}_{q(S_{1:T},\Theta)}$. As a visualization, some estimated Normal distributions are plotted in Figure 2.2. The distance between these dashed proposal distributions and a prior distribution $p(\theta) = \mathcal{N}(\theta; 0, 1)$ is shown in the legends. Other than introducing this quantitative measure about dissimilarity, this K-L divergence $D_{KL}\{q(\Theta) \mid\mid p(\Theta \mid \eta)\}$ is also useful when implementing the constraint on model parameters. For example, if some parameters are defined as positive values only, the prior distribution will have $p(\theta < 0) = 0$. According to the definition of K-L divergence in (2.25), any proposal distribution with non-zero probability for $\theta < 0$ will make $D_{KL}\{q(\theta) \mid\mid p(\theta)\}$ towards the positive infinity:

$$\begin{aligned} D_{KL}\{q(\theta) \mid\mid p(\theta)\} &= \int_{-\infty}^{0} q(\theta) \ln \frac{q(\theta)}{p(\theta)} d\theta + \int_{0}^{+\infty} q(\theta) \ln \frac{q(\theta)}{p(\theta)} d\theta \\ &= +\infty + \int_{0}^{+\infty} q(\theta) \ln \frac{q(\theta)}{p(\theta)} d\theta. \end{aligned} \tag{2.47}$$

From the optimization point of view, this constraint of model parameters has been integrated into the objective function, and has been transferred to the updated proposal distribution $q(\theta)$.

2.2.2 Limitations of Variational Bayesian Inference

In the above introduction of variational Bayesian inference, the mean field assumption in (2.42) is necessary for tractability of the inference procedure. Without this assumption, the iterative updating is not amenable. However, this assumption also introduces a gap between the optimal proposal distribution and the exact posterior distribution. According to the Bayes rule, even if θ_1 and θ_2 are independent in their prior distribution, the posteriors $p(\theta_1 \mid X)$ and $p(\theta_2 \mid X)$ are mutually dependent given the likelihood $p(X \mid \theta_1, \theta_2)$. Thus, the divergence in (2.38) can never become zero after introducing the independence assumption. As an intuitive example, if the exact posterior is a multivariate Normal distribution with a non-zero covariance term, $Cov(\theta_1, \theta_2) \neq 0$, the two independent distributions $q(\theta_1)$ and $q(\theta_2)$ can never converge to this posterior.

Without the mean field assumption (2.42), the proposal distribution $q(S_{1:T}, \Theta)$ has the ability to reach the exact posterior distribution. However, from practical consideration, this assumption (or approximation) is necessary. First, the resultant model is usually expected to be used for the prediction of future data. In particular, the model parameters $\Theta = \{\theta_{tr}, \theta_{ob}\}$ will be used to extract features from future observations. However, $q(S_{1:T})$ is learned only for the past (training) samples. The proposed feature extraction method should use Θ independently from $q(S_{1:T})$. Comparing to separating Θ from $q(S_{1:T}, \Theta)$ after the optimization procedure, directly optimizing the independent distributions will be more efficient for model interpretation. Also, from the computational point of view, without the independence assumption for $q(S_{1:T})$ and $q(\Theta)$, the complexity of updating $q(\Theta)$ will also increase with the number of training samples. Thus, the independence assumption is necessary for the variational Bayesian inference in order to yield an implementable result.

2.3 Particle Based Algorithms

One important advantage of variational Bayesian inference is the flexibility of selecting proposal distributions. To achieve the approximation (2.37), any type of probability distributions may be used as the proposal distribution in (2.42). In conventional applications, the proposal distributions are usually selected for computational convenience. For example, when the conjugate prior is selected, the proposal distribution will be in the same distribution family [39]. However, as discussed in Chapter 1, the prior distribution in this thesis will not necessarily be limited by the conjugate prior. There are two main reasons: (1) the likelihood formulation of a specified feature extraction model may not always have a conjugate prior, (2) the prior distribution will be determined by certain constraints and preferences for the specific problem.

To have more flexibility for modelling process data, the particle-based representation is introduced for the proposal distribution:

$$q_N(\theta) = \frac{1}{N}\sum_{j=1}^{N} \delta(\theta = \theta^{(j)}), \tag{2.48}$$

where N is the number of particles, and $\delta(\cdot)$ is the Dirac delta function. In this section, three particle-based inference approaches are reviewed.

2.3.1 Sampling Algorithms

Based on the representation in (2.48), most applications of a probability distribution can be realized through the sampled particles. Specifically, if a sufficient number of particles $\theta^{(1,\cdots,N)}$ have been sampled from $q(\theta)$, any $q(\theta)$ related integration can be approximated by the particle representation $q_N(\theta)$:

$$\frac{1}{N}\sum_{j=1}^{N} f(\theta^{(j)}) \overset{N\to\infty}{\longrightarrow} \int f(\theta)\ q(\theta)\ d\theta. \tag{2.49}$$

For example, in the variational updating step (2.44), the expectation $\mathbb{E}(\theta)$ and $\mathbb{E}(\theta \cdot \theta')$ can be approximated by using $q_N(\theta)$. According to the central limit theorem, the approximation error of (2.49) will follow a converged distribution:

$$[\mathbb{E}_{q_N(\theta)} f(\theta) - \mathbb{E}_{q(\theta)} f(\theta)] \sim \mathcal{N}(0, \frac{\sigma_f^2}{N}), \tag{2.50}$$

as long as the variance of $f(\theta)$ is not diverged with respect to $q(\theta)$:

$$\sigma_f^2 = \mathbb{E}_{q(\theta)} f^2(\theta) - [\mathbb{E}_{q(\theta)} f(\theta)]^2 < +\infty. \tag{2.51}$$

To achieve this convenience of approximating $q_N(\theta)$, it is important to ensure that a sufficient number of particles are generated exactly from $q(\theta)$.

It is known that given the cumulative distribution function $F(\theta)$ of a probability distribution, sampling from this distribution can be easily achieved. In this approach, the inverse function F^{-1} is used to yield the target samples by manipulating samples of the uniform distribution "U" [23]:

$$\theta = F^{-1}(u) \sim dF(\theta), \qquad if : u \sim U(0,1).$$

However, for our inference tasks, the target distribution cannot be known by its cumulative distribution function. For example, the normalizing constants in (2.45) and (2.46) are generally unavailable. To solve this problem, the importance sampling technique is introduced.

In the importance sampling algorithm, the particle based distribution $q_N(\theta)$ is extended to include a weight for each particle:

$$q_N(\theta) = \sum_{j=1}^{N} w^{(j)} \; \delta(\theta = \hat{\theta}^{(j)}). \tag{2.52}$$

To be consistent with (2.48), these weights are constrained to have $\sum_{j=1}^{N} w^{(j)} = 1$. By using these additional parameters, it is not required that the particles are generated from $q(\theta)$. For instant, a distribution $p_0(\theta)$ can be used if eligible samples $\hat{\theta}^{(1,\cdots,N)}$ can be drawn from it. Then, the difference between $p_0(\theta)$ and $q(\theta)$ will be accounted by the weights $w^{(1,\cdots,N)}$, as long as $q(\theta)$ can be evaluated at any specified point of θ. Specifically, these weights will be calculated from

$$\sum_{j=1}^{N} w^{(j)} = 1, \quad \forall \, j \, : \; w^{(j)} \propto q(\hat{\theta}^{(j)}) \; / \; p_0(\hat{\theta}^{(j)}). \tag{2.53}$$

With these weighted particles, the integration procedure becomes

$$\sum_{j=1}^{N} w^{(j)} \; f(\hat{\theta}^{(j)}) \overset{N\to\infty}{\longrightarrow} \int f(x) \; w(\theta) \; p_0(\theta) \; d\theta = \int f(x) \; q(\theta) \; d\theta. \tag{2.54}$$

It can be observed that a weighting function is defined in the importance sampling: $w(\theta) = \frac{q(\theta)}{p_0(\theta)}$. While applied in variational Bayesian inference, for example to estimate $q(\theta_{tr})$, this weighting function can be separated from the variational updating step (2.45):

$$\begin{aligned} \ln q(\theta_{tr}) &= \ln w(\theta) + \ln p_0(\theta) \\ &= \mathbb{E}_{q(S_{1:T})} \ln p(S_{1:T} \mid \theta_{tr}) + p(\theta_{tr} \mid \eta_{tr}) + const., \\ \Rightarrow \;\; \ln w^{(j)} &= \mathbb{E}_{q(S_{1:T})} \ln p(S_{1:T} \mid \theta_{tr}^{(j)}) + const. \\ \theta_{tr}^{(j)} &\sim p_0(\theta_{tr}) = p(\theta_{tr} \mid \eta_{tr}). \end{aligned} \tag{2.55}$$

Thus, the importance sampling technique becomes a suitable solution for the variational updating step. In the next chapter, more specific discussion will be provided for incorporating the importance sampling into the variational Bayesian inference.

2.3.2 Particle Filter and Smoother

After the importance sampling is introduced for updating model parameters, another particle-based algorithm can be adopted for estimating $q(S_{1:T})$. According to (2.44), this updating procedure is amount to deriving a state estimation algorithm with the additional consideration of parameter distributions:

$$\begin{aligned} \ln q^*(S_{1:T}) &= \mathbb{E}_{q(\theta)} \ln p(X_{1:T}, S_{1:T} \mid \theta) + const. \\ &= \mathbb{E}_{q(\theta)} \ln p(X_{1:T}, S_{1:T} \mid \theta) - \mathbb{E}_{q(\theta)} \ln p(X_{1:T} \mid \theta) + const., \\ \Rightarrow \;\; q^*(S_{1:T}) &= c \cdot e^{\mathbb{E}_{q(\theta)} \ln p(S_{1:T} \mid X_{1:T}, \theta)}. \end{aligned} \tag{2.56}$$

Since $\mathbb{E}_{q(\theta)} \ln p(X_{1:T} \mid \theta)$ is independent of $S_{1:T}$, it is omitted in the following derivation of learning $q(S_{1:T})$. One can also treat this case as replacing $q(\theta)$ with the Dirac delta function [23]. Also, the one-dimensional latent feature $s_{1:T}$ is considered in the following illustration.

Similar to the aforementioned sequential forward-backward algorithm, the particle-based state estimation consists of two sequential paths: particle filtering (forward) and particle smoothing (backward). There exist several variants of the filtering path, as well as of the smoothing path [40]. Here, a sample algorithm is included to illustrate its applicability in extracting the dynamic latent feature. First, the particle representation of each latent state can be illustrated with a set of particle-weight pairs:

$$\begin{aligned} q_N^{(f)}(s_t) &= \sum_{j=1}^{N} \hat{w}_t^{(j)} \;\; \delta(s_t - s_t^{(j)}) \to p(s_t \mid X_{1:t}), \\ q_N^{(s)}(s_t) &= \sum_{j=1}^{N} w_t^{(j)} \;\; \delta(s_t - s_t^{(j)}) \to p(s_t \mid X_{1:T}). \end{aligned} \tag{2.57}$$

In the filtering path, both particles and their weights will be determined, but in the smoothing path, only the weights will be recalculated. Here, a common assumption has been added that the transition function should be available for sampling the particles, which has actually been indicated by the form in (2.1).

Similar to the Kalman filtering algorithm, the filtering path updates $q_N(s_t \mid X_{1:t})$ recursively. First, the predicted distribution is obtained based on sampling procedure:

$$s_t^{(j)} \sim p(s_t \mid s_{t-1}^{(j)}, \theta_{tr}), \quad \forall \; j = 1, ..., N, \tag{2.58}$$

where the initial particles can be generated from $p(s_1 \mid \theta_{tr})$. Then, these particles are weighted by the observation function for the filtered distribution:

$$\hat{w}_t^{(j)} \propto p(X_t \mid s_t^{(j)}, \theta_{ob}), \quad \forall \; j = 1, ..., N. \tag{2.59}$$

The efficiency of this filtering algorithm is usually determined by the distribution of $\hat{w}_t^{(1,\dots,N)}$ [40], such as through an effective number: $1/\sum_{j=1}^{N}(\hat{w}_t^{(j)})^2$. If the sampled particles have diverged too much, this indicator will become small, and the re-sampling algorithm should be applied to update the particles.

After the particles are generated for the entire sequence and the filtered distributions are ready, a recursive reweighing function can be applied for obtaining the smoothed distribution. According to the general backward path in (2.16) and (2.17), this smoothing path can be derived as

$$\begin{aligned}
p(s_t \mid X_{1:T}) &= \int p(s_{t+1} \mid X_{1:T}) \, \frac{p(s_{t+1} \mid s_t) \; p(s_t \mid X_{1:t})}{p(s_{t+1} \mid X_{1:t})} ds_{t+1} \\
&= p(s_t \mid X_{1:t}) \int \frac{p(s_{t+1} \mid s_t) \; p(s_{t+1} \mid X_{1:T})}{\int p(s_{t+1} \mid s_t) \; p(s_t \mid X_{1:t})} ds_{t+1} \\
&\approx \sum_{j=1}^{N} \hat{w}_t^{(j)} \; \delta(s_t - s_t^{(j)}) \int \frac{p(s_{t+1} \mid s_t) \; \sum_{k=1}^{N} w_{t+1}^{(k)} \; \delta(s_{t+1} - s_{t+1}^{(k)})}{\int p(s_{t+1} \mid s_t) \; \sum_{l=1}^{N} \hat{w}_t^{(l)} \; \delta(s_t - s_t^{(l)}) ds_t} ds_{t+1} \\
&= \sum_{j=1}^{N} \hat{w}_t^{(j)} \; \delta(s_t - s_t^{(j)}) \sum_{k=1}^{N} \frac{w_{t+1}^{(k)} \; p(s_{t+1}^{(k)} \mid s_t)}{\int p(s_{t+1}^{(k)} \mid s_t) \; \sum_{l=1}^{N} \hat{w}_t^{(l)} \; \delta(s_t - s_t^{(l)}) ds_t} \\
&= \sum_{j=1}^{N} \hat{w}_t^{(j)} \; \sum_{k=1}^{N} \frac{w_{t+1}^{(k)} \; p(s_{t+1}^{(k)} \mid s_t^{(j)})}{\sum_{l=1}^{N} \hat{w}_t^{(l)} \; p(s_{t+1}^{(k)} \mid s_t^{(l)})} \; \delta(s_t - s_t^{(j)}).
\end{aligned} \tag{2.60}$$

In the third row above, the approximation is made by using $q_N^{(f)}$ and $q_N^{(s)}$ in (2.57). Comparing the final result to the representation in (2.57), the adjusted weight for smoothing path can be obtained as

$$w_t^{(j)} = \sum_{k=1}^{N} w_{t+1}^{(k)} \; \frac{\hat{w}_t^{(j)} \; p(s_{t+1}^{(k)} \mid s_t^{(j)})}{\sum_{l=1}^{N} \hat{w}_t^{(l)} \; p(s_{t+1}^{(k)} \mid s_t^{(l)})}. \tag{2.61}$$

With the above estimation procedure for obtaining particles and their weights, a particle-based representation $q(S_{1:T})$ can be updated in the variational Bayesian inference. In the forth chapter of this thesis, the above inferencing approach will be modified to extract the constrained dynamic latent feature.

2.3.3 Stochastic Variational Inference

Other than being used to represent $q(\Theta)$ and $q(S_{1:T})$, the particle-based algorithms can also be applied to optimize the parameters in the proposal distributions. For example, if the proposal distribution can be parametrised by the variational parameter $\hat{\eta}$, optimizing this proposal distribution $q(\Theta \mid \hat{\eta})$ will be equivalent to optimizing the variational parameter $\hat{\eta}$.

Based on the objective function defined in (2.40), the optimal proposal distribution will be obtained at the maxima point of $\hat{\eta}^*$.

If the corresponding gradient $\nabla_{\hat{\eta}}\mathcal{L}_{q(S_{1:T},\Theta)}$ can be calculated with respect to $\hat{\eta}$, the following stochastic optimization approach can be applied:

$$\begin{aligned}\hat{\eta}^{(z+1)} &= \hat{\eta}^{(z)} + \rho_z \cdot \nabla_{\hat{\eta}}\mathcal{L}_{q(S_{1:T},\Theta)} \\ &= \hat{\eta}^{(z)} + \rho_z \cdot [\nabla_{\hat{\eta}}\mathbb{E}_{q(S_{1:T},\Theta)} \ln p(X_{1:T}, S_{1:T} \mid \Theta) \\ &\qquad\qquad - \nabla_{\hat{\eta}} D_{KL}\{q(\Theta \mid \hat{\eta}) \mid\mid p(\Theta \mid \eta)\}], \end{aligned} \tag{2.62}$$

where ρ_z is a step-size variable for the updating step z. For the second part in this gradient term, the K-L divergence $\nabla_{\hat{\eta}} D_{KL}\{q(\Theta \mid \hat{\eta}) \mid\mid p(\Theta \mid \eta)\}$ is usually an analytical function of $\hat{\eta}$. For the first gradient part, the following particle-based algorithm can be used to approximate this gradient calculation:

$$\begin{aligned} &\nabla_{\hat{\eta}}\ \mathbb{E}_{q(S_{1:T},\Theta)} \ln p(X_{1:T}, S_{1:T} \mid \Theta) \\ =&\nabla_{\hat{\eta}} \int \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta)\ q(\Theta \mid \hat{\eta}) d\theta \\ =&\int \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta)\ \nabla_{\hat{\eta}} q(\Theta \mid \hat{\eta}) d\theta \\ =&\int \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta)\ q(\Theta \mid \hat{\eta})\ \nabla_{\hat{\eta}} \ln q(\Theta \mid \hat{\eta}) d\theta \\ \approx&\frac{1}{N}\sum_{j=1}^{N} \mathbb{E}_{q(S_{1:T})} \ln p(X_{1:T}, S_{1:T} \mid \Theta^{(j)}) \nabla_{\hat{\eta}} \ln q(\Theta^{(j)} \mid \hat{\eta}), \end{aligned} \tag{2.63}$$

where the particle $\Theta^{(j)}$ is independently sampled from $q(\Theta \mid \hat{\eta})$. Therefore, given $\hat{\eta}^{(z)}$, a sufficient number of parameter particles $\Theta^{(1,\ldots,N)}$ can be generated for this approximation.

In practice, after $q(\Theta \mid \hat{\eta})$ being parametrised within the exponential distribution family, $\nabla_{\hat{\eta}} \ln q(\Theta^{(j)} \mid \hat{\eta})$ can also be calculated analytically for each $\hat{\eta} = \hat{\eta}^{(z)}$ [41]. Thus, the stochastic optimization approach in (2.62) can be realized. By adopting this particle-based inferencing approach, the variational Bayesian inference has become more attractive in dealing with large data sets [41] and has been widely used for improving the deep neural networks [28].

Chapter 3

Bayesian Learning for Slow Feature Analysis *

In this chapter, based on the probabilistic formulation of slow feature analysis, a novel learning framework is proposed to extract dynamic latent features. Under this full Bayesian framework, the prior knowledge of process dynamics is materialized as a Beta distribution for the transition parameters. Unlike the conventional method dealing with a static model, in this study, a latent dynamic model is learned by considering the nominal velocities of the latent features. By applying the variational Bayesian inference, the estimation uncertainties can be accounted for by using a probabilistic description for model parameters. By using a variational lower bound as the evaluation indicator for this learning process, the number of latent features can be determined automatically. Through the modelling tasks on simulated datasets and an industrial application case, the effectiveness and practicability of this feature extraction method have been demonstrated.

3.1 Introduction

In the process industry, the on-line measurement of quality variables constitutes an indispensable part of process control and optimization. Due to the unsatisfactory performance of real-time hardware instruments, inferential sensing technologies, which make predictions with commonly available process data, became favourable alternative solutions. Because of unattainable assumptions and simplifications in first-principles formulations, they usually cannot be effectively applied to prediction tasks. Meanwhile, the increasing number of instrumented sensors and a large number of historical records make the process data analytic

[*] A version of this chapter has been published in Ma, Yanjun, and Biao Huang. “Bayesian learning for dynamic feature extraction with application in soft sensing.” IEEE Transactions on Industrial Electronics 64, no. 9 (2017): 7171-7180.

a strong modelling alternative to build inferential sensors [7]. As a data-driven project, the most challenging task is to develop a model for accurate real-time measurement for the quality variables.

In order to build predictive models, the properties of historical data are worth deliberately exploring, such as discussed in Chapter 1. Specifically, an effective dimensionality reduction method and appropriate feature selection criteria are required to avoid misleading regression coefficients and avoid losing informative features for quality prediction. Based on the above considerations, the modelling approach with the dynamic latent feature, as discussed in Section 2.1, is considered in this chapter. From the modelling point of view, these applications formulate a latent space to filter data and propagate information from the input space to the output space, instead of directly building an input-output model, and expect to achieve more effectiveness and robustness of the predictions through building relation on this latent layer.

Since the inferential sensing usually serves as the supplement for a control improvement project, where the output of inferential sensor could be used as the input for a feedback controller, the original target system is often operated by less satisfactorily designed controllers. Along with the inevitable fluctuations and the inertia of processes, process data contains dynamic properties encapsulated as temporal correlations among process variables. Ignoring process dynamics will inevitably limit the ability to discover the statistical mapping, and lose important information conveyed from stochastic data sequences. Most of the current considerations of process dynamics focus on the temporal correlation in the observed sequences directly, such as including lagged data samples for a dynamic augmentation [42, 43]. However, as the representation of observed process data, the extracted latent features can also contain and in fact better represent dynamic properties, which makes them more suitable to capture the hidden driving causes of process variations. Unfortunately, for general process data analytic with the assumption of the steady state, inadequate efforts have been made towards revealing data dynamics other than the statistical analysis of process data. One of the approaches considering the dynamic connection in latent space is the subspace identification [44, 45], in which the definition of states is inherited from system identification, influenced by inputs and revealed by outputs.

On the other hand, the process variables in inferential sensing applications may not be the inputs of a physical system. Instead, they are usually measured outputs of other variables. The features that are mathematically projected from these measurements will be a better choice to effectively and efficiently capture process trends, sharing the similarity with feature extraction in pattern recognition. Another approach, named as slow feature analysis (SFA) [17], constructs the latent space by minimizing the averaged difference in the sequence of latent features, formulated as a variant of independent component analysis (ICA). This approach combines the noise removal advantage of ICA and the slowness preference in process modelling, based on which the performance of quality variable prediction and process monitoring can be improved through modelling and monitoring in the latent space [46].

As the mainstay of engineering and computer science, the probability theory is viewed as the best tool to formulate models and estimate parameters from data [23]. As presented with the probabilistic in Section 2.2, the model can be learned with an explicit prior preference, the structure can be selected according to specific evaluating indices, and the predictions can be assigned with specific beliefs in application [8]. With a quadratic objective function and relatively clean data sets, the deterministic methods, such as eigenvalue based approaches can solve the modelling problem. However, as a deterministic solution, the objective is dictatorial and difficult to evaluate in the whole context of feature extraction. Moreover, the projection matrix is likely to down-weigh more noisy measurements of the inputs. In order to provide better insights into feature extraction, the knowledge-based modelling preference should be mathematically merged into the model evaluation, where a probabilistic framework is naturally adopted. Most of deterministic data analysis approaches also have their probabilistic extensions [22, 24] to be robust to abnormal sampled data and allow more extensions from the original version. In the dynamic latent variable extraction of this chapter, the probabilistic approach and the Bayesian framework can provide compact interpretation for dynamics in the latent space, and related inference algorithms can merge the prior preference with the model of observations.

3.2 Problem Statement

In this chapter, a novel Bayesian dynamic feature extraction method is proposed to discover latent features along with auto-regressive properties, sufficiently capture process fluctuations, and predict quality variables from latent features. The feature extraction established in the Bayesian framework is capable of bringing two significant advantages: the integration of process knowledge and historical data, and the completeness of simultaneous parameter and states estimation. The first one is to be achieved by using a random variable to describe the varying velocity of each latent feature and formulating its probabilistic dependency. The second one is to be obtained through a compact Bayesian approximation algorithm, which not only approximates the posterior of model parameters but also evaluates the model structures with an explicit index.

This chapter proceeds as follows. In the next section, the feature extraction is formulated with a probabilistic framework, and detailed explanations are provided. In the third section, the parameter estimation and the model learning technologies are introduced, along with the proposed modifications and extensions. The fourth section demonstrates the strength of our modelling algorithm in a simulated data, a benchmark simulation and a real application in industry. Finally, concluding remarks and future perspectives are presented.

3.3 Probabilistic Formulation

In this section, a probabilistic framework is formulated to connect data and the latent space, as well as a layer of prior distributions for parameter estimation. Based on the modelling objective, the feature extraction is designed to represent the process dynamics by sequences of autocorrelated latent variables. With the probabilistic interpretation of SFA [25], further characteristics of the latent features and the Bayesian framework are discussed with the following state space formulation:

$$S_t = A \cdot S_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{3.1}$$

$$X_t = H \cdot S_t + v_t, \qquad v_t \sim \mathcal{N}(0, R), \tag{3.2}$$

where A is a d-by-d matrix to model the auto-correlation, H is a m-by-d matrix describing the projection from latent space to observations. Both the transition noise and the observation noise are assumed to have the multivariate Normal distribution $\mathcal{N}$, with the covariance matrices Q and R respectively.

For the aforementioned model, imposing the independence condition forces the transition matrix and the covariance matrix in (3.1) to be diagonal: $A = diag\{a_1, ..., a_d\}$ and $Q = diag\{\rho_1^{(-1)}, ..., \rho_d^{(-1)}\}$. In addition to de-correlating input space, the model with independent latent features allows the process dynamics to be represented among different frequency bands, without interfering each other [47].

3.3.1 Statistical Dependency

In order to incorporate the modelling preference and describe dynamic properties explicitly, some characteristics of the latent features need to be investigated. For the proposed feature extraction model, the preference of different velocities can be used as a guide. To exploit the varying velocity, or the "slowness", explicitly, the stationary constraint is considered for the latent feature S in (3.1).

The stationary constraint forces the latent variable sample S_t to have an invariant mean and an invariant variance for prior, which states that given a set of parameters, $p(S_t) = p(S_{t-1})$ for any time t. As inherited from conventional constrained in slow feature analysis, the zero-mean in (1.2) and the uni-variance in (1.3) are adopted. This constraint is not only inherited from traditional latent variable models but also based on physical understandings about target processes. In most of the inferential sensing projects, quality variables are operated in a continuous manner, requiring the extracted features to capture the continued variations.

To impose these constraints in the state space form, an additional equation for transition parameters are introduced. Since the latent features are considered as independent, this constraint is also discussed for one-dimensional case. For the i-th feature $s^{(i)}$, the stationary constraint is implemented as (3.3) and (3.4):

$$s_t^{(i)} = a_i \; s_{t-1}^{(i)} + w_t^{(i)}, \quad w_t^{(i)} \sim \mathcal{N}(0, \rho_i^{-1}), \tag{3.3}$$

$$where: \quad a_i^2 + \rho_i^{-1} = 1, \quad \rho_i > 0. \tag{3.4}$$

By imposing (3.4), if $s_{t-1}^{(i)}$ follows the standard Normal distribution $\mathcal{N}(0,1)$, it can be verified that $s_t^{(i)}$ will then follow the same standard Normal distribution according to the first order Markovian property. Thus, the original constraints in SFA, (1.2) and (1.3), are satisfied now with probabilistic interpretation [47]. Other than constraint (3.4), to better represent the concept of "slow" feature, the support domain of the transition coefficient is limited as

$$a_i \in (0,1), \quad \forall \; i \in \{1, ..., d\}. \tag{3.5}$$

Since ρ_i stays positive, $|a_i|$ must be smaller that 1. The negative part is eliminated in this study to avoid the extreme noisy variations.

Intuitively, this constraint states that to maintain a constant variation level, the feature with weaker auto-regressive strength will be affected more by the noise. As one can see from (3.4) that a smaller a_i corresponds to a larger variance of noise ρ_i^{-1}. Based on this, each pair of diagonal terms in A and Q shrinks to one free random variable ($\rho_i = \frac{1}{1-a_i^2}$), and the averaged varying velocity in (1.1) can be represented by and controlled through the auto-regressive parameter a_i:

$$v(s^{(i)}) = \mathbb{E}\left[\|s_t^{(i)} - s_{t-1}^{(i)}\|^2\right] = 2\;(1 - a_i). \tag{3.6}$$

For a visual illustration, two independent features under invariant distribution assumption are shown in Figure 3.1. The one with larger value of a_i demonstrates an obviously slower trend comparing to the other.

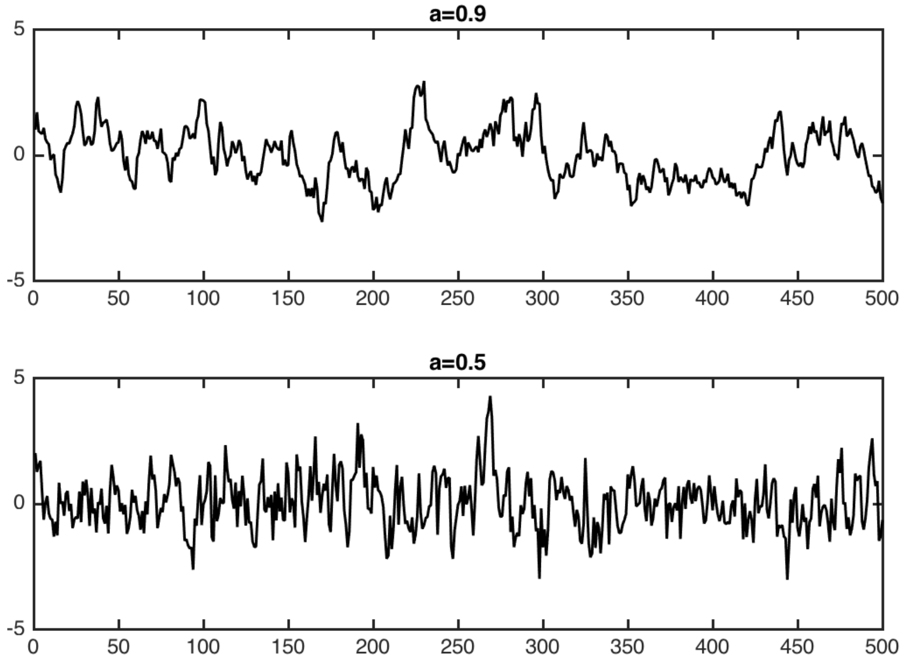


Figure 3.1: Latent Features with Different Varying Velocities

This random variable based representation of varying velocity allows the modelling strategy to optimize each a_i within a probabilistic framework for all observed data, rather than only maximizing it without considering other performances, such as SFA does [17]. In the proposed Bayesian framework, by assigning a prior distribution to each coefficient parameter a_i, the prior knowledge of auto-regressive dynamics can be integrated mathematically. For example, while there exist a prior guess for the time constant of informative trends, it can be converted as a prior distribution for this parameter:

$$\begin{aligned} & G_p = \frac{S(s)}{U(s)} = \frac{K}{\tau\ s+1} \\ \Rightarrow\ & s_t = e^{-\frac{T_S}{\tau}}\ s_{t-1} + k(1-e^{-\frac{T_S}{\tau}})\ u_{t-1} \\ \Rightarrow\ & a = e^{-\frac{T_S}{\tau}}, \end{aligned} \tag{3.7}$$

where G_p is a transfer function between U and S in the frequency domain and T_S is the system sampling interval for its discrete counterpart. It can be observed that for this first order system, its time constant is related to the transition coefficient of auto-regression function (3.3).

As discussed in Chapter 2, the joint distribution of observations $X_{1:T}$ and hidden features $S_{1:T}$ represented by (3.1) and (3.2) plays a central role in statistical inference, which is expressed as following conditional distribution given a set of parameters $\{A, H, R\}$:

$$\begin{aligned} p(X_{1:T}, S_{1:T} \mid A, H, R) &= p(X_{1:T} \mid S_{1:T}, H, R) \cdot p(S_{1:T} \mid A) \\ &= \prod_{t=1}^{T} p(X_t \mid S_t, H, R) \cdot \prod_{t=2}^{T} \ln p(S_t \mid S_{t-1}, A) \cdot p(S_1). \end{aligned} \tag{3.8}$$

Based on joint likelihood in (3.8), the Maximum-Likelihood is amenable [47]. However, in order to fully utilize the integrity of probabilistic framework and incorporate the parsimony principle, the proposed feature extraction strategy is formulated in a Bayesian framework with the prior distributions for parameters.

3.3.2 Prior Distributions

One major innovation of this chapter is introducing a prior distribution for the constrained transition parameter A. Based on that, the modelling preference of varying velocity of latent feature S can be implemented through the connection in (3.6). For each element a_i, Beta distribution is chosen as the prior distribution to govern the continuous value of a_i in the range of $(0, 1)$:

$$Beta(a_i \mid \alpha_a, \beta_a) = \frac{(a_i)^{\alpha_a - 1}(1 - a_i)^{\beta_a - 1}}{\mathcal{B}\{\alpha_a, \beta_a\}}. \tag{3.9}$$

The parameters of Beta distribution, α_a and β_a, as the hyper-parameters for entire Bayesian model can determine the preferred varying velocity. In Figure 3.2, the left plot shows the cases that larger a_i possesses larger prior probability, representing a preference of slower features; the right one shows the other way around with a specified varying velocity ($\mathbb{E}[a_i] = 0.3$).

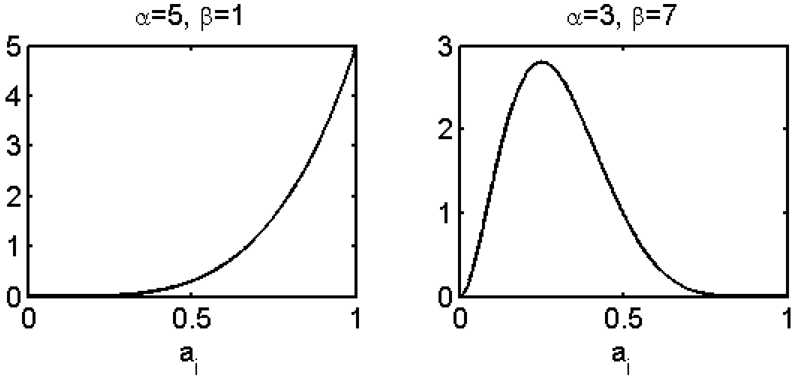


Figure 3.2: Different Choices of Hyper-Parameters in the Beta Distribution

Another important component is the parsimony principle in formulating multiple latent features. In most tasks of modelling [48] and identification [49], since the divergence caused by over-fitting is not desired, the model with less complexity or a lower order is preferred. In the proposed structure of observation function (3.2), this preference is translated to using zero as an expectation for the prior mean of emission matrix H. For one row vector h_i in H, the prior distribution can be selected as the Normal distribution:

$$h_i \sim \mathcal{N}(\vec{0}, \Lambda_0^{-1}), \tag{3.10}$$

where Λ_0 is a d-by-d precision matrix for the strength of prior information. With such prior distributions, the posterior estimation of H tends to be zero while the likelihood from observation, $p(X_{1:T} \mid S_{1:T}, H, R)$, is not significant for non-zero H. This mechanism allows the Automatic Relevance Determination [50], which can reduce the weight of irrelevant latent features during the inference procedure. Besides, particular prior distributions, such as the exponential distribution, can be chosen for stronger dimensionality shrinkage [51].

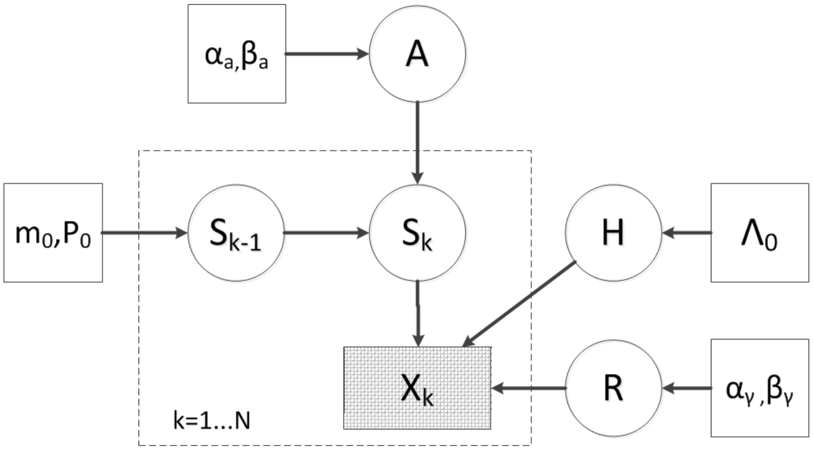


Figure 3.3: Proposed Bayesian design for slow feature extraction

At this point, the hierarchical probabilistic graphical model corresponding to model (3.1)-(3.2) can be summarized in Figure 3.3. The probability expression of its directed connections are summarized as

$$p(S_t \mid S_{t-1}, A) = \mathcal{N}(S_t;\ A \cdot S_{t-1}, \mathbb{I}_d - A^2), \quad \forall\ t = 2, ..., T, \tag{3.11}$$

$$p(X_t \mid S_t, H, R) = \mathcal{N}(X_t;\ H \cdot S_t, R), \qquad \forall\ t = 1, ..., T, \tag{3.12}$$

where $\mathbb{I}_d$ is the d-dimensional identity matrix and the covariance of transition noise Q is replaced with $\mathbb{I}_d - A^2$ for simplicity. These two likelihood terms are detailed probabilistic connections for latent features and observations in (3.8). The prior distribution or the hierarchical modelling is expressed as

$$p(A \mid \alpha_a, \beta_a) = p(diag\{a_1, ..., a_d\} \mid \alpha_a, \beta_a) = \prod_{i=1}^{d} Beta(a_i;\ \alpha_a, \beta_a), \tag{3.13}$$

$$p(H \mid \Lambda_0) = p([h_1, ..., h_m]' \mid \Lambda_0) = \prod_{i=1}^{m} \mathcal{N}(h_i;\ \vec{0}, \Lambda_0^{-1}), \tag{3.14}$$

$$p(R \mid \alpha_\gamma, \beta_\gamma) = p(\gamma^{-1} \cdot \mathbb{I}_m \mid \alpha_\gamma, \beta_\gamma) = \mathcal{G}(\gamma;\ \alpha_\gamma, \beta_\gamma), \tag{3.15}$$

$$p(S_1 \mid m_0, P_0) = \mathcal{N}(S_1;\ \vec{0}, \mathbb{I}_d). \tag{3.16}$$

As illustrated before, distribution in (3.13) provides the prior distribution for varying velocities, and the prior of row vectors of emission matrix H is expressed in (3.14). The covariance of emission noise R is set as the scaled identity matrix, where $\mathbb{I}_m$ is the m-dimensional identity matrix. This prior distribution is adopted from the probabilistic mapping strategy [22] to model each observation dimension with an equal weight; the usage of Gamma distribution $\mathcal{G}(\cdot)$ as conjugate prior in (3.15) facilitates the parameter estimation procedure. Regarding the prior distribution for initial state in (3.16), the standard Normal distributions are assigned for each dimension, $m_0 = \vec{0},\ P_0 = \mathbb{I}_d$, which is accordance with the original slow feature analysis.

3.4 Variational Inference Methods

Regarding the parameter inference work, Bayesian learning methods gain their popularity because of the explicit synthesis of data and modelling preference. Based on a set of given probability dependencies, specific estimation algorithms can be formulated to incorporate different uncertainties. As the feature extraction model has been formulated as the directed probability graphical model shown in Figure 3.3, which is an extended model from Dynamic Bayesian Network [36], the posterior distribution of parameters $\{A, H, R\}$ and latent features $S_{1:T}$ can be determined simultaneously. Because of the high dimension of parameters, the analytical derivation of posterior distribution is prohibitively expensive. As illustrated in Chapter 2, the approximate Bayesian inference is applied to obtain the applicable posterior distributions.

The joint log-likelihood of observations $X_{1:T}$, latent features $S_{1:T}$, and model parameters $\{A, H, R\}$ is presented as

$$\begin{aligned}
& \ln p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma) \\
= \ & \ln p(S_1 \mid m_0, P_0) + \sum_{t=2}^{T} \ln p(S_t \mid S_{t-1}, A) + \sum_{t=1}^{T} \ln p(X_t \mid S_t, H, R) \\
& \quad + \sum_{i=1}^{d} \ln p(a_i \mid \alpha_a, \beta_a) + \sum_{i=1}^{m} \ln p(h_i \mid \vec{0}, \Lambda_0) + \ln p(\gamma \mid \alpha_\gamma, \beta_\gamma).
\end{aligned} \tag{3.17}$$

Based on introduction of variational algorithms in Section 2.2, proposal distribution of the latent features and model parameters can be used to approximate their true posterior:

$$q(S_{1:T}, A, H, R) = q(S_{1:T})\ q(A)\ q(H)\ q(R). \tag{3.18}$$

The updating for all components will be derived from above joint likelihood, such as for the proposal distribution of latent features $q^*(S_{1:T})$:

$$\ln q^*(S_{1:T}) = \mathbb{E}_{q(A)q(H)q(R)} \left[\ln p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)\right] + const., \tag{3.19}$$

where the $q(A)q(H)q(R)$ is obtained from the previous updating step.

In the linear case, most of distribution updating steps can be facilitated by assigning the prior as conjugate of the likelihood [52]. For example, we choose the prior of elements in R to be inverse Gamma distribution. According to the Bayes rule, these prior distributions will uniquely yield posterior distributions with the same form as their corresponding priori, leaving only their parameters to be updated. Typically in this chapter, the updating equations for H and R are obtained as

$$q^*(H) = \prod_{i=1}^{m} q^*(h_i) = \prod_{i=1}^{m} \mathcal{N}(h_i;\ \hat{m}_i, \hat{\Lambda}_i^{-1}), \tag{3.20}$$

$$q^*(R) = q^*(\gamma) = \mathcal{G}(\gamma;\ \hat{\alpha}_\gamma, \hat{\beta}_\gamma). \tag{3.21}$$

The stepwise optimal hyper-parameters are derived as

$$\hat{\Lambda}_i = \Lambda_0 + \langle\gamma\rangle \sum_{t=1}^{T} \langle S_t S_t'\rangle, \tag{3.22}$$

$$\hat{m}_i = \Lambda_i^{-1}\ \langle\gamma\rangle \sum_{t=1}^{T} \langle S_t\rangle \cdot X_t^{(i)}, \tag{3.23}$$

$$\hat{\alpha}_\gamma = \alpha_\gamma + \frac{T \cdot m}{2}, \tag{3.24}$$

$$\hat{\beta}_\gamma = \beta_\gamma + \frac{1}{2}\sum_{i=1}^{m}\left[\sum_{t=1}^{T} X_k^{(i)} X_t^{(i)} - 2\sum_{t=1}^{T} X_t^{(i)} \langle S_t\rangle' \langle h_i\rangle + tr\{\sum_{t=1}^{T}\langle S_t S_t'\rangle\langle h_i h_i'\rangle\}\right]. \tag{3.25}$$

In these equations, $\langle\cdot\rangle$ is the shorthand for statistical expectation of a random variable under its own distribution.

While the conjugate exponential family significantly reduces the load of calculation, solutions to updating equations for $q^*(S_{1:T})$ and $q^*(A)$ are not straightforward: the first one requires the uncertainty compensation in the sequenced data, and the difficulty for the second one is from handling non-conjugate prior (Beta distribution). In what follows, the detailed development for these two updating steps will be illustrated.

3.4.1 Estimation of Dynamic Latent Features

In order to obtain the updating equation for latent features $S_{1:T}$, the following transformation is made according to the Bayes rule:

$$\begin{aligned}
&\ln q^*(S_{1:T}) \propto \mathbb{E}_{q(A)q(H)q(R)}\left[\ln p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)\right] \\
= &\ \mathbb{E}_{q(A)q(H)q(R)}\Bigg[\ln \frac{p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)}{p(X_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)} \\
&\qquad + \ln p(X_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)\Bigg] \\
= &\ \mathbb{E}_{q(A)q(H)q(R)}\left[\ln p(S_{1:T} \mid X_{1:T}, A, H, R)\right]. \qquad (3.26)
\end{aligned}$$

While given a set of distributions of parameters, the expectation of $\ln p(X_{1:T}, A, H, R \mid ...)$ is independent from the distribution of latent features. The final term of this transformation $p(S_{1:T} \mid X_{1:T}, A, H, R)$ brings out a state estimation problem based on the model in (3.1) and (3.2). In the case with deterministic value of model parameters, the optimal solution can be obtained through Kalman smoothing algorithm [53]. However, under this Bayesian framework, the parameters are given as the probability distributions instead of the point estimations, where the uncertainty of parameters implies following inequality:

$$\begin{aligned}
&\mathbb{E}_{q(A)q(H)q(R)}\left[\ln p(S_{1:T} \mid X_{1:N}, A, H, R)\right] \\
&\qquad \neq \ln p(S_{1:T} \mid X_{1:T}, \langle A\rangle, \langle H\rangle, \langle R\rangle). \qquad (3.27)
\end{aligned}$$

In the linear case with Gaussian distributed latent states, the difference between two sides of inequality (3.27) can be determined through the *fluctuation terms* [54]:

$$F_t = \langle A' \cdot (\mathbb{I}_d - A^2)^{-1} \cdot A\rangle - \frac{\langle A \cdot (\mathbb{I}_d - A^2)^{-1}\rangle \langle (\mathbb{I}_d - A^2)^{-1} \cdot A\rangle}{\langle (\mathbb{I}_d - A^2)^{-1}\rangle}, \qquad (3.28)$$

$$F_o = \langle H' \cdot R^{-1} \cdot H\rangle - \langle H'\rangle \langle R^{-1}\rangle \langle H\rangle, \qquad (3.29)$$

where F_t stands for fluctuation from the transition function, and F_o describes fluctuation in the observation function. Because of the non-zero property of these fluctuation terms, using the right side of (3.27) directly from the Kalman smoothing algorithm will introduce error to the latent feature. In the study of [54], factorized components of these *fluctuation terms* are applied to compensate this differences, making the original Kalman smoothing algorithm feasible for the left side of (3.27).

With the *fluctuation terms* identified, it can be factorized as U_A and U_B through the Cholesky decomposition:

$$U_A \cdot U_A' = F_t,$$

$$U_B \cdot U_B' = F_o.$$

The inequality can thus be compensated by augmenting each observed data vector X_t, the emission matrix H, and the covariance matrix R with these components:

$$\widetilde{X}_t = \left[X_t', \ \vec{0}_{1\times d}, \ \vec{0}_{1\times d}\right], \qquad \forall\ t = 1, ..., T, \tag{3.30}$$

$$\widetilde{H}_t = \begin{cases} [\langle H\rangle, \ U_A, \ U_B], & \forall\ t = 1, ..., T-1, \\ [\langle H\rangle, \ \vec{0}, \ \ U_B], & \quad \mathit{if}: t = T, \end{cases} \tag{3.31}$$

$$\widetilde{R} = diag\{\langle R\rangle, \ \mathbb{I}_d, \ \mathbb{I}_d\}. \tag{3.32}$$

Regarding the transition parameters, the adjustments are performed as

$$\widetilde{A} = \frac{\langle A \cdot (\mathbb{I}_d - A^2)^{-1}\rangle}{\langle(\mathbb{I}_d - A^2)^{-1}\rangle}, \tag{3.33}$$

$$\widetilde{Q} = \langle(\mathbb{I}_d - A^2)^{-1}\rangle^{-1}. \tag{3.34}$$

It should be noted this transformation is a little bit twisted comparing to the direct transformation, such as $\langle A\rangle$ or $\langle \mathbb{I}_d - A^2\rangle$. However, these replacements are based on the expectation terms on the logarithm expansion for the state likelihood in (3.11). Based on above replacements, a time-varying state space model with augmented observations and augmented model parameters can be derived as the following model:

$$\begin{aligned} S_t &\sim \mathcal{N}(\widetilde{A} \cdot S_{t-1}, \widetilde{Q}), \\ X_t &\sim \mathcal{N}(\widetilde{H}_t \cdot S_t, \widetilde{R}), \end{aligned} \tag{3.35}$$

where these parameters are all deterministic values. Based on this augmented model, a solution equivalent to (3.26) is given below:

$$\begin{aligned} &\mathbb{E}_{q(A)q(H)q(R)}\left[\ln p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma)\right] \\ = \ &\ln p\left(S_{1:T} \mid \widetilde{X}_{1:T}, \widetilde{A}, \widetilde{Q}, \widetilde{H}_{1:T}, \widetilde{R}\right) + const.. \end{aligned} \tag{3.36}$$

Thus, solving the posterior on the right side of (3.36) will provide updated distributions for latent features $S_{1:T}$ in VB framework, where the optimal solution for this posterior in the model of (3.35) is the Kalman filtering and smoothing algorithm [33].

The updated distribution of $q^*(S_{1:T})$ will then be used to calculated the expectation of latent states. In the learning framework of this chapter, the required statistics include $\langle S_t\rangle$, $\langle S_t S_t'\rangle$, and $\langle S_{t-1} S_t'\rangle$. Based on the illustration in Section 2.1, the posterior distribution of latent states can be obtained in the form of $p(S_t \mid \widetilde{X}_{1:T}, \widetilde{A}, \widetilde{Q}, \widetilde{H}_{1:T}, \widetilde{R})$ and $p(S_{t-1}, S_t \mid \widetilde{X}_{1:T}, \widetilde{A}, \widetilde{Q}, \widetilde{H}_{1:T}, \widetilde{R})$. As in the linear model with Normal distributions, these statistics are available from result multivariate Normal distributions.

3.4.2 Estimation of Transition Parameters

The last set of parameters to be estimated is diagonal elements in the transition matrix A. The prior distribution in (3.13) is determined by considering both the support domain in (3.5) and the modelling preference on dynamic properties. However, the selection of Beta distribution makes it a non-conjugate prior for the likelihood in (3.11). Consequently, the posterior distribution does not belong to any known analytical distributions. In Section 2.3, particle-based algorithms have been reviewed for such challenges. In this chapter, the importance sampling methods are adopted for the transition parameters.

Since each latent dimension is assumed as independent from each other, it will be updated individually then. The updating equation for the i-th term can be detailed with an explicit form:

$$\ln q(a_i) = \ln p(\langle s_{1:T}^{(i)} \rangle \mid a_i) + \ln p(a_i \mid \alpha_a, \beta_a) + const., \tag{3.37}$$

$$\begin{aligned} where: \ \ln p(\langle s_{1:T}^{(i)} \rangle \mid a_i) = &\frac{T-1}{2}\ \ln \frac{1}{1-a_i^2} - \frac{1}{2}\ \sum_{t=2}^{T} \langle s_t^{(i)} s_t^{(i)} \rangle \frac{1}{1-a_i^2} \\ &+ \sum_{t=2}^{T} \langle s_{t-1}^{(i)} s_t^{(i)} \rangle \frac{a_i}{1-a_i^2} - \frac{1}{2}\ \sum_{t=1}^{T-1} \langle s_t^{(i)} s_t^{(i)} \rangle \frac{a_i^2}{1-a_i^2}. \end{aligned} \tag{3.38}$$

The elemental statistics $\langle s_{t-1}^{(i)} s_t^{(i)} \rangle$ and $\langle s_{t-1}^{(i)} s_t^{(i)} \rangle$ are obtained as the diagonal terms of whole statistic of $\langle S_t S_t' \rangle$ and $\langle S_{t-1} S_t' \rangle$. It is true that the rest off-diagonal terms are ignored in learning the transition coefficients, which becomes an inevitable assumption in this proposed learning structure.

As a sampling technique, importance sampling uses samples to stochastically approximate the exact posterior distribution. In its procedure, a support distribution ($p_0(x)$), which has a wider non-zero domain than the target distribution ($q(x)$), is proposed to generate Monte Carlo samples. When the importance vector of the target distribution over the support distribution ($w(x) = \frac{q(x)}{p_0(x)}$) is achievable for each sample point, any expectation value of the target distribution can be asymptotically obtained with these independently drawn samples:

$$\sum_{j=1}^{N} f\left(x^{(j)}\right) w\left(x^{(j)}\right) \overset{N\to\infty}{\longrightarrow} \int f\left(x\right) w\left(x\right) p_0\left(x\right) dx = \int f\left(x\right) q\left(x\right) dx, \tag{3.39}$$

where N denotes the total number of samples. Regarding the updating step for $q(a_i)$, the support distribution is chosen as the prior distribution $p_0 = p(a_i \mid \alpha_a, \beta_a)$, and the importance weight can be evaluated from the likelihood term $p(\langle s_{1:T}^{(i)} \rangle \mid a_i)$ in (3.38). The required statistics should also be estimated from the sample based distribution $q(a_i)$. Here, three required statistics are derived below to utilize the $q(a_i)$ in VB formulation:

$$\langle(\mathbb{I}_d - A^2)^{-1}\rangle = diag\{\langle\rho_1\rangle, ..., \langle\rho_d\rangle\}, \qquad \langle\rho_i\rangle = \sum_{j=1}^{N} \frac{1}{1-[a_i^{(j)}]^2} w\left(a_i^{(j)}\right), \tag{3.40}$$

$$\langle A(\mathbb{I}_d - A^2)^{-1}\rangle = diag\{\langle\rho_1\ a_1\rangle, ..., \langle\rho_d\ a_d\rangle\}, \qquad \langle\rho_i\ a_i\rangle = \sum_{j=1}^{N} \frac{a_i^{(j)}}{1-[a_i^{(j)}]^2} w\left(a_i^{(j)}\right), \tag{3.41}$$

$$\langle A'(\mathbb{I}_d - A^2)^{-1}A\rangle = diag\{\langle\rho_1\ a_1^2\rangle, ..., \langle\rho_d\ a_d^2\rangle\}, \quad \langle\rho_i\ a_i^2\rangle = \sum_{j=1}^{N} \frac{[a_i^{(j)}]^2}{1-[a_i^{(j)}]^2} w\left(a_i^{(j)}\right), \tag{3.42}$$

where $a_i^{(j)}$ is the j-th sample drawn from Beta distribution for a_i, and N the total number of samples.

3.4.3 Variational Lower Bound

The learning objective of variational Bayesian learning algorithm is the variational lower bound of evidence probability, for which the general formulation has been given in Section 2.2. Here the specified formulation for the proposed feature extraction model is detailed as

$$\begin{aligned}\mathcal{L}_{q(S1:T)q(A)q(H)q(R)} = {} & \mathbb{E}_{q(S1:T)q(A)q(H)q(R)} \ln p(X_{1:T}, S_{1:T}, A, H, R \mid \alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \beta_\gamma) \\ & + \mathcal{H}\{q(S1:T)\} + \mathcal{H}\{q(A)\} + \mathcal{H}\{q(H)\} + \mathcal{H}\{q(R)\}.\end{aligned} \tag{3.43}$$

The first term is about the expected complete likelihood. Based on the formation in (3.17), it is achievable with the aforementioned proposal distributions. For entropy of distribution $q(S_{1:T})$, a representation of chained Normal distributions are utilized as

$$q(S_{1:T}) = \frac{q(S_1, S_2)\ q(S_2, S_3) \cdots q(S_{T-1}, S_T)}{q(S_2) \cdots q(S_{T-1})}. \tag{3.44}$$

With above factorization, the entropy $\mathcal{H}\{q(S1:T)\}$ can be obtained as a summation of individual entropies. For the parameters with conjugate prior (H and R), the required moments can be obtained directly from estimated probability density function. For transition parameter a_i, the entropy is approximated through a histogram based estimation [55].

Typically, in our feature extraction model, the dimension of latent space d is selected according to the corresponding lower bound value. Although with different values of d, the lower bound is maximized with different realizations of hyper-parameters, its maximized value still represents the best data explanation we could obtain given d [29]. In the next section, this dimension determination is examined on a constructed data set, and further utilized in a regression task.

3.5 Simulations

In this section, properties of the dynamic latent feature extraction model and its application in prediction, namely the Bayesian dynamic feature regression (BDFR), are investigated. The principal component analysis with its regression application (PCR) and slow feature analysis with its regression application (SFR) are also performed as competing algorithms. First, three feature extraction methods are compared in constructed data for dimensionality reduction. After that, the predictive abilities are tested on the simulated Tennessee Eastman Process (TEP) and a real inferential sensor modelling problem in the oil sands industry. Throughout this section, the hyper-parameters are chosen as $\{\alpha_a, \beta_a, \Lambda_0, \alpha_\gamma, \alpha_\gamma\} = \{5, 1, diag\{1, ..., d\}, 1, 1\}$.

3.5.1 Numerical Examples

In the constructed data, five sequences of the latent feature S^* are generated with different varying velocities: $A^* = diag\{0.99, 0.90, 0.50, 0.43, 0.24\}$, and a randomly generated emission matrix $H^*_{10\times5}$ projects S^* to the 10-dimensional observation data X^*, along with a certain degree of emission noise: $X^*_k = H^*_{10\times5} \cdot S^*_k + v^*_k$. Besides that, a regression vector $b^*_{1\times5} = [0.4, 3, 0.2, 1, 6]$ provides a sequence of quality variable y^* from the latent features: $y^*_k = b^*_{1\times5} \cdot S^*_k + t^*_k$. This contrived data is constructed from the linear dynamic model in (3.1) and (3.2), where a good result of BDFR will prove that the Bayesian dynamic feature extraction is able to effectively model this generated data. Besides this verification for Bayesian estimation, further discussions on the applicability of the proposed model in Figure 3.3 will be performed later on.

As the fundamental property of feature extraction methods, the evaluation of each latent sequence helps to determine the optimal dimension of the latent space. In Principal Component Analysis, the explained variance of each feature (σ^2) forms an importance index. In Slow Feature Analysis, the latent features are formulated according to the velocity (λ), which is the importance index in measuring the slowness. In our method, values of lower bound for each latent dimension d are calculated for the dimension reduction. Figure 3.4 compares these three indices that determine the latent dimension in different methods.

The significant difference between these three indices is that the first two generated by PCA and SFA are monotonously increasing or decreasing, while the third one has a maximal point. The latent dimension of PCA and SFA cannot be determined by their extreme points from this unsupervised procedure. However, as explained in the probabilistic framework, the

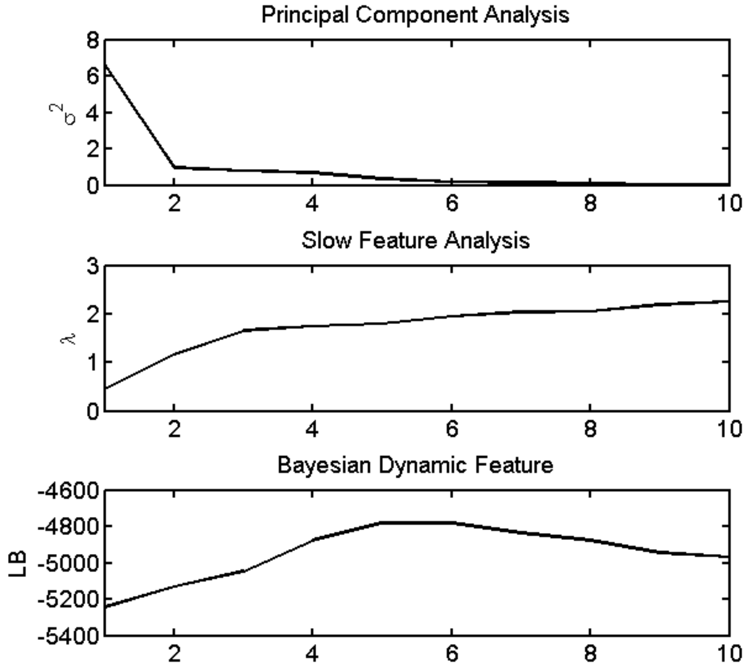


Figure 3.4: Unsupervised indices to determine the latent dimension

maximal point in our evaluating index indicates an optimal value of the latent dimension. In the third plot of Figure 3.4, the dimension of 5 (or 6) is more suitable since a relatively high value of lower bound is reached, which matches better to the theoretical value.

Latent dimension can also be determined by their predictive abilities. With different latent dimensions, the prediction result is evaluated through the Pearson correlation coefficient on a validation data set. The performance is shown in Figure 3.5, where BDFR can give the highest correlation coefficient for the same dimension selected according to Figure 3.4. While the other two methods require cross-validation, the Bayesian dynamic feature extraction is capable of utilizing the probabilistic lower bound to determine the optimal latent dimension, and its optimality is reflected by prediction performance.

3.5.2 Process Data Examples

In this section, the proposed Bayesian modelling strategy is tested on benchmark data, which is from a simulation of the Tennessee Eastman Process [56]. This process is about a plant of synthesis reactions, and the simulated data contains process variables from sensors, controllers and analyzers as described in [2]. In this case study, the objective is to build a predictive model from 33 commonly measured process variables to predict three quality

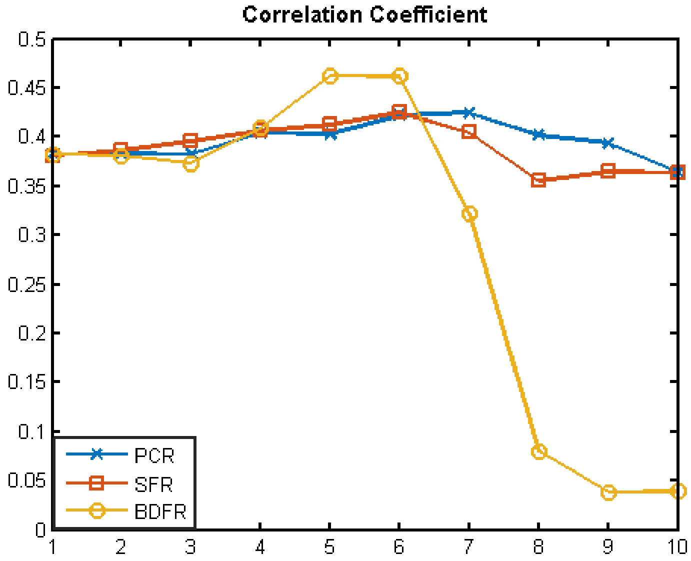


Figure 3.5: Supervised indices to determine the latent dimension

variables in the normal operating region. To have a fair competition, the inputs and outputs for both three competitors are normalized with the same mean vector and the same variance matrix.

To build a latent variable model, the dimension of the latent space ($\#LVs$) is selected by 5-fold cross-validation for PCR and SFR and by the lower bound value for BDFR respectively. The model is trained on 500 input-output samples and tested on 960 data samples. The model prediction is evaluated by the correlation coefficient (r-value), as listed in Table 3.1.

Table 3.1: Comparison in the Tennessee Eastman Process

Corr-Coef	PCR		SFR		BDFR	
	#LVs	r-value	#LVs	r-value	#LVs	r-value
Product A	24	0.6319	8	0.6354	23	0.6336
Product B	22	0.2389	9	0.2823	23	0.2389
Product C	2	0.6493	6	0.6705	23	0.6757

From this table, all three methods show similar predictive abilities for product A, while SFR is better for product B, and SFR and BDFR are better for the product C. Based on the comparison result of benchmark data, it is verified that this Bayesian modelling approach can extract meaningful features in this benchmark process simulation.

However, in the general comparison between deterministic modelling algorithm (PCA and SFA) and probabilistic approach, while the probabilistic approach always brings the ability to handle abnormal data [22], the estimated parameters from iterative optimization may be more sensitive to initial points. Therefore, the Monte Carlo experiments with different initial points are usually conducted to reduce this sensitivity for the probabilistic approach, but it still suffers from the inefficiency caused by the high dimensional parameters. In our proposed modelling approach, the complete Bayesian framework can reduce this uncertainty by assigning a layer of prior distributions. In order to investigate the stability of parameter estimation, the value of lower bound in each optimization iteration is selected as an indicating index for the training procedure. Three runs from different initial points are performed and plotted in Figure 3.6.

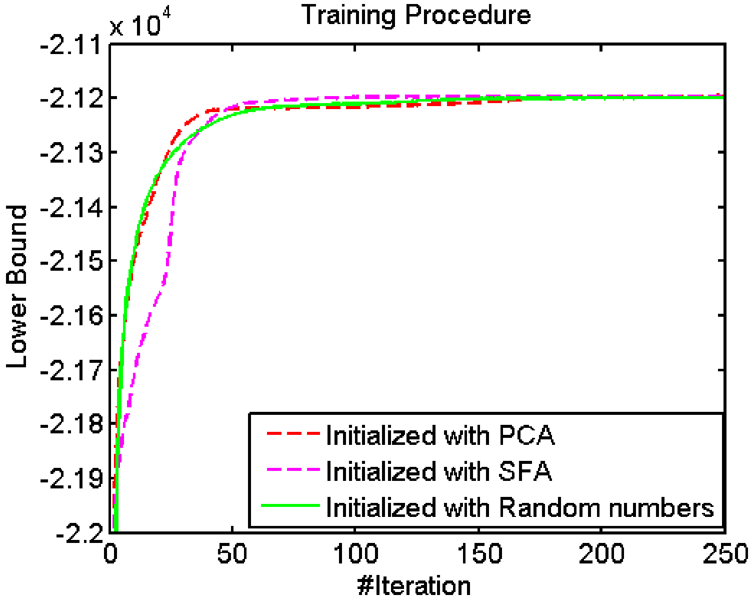


Figure 3.6: Consistency of the proposed variation methods

Along with the randomized initial point, the estimation results from PCA and SFA are also selected as the two other initial points for comparison. In Figure 3.6, three paths from different initial points reach a similar level of the final value, representing similar optimized results. As a consequence, given the prior distributions of parameters in Figure 3.3, the predicting result is robust to the changes in the initial point.

3.5.3 Industrial Case Study

The strength of the Bayesian dynamic feature extraction method is further tested on an industrial modelling problem. The process to be studied is the Steam Assisted Gravity Drainage (SAGD) in the oil sands industry. Two horizontal wells are drilled into underground bitumen reservoir, where the steam is injected through the top injector well to heat the heavy crude oil and a mixture of oil, water and gas is pumped out through wellbores on the bottom producer well. Figure 1.1 shows a typical topology for one well-pair. Two on-line measurements of the produced emulsion (water and oil), emulsion flow rate (EF) and the proportion of water, water content (WC), are required for subsequent control application. There are 18 process variables available for predicting the two quality variables, consisting of pump frequency, sampled underground pressure and emulsion temperatures. For off-line validation, the reference is provided by a limited number of analyzers with nuclear detection technology, and the data is averaged within an acceptable time range to reduce the delay effects.

Table 3.2: Comparison in the SAGD Data 1

	PCR			SFR			BDFR		
	#LVs	r-value	MAE	#LVs	r-value	MAE	#LVs	r-value	MAE
WC	4	0.61	0.70	10	0.87	0.36	9	0.85	0.31
EF	6	0.91	0.51	13	0.95	0.32	9	0.98	0.30

Table 3.3: Comparison in the SAGD Data 2

	PCR			SFR			BDFR		
	#LVs	r-value	MAE	#LVs	r-value	MAE	#LVs	r-value	MAE
WC	3	0.76	0.50	4	0.78	0.55	10	0.82	0.50
EF	14	0.91	0.25	11	0.93	0.26	10	0.97	0.20

Following the same approach as in the previous example, $\#LVs$ is selected by 5-fold cross-validation as well as the value of lower bound for different competing algorithms respectively. In addition to the correlation coefficient, the mean absolute error (MAE) forms as another evaluating index, which is usually required by practitioners. In this study, two sets of SAGD data are selected from two well-pairs in the same operation site to test the prediction performance. The comparison results are illustrated in the following tables.

From these two tables, considering both correlation coefficient and mean absolute error, the BDFR gives better performance, indicating the Bayesian dynamic features extraction can generate more informative features from data. Also, BDFR provides similar numbers of latent features in different data sets, which is promising to develop a general model structure for different well-pairs rather than determining the structure for each well-pair. Besides the quantified evaluation, the visual examination is a meaningful way to validate the performance. In Figure 3.7 and Figure 3.8, the predicting results are compared with the reference on the testing data sets, where BDFR can track the trend of the reference better than other competing algorithms. Especially for the part of first 300 samples in Figure 3.7

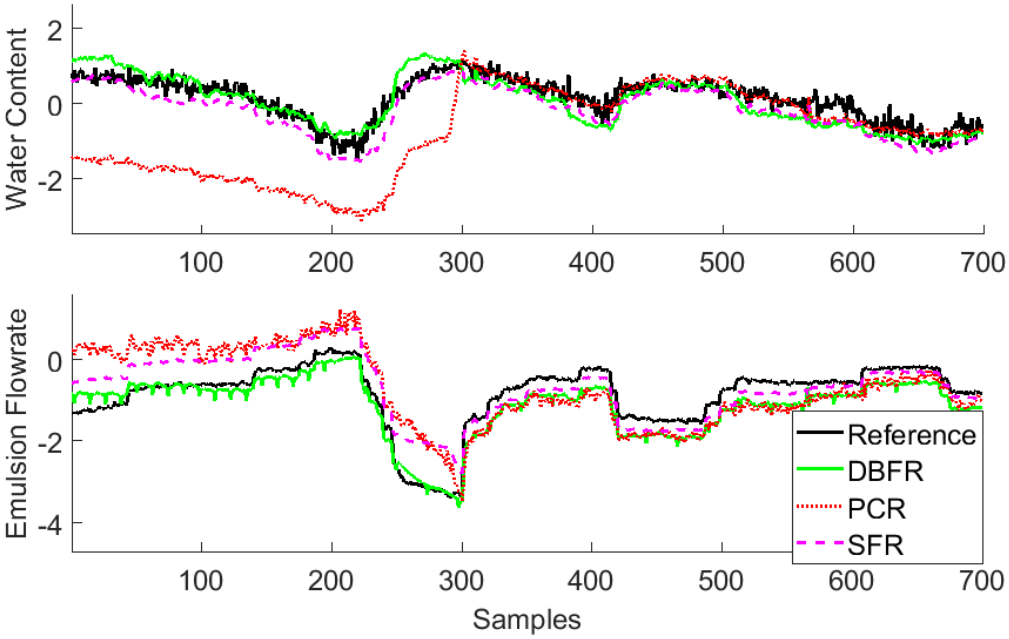


Figure 3.7: Validation Results for Modelling SAGD Data 1

and the part around 350 samples in Figure 3.8, both flow rate and water content can be captured by BDFR model, while significant deviations are observed in PCR and SFR.

3.6 Conclusion

In this chapter, a new data-driven modelling approach for dynamic feature extraction is developed. The observed process variables are assumed as a projection from the dynamic latent features. Constraints and prior distributions of model parameters are implemented to formulate a Bayesian probabilistic graphical model for feature extraction. Then the estimation algorithm for the proposed graphical model is elaborated from the variational Bayesian inference and importance sampling, which has utilized both historical data and prior modelling preferences. One of the advantages in the Bayesian extraction framework is to determine the dimension of latent space automatically. In this study, approximate Bayesian inference provided a lower bound value for the marginal likelihood of observations, which is validated to be useful in model selection. Later, the prediction strength of the extracted latent features is demonstrated with the simulated data, the benchmark simulation, and two cases of industrial data.

These verification results state that, by formulating informative latent features, the proposed Bayesian dynamic feature extraction and related estimation algorithm are applicable

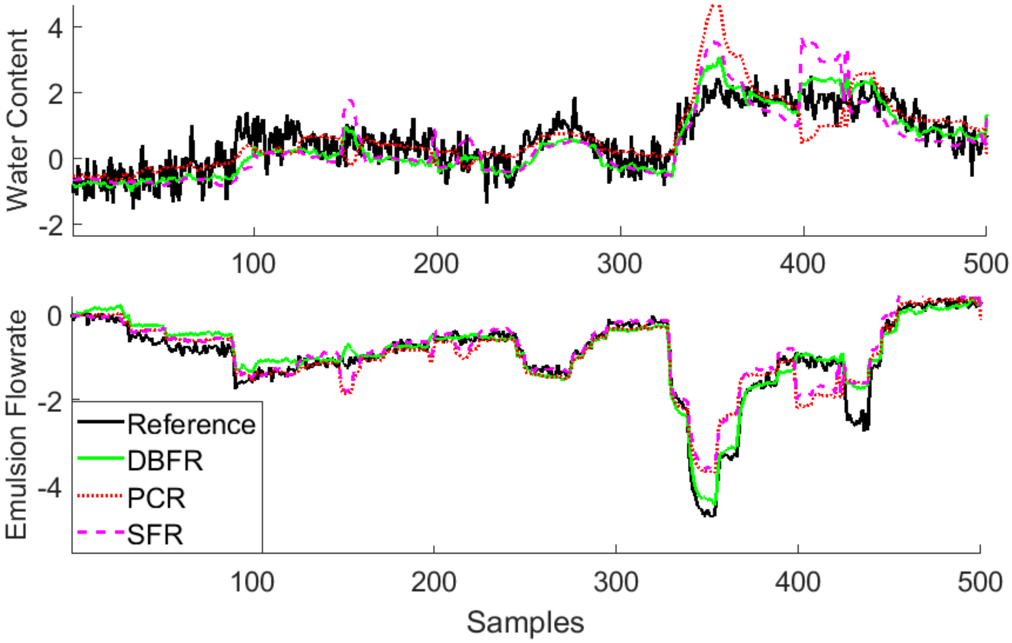


Figure 3.8: Validation Results for Modelling SAGD Data 1

for prediction applications such as inferential sensing in a SAGD related process. In the future, the proposed Bayesian dynamic feature extraction can be extended with a supervised learning strategy, which could extract features with specified dynamic properties. Besides, the estimated feature velocities can be incorporated in the online prediction, which could make the model more robust to the abnormal observations.

Chapter 4

Bayesian Generalization of Markov Transition Model [*]

When the dynamic latent feature is modelled with discrete variables, the transition function is commonly formulated as the hidden Markov model (HMM). As a structural extension, a dynamic model for the parameters of original discrete states, which represent the probability of discrete events, is considered this chapter. In order to formulate the transition function for this constrained dynamic latent features, a new probabilistic model is developed by using either the Dirichlet distribution or the Beta distribution for the transition noise. Properties of this new transition model have been discussed and compared with the conventional state transition function. By incorporating a non-linear observation function, a feature extraction model is proposed for extracting constrained dynamic features from unconstrained observations. For the detailed inference procedure, two novel learning algorithms are developed. To demonstrate the effectiveness of proposed algorithms and the applicability of the proposed model, the validation part of this chapter has used numerical examples, a benchmark simulation, and industrial data sets.

4.1 Introduction

Extracting informative features is a critical step in data analysis [10, 11]. Originally, feature extraction mainly serves for image processing [13] to find specific objects. For example, in geographical image processing, analyzers are designed to detect linear objects such as roads and airport runways, or non-linear objects such as rivers and trails. While applied for general latent variable models, its scope has been extended, and features are not necessarily related to real objects. In the most celebrated latent variable extraction, principal component

[*] A version of this chapter has been accepted as Ma, Yanjun, Shunyi Zhao, and Biao Huang. "Feature Extraction of Constrained Dynamic Latent Variables." by IEEE Transactions on Industrial Informatics

analysis (PCA) [14], the features (or latent scores) are extracted to represent the variability of multivariate observations. In the regression-oriented latent variable models such as partial least squares (PLS) [15], the features are extracted to capture common variations in both inputs and outputs. These methods formulate the latent features with specified preferences, for example, pursuing larger variance in PCA and larger mutual correlation in PLS. Thus, enrichment of modelling preferences and detailed descriptions of latent space will extend application scope and amplify efficiency of the feature extraction model.

Feature extraction techniques have been widely used for modelling key quality variables in process industries. Since accurate measurements of quality variables are usually obtained in a time-consuming and slow-rate sampling manner, an on-line estimation solution is desirable. It has the regression model to use related and reliable process variables for quality variable estimation in real-time. As illustrated in Chapter 1, the emphasis on temporal correlation between consecutive latent samples does contribute to process data analytics. On the other hand, constraints commonly exist in practice [57, 58]. One motivation comes from the commonly used position limits in industry, such as the saturation constraint in control valves and boundaries for pressures in oil extractions well [16]. Given this, a forward step is to investigate the case such that the latent features not only contain temporal correlation but also are constrained within certain boundaries.

When formulating the constrained latent features, there are two approaches. One is modelling with an encoding format $s = g(x; \theta)$, which projects unconstrained observation x to the constrained latent feature s, where θ denotes parameters. The other is modelling with a decoding format $x = f(s; \theta)$, where the latent features are generated within constraints. In this study, the latter is selected based on the following considerations.

- Parameters of the encoding format are usually learned as deterministic values [59, 60] and through extra trim algorithms [61]. To obtain consistent and robust results, the number of training samples should be sufficient for asymptotic convergence.
- The decoding format allows the probabilistic and Bayesian interpretation for parameter estimation [23]. More importantly, only desirable features are generated to improve learning efficiency.

Following the decoding format, a new description of the constrained dynamic latent feature is proposed in this work, which is combined with a non-linear observation function for latent variable modelling. A distinct characteristic of the proposed framework is that uncertainties of features and parameters are considered jointly, where appropriate probability distributions can be used to increase robustness.

A good example of constrained latent features with the decoding format can be found at the probabilistic PCA model [22], where each latent feature is restricted to be independent, and the observation noises of each dimension are constrained to be independent. By incorporating these regulations, the latent features become capable of decorrelating multivariate observations. For the dynamic model, besides, constraints have been imposed to form the probabilistic slow feature model [47]. Specifically, the transition coefficient a and variance σ^2 in (2.4) are restricted to the following constraints:

$$a \in (0,1), \qquad a^2 + \sigma^2 = 1. \tag{4.1}$$

With constraint (4.1), the standard Normal distribution $\mathcal{N}(0,1)$ will be a solution for the stationary distribution of the transition model (2.4). Thus, the zero-mean and unit-variance regulation in slow feature analysis [17] has been realized. More importantly, the constrained parameter "a" ensures positive temporal correlation and illustrates the "slowness" mathematically, which has been elaborated in Chapter 3. Thus, the aforementioned feature selection criterion, namely preference for smaller first order difference, can be materialized through a prior distribution of "a" in the previous chapter.

Because of the low quality and auto-correlated behaviours of raw process data, constructing a latent dynamic model has become a favourable option for extracting informative latent features. Other than the state space formulation in Chapter 3, the Markovian model [35] has also been widely applied for the transition function of latent features. This model describes the dynamic characteristics of discrete variables. However, the finite number of possible values limits its general modelling strength. For example, each possible value of the latent state s_t can only represent one possible transit mode. It reveals a switching behaviour, where different modes cannot happen at the same time. However, there are other cases where multiple models could contribute at the same time. In such cases, the latent state should no longer represent an index for the possible model but represent a set of probabilities for all possible models. It is another more important motivation for this study, where the latent feature of the conventional hidden Markov model is generalized to their possibilities.

4.2 Problem Statement

Instead of imposing a constraint on transition parameters, a constraint on the dynamic features s_t is considered. By considering a constraint on the $l1$ norm for the positive latent states, the conventional hidden Markov model is extended hierarchically. By considering boundaries for multiple latent features, robustness against significant deviations in raw observations can be achieved through the formulated constrained features. Consequently, in this chapter, a new transition model is proposed to describe the constrained dynamic features in a probabilistic way. Furthermore, we incorporate the proposed transition model into a feature extraction model and learn the corresponding parameters through the variational Bayesian inference.

The remainder of this paper is organized as follows. In the next section, a mathematical formulation for the constrained dynamic features is described through the comparison with its unconstrained counterpart, and the proposed model for feature extraction is also introduced. Following that, a probabilistic inference method is elaborated for the model fitting task, where particle methods are integrated within the variational Bayesian inference [39] to realize the dynamic features in the constrained domain. In the simulation and validation section, the proposed learning algorithm and the feature extraction model are validated through numerical examples, benchmark simulations, and an industrial case study.

4.3 Probabilistic Formulation

In this section, the probabilistic framework of the proposed feature extraction model for dynamic latent features will be provided. First, two approaches to consider the constrained dynamic latent feature are discussed along with constructing their transition functions. Following that, one possible observation function is presented with a non-linear form. The feature extraction model is then finalized.

4.3.1 Apply Dirichlet Distribution for Hierarchical Hidden Markov Model

In hidden Markov model, a transition matrix is usually used to model the transition behaviour, such as reviewed in Chapter 2. For example, if we redefine the discrete latent state as q_t, it transition behaviour is modelled with:

$$\Pr(q_t = i \mid q_{t-1} = j) = a_{i,j}, \quad \forall\ i,\ j \in \{1, ..., d\}, \tag{4.2}$$

where Pr stands for a probability, and the transition matrix $A_{d\times d} = [a_{i,j}]_{i,j} = 1, ..., d$ is the transition parameters. While parametrising the latent state q_t by its probabilities on each possible value, a continuous and constrained latent state s_t can be formed as

$$\Pr(q_t = i \mid s_t) = s_t^{(i)}, \quad \forall\ i \in \{1, ..., d\}, \tag{4.3}$$

$$where : s_t^{(i)} \geq 0, \quad \forall\, i \in \{1, ..., d\}, \tag{4.4}$$

$$\|s_t\|_1 = \sum_{i=1}^{d} |s_t^{(i)}| = \sum_{i=1}^{d} s_t^{(i)} = 1. \tag{4.5}$$

Regarding such continuous latent variable as a column random vector, the transition function in (4.2) is translated as

$$s_t^{(i)} = \sum_{j=1}^{d} \Pr(q_t = i \mid q_{t-1} = j) \Pr(q_{t-1} = j \mid s_{t-1}) = \sum_{j=1}^{d} a_{i,j}\ s_{t-1}^{(j)} = A_{i,:} \cdot s_{t-1},$$

$$\Rightarrow \quad s_t = A \cdot s_{t-1}. \tag{4.6}$$

In above transition function (4.6), the A is the only transition parameter. Comparing to the transition function for general continuous states, such as in (2.6), it lacks a parameter for the uncertainty, such as the covariance parameter Q. While the latent state s_t, instead of q_t, is directly involved in the observation function, the above transition function (4.6) limited the modelling strength by not considering the stochastic property of the latent features. Therefore, a novel transition function is developed for the latent state s_t, which representing the probability vector for the latent state in the conventional hidden Markov model. An illustration of such extending proposal is shown in Figure 4.1, where the connection of blue dots arrow will be presented.

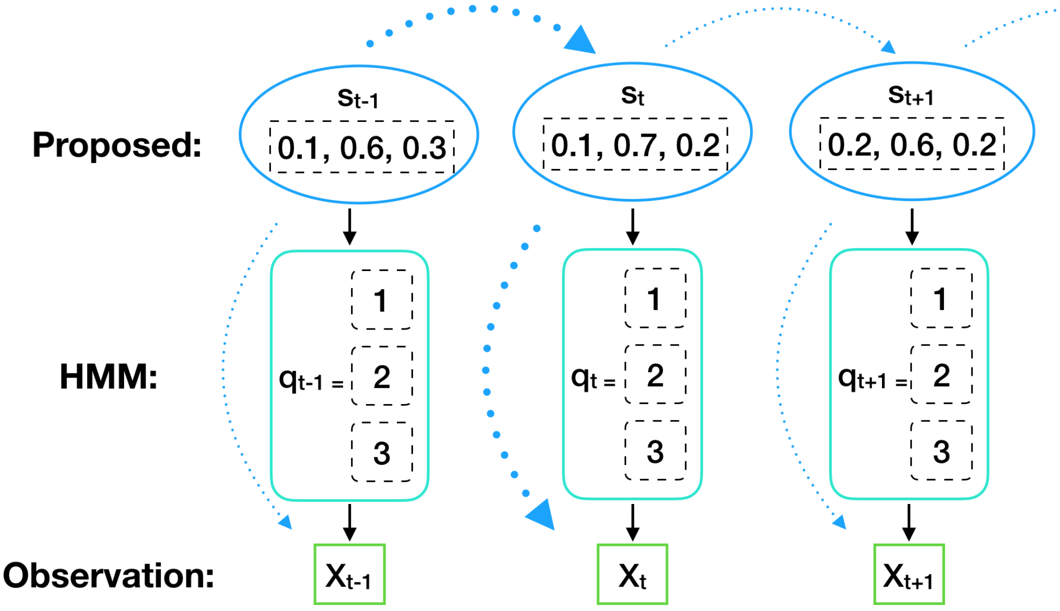


Figure 4.1: Extending Hidden Markov Model with Constrained Dynamic Latent Feature

From (2.4), we can state either that the transition noise is sampled from a zero-mean distribution or that s_t is distributed around the expected location $a \cdot s_{t-1}$. To ensure that

S_t satisfies the constraints in (4.4) and (4.5) at any given time t, Dirichlet distribution is introduced to replace the Normal distribution. The d dimensional Dirichlet distribution is defined as

$$Dir(x;\ \alpha) = \Gamma\{\sum_{i=1}^{d} \alpha_i\} \prod_{i=1}^{d} \frac{[x^{(i)}]^{\alpha_i - 1}}{\Gamma\{\alpha_i\}}, \tag{4.7}$$

where $\Gamma\{\cdot\}$ is the Gamma function. This distribution makes the mean of each dimension distributed around $\mathbb{E}[x^{(i)}] = \alpha_i / \sum \alpha_i$. Dirichlet distribution is utilized because it has the same support as the proposed states in (4.4) and (4.5). Following the conventional design of HMM, the transition equation (4.6) is used to model the expected position $\mathbb{E}[S_t]$. In addition, a precision variable ρ is introduced to make a stochastic transition function. The proposed process then has the transition probability as

$$p(S_t \mid S_{t-1}) = Dir(S_t;\ \rho \cdot A \cdot s_{t-1}). \tag{4.8}$$

The precision parameter ρ is a positive value determining how s_t distributes around the centre of $A \cdot s_{t-1}$. Comparing to the transition model designed in the previous Chapter 3, the transition matrix A can also be limited with one degree of freedom, such as

$$\begin{aligned} A_{i,i} &= a, \qquad a \in (0,1), \qquad \forall\ i \in 1...d, \\ A_{i,j} &= \frac{1-a}{d-1}, \qquad \forall\ i \neq j. \end{aligned} \tag{4.9}$$

Thus, the inertia represented by this stochastic process can be uniquely determined by the coefficient a. Besides that, the learning procedure for a single scaler a requires much less excitation than inferencing a complete transition matrix A.

This transition function has generalized the hidden Markov transition model by introducing the stochastic uncertainty for the probability vector of each mode. However, on the other hand, it also contains the dependency in the latent space. It means each latent dimension is not independent of each other: $S_{1:T}^{(i)} \not\perp S_{1:T}^{(j)}$, which could result with difficult in modelling the observations. Also, when it comes to high latent dimension cases, the effect of constraints becomes trivial for the feature dynamics. Thus, in the following content, above constrains (4.4) and (4.5) are applied to $d = 2$ case. Actually, for this case, the degree of freedom is one: $S_t^{(2)} = 1 - S_t^{(1)}$, and the latent feature can be treated as in the single latent dimension.

4.3.2 Transition Model Design with Beta distribution

In this part, the single dimensional dynamic feature $s_{1:T}$ with constraint $s_t \in (0,1)$, $1 \leqslant t \leqslant T$, is formulated from the probabilistic perspective. As can be seen, the primary constraint is selected as

$$s_t \in (0,1). \tag{4.10}$$

Other specified ranges can be regarded as linear projections from this primary range of $(0,1)$. To this end, the Beta distribution [62] is used to model the transition probability $p(s_t|s_{t-1})$, and two transition parameters, coefficient a and precision ρ, are used to govern the dynamic behaviours.

To ensure that s_t satisfies the constraint (4.10) at any given time t, Beta distribution is introduced to model the transition probability:

$$p(s_t \mid s_{t-1}) = Beta(s_t;\ \alpha_t, \beta_t) = \frac{(s_t)^{\alpha_t - 1}(1 - s_t)^{\beta_t - 1}}{\mathcal{B}\{\alpha_t, \beta_t\}}, \tag{4.11}$$

where $\alpha_t > 0$, $\beta_t > 0$, and $\mathcal{B}\{\cdot\}$ is the Beta function [62]. By interpreting α_t and β_t with $s_t^{(c)}$ and ρ, we can rewrite (4.11) as

$$p(s_t \mid s_{t-1}) = Beta(s_t \mid \underbrace{\rho\ s_t^{(c)}}_{\alpha_t}, \underbrace{\rho\ (1 - s_t^{(c)})}_{\beta_t}), \tag{4.12}$$

where $s_t^{(c)} = \frac{\alpha_t}{\alpha_t + \beta_t}$ denotes the expected location of (4.11), and $\rho = \alpha_t + \beta_t$ is considered as the precision of (4.11). Using the property of Beta distribution, the variance of (4.12) is computed:

$$Var\left[s_t \mid s_{t-1}\right] = \frac{s_t^{(c)}(1 - s_t^{(c)})}{\rho + 1}. \tag{4.13}$$

By adopting (4.9), the function for $s_t^{(c)}$ can be formulated as

$$\begin{bmatrix} s_t^{(c)} \\ 1 - s_t^{(c)} \end{bmatrix} = \begin{bmatrix} a & (1-a) \\ (1-a) & a \end{bmatrix} \begin{bmatrix} s_{t-1} \\ 1 - s_{t-1} \end{bmatrix}, \tag{4.14}$$

where $a \in (0,1)$. The expected position $s_t^{(c)}$ can thus preserve the same constraint, i.e., $s_t^{(c)} \in (0,1)$. Similar to the conventional models, in the proposed transition model, the expected location is positively correlated with s_{t-1}, and the variance is negatively correlated with ρ.

With the transition equation (4.12), the feature will arrive at a stationary state has given sufficient time, and information of the initial s_0 will gradually vanish because of the stabilizing coefficient a. After the arrival at the stationary state, statistical measures can be determined uniquely by a and ρ. In order to demonstrate these, several simulations are conducted with different parameters. Intuitively, the coefficient a determines the memory strength of a dynamic feature. In Figure 4.2, the estimated time to the stationary stage is plotted for different a, where the precision ρ is neglected because of no contribution of it on average. As expected, the memory strength (the amount of time to forget s_0) is proportional to coefficient a: the larger the parameter a, the stronger the memory is. Also, individual runs from three specific a's are plotted in sub-figures, where the mean trends of them are in black. With this figure, a range of the underlying a can be selected based on the preferred transition behaviour.

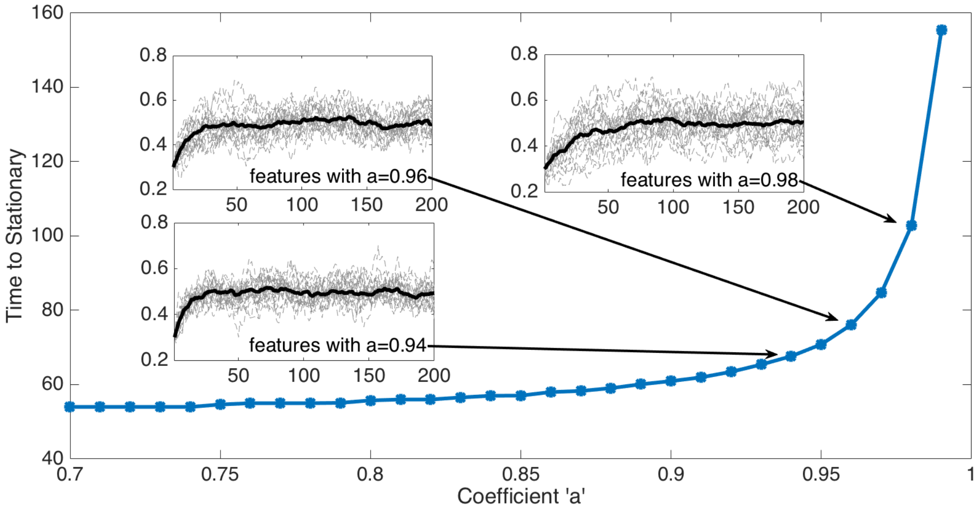


Figure 4.2: Transition Properties of Constrained Dynamic Features

At the stationary stage, the standard deviation (STD) $\sqrt{Var[s_{t\to\infty}]}$ and the expected velocity (mean squared first order difference) $\mathbb{E}\|s_t - s_{t-1}\|^2$ are studied to represent the variance level and the inertia level accordingly. Similar as in the aforementioned simulations, the averaged statistics are plotted in Figure 4.3. With the increase of precision ρ, both STD and velocity show a decreasing trend. This point can be interpreted by the fact that a less noisy transition model (4.12) gives smaller stationary variance and larger inertia. With the increase of coefficient a, on the other hand, STD shows an increasing trend, indicating that a stronger memory can distribute $s_{1:T}$ in a wider range. Because a larger a introduces larger inertia, a decreasing trend can be observed in the velocity curve. To visualize these statistics more clearly, two nominal features (one is from the uniform distribution on $[0.25, 0.75]$, and the other is the sinusoidal signal) are also plotted in Figure 4.3. As can be seen, a range of the underlying parameters can be selected roughly in advance, by comparing an observed feature with stationary behaviours shown in the sub-figures of Figure 4.3.

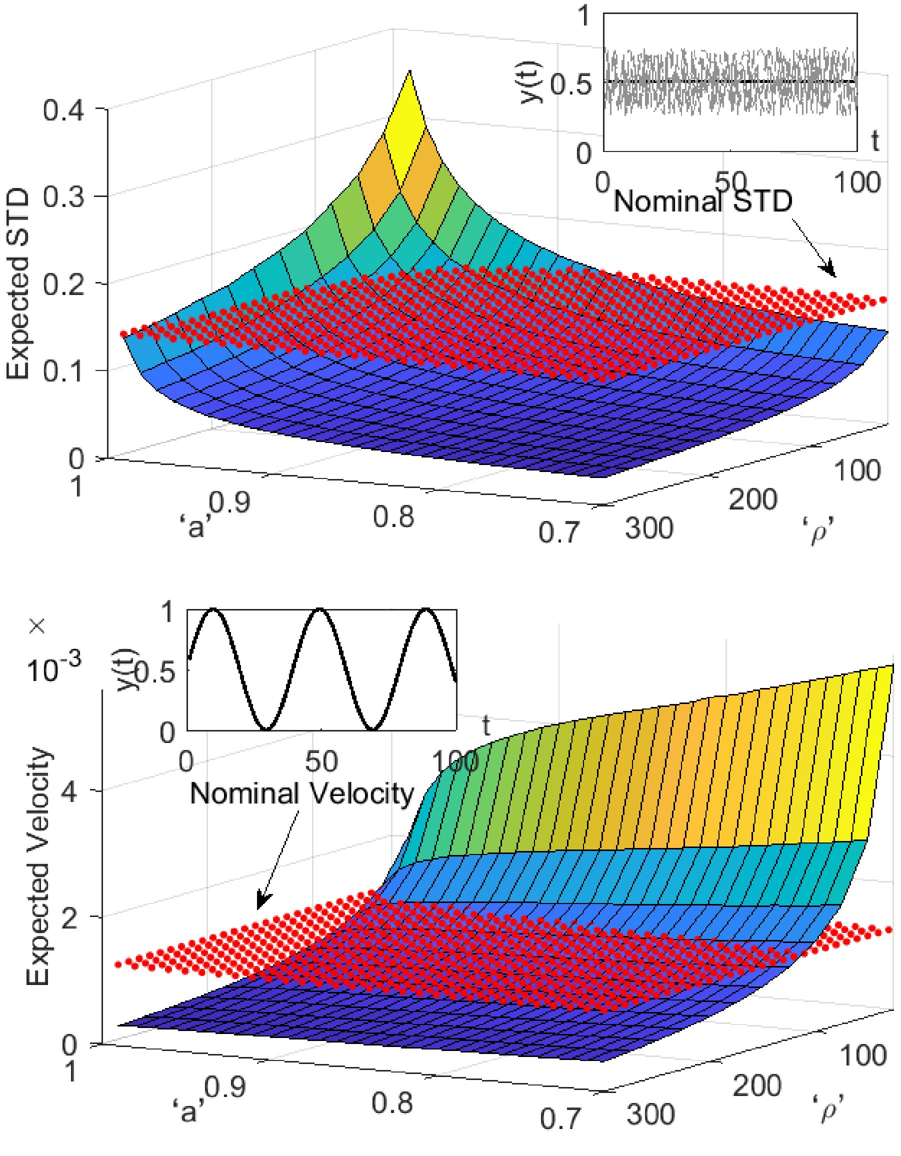


Figure 4.3: Stationary Properties of Constrained Dynamic Features

4.3.3 Emission Model Design

Based on the proposed transition model (4.12) and (4.14), the stochastic feature retains the ability to cover the constrained domain $(0, 1)$. While applied in latent variable modelling, this constrained domain shall be projected to the general unconstrained observation space $(-\infty, +\infty)$. The emission function considered in this paper is formulated as

$$X_t = b + H \cdot \ln S_t + v_t, \tag{4.15}$$

where the latent feature is introduced as multiple constrained dynamic features $S_t = [s_t^{(1)}, ..., s_t^{(d)}]'$, and the observed data sample X_t has dimension m. The logarithm operator is applied on each element to project $(0, 1)$ to $(-\infty, 0)$. Together with matrix H and bias b, the latent space is connected with the unconstrained observation space. Corresponding uncertainty is introduced by the noise term $v_t \sim \mathcal{N}(0, R)$.

Comparing (4.15) with the encoding feature learning cell with logistic function [11, 59]:

$$s_t = \frac{e^{x_t}}{1 + e^{x_t}} \;\Leftrightarrow\; x_t = \ln s_t - \ln(1 - s_t),$$

a difference is that $\ln(1 - s_t)$ is neglected. In latent variable models, since the dimension of latent features, d, is determined from data, both s_t and $(1 - s_t)$ can be potentially extracted by the learning procedure. Besides that, the element H_{ij} of emission matrix determines the direction from latent dimension j to observation dimension i. By extracting multiple latent features and learning the latent dimension d from data, our proposed emission model (4.15) shares the similar modelling scope as this widely applied encoding function.

4.3.4 Prior Distributions

Now, the feature extraction model considered in this paper has been formulated as following: Equation (4.12) gives the linear but non-Gaussian transition model, and Equation (4.15) denotes the non-linear emission model. To perform parameter estimation and feature learning, the Bayesian inference approach is adopted. To this end, prior distributions of the parameters in (4.15) are assigned for automatic relevance determination [36]:

$$p(b \mid \Sigma_b) = \mathcal{N}(b;\; 0, \Sigma_b), \tag{4.16}$$

$$p(H \mid \Sigma_h) = p([h_1, ..., r_m]' \mid \Sigma_h) = \prod_{j=1}^{m} \mathcal{N}(h_j;\; 0, \Sigma_h), \tag{4.17}$$

$$p(R \mid \alpha_r, \beta_r) = p(diag\{r_1^{-1}, ..., r_m^{-1}\} \mid \alpha_r, \beta_r) = \prod_{j=1}^{m} \mathcal{G}(r_j;\; \alpha_r, \beta_r). \tag{4.18}$$

Here, similar to the probabilistic PCA [22], the noise in each observation dimension is assumed to be independent, making the covariance R a diagonal matrix specified in (4.18) with Gamma distribution $\mathcal{G}(\cdot)$. The hyper-parameters, matrices Σ_b and Σ_h, as well as α_r and β_r are given as the prior information.

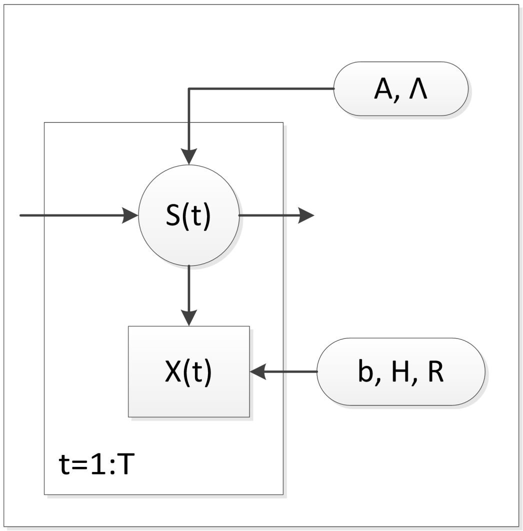


Figure 4.4: Proposed Bayesian design for constrained dynamic feature extraction

Since the multiple latent features are expected to be independent of each other, the parameters of transition model (4.12) and (4.14) can be presented as diagonal matrices $A = diag\{a_1, ..., a_d\}$ and $\Lambda = diag\{\rho_1, ..., \rho_d\}$. To cover all the corresponding support domain, the Beta distribution and the Gamma distribution are selected as the prior of the transition coefficients and precision parameters, respectively; as for the initial latent state, the uniform distribution $U(0, 1)$ is used to cover its support domain $(0, 1)$:

$$p(A \mid \alpha_a, \beta_a) = \prod_{i=1}^{d} Beta(a_i;\ \alpha_a, \beta_a), \tag{4.19}$$

$$p(\Lambda \mid \alpha_\rho, \beta_\rho) = \prod_{i=1}^{d} \mathcal{G}(\rho_i;\ \alpha_\rho, \beta_\rho), \tag{4.20}$$

$$p(S_1) = \prod_{i=1}^{d} U(S_1^{(i)};\ 0, 1). \tag{4.21}$$

According to Figure 4.2 and Figure 4.3, the hyper-parameters, α_a, β_a, α_ρ, and β_ρ, can be selected to represent preferred ranges of a and ρ. With these prior distributions, a probabilistic graphical model of the problem considered is shown in Figure 4.4, from which the complete joint probability distribution between observations and all random variables,

$\mathcal{P} \equiv p(X_{1:T}, S_{1:T}, A, \Lambda, b, H, R)$, can be written as

$$\mathcal{P} = \prod_{t=1}^{T} p(X_t \mid S_t, b, H, R) \cdot p(b) \cdot p(H) \cdot p(R)$$
$$\cdot \prod_{t=2}^{T} p(S_t \mid S_{t-1}, A, \Lambda) \cdot p(S_1) \cdot p(A) \cdot p(\Lambda). \tag{4.22}$$

4.4 Variational Inference Methods

In this section, we adopt the variational Bayesian inference (VB) approach [39] to approximate the posterior distribution of the parameters and dynamic latent features. A general introduction of this approach can be found in Chapter 2. In Figure 4.4, these variables have been grouped into three sets based on their probabilistic dependencies, thus the proposal distribution is factorized as

$$q(S_{1:T}) \cdot q(A, \Lambda) \cdot q(H, b, R)$$
$$\to p(S_{1:T}, A, \Lambda, b, H, R \mid X_{1:T}), \tag{4.23}$$

where latent features $S_{1:T}$ form their own group, A and Λ form a second group, and the emission parameters form the last one. This "mean field" approximation implements the independence among groups to facilitate the learning procedure [39]. In addition, this factorization for dynamic latent features can make $q(A, \Lambda)$ a compact result to make predictions on new data. Mathematically, the approximation (4.23) is usually evaluated by the Kullback-Leibler divergence $D_{KL}[q||p]$ for tractability. The variational Bayesian Expectation-Maximization strategy [39] is then applied to update $q(S_{1:T})$, $q(A, \Lambda)$, and $q(H, b, R)$ by minimizing this divergence. For example, the updating equation for $q(H, b, R)$ is presented as

$$\begin{aligned}
&\min_{q(H,b,R)} D_{KL}\left[q(H,b,R)q(A,\Lambda)q(s_{1:T}) \parallel p(s_{1:T}, A, \Lambda, H, b, R \mid x_{1:T})\right] \\
\Leftrightarrow &\max_{q(H,b,R)} \int_{q_{H,b,R}} \int_{q_{A,\Lambda}} \int_{q_{S_{1:T}}} \ln \frac{p(X_{1:T}, S_{1:T}, A, \Lambda, b, H, R)}{q(H,b,R)\ q(A,\Lambda)\ q(S_{1:T})} \\
\Leftrightarrow &\max_{q(H,b,R)} \int_{q_{H,b,R}} \left[\mathbb{E}_{q(A,\Lambda)q(S_{1:T})} \ln \mathcal{P} - \ln q(H,b,R)\right] \\
\Leftrightarrow &\min_{q(H,b,R)} D_{KL}\left[q(H,b,R) \parallel c \cdot e^{\mathbb{E}_{q(A,\Lambda)q(S_{1:T})} \ln \mathcal{P}}\right].
\end{aligned} \tag{4.24}$$

The first equivalence in above derivations uses the Bayes' rule as well as the fact that the evidence probability $p(X_{1:T})$ is irrelevant with these proposal distributions [23]. The second equivalence is based on the fact that $q(S_{1:T})$ and $q(A, \Lambda)$ stay constant in this step. With re-organization of the third equivalence, the optimal solution can be obtained by minimizing this step-wise $D_{KL}[q||p]$ in the last row:

$$\ln q^*(H,b,R) = \mathbb{E}_{q(s_{1:T})q(A,\Lambda)} \ln \mathcal{P} + const.. \tag{4.25}$$

Other updating equations will be formulated similarly, and an overall diagram for the proposed learning algorithm is summarized in Figure 4.5. For the initialization procedure, three components are realized sequentially to start the iterative updating. It will ensure that only desirable latent features will be utilized; also, with such initialized emission parameters, the following estimation for latent features will not result with most near-boundary samples. Since the transition parameters and emission parameters are not dependent on each other, the corresponding updating can be performed in parallel. With the proposed constrained features, technical challenges appear when formulating $q^*(s_{1:T})$ and $q^*(A,\Lambda)$, which will be illustrated in details.

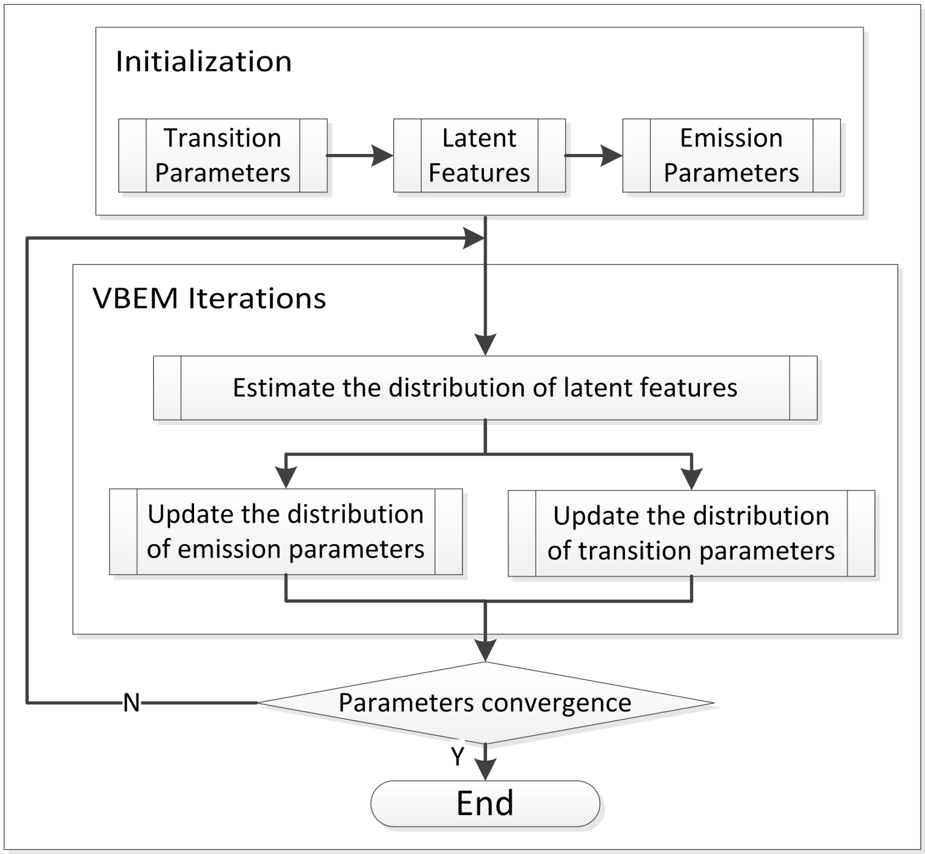

Initialization
Transition Parameters
Latent Features
Emission Parameters
VBEM Iterations
Estimate the distribution of latent features
Update the distribution of emission parameters
Update the distribution of transition parameters
N
Parameters convergence
Y
End

Figure 4.5: Learning Algorithm for the Proposed Feature Extraction Model

4.4.1 Estimate Latent Features with Revised Particle Smoother

Estimating the proposal distribution for a hidden Markovian sequence $q(s_{1:T})$ usually shares the same objective as the smoothing task for $p(s_{1:T} \mid x_{1:T})$. However, in the VB framework, this estimation interacts with the other proposal distributions:

$$\begin{aligned} \ln q^*(S_{1:T}) &= \mathbb{E}_{q(H,b,R)q(A,\Lambda)} \ln \mathcal{P} + const. \\ = \; & \mathbb{E}_{q(H,b,R)q(A,\Lambda)} \ln p(S_{1:T} \mid X_{1:T}, A, \Lambda, H, b, R) + const'. \\ = \; & \sum_{t=2}^{T} \mathbb{E}_{q(A,\Lambda)} \ln p(S_t \mid S_{t-1}, A, \Lambda) + \ln p(S_1) \\ & + \sum_{t=1}^{T} \mathbb{E}_{q(H,b,R)} \ln p(X_t \mid S_t, H, b, R) + const'.. \end{aligned} \tag{4.26}$$

In unconstrained case, the linear transition model (2.5) propagates the Normal distribution from S_{t-1} to S_t, and the linear emission function retains the Normal distribution for the posterior $p(S_t \mid X_{1:T})$. Therefore, $q(S_{1:T})$ can be parametrised with their mean and covariance, which will be updated explicitly through forward and backward algorithms. In this study, however, the proposed transition model (4.12) does not preserve above features because of the non-conjugate connections. To solve this problem, particle based approaches are adopted.

Generally, the particle state estimation consists of two basic mechanisms, sampling and weighing [40]. To simplify the statistical expectation in terms of $q(H, b, R)$, a nominal $\tilde{X}_t$ is introduced to replace a part of (4.26):

$$\ln p(\tilde{X}_t \mid S_t) = \mathbb{E}_{q(H,b,R)} \ln p(X_t \mid S_t, H, b, R). \tag{4.27}$$

By using $\tilde{X}_t$ to represent the sufficient information of $q(H, b, R)$ (without any approximation), the weighing procedure of generated particles can be realized in the conventional way.

The emission function contributes to the updating of $q(S_{1:T})$ through the following function:

$$\begin{aligned} \mathbb{E}_{q_{(H,b,R)}} \ln p(X_t \mid S_t, H, b, R) = & -\frac{1}{2} tr\{\langle H' R^{-1} H \rangle \ln S_t \ln S_t'\} \\ & + \langle (x_t - b)' R^{-1} H \rangle \ln S_t - \frac{1}{2} \langle (X_t - b)' R^{-1} (X_t - b) \rangle, \end{aligned} \tag{4.28}$$

where involved statistics, $\langle H' R^{-1} H \rangle$, $\langle b' \; R^{-1} H \rangle$, $\langle R^{-1} H \rangle$, $\langle b' \; R^{-1} b \rangle$, $\langle b' \; R^{-1} \rangle$, and $\langle R^{-1} \rangle$ can be obtained from above derivations.

As for the other part of (4.26), the expected diagonal elements of A and Λ are used to approximate the expectation in terms of $q(A, \Lambda)$:

$$\begin{aligned} \ln \tilde{p}(S_t \mid S_{t-1}) = \textstyle\sum_{i=1}^{d} \ln p(S_t^{(i)} \mid S_{t-1}^{(i)}, \langle a_i \rangle, \langle \rho_i \rangle) \\ \approx \mathbb{E}_{q(A,\Lambda)} \ln p(S_t \mid S_{t-1}, A, \Lambda). \end{aligned} \tag{4.29}$$

Thus, $\langle a_i \rangle$ and $\langle \rho_i \rangle$ can be used as deterministic values to generate and propagate particles. We will illustrate these two statistics within the updating procedure of $q(A, \Lambda)$ shortly.

Besides (4.27) and (4.29), a new strategy is proposed for state representation. Instead of completely relying on particles and their weights to describe latent states[40], intermediate state distributions are fitted to represent the particles. Specifically, after assigning weights at each time instant, the traditional re-sampling procedure is replaced by a procedure of fitting an intermediate distribution. By doing this, extreme particle weights can be avoided, and more importantly, the backward propagation can be simplified considerably.

Accordingly, the following intermediate distributions are proposed: the (one-step) predicted distribution $q(S_t|X_{1:t-1})$ and the filtered distribution $q(S_t|X_{1:t})$ are approximated with Beta distribution in forward path; as well as $q(S_t|X_{1:T})$ in backward path, accordingly:

$$q(S_t \mid X_{1:t-1}) \approx \prod_{i=1}^{d} Beta\left(S_t^{(i)} \mid [\alpha_t^{(p)}]^{(i)}, [\beta_t^{(p)}]^{(i)}\right),$$

$$q(S_t \mid X_{1:t}) \approx \prod_{i=1}^{d} Beta\left(S_t^{(i)} \mid [\alpha_t^{(f)}]^{(i)}, [\beta_t^{(f)}]^{(i)}\right),$$

$$q(S_t \mid X_{1:T}) \approx \prod_{i=1}^{d} Beta\left(S_t^{(i)} \mid [\alpha_t^{(s)}]^{(i)}, [\beta_t^{(s)}]^{(i)}\right).$$

The above usage of the Beta distribution is motivated from its support domain $[0, 1]$ as well as its ability to capture the uni-mode distribution. Based on the transition model (4.12), for each single dimensional feature $s_{1:T}$, if $q(s_{t-1})$ can be formed as a uni-mode distribution, $q(s_t)$ will preserve this mode due to transition behaviour:

$$\begin{aligned} s_t^{(mode)} \mid s_{t-1} = \arg\max \int_{s_{t-1}} p(s_t|s_{t-1})p(s_{t-1}) \\ = (2a-1)\ s_{t-1}^{(mode)} + 1 - a. \end{aligned} \tag{4.30}$$

Assuming the predicted distribution $p(s_t \mid X_{1:t-1})$ as a uni-mode distribution and has been approximated by $Beta(s_t \mid \alpha_t^{(p)}, \beta_t^{(p)})$. The observation function is simplified from (4.28) to the one dimension case. The filtered distribution is then presented as

$$\ln p(s_t \mid x_{1:t}) = \ln p(s_t \mid X_{1:t-1}) + \ln p(X_t \mid s_t)$$

$$\approx -\frac{k_1}{2}(\ln s_t)^2 + k_2 \ln s_t + (\alpha_t^{(p)} - 1)\ln s_t + (\beta_t^{(p)} - 1)(1 - \ln s_t), \qquad (4.31)$$

where $k_1(> 0)$ and k_2 are generalized from (4.28). As long as $\beta_t^{(p)} > 1$, the first order derivative of (4.31) will be monotonously decreasing and will have only one feasible root, which leads $p(s_t \mid X_{1:t})$ to be the uni-mode distribution. Thus, with parameters α and β fitted accurately, for example by the algorithm from [62], Beta distribution can represent those weighted particles within $(0, 1)$ reasonably well.

Based on above modifications for adopting particle state estimation under the VB framework, the learning procedure for $q(S_{1:T})$ can now be formulated as a standard forward-backward algorithm. In the following, i-th latent dimension is denoted with superscript "(i)", and k-th individual particle is denoted by "$[k]$". The prediction step consists of sampling particles for S_{t-1}, propagating them to S_t, and estimating intermediate parameters:

$$\begin{aligned}
&S_{t-1}^{(i)}[k] \sim Beta([\alpha_t^{(f)}]^{(i)}, [\beta_t^{(f)}]^{(i)}), && \forall\ k = 1, ..., N,\\
&S_t^{(i)}[k] \sim \tilde{p}(s_t \mid s_{t-1}^{(i)}(k)), && \forall\ k = 1, ..., N,\\
&[\alpha_t^{(p)}]^{(i)},\ [\beta_t^{(p)}]^{(i)} \xleftarrow{Fit} \{S_t^{(i)}[1, ..., N]\}, && \forall\ i = 1, ..., d. \qquad (4.32)
\end{aligned}$$

Note that $\tilde{p}$ is introduced in (4.29), where $\langle a_i \rangle$ and $\langle \rho_i \rangle$ are involved. In the correction step, weights are provided to these samples by considering all d dimensions, and individual Beta distributions are then estimated for each latent dimension:

$$\begin{aligned}
&w_t[k] \propto p(\tilde{X}_t \mid S_t[k]), && \forall\ k = 1, ..., N,\\
&[\alpha_t^{(f)}]^{(i)},\ [\beta_t^{(f)}]^{(i)} \xleftarrow{Fit} \{s_t^{(i)}[1, ..., N],\ w_t[1, ..., N]\}, && \forall\ i = 1, ..., d. \qquad (4.33)
\end{aligned}$$

One typical advantage of introducing these intermediate Beta distributions is to simplify the backward propagation (or smoothing) step. In the conventional particle smoothers, the updating equation for $p(S_t \mid X_{1:T})$ is computational intense, where two sampling steps are required [40]. In the proposed method, the smoothing step becomes more explicit with the imposed Beta distribution:

$$\begin{aligned}
p(S_t \mid X_{1:T}) &= p(S_t \mid X_{1:t}) \int \frac{p(S_{t+1} \mid S_t)\ p(S_{t+1} \mid X_{1:T})}{p(S_{t+1} \mid X_{1:t})} dS_{t+1}\\
&\approx p(S_t \mid X_{1:t})\ \mathbb{E}_{q(S_{t+1} \mid \alpha_{t+1}^{(s)}, \beta_{t+1}^{(s)})} \left[\frac{p(S_{t+1} \mid S_t)}{q(S_{t+1} \mid \alpha_{t+1}^{(p)}, \beta_{t+1}^{(p)})} \right]. \qquad (4.34)
\end{aligned}$$

At each time step, since $\{\alpha_{t+1}^{(s)}, \beta_{t+1}^{(s)}\}$ and $\{\alpha_{t+1}^{(p)}, \beta_{t+1}^{(p)}\}$ are available, only one sampling and one weighing are required to compute (4.34) with particles, which is specified by

$$\hat{S}_t^{(i)}[k] \sim Beta([\alpha_t^{(f)}]^{(i)}, [\beta_t^{(f)}]^{(i)}), \qquad \forall\ k = 1, ..., N, \quad \forall\ i = 1, ..., d, \qquad (4.35)$$

$$\hat{w}_t[k] \propto \mathbb{E}_{q(S_{t+1}|\alpha^{(s)}_{t+1},\beta^{(s)}_{t+1})}\left[\frac{p(S_{t+1}\mid \hat{S}_t[k])}{q(S_{t+1}\mid \alpha^{(p)}_{t+1},\beta^{(p)}_{t+1})}\right], \qquad \forall\, k=1,...,N, \tag{4.36}$$

$$[\alpha_t^{(s)}]^{(i)},\ [\beta_t^{(s)}]^{(i)} \xleftarrow{Fit} \{\hat{s}_t^{(i)}[1,...,N],\ \hat{w}_t[1,...,N]\}, \qquad \forall\, i=1,...,d. \tag{4.37}$$

Since this sampling and weighing only depend on the adjacent feature states, its computational efficiency can be further improved by parallel computing among latent dimensions. A flow diagram shown in Figure 4.6 summarizes the proposed algorithm for feature estimation.

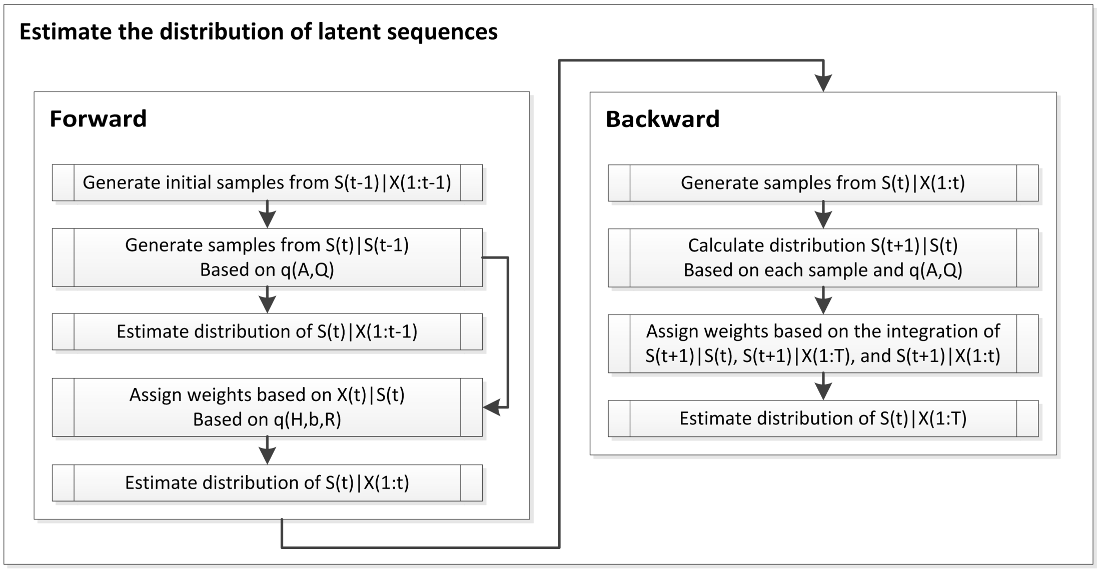

Estimate the distribution of latent sequences
Forward
Generate initial samples from S(t-1)|X(1:t-1)
Generate samples from S(t)|S(t-1)
Based on q(A,Q)
Estimate distribution of S(t)|X(1:t-1)
Assign weights based on X(t)|S(t)
Based on q(H,b,R)
Estimate distribution of S(t)|X(1:t)
Backward
Generate samples from S(t)|X(1:t)
Calculate distribution S(t+1)|S(t)
Based on each sample and q(A,Q)
Assign weights based on the integration of
S(t+1)|S(t), S(t+1)|X(1:T), and S(t+1)|X(1:t)
Estimate distribution of S(t)|X(1:T)

Figure 4.6: Updating Latent Feature with Particle Based State Estimation

4.4.2 Estimate Transition Parameters with Revised Importance Sampling

Since each latent feature $S^{(i)}_{1:T}$ is modelled as independent random variables, the corresponding updating procedure of transition parameter $a^{(i)}$ and $\rho^{(i)}$ can be derived individually. Based on the probability dependency in Figure 4.4, the updating equation is simplified to

$$\begin{aligned}\ln q^*(a^{(i)},\rho^{(i)}) &= \mathbb{E}_{q(H,b,R)q(S_{1:T})} \ln \mathcal{P} + const. \\ &= \mathbb{E}_{q(S_{1:T})} \ln p(S^{(i)}_{1:T} \mid a^{(i)},\rho^{(i)}) + \ln p(a^{(i)},\rho^{(i)}) + const'., \end{aligned} \tag{4.38}$$

where *const.* and *const*′. are normalizing constants. In the following, the superscript related to the latent dimension is omitted, and the single dimensional feature $s_{1:T}$ is investigated for simplicity. Since the updated $q^*(a,\rho)$ will only be utilized for calculating $\langle a\rangle$ and $\langle\rho\rangle$ as in the previous discussions, this section will focus on deriving the representative estimates for these two statistics.

First, the log-likelihood of $s_{1:T}$ in (4.38) can be factorized as

$$\ln p(s_{1:T} \mid a, \rho) = \ln p(s_1) + \sum_{t=2}^{T} \ln p(s_t \mid s_{t-1}, a, \rho).$$

To investigate the likelihood on transition parameters, the logarithm of conditional probability distribution between consecutive states is expended as

$$\begin{aligned}\ln p(s_t \mid s_{t-1}, a, \rho) = &- \ln\Gamma\{\rho(2a-1)s_{t-1} + \rho(1-a)\} \\ &- \ln\Gamma\{\rho a - \rho(2a-1)s_{t-1}\} + \rho(2a-1)\ s_{t-1}\ \ln\frac{s_t}{1-s_t} \\ &+ [\rho - \rho a - 1]\ln s_t + [\rho a - 1]\ \ln(1-s_t) + \ln\Gamma\{\rho\}. \end{aligned} \tag{4.39}$$

With the objective function defined as $Q(a, \rho)$, the gradients of a and ρ are derived as follows for Maximum a Posterior estimation:

$$\begin{aligned}\frac{\partial Q}{\partial a} =& \rho \cdot \sum_{t=2}^{T} (2\langle s_{t-1}\rangle - 1)\,[\ln\langle s_t\rangle - \psi\{\rho\langle c_t\rangle\}] \\ &+ (1 - 2\langle s_{t-1}\rangle)\ \ [\ln(1-\langle s_t\rangle) - \psi\{\rho(1-\langle c_t\rangle)\}] \\ &+ (\alpha_a - 1)/a - (\beta_a - 1)/(1-a), \end{aligned} \tag{4.40}$$

$$\begin{aligned}\frac{\partial Q}{\partial \rho} =& \sum_{t=2}^{T} \langle c_t\rangle \cdot \ln\langle s_t\rangle + (1-\langle c_t\rangle)\cdot\ln(1-\langle s_t\rangle) \\ &- \langle c_t\rangle \cdot \psi\{\rho\langle c_t\rangle\} - (1-\langle c_t\rangle)\cdot\psi\{\rho(1-\langle c_t\rangle)\} \\ &+ (1-\alpha_\rho)/\rho + \beta_\rho/\rho^2, \end{aligned} \tag{4.41}$$

where $\langle c_t\rangle = (2a-1)\langle s_{t-1}\rangle + 1 - a$ and $\psi\{\cdot\}$ is the Di-gamma function.

The expansion of $\ln p(s_t \mid s_{t-1}, a, \rho)$ is provided in Equation (4.39). In the VB framework, joint terms about two consecutive latent states, s_t and s_{t-1}, should be evaluated from $q(s_{1:T})$. In the linear and unconstrained case, the joint distribution $q(s_t, s_{t+1})$ can be derived for corresponding statistical expectations. As for the constrained case, the expectation of above log-likelihood is approximated as

$$\mathbb{E}_{q(s_{1:T})} \ln p(s_{1:T} \mid a, \rho) \approx \ln p(\langle s_{1:T}\rangle \mid a, \rho) \equiv Q(a, \rho), \tag{4.42}$$

where $\langle\cdot\rangle$ stands for expected value from $q(s_{1:T} \mid \alpha^{(s)}_{1:T}, \beta^{(s)}_{1:T})$. The main rationale for this approximation consists of two aspects. First, since the intermediate distributions have been developed for $q(s_{1:T})$, the joint distribution $q(s_t, s_{t+1})$ should also be used. However, as being presented by (4.39), complicated consecutive terms, such as $s_{t-1}\ln s_t$ and $\ln\Gamma\,[\rho(2a-1)s_{t-1} + \rho(1-a)]$, prevent an explicit integration. Second, the theoretically compact method, sampling whole sequences from $q(s_{1:T})$, is inefficient for the long and multivariate sequence, where the sampled sequences can be easily distorted. Instead, with correct parameters for intermediate state distribution, the expected sequence $\langle s_{1:T} \rangle$ would be representative. Thus, above approximation is considered as an effective and efficient way to represent the latent sequence for updating the transition parameters.

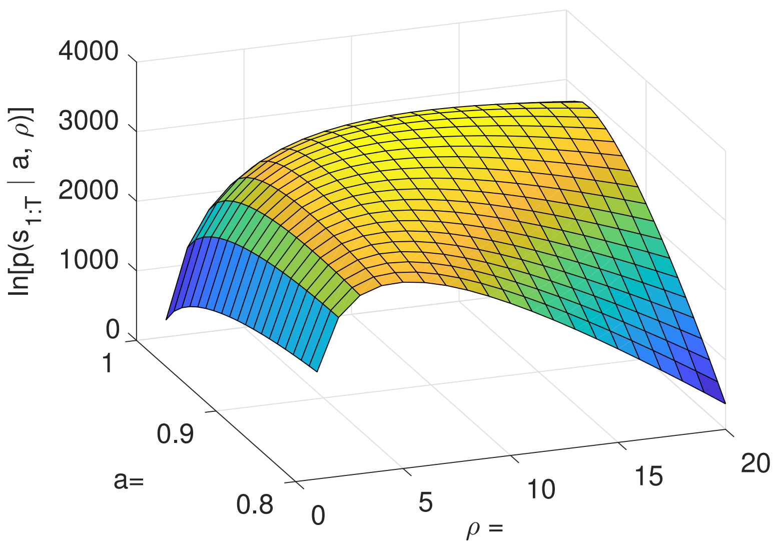


Figure 4.7: Log-Likelihood Function for a and ρ

To estimate $\langle a \rangle$ and $\langle \rho \rangle$ from (4.42), a preliminary solution can be provided through Maximum-a-Posterior estimation. With the detailed equation for $\ln p(s_{1:T}|a, \rho)$, it can be easily proved to be a differentiable and concave function, which is illustrated in Figure 4.7 as an example. The objective function, as a combination of $Q(a, \rho)$ and the logarithm of prior distributions (4.19) and (4.20), is still a differentiable and concave function. Therefore, a global optimum can be obtained through gradient-based optimization methods. The required derivatives are provided in (4.40) and (4.41). As for the proposed variational inference approach, we use a modified importance sampling method to estimate the distribution $q(a)$ and $q(\rho)$.

Basic importance sampling method [63] can be applied to generate samples from the prior distribution and assign weights according to likelihood. However, for the two dimensional weighing function shaped as in Figure 4.7, it is not efficient to calculate the weight for each $\rho^{(j)}$ through the conventional marginalization method:

$$w_\rho^{(j)} = \sum_{i=1}^{N_a} Q(a^{(i)}, \rho^{(j)}), \tag{4.43}$$

where N_a is the total number of particles associated with each $\rho^{(j)}$. The main reason is that weighing function $\ln p(\langle s_{1:T}\rangle | a, \rho)$ can vary in a very large range for a given ρ. See the case with $\rho^{(j)} = 15$ in Figure 4.7. The result from (4.43) will be almost completely determined by the area with lower (log-)likelihood. Thus, the high likelihood area, the yellow part in Figure 4.7, will be less represented. To develop a more meaningful estimate of $\langle \rho \rangle$, the maximization operation is utilized as another marginalising method, resulting in

$$w_\rho^{(j)} = \max_{i=1\ldots N_a} Q(a^{(i)}, \rho^{(j)})\,. \tag{4.44}$$

Graphically, the highest ridge of each direction in Figure 4.7 is selected as the marginalized distribution. Thus, the "meaningful" (with higher weight) area will no longer be concealed by the "meaningless" area and will receive more attention after marginalization. Based on the marginalized weights, a Gamma distribution is fitted to summarize N_ρ number of particles $\rho^{(1:N_\rho)}$ and their weights $w_\rho^{(1:N_\rho)}$, and a Beta distribution is obtained from $a^{(1:N_a)}$ and $w_a^{(1:N_a)}$. Corresponding $\langle a \rangle$ and $\langle \rho \rangle$ can then be determined as expectation values from the fitted distributions.

4.5 Simulations

This section provides validation of the proposed designs from three aspects. First, a simulated example is designed to validate the proposed estimation algorithm, illustrating the learning ability for latent feature and model parameters. Then, a practical feature extraction procedure is established and applied to a benchmark dataset for regression task on time series data. At last, an application to industrial data is performed to validate the usefulness of the proposed feature extraction model.

4.5.1 Numerical Examples

To validate proposed estimation algorithms, constrained latent features and observations are simulated with given transition model (4.12) and emission model (4.15). For instant, 3-dimension latent features with 5-dimension observations are generated for 100 samples.

To test the core part of the proposed learning algorithm as shown in Figure 4.5, the estimate of latent feature $q(s_{1:T})$ is compared first with actual model parameters. One

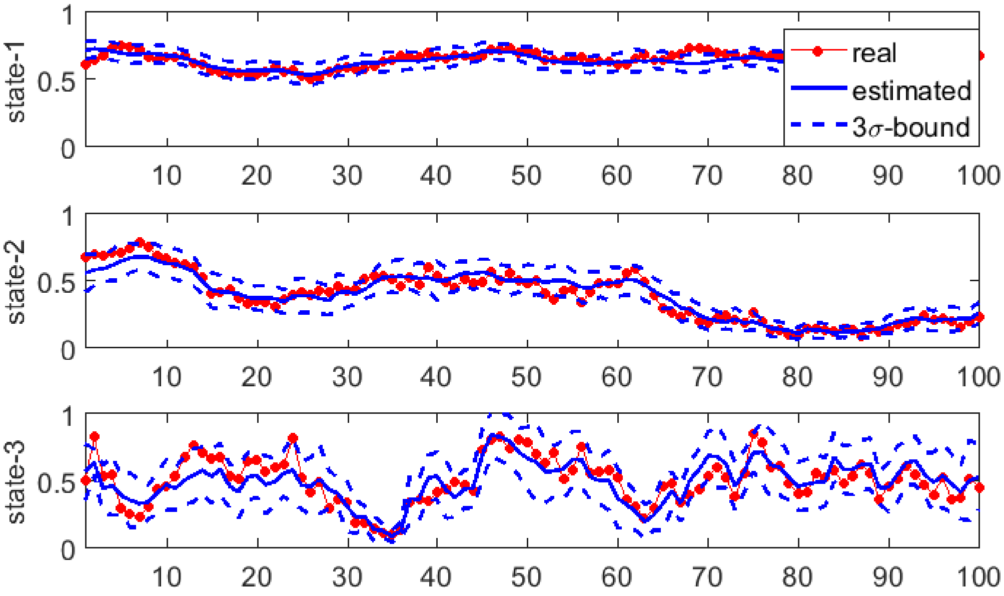


Figure 4.8: Estimation of Constrained Dynamic Latent Features with Model

example of the estimated results is presented in Figure 4.8, where the three features possess distinct transition parameters: $[a_1 = 0.99, \rho_1 = 300]$, $[a_2 = 0.95, \rho_2 = 100]$, and $[a_3 = 0.9, \rho_3 = 20]$. It can be observed that for these different latent dynamics, all the estimated features (blue curves) are able to capture the corresponding real value (red curves). Also the estimated bounds (based on the standard deviation of fitted Beta distribution) can cover most of values.

Next, the proposed updating procedure for transition parameters $q(a, \rho)$ is verified. In particular, the proposed modification on importance sampling (4.44) is compared with basic importance sampling, where the target feature $s_{1:T}$ is generated with the real transition parameters. In Figure 4.9, the two updated distributions for a and ρ are plotted along with their real values and the utilized prior distributions. The numbers in legend boxes represent the corresponding estimates of $\langle a \rangle$ and $\langle \rho \rangle$. It can be observed that the maximization based marginalization (pos-max) can capture the real parameter value from the relatively trivial prior for both the coefficient a and the precision ρ. In the top plot for the probability density function $p(a)$, the result of the conventional marginalizing method (pos-sum) is roughly distributed around the real value (the red dot). Although its statistic $\langle a \rangle$ captured the real value accurately, it can be seen that the shape of the marginalized distribution is skewed. As for the result from "pos-max", the shape of the density function is well balanced,

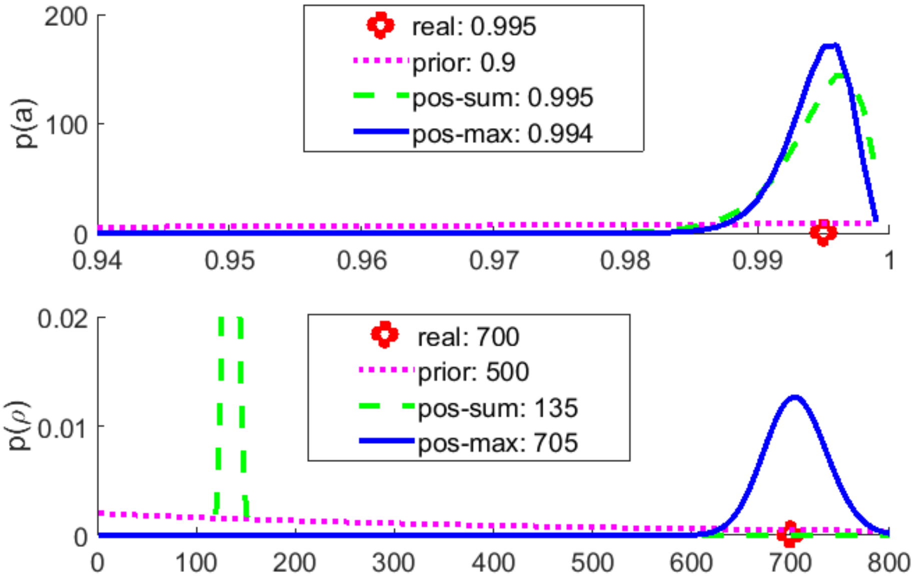


Figure 4.9: Estimation of Transition Parameters for Constrained Dynamic Feature

and a more narrow bell-shape is given to bring the maximum point closer to the real value. In the bottom plot for the probability density function $p(\rho)$, it is clear that "pos-sum" is failed in estimating the precision parameter ρ. Whereas, "pos-max" has demonstrated its strength. Furthermore, the Monte Carlo tests (20 runs for each) are performed for multiple ρ's and the fixed $a = 0.995$. Table 4.1 compares the mean and standard deviation from these two marginalization methods. Based on these, the effectiveness of proposed method (4.44) has been validated.

Table 4.1: Validation Results of Estimating Precision Parameter

ρ^*	100	300	500	700	900
$\langle\rho\rangle_{sum}$	18±7.9	59±22	95.7±38	134±49	176±77
$\langle\rho\rangle_{max}$	101±3.9	302±20	511±24	702±32	904±41

Based on above verifications, the complete learning algorithm in Figure 4.5 can now be performed to extract the latent features and learn the parameters simultaneously. Other than aforementioned three sets of transition parameters, the observation noise is simulated with $R = diag\{10, 10, 50, 50, 400\}$. As for the proposed VB inference algorithm, the hyperparameters are selected as $\{\alpha_a = 9, \beta_a = 1, \alpha_\rho = 1, \beta_\rho = 10^{-3}, \Sigma_b = 10^{-8}, \Sigma_h = 10^{-4}, \alpha_r = 0.1, \beta_r = 0.1\}$ for a non-informative prior, and the initial values are randomized as in

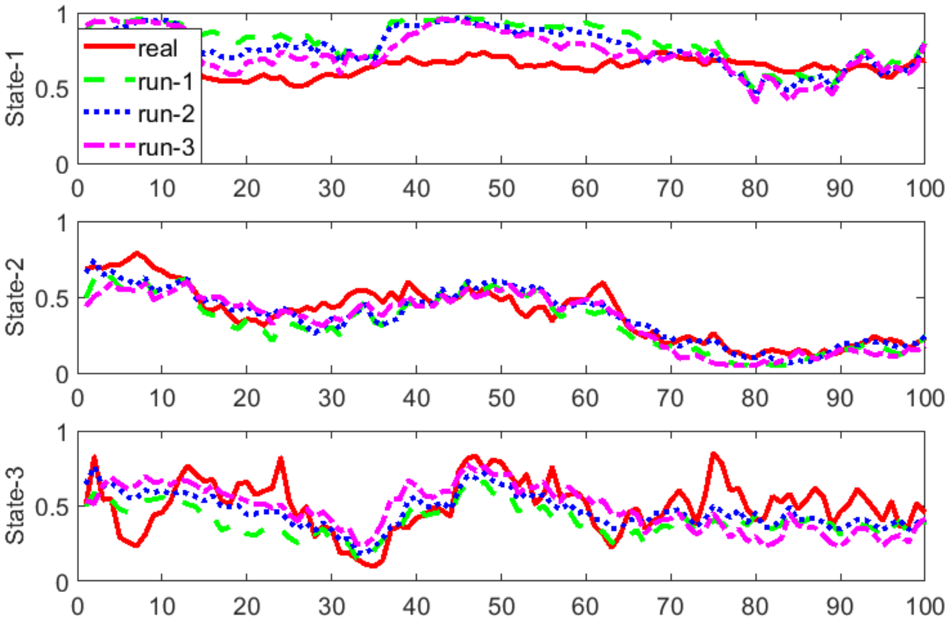


Figure 4.10: Estimation of Constrained Dynamic Latent Features without Model

Figure 4.5. Because of the inherit stochastic properties, the results of individual simulations could be different.

In Figure 4.10, the extracted dynamic latent features from three sampled individual runs are plotted. In this figure, the real features are simulated by the proposed transition model (4.12). "run-1", "run-2", and "run-3" are converged learning results with the aforementioned hyper-parameter set. It can be observed that the differences among individual realizations are not significant, indicating a numerically consistent learning procedure from fully random initial values. Specifically, the second latent feature, which possesses significant variations and has a certain level process inertia, has been well captured. Other than Figure 4.10, one additional visualization of extracted dynamic latent feature is shown in Figure 4.11. In this figure, the state estimation result, with the actual model parameter θ^* in simulating, is also plotted. These state estimation results have close accordance with real features. Similarly, the significant variations of "State-2" are captured with more accuracy. However, the third feature, which has the smallest auto-correlation coefficient a_3 and the smallest precision ρ_3, cannot be captured accurately for the given hyper-parameter set, especially for the part around $t = 160$. The main reason for this deviation is that the estimated $\hat{a}_3$ is not as small as the given one, where the fast fluctuations are not distinguished from observation noise. Another particular deviation is at the part around $t = 280$

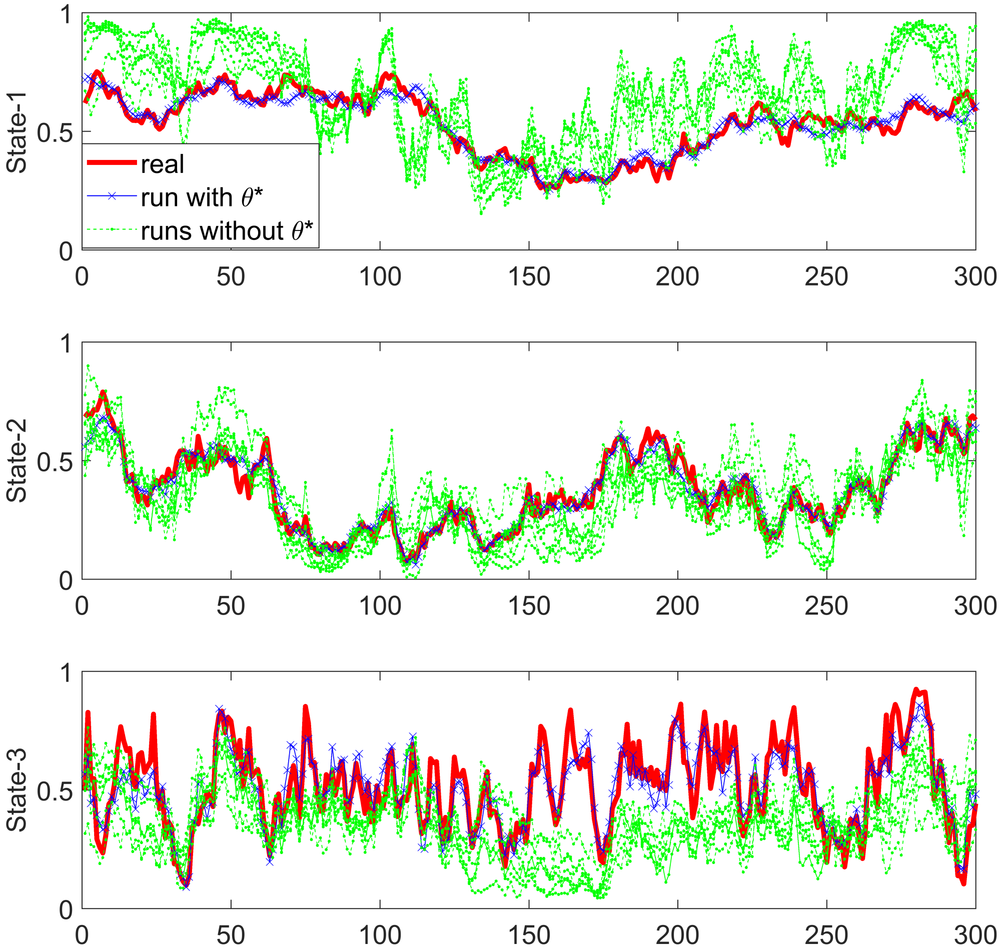


Figure 4.11: Estimation of Emission Matrix for Constrained Dynamic Feature

in the top plot. Roughly speaking, this peak shares similar shape as the concurrent part in “State-3”. It reveals the lacking of mutual independence in the latent space. In the following application section, this shortness will be overcome by iterative feature extraction procedure.

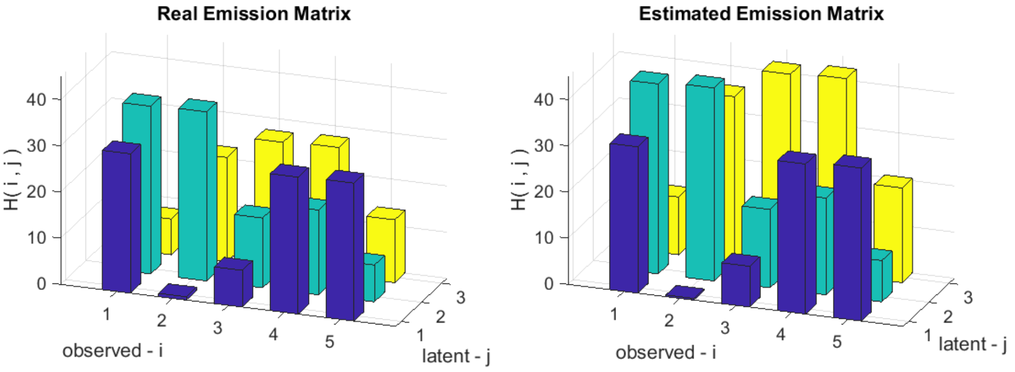


Figure 4.12: Example-1 Estimation of Emission Matrix for Constrained Dynamic Feature

In addition to the learned features, the estimated emission matrix is also compared with the real emission matrix H. For a better illustration, two examples are considered. The first one contains only positive entries, which has been plotted in Figure 4.12. In this plot, two axes denote the position in matrix H, and the vertical axis shows the value of this element. It can be observed that the bars with same colour share a similar shape. In other words, the relative sizes of elements in each column $H_{:,j}$ in H have been captured correctly. On the other hand, for different latent dimension, the strength of emission magnitude is not learned well. That is, while the yellow bar is lower than the green bar for the real case, the estimated yellow bar is higher than the green one.

The second one contains both positive coefficients and negative coefficients and has been plotted in Figure 4.13. We can observe that for the first two slowest features, the significant emission weights, such as $H_{2,1}$, $H_{4,1}$, and $H_{4,2}$, are identified correctly. Based on this, the proposed algorithm has shown abilities to extract constrained latent features and learn correct parameters for the features with significant inertia. However, due to the insufficient excitation of short observation sequences, some estimated parameters, typically for the features with small inertia, have distance from corresponding real ones. Besides, the estimated features still have certain correlations. In order to overcome such drawbacks, the proposed approach will be applied to extract the latent features iteratively.

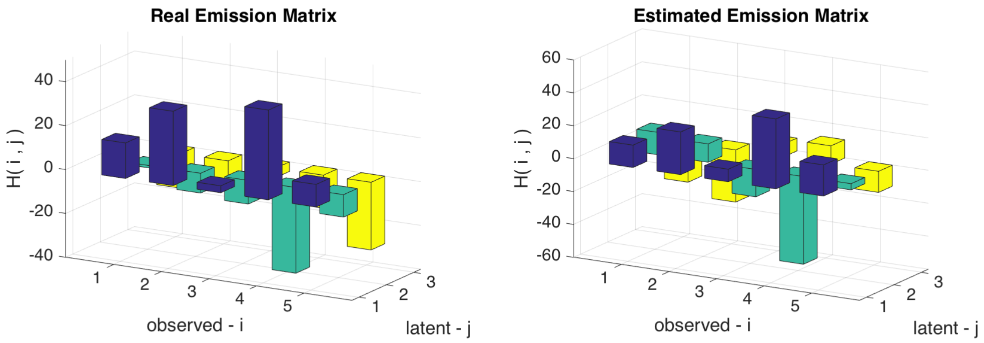


Figure 4.13: Example-2 Estimation of Emission Matrix for Constrained Dynamic Feature

4.5.2 Process Data Examples

In this section, the proposed feature extraction model is applied to process data. Since there are no real reference features to compare with (as in most practical cases), the applicability of the proposed feature extraction model will be verified through the regression tasks. As stated in the introduction section, the latent features are extracted for inferential sensor modelling and used as regressors to predict the quality variable. In this application, the constrained dynamic latent features are extracted iteratively to avoid possible mutual correlation. More specifically, one single dimensional feature will be extracted at one time; after convergence, the extracted component $\langle H_{:,j} \rangle \cdot \ln\langle s_{1:T}^{(j)} \rangle$ will be removed, leaving the 'residual' for the next feature extraction. Following the general feature extraction procedure, this iterative learning can be stopped according to either unsupervised criteria such as the level of estimation noise [23] or supervised criteria such as the correlation with training outputs [47]. To extract the latent features on testing data with the estimated model, the revised particle filter based on (4.32) and (4.33) is utilized.

The first validation is on the benchmark simulation of the Tennessee Eastman Process. In this problem, 33 variables are used as inputs to predict the concentration of chemical components [2]. Similar to the description in the introduction section, the training outputs are down-sampled by ten sampling units to mimic real inferential sensing application, where the outputs are typically sampled much more slowly than the inputs. Widely applied feature extraction based regression models, including principal components based regression (PCR), slow features based regression (SFR), and partial least squares (PLS), are used for comparison, while the multivariate linear regression (MLR) is selected as a baseline. The prediction performance on the testing data is evaluated by two statistics: (Pearson) Corre-

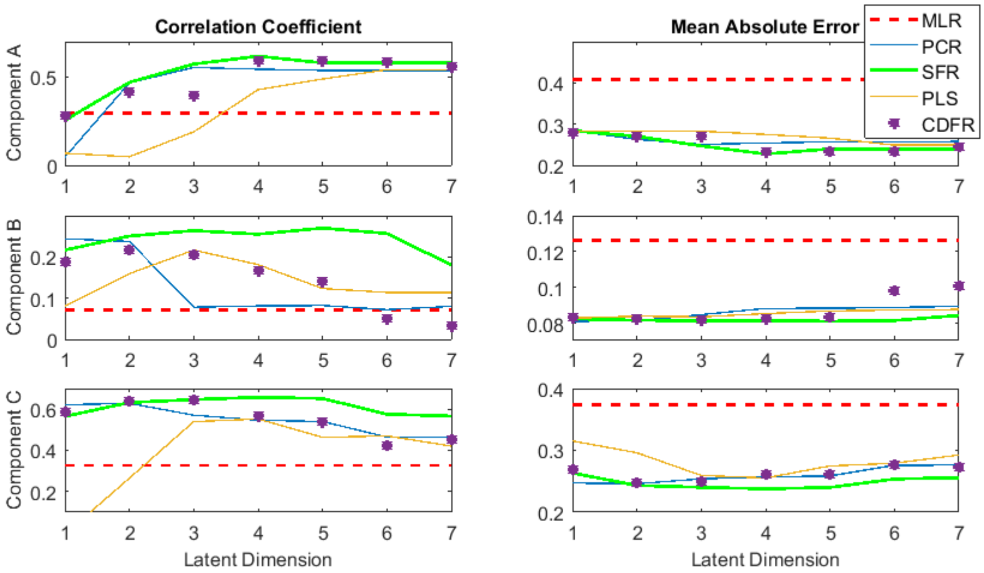


Figure 4.14: Prediction Performance on Tennessee Eastman Process Simulations

lation Coefficient for trend capture and the Mean Absolute Error for a robust measure of accuracy. The performance on Component-A is presented in Figure 4.14. The performance of the proposed method (CDFR) is comparable with the best one (SFR) of the considered references in both statistics. Based on this benchmark, applicabilities of the proposed feature extraction have been validated. To further demonstrate its advantages, another data set from a real-world process is used, where latent features have constrained variation ranges.

4.5.3 Industrial Case Study

This practical application is on Steam Assisted Gravity Drainage process, an enhanced oil recovery approach. The heated steam is injected into the underground reservoir to form a steam chamber, where the heavy oil/bitumen can be extracted with Electric Submersible Pumps. To configure a better control strategy, the quality of produced emulsion, such as the water content, needs a granular on-line estimation. Current measurements are obtained from a test separator, which is available to measure the water content every second week. Due to the complex and uncertain mechanism of the underground heating and drainage, the data-driven model is used, where primarily measured process variables, such as the flow rate, temperature, and pressure measurements are used as inputs to the data-driven model. Based on first-principles knowledge, 14 related variables are utilized as inputs. Based on process knowledge, the water content is affected by several underground variables, such as bitumen viscosity and “down-hole” pressures. Because their values are not the same for an extended heat chamber (around 1 kilometre), it was physically challenging to estimate them in a distributed manner. However, it is known that their variations must be contained in an operation range. In this application, the proposed constrained latent features are expected to reveal the variations of these unmeasurable variables within certain ranges. In order to illustrate the advantage/usage of the proposed model, an industrial case with potential constrained variations is selected as a regression exercise. The process is called Steam Assisted Gravity Drainage (SAGD) process, where the hot steam is injected to underground bitumen reservoir, and a mixture of oil and water (emulsion) is lifted through Electric Submersible Pumps. Real-time estimation of the emulsion water content is expected for process optimization, for which the current measurement is from an off-line separator.

Table 4.2: Validation Results of Estimating SAGD Water Content

d	Correlation Coefficient (Mean Absolute Error Improvement %)				
	PCA	SFA	PLS	sLDS	CDLF
2	0.19(9.4)	0.25(8.0)	-0.18(2.7)	0.11(7.2)	0.06(7.7)
3	0.30(9.9)	0.24(7.4)	-0.28(1.4)	0.14(7.4)	0.10(8.7)
4	0.09(8.5)	0.20(6.4)	-0.33(-3.7)	0.03(6.3)	0.26(9.8)
5	0.01(7.9)	0.11(2.7)	-0.25(-1.6)	0.13(2.2)	**0.32(11.8)**
6	-0.14(5.6)	-0.02(4.5)	-0.21(-4.0)	0.28(1.8)	0.30(8.0)

In Table 4.2, the Correlation Coefficient and the relative improvement of Mean Absolute Error (based on OLS) are compared over the retained latent dimension d (no further improvement). In Fig. 4.15, a trend visualization is plotted to compare the proposed model with the best competitor in Table 4.2. Besides the methods mentioned above, an additional competitor (sLDS) is realized, which also extracts dynamic latent features with a probabilistic model. It can be observed that the proposed method provides the highest correlation and the most significant improvement with five latent features. The widely applied PCA and SFA yield large deviations on testing data, which are exampled as peaks of the blue dash in Fig. 4.15. Although SFA produces a better correlation with small latent dimensions, its best performance is not competitive because of its only preference on “slowness”. As a supervised method, the less satisfactory performance of PLS is due to the sparsity and randomness of training outputs. As one of its probabilistic extensions, sLDS improves the performance in this application case, but it is still not as good as other unsupervised ones. Instead, CDLF has considered both probabilistic dynamic modelling and potential constraints.

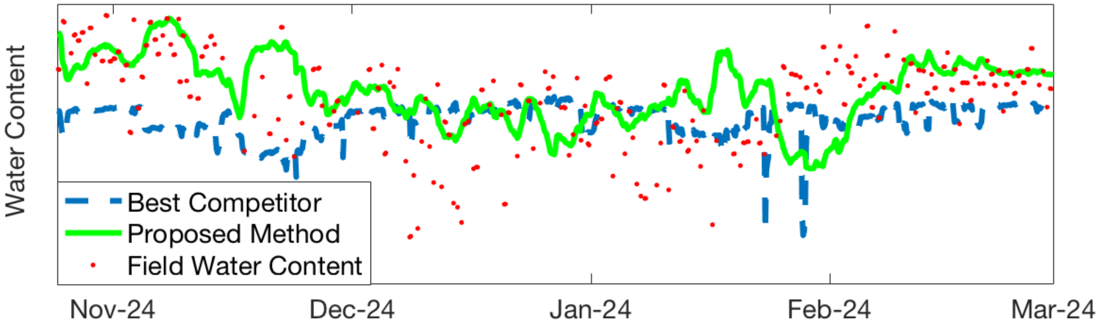


Figure 4.15: Estimation of Water Content in SAGD Process

4.6 Conclusions

In this chapter, a learning framework for extracting constrained dynamic features has been developed, where a novel latent transition function is designed. Based on typical characteristics of process data, the latent features are modelled to have the constrained range and the dynamic behaviour. By replacing the Normal distribution with the Beta distribution in the state transition model, the latent states are limited within $(0, 1)$. Through the comparison with the conventional unconstrained features, both stationary and transition properties of the proposed dynamic feature have been shown to be comparative to the unconstrained features.

To apply this constrained dynamic feature to latent variable modelling, an emission model is added, and the associated learning algorithm is developed through the variational Bayesian inference, where novel solutions are provided for the non-conjugate challenges. By testing it on numerical examples, the proposed combination of the variational Bayesian inference and particle state estimation methods is shown to be useful for parameter estimation. By testing it on benchmark examples and industrial data, the practical advantage of the proposed feature extraction approach has been demonstrated through the predictive modelling tasks.

Chapter 5

Dynamic Latent Feature for Multiple Model Problems *

In this chapter, the dynamic feature extraction is considered in the context of multi-mode observations. To address switching behaviours in industrial processes, multiple emission models are used to formulate the observation function. To address the temporal correlation from continuously operating processes, the transition function for continuous variables is implemented to describe the dynamic latent feature. Based on the variational Bayesian framework, a novel learning algorithm is developed to estimate the latent feature along with the multiple sets of emission parameters, where the estimation uncertainties are considered in variational updating steps. The effectiveness and practicability of the proposed model are illustrated through benchmark simulations and an industrial case study. In addition to the off-line modelling, the Bayesian state estimation algorithm is also extended to the online filtering application, where multiple models are considered in both transition function and observation function. Through the application of tracking problems, the proposed variational Bayesian filter has demonstrated its advantages.

5.1 Introduction

Sensors and transmitters are abundantly installed for process control and monitoring. With the development of data analysis technologies, historical data records become more valuable for monitoring and optimization. Typically, on-line estimation of quality variables can be improved by modelling process data and developing inferential sensors [4, 5, 6]. Similar

*A version of this chapter has been published in Ma, Yanjun, and Biao Huang. "Extracting dynamic features with switching models for process data analytics and application in soft sensing." AIChE Journal 64, no. 6 (2018): 2037-2051.
Section 5.6 of this chapter has been published in Ma, Yanjun, Shunyi Zhao, and Biao Huang. "Multiple-Model State Estimation Based on Variational Bayesian Inference." IEEE Transactions on Automatic Control (2018)

to the hardware instruments. However, knowledge-based soft sensors can frequently fail due to harsh operating conditions and unattainable assumptions. Data-driven approaches have proven to be a good alternative for inferential sensor development [7, 8, 9]. The main advantage of data-driven methods is their flexibility for various applications. Especially in the chemical process, with a variety of operating conditions, the same process can exhibit quite different behaviours. Rather than establishing first-principles-based models for various conditions, modelling with specific process data can learn the multiple operating conditions automatically and then provides more accuracy.

As a valuable package for industrial applications, inferential sensing requires the studies from both practical and theoretical point of view. The accuracy of its predictive model (estimating quality variables with regularly measured process variables) is essential. Although there exist plenty of modelling approaches for this regression task, many of them are facing challenges in dealing with process data. First, the excitation of model inputs, which are typically measured in a routine operating process, is usually insufficient and there is a common occurrence of co-linearity [64, 65]. Second, the reference outputs of inferential sensors are usually sparse and contain time-domain uncertainty. For an output as a quality variable, its reference value is often obtained from infrequent laboratory analysis, which has large sampling intervals and usually is lagged from the actual sample time. Such difficulties can make classical regression methods less practical. To overcome aforementioned challenges, latent variable models were proposed and have become popular in process data analytics [66, 67, 68]. In these approaches, the raw measurements are projected to construct informative latent features with linearly or non-linearly functions. The prediction is then performed with these features. Appropriate latent feature can improve the modelling effectiveness and robustness through a variety of methods [69, 70]. In particular, some improvements are achieved by considering the ubiquitous system inertia in chemical processes [71]. For example, the Markovian model can be added to capture temporal data correlations in the latent space [72, 73]. In Chapter 2, the general formulation of dynamic feature extraction has been presented.

In addition to the challenges above, various operating modes are often observed in historical data. In chemical engineering, operating modes usually vary with the changes in feed or product quality. From the modelling point of view, a single model is usually insufficient for multiple modes, and divide-and-conquer strategies are expected to make improvements. Under some scenarios, scheduling variable(s) can be selected to identify the switching behaviours [74, 75]. However, in many other cases, the identification of operating mode requires additional analysis. As for modelling an input-output system, the model identity of each data point is usually determined together with parameter estimation [76, 77, 78]. As for the latent feature extraction, several structures have also been proposed from different assumptions. If switching behaviours are assumed to follow a Markovian transition, Hidden Markov Models [79, 80] and other extended hierarchical structures [81, 82] can be used to estimate the model identity sequence. If observations were assumed as representation of specific latent variables, switching states space models [83, 84, 85] can be constructed. In this study of feature extraction for multi-mode process data, we proposed to use switched emission models and unique feature variables to capture switching and dynamic behaviour of the processes.

Other than from the modelling perspective, the multiple model scenario has also been widely studied in the state estimation perspective. During the last three decades, state estimation from various types of noisy measurements has drawn considerable attention. The most well-known method is inarguably the Kalman filter (KF) [86], which provides the optimal Bayesian estimates for the linear state-space model with white Gaussian noises. For the state estimation of multiple-model systems, it has been proved that the optimal Bayesian estimation is computationally intractable due to the exponentially increasing number of possible hypotheses (trajectories). Therefore, some suboptimal approaches have been developed [87, 88, 89], and the critical point is to limit the number of underlying hypotheses with an approximation of the respective probability density functions (PDFs).

For example, randomly selecting a predetermined number of hypotheses and discarding the remaining ones was proposed in [90], and a selection of the most likely hypothesis was developed in [91, 92]. Later, the generalized pseudo-Bayesian (GPB) algorithm that approximates the posterior Gaussian mixture distribution by a single Normal distribution after filtering was proposed in [83, 93]. Different from the GPB method, the interacting multiple model (IMM) [94] algorithm conducts Normal distribution approximation before filtering, which results in a good trade-off between accuracy and robustness. This method has been used in many practical applications including moving target tracking, fault detection, signal processing, etc., and one can also find various extensions and modifications [95, 96]. To name a few, the IMM strategy has been extended to the non-linear state-space model in [97] using the particle approximation, and the exponential cost function is minimized for multiple-model state-space models to get the risk-sensitive IMM filter, which is further extended to the non-linear case in [98]. The multiple-model estimation with variable structure is developed in [99] to deal with uncertain model parameters.

Moreover, an improvement of the IMM method using the expectation-maximization (EM) approach is developed in [100], with the aim of finding more accurate mode weights. A deterministic algorithm, as well as a stochastic algorithm, is given in [101] to obtain the marginal maximum of a sequential posterior state estimate of jump Markov linear systems. On-line Bayesian estimation of model transition probabilities is proposed and realized in the framework of multiple-model estimation, and the corresponding maximum likelihood modifications are further proposed in [102]. For more details, one can refer to [103] and references therein.

Probabilistic learning methods have been considered effective to accommodate various structures and improve performance in comparison with deterministic approaches [23]. For mixture models, clustering with probabilities allows obtaining the labels and model parameters more efficiently [104]. For latent variable models, properly assigned probability distributions help in dealing with noisy observations [22] and infrequent measurements [105], and increase the robustness against outliers [24]. The probabilistic inference and associated Bayesian framework are also preferred in dealing with many challenges in process data analytics [8, 26]. In these approaches, the background knowledge or process understanding can be implemented mathematically based on Bayes rule, and model parameters can be estimated with probability distributions rather than point estimations. As for modelling with latent features, Bayesian inference methods not only improve efficiency by considering parameters uncertainties but can also provide an explicit indicator of modelling performance [29], leading to an automatic selection of proper model structures.

Since the continuous latent feature utilized in this chapter is modelled similarly as in the general state space model, the proposed variational learning algorithm become a suitable candidate to solve the state estimation problem. It has revealed in the iterative updating procedure in Section 2.2. In order to compute the overall state estimates and error covariances in multiple-model estimation, a strategy that extracts the exact posterior PDF from all the possible trajectories is necessary. By analyzing methods mentioned above, it can be observed that approximating the posterior PDF (Gaussian mixture distribution in linear Gaussian systems) by a single Gaussian distribution with the mean and variances calculated through weighted summation is a common practice. Therefore, the proposed variational learning step for latent feature $S_{1:T}$ can be applied as a better method to obtain the overall estimates of the state and error covariance from a Gaussian mixture distribution, which motives this work.

5.2 Problem Statement

In this study, the feature extraction method is considered to have multiple emission models and autoregressive latent features. While those emission models describe the switching properties, a unique set of latent features is modelled to capture the continuous process dynamics. To learn this structure from process data, a Bayesian inference algorithm is developed to increase the robustness for parameter estimation. With the proposed learning framework, the structural parameters, such as the number of switched models and the dimension of latent space, can also be automatically determined in unsupervised learning.

Considering the switching scenarios, the variational learning method for dynamic latent features is novel. This variational Bayesian (VB) inference is used to approximate the joint PDF of the state and the model identity variable. Based on this strategy, the approximation accuracy in the tracking problem is expected to be improved.

The remainder of this chapter proceeds as follows. In the next section, the problem formulation and detailed explanations are given for the proposed structure of feature extraction. In Section 5.4, the specific learning method is developed for the (off-line) feature extraction model, where the state estimation is performed as a smoothing problem. Two numerical simulations are used here to validate the off-line feature extraction model. In Section 5.5, the corresponding on-line estimation algorithm is developed. The subsequent application demonstrates the proposed modelling approach through an industrial case study. Besides, the state estimation step is applied as a filtering algorithm in Section 5.6, which is then demonstrated through tracking problems. Finally, concluding remarks are presented in Section 5.7.

5.3 Probabilistic Formulation

In this section, multiple emission models are introduced to describe the changing operation regions, and the probability graphical model is proposed for our feature extraction task.

5.3.1 Multiple Emission Models

Switching of operating mode is commonly seen in process industries. Usually in the context of inferential sensing, these changing modes are results of unknown and random events, such as the changing of product price or the composition of feed materials. In this study, the proposed feature extraction model generalizes the emission function (2.7) with multiple emission models. While k-th model $M^{(k)} = \{\mu^{(k)}, H^{(k)}, v_{\cdot}^{(k)}\}$ is selected, the observation is described with

$$X_t = \mu^{(k)} + H^{(k)} \cdot S_t + v_t^{(k)}, \qquad if : r_t = k, \tag{5.1}$$

where $\mu^{(k)}$ is the mean of observations, $H^{(k)}$ is the projection matrix, and $v_t^{(k)}$ is observation noise. The model identity variable $r_t \in \{1, ..., K\}$ assigns one emission model to each observation.

As for the transition function, the model and the constrained described in Chapter 3 is retained.

$$S_t = A \cdot S_{t-1} + w_t, \quad w_t \sim \mathcal{N}(\vec{0}, Q), \tag{5.2}$$

$$where : A = diag\{a_1, ..., a_d\} \quad Q = diag\{1 - a_1^2, ..., 1 - a_d^2\}. \tag{5.3}$$

The transition matrix A and the noise covariance Q for dynamic latent features are fixed as diagonal and constrained Q in (5.3). It assures that the prior distribution of each dimension of latent features is mutual independent, and the standard Normal distribution is its stationary distribution. It should be noted that the constraint (5.3) is implemented for the continuous or relatively steady process, providing the benefit as a regulator. While it is necessary to assume significant transition, such as batch process, this constraint could be removed, and the switching structure is not limited to (5.3). Thus, an illustration of this extraction model can be presented in Figure 5.1: (1) $S_{1:T}$ is connected with the transition model to capture common driving forces among different modes; (2) multiple emission models $M^{(1:K)}$ map the latent feature to observations $X_{1:T}$ and reflect the mode changes.

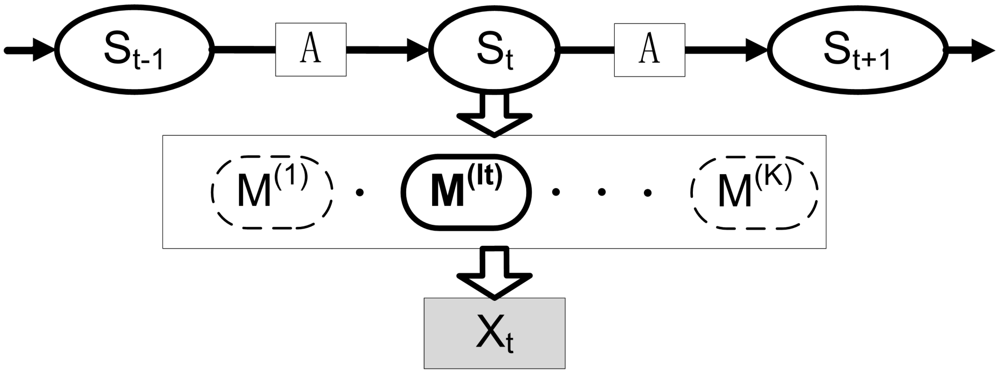


Figure 5.1: Multiple Emission Model for Dynamic Latent Feature

The feature extraction objective of this chapter is thus formed as that based on a sequence of observations $X_{1:T}$, identify the multiple emission models $M^{(1:K)}$ and the transition matrix A. The latent feature $S_{1:T}$ in this model is independent of the model identity sequence $r_{1:T}$, and follows a single transition matrix.

This structure is different from the common assumption of switching state space model or switched dynamical systems, which usually contains multiple state transition models and multiple emission models. Although the conventional switching structure is more general in representation, it could be difficult in inferential sensing practice. On the one hand, the historical records from an operating plant, especially for continuous operations, often contain insufficient excitations to learn over multiple model parameters. On the other hand, since both $H^{(k)}$ and the dimension of latent space are determined from data, the switched dynamics could also be captured by different latent sequences and presented through $H^{(k)}$. Thus, an appropriate model that is desired in practical applications should have a simpler structure and less number of parameters. In the following development, we will adopt a robust noise model to reduce the sensitivity to outliers, and use a Bayesian learning method to allow temporal correlations in the posterior distribution of the model identity sequence $r_{1:T}$.

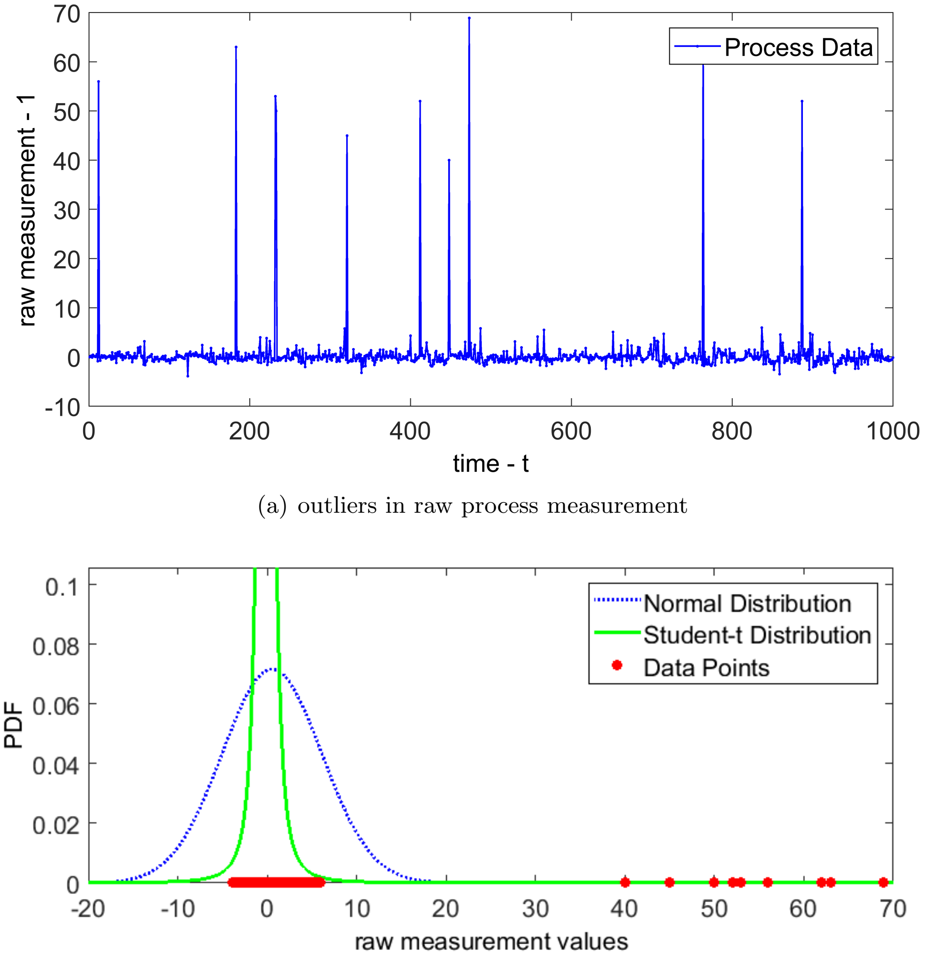
5.3.2 Robust Realization of Noise Model

While establishing a probabilistic model, the distribution of noise should be carefully selected [23]. As for feature extraction, the multi-dimensional observations are to be decorrelated by emission models, leaving the residual noises to follow independent Normal distributions such as the case in the probabilistic principal component analysis (PPCA) [22]. Depending on different applications, the magnitude of the variance of each output can be either the same or different [106]. In this study, $v_{\cdot}^{(k)}$ is modelled with a diagonal and isotropic covariance matrix $(\gamma^{(k)})^{-1} \cdot \mathbb{I}$ for each emission model.

Noise modelling should also be robust, especially for industrial process data. Here the robustness refers to less sensitive to the outlier and abnormal observations [107]. Student-t distribution has been considered as a proper probabilistic description of data with outliers [108, 109]. Figure 5.2 shows an illustrative example of the advantage of using Student-t distribution. In its top plot, one example of raw process measurement is shown, where several peaks can be observed. In the bottom plot, the two fitted distributions are compared, and one can see Student-t distribution can accommodate the outliers better than the Normal distribution.

One mathematical explanation of this robustness is based on the Normal-Gamma factorization for the distribution of $v_t^{(k)}$:

$$v_t^{(k)} \sim \mathcal{S}(\vec{0}, \frac{1}{\gamma^{(k)}}\mathbb{I}, \nu) = \int_0^{+\infty} \mathcal{N}(\vec{0}, \frac{1}{u\,\gamma^{(k)}}\mathbb{I}) \cdot \mathcal{G}(u;\ \frac{\nu}{2}, \frac{\nu}{2}) \cdot du, \tag{5.4}$$



(a) outliers in raw process measurement

(b) comparison between Normal and Student-t distribution

Figure 5.2: Handling Outliers with Student-t distribution

where ν is the degree of freedom and $\mathcal{G}$ stands for Gamma distribution. The intermediate variable u can be introduced to scale the variance of Normal distribution, and the Student-t distribution becomes an infinite summation of scaled Normal distributions. Hierarchically, u should distribute according to a Gamma distribution with parameter ν. In application, a certain value of u can be assigned for each data sample. Thus, the integration is simplified according to an application of the Mean Value Theorem [110]. It should be noted that even for multivariate Student-t distribution, this decomposition is always performed with one-dimensional u. This means that ν controls an isotropic fading rate of probability density, and the covariance matrix in the Normal distribution determines the direction. Similar as in Chapter 3, the covariance matrix is set as identity matrix $\mathbb{I}$, which makes the dimension reduction feasible.

As the result, one additional latent variable, defined as $U_t^{(k)}$ for each emission model $M^{(k)}$, is introduced to the proposed model:

$$p(X_t \mid \mu^{(k)}, H^{(k)}, S_t, \gamma^{(k)}, U_t^{(k)}) = \mathcal{N}(X_t;\ \mu^{(k)} + H^{(k)} \cdot S_t,\ \frac{1}{U_t^{(k)}\gamma^{(k)}}\mathbb{I}), \tag{5.5}$$

$$p(U_t^{(k)} \mid \nu) = \mathcal{G}(U_t^{(k)};\ \frac{\nu}{2}, \frac{\nu}{2}). \tag{5.6}$$

The effect of $U_t^{(k)}$ is to calibrate the likelihood of X_t for the k-th model, and the actual value of it will be learned from both data and the prior in (5.6). When a small value of $U_t^{(k)}$ is determined, the observation will be considered as an outlier, and its likelihood will be down-weighed. Since there are multiple emission models, different calibration levels will be associated with different $M^{(1:K)}$. Based on this calibrated likelihood (5.5), the number of false switching actions (caused by outliers) will be generally reduced, and the estimation of model parameters, such as $\mu^{(k)}$ and $H^{(k)}$, can be improved.

5.3.3 Use Identity Vector for Switching Mechanism

As is common in the statistical learning and probabilistic inference, the log-likelihood plays a crucial role in the derivation. To avoid inconvenience of deriving a logarithm before a summation operator, the discrete scalar r_t in (5.1) is re-defined by a vector variable I_t specified as

$$\begin{aligned} I_t &= \left[I_t^{(1)}, I_t^{(2)}, ..., I_t^{(K)} \right], \\ where: I_t^{(k)} &\in \{0, 1\}, \qquad \forall\ k = 1, ..., K, \\ \sum_{k=1}^{K} I_t^{(k)} &= 1. \end{aligned} \tag{5.7}$$

The activation of the k^{th} model is now translated as $I_t^{(k)} = 1$. With r_t governed by the categorical distribution, I_t can be described with the Multinomial distribution (Mul):

$$Mul\left(I_t \mid \pi_{1:K}\right) = \prod_{k=1}^{K} \left[\pi_k\right]^{I_t^{(k)}}, \tag{5.8}$$

where the parameter $\pi_{1:K}$ is a sum-one vector with all elements being positive. Each π_k stands for the statistical expectation of $I_t^{(k)}$, which also represents the responsible weight of the k^{th} model. Thus, it has the following properties:

$$\Pr\left(r_t = k\right) = \mathbb{E}_{Mul(I_t|\pi_{1:K})}[I_t^{(k)}] = \Pr\left(I_t^{(k)} = 1\right) = \pi_k \geq 0, \tag{5.9}$$

$$\sum_{k=1}^{K} \pi_k = \sum_{k=1}^{K} \Pr\left(r_t = k\right) = 1. \tag{5.10}$$

This vectorial representation of the model identity can provide a compact probability description of $\{S_t, I_t\}$ in the log-likelihood formulation, which facilitates the approximation of posterior considerably. For example, the conditional probability of the observation function becomes

$$p(X_t \mid S_t, I_t, M) = \prod_{k=1}^{K} \left[p(X_t \mid S_t, I_t, M^{(k)})\right]^{I_t^{(k)}}. \tag{5.11}$$

With the product operator, I_t serves as the exponential terms for S_t. Considering that I_t described by equation (5.8) belongs to a conjugate prior distribution of the likelihood, the posterior of I_t will have a closed form and can be derived conveniently [111]. Besides, by using its statistical expectation, I_t will be free from discrete values in the inference procedure. Thus, a more granular interaction between the model identity and S_t is introduced by this vectorial representation I_t, which merges multiple models without waiting until the final calculation step.

5.3.4 Prior Distributions

To complete this feature extraction in probabilistic approaches and formulate it under the Bayesian framework, other probability dependencies and modelling preferences should also be clarified. The first one is about reducing the risk of over-fitting from the increasing number of sub-models. To avoid generating unnecessary sub-models, the following prior distributions are assigned to parameters $\mu^{(k)}$, $H^{(k)}$ and $\gamma^{(k)}$ as regulators:

$$p(\mu^{(k)} \mid \Lambda_\mu) = \mathcal{N}(\mu^{(k)};\ \vec{0}, \Lambda_\mu^{-1}), \tag{5.12}$$

$$p(H^{(k)} \mid \Lambda_H) = p(\left[h_1^{(k)}, ..., h_m^{(k)}\right]' \mid \Lambda_H^{-1}) = \prod_{j=1}^{m} \mathcal{N}(h_j^{(k)} \mid 0, \Lambda_H^{-1}), \tag{5.13}$$

$$p(\gamma^{(k)} \mid \alpha_\gamma, \beta_\gamma) = \mathcal{G}(\gamma^{(k)};\ \alpha_\gamma, \beta_\gamma), \tag{5.14}$$

where m is the dimension of X_t, and $h_i^{(k)}$ is the transpose of the i-th row in $H^{(k)}$. Hyperparameters Λ_μ, Λ_H, $\alpha_\gamma$and β_γ are used to represent a non-informative prior distribution, which are only preventing the weird values. With these regulatory prior distributions, the parameters of an insignificant emission model will become trivial, such as discussed in Chapter 3.

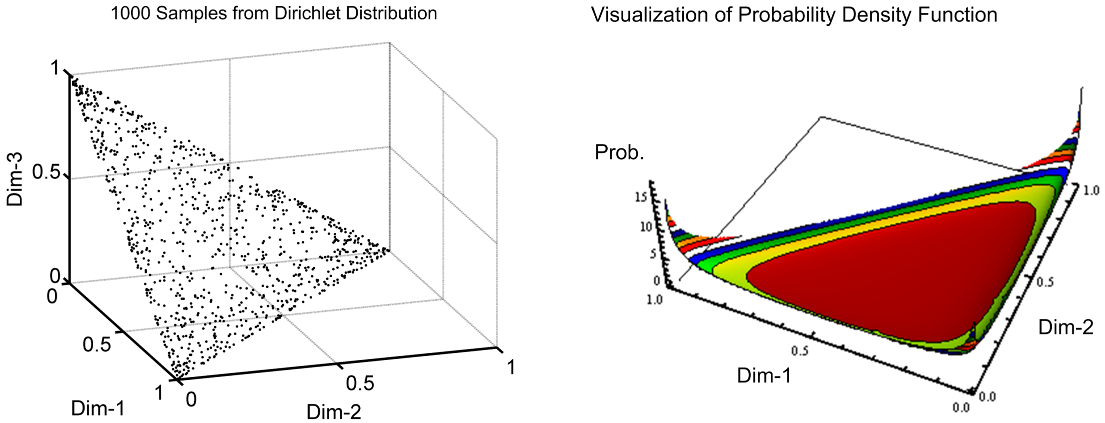


Figure 5.3: Selecting model identity from Dirichlet distribution

To set the preference for a smaller number of sub-models, $\pi_{1:K}$ is described by a prior of Dirichlet distribution Dir to allow Automatic Relevance Determination (ARD):

$$p(\pi_{1:K} \mid \alpha_{1:K}) = Dir(\pi_{1:K}; \alpha_{1:K}), \quad \forall\ k = 1, ..., K,\ \alpha_k < 1. \tag{5.15}$$

With the fixed K and $\alpha_k < 1$, the sample with $I_t^{(k)}$ near either zero or one will receive high value from this probability distribution. As a three-dimensional ($K = 3$) example, Figure 5.3 shows the samples and a collapsed density function of the case with $\alpha_{1:3} = [0.4, 0.4, 0.4]$. Thus, the number of "useful", assigned with a significant value for the weight, models is preferred to be smaller, which means ARD can be performed through Bayesian inference with this prior distribution.

To summarize all the probabilistic dependencies, a probability graphical model is drawn in Figure 5.4. The probabilistic description for the dynamic latent feature $S_{1:T}$ ($S_t \in \mathbb{R}^d$) and the transition matrix A ($= diag\{a_{1:d_S}\}$) is adopted as in Chapter 3:

$$p(S_{1:T} \mid A) = P(S_0) \cdot \prod_{t=1}^{T} P(S_t \mid S_{t-1}, A)$$

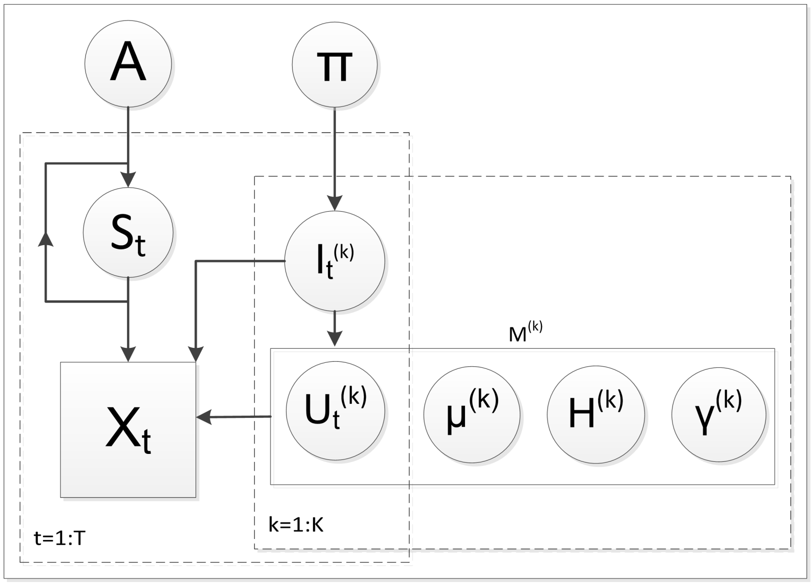


Figure 5.4: Proposed Bayesian design for Multi-Model Dynamic Feature

$$= \mathcal{N}(S_0;\ \vec{0}, \mathbb{I}_d) \cdot \prod_{t=1}^{T} \mathcal{N}(S_t;\ A \cdot S_{t-1}, Q), \tag{5.16}$$

$$p(A \mid \alpha_a, \beta_a) = \prod_{j=1}^{d} P(a_j) = \prod_{j=1}^{d} Beta(a_j;\ \alpha_a, \beta_a), \tag{5.17}$$

where Q is defined in (5.3). The robust noise modelling (presented in earlier section) with multi-model results in the relation between $U_t^{(1:K)}$ and $I_t^{(1:K)}$ as

$$p(U_t^{(1:K)} \mid I_t^{(1:K)}, \nu) = \prod_{k=1}^{K} \left[\mathcal{G}(U_t^{(k)};\ \frac{\nu}{2}, \frac{\nu}{2}) \right]^{I_t^{(k)}}, \tag{5.18}$$

which is detailed in [75]. Then, the likelihood of (finite length) observations conditioned on $S_{1:T}$ and $M^{(1:K)}$ is given by

$$\begin{aligned} & p(X_{1:T} \mid S_{1:T}, I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)}, M^{(1:K)}) \\ = \ & p(X_{1:T} \mid S_{1:T}, I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)}, \{\mu, H, \gamma\}^{(1:K)}) \\ = \ & \prod_{t=1}^{T} \prod_{k=1}^{K} \left[\mathcal{N}(X_t;\ \mu^{(k)} + H^{(k)} \cdot S_t, \frac{1}{U_t^{(k)} \gamma^{(k)}} \mathbb{I}_m) \right]^{I_t^{(k)}}. \end{aligned} \tag{5.19}$$

Finally, the objective of feature extraction is transformed as to inference the following posterior:

$$p(\theta \mid X_{1:T}, \eta) = p(S_{1:T}, I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)}, \mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)}, A, \pi_{1:K} \mid X_{1:T}, \eta). \tag{5.20}$$

where θ is a short notation for the whole parameter set. It should be noted that, usually the outlier in process measurements are not believed to be sufficient enough to learn the degree of freedom in the Student-t distribution. Thus, in this study, the parameter ν will not be learned from data; instead, it will be included into the set of hyper-parameters:

$$\eta = \{\Lambda_\mu, \Lambda_H, \alpha_\gamma, \beta_\gamma, \alpha_a, \beta_a, \alpha_{1:K}\}. \tag{5.21}$$

5.4 Off-line Variational Inference Methods

In the learning procedure, the proposal distribution can be updated through coordinate ascent strategies or sampling methods [112]. For our proposed model, the coordinate ascent approach, as introduced in Chapter 2, is chosen to avoid unnecessary computational load and retain a meaningful interpretation, where $q(\theta)$ is defined with assumption:

$$q(\theta) = q(S_{1:T}) \cdot q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)}) \cdot q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)}) \cdot q(A) \cdot q(\pi_{1:K}). \tag{5.22}$$

Following the general derivation in Section 2.2, the objective of variation learning algorithm can be formulated as

$$\begin{aligned}\mathcal{L}(q(\theta)) &= p(X_{1:T} \mid \eta) - D_{KL}\{q(\theta) \parallel p(\theta \mid X_{1:T}, \eta)\} \\ &= \mathbb{E}_{q(\theta)} p(X_{1:T}, \theta \mid \eta) + \mathcal{H}\{q(\theta)\}. \end{aligned} \tag{5.23}$$

Then the coordinate ascent algorithm will maximize variational lower bound $\mathcal{L}(q(\theta))$ based on the following three equations:

$$\ln q^*(A) = \mathbb{E}_{q(S_{1:T})} [\ln p(X_{1:T}, \theta \mid \eta)] + c_A, \tag{5.24}$$

$$\ln q^*(S_{1:T}) = \mathbb{E}_{q(A)q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})} [\ln p(X_{1:T}, \theta \mid \eta)] + c_S, \tag{5.25}$$

$$\ln q^*(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)}) = \mathbb{E}_{q(S_{1:T})q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})} [\ln p(X_{1:T}, \theta \mid \eta)] + c_M, \tag{5.26}$$

$$\ln q^*(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)}) = \mathbb{E}_{q(S_{1:T})q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})q(\pi_{1:K})} [\ln p(X_{1:T}, \theta \mid \eta)] + c_{IU}, \tag{5.27}$$

$$\ln q^*(\pi_{1:K}) = \mathbb{E}_{q^*(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})} [\ln p(X_{1:T}, \theta \mid \eta)] + c_\Pi, \tag{5.28}$$

where c_A, c_S, c_{IU}, c_M, c_Π are normalization constants. By iteratively applying these updating equations, the variational Lower Bound will increase monotonically. It can be observed that the dynamics of latent feature $S_{1:T}$ is modelled similar as in Chapter 3, the updating procedure of (5.24) can be solved similarly. With the usage of importance sampling, the required statistics in this chapter become available from equations in (3.40), (3.41), and (3.42). As for the updating step (5.45), it depends on the most of proposal distributions, also its result will be used for nearly every other updating steps. In later this section, the innovative steps will be detailed for it. With aforementioned probability dependencies, the updating equations from (5.26) to (5.28) can be derived based on conjugate prior distributions.

5.4.1 Learning Multiple Robust Emission Models

In this part, the updating equation for the parameters in the multiple emission models will be detailed. Following the detailed equations for (5.26), the updating procedure of (5.27) and (5.28) will be provided. Since the feature extraction model is developed in an off-line manner, and the model identity variables are assumed as temporal independent, the proposal distribution $q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})$ and the proposal distribution of the latent features $q(S_{1:T})$ can be learned individually.

Here, an iterative updating procedure for multiple emission models is presented.

1. For $\mu^{(k)}$, the updated proposal distribution is still a Normal distribution $q(\mu^{(k)}) = \mathcal{N}(\mu^{(k)};\ m_\mu^{(k)}, P_\mu^{(k)})$:

$$m_\mu^{(k)} = P_\mu^{(k)} \cdot \langle\gamma^{(k)}\rangle \sum_{t=1}^{T} \langle U_t^{(k)} I_t^{(k)}\rangle (X_t - \langle H^{(k)}\rangle\langle S_t\rangle), \tag{5.29}$$

$$P_\mu^{(k)} = \left[\langle\gamma^{(k)}\rangle \sum_{t=1}^{T} \langle U_t^{(k)} I_t^{(k)}\rangle \cdot \mathbb{I}_m + \Lambda_\mu\right]^{-1}. \tag{5.30}$$

2. For $H^{(k)}$, the transpose of i-th row $h_i^{(k)}$ is updated with a Normal distribution $q(h_i^{(k)}) = \mathcal{N}(h_i^{(k)};\ m_{h_i}^{(k)}, P_{h_i}^{(k)})$:

$$m_{h_i}^{(k)} = P_{h_i}^{(k)} \cdot \langle\gamma^{(k)}\rangle \sum_{t=1}^{T} \langle U_t^{(k)} I_t^{(k)}\rangle \left[X_t - \langle\mu^{(k)}\rangle\right]_{(i)} \langle S_t\rangle, \tag{5.31}$$

$$P_{h_i}^{(k)} = \left[\langle\gamma^{(k)}\rangle \sum_{t=1}^{T} \langle U_t^{(k)} I_t^{(k)}\rangle \langle S_t S_t'\rangle + \Lambda_H\right]^{-1}, \tag{5.32}$$

where $[\cdot]_{(i)}$ denotes the i^{th} elements of the inside vector. Thus, the expectation statistics for the overall emission matrix are formed as

$$\langle H^{(k)}\rangle = \left[m_{h_1}^{(k)}, ..., m_{h_m}^{(k)}\right]', \tag{5.33}$$

$$\langle H^{(k)\prime} \cdot H^{(k)}\rangle = \sum_{j=1}^{m} m_{h_j}^{(k)} \cdot (m_{h_j}^{(k)})' + P_{h_i}^{(k)}. \tag{5.34}$$

3. The proposal distribution for $\gamma^{(k)}$ is updated as a Gamma distribution with two parameters $q(\gamma^{(k)}) = \mathcal{G}(\gamma^{(k)};\ \alpha_{\gamma}^{(k)}, \beta_{\gamma}^{(k)})$:

$$\alpha_{\gamma}^{(k)} = \alpha_{\gamma} + \frac{m}{2} \sum_{t=1}^{T} \langle I_t^{(k)} \rangle, \tag{5.35}$$

$$\beta_{\gamma}^{(k)} = \beta_{\gamma} + \sum_{t=1}^{T} \frac{\langle U_t^{(k)} I_t^{(k)} \rangle}{2} \langle \sigma_t^2 \rangle. \tag{5.36}$$

The residual term $\langle \sigma_t^2 \rangle$ is an expected observation noise, which has the form as

$$\begin{aligned}\langle \sigma_t^2 \rangle =& X_t' X_t + tr\{\langle \mu^{(k)} (\mu^{(k)})' \rangle\} + tr\{\langle S_t S_t' \rangle \langle (H^{(k)})' H^{(k)} \rangle\} \\ & - 2X_t' \langle H^{(k)} \rangle \langle S_t \rangle - 2X_t' \langle \mu^{(k)} \rangle + 2 \langle \mu^{(k)} \rangle' \langle H^{(k)} \rangle \langle S_t \rangle,\end{aligned} \tag{5.37}$$

where $tr\{\cdot\}$ stands for the matrix trace operator.

Regarding the model identity variable $I_{1:T}$ and the intermediate variable $U_{1:T}$, their proposal distribution $q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})$ will not be further factorized. It is based the original Student-t distribution of each mission noise, which requires the dependency between these two sets of random variables[109]. For the conditional uncertainty $U_t^{(k)} \mid I_t^{(1:K)}$, the posterior is developed with a Gamma distribution based on given $I_t^{(1:K)}$:

$$q(U_t^{(k)} \mid I_t^{(1:K)}) = \mathcal{G}(U_t^{(k)} \mid I_t^{(1:K)};\ \alpha_{U_t}^{(k)}, \beta_{U_t}^{(k)}).$$

For $I_t^{(k)}$, it follows a Multinomial distribution:

$$q(I_t^{(1:K)}) = Mul(I_t^{(1:K)};\ \alpha_{I_t}^{(1:K)}).$$

The corresponding likelihood term has marginalized the intermediate variable $U_t^{(1:K)}$ out to making the actual Student-t distribution for observation noise:

$$\alpha_{U_t}^{(k)} = \frac{\nu}{2} + \frac{m}{2}, \tag{5.38}$$

$$\beta_{U_t}^{(k)} = \frac{\nu}{2} + \frac{\langle \gamma^{(k)} \rangle}{2} \langle \sigma_t^2 \rangle, \tag{5.39}$$

$$\alpha_{I_t}^{(k)} = e^{\langle \ln \pi_k \rangle} \cdot \frac{\Gamma(\frac{\nu+m}{2})}{\Gamma(\frac{\nu}{2})(\nu\pi)^{\frac{m}{2}}} \cdot \langle \gamma^{(k)} \rangle^{\frac{m}{2}} \cdot [1 + \frac{\langle \gamma^{(k)} \rangle}{\nu} \langle \sigma_t^2 \rangle]^{-\frac{\nu+m}{2}}, \tag{5.40}$$

where $\langle \sigma_t^2 \rangle$ is the expected residual that defined in (5.37). As for this joint proposal distribution, the required statistics from other variational updating steps can be determined as

$$\langle I_t^{(k)} \rangle = \frac{\alpha_{I_t}^{(k)}}{\sum_{k'=1}^{K} \alpha_{I_t}^{(k')}}, \tag{5.41}$$

$$\langle U_t^{(k)}\ I_t^{(k)} \rangle = \frac{\alpha_{U_t}^{(k)}}{\beta_{U_t}^{(k)}} \cdot \langle I_t^{(k)} \rangle, \tag{5.42}$$

$$\langle \ln U_t^{(k)}\ I_t^{(k)} \rangle = \Big[\psi\{\alpha_{U_t}^{(k)}\} - \ln\{\beta_{U_t}^{(k)}\}\Big] \cdot \langle I_t^{(k)} \rangle. \tag{5.43}$$

In order to update the distribution for $\pi_{1:K}$, aforementioned parameters of $\alpha_{I_{1:T}}^{(1:K)}$ can be treated as samples from $q(\pi_{1:K})$. Based on the statistical conjugacy, a Dirichlet distribution is selected: $q(\pi_{1:K}) = Dir(\pi_{1:K};\ \alpha_\pi^{(1:K)})$, and the parameters are updated as

$$\alpha_\pi^{(k)} = \alpha_k + \sum_{t=1}^{T} \langle I_t^{(k)} \rangle, \quad \forall\ k \in \{1, ..., K\}. \tag{5.44}$$

Other than the introduced statistics for upcoming variation learning step, other conventional self-expected statistics, such as the first order or second order moments, can be easily found with the estimated variable posteriors.

5.4.2 Smoothing States with Multiple Emission Models

The updating equation for $q^*(S_{1:T})$ is considered as

$$\ln q^*(S_{1:T}) = \mathbb{E}_{q(A)q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})} \left[\ln p(X_{1:T}, \theta \mid \eta)\right] + c_S, \tag{5.45}$$

which requires a study about the probability distribution over a sequenced latent variables $S_{1:T}$. Typically, calculating the sufficient statistics and entropy for this latent sequence is solved by filtering and smoothing [40]. By considering the distribution of model parameters, updating $q(S_{1:T})$ has been developed for a single emission model in Chapter 3. By considering multiple models with deterministic parameters, the problem can be solved by interacting multiple model methods [103]. However, with multiple emission models along with their uncertainties (described with probability distributions), this estimation problem becomes more challenging. In this study, a novel algorithm is developed to merge different models based on the distribution of model parameters $q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})$, as well as the uncertainty of model identity sequence $q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})$.

For a better illustration, (5.45) needs to be explicitly expanded as

$$\begin{aligned}
\ln q^*(S_{1:T}) &= \mathbb{E}_{q(I_{1:T}^{(1:K)}, U_{1:T}^{(1:K)})q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})} \left[\ln p(X_{1:T} \mid S_{1:T}, \theta)\right] \\
&\qquad + \mathbb{E}_{q(A)} \left[\ln P(S_{1:T} \mid A)\right] + const., \\
= &\sum_{t=1}^{T} \langle \ln P(X_t \mid S_t, I_t^{(1:K)}, U_t^{(1:K)}, \mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)}) \rangle + \sum_{t=1}^{T} \langle \ln P(S_t \mid S_{t-1}, A) \rangle + const. \\
= &\sum_{t=1}^{T} -\frac{1}{2} S_t' \sum_{k=1}^{K} \langle U_t^{(k)} I_t^{(k)} \rangle \langle \gamma^{(k)} (H^{(k)})' H^{(k)} \rangle S_t + S_t' \sum_{k=1}^{K} \langle H^{(k)} \rangle' \langle U_t^{(k)} I_t^{(k)} \rangle \langle \gamma^{(k)} (X_t - \mu^{(k)}) \rangle
\end{aligned}$$

$$+\sum_{t=1}^{T} -\frac{1}{2}S_t'\langle\frac{1}{\mathbb{I}-A^2}\rangle S_t + S_{t-1}'\langle\frac{A}{\mathbb{I}-A^2}\rangle S_t - \frac{1}{2}S_{t-1}'\langle\frac{A^2}{\mathbb{I}-A^2}\rangle S_{t-1} + const., \tag{5.46}$$

where $\langle\cdot\rangle$ is the expectation operator. The covariance of transition noise Q is replaced with $\mathbb{I}-A^2$, where the square operator is applied to each element of A. Based on this quadratic function of each individual latent sample S_t and adjacent latent samples (S_{t-1}, S_t), it can be concluded that their posteriors have Normal distributions. For deriving the Normal distribution over the latent variable sequence, the original Kalman filter and smoother have proven to be optimal. To benefit from its optimality, a parameter calibration procedure is developed to factorize (5.46) into traditional posterior formula of $S_{1:T}$ [40].

The parameters for transition function (5.2) are calibrated similar as in (3.33) and (3.34):

$$\tilde{A} = \tilde{A}' = \langle\frac{A}{\mathbb{I}-A^2}\rangle\langle\frac{1}{\mathbb{I}-A^2}\rangle^{-1} = diag\{\langle\frac{a_1}{1-a_1^2}\rangle/\langle\frac{1}{1-a_1^2}\rangle, ..., \frac{a_d}{1-a_d^2}\rangle/\langle\frac{1}{1-a_d^2}\rangle\}, \tag{5.47}$$

$$\tilde{Q}^{-1} = \langle\frac{1}{\mathbb{I}-A^2}\rangle = diag\{\langle\frac{1}{1-a_1^2}\rangle, ..., \langle\frac{1}{1-a_d^2}\rangle\}. \tag{5.48}$$

Different from the Chapter 3, the observations and the emission parameters are not augmented directly because of the multiple model cases. On the hand, since the latent features are modelled with a single set of $S_{1:T}$, and it has posterior distribution as sequenced Normal distributions, there always exist two coefficients to summarize the observation function: one for first order moment, and another for the second moment. In this study, for the first summation in (5.46), these two coefficients are calibrated as

$$\tilde{F}_t^{(P)} = \sum_{k=1}^{K}\langle U_t^{(k)} I_t^{(k)}\rangle\langle\gamma^{(k)}(H^{(k)})'H^{(k)}\rangle + \langle\frac{A^2}{1-A^2}\rangle - \tilde{A}'\tilde{Q}^{-1}\tilde{A}, \quad \forall\ t < T, \tag{5.49}$$

$$\tilde{F}_T^{(P)} = \sum_{k=1}^{K}\langle U_T^{(k)} I_T^{(k)}\rangle\langle\gamma^{(k)}(H^{(k)})'H^{(k)}\rangle, \tag{5.50}$$

$$\tilde{F}_t^{(m)} = \sum_{k=1}^{K}\langle H^{(k)}\rangle'\langle U_t^{(k)} I_t^{(k)}\rangle\langle\gamma^{(k)}(X_t - \mu^{(k)})\rangle, \qquad \forall\ t = 1...T. \tag{5.51}$$

With these intermediate parameters, (5.45) can be rewritten as

$$\ln q^*(S_{1:T}) = \sum_{t=1}^{T} -\frac{1}{2}S_t'\cdot\tilde{F}_t^{(P)}\cdot S_t + S_t'\cdot\tilde{F}_t^{(m)}$$

$$+\sum_{t=1}^{T} -\frac{1}{2}\left[S_t - \tilde{A}\cdot S_{k-1}\right]'\tilde{Q}^{-1}\left[S_t - \tilde{A}\cdot S_{k-1}\right] + const.. \tag{5.52}$$

This formulation has factorized adjacent samples, allowing recursive filtering and smoothing steps to obtain the posterior of $S_{1:T}$. Before introducing the updating steps, the definition of forward and backward paths is given as

$$P(S_t \mid \tilde{A}, \tilde{Q}, \tilde{F}^{(P)}_{1:t-1}, \tilde{F}^{(m)}_{1:t-1}) = \mathcal{N}(S_t;\ m_t^{(p)}, P_t^{(p)}), \tag{5.53}$$

$$P(S_t \mid \tilde{A}, \tilde{Q}, \tilde{F}^{(P)}_{1:t}, \tilde{F}^{(m)}_{1:t}) = \mathcal{N}(S_t;\ m_t^{(f)}, P_t^{(f)}), \tag{5.54}$$

$$P(S_t \mid \tilde{A}, \tilde{Q}, \tilde{F}^{(P)}_{1:T}, \tilde{F}^{(m)}_{1:T}) = \mathcal{N}(S_t;\ m_t^{(s)}, P_t^{(s)}). \tag{5.55}$$

The proposed estimation procedure is then presented with the following three steps.

1. Prediction step (inherited from Kalman filter):

$$m_t^{(p)} = \tilde{A} \cdot m_{t-1}^{(f)}, \tag{5.56}$$

$$P_t^{(p)} = \tilde{A} \cdot P_{t-1}^{(f)} \cdot \tilde{A}' + \tilde{Q}. \tag{5.57}$$

2. Update step (revised updating step):

$$m_t^{(f)} = P_t^{(p)} \cdot \left[\tilde{F}_t^{(m)} - C_t \cdot \tilde{F}_t^{(m)} \right] + L_t \cdot m_t^{(p)}, \tag{5.58}$$

$$P_t^{(f)} = L_t \cdot P_t^{(p)}, \tag{5.59}$$

$$C_t = \tilde{F}_t^{(P)} \cdot \left[(P_t^{(p)})^{-1} + \tilde{F}_t^{(P)} \right]^{-1}, \tag{5.60}$$

$$L_t = \mathbb{I}_d - P_t^{(p)} \cdot \left[\tilde{F}_t^{(P)} - C_t \cdot \tilde{F}_t^{(P)} \right], \tag{5.61}$$

where $\mathbb{I}_d$ is the identity matrix in d-dimensional space.

3. Smoothing step (inherited from Kalman smoother):

$$m_t^{(s)} = m_t^{(f)} + G_t * \left[m_{t+1}^{(s)} - m_{t+1}^{(p)} \right], \tag{5.62}$$

$$P_t^{(s)} = P_t^{(f)} + G_t * \left[P_{t+1}^{(s)} - P_{t+1}^{(p)} \right] * G_t', \tag{5.63}$$

$$G_t = P_t^{(f)} \cdot \tilde{A}' \cdot \left[P_{t+1}^{(p)} \right]^{-1}. \tag{5.64}$$

This proposed estimation procedure maintains the optimality of Kalman filter and smoother within this variational updating step. The proof can be verified by integrating Normal distributions (5.53)-(5.55) and applying the Woodbury matrix identity [113]. In fact, this updating procedure handled the uncertainty in the transition model (5.2) with calibrated parameters $\tilde{A}$ and $\tilde{Q}$, and simultaneously considered the probability of different emission models with expectation terms $\tilde{F}_t^{(P)}$ and $\tilde{F}_t^{(m)}$. As a result, the switching uncertainty is addressed within the standard forward-backward algorithm. To describe $q^*(S_{1:T})$ in other variational updating steps, the following sufficient statistics are prepared as

$$\langle S_t \rangle = m_t^{(s)}, \tag{5.65}$$

$$\langle S_t \cdot S_t' \rangle = P_t^{(s)} + \langle S_t \rangle \cdot \langle S_t \rangle', \tag{5.66}$$

$$\langle S_{t+1} \cdot S_t' \rangle = P_{t+1}^{(s)} \cdot G_t' + \langle S_{t+1} \rangle \cdot \langle S_t \rangle'. \tag{5.67}$$

5.4.3 Selecting Model Structures with Variational Lower Bound

In this probability model, hyper-parameters in (5.12)-(5.14) and (5.15)-(5.18) are fixed as constant to represent the modelling preference. As discussed before, by assigning zero-mean prior for $\mu^{(k)}$ and $H^{(k)}$, only "necessary" latent feature dimensions remain active in the resulting posteriors. By assigning a specific Dirichlet distribution to $\pi_{1:K}$, the number of "active" emission models tends to be smaller. Therefore, process knowledge has been mathematically integrated, but the iterative optimization algorithm may still stop at different local maxima and result in several possible solutions. However, according to (5.23), solution with the highest $\mathcal{L}(q(\theta))$ is said to have achieved the best approximation of $p(\theta \mid X_{1:T}, \eta)$ [39]. Thus, the optimized value of $\mathcal{L}(q(\theta))$ will be used for model selection.

The detailed expression is of the variational lower bound is repeated as

$$\mathcal{L}(q(\theta)) = \int q(\theta) \cdot \ln p(X_{1:T}, \theta \mid \eta) \cdot d\theta - \int q(\theta) \cdot \ln q(\theta) \cdot d\theta.$$

The first term can be calculated from the probability dependencies in Figure 5.4 and explicit forms in (5.12)-(5.19); the second term is determined individually by each proposal distribution. Here, a detailed presentation of these two statistics is listed for a reference

The first term is explicitly presented as in the following equation, where $\langle \cdot \rangle$ is the expectation based on $q(\theta)$:

$$\begin{aligned}
& \int q(\theta) \cdot \ln p(X_{1:T}, \theta \mid \eta) \cdot d\theta \\
= \; & \sum_{t=1}^{T} \sum_{k=1}^{K} \Big[\langle \ln p(X_t \mid S_t, U_t^{(k)}, I_t^{(k)}, \{\mu, H, \gamma\}^{(k)}) \rangle + \langle \ln p(U_t^{(k)} \mid I_t^{(k)}, \nu) \rangle + \langle \ln p(I_t^{(k)} \mid \pi_{1:K}) \rangle \Big] \\
& + \sum_{t=1}^{T} \langle p(S_t \mid S_{t-1}, A) \rangle + \langle \ln p(\pi_{1:K} \mid \alpha_{1:K}) \rangle + \langle \ln p(A \mid \alpha_a, \beta_a) \rangle \\
& + \sum_{k=1}^{K} \Big[\langle \ln p(\mu^{(k)} \mid \Lambda_\mu) \rangle + \langle \ln p(H^{(k)} \mid \Lambda_H) \rangle + \langle \ln p(\gamma^{(k)} \mid \alpha_\gamma, \beta_\gamma) \rangle \Big] \\
= \; & \sum_{t=1}^{T} \sum_{k=1}^{K} \Big[-\frac{m \cdot \ln 2\pi}{2} \langle I_t^{(k)} \rangle + \frac{m}{2} \langle \ln \gamma^{(k)} \rangle \langle \ln U_t^{(k)} I_t^{(k)} \rangle - \frac{1}{2} \langle U_t^{(k)} I_t^{(k)} \rangle \langle \sigma_t^2 \rangle \Big] \\
& + \sum_{t=1}^{T} \sum_{k=1}^{K} \Big[\langle I_t^{(k)} \rangle [\frac{\nu}{2} \ln \frac{\nu}{2} - \ln \Gamma(\frac{\nu}{2})] + (\frac{\nu}{2} - 1) \langle \ln U_t^{(k)} I_t^{(k)} \rangle - \frac{\nu}{2} \langle U_t^{(k)} I_t^{(k)} \rangle + \langle I_t^{(k)} \rangle \langle \ln \pi_k \rangle \Big] \\
& + \sum_{t=1}^{T} \sum_{j=1}^{d} \Bigg[\frac{1}{2} \langle \ln \frac{1}{1 - a_j^2} \rangle - \frac{1}{2} \langle S_t S_t' \rangle_j \langle \frac{1}{1 - a_j^2} \rangle + \langle S_t S_{t-1}' \rangle_j \langle \frac{a_j}{1 - a_j^2} \rangle - \frac{1}{2} \langle S_{t-1} S_{t-1}' \rangle_j \langle \frac{a_j^2}{1 - a_j^2} \rangle \Bigg]
\end{aligned}$$

$$
\begin{aligned}
&- \frac{T \cdot d}{2} \ln 2\pi + \sum_{k=1}^{K} \left[-\frac{m}{2} \ln 2\pi + \frac{1}{2} \ln |\Lambda_\mu| - \frac{1}{2} tr\{\langle \mu^{(k)} (\mu^{(k)})' \rangle \cdot \Lambda_\mu\} \right] \\
&+ \sum_{k=1}^{K} \left[-\frac{m \cdot d}{2} \ln 2\pi + \frac{d_X}{2} \ln |\Lambda_H| - \frac{1}{2} tr\{\langle (H^{(k)})' H^{(k)} \rangle \cdot \Lambda_H\} \right] \\
&+ \sum_{k=1}^{K} \left[\alpha_\gamma \ln(\beta_\gamma) - \ln \Gamma(\alpha_\gamma) + (\alpha_\gamma - 1) \langle \ln \gamma^{(k)} \rangle - \beta_\gamma \langle \gamma^{(k)} \rangle \right] \\
&+ \sum_{j=1}^{d} [-\ln B(\alpha_a, \beta_a) + (\alpha_a - 1) \langle \ln a_j \rangle - (\beta_a - 1) \langle \ln(1 - a_j) \rangle] \\
&+ \ln \Gamma(\sum_{k=1}^{K} \alpha_k) + \sum_{k=1}^{K} [(\alpha_k - 1) \langle \ln \pi_k \rangle - \ln \Gamma(\alpha_k)], \qquad (5.68)
\end{aligned}
$$

where $|\cdot|$ stands for the determinant of a matrix. and $\ln \Gamma(\cdot)$ is the Log-Gamma function. The residual term $\langle \sigma_t^2 \rangle$ is defined in (5.37).

The second term in the variational lower bound (5.23) is presented as the summation of individual entropies:

$$\mathcal{H}\{q(\theta)\} = \sum_{j=1}^{d} (a_j)_{ent} + (U \mid I)_{ent} + I_{ent} + S_{ent} + \sum_{k=1}^{K} \left[\mu_{ent}^{(k)} + H_{ent}^{(k)} + \gamma_{ent}^{(k)} \right] + \pi_{ent}. \qquad (5.69)$$

The entropy $(a_j)_{ent}$ of sample-based posteriors is calculated by distributing samples into N_{bin} bins, aggregating the weights $f_n(a_j)$ for the n-th bin, and calculating entropy with the width wid_n of n-th bin as

$$(a_j)_{ent} \approx - \sum_{n=1}^{N_{bin}} f_n(a_j) \cdot \ln \frac{f_n(a_j)}{wid_n}, \qquad (5.70)$$

The entropy for the conditional random variable $U \mid I$ is obtained from the estimated Gamma distributions:

$$(U \mid I)_{ent} = \sum_{t=1}^{T} \sum_{k=1}^{K} \langle I_t^{(k)} \rangle \left[\alpha_{U_t}^{(k)} - \ln\{\beta_{U_t}^{(k)}\} + \ln \Gamma(\alpha_{U_t}^{(k)}) + (1 - \alpha_{U_t}^{(k)}) \psi\{\alpha_{U_t}^{(k)}\} \right]. \qquad (5.71)$$

Regarding identity variables, the entropy calculation is simply enumerating the possibilities of discrete events:

$$I_{ent} = \sum_{t=1}^{T} \sum_{k=1}^{K} -\langle I_t^{(k)} \rangle \cdot \ln \langle I_t^{(k)} \rangle. \qquad (5.72)$$

Similar as in Chapter 3, the entropy of latent feature $S_{1:T}$ is based on a representation of sequenced Normal distributions. Here, the detailed equation is listed, where the results from aforementioned smoothing step are utilized:

$$S_{ent} = \left[T \cdot d - \frac{(T-2) \cdot d}{2} \right] \ln(2\pi e) - \frac{1}{2} \sum_{t=2}^{T-1} \ln |P_t^{(s)}|$$

$$+\frac{1}{2}\sum_{t=1}^{T-1}\ln\begin{vmatrix} P_{t+1}^{(s)} & P_{t+1}^{(s)}\cdot G_t' \\ G_t\cdot P_{t+1}^{(s)} & P_t^{(s)} \end{vmatrix}. \tag{5.73}$$

The entropies from multiple emission models are mutual independent, for each model $M^{(k)}$, the entropy consists of following terms:

$$\mu_{ent}^{(k)} = \frac{m}{2}(1+\ln(2\pi)) + \frac{1}{2}\ln|P_\mu^{(k)}|, \tag{5.74}$$

$$H_{ent}^{(k)} = \sum_{i=1}^{m}\left[\frac{d_S}{2}(1+\ln(2\pi)) + \frac{1}{2}\ln|P_{h_i}^{(k)}|\right], \tag{5.75}$$

$$\gamma_{ent}^{(k)} = \alpha_\gamma^{(k)} - \ln\{\beta_\gamma^{(k)}\} + \ln\Gamma(\alpha_\gamma^{(k)}) + (1-\alpha_\gamma^{(k)})\psi\{\alpha_\gamma^{(k)}\}. \tag{5.76}$$

As for the overall weight of model identities, the entropy is calculated from the Dirichlet distribution:

$$\begin{aligned}\pi_{ent} = &-\ln\Gamma(\sum_{k=1}^{K}\alpha_\pi^{(k)}) - (K-\sum_{k=1}^{K}\alpha_\pi^{(k)})\cdot\psi\{\sum_{k=1}^{K}\alpha_\pi^{(k)}\} + \sum_{k=1}^{K}[\ln\Gamma(\alpha_\pi^{(k)}) \\ &-(\alpha_\pi^{(k)}-1)\psi\{\alpha_\pi^{(k)}\}].\end{aligned} \tag{5.77}$$

In addition to the selection among multiple results, two structural parameters d (the dimension of latent space) and K (the number of emission models) can also be optimized by the converged $\mathcal{L}(q(\theta))$. Usually, d and K are initialized with large values for a comprehensive description of data. Then, benefiting from ARD, the learning procedure can be accelerated with the preliminary results from Variational Inference. While different heuristic strategies can be adopted to search d and K [39], the principle is always to maximize $\mathcal{L}(q(\theta))$. A practical realization will be discussed in the following application section.

5.4.4 Numerical Simulations

In this section, the learning strength of the proposed algorithm is demonstrated with two simulation examples. The first simulation will illustrate the abilities of the proposed model in fitting non-linear data. The second simulation shows that our algorithm can determine the number of modes and the dimension of latent space simultaneously.

In the first example, a set of non-linear data is generated to illustrate the procedure of fitting local linear models. By this simulation, practical steps for applying this learning algorithm are discussed. The data, consisting of 600 samples from a noisy shrinking spiral, is produced as [39]

$$X_i = [(13-0.5t_i)\cos t_i, \quad -(13-0.5t_i)\sin t_i, \quad t_i] + w_i,$$

$$t_i = \frac{4\pi}{T} \cdot i, \quad T = 600, \quad w_i \sim N(0, diag[0.5, 0.5, 0.5]), \tag{5.78}$$

where the parameter t_i is the temporal position along the spiral. Visualization from two angles is presented in Figure 5.5.

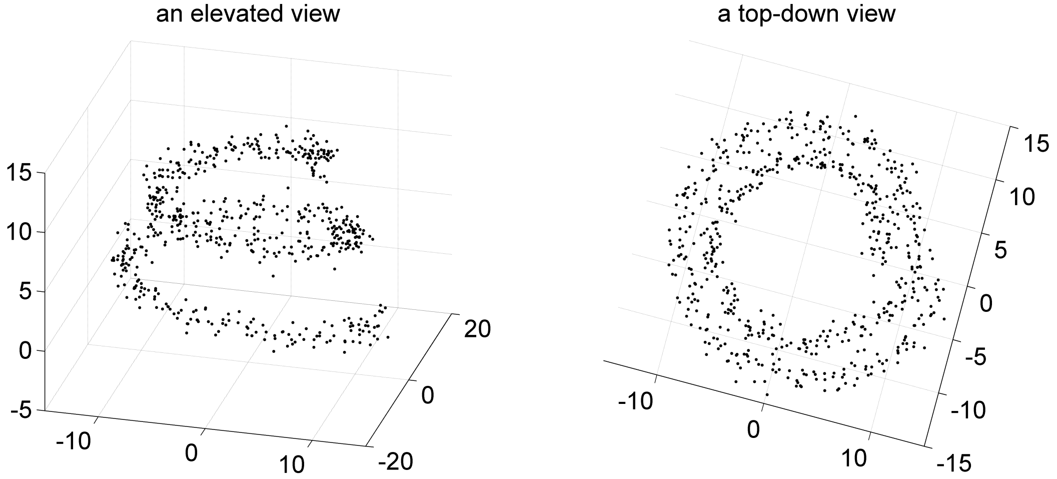


Figure 5.5: The Spiral Data in 3-Dimension Space with a 1-Dimension Non-Linearly Embedded

As the dimension of observations m is 3, the modelling procedure starts with two-dimensional latent space ($d = 2$), similar to the application of PPCA [22]. The initial number of emission models is set to seven ($K = 7$) for a sufficient description of the data. Figure 5.6 shows five individual evolutions of parameter learning, where the x-axis is the updating iterations and the y-axis is $\mathcal{L}(q(\theta))$. Although all these $\mathcal{L}(q(\theta))$s correctly (monotonically) increase, the converged values are different, representing several local minima. The optimal model will be selected according to these converged values of variational Lower Bound, such as selecting the third iteration in this example.

In order to explore further structural settings, the model with smaller latent dimension and less number of models could also be tried. As one advantage of ARD, the number of active models can be determined from the posterior of $\pi_{1:K}$. If the k-th cluster is not responsible for any observation, the estimated $\langle \pi_k \rangle$ will be smaller than $\frac{1}{T}$, indicating a possible removal of this cluster. For example, with four modes being useful in the first trial, following tests for K shall start from four. Several values of $\mathcal{L}(q(\theta))$ are listed in Table 5.1 for additional structure sets, and typical differences are visualized in Figure 5.7. In addition, it can be observed that the converged $\mathcal{L}(q(\theta))$ is neither dependent on initial $\mathcal{L}(q(\theta))$ nor the optimizing speed, which is from the inherit complexity of this feature extraction model.

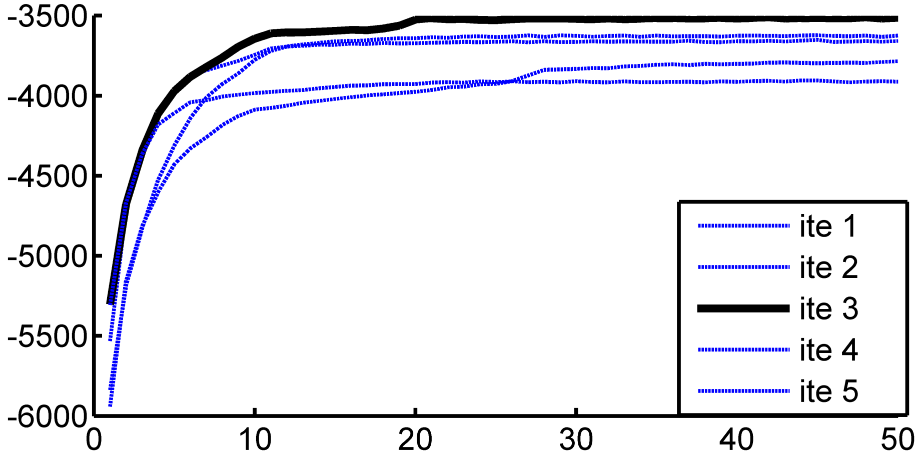


Figure 5.6: Evolutions of the variational Lower Bound with Randomized Initializations

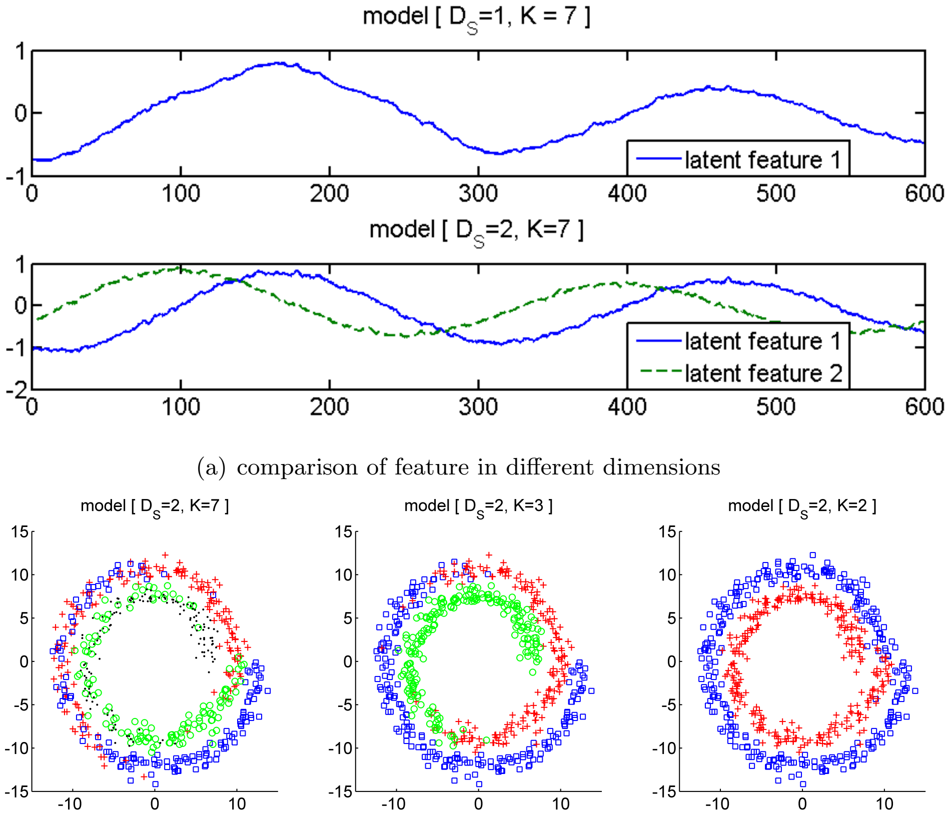
By comparing different latent dimensions, it is found that both one-dimensional and two-dimensional features can represent the driving force well. Particularly, the two-dimensional features have a good approximation of sinusoidal signals. When the latent dimension is reduced to one, distortion in $S_{1:T}$ is observed, which is also reflected by a less optimal value in Table 5.1. For the comparison of different K, labelling results are visualized in the bottom plots of Figure 5.7, where different models are well clustered along the spiral. As indicated by $\mathcal{L}(q(\theta))$, either $K = 3$ or $K = 2$ can be considered appropriate for the proposed model. Conventional Bayesian mixture of factor analysers yields 14 clusters [39]. To summarize, when the non-linearity of process data is from the variations along time, the estimated dynamic latent features can capture it through temporal relations and reduce the number of emission models (or analyzers) significantly.

model	[d_S=2, K=7]	[d_S=1, K=7]	[d_S=2, K=4]	[d_S=2, K=3]	[d_S=2, K=2]
$\mathcal{L}$	-3512	-4677	-3494	-3469	-3622

Table 5.1: Comparison of the Variational Lower Bounds

In the second example, a synthetic data set was generated to show the strength of identifying true model identities by the proposed method. Here, the data is projected from 3 sequences of latent features through 3 emission models, generating 5-dimensional observations. The latent features are generated with a transition matrix $A = diag\{0.97, 0.9, 0.85\}$, and emission models are given as

$$\mu^{(1)} = [10, 8, 20, -10, 0], \qquad \mu^{(2)} = [10, 8, 10, 0, 6], \qquad \mu^{(3)} = [6, 8, 2, 6, 6],$$



(a) comparison of feature in different dimensions

(b) comparison of different clustering results

Figure 5.7: Comparisons of Features and Clusters for Different Identified Models

$$H^{(1)} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \\ 2 & 3 & 1 \\ 4 & -2 & 0 \end{bmatrix}, \quad H^{(2)} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -2 & 0 \\ 2 & 0 & 1 \\ 0 & 0 & 3 \\ 0 & 2 & -4 \end{bmatrix}, \quad H^{(3)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 4 & 0 \\ 4 & 0 & 2 \\ -1 & -1 & 2 \\ 1 & 3 & 2 \end{bmatrix},$$

$$\gamma^{(1)} = 1, \qquad \gamma^{(2)} = 3, \qquad \gamma^{(3)} = 1.5,$$

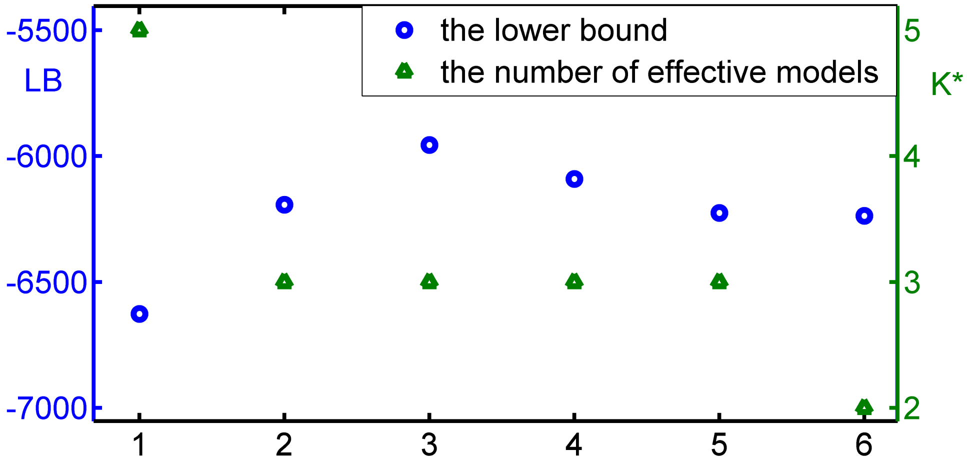


Figure 5.8: Selecting the Number of Latent Dimension and the Number of Emission Models

Similar to the previous example, the inference procedure starts from $K = 6$ and gradually attempts latent dimensions from $d = 1$ until $d = 6$. The simulation results are presented in Figure 5.8, where the variational Lower Bound $\mathcal{L}(q(\theta))$ and the number of "active" models (K^*) are plotted against the dimension of latent space (d). Based on these two trends, the latent dimension can be selected at the peak of $\mathcal{L}(q(\theta))$ ($d = 3$), and the corresponding K^* is 3. From this demonstration, it is shown that the proposed learning algorithm can identify the number of emission models and the dimension of latent space correctly.

To validate the result of learning the model identity sequence, the proposed method is compared with three popular methods for modelling mixtures. The first one is the "K-means" method, which fits a given number of Gaussian mixtures in input space. The second is the mixture of probabilistic principal component analysis (MPPCA) [114], which can learn both latent features and the number of models. The last one is the robust mixture of probabilistic principal component analysis (RMPPCA) [109], which enhances MPPCA by implementing noise modelling with Student-t distributions. The dimension of latent space for MPPCA and RMPPCA is set to the real value of three, and multiple trials are conducted to ensure that only their optima are compared with the proposed method for

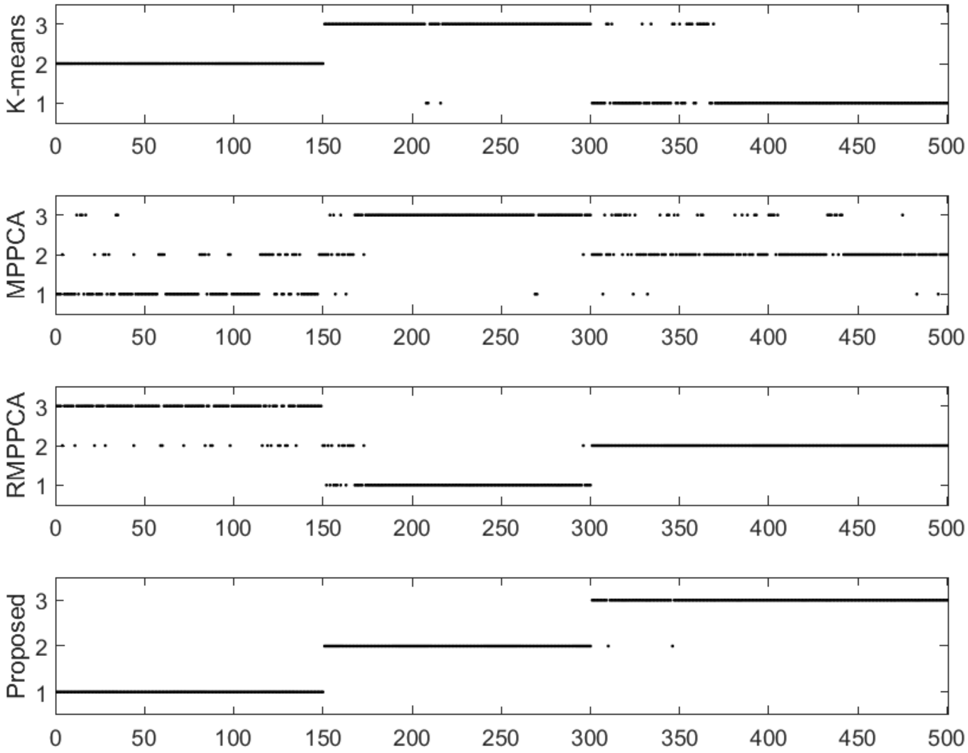


Figure 5.9: Comparison of Estimated Model Identities

fairness. The estimated result of model identities is plotted in Figure 5.9. It can be observed that the proposed feature extraction method can provide the best identification result for model identities, which has only two misclassified samples.

5.5 On-line Application with Variational Inference Results

Based on the above two simulations, the proposed method is validated in the off-line learning procedure. In order to apply this off-line estimated model, an online feature extraction procedure is expected. In this section, the result from the aforementioned Bayesian learning is used to learn the latent feature for new measurement samples. In the industrial application of a steam generation process, the corresponding latent features are shown to be informative in estimating the steam quality.

5.5.1 Filtering States with Results of Variational Inference

To apply the result from Bayesian extraction of dynamic latent features, the distribution of the time-invariant parameters will no longer be updated. In this study, the converged proposal distribution $q(A)$, $q(\pi_{1:K})$, and $q(\mu^{(1:K)}, H^{(1:K)}, \gamma^{(1:K)})$ will be utilized.

While applying mixture models with static latent features, such as in MPPCA, the estimation of current model identity I_t only depends on the current observation X_t (or some predefined prior belief in Bayesian modelling). As an enhancement, model identities will be determined from both observations and dynamic latent features in our proposed method. As a part of the off-line learning procedure, the online feature extraction strategy can be derived as follows.

- Step-1, estimate the predicted distribution for latent state S_t, which is Normal distribution $\mathcal{N}(m_t^{(p)}, P_t^{(p)})$. For the initial case, set $m_1^{(p)} = \vec{0}$ and $P_1^{(p)} = \mathbb{I}_d$; For the case with $t > 1$, calculate $m_t^{(p)}, P_t^{(p)}$ based on (5.56) and (5.57).

- Step-2, estimate the identity variable I_t and associated $U_t \mid I_t$:

$$\langle I_t^{(k)} \rangle \propto \langle \mathcal{S}(X_t \mid \mu^{(k)} + H^{(k)} \cdot S_t^{(p)}, (\gamma^{(k)})^{-1}\mathbb{I}_m, \nu) \rangle, \qquad \sum_{k=1}^{K} \langle I_t^{(k)} \rangle = 1,$$
$$\langle U_t^{(k)} \ I_t^{(k)} \rangle = \langle I_t^{(k)} \rangle \cdot \frac{\nu + m}{\nu + \langle \gamma^{(k)} \rangle \langle \sigma_t^2 \rangle},$$

  where $\langle \sigma_t^2 \rangle$ can be obtained from (5.37).

- Step-3, calculate $\tilde{F}_T^{(P)}, \tilde{F}_T^{(m)}$ based on (5.50), (5.51), and $\langle U_t^{(k)} \ I_t^{(k)} \rangle$.

- Step-4, estimate $S_t = S_t^{(f)}$ based on (5.60) and (5.59).

In this online learning strategy, the forward path (5.54) is performed with updating equation for I_t and U_t, where the distribution of parameters and dynamic properties of $S_{1:T}$ can be fully utilized. By implementing this on-line algorithm, I_t and the current latent feature S_t will be estimated jointly. It should be noted that since future information cannot be used to smooth current I_t in the online application, the expectation terms such as $\tilde{F}_T^{(P)}$ and $\tilde{F}_T^{(m)}$ are only calculated with the $t = T$ case. Also, the overall weight of $\pi_{1:K}$ is not applicable to tune the identity variable, such as in the updating equation (5.40). Thus, slight differences may exist between off-line and online results. Figure 5.10 shows model identities from these two learning procedures. In this figure, the green line shows the result of the off-line training algorithm in the previous section; the blue dot line shows a repeat

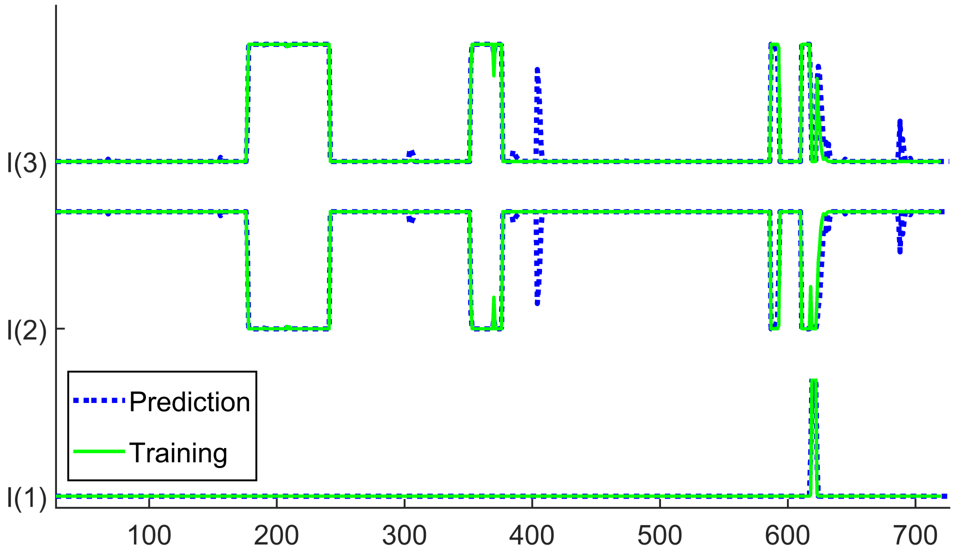


Figure 5.10: Comparison between Off-line and On-line Model Identity Estimations

application with the online learning algorithm. It can be observed that most variations in the model identity sequence can be captured with the above on-line algorithm, and the off-line training method indeed prevented some unnecessary peaks.

5.5.2 Process Description of Steam Quality Inferential Sensor

Equipped with on-line feature extraction, the proposed method can be tested with industrial applications. In this case study, an inferential sensing project in steam generation processes is introduced to test our algorithm. Figure 5.11 and Table 5.2 illustrate a simplified diagram and sensor network for the steam generation plant, called Once-Through Steam Generator (OTSG). In this process, boiler feed water (BFW) and fuel gas (FG) are sent to the heat exchanger (including both convection section and radiation section) to produce wet steam (combination of steam and water). The water fluid is split into several passes within the heat exchanger and recombined at the outlet end for the further separation. In order to reduce unnecessary fuel consumption, the steam quality (mass fraction of steam) needs to be monitored.

The current solution relies on the slow-rate manual sampling, shown as the yellow circle in Figure 5.11. On the other hand, there are fast-sampled process variables related to steam quality: BFW-T, BFW-P, and In-F measure temperature, pressure and flow rate for the inlet fluid; Skin-T, WS-P, and DP measure temperature, pressure and differential pressure for the outlet fluid; FG-F shows the flow rate of energy flow. Above measurements have

Variable	Description	Unit	Sampling interval
BFW-T	Temperature of boiler feed water	°C	1 minute
BFW-P	Pressure of boiler feed water	kPa	1 minute
In-F	Inlet flowrate of individual passes	t/hr	1 minute
Skin-T	Outlet temperature of individual passes	°C	1 minute
DP	Differential pressure of individual passes	kPa	1 minute
WS-P	Pressure of recombined wet steam	kPa	1 minute
FG-F	Flowrate of input fuel gas	kg/hr	1 minute
SQ	Reference output of steam quality	%	∼ 6 hours

Table 5.2: List of available process variables

included all the direct measurements for the individual passes and were suggested as the influential variables by engineers and also by our previous studies [115]. The objective is to provide a reliable on-line estimation for individual pass SQ by using these fast-rate process measurements. From the background information, the switched behaviour is mainly due to the multiple operation regions and the composition of energy flow. Because of the fixed equipment size and a relatively steady set-point (SP), the latent features can be assumed to follow a unique dynamic model.

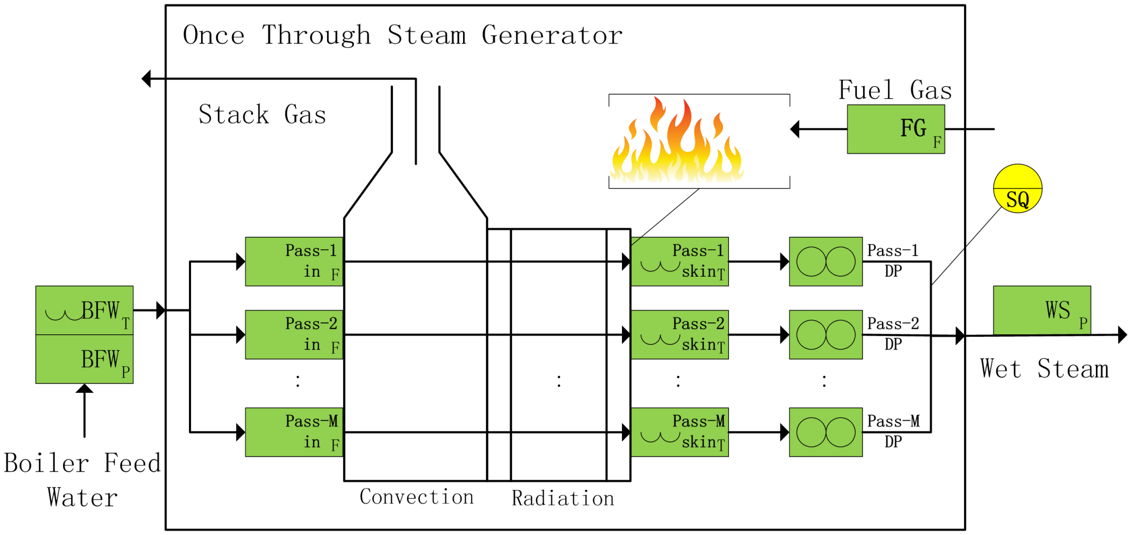


Figure 5.11: Flow Diagram of Once Through Steam Generator

5.5.3 Apply the Bayesian Feature Extractions

From the data visualization in Figure 5.12, there are three significant challenges in this project: (1) unknown disturbances, the composition of fuel gas is not measured and will result in different regions for FG-F; (2) frequent occurrence of outliers (also shown in Figure 5.2), since the DP measurements are unreliable and possess relatively large noise; (3) the random sampling of reference outputs, since although the 7 inputs are regularly measured at 1-min interval, the reference output is manually measured at roughly 6-hour interval. Considering the significant difference in sampling intervals between the model inputs and output, unsupervised learning for feature extraction is preferred.

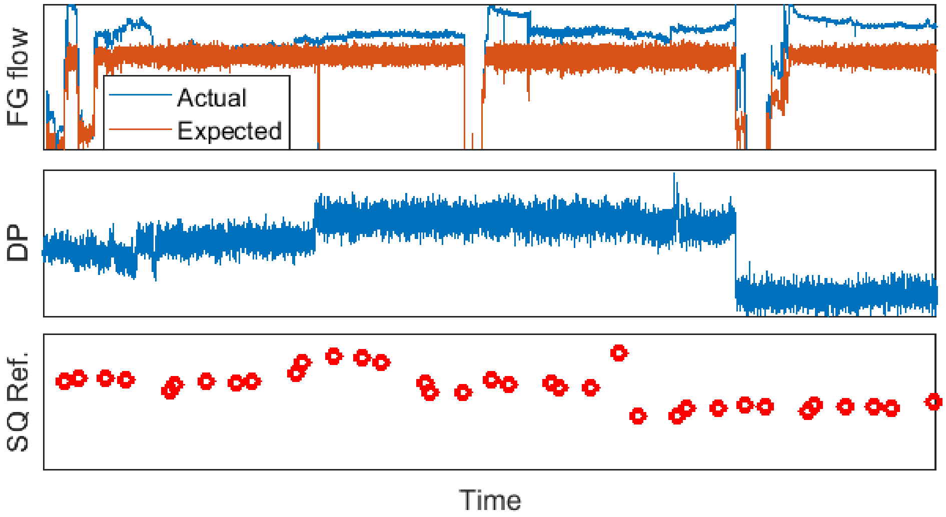


Figure 5.12: Typical Measurements from an Operating Steam Generator

As discussed in previous sections, the robust emission models and the probabilistic learning algorithm are designed to overcome the challenge from these noisy measurements and outliers. To address the unknown disturbances, the switched emission models are learned along with dynamic latent features in the proposed method. Due to the limited computing power in the industrial DCS system, complete online learning strategies are deemed not suitable. In this study, three developed and widely applied algorithms: principal components regression (PCR), partial least squares (PLS), and slow feature regression (SFR)[46] are compared with our proposed algorithm. As a natural enhancement, lagged inputs are incorporated to extend the above methods to their dynamic versions.

In this validation, 2-month data from 2 OTSGs are available; the first quarter of data is used for training and the rest for verification. In the off-line learning procedure, the proposed method identified the latent feature as in four-dimensional space ($d_S = 4$) for both 2 OTSGs, representing the number of hidden correlations. It also identified three switched models ($K = 3$) for OTSG-1 and two switched models ($K = 2$) for OTSG-2, relaxing the traditional modelling assumption of linear and unchanged correlations among

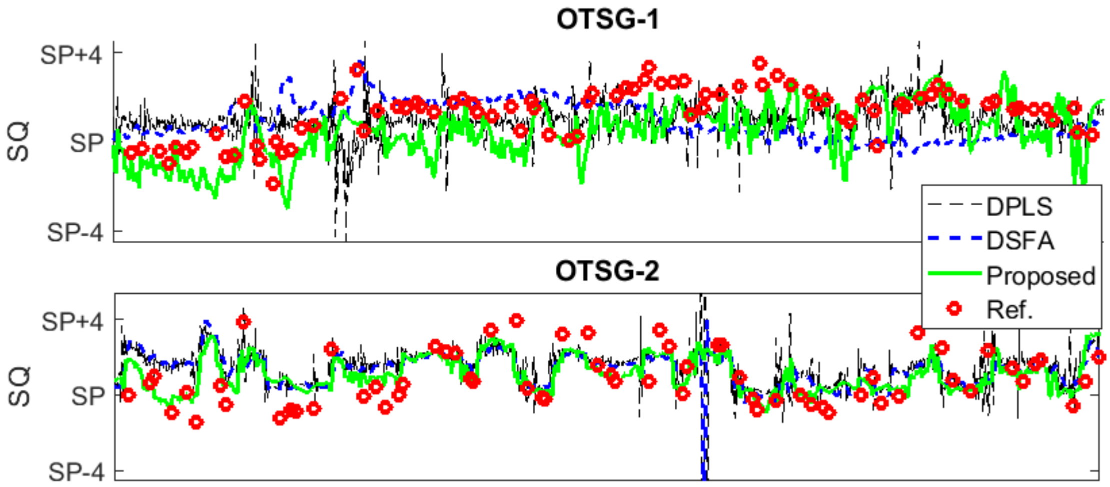


Figure 5.13: Comparison of Prediction from Different Latent Features

process variables. As the switching behaviours have been addressed in this feature extraction procedure, the regression model can then be established between the extracted features and the reference output.

	OTSG-1			OTSG-2		
	CC	MAE	RMSE	CC	MAE	RMSE
DPCR	0.23	1.14	1.28	¡0.1	0.93	0.95
DPLS	0.23	1.22	1.28	¡0.1	1.04	1.03
DSFA	0.39	1.03	1.30	0.30	0.64	0.94
Proposed	0.52	1.10	1.24	0.32	0.69	0.86

Table 5.3: Validation Results of Steam Quality Inferential Sensing

As for the aforementioned MPPCA and RMPPCA, reference data points have to be distributed to different clusters. With a limited number of reference output samples, this will severely harm the regression results (in fact even worse than DPCR). In the on-line validation, each predicted SQ is evaluated with the Pearson correlation coefficient(CC), mean absolute error (MAE), and root mean square error (RMSE) respectively. Table 5.3 summarizes the optimal performance of each method, where CC is used to determine the optimal latent dimension for each algorithm. It can be observed that our proposed method can provide the highest correlation and the lowest RMSE for both OTSGs. Figure 5.13 shows the time trend of each prediction against the reference (DPCR is omitted for clarity), where the y-axis is the steam quality value. Although the reference itself is relatively steady but noisy, the proposed method still captured the overall trend of steam quality. The testing results on these two sets of industrial data indicate that the features extracted from our prosed method can predict steam quality better than other competing algorithms.

5.6 Extending On-line Feature Learning Algorithm States to General Multiple Models

In this part, we convert the above on-line learning algorithm to a new approach to state estimation of multiple state-space models. Unlike the traditional methods (including the interacting multiple model (IMM) algorithm) that approximate a Gaussian mixture distribution with a single Normal distribution, the proposed method approximates the joint probability density functions (PDFs) of state and model identity through Bayesian inference. It is shown that the proposed method reduces the approximation error considerably, and improves estimation accuracy without increasing computational cost. Analysis of its specific features as well as a potential extension is also presented. Numerical examples with a practical-oriented simulation are employed to illustrate the effectiveness of our proposed method.

5.6.1 General Linear Multiple State-Space Models

By assuming that the switching actions are independent of state propagation and denoting the model identity as $r_t \in \{1, ..., K\}$, the linear multiple-model state-space system under consideration is specified by

$$x_t = F^{(r_t)} \cdot x_{t-1} + T^{(r_t)} \cdot v_t, \tag{5.79}$$

$$y_t = H^{(r_t)} \cdot x_t + w_t, \tag{5.80}$$

where k is the time instant, x_t denotes the state, y_t is the observation, and $v_t \sim \mathcal{N}(0, Q^{(r_t)})$ and $w_t \sim \mathcal{N}(0, R^{(r_t)})$ denote the process and measurements noise with Gaussian distributions, respectively. In this paper, it is assumed that all the parameters involving $F^{(r_t)}$, $T^{(r_t)}$, $H^{(r_t)}$, $Q^{(r_t)}$, and $R^{(r_t)}$ are available once the model identity r_t is determined. Here, r_t can be described with the Markovian dynamics:

$$P_r(r_t = j \mid r_{t-1} = i, r_{t-2}, ...) = P_r(r_t = j \mid r_{t-1} = i) = p_{ij}, \tag{5.81}$$

where $i, j = 1, 2, ..., K$.

Following a general definition, the objective of a state estimator is to estimate the state variable x_t given the observations $Y_{1:t} \triangleq \{y_1, y_2, ..., y_t\}$. In other words, the state estimator shall provide the posterior distribution $p(x_t \mid Y_{1:t})$ under the probabilistic framework. A predefined cost function is then employed for the point estimate. For the models considered, if the model identity r_k is known, the conventional KF can provide the optimal estimates conveniently. Otherwise, estimation algorithms should consider the uncertainty of r_t during filtering.

The problem considered in this paper is formulated as follows. Given the system model (5.79) and (5.80), derive an estimator by calculating the posterior distribution $p\left(x_t \mid Y_{1:t}\right)$ using the VBI approach. The main purpose is to reduce the approximation error introduced in the IMM method without increasing computational load.

5.6.2 State Estimation with Variational Methods

This vectorial representation of the model identity, in Section 5.3, is able to provide a compact probability description of $\{x_t, I_t\}$ in the log-likelihood formulation, which facilitates the approximation of posterior considerably. Using I_t in (5.7), the probabilistic description for the transition and emission function will be

$$p(x_t \mid x_{k-1}, I_t) = \prod_{k=1}^{K} \left[\mathcal{N}(x_t;\ F^{(k)}x_{t-1},\ \hat{Q}^{(k)})\right]^{I_t^{(k)}}, \tag{5.82}$$

$$p(y_t \mid x_t, I_t) = \prod_{k=1}^{K} \left[\mathcal{N}(y_t;\ H^{(k)}x_t,\ R^{(k)})\right]^{I_t^{(k)}}, \tag{5.83}$$

where $\hat{Q}^{(k)} \triangleq T^{(k)}Q^{(k)}(T^{(k)})'$. With the product operator, I_t serves as the exponential terms for x_t.

Since we are interested in the posterior PDF $p\left(x_t \mid Y_{1:t}\right)$ and $p\left(I_t \mid Y_{1:k}\right)$, two proposal distributions $q(I_t)$ and $q(x_t)$ are used to approximate the joint posterior distribution as

$$q(x_t) \cdot q(I_t) \to p\left(x_t, I_t \mid Y_{1:t}\right). \tag{5.84}$$

The quality of this approximation is evaluated by the Kullback-Leibler (K-L) divergence as

$$D_{KL}\{q(I_t)\ q(x_t)\ ||\ p\left(x_t, I_t \mid Y_{1:t}\right)\} = \int q(I_t)\ q(x_t) \cdot \ln \frac{q(I_t)\ q(x_t)}{p(x_t, I_t \mid Y_{1:t})} \cdot dx_t \cdot dI_t. \tag{5.85}$$

Note that this divergence can be minimized to zero if both sides of (5.84) are exactly same; otherwise, it remains positive. To avoid calculating with the unknown distribution $p\left(x_t, I_t \mid Y_{1:t}\right)$, VB uses the following equality to increase explicitness:

$$\ln p\left(y_t \mid Y_{1:t-1}\right) = D_{KL}\{q(I_t)\ q(x_t)\ ||\ p\left(x_t, I_t \mid Y_{1:t}\right)\} + \mathcal{L}_{q(I_t)q(x_t)}\left[y_t \mid Y_{1:t-1}\right], \tag{5.86}$$

where $\mathcal{L}_{q(I_t)q(X_t)}[\cdot]$ is defined as the variational Lower Bound (LB):

$$\mathcal{L}_{q(I_t)q(x_t)}\left[y_t \mid Y_{1:t-1}\right] = \int q(I_t)\ q(x_t) \cdot \ln \frac{p\left(y_t, x_t, I_t \mid Y_{1:t-1}\right)}{q(I_t)\ q(x_t)} \cdot dx_t \cdot dI_t. \tag{5.87}$$

Combining (5.85) and (5.87), equation (5.86) can be verified through the Bayes' rule. Since the recurrent likelihood $p\left(y_t \mid Y_{1:t-1}\right)$ is independent from states, it can be treated as a constant. Under this framework, the proposal distributions $q(I_t)$ and $q(x_t)$ that provide the highest value of $\mathcal{L}_{q(I_t)q(x_t)}\left[y_t \mid Y_{1:t-1}\right]$ will be treated as the best approximation.

5.6.3 Approximate the Filter with Proposal Distributions

The realization of above approximation consists of two steps: deriving the posterior distribution $p\left(I_t \mid Y_{1:t}\right)$ and estimating the proposal distribution $q(x_t)$. Before describing these estimation equations, it would be helpful to clarify all the available information in the joint distribution $p\left(y_t, x_t, I_t \mid Y_{1:t-1}\right)$. The superscripts $^{(\mathrm{p})}$ and $^{(\mathrm{f})}$ will be used to denote the predicted (given $Y_{1:t-1}$) and filtered (given $Y_{1:t}$) results, respectively. First, the model identity can be separated out through

$$p(y_t, x_t, I_t \mid Y_{1:t-1}) = p(y_t, x_t \mid Y_{1:t-1}, I_t) \cdot p(I_t \mid Y_{1:t-1}),$$

where $p(I_t \mid Y_{1:t-1}) = Mul(I_t;\ \pi_{1:K}^{(\mathrm{p})})$ can be obtained from the transition function for model identity in (5.81). Then, $p\left(y_t, x_t \mid Y_{1:t-1}, I_t\right)$ can be written as

$$\int_{x_{t-1}} p(y_t \mid x_t, I_t) \cdot p(x_t \mid x_{k-1}, I_t) \cdot p(x_{t-1} \mid Y_{1:t-1}).$$

Since we focus on the mean and variance of the estimated state, $p\left(x_{t-1} \mid Y_{1:t-1}\right)$ is denoted as $\mathcal{N}(m_{t-1}^{(\mathrm{f})}, P_{t-1}^{(\mathrm{f})})$. With the likelihood terms introduced in equations (5.82) and (5.83), the joint distribution can be expanded as

$$\begin{aligned} p(y_t, x_t, I_t \mid Y_{1:k-1}) &= \prod_{k=1}^{K} \left[\pi_j^{(\mathrm{p})} \cdot \mathcal{N}(y_t;\ H^{(k)} \cdot x_t, R^{(k)})\right]^{I_t^{(k)}} \\ &\times \int_{x_{t-1}} \prod_{k=1}^{K} \left[\mathcal{N}(x_t;\ F^{(k)} \cdot x_{t-1}, \hat{Q}^{(k)})\right]^{I_t^{(k)}} \cdot \mathcal{N}(x_{t-1};\ m_{t-1}^{(\mathrm{f})}, P_{t-1}^{(\mathrm{f})}) \\ &= \prod_{k=1}^{K} \left[\pi_j^{(\mathrm{p})} \cdot \mathcal{N}(y_t;\ H^{(k)} \cdot x_t, R^{(k)}) \cdot \mathcal{N}(x_t;\ m_t^{(\mathrm{p})}[k], P_t^{(\mathrm{p})}[k])\right]^{I_t^{(k)}}. \end{aligned}$$

The second equality is because of the binary property of $I_t^{(k)}$, allowing the exchange of integration and product operator. Thus, the predicted parameters can be derived as

$$m_t^{(\mathrm{p})}[k] = F^{(k)} \cdot m_{t-1}^{(\mathrm{f})}, \tag{5.88}$$

$$P_t^{(\mathrm{p})}[k] = F^{(k)} \cdot P_{t-1}^{(\mathrm{f})} \cdot (F^{(k)})' + \hat{Q}^{(k)}. \tag{5.89}$$

Based on this joint distribution, the posterior probability of I_t is obtained by applying Bayes' rule and integrating x_t, which is given by

$$p(I_t \mid Y_{1:t}) = \frac{p(I_t, y_t \mid Y_{1:t})}{p(y_t \mid Y_{1:t})} \propto \int_{x_t} p\left(y_t, x_t, I_t \mid Y_{1:t-1}\right),$$

where $p(y_t \mid Y_{1:t})$ is the normalizing constant. This integration is straightforward within conjugate distributions. First, the support of identity vector in allows a move-in of predict distribution of X_t:

$$\begin{aligned} & p(Y_t, X_t, I_t \mid Y_{1:t-1}) \\ & \quad = \prod_{k=1}^{K} \Big[\mathcal{N}(Y_t;\ \mu^{(k)} + H^{(k)} \cdot X_t, R^{(k)}) \cdot \pi_k^{(\mathrm{p})} \cdot \mathcal{N}(X_t;\ m_t^{(\mathrm{p})}, P_t^{(\mathrm{p})})\Big]^{I_t^{(k)}}. \end{aligned} \tag{5.90}$$

Then, switching the integration and summation will provided the result as

$$\int_{x_t} p(y_t, x_t, I_t \mid Y_{1:t-1}) = \prod_{k=1}^{K} \left[\pi_k^{(\mathrm{p})} \cdot \Lambda_k(y_t)\right]^{I_t^{(k)}},$$

where $\Lambda_k(y_t) \triangleq \mathcal{N}(y_t \mid H^{(k)} m_t^{(\mathrm{p})}[k], R^{(k)} + H^{(k)} P_t^{(\mathrm{p})}[k](H^{(k)})')$. After normalization, the posterior (or filtered) distribution is formed as $p\left(I_t \mid Y_{1:t}\right) = Mul(I_t;\ \pi_{1:K}^{(\mathrm{f})})$, where

$$\pi_k^{(\mathrm{f})} \propto \pi_k^{(\mathrm{p})} \cdot \Lambda_k(y_t). \tag{5.91}$$

As can be seen, for the model identity, this derivation shares the same formulation as the counterpart in the IMM algorithm. Since there is no further approximation conducted, we will use (5.91) in the derivation of approximate $p\left(x_t \mid Y_{1:t}\right)$.

Instead of updating and merging mixtures of Gaussian distribution, the proposed estimator formulates $q(x_t)$ as a single Gaussian distribution and directly estimates it. Following the updating equation , this proposal distribution can be calculated with the parameter from (5.91), which is

$$\ln q(x_t) = \mathbb{E}_{p(I_t \mid Y_{1:t})}\left[\ln p(y_t, x_t, I_t \mid Y_{1:t-1})\right] + c.$$

Taking the logarithm of $p\left(y_t, x_t, I_t \mid Y_{1:t-1}\right)$ results in a quadratic function of x_t. After calculating the expectation, i.e., replacing I_t with $\pi_j^{(\mathrm{f})}$, the 2-nd order and 1-st order moments of x_t can be derived explicitly. At this point, $q(x_t)$ is updated as $\mathcal{N}(x_t;\ m_t^{(\mathrm{f})}, P_t^{(\mathrm{f})})$ with

$$m_t^{(\mathrm{f})} = P_t^{(\mathrm{f})} \sum_{k=1}^{K} \pi_k^{(\mathrm{f})} \left[(H^{(k)})' {R^{(k)}}^{-1} y_t + {P_t^{(\mathrm{p})}[k]}^{-1} m_t^{(\mathrm{p})}[k]\right],$$

$$P_t^{(\mathrm{f})^{-1}} = \sum_{k=1}^{K} \pi_k^{(\mathrm{f})} \left[(H^{(k)})' R^{(k)^{-1}} H^{(k)} + P_t^{(\mathrm{p})}[k]^{-1} \right]. \tag{5.92}$$

To conduct a sufficient optimization for both $q(I_t)$ and $q(x_t)$, normally adequate updating steps should be performed iteratively, until "$\mathcal{L}$" converges. In this study, however, the optimal posterior $p(I_t \mid Y_{1:t})$ can be obtained analytically as (5.91). Based on this result, the optimization for $q(x_t)$ can then be achieved in one single step. In other words, as long as the optimal estimate for model identity I_t is kept, the above approximation $q(x_t)$ will be the optimal solution with respect to the K-L divergence to achieve

$$q(x_t) \cdot p\left(I_t \mid Y_{1:t}\right) \rightarrow p\left(x_t, I_t \mid Y_{1:t}\right). \tag{5.93}$$

The core algorithm for our proposed identity expectation estimator (IEE) can now be summarized as follows:

1. Predict: Get distributions of (5.88) and (5.89);
2. Identify: Derive the posterior $p\left(I_t \mid Y_{1:t}\right)$ with (5.91);
3. Approximate: Use (5.92) to estimate $p\left(x_t \mid Y_{1:t}\right)$.

A recurrent filtering algorithm can always be written as a function of the prediction term and the correction term. To have a clear picture, the same state transition parameters are assumed for all models, making $x_t^{(\mathrm{p})}$ as the unique solution in the prediction step. Its updating step with multiple emission models is then formulated as

$$x_t^{(\mathrm{f})} = x_t^{(\mathrm{p})} + \sum_{k=1}^{K} \pi_k^{(\mathrm{f})} \cdot G_k \cdot \left[y_t - H^{(k)} \cdot x_t^{(\mathrm{p})} \right], \tag{5.94}$$

where $\pi_k^{(\mathrm{f})}$ is the posterior weight for different emission models, which shares the same results between IMM and IEE. The differences exist in the associated filter gains G_k, compared as

$$\begin{aligned}
G_k^{(IMM)} &= P_t^{(\mathrm{p})} \cdot (H^{(k)})' [H^{(k)} P_t^{(\mathrm{p})} (H^{(k)})' + R^{(k)}]^{-1}, \\
G_k^{(IEE)} &= [P_t^{(\mathrm{p})^{-1}} + \sum_{k'=1}^{K} \mu_{j'}^{(\mathrm{f})} (H^{(k')})' R^{(k')^{-1}} H^{(k')}]^{-1} (H^{(k)})' R^{(k)^{-1}}.
\end{aligned}$$

It can be noted that, for different models, IMM has a factor of $[H^{(k)} P_k^{(\mathrm{p})} (H^{(k)})' + R^{(k)}]^{-1}$, and IEE has $(H^{(k)})'[R^{(k)}]^{-1}$. Recalling the result of model identity in (5.91), the variance factor $[H^{(k)} P_t^{(\mathrm{p})} (H^{(k)})' + R^{(k)}]$ has been engaged in the Gaussian likelihood $\Lambda_k(y_t)$. Therefore, IMM amplifies this variance factor again in each filter gain G_k, causing a sensitive estimate of $x_t^{(\mathrm{f})}$; whereas the result of IEE avoids this problem and leads to a smoother state estimate. Another advantage is the reduction in computational load. Within the formulation (5.94), matrix inverse operator is performed K times in IMM filter, whereas only twice in IEE.

5.6.4 Extension for Other Latent Variables

This framework of utilizing the mathematical expectation can be easily extended to a broader application scope. As is often the case, additionally hidden states or more complicated models could be necessary for some specific requirements. By adopting a derivation similar to that in the previous section, the approximate state estimation can be conducted as long as the probabilistic description is available. As an example, the extension to robust filtering is illustrated below.

In order to be more robust to the observation outliers, the Student's t-distribution has been widely used for modelling the noise [116]. In this extension, the measurements noise w_t is assumed to be independently and identically distributed as

$$w_t \sim \mathcal{S}(0, R^{(k)}, \nu), \quad \mathit{if} : r_t = k, \tag{5.95}$$

where ν is the degree of freedom (DoF) for Student's t-distribution ($\mathcal{S}$). Rather than investigating this DoF further, this extension focuses on the procedure for a robust filtering. When applied to data analysis, the Student's t-distribution is usually split into an integration of Gaussian distribution and Gamma distribution ($\mathcal{G}$) [117]. In the realization, if the additional hidden variable U_t is introduced at each sample point to represent the sampled scaling of variance, such as described in Section 5.3, the noise w_t in the j-th model can be described as

$$w_t \sim \int_0^\infty \mathcal{N}(0, \frac{R^{(k)}}{u}) \cdot \mathcal{G}\left(u;\ \frac{\nu}{2}, \frac{\nu}{2}\right) du = \mathcal{N}(0, \frac{R^{(k)}}{U_t^{(k)}}). \tag{5.96}$$

While formulating with VBI, $U_t^{(k)}$ is estimated with the prior of Gamma distribution and the corresponding likelihood. As discussed in previous sections, the posteriori expectation can be derived with the time-variant assumption for U_t:

$$\langle U_t^{(k)} \rangle = \frac{\nu + m}{\nu + (y_t - H^{(k)} \cdot x_t^{(\mathrm{p})})' {R^{(k)}}^{-1} (y_t - H^{(k)} \cdot x_t^{(\mathrm{p})})}, \tag{5.97}$$

where m is the dimension of y_t. Then, the posterior of model identity vector I_t will be able to incorporate $U_t^{(k)}$ as

$$\mathbb{E}_{p(I_t^{(k)}|Y_{1:k})}[I_t^{(k)}] \approx \pi_k^{(\mathrm{f})} \propto \pi_j^{(\mathrm{p})} \cdot \mathcal{N}(y_t;\ H^{(k)} \cdot x_t^{(\mathrm{p})}, \frac{R^{(k)}}{\langle U_t^{(k)} \rangle} + H^{(k)} P_t^{(\mathrm{p})} (H^{(k)})'). \tag{5.98}$$

Since $u_t^{(k)}$ has been introduced, the posterior of the identity vector is also an approximation from $\pi_k^{(\mathrm{f})}$. With above two expectations, the implicit integration of Gaussian and Student's t-distribution can be avoided. The estimation of x_t is then turned out as

$$x_t^{(\mathrm{f})} = P_t^{(\mathrm{f})}\left[P_t^{(\mathrm{p})^{-1}} x_t^{(\mathrm{p})} + \sum_{k=1}^{K} \pi_k^{(\mathrm{f})} \langle u_t^{(k)}\rangle (H^{(k)})' R^{(k)^{-1}} y_t\right],$$
$$P_t^{(\mathrm{f})} = \left[P_t^{(\mathrm{p})^{-1}} + \sum_{k=1}^{K} \pi_k^{(\mathrm{f})} \langle u_t^{(k)}\rangle (H^{(k)})' R^{(k)^{-1}} H^{(k)}\right]^{-1}.$$

5.6.5 Application Examples of Filtering Algorithm

In the first set of simulations, the proposed algorithm is tested through a set of scalar jump Markov linear systems (JMLS). Then, a manoeuvring target tracking example is used for validation in the multi-dimensional case. After that, the strength for a robust estimation will be discussed for the cases with outliers.

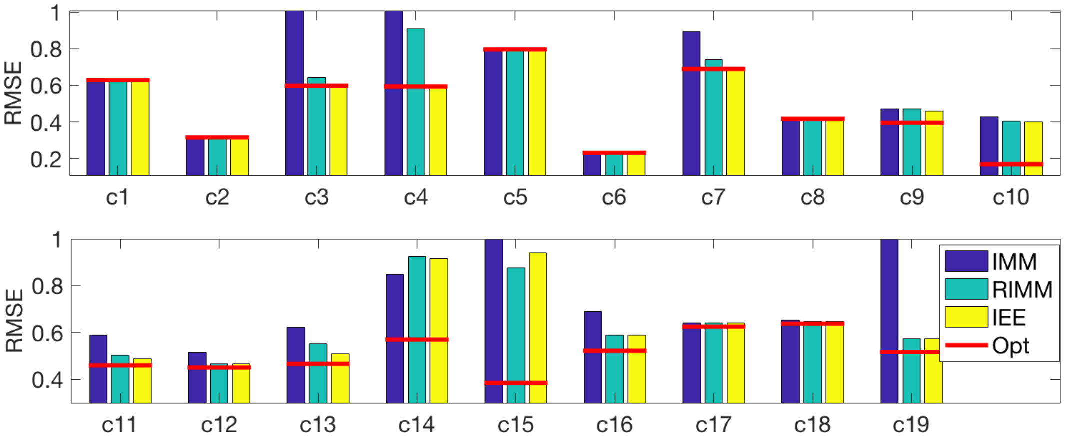


Figure 5.14: Root Mean Squared Error for 19 Examples

In this simulation, examples in [94] are adopted to compare the proposed IEE with IMM as well as a re-weighed IMM algorithm (RIMM) [100]. The switched model identities are generated with transition function according to (5.81), where the transition probabilities are represented as

$$p_{11} = 1 - 1/\tau_1, \quad p_{12} = 1/\tau_1, \quad p_{22} = 1 - 1/\tau_2, \quad p_{21} = 1/\tau_2. \tag{5.99}$$

The states and observations are generated with equation (5.79) and (5.80), where the parameters are assigned as same as in [94] for 19 individual cases. For each case, 100 Monte Carlo tests were performed with randomized initial points generated from $\mathcal{N}(0,1)$. The root mean square errors (RMSE) are compared in Figure 5.14, where the Opt represents the estimate with known model identities. The typical improved performance against IMM and RIMM can be observed for case 3, 4, and 13. These three cases either have a large observation noise or possess an infrequent jump behaviour, indicating that our proposed algorithm has a stronger ability to eliminate observation noises for systems with large inertia (large $\tau_{1:2}$). The step-wise root mean squared errors (RMSE) are compared in Figure 5.15 for this case.

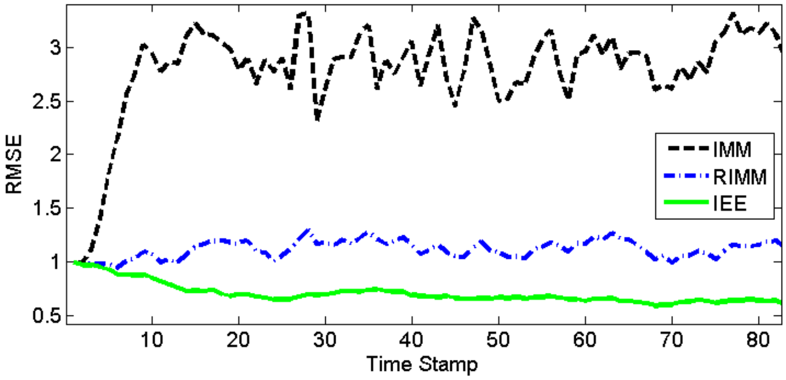


Figure 5.15: Root Mean Squared Error for Example-4

To test the proposed algorithm in multi-dimensional cases, a target tracking problem is adopted from [118]. The target is moving in 2D space, and the positions and velocities forms 4D states, where the positions are measured with noise. Three transition models are used to describe the straight, left-turn, and right-turn moves:

$$F(\omega) = \begin{bmatrix} 1 & 0 & \frac{sin(\omega)}{\omega} & \frac{cos(\omega)-1}{\omega} \\ 0 & 1 & \frac{cos(\omega)-1}{\omega} & \frac{sin(\omega)}{\omega} \\ 0 & 0 & \frac{9cos(\omega)}{10} & -\frac{9sin(\omega)}{10} \\ 0 & 0 & \frac{9sin(\omega)}{10} & \frac{9cos(\omega)}{10} \end{bmatrix}, \qquad T = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$$F^{(1)} = F(\omega \to 0), \qquad F^{(2)} = F(\omega), \qquad F^{(3)} = F(-\omega);$$

$$T^{(1)} = T^{(2)} = T^{(3)} = T, \qquad Q^{(1)} = Q^{(2)} = Q^{(3)} = \sigma_q^2 \cdot \mathbb{I}_2;$$

$$H^{(1)} = H^{(2)} = H^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$R^{(1)} = R^{(2)} = R^{(3)} = \sigma_r^2 \cdot \mathbb{I}_2.$$

In this simulation, a nominal setting of $\{\omega = 0.01; \sigma_q = 0.5; \sigma_r = 5\}$ was utilized, and the transition of model identities in (5.81) is described by matrices tr_1 and tr_2 for two types of dynamics:

$$tr_1 = \begin{bmatrix} 0.99 & 0.05 & 0.05 \\ 0.05 & 0.99 & 0.05 \\ 0.05 & 0.05 & 0.99 \end{bmatrix}, tr_2 = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.8 & 0.1 & 0.1 \\ 0.8 & 0.1 & 0.1 \end{bmatrix}.$$

Case	IMM		RIMM		IEE	
	EP	EV	EP	EV	EP	EV
$tr_1, \omega, \sigma_q, \sigma_r$	5.49	2.38	4.17	1.51	3.53	1.06
$tr_2, \omega, \sigma_q, \sigma_r$	5.45	2.36	3.52	1.06	3.52	1.06
$tr_1, 10\omega, \sigma_q, \sigma_r$	5.49	2.39	4.12	1.48	3.56	1.08
$tr_1, \omega, 2\sigma_q, \sigma_r$	5.49	2.70	4.48	2.21	4.12	1.85
$tr_1, \omega, \sigma_q, 0.6\sigma_r$	3.32	1.57	2.66	1.21	2.38	0.97

Table 5.4: Error from Monte Carlo Tests for the Tracking Problem

Each estimation algorithm was tested 100 times in several different cases around the nominal setting, and the MAE was calculated for the Euclidean distance of position (EP) and velocity (EV) from their true values respectively. The detailed settings and related performance are presented in Table 5.4. It shows that IEE outperformed IMM in all these five cases, and also performs better than RIMM in cases with tr_1 as the transition matrix. Two samples of the tracking results are shown in Figure 5.16 and Figure 5.17. IEE approach provided the closest tracking to the real one, and it was also the smoothest tracking comparing to IMM and RIMM. It can be said that when the target moves with resistance and turns with a certain level of consistency, IEE is expected to estimate more precisely and be less affected by noise and misclassification.

In the last application example, the proposed estimator (IEE) and its extension to Student's t-distribution (IEE-t) are tested for a 2D state estimation. To focus on the extra latent variable in (5.97), the system is built with identical transition models and distinct emission models:

$$F = \begin{bmatrix} 0.95 & 2 \\ 0 & 0.84 \end{bmatrix}, \; T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; Q = \begin{bmatrix} 10 & 0.3 \\ 0.3 & 0.5 \end{bmatrix};$$
$$H^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \; H^{(2)} = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}, \; R^{(1)} = R^{(2)} = 150 \cdot I_2.$$

The outliers are generated from a contaminated Gaussian distribution: $(1-\epsilon)\cdot\mathcal{N}(0,R) + \epsilon\cdot\mathcal{N}(0,k\cdot R)$, where ϵ controls the proportion of outliers and k determines the magnitude of outliers. The DoF of t-distribution could be determined from these parameters, but in this study, small values of ϵ and a fixed value of ν $(=10)$ are used for a basic verification of IEE-t.

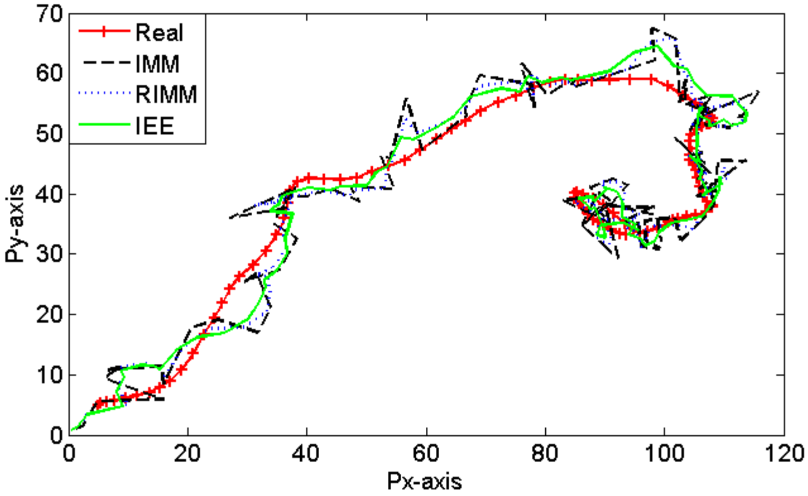


Figure 5.16: Example-1 of Tracking Results

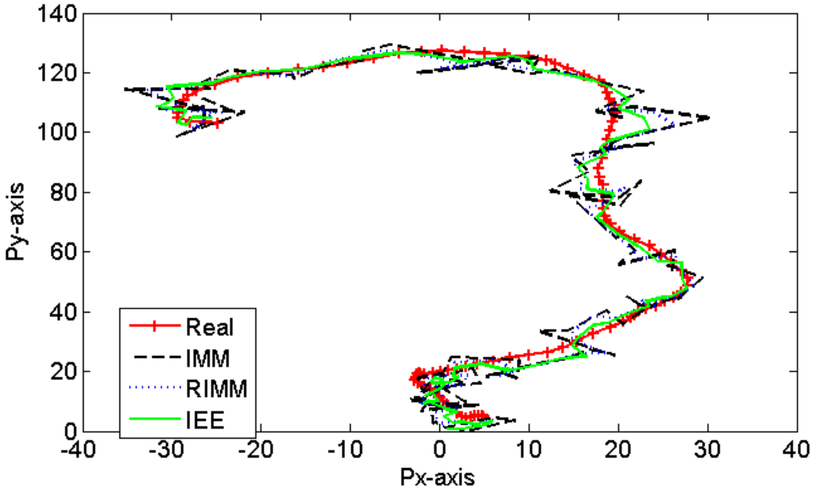


Figure 5.17: Example-2 of Tracking Results

Methods	$\epsilon = 0.05, k = 9$		$\epsilon = 0.1, k = 9$		$\epsilon = 0.1, k = 16$	
	State 1	State 2	State 1	State 2	State 1	State 2
IMM	8.44	1.39	9.62	1.47	11.75	1.68
IEE	7.22	1.15	7.93	1.16	9.08	1.21
IEE-t	7.12	1.14	7.69	1.15	8.31	1.18
ID-KF	6.37	1.13	6.96	1.13	8.09	1.18

Table 5.5: Error from Monte Carlo Tests for Different Outliers

With 200 times of Monte Carlo test, the errors of each state are presented in Table 5.5 for three different outlier settings, where ID-KF is the estimation assuming known true model identities but still unaware of noise contamination. Compared to the regular IEE, introducing t-distribution has improved the performance and becomes more favourable when more severe outliers present. The step-wise RMSE for the third outlier setting is shown in Figure 5.18. It can be observed that ID-KF is not always optimal because of the existence of outliers, and the IEE-t can even provide better performance.

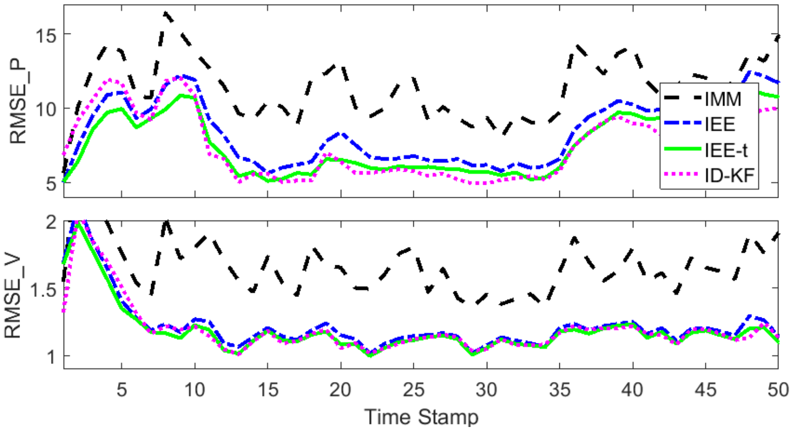


Figure 5.18: Step-wise RMSE of Estimation with Outliers

5.7 Conclusions

In this chapter, a feature extraction method with multiple emission models is developed for inferential sensor modelling. Based on the objective of solving the switching problem in process data analytics, the proposed approach introduced multiple robust models to analyze observations in different operation regions and formulated the latent features with temporal correlation. By implementing a probabilistic formulation, an approximate Bayesian inference algorithm is developed. In particular, a novel variational method is proposed for state updating with multiple emission models along with their uncertainties. Through two simulation examples, the proposed learning methods are shown to have the advantage of automatic structure selection: the latent dimension and the number of active models can be determined according to the variational lower bound. Through the industrial example, an online feature extraction procedure is developed, and the extracted features are demonstrated to be informative for the inferential sensing task. These simulation and industrial application results conclude that our proposed feature extraction has advantages in both learning model identities and predicting the output.

In the extension part of this chapter, the on-line state estimation algorithm is developed for the multiple model state estimation. Based on the objective of minimizing the statistical distance from the exact but implicit posterior distribution, a novel state estimator is designed from a vectorial representation of model identities and the variational learning strategy. The strength of our proposed estimator is verified through benchmark studies for the scalar JMLS and a problem of manoeuvring target tracking. Additionally, the usage of the expected term for intermediate variables, like model identity, is proved to be useful for more complicated models in state estimation. Most significant advantages are found for the system with large inertia and considerable measurement noise.

Chapter 6

Dynamic Latent Feature for Time Delay Problems [*]

The uncertain time delay is a challenging problem in system identification. Accurate estimates of input-output delay can bring significant improvement for system parameter estimation. In this paper, a probabilistic identification framework is developed with consideration of both time-invariant and time-variant time delays. Especially, the autoregressive moving average model with exogenous inputs (ARMAX) is selected as the model structure, and its estimation is conducted under the Bayesian inference framework. Within this framework, a static model, a Markov model, and a hierarchical model are proposed to describe random time delays under different application scenarios. For each scenario, the corresponding learning algorithm is developed based on the variational Bayesian inference, which estimates time delay and model parameters simultaneously. The proposed identification algorithms are validated with numerical simulations along with an industrial application example.

6.1 Introduction

Identification performance of a data-driven model critically depends on how to model the error (the noise). In a static input-output model $y = f(x) + e$, noise e denotes a distance between model output $f(x)$ and actual output y, and samples of e are usually assumed to be independent and identically distributed. With this formulation of the probability distribution, the optimal estimate of $f(\cdot)$ can be obtained by maximizing the likelihood function. Generally, the modelling performance can be improved by a more accurate description of the noise distributions. For example, high-order moments of the noise can be considered when dealing with outliers [119, 120]. When it comes to time series analysis, coloured noise

[*] A version of this chapter will be submitted for a journal publication

ϵ is considered as a more appropriate description for error/disturbance [49], which describes the temporal dependency as the auto-correlation: $\epsilon_t \not\perp \epsilon_{t-1}$. As a typical example [121], the autoregressive moving average model with exogenous inputs (ARMAX) is given as

$$\begin{aligned} &y_t + a_1 \cdot y_{t-1} + ... + a_{n_a} \cdot y_{t-n_a} \\ &= b_1 \cdot u_{t-k} + ... + b_{n_b} \cdot u_{t-k-n_b+1} \\ &\quad + c_0 \cdot e_t + c_1 \cdot e_{t-1} + ... + c_{n_c} \cdot e_{t-n_c}, \end{aligned} \tag{6.1}$$

where the coloured noise is described through a weighted moving average of white noise e. In this dynamic model, an input-output time delay k is also considered which is typically assumed to be a constant. To take the randomness of time delay k into consideration, a probabilistic framework for ARMAX identification should be developed.

6.1.1 Random Time Delays

In model (6.1), k is used to align unsynchronized input-output pairs. Usually, it is considered as a deterministic parameter, where either experimental approaches or data-driven approaches can be used to infer it [122, 123]. In the data-driven approaches, both the prediction error methods [122] and the impulse response methods [124] are used. To accommodate non-linear systems, a sliding mode method has also been developed to identify time delay within certain boundaries [125]. In the on-line identification case, a modified least squares method can be used [126]. As for the state-space formulation, identifiability of the delay parameter has also been investigated [127].

In a more challenging case, unknown time delay may show randomness at different time instants [128, 129]. For example, in some cases where output data are collected from different locations, the time delay is determined by the transportation time of materials [130]. Figure 6.1 illustrates such random time delay, where two different bars represent a time-variant time delay. In other cases where the output variables are measured in an off-line manner, the input-output time delay can be affected by the time spent in laboratory analysis. In these scenarios, to facilitate parameter estimation, time delay should be identified as a sequence of unknown parameters [131, 132].

When being treated as deterministic values, this time delay sequence may be determined by dynamic time warping algorithms [133] in off-line estimation, or a recursive filter [134] in on-line estimation. However, the aforementioned causes of time delay, either due to the uncertain material transportation or due to uncertain off-line analysis delay, suggest it is more appropriate interpreting time delay k as sequenced random variables. Comparing to

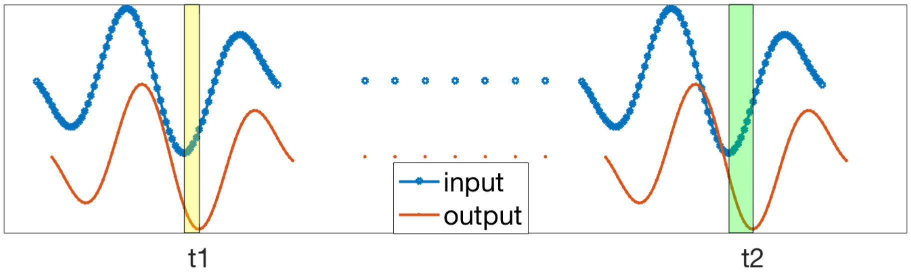


Figure 6.1: Random Time Delays

the previous interpretation as an unknown constant parameter, this treatment does not demand a different set of model parameters for different time delays. More importantly, probability models [23] can be developed to incorporate modelling assumptions to describe the random time delay. For example, the hidden Markov model can be used to describe a dynamically varying time delay [135, 136].

6.1.2 Probabilistic Identification

In terms of the system identification objective, conventional algorithms usually search for the optimal point for each parameter, such as the prediction error methods [122] and the covariance analysis based methods [137]. In other modelling problems, such as regression analysis, probabilistic learning approaches have been widely utilized in the recent decades [38, 28]. Among these approaches, the Bayesian methods extend the point-estimation by introducing the probability distribution for model parameters [23]. Through this probabilistic description, modelling preferences and constraints can be integrated mathematically [38]. For example, the Laplace distribution can be used to describe a sparse parameter set [51], and the Beta distribution can be used to describe constrained parameters [138]. To improve the learning efficiency of Bayesian methods, the variational Bayesian inference [39] has been commonly used. As an emerging learning method [28], it uses an explicit proposal distribution $q(\cdot)$ to approximate the implicit posterior [39], which makes full use of estimation uncertainty in the learning procedure.

For the discrete time delay variables considered in this paper, a vector representation of time delay along with its corresponding probability distribution can be utilized [82, 139]. In this representation, a random vector I_t is used to represent time delay at time t: $I_t = [I_t^{(0)}, I_t^{(1)}, ..., I_t^{(K)}]'$, where each element determines a possible realization of time delay k. While the formal definition will be presented in the next section, we use Figure 6.2 to illustrate this formulation conceptually. Here, $I_t^{(k)}$ determines whether the output sample

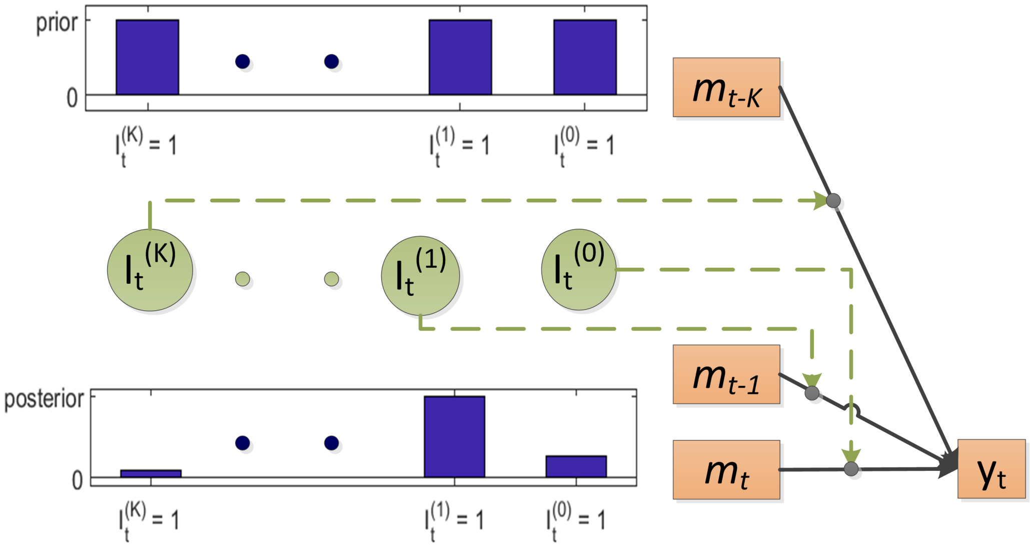


Figure 6.2: Modelling Time Delay with Latent Variable

y_t should be aligned with the prediction $\hat{y}_{t-k}$. In the Bayesian approach, the prior $p(I_t)$ can be updated according to the likelihood of each alignment, and the resultant posterior of I_t can be used to assist identification algorithms [140, 141]. Unlike building multiple models corresponding to different time delays, only one set of model parameters is identified owing to the use of I_t [135, 136].

6.2 Problem Statement

In this study, system identification of the ARMAX model is developed with variational Bayesian inference, where the proposal distribution $q(\cdot)$ will be utilized to approximate exact posterior distributions. Within this framework, three modelling assumptions about those delay variables will be realized, and advanced identification algorithms [141, 140] can be integrated. In this chapter, the proposed time delay formulation and learning algorithms are illustrated for the single-input-single-output ARMAX model in (6.1). It should be noted that there is no significant barrier preventing the extension to non-linear or multivariate processes.

The remainder of this chapter is organized as follows: Section 6.3 describes the proposed probabilistic identification framework for ARMAX modelling. Section 6.4 discusses three modelling assumptions for uncertain time delays and formulates a model for each assumption. Section 6.5 and Section 6.6 illustrates novel learning algorithms for the time-invariant time delay the time-variant time delays, respectively. Besides proper validations for each algorithm, the overall performance is presented with a simulated example and a practical case study in Section 6.8. Concluding remarks are then presented in Section 6.9.

6.3 Bayesian Interpretation of ARMAX Identification

As introduced above, the ARMAX model is a representative example in the system identification algorithms [121]. To investigate random time delay from the perspective of a dynamic latent feature, the likelihood function of the ARMAX model should be first reflected as the corresponding observation function of time delay. Also, in this section, its estimation procedure is formed within the variational Bayesian inference. The proposed structure provides a flexible probabilistic framework for the specific design for time delay modelling, and the proposed inference methods have superior performance to the conventional gradient-based optimization methods in system identification. Its usefulness is demonstrated by comparison with the identification algorithm in MATLAB system identification toolbox.

To have a compact representation for the data samples and the model parameters in (6.1), the following notations are introduced:

$$\psi_t^{(k)} \equiv [y_t, ..., y_{t-n_a}, -u_{t-k}, ..., -u_{t-k-n_b+1}]', \tag{6.2}$$

$$\theta_{ab} \equiv [1, a_1, ..., a_{n_a}, b_1, ..., b_{n_b}]', \tag{6.3}$$

$$\epsilon_t \equiv c_0 \cdot e_t + c_1 \cdot e_{t-1} + ... + c_{n_c}, \tag{6.4}$$

where $\psi_t^{(k)}$ is an "$n_a + n_b + 1$" dimensional column vector stacking all related inputs and outputs, and θ_{ab} is the associated parameter vector. In this section, the models of time delays will not be discussed for the time being and the superscript (k) is temporarily neglected. Thus, the model is now abbreviated as $\theta'_{ab} * \psi_t = \epsilon_t$. The likelihood of training samples $(u_{1:T}, y_{1:T})$ can be formulated through the probability distribution of noise terms $\epsilon_{1:T}$:

$$p(\epsilon_{1:T} \mid \theta_c) = p(y_{1:T} \mid u_{1:T}, \theta_{ab}, \theta_c), \tag{6.5}$$

where $\theta_c = [c_0, ..., c_{n_c}]'$. In the above probability distribution expression for ϵ_t, the likelihood function for the ARMAX identification has been converted to considering a distribution of coloured noises $\epsilon_{1:T}$.

6.3.1 Probability Density of Coloured Noise

Based on the definition in (6.4) and the assumption that $e_{t-n_c:t}$ are independently and identically distributed, ϵ_t is auto-correlated but only correlated with n_c's preceding instants.

If e_t is assumed to have a standard Gaussian distribution, namely $e_t \sim \mathcal{N}(0,1)$, the following statistics can be used for a sufficient description of ϵ_t:

$$r_i \equiv \mathbb{E}[\epsilon_t \ \epsilon_{t-i}] = \begin{cases} \sum_{j=0}^{n_c-i} c_j \cdot c_{j+i}, & \forall \ i \in \{0, ..., n_c\}, \\ 0, & otherwise. \end{cases} \tag{6.6}$$

According to the Gaussian distribution and the linear transformation in (6.4), these 2$^{\text{nd}}$ order moments are sufficient to represent the original noise model parameter θ_c [122].

Based on this finite length of auto-correlation, a proposition is presented here to describe the probability density function of coloured noise.

Proposition 1 *For n_c order colour noise defined in* (6.4)*, the temporal dependence can be modelled as an $n_c + 1$ order Markov sequence:*

$$p(\epsilon_t \mid \epsilon_{t-1}, ..., \epsilon_1) = p(\epsilon_t \mid \epsilon_{t-1}, ..., \epsilon_{t-n_c}). \tag{6.7}$$

Furthermore, by defining adjacent noise samples with $\Upsilon_t \equiv [\epsilon_t, \epsilon_{t-1}, ..., \epsilon_{t-n_c}]'$ and $\bar{\Upsilon}_t \equiv [\epsilon_{t-1}, ..., \epsilon_{t-n_c}]'$ the probability density function of sequenced noises $\epsilon_{1:T}$ is formulated as

$$p(\epsilon_{1:T}) = \prod_{t=n_c+1}^{T} p(\Upsilon_t) \ \ / \prod_{t=n_c+2}^{T} p(\bar{\Upsilon}_t). \tag{6.8}$$

Proof. It can be verified through the Bayes' rule that

$$\begin{aligned} p(\epsilon_{1:T}) &= \prod_{t=2}^{T} p(\epsilon_t \mid \epsilon_{t-1}, ..., \epsilon_1) \\ &= \prod_{t=n_c+1}^{T} p(\epsilon_t \mid \epsilon_{t-1}, ..., \epsilon_{t-n_c}) \cdot p(\epsilon_{n_c}, ..., \epsilon_1). \end{aligned}$$

■

Thus, the distribution of Υ_t and the distribution of $\bar{\Upsilon}_t$ can be used to describe the coloured noise. In this study, these stacked noises follow the multivariate Gaussian distribution: $\Upsilon_t \sim \mathcal{N}(0, R)$ and $\bar{\Upsilon}_t \sim \mathcal{N}(0, \bar{R})$. The covariance matrix R, as well as its principal sub-matrix $\bar{R}$, is a semi-positive definite symmetric Toeplitz matrix, comprising of n_c+1 (or n_c) distinct parameters shown in (6.6):

$$R = Toeplitz(r_0, r_1, ..., r_{n_c}), \tag{6.9}$$

$$\bar{R} = Toeplitz(r_0, r_1, ..., r_{n_c-1}). \tag{6.10}$$

By replacing the original noise parameter θ_c with these covariance matrices, the probability distribution of $\epsilon_{1:T}$ can be formulated through the covariance matrix R. Comparing to using the covariance of the whole sequence, such as $\Omega \equiv cov(Y|\theta)/\lambda^2$ in Complement C7.7 of [122], the usage of R has eliminated those unnecessary zero elements, and only leaves a minimum number of matrix elements in identification algorithms. In addition, it also allows use of other probability distributions to describe Υ_t. In more general cases, where noises are correlated in a non-linear way or distributed with non-Gaussian distribution, the probability distribution for Υ_t is not necessarily multivariate Gaussian distribution. For example, multivariate student-t distributions could be used to help increasing robustness. In this study, we will use this probabilistic description to formulate a likelihood function for the ARMAX model.

6.3.2 Log-Likelihood and Bayesian formulation of ARMAX Model

Similar to the coloured noise, the input-output pairs are also stacked for this likelihood formation:

$$\Psi_t \equiv [\psi_t, ..., \psi_{t-n_c}]', \tag{6.11}$$

$$\bar{\Psi}_t \equiv [\psi_{t-1}, ..., \psi_{t-n_c}]'. \tag{6.12}$$

According to ψ_t in (6.2), eligible samples of Ψ_t are drawn from

$$T_s = n_c + \max\{n_a,\ n_b + k - 1\} + 1. \tag{6.13}$$

Based on Proposition 1, the log-likelihood function of ARMAX model (6.1), with a given time delay k and $e_t \sim \mathcal{N}(0,1)$, can be formed as

$$\begin{aligned}\mathcal{L} &\equiv \ln p(y_{1:T} \mid u_{1:T}, a_1, ..., a_{n_a}, b_1, ..., b_{n_b}, c_0, ..., c_{n_c}) \\ &= \sum_{t=T_s}^{T} \ln p\left(\Upsilon_t \mid R\right) - \sum_{t=T_s+1}^{T} \ln p\left(\bar{\Upsilon}_t \mid \bar{R}\right) \\ &= \sum_{t=T_s}^{T} \ln \mathcal{N}\left(\Psi_t \cdot \theta_{ab};\ 0, R\right) - \sum_{t=T_s+1}^{T} \ln \mathcal{N}\left(\bar{\Psi}_t \cdot \theta_{ab};\ 0, \bar{R}\right).\end{aligned} \tag{6.14}$$

In terms of the model parameters θ_{ab}, this log-likelihood is a quadratic function. Thus, conventional challenges from the non-linearity is converted to the challenges in the estimation of covariance matrix R. It can be seen that $\bar{R}$ is part of R.

In Figure 6.3, one example of $\mathcal{L}$ is plotted as a function of two-dimensional R from the definition (6.9). Based on this formulation of log-likelihood, the covariance parameters, for example $\{r_0, r_1\}$, can be optimized by using partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial r_i} = -\frac{T - T_s + 1}{2} tr\{R^{-1} \cdot \frac{\partial R}{\partial r_i}\} + \frac{T - T_s}{2} tr\{\bar{R}^{-1} \cdot \frac{\partial \bar{R}}{\partial r_i}\}$$

$$-\frac{1}{2}tr\{-R^{-1}\sum_{t=T_s}^{T}\Upsilon_t\Upsilon_t'\ R^{-1}\cdot\frac{\partial R}{\partial r_i}\}+\frac{1}{2}tr\{-\bar{R}^{-1}\sum_{t=T_s+1}^{T}\bar{\Upsilon}_t\bar{\Upsilon}_t'\ \bar{R}^{-1}\cdot\frac{\partial \bar{R}}{\partial r_i}\}, \tag{6.15}$$

where $tr\{\cdot\}$ is the trace operator for a matrix. Υ_t and $\bar{\Upsilon}_t$ are defined in Theorem 1. The matrices $\partial R/\partial r_i$ and $\partial \bar{R}/\partial r_i$ contain binary elements of $\{0,1\}$, indicating the position of r_i in corresponding matrix. For example, $\partial \bar{R}/\partial r_0$ is the n_c-dimensional identity matrix. Thus, the derivative based algorithm can be realized for searching the optimal point of $[r_0, r_1, ..., r_{n_c}]$. An feasible initial point can be selected as the identity matrix as the rule-of-thumb, representing the prior assumption about white noise. From the optimized covariance parameters, the original model parameters can be recovered uniquely according to the n_c+1 equations from (6.6). Thus, a deterministic optimization becomes a feasible solution to the maximum likelihood estimation of the ARMAX model.

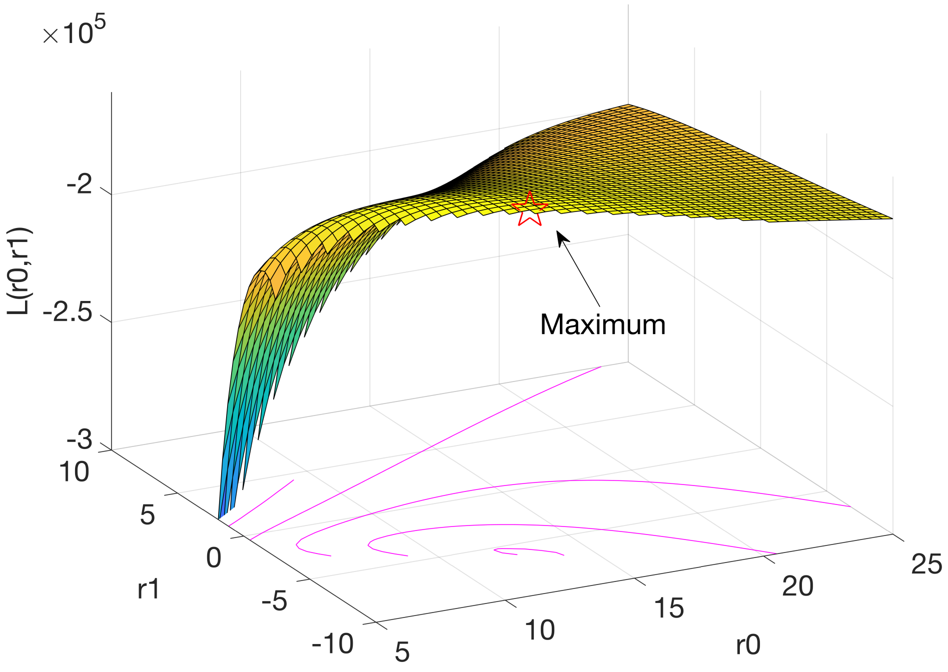


Figure 6.3: Log-Likelihood Function of ARMAX Covariance Parameters

However, for our probabilistic modelling within the Bayesian framework, the prior distributions are needed. As for the prior of θ_{ab}, to deal with this quadratic log-likelihood function, the Gaussian distribution can be used as a conjugate prior in the variational Bayesian inference:

$$p(\theta_{ab}) = p(a_1, ..., a_{n_a}, b_1, ..., b_{n_b}) = \mathcal{N}(0, P_{ab}). \tag{6.16}$$

In this distribution, P_{ab} as an $n_a + n_b$ dimensional covariance matrix can be assigned with large magnitude to represent a non-informative prior.

On the other hand, selecting a distribution for the covariance matrix R is not straightforward. The specific properties about R – symmetric, semi-positive definite, Toeplitz – make the common matrix distributions unsuitable. For example, Inverse-Wishart distribution violates the definition of a Toeplitz matrix. Also, other matrix decomposition techniques will introduce more additional independent parameters, which reduces learning efficiency. In this study, within the variational Bayesian inference, a particle-based representation of $p(R)$ is proposed as

$$\begin{aligned} p(R) &= p(r_0 \cdot R_0) = p(r_0) \cdot P(R_0) \\ &= \mathcal{G}^{-1}(r_0;\ \alpha_{r_0}, \beta_{r_0}) \cdot \mathcal{U}_{n_c}(R_0;\ -1, 1). \end{aligned} \tag{6.17}$$

The first element (the scale variable) r_0 is described with Inverse-Gamma distribution $\mathcal{G}^{-1}$, and the normalized correlation matrix $R_0 = R/r_0$ is described through a particle generating procedure $\mathcal{U}_{n_c}$, where the Uniform distribution $\mathcal{U}(-1,1)$ is used. Here, the specific sampling algorithm [142] of $R_0 \sim \mathcal{U}_{n_c}(-1,1)$ serves as a prior for the following Bayesian inference.

Algorithm 1 Sample (n_c+1)-dim Correlation Matrix

```
for $i = 1$ to $n_c$ do
    $\varphi_{i,i} = \alpha_i \sim \mathcal{U}(-1,1)$;
    for $j = 1$ to $i - 1$ do
        $\varphi_{i,j} = \varphi_{i-1,j} - \alpha_i * \varphi_{i-1,i-j}$;
    $\rho_i = \sum_{j=1}^{i} \varphi_{i,j} \cdot \rho_{i-j}$;    $\rho_0 = 1$;
return $R_0 \leftarrow Toeplitz(1, \rho_1, ..., \rho_{n_c})$;
```

6.3.3 Identification with Variational Bayesian Inference

To conduct approximate inference, the parametrization of proposal distributions should be clarified. Based on the probabilistic conjugacy [38], the proposal distribution of θ_{ab} can be chosen as

$$q(\theta_{ab}) = \mathcal{N}(\theta_{ab};\ \hat{m}_{ab}, \hat{P}_{ab}), \tag{6.18}$$

where $\hat{m}_{ab}$ and $\hat{P}_{ab}$ are hyper-parameters in this proposal distribution. With particles of R_0 being generated from $\mathcal{U}_{n_c}(R_0)$, the log-likelihood (6.14) can provide the corresponding weights for these particles. Thus, based on the concept of importance sampling, the approximated posterior is represented by N pairs of sample $R_0^{(i)}$ and weight $\hat{w}^{(i)}$:

$$q(R) \approx \mathcal{G}^{-1}(r_0;\ \hat{\alpha}_{r_0}, \hat{\beta}_{r_0}) \sum_{i=1}^{N} \hat{w}^{(i)} \delta(R_0 - R_0^{(i)}). \tag{6.19}$$

With sufficient number of particles, $q(R)$ can be well represented by the particles [142]. where r_0 still follows the Inverse-Gamma distribution because of conjugacy, and $\delta(\cdot)$ stands for the Dirac delta function.

The learning objective is then formulated through an approximation of posterior:

$$q(\theta_{ab}) \cdot q(R) \rightarrow p(\theta_{ab}, R \mid u_{1:T}, y_{1:T}). \tag{6.20}$$

Instead of searching the posterior directly, the proposal distributions, specifically their hyper-parameters, can be manipulated to minimize the statistical difference between the two sides of (6.20). According to the variational Bayesian inference, the best estimation can be obtained when the Kullback-Leibler (K-L) divergence is minimized:

$$\min_{q(\theta_{ab})q(R)} D_{KL}\{q(\theta_{ab})\ q(R) \parallel p(\theta_{ab}, R \mid u_{1:T}, y_{1:T})\}. \tag{6.21}$$

With the use of factorized and parametrised proposal distribution in (6.20), the inference problem becomes tractable.

Based on the multivariate Normal distribution adopted in the objective $\mathcal{L}$, two key intermediate statistics are crucial for solving this learning problem. The first one is the trace function for evaluating covariance parameters:

$$\begin{aligned} f_{tr}(R_0 \mid q(\theta_{ab})) = tr\{R_0^{-1} \sum_{t=T_s}^{T} \Psi_t \langle \theta_{ab} \cdot \theta'_{ab} \rangle \Psi'_t\} \\ - tr\{\bar{R}_0^{-1} \sum_{t=T_s+1}^{T} \bar{\Psi}_t \langle \theta_{ab} \cdot \theta'_{ab} \rangle \bar{\Psi}'_t\}. \end{aligned} \tag{6.22}$$

It depends on the proposal distribution $q(\theta_{ab})$ in (6.18), where $\langle \theta_{ab} \cdot \theta'_{ab} \rangle$ is achieved as the second moment from this Normal distribution. The second one is about an expectation matrix for updating model parameters:

$$\begin{bmatrix} \langle \lambda_0 \rangle & \langle m_\theta \rangle' \\ \langle m_\theta \rangle & \langle \Lambda_\theta \rangle \end{bmatrix} = \sum_{t=T_s}^{T} \Psi'_t \langle R^{-1} \rangle \Psi_t - \sum_{t=T_s+1}^{T} \bar{\Psi}'_t \langle \bar{R}^{-1} \rangle \bar{\Psi}_t. \tag{6.23}$$

It depends on the proposal distribution $q(R)$ in (6.19). where $\langle R^{-1} \rangle$ and $\langle \bar{R}^{-1} \rangle$ are obtained from this particle based distribution.

Thus, the detailed variational updating steps for the proposed objective (6.21) can be realized. Based on the step-wise KL divergence, the optimal $q^*(\theta_{ab})$ is obtained as

$$\begin{aligned} \ln q^*(\theta_{ab}) &= \langle \ln p(y_{1:T}, \theta_{ab}, R \mid u_{1:T}) \rangle + const. \\ &= \langle \mathcal{L} \rangle + \ln p(\theta_{ab}) + const. \\ \Rightarrow \ \hat{m}^*_{ab} &= -\hat{P}^*_{ab} \cdot \langle m_\theta \rangle; \quad \hat{P}^*_{ab} = (P^{-1}_{ab} + \langle \Lambda_\theta \rangle)^{-1}, \end{aligned} \tag{6.24}$$

where $\mathcal{L}$ is obtained from (6.14). Based on the quadratic form of $\mathcal{L}$ and Gaussian distributions for both prior and proposal distribution, the variational parameters can then be determined uniquely. The necessary statistics are obtained as

$$\langle \theta_{ab} \theta'_{ab} \rangle = \begin{bmatrix} 1 & \hat{m}'_{ab} \\ \hat{m}_{ab} & \hat{m}_{ab} \cdot \hat{m}'_{ab} + \hat{P}_{ab} \end{bmatrix}. \tag{6.25}$$

Similarly, the updating equation for $q(R)$ is obtained as

$$\begin{aligned} &\ln q^*(R) = \langle \mathcal{L} \rangle + \ln p(r_0) + \ln p(R_0) + const. \\ \Rightarrow \ &\hat{\alpha}_{r_0} = \alpha_{r_0} + (T - T_s + n_c + 1)/2, && (6.26) \\ &\hat{\beta}_{r_0} = \beta_{r_0} + \frac{1}{2} \sum_{i=1}^{N} \hat{w}^{(i)} \cdot f_{tr}(R_0^{(i)}), && (6.27) \\ &\ln \hat{w}^{(i)} = -\frac{T - T_s + 1}{2} \ln |R_0^{(i)}| + \frac{T - T_s}{2} \ln |\bar{R}_0^{(i)}| - \frac{\hat{\alpha}_{r_0}}{2\hat{\beta}_{r_0}} \cdot f_{tr}(R_0^{(i)}). && (6.28) \end{aligned}$$

Essentially, r_0 and R_0 are assumed to be independent in the proposal distribution, which makes it possible to update them separately. The necessary statistics for other variational updating steps can be obtained as follows:

$$\begin{aligned} \langle R^{-1} \rangle &= \hat{\alpha}/\hat{\beta} \cdot \sum_{i=1}^{N} \hat{w}^{(i)} \cdot [R_0^{(i)}]^{-1}, \\ \langle \bar{R}^{-1} \rangle &= \hat{\alpha}/\hat{\beta} \cdot \sum_{i=1}^{N} \hat{w}^{(i)} \cdot [\bar{R}_0^{(i)}]^{-1}. \end{aligned} \tag{6.29}$$

Especially, in terms of the particle-based representation of the covariance parameter R, one empirical re-sampling algorithm is also proposed to increase the efficiency of importance sampling. It has been illustrated in Algorithm 2. In practice, it can concentrate particles around the interested area, and thus forms a better sample distribution.

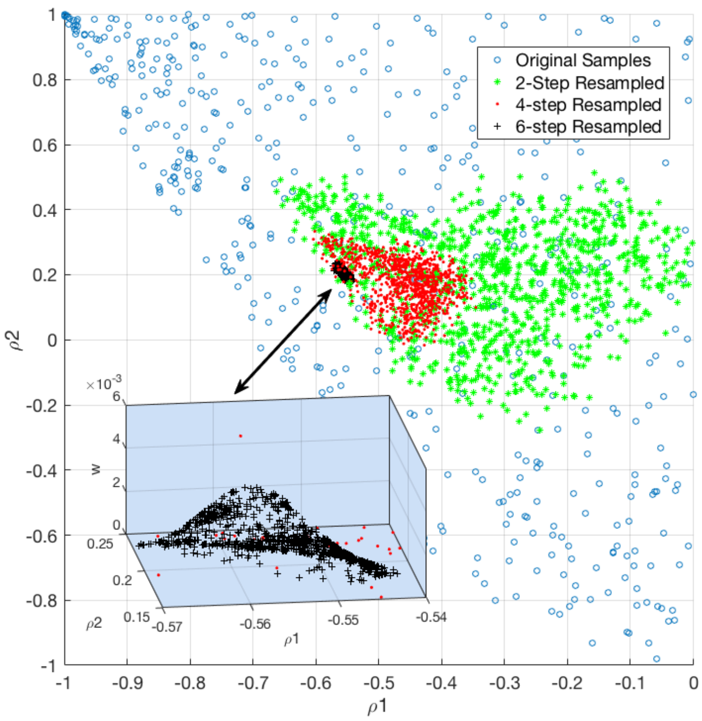


Figure 6.4: Estimating Covariance Parameters through Re-Sampling Methods

Algorithm 2 Re-Sampling Correlation Matrices

```
data particles $p^{(1:N)}$ and weights $w^{(1:N)}$;
require $\alpha_p$ for $w_{th}$ : $\sum_{i=1}^{N} 1(w^{(i)} < w_{th}) = \alpha_p \cdot N$;
for $i = 1$ to $N$ do
    if $w^{(i)} < w_{th}$ then
        $p_{good} \leftarrow rand\{p^{(j)} : w^{(j)} > w_{th}\}$;
        $p^{(i)} \leftarrow \alpha_p \cdot p^{(i)} + (1 - \alpha_p) \cdot p_{good}$;
    else
        $p^{(i)} \leftarrow p^{(i)}$;
```

A validation example is shown in Figure 6.4 for two-dimensional $R_0 = Toeplitz(\rho_0, \rho_1)$. The initial particles (blue circles) are distributed sparsely according to $\mathcal{U}_{n_c}$, but the re-sampled ones are gradually concentrated on the high probability area. Typically, as shown in the sub-figure, particles from the 6th re-sampling step (black crosses) have distributed around the "useful" area, and present a granular shape for $q(R)$.

In order to recover the original "C-polynomial" parameters in (6.4), the mean estimate from these particles: $\langle R \rangle$, is used to solve equations in (6.6). Numerical simulations have been conducted to validate this procedure with both methods ($\mathcal{M}$) including the partial

Table 6.1: Estimate C Parameters with Different Orders

n_c^*	$\mathcal{M}$	$n_c = 2$	$n_c = 3$
1	Opt	$2.97, -1.91, 0.01$	$2.99, -1.87, 0.01, 0.06$
	Sam	$2.98, -1.87, 0.02$	$3.00, -1.85, 0.01, 0.06$
2	Opt	$2.98, -1.92, 0.92$	$2.96, -1.92, 0.99, 0.08$
	Sam	$2.99, -1.86, 0.89$	$2.97, -1.87, 0.96, 0.09$
3	Opt	$5.74, -4.29, 5.74$	$2.98, -1.90, 0.97, -0.41$
	Sam	$2.92, -1.85, 1.21$	$2.98, -1.89, 0.97, -0.44$

derivative based optimization (Opt) and the proposed sampling algorithm (Sam). Table 6.1 shows a comparison result for the second order case and the third order case. The true noise is generated for $n_c^* \in \{1, 2, 3\}$ cases by using the true parameters $c^* = [3, -2]$, $c^* = [3, -2, 1]$, and $c^* = [3, -2, 1, -0.5]$, respectively. In the inference procedure, the noise polynomial order is assumed to be $n_c = 2$ or $n_c = 3$. From the result in Table 6.1, it can be observed that both methods can provide accurate results when the assumed noise order is equal to or higher than the true order. However, as for the case when the assumed order is lower than the true one, the sampling based method shows more robustness.

In addition to validating the identification algorithm for "C-polynomial" parameters, the advantage of proposed probabilistic identification of the ARMAX model is compared with the identification toolbox in MATLAB. A benchmark Stirred Tank Heater simulation is used to generate data, using a Random Binary Signal at the steam valve as the input and collecting temperature data as the output. With a given set-point of water level, 1000 samples are collected as the training data and 500 samples as the validation data. For a fair comparison, our proposed algorithm (VI-ID) is compared with the MATLAB built-in function for the ARMAX model (denoted as MATLAB) for a given time delay value (k^*=3). The performance is evaluated for several possible combinations of $\{n_a, n_b, n_c\}$. The "goodness-of-fit" is calculated by using the normalized root mean square error between the actual output y and the ∞-step prediction $\hat{y}$:

$$Fit\% = 1 - \frac{\|y - \hat{y}\|_2}{\|y - mean(y)\|_2}.$$

The result is shown in Figure 6.5, where the sufficiency of our proposed identification algorithm is validated. Comparing to the existing method, VI-ID also has a wider range of good performance, which indicates that by applying the proposed probabilistic algorithm, identification results show less sensitive to variations in structural parameters n_a and n_b.

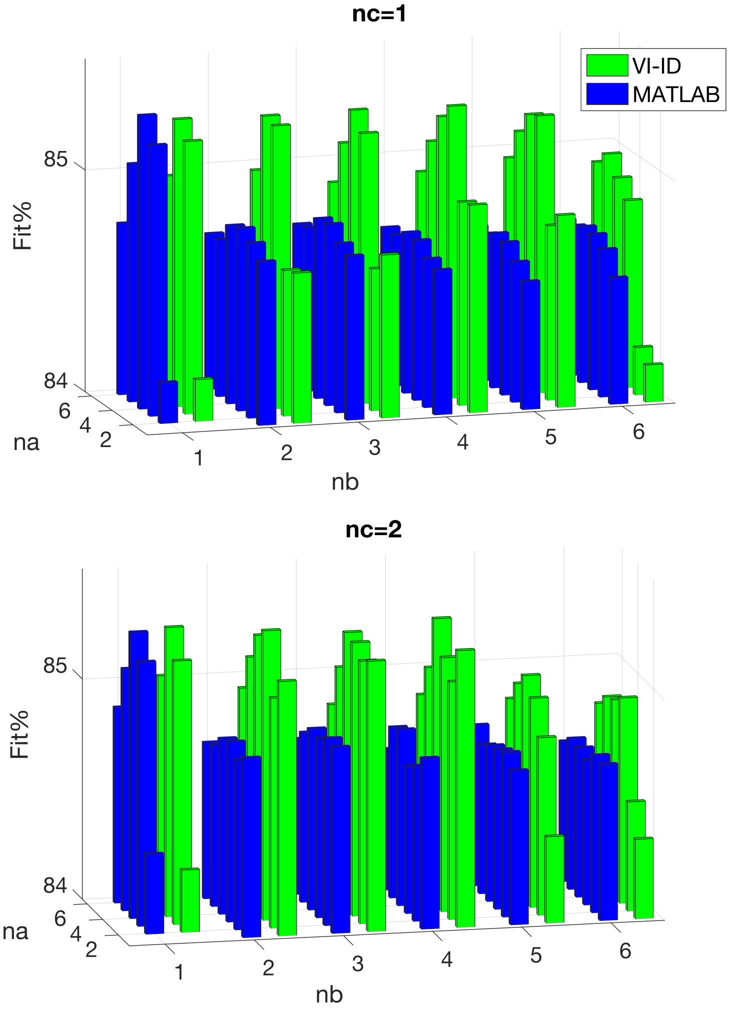


Figure 6.5: Validating Identification Performance of Bayesian ARMAX

6.4 Probabilistic Formulation of Time-Invariant Time Delays

The proposed probabilistic framework has provided a solid identification performance, making itself a good foundation for the probabilistic modelling of ARMAX with random time delay k. Besides the previous introductory description about the lag variable I_t, here we present a formal definition for time delay variables and their functionality.

Definition 2 *For the bounded integer time delay parameter, $k \in \{0, ..., K\}$, in the discrete-time ARMAX model* (6.1)*, a lag vector I_t with K+1 binary elements:*

$$I_t = [I_t^{(0)}, I_t^{(1)}, ..., I_t^{(K)}]', \quad \forall\ k:\ I_t^{(k)} \in \{0, 1\}, \tag{6.30}$$

can be defined with a constraint that $\|I_t\|_0 = 1$. By using I_t to replace k in (6.2)*:*

$$\psi_t = \psi_t^{(k)} \Leftrightarrow I_t^{(k)} = 1, \tag{6.31}$$

the dependency between the noise ϵ_t and this lag vector I_t is then formulated as

$$p(\epsilon_t \mid I_t) = \prod_{k=0}^{K} \left[p(\theta'_{ab} \cdot \psi_t^{(k)}) \right]^{I_t^{(k)}}. \tag{6.32}$$

In this definition, $\|\cdot\|_0$ stands for l_0 norm, indicating the number of non-zero elements. It can be verified that I_t is only defined at K+1 possible values, which is consistent with the original time delay k. By using this lag vector, the time delay is introduced with the probability dependency, as shown in (6.32), which facilitates the probabilistic identification of the ARMAX model.

The motivation of this lag vector representation is from the study on multi-modes observations, which has been presented in Chapter 5. Instead of being used as weighting coefficients for all possible cases, this vectorial realization of time delay allows more effective development of probabilistic algorithms. Technically, the location of the exponential term of original probability can simplify the derivation, which becomes the summation of coefficients in the log-likelihood term.

Based on its discrete support domain, the Multinomial distribution *Mul* is usually used to describe I_t:

$$p(I_t \mid \pi) = Mul(I_t;\ \pi) = \prod_{k=0}^{K} [\pi^{(k)}]^{I_t^{(k)}}, \tag{6.33}$$

where the parameter $\pi = [\pi^{(0)}, \pi^{(1)}, ..., \pi^{(K)}]'$ is a sum-one vector with non-negative elements:

$$\textstyle\prod_{k=0}^{K} [\pi^{(k)}] = 1, \quad \forall\ k \in \{0, ..., K\}:\ \pi^{(k)} \geq 0.$$

This parameter actually represents the occurrence possibility of each delay value:

$$\Pr[I_t^{(k)} = 1] = \pi_t^{(k)}.$$

Based on this parametrization, the prior knowledge of I_t can be imposed through a parameter vector π_0, and its updated posterior can be described by a specified parameter vector $\hat{\pi}_t$, such as those visualized by the blue bars in Figure 6.2.

However, for the sequenced input-output pairs, the proposal distribution should be defined on the sequenced $I_{1:T}$:

$$q(I_{1:T}) \to p(I_{1:T} \mid y_{1:T}, u_{1:T}).$$

The formation for $q(I_{1:T})$ as well as for $p(I_{1:T})$ depends on the modelling assumptions and application scenarios. In this study, three probability models are used to describe the behaviour of time delays in distinct conditions. As for this section, the probability distribution for a time-invariant time delay is illustrated, and one example application of this case is provided to illustrate probabilistic ARMAX modelling with time-invariant time delay.

6.4.1 Modelling Assumption

This basic case deals with the static time delay, by assuming that all lag variables share the same value. Under the proposed probabilistic framework, this condition is presented by a single delay vector I_{n_k}. Consequently, the parametrization of prior distribution $p(I_{1:T})$ becomes

$$\begin{aligned} & I_1 = I_2 = \cdots = I_T = I_{n_k} \\ \Rightarrow\ & p(I_{1:T}) = p(I_{n_k}) = Mul(I_{n_k};\ \pi_0). \end{aligned} \tag{6.34}$$

With the use of the elementary likelihood defined in Definition 2, the variational updating procedure will result in a conjugate posterior, for which the proposal distribution $q(I_{1:T})$ is formed as

$$q(I_{1:T}) = q(I_{n_k}) = Mul(I_{n_k};\ \hat{\pi}). \tag{6.35}$$

Based on such parametrization, each pairing possibility introduced in Figure 6.2 can now be summarized by the vector $\hat{\pi}$. The probability graphical model of this assumption is shown in Figure 6.6, where green circles denote random variables and blue diamonds are their parameters to be estimated. After a joint learning procedure for both model parameters and the lag variables, the most likely time delay value can be taken from the converged $\hat{\pi}$.

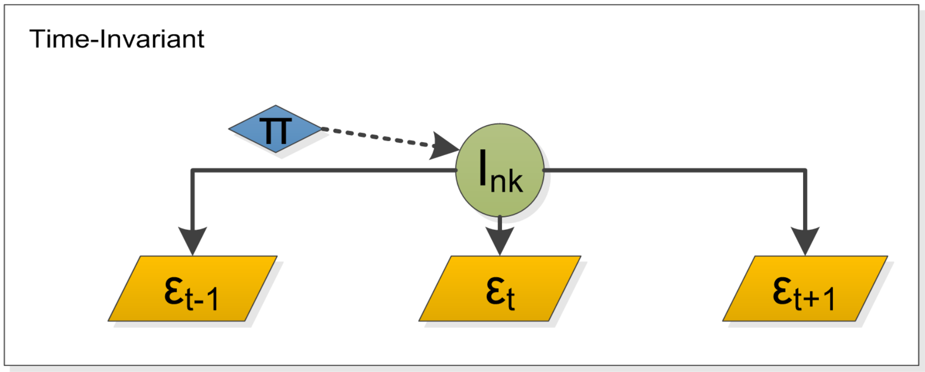


Figure 6.6: Modelling Assumptions for Time-Invariant Time Delays

6.4.2 Variational Inference

Before the discussion about identification procedure for time-variant cases, the time-delay model in Figure 6.6 should be connected with probabilistic ARMAX model. Since all the lag vectors share a same value, both data vector ψ_t and stacked data matrix Ψ_t have $K+1$ different time delay realizations:

$$\Psi_t = \Psi_t^{(k)} = [\psi_t^{(k)}, \psi_{t-1}^{(k)}, ..., \psi_{t-n_c}^{(k)}]' \iff I_{n_k}^{(k)} = 1,$$

where the superscript (k) denotes delay unit. From Theorem 2, the conditional probability for the noise vectors becomes

$$p\left(\Upsilon_t \mid I_{n_k}, R\right) = \prod_{k=0}^{K} \left[\mathcal{N}(\Psi_t^{(k)} \cdot \theta_{ab};\ 0, R)\right]^{I_{n_k}^{(k)}}, \tag{6.36}$$

$$p\left(\bar{\Upsilon}_t \mid I_{n_k}, \bar{R}\right) = \prod_{k=0}^{K} \left[\mathcal{N}(\bar{\Psi}_t^{(k)} \cdot \theta_{ab};\ 0, \bar{R})\right]^{I_{n_k}^{(k)}}. \tag{6.37}$$

Thus, the original log-likelihood in (6.14) can be extended with the condition on I_{n_k}, and the log-likelihood with time-invariant time delay becomes

$$\begin{aligned} \mathcal{L}_{TI} &\equiv \ln p(y_{1:T} \mid I_{n_k}, u_{1:T}, \theta_{ab}, R) \\ &= \sum_{t=T_s}^{T} \ln p\left(\Upsilon_t \middle| \, I_{n_k}, R\right) - \sum_{t=T_s+1}^{T} \ln p\left(\bar{\Upsilon}_t \middle| \, I_{n_k}, \bar{R}\right) \\ &= \sum_{k=0}^{K} I_{n_k}^{(k)} \cdot \left[\sum_{t=T_s}^{T} \ln \mathcal{N}(\Psi_t^{(k)} \cdot \theta_{ab};\ 0, R) - \sum_{t=T_s+1}^{T} \ln \mathcal{N}(\bar{\Psi}_t^{(k)} \cdot \theta_{ab};\ 0, \bar{R}) \right]. \end{aligned} \tag{6.38}$$

It can be observed that the elements of lag vector I_{n_k} appear as the summation coefficients before $\ln \mathcal{N}$ terms, which facilitates the learning procedure of I_{n_k}:

$$\ln q^*(I_{n_k}) = \langle \mathcal{L}_{TI} \rangle_{(\theta_{ab}, R)} + \ln p(I_{n_k}) + const.$$

$$\Rightarrow \quad \ln \hat{\pi}^{(k)} = \ln \pi_0^{(k)} - \frac{1}{2} tr\{(\sum_{t=T_s}^{T} \Psi_t^{(k)'} \langle R^{-1} \rangle \Psi_t^{(k)} - \sum_{t=T_s+1}^{T} \bar{\Psi}_t^{(k)'} \langle \bar{R}^{-1} \rangle \bar{\Psi}_t^{(k)}) \langle \theta_{ab} \theta'_{ab} \rangle\}, \tag{6.39}$$

where $\langle \cdot \rangle$ denotes the expectation operator with respect to $q(\theta_{ab})$ and $q(R)$.

As presented in the previous section, learning θ_{ab} and R only depends on two key intermediate statistics in (6.23) and (6.22). With an unknown but time-invariant time delay, these two statistics shall be revised by considering a dependency on $q(I_{n_k})$:

$$f_{tr}^{(TI)}(R_0) = tr\{R_0^{-1} \cdot \sum_{k=0}^{K} \langle I_{n_k}^{(k)} \rangle \sum_{t=T_s}^{T} \Psi_t \langle \theta_{ab} \theta'_{ab} \rangle \Psi'_t\} - tr\{\bar{R}_0^{-1} \cdot \sum_{k=0}^{K} \langle I_{n_k}^{(k)} \rangle \sum_{t=T_s+1}^{T} \bar{\Psi}_t \langle \theta_{ab} \theta'_{ab} \rangle \bar{\Psi}'_t\},$$

$$\begin{bmatrix} \langle \lambda_0 \rangle & \langle m_\theta \rangle' \\ \langle m_\theta \rangle & \langle \Lambda_\theta \rangle \end{bmatrix}^{(TI)} = \sum_{k=0}^{K} \langle I_{n_k}^{(k)} \rangle \{ \sum_{t=T_s}^{T} \Psi'_t \langle R^{-1} \rangle \Psi_t - \sum_{t=T_s+1}^{T} \bar{\Psi}'_t \langle \bar{R}^{-1} \rangle \bar{\Psi}_t \},$$

where $\langle I_{n_k}^{(k)} \rangle$ is calculated from the Multinomial parametrization in (6.33). Actually, above two equations have revealed the essential advantage of introducing a probability modelling for time delay: learning model parameters with a consideration about the time delay uncertainty. In each updating step, $\langle I_{n_k}^{(k)} \rangle$ is obtained from the latest estimation of time delay. With these additional weighting factors, use of the mode and uncertainty of time delay can make improvements for optimization effectiveness [39].

For validation, this inference algorithm is tested by the aforementioned Stirred Tank Heater simulation example. As indicated by Figure 6.5, a set of structural parameters $[n_a, n_b, n_c]$ is selected as $[3, 1, 1]$. In Figure 6.7, the learning procedure for the case $K = 9$ has been shown with the first three evolutions of $\hat{\pi}$ and $q(b_1)$. It can be observed that the mode of $\hat{\pi}$ has quickly converged to the optimal time delay $k = 3$, and the proposal distribution of b_1 has evolved correspondingly.

In addition to the use of the non-informative initial guess, a wrong initial guess ($k = 8$) is used deliberately for a robustness test. Similar plots are shown in Figure 6.8, where $q(b_1)$ was deviated initially but was gradually corrected along with the convergence of $\hat{\pi}$. Thus, it has been validated that by considering the randomness of time-invariant time delay, the proposed algorithm can simultaneously learn the parameters and unknown time delay k.

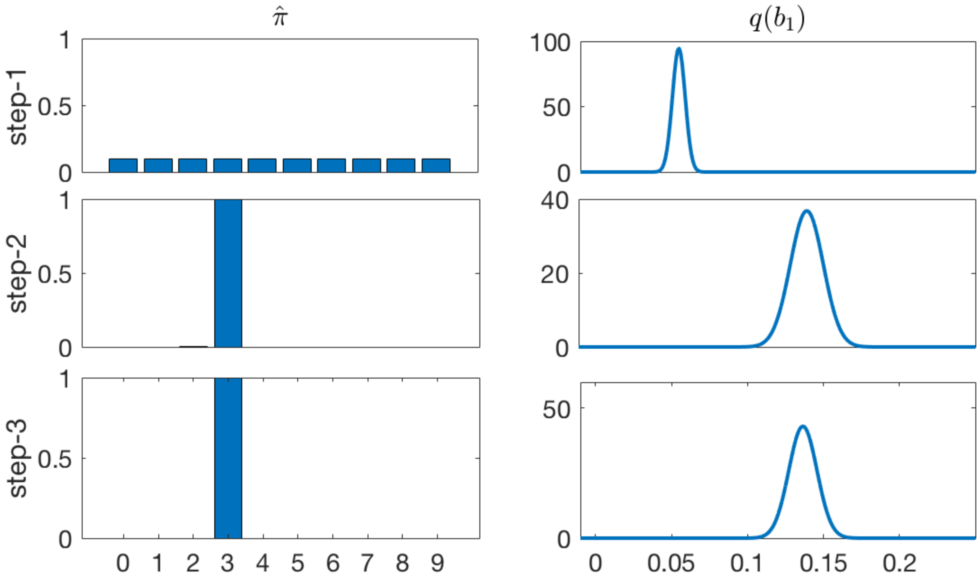


Figure 6.7: Learning Time Invariant Delay with Trivial Initial

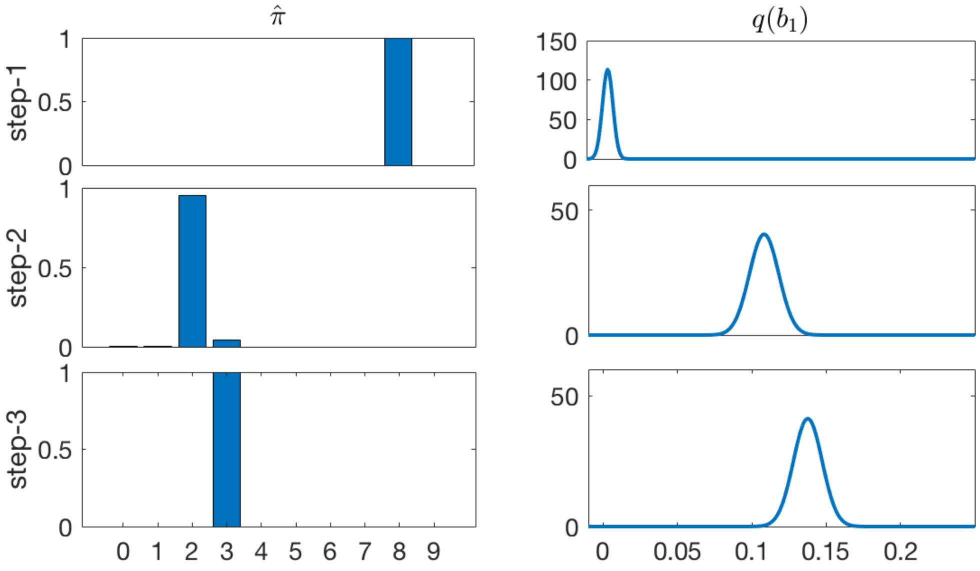


Figure 6.8: Learning Time Invariant Delay with Wrong Initial

6.5 Probabilistic Formulation of Independent Time-Variant Time Delays

Similar as in the previous discussion for the time delay inference, the connection between time delay models and probabilistic ARMAX model should be elaborated. As for the ARMAX modelling with time-variant time delays, the stacked data matrix will be associated with several time delays, making the indexing as

$$\Psi_t^{(k_0,\dots,k_{n_c})} \equiv [\psi_t^{(k_0)}, \psi_{t-1}^{(k_1)}, \dots, \psi_{t-n_c}^{(k_{n_c})}]'.$$

Therefore, the noise vectors Υ_t and $\bar{\Upsilon}_t$ also depend on several lag vectors. The log-likelihood of ARMAX model with time-variant time delays is thus formulated as

$$\begin{aligned}\mathcal{L}_{TV} &\equiv \ln p(y_{1:T} \mid I_{1:T}, u_{1:T}, \theta_{ab}, R) \\ &= \sum_{t=T_s}^{T} \ln p\left(\Upsilon_t \mid I_{t:t-n_c}, R\right) - \sum_{t=T_s+1}^{T} \ln p\left(\bar{\Upsilon}_t \mid I_{t-1:t-n_c}, \bar{R}\right),\end{aligned} \tag{6.40}$$

where the calculation of each conditional distribution of noise vectors requires an enumeration of all possible delays:

$$p(\Upsilon_t \mid I_{t:t-n_c}, R) = \prod_{k_0=0}^{K} \cdots \prod_{k_{n_c}=0}^{K} \left[\mathcal{N}(\Psi_t^{(k_0,\dots,k_{n_c})} \cdot \theta_{ab};\ \vec{0}, R)\right]^{I_t^{(k_0)} \cdots I_{t-n_c}^{(k_{n_c})}}, \tag{6.41}$$

$$p(\bar{\Upsilon}_t \mid I_{t-1:t-n_c}, \bar{R}) = \prod_{k_1=0}^{K} \cdots \prod_{k_{n_c}=0}^{K} \left[\mathcal{N}(\bar{\Psi}_t^{(k_1,\dots,k_{n_c})} \cdot \theta_{ab};\ \vec{0}, \bar{R})\right]^{I_{t-1}^{(k_1)} \cdots I_{t-n_c}^{(k_{n_c})}}. \tag{6.42}$$

Based on this log-likelihood, the remaining learning procedure, specifically the calculation of those key statistics $f_{tr}^{(TV)}(R_0)$, $\langle m_\theta \rangle^{(TV)}$ and $\langle \Lambda_\theta \rangle^{(TV)}$, requires the joint estimation of $\langle I_t^{(k_0)}, \dots, I_{t-n_c}^{(k_{n_c})} \rangle$ and $\langle I_{t-1}^{(k_1)}, \dots, I_{t-n_c}^{(k_{n_c})} \rangle$. It can be observed that a large number of enumerations are involved in the parameter estimation with time-variant time delay. In the later part of this section, a practical constraint will be proposed to prune unnecessary calculations.

6.5.1 Modelling Assumption

When time delays are time-varying but sequentially independent, there is no need to model their dynamics, for example, when the actual delay comes from the human error or the inconsistent sampling procedure. In this case, the prior distribution $p(I_{1:T})$ and the proposal distribution $q(I_{1:T})$ are parametrised as independent individual distributions:

$$p(I_{1:T}) = \prod_{t=1}^{T} p(I_t) = \prod_{t=1}^{T} Mul(I_t;\ \pi), \tag{6.43}$$

$$q(I_{1:T}) = \prod_{t=1}^{T} q(I_t) = \prod_{t=1}^{T} Mul(I_t;\ \hat{\pi}_t). \tag{6.44}$$

Different from the proposal distribution (6.35) in time-invariant case, here the proposal distribution (6.44) is parametrised with multiple parameters $\hat{\pi}_{t|t=1,\cdots,T}$. Each of these parameters determines a posterior of time delay at one time instant. By investigating the probability distribution of these posterior realizations, a general behaviour of time delays can be summarized.

To realize such hierarchical modelling, the parameter π in (6.43) is first modelled as a random vector, as denoted with the green circle in Figure 6.9. Based on its definition as the sum-one vector with all positive elements, the prior distribution π is selected as a Dirichlet distribution for the probabilistic conjugacy:

$$p(\pi \mid \alpha) = Dir(\pi;\ \alpha) = \frac{1}{B(\alpha)} \prod_{k=0}^{K} [\pi^{(k)}]^{\alpha^{(k)}-1}, \tag{6.45}$$

where $B(\cdot)$ is the multivariate Beta function. Similarly, the proposal distribution for π is formulated as

$$q(\pi) = Dir(\pi;\ \hat{\alpha}). \tag{6.46}$$

During the variational Bayesian inference, this hyper-parameter $\hat{\alpha}$ will be updated with the estimated $\hat{\pi}_t$, resulting the collection of primary modelling results.

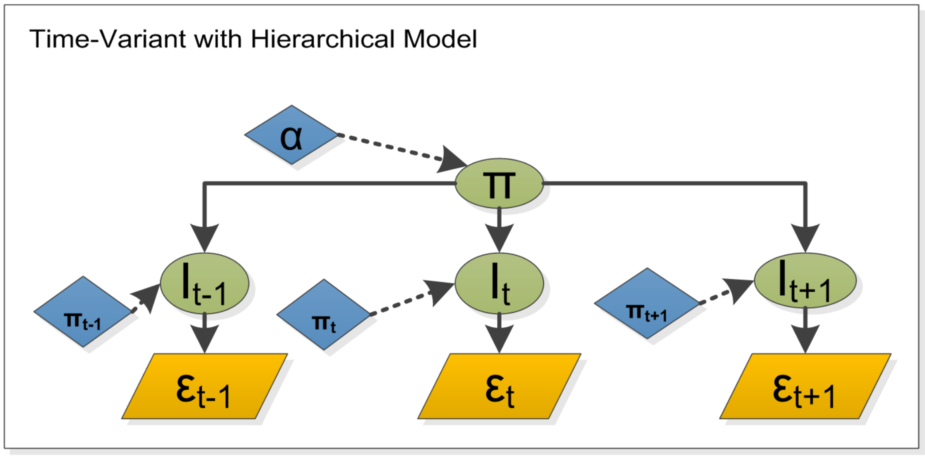


Figure 6.9: Modelling Assumptions for Independent Time-Variant Time Delays

6.5.2 Variational Inference

In this part, the estimation of $q(I_{1:T})$ is obtained based on $\mathcal{L}_{TV}$ in (6.40) and the proposed hierarchical model. Since the prior distributions of $I_{1:T}$ and π have been defined in (6.43) and (6.45) respectively, the updating procedure of $q(I_{1:T})$ becomes

$$\ln q^*(I_{1:T}) = \langle \mathcal{L}_{TV} \rangle_{(\theta_{ab},R)} + \langle \ln p(I_{1:T} \mid \pi) \rangle_{(\pi)} + const..$$

Based on the independence assumption in (6.44), the detailed updating equation for each individual $q(I_t)$ can be formulated based on the knowledge of all the other lag vectors $q(I_{\{1:T\}\backslash t})$:

$$\begin{aligned}
\ln q^*(I_t) =& \langle \mathcal{L}_{TV} \rangle_{(\theta_{ab},R,I_{\{1:T\}\backslash t})} + \langle \ln p(I_t \mid \pi) \rangle_{\pi} + const. \\
=& -\frac{1}{2} \sum_{t=T_s}^{T} \sum_{k_0,\ldots,k_{n_c}}^{K^{n_c}} \langle V_t^{(k_0,\ldots,k_{n_c})} \rangle_{q(I_{\{1:T\}\backslash t})} \cdot tr\{ \langle \Psi_t^{(k_0,\ldots,k_{n_c})} R^{-1} (\Psi_t^{(\ldots)})' \rangle \langle \theta_{ab} \theta'_{ab} \rangle \} \\
& + \frac{1}{2} \sum_{t=T_s+1}^{T} \sum_{k_1,\ldots,k_{n_c}}^{K^{n_c}} \langle \bar{V}_t^{(k_1,\ldots,k_{n_c})} \rangle_{q(I_{\{1:T\}\backslash t})} \cdot tr\{ \langle \bar{\Psi}_t^{(k_1,\ldots,k_{n_c})} \bar{R}^{-1} (\bar{\Psi}_t^{(\ldots)})' \rangle \langle \theta_{ab} \theta'_{ab} \rangle \} \\
& + \sum_{k=0}^{K} I_t^{(k)} \cdot \langle \ln \pi_t^{(k)} \rangle + const. \\
=& \sum_{k=0}^{K} I_t^{(k)} \cdot [S_t^{(k)} + \langle \ln \pi^{(k)} \rangle] + const.,
\end{aligned} \tag{6.47}$$

where $\langle \cdot \rangle_{q(I_{\{1:T\}\backslash t})}$ stands for the expectation with respect to distribution of other lag vectors. Thus, the optimal $\hat{\pi}_t^{(k)}$ for proposal distribution $q(I_t)$ can be obtained by normalizing the factor $[S_t^{(k)} + \langle \ln \pi_t^{(k)} \rangle]$. The innovation part $S_t^{(k)}$ is obtained from the trace terms in which I_t is involved and ψ_t is fixed as $\psi_t^{(k)}$. Mathematically, it is defined as

$$S_t^{(k)} = \sum_{t_n=t}^{t+n_c} \sum_{I_{\{t_n:t_n-n_c\}\backslash t}} \langle I_{\{t_n:t_n-n_c\}\backslash t} \rangle \cdot \ln\{ \; \Gamma_t^{(k_0,\ldots,k_{n_c})} \mid \psi_t = \psi_t^{(k)} \; \}. \tag{6.48}$$

A complete formulation of the observation term $\Gamma_t^{(k_0,\ldots,k_{n_c})}$ will be illustrated later in (6.57), where the proposal distribution $q(\theta_{ab})$ and $q(R)$ are utilized. Here, two special cases are presented as illustrative examples. When applied to the white noise case, where $n_c = 0$, it becomes

$$S_t^{(k)} = -\frac{1}{2} tr\{ \; \psi_t^{(k)'} \langle R^{-1} \rangle \psi_t^{(k)} \langle \theta_{ab} \theta'_{ab} \rangle \; \}.$$

When it comes to the first order colour noise, namely $n_c = 1$, the innovative term becomes

$$\begin{aligned}
S_t^{(k)} =& \sum_{k'=0}^{K} \langle I_{t-1}^{(k')} \rangle \ln\{ \Gamma_t^{(k,k')} \} + \sum_{k''=0}^{K} \langle I_{t+1}^{(k'')} \rangle \ln\{ \Gamma_t^{(k'',k)} \} \\
=& \sum_{k'=0}^{K} \langle I_{t-1}^{(k')} \rangle tr\{ [\psi_t^{(k)}, \psi_{t-1}^{(k')}]' \langle R^{-1} \rangle [\psi_t^{(k)}, \psi_{t-1}^{(k')}] \langle \theta_{ab} \theta'_{ab} \rangle \}
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{k''=0}^{K} \langle I_{t+1}^{(k'')} \rangle tr\{[\psi_{t+1}^{(k'')}, \psi_t^{(k)}]' \langle R^{-1} \rangle [\psi_{t+1}^{(k'')}, \psi_t^{(k)}] \langle \theta_{ab} \theta'_{ab} \rangle\} \\
&- tr\{\psi_t^{(k)'} \langle \bar{R}^{-1} \rangle \psi_t^{(k)} \langle \theta_{ab} \theta'_{ab} \rangle\}.
\end{aligned}
$$

The independence among the proposal distributions in (6.44) also leads to a production formation for the joint statistic:

$$
\langle I_t^{(k_0)} \; I_{t-1}^{(k_1)} \cdots I_{t-n_c}^{(k_{n_c})} \rangle = \langle I_t^{(k_0)} \rangle \langle I_{t-1}^{(k_1)} \rangle \cdots \langle I_{t-n_c}^{(k_{n_c})} \rangle. \tag{6.49}
$$

A similar production applies to $\langle I_{t-1}^{(k_1)} \cdots I_{t-n_c}^{(k_{n_c})} \rangle$. As for updating the proposal distribution for parameter π defined in (6.46), the prior distribution (6.45) will be used to form the following updating equation:

$$
\begin{aligned}
\ln q^*(\pi) &= \langle \ln p(I_{T_s - n_c : T} \mid \pi) \rangle_{I_{1:T}} + \ln p(\pi \mid \alpha) + const. \\
\Rightarrow \; \hat{\alpha}^{(k)} &= \alpha^{(k)} + \sum_{t=T_s-n_c}^{T} \langle I_t^{(k)} \rangle.
\end{aligned} \tag{6.50}
$$

Based on the Dirichlet distribution in (6.45), the necessary statistic in updating equation (6.47) is obtained as

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

where $\psi\{\cdot\}$ stands for the Digamma function. The validation of above modelling algorithms will be discussed later, together with the case of sequentially dependent time-variant time delays.

6.6 Probabilistic Formulation of Dependent Time-Variant Time Delays

Similar to the discussion in Chapter 2, a dynamic model can be used to describe sequentially dependent time-variant time delays and improve the representation of latent variables. In the above probability model of time delay, the assumption/parametrization of time-variant time delay in (6.43) and (6.43) is only a special probability description of the latent sequence $I_{1:T}$. With a dynamic model, in particular the hidden Markov model, the probability description can be generalized by considering the joint distribution of consecutive lag vectors, namely $p(I_{t-1}, I_t)$.

6.6.1 Modelling Assumption

When time delays are sequentially dependent, it is necessary to model their dynamics. For example, the time delay can be caused by material transportation or other related processes. In this case, the probability distribution of the time delays $I_{1:T}$ cannot be assumed to be independent. As a conventional dynamic model for discrete variables, the (first-order) Markov model with a probability transit matrix Π is usually imposed [82, 136]:

$$p(I_t \mid I_{t-1}, \Pi) = \prod_{i=0}^{K} \prod_{j=0}^{K} [\Pi_{i,j}]^{I_{t-1}^{(i)} \cdot I_t^{(j)}} .$$

where $\Pi_{i,j}$ defines the transition probability from i^{th} case to j^{th} case. The prior for $I_{1:T}$ is then determined by this matrix Π and a parameter vector π_0 for the initial state I_1, which is formed as

$$p(I_{1:T} \mid \pi_0, \Pi) = Mul(I_1;\ \pi_0) \cdot \prod_{t=2}^{T} p(I_t \mid I_{t-1}, \Pi). \tag{6.51}$$

When applied in the conventional hidden Markov model, the (step-wise) posterior of $I_{1:T}$ can be learned by the forward-backward algorithm, which has been detailed in Chapter 2. Actually, this method learns the posterior of $I_{1:T}$ with a form of

$$q(I_{1:T}) = \frac{\prod_{t=2}^{T} q(I_t, I_{t-1} \mid \hat{\pi\pi}_{t,t-1})}{\prod_{t=2}^{T-1} q(I_t \mid \hat{\pi}_t)}. \tag{6.52}$$

The parameters $\hat{\pi\pi}_{t,t-1}$ and $\hat{\pi}_t$ determine the joint distribution of consecutive states and the marginal distribution of single state, respectively. Thus, a probability graphical model can be illustrated in Figure 6.10. However, for modelling the coloured noise in ARMAX model, the "observation" ϵ_t needs to be considered in the stacked noise vector Υ_t, where multiple lag vectors will be involved. To solve this "high-order" emission model challenge, a modification on conventional HMM inference will be proposed as a novel learning algorithm.

6.6.2 Variational Inference

With the transition function defined as the Markov model in (6.51) and the observation function defined as $\mathcal{L}_{TV}$ in (6.40), the updating equation of $q(I_{1:T})$ will be solved through a novel algorithm for the hidden Markov model with a "high-order" emission model. In order to achieve this, the conventional Markov chain is first transferred to a higher state dimension by using the following result.

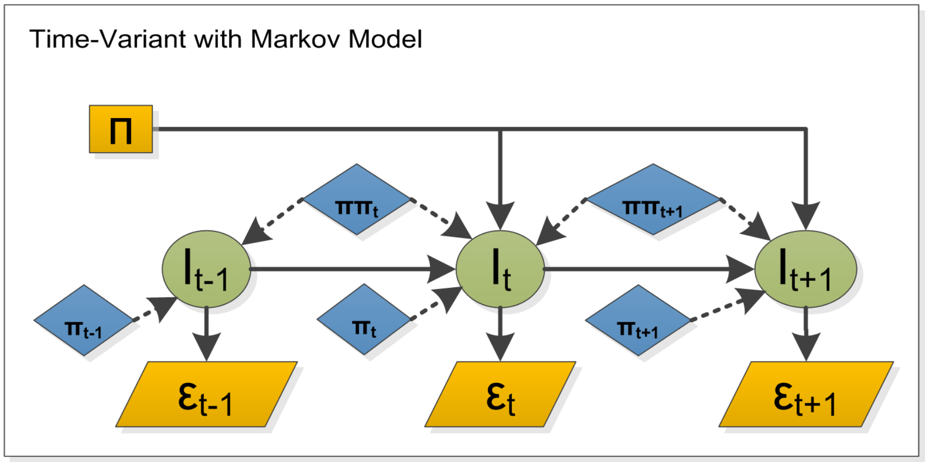


Figure 6.10: Modelling Assumptions for Independent Time-Variant Time Delays

Proposition 3 *Define an augmented state* $V_t \equiv [I_t, ..., I_{t-n_c}]$*, and parametrise its probabilistic description with a multi-dimensional* $\boldsymbol{\pi}_t$*:*

$$p(V_t \mid \boldsymbol{\pi}_t) = \prod_{k_0=0}^{K} \cdot\cdot \prod_{k_{n_c}=0}^{K} \left[\boldsymbol{\pi}_t^{(k_0,..,k_{n_c})}\right]^{I_t^{(k_0)} \cdot\cdot I_{t-n_c}^{(k_{n_c})}} . \tag{6.53}$$

and formulate a nominal transition probability $p(V_t \mid V_{t-1}, \Pi)$ *between* $\boldsymbol{\mathcal{M}}(V_{t-1};\ \boldsymbol{\pi}_{t-1})$ *and* $\boldsymbol{\mathcal{M}}(V_t;\ \boldsymbol{\pi}_t)$ *as*

$$\boldsymbol{\pi}_t^{(k_0,...,k_{n_c})} = \Pi_{k_1,k_0}\ \textstyle\sum_{k_{n_c+1}=0}^{K} \boldsymbol{\pi}_{t-1}^{(k_1,...,k_{n_c+1})}. \tag{6.54}$$

Then, a prior distribution equivalent to $p(I_{1:T} \mid \pi_0, \Pi)$ *in* (6.51) *can be written as*

$$p(V_{n_c+1:T}|\pi_0, \Pi) = \boldsymbol{\mathcal{M}}(V_{n_c+1}; \boldsymbol{\pi}_0) \prod_{t=n_c+2}^{T} p(V_t|V_{t-1}, \Pi),$$

where each element of $\boldsymbol{\pi}_0$ *is obtained from*

$$\boldsymbol{\pi}_0^{(k_0,...,k_{n_c})} = \pi_0^{(k_{n_c})} \cdot \Pi_{k_{n_c},k_{n_c-1}} \cdots \Pi_{k_1,k_0}. \tag{6.55}$$

Proof. As an operator between two probability distributions, $p(V_t \mid V_{t-1}, \Pi)$ is formulated according to

$$\boldsymbol{\mathcal{M}}(V_t;\ \boldsymbol{\pi}_t) = \int_{I_{t-n_c-1}} p(V_t \mid V_{t-1}, \Pi)\ \boldsymbol{\mathcal{M}}(V_{t-1};\ \boldsymbol{\pi}_{t-1}).$$

As a conditional probability, it can be simplified as

$$\begin{aligned} p(V_t \mid V_{t-1}, \Pi) &= p(I_t, ..., I_{t-n_c} \mid I_{t-1}, ..., I_{t-n_c-1}, \Pi) \\ &= p(I_t \mid I_{t-1}, ..., I_{t-n_c-1}, \Pi) = p(I_t \mid I_{t-1}, \Pi), \end{aligned}$$

where $V_t \mid V_{t-1}$ has a same support as $I_t \mid I_{t-1}$. Thus, the equivalence can be formulated as

$$\begin{aligned} p(I_{1:T} \mid \pi_0, \Pi) = &\{ \ \mathcal{M}(I_1; \pi_0) \textstyle\prod_{t=2}^{n_c+1} p(I_t \mid I_{t-1}, \Pi) \ \} \\ &\cdot \{ \ \textstyle\prod_{t=n_c+2}^{T} p(I_t \mid I_{t-1}, ..., I_{t-n_c-1}, \Pi) \ \} \\ = &\boldsymbol{\mathcal{M}}(V_{n_c+1}; \boldsymbol{\pi}_0) \cdot \textstyle\prod_{t=n_c+2}^{T} p(V_t \mid V_{t-1}, \Pi). \end{aligned}$$

■

Conceptually, the augmented state V_t is designed to redeem the independence in observation equations. The updating equation can thus have a standard formation:

$$\ln q^*(I_{1:T}) = \sum_{t=T_s+1}^{T} \ln p(V_t \mid V_{t-1}, \Pi) + \ln p(V_{T_s} \mid \boldsymbol{\pi}_0) + \sum_{t=T_s}^{T} \ln p(\Gamma_t \mid V_t), \tag{6.56}$$

where the initial index T_s is obtained by substituting k with K in (6.13), and all the previous samples of lag variable are treated as redundant information. The observation term is derived from $\langle \mathcal{L}_{TV} \rangle_{(\theta_{ab}, R)}$:

$$p(\Gamma_t \mid V_t) = \prod_{k_0=0}^{K} \cdots \prod_{k_{n_c}=0}^{K} \left[\Gamma_t^{(k_0, ..., k_{n_c})} \right]^{I_t^{(k_0)} \cdots I_{t-n_c}^{(k_{n_c})}}, \tag{6.57}$$

$$\begin{aligned} \textit{where}: \ \ln \Gamma_{T_s}^{(k_0,...,k_{n_c})} &= -\frac{1}{2} tr\{ \langle \theta_{ab} \theta'_{ab} \rangle \cdot \Psi_t^{(k_0,...,k_{n_c})'} \langle R^{-1} \rangle \Psi_t^{(..)} \}, \qquad \textit{if}: t = T_s, \\ \ln \Gamma_t^{(k_0,...,k_{n_c})} &= -\frac{1}{2} tr\{ \langle \theta_{ab} \theta'_{ab} \rangle [\ \Psi_t^{(k_0,...,k_{n_c})'} \langle R^{-1} \rangle \Psi_t^{(..)} \\ &\qquad - \frac{1}{(K+1)} \bar{\Psi}_t^{(k_1,...,k_{n_c})'} \langle \bar{R}^{-1} \rangle \bar{\Psi}_t^{(..)} \ ] \}, \quad \forall \ t > T_s, \end{aligned}$$

where $q(\theta_{ab})$ and $q(R)$ are used to calculate the expectations $\langle \cdot \rangle$.

With the above setting, calculating the hyper-parameters in $q^*(I_{1:T})$ is equivalent to estimating the posterior distribution for hidden Markov sequence. An illustrative plot is shown in Figure 6.11 for an example of the second order case. Accordingly, the modified forward-backward algorithm is proposed as follows. Based on the fact that $I_t \perp \Gamma_{t-i}, \forall \ i > 0$, the predicting path is defined as follows:

$$\begin{aligned} \alpha_t^{(p)} &\equiv p(V_t \mid \Gamma_{1:t-1}) = p(I_t, ..., I_{t-n_c} \mid \Gamma_{1:t-1}) \\ &= \sum_{I_{t-n_c-1}} p(I_t, I_{t-1}, ..., I_{t-n_c-1} \mid \Gamma_{1:t-1}) \\ &= \sum_{I_{t-n_c-1}} p(I_t \mid I_{t-1}, ..., I_{t-n_c-1}, \Gamma_{1:t-1}) \cdot p(I_{t-1}, ..., I_{t-n_c-1} \mid \Gamma_{1:t-1}) \\ &= \sum_{I_{t-n_c-1}} p(I_t \mid I_{t-1}) \cdot \alpha_{t-1}. \end{aligned} \tag{6.58}$$

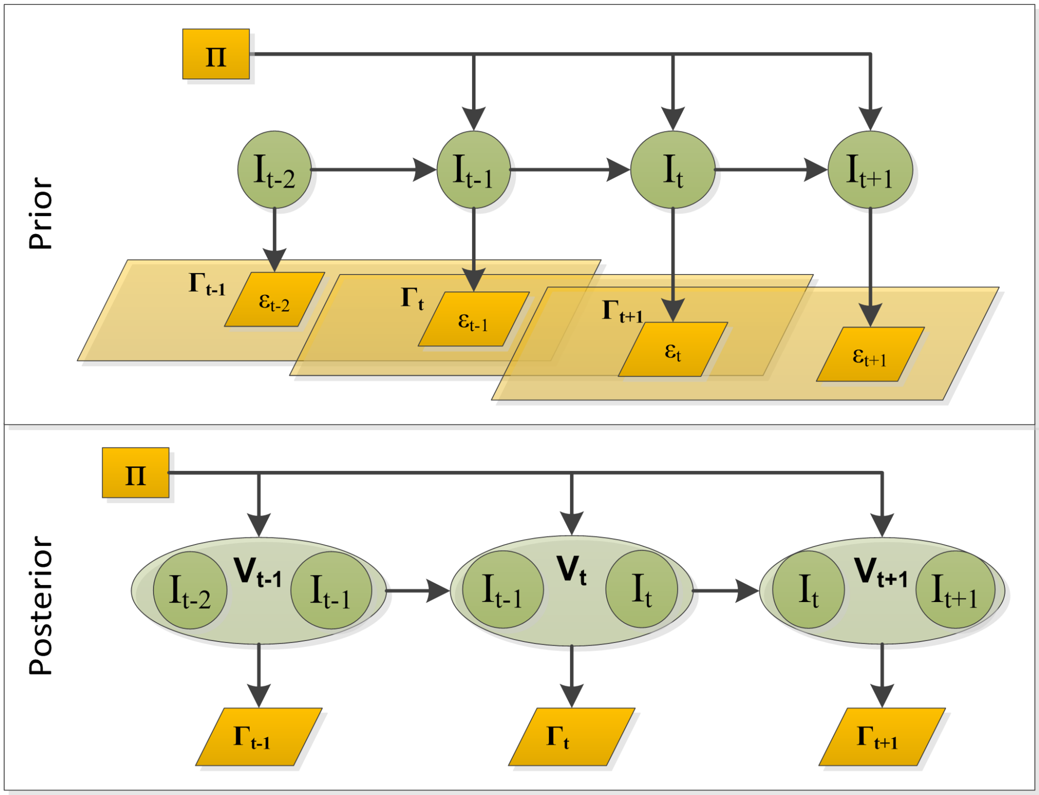


Figure 6.11: Estimate Hidden Markov Sequence with High Order Observations

Based on the fact that all $\Gamma_{1:T}$ are given values in this variational updating step, the forward (filter) path is derived as

$$\begin{aligned}\alpha_t \equiv p(V_t \mid \Gamma_{1:t}) &= p(I_t, ..., I_{t-n_c} \mid \Gamma_{1:t}) \\ &= \frac{p(\Gamma_t, I_t, ..., I_{t-n_c} \mid \Gamma_{1:t-1})}{p(\Gamma_t \mid \Gamma_{1:t-1})} \\ &= \frac{p(\Gamma_t \mid I_t, ..., I_{t-n_c}, \Gamma_{1:t-1}) \cdot p(I_t, ..., I_{t-n_c} \mid \Gamma_{1:t-1})}{p(\Gamma_t \mid \Gamma_{1:t-1})} \\ &\propto p(\Gamma_t \mid I_t, ..., I_{t-n_c}) \cdot \alpha_t^{(p)}. \end{aligned} \tag{6.59}$$

The final posterior is obtained from the backward (smoother) path based on all the predict and forward statistics:

$$\begin{aligned}\gamma_t \equiv p(V_t \mid \Gamma_{1:T}) &= p(I_t, ..., I_{t-n_c} \mid \Gamma_{1:T}) \\ &= \sum_{I_{t+1}} p(I_{t+1}, I_t, ..., I_{t-n_c} \mid \Gamma_{1:T}) \\ &= \sum_{I_{t+1}} p(I_{t+1}, ..., I_{t-n_c+1} \mid \Gamma_{1:T}) \cdot p(I_{t-n_c} \mid I_{t+1}, ..., I_{t-n_c+1}, \Gamma_{1:T}) \\ &= \sum_{I_{t+1}} \gamma_{t+1} \cdot p(I_{t-n_c} \mid I_{t+1}, ..., I_{t-n_c+1}, \Gamma_{1:t})\end{aligned}$$

$$
\begin{aligned}
&= \sum_{I_{t+1}} \gamma_{t+1} \cdot \frac{p(I_{t+1}, ..., I_{t-n_c+1}, I_{t-n_c} \mid \Gamma_{1:t})}{p(I_{t+1}, ..., I_{t-n_c+1} \mid \Gamma_{1:t})} \\
&= \sum_{I_{t+1}} \gamma_{t+1} \cdot \frac{p(I_{t+1} \mid I_t, ..., I_{t-n_c+1}, I_{t-n_c}, \Gamma_{1:t}) \cdot p(I_t, ..., I_{t-n_c+1}, I_{t-n_c} \mid \Gamma_{1:t})}{p(I_{t+1}, ..., I_{t-n_c+1} \mid \Gamma_{1:t})} \\
&= \alpha_t^{(p)} \cdot \sum_{I_{t+1}} \gamma_{t+1} \cdot \frac{p(I_{t+1} \mid I_t)}{\alpha_{t+1}^{(p)}}. \qquad (6.60)
\end{aligned}
$$

In order to connect the forward path and the backward path, the fact that $\gamma_T = \alpha_T$ is utilized.

The updated hyper-parameter $\hat{\boldsymbol{\pi}}_t$ is for each augmented state in above backward path $p(V_t \mid \Gamma_{1:T})$, which is ready for $\langle I_t^{(k_0)} \cdots I_{t-n_c}^{(k_{n_c})} \rangle$ to do variational updating. From Figure 6.11, it can be observed that I_t exists in both V_t and V_{t+1} in the posterior. Based on the backward path derived above, the marginalized estimate of $\langle I_t \rangle$ from either augmented state will be identical, which guarantees a consistent calculation for $\langle I_t^{(k_0)} \cdots I_{t-n_c}^{(k_{n_c})} \rangle$ and $\langle I_{t-1}^{(k_1)} \cdots I_{t-n_c}^{(k_{n_c})} \rangle$. The numerical demonstration will be provided shortly.

6.7 Practical Consideration for Time-Variant Time Delays

A key idea of the proposed probabilistic identification method is to formulate the random time delay as a multi-mode problem. For its practical application, a constraint on the dynamic behaviour of time-variant time delay should be considered. If the system is causal, meaning that model outputs only depend on the past (and current) inputs, the non-negativity in Definition 2 will be sufficient. However, if the system is assumed to be incapable of using the input signal in reverse order, some controversial cases may be found. For example, the case that $I_{t-1}^{(1)} = I_t^{(3)} = 1$ should not occur in reality. In other words, if the system has generated y_{t-1} by the inputs up to u_{t-2}, it cannot miss u_{t-2} and only use inputs up to u_{t-3} in generating y_t.

This kind of "first-come-first-serve" rule can be informally defined as follows. The effect of the latest input sample on a future output sample must occur later than the effect of the latest input sample on the current output sample. As for model (6.1) with the unit sampling interval, this rule can be interpreted as an additional constraint for the lag vector defined in Definition 2:

$$
k_{t+1} - k_t \leq 1, \quad \text{if}: I_{t+1}^{(k_{t+1})} = I_t^{(k_t)} = 1. \qquad (6.61)
$$

A similar concept has been discussed in a study of the stability issue in [132]. As for the identification objective, this constraint can be realized by limiting the supporting domain of $I_{t:t-n_c}$. Practically, the integration operator in (6.41) can be trimmed as

$$\prod_{k_0=0}^{K}\prod_{k_1=0}^{K}\cdot\prod_{k_{n_c}=0}^{K} \Rightarrow \prod_{k_0=0}^{K}\prod_{k_1=k_0-1}^{K}\cdot\prod_{k_{n_c}=k_{n_c-1}-1}^{K}. \tag{6.62}$$

In this study, it has been realized for the sequentially independent delay case.

As for the sequentially dependent time delays, this constraint is further interpreted through the parameter of lag variables:

$$\textstyle\sum_{k=0}^{K} k\cdot[\pi_{t+1}^{(k)}-\pi_t^{(k)}]\leq 1. \tag{6.63}$$

Strictly speaking, this condition should be satisfied in the converged result of each I_t, where the constrained state estimation algorithms [61] can be considered. In this study, the above constraint (6.63) is transferred to the transition matrix Π in (6.51) – that is, an eligible Π must have the following inequality hold for any π_t defined in (6.33):

$$\pi_t'\cdot(\Pi-\mathbb{I}_{K+1})\cdot[0,1,...,K]'\leq 1, \tag{6.64}$$

where $\mathbb{I}_{K+1}$ is the $K+1$ dimensional identity matrix. The effectiveness of the constrained matrix Π will be shown before validating the inferencing algorithm for $q(I_{1:T})$.

Table 6.2: Validation Results for Constrained Transition Matrix

α	0.9	0.8	0.7	0.6
$\langle I_{1:T}\rangle_{ss}$	0.6605	0.7710	0.8500	0.9073
$\langle I_{1:T}\rangle_{base}$	0.7468	0.8362	0.9127	0.9591
$\langle I_{1:T}\rangle_{cons}$	0.7040	0.7962	0.8805	0.9346

In particular, the tridiagonal matrix Π: $\Pi_{i,j}=0,\forall|i-j|>1$ is suitable for the constraint (6.64), and has been used in this study. Here, we give a brief demonstration on using such tridiagonal Π to improve the estimation of $I_{1:T}$. In this simulation, the actual $I_{1:T}$ with $T=1000$ and $K=5$ is generated by a tridiagonal matrix. The principal diagonal elements are set to α, and the other element(s) uses $1-\alpha$. Given the model parameters θ, $I_{1:T}$ is estimated by the forward-backward algorithm in three cases: (1) with the actual Π, (2) learning an unconstrained Π [82], and (3) learning a tridiagonal Π. As a result, the mean absolute error is evaluated for the three corresponding $I_{1:T}$ estimates: $\langle I_{1:T}\rangle_{ss}$, $\langle I_{1:T}\rangle_{base}$, and $\langle I_{1:T}\rangle_{cons}$. In Table 6.2, the performance from Monte Carlo simulations (×100) is presented for different α values. A clear deficiency can be observed in the unconstrained case, indicating that when (6.61) is considered as a necessary constraint, restricting Π can improve the estimation of $I_{1:T}$.

6.8 Simulation

By integrating the probabilistic model and the Bayesian inference method, this proposed approach can improve the estimation of model parameters. In this section, applications of the proposed time delay structure will be demonstrated, and the numerical example will be presented to illustrate the detailed learning procedure. To further validate the proposed approach, two industrial applications will be used for demonstrating the advantages.

6.8.1 Validation for Time Delay Estimation

In order to validate the updating procedure of $q(I_{1:T})$ for both independent and dependent time-delay assumptions, a short sequence of deterministic time delays is used to generate $\Gamma_{1:T}$ with $n_c = 1$. With these nominal "observations" $\Gamma_{1:T}$, two modelling approaches of time delay can recover the latent states from different perspectives. A preliminary result is shown in Figure 6.12, where the time delays are plotted with cumulative bars to reveal the probability of each estimated time delay value. It can be observed that the modes of both two posteriors have captured the correct time delays for these seven samples, if maximum-a-posterior criterion is used to determine the time delays. As for the result from Markov modelling, the estimation at the fifth sample is less assured because of a conflicting likelihood between "transition" and "observation". The preference would to be $k = 1$ according to the transit matrix, but it would be $k = 2$ based on Γ_t and Γ_{t+1}. As for the result of hierarchical modelling, there exists a small probability of $k = 0$ for the fourth and sixth sample, which is caused by the updated prior distribution of π.

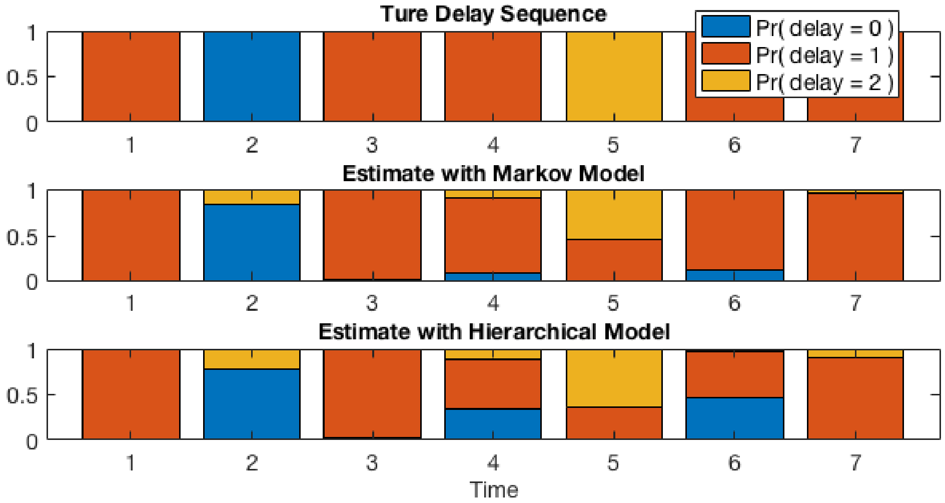


Figure 6.12: Effects of Two Assumptions for Time Variant Time Delays

For the hyper-parameter π in the hierarchical modelling, its prior distribution in (6.45) and the updated proposal distribution in (6.46) can be visualized in Figure 6.13. It can be observed that for the posterior estimation, the mode has been brought closer to the vertex $[\pi^{(0)} = 0, \pi^{(1)} = 1, \pi^{(2)} = 0]$, representing that the most frequent case $k = 1$ has been captured with this distribution.

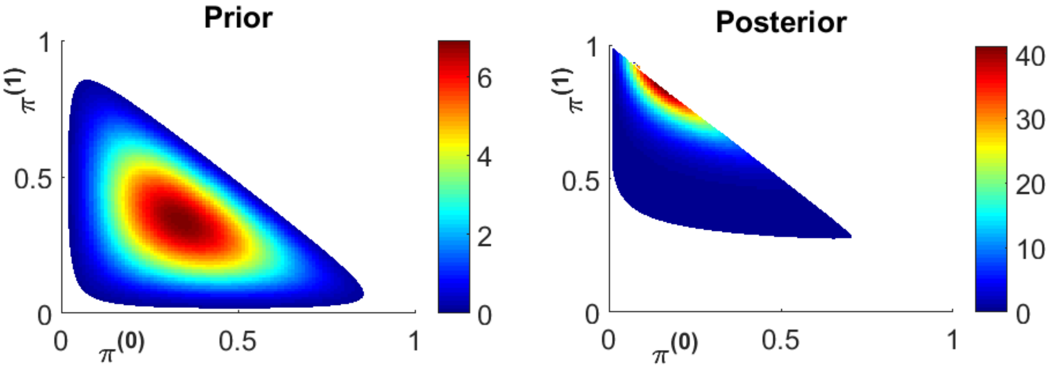


Figure 6.13: Describe Random Delays with Dirichlet Distribution

Another example of the updating results of the hierarchical modelling is shown in Figure 6.14. The prior distribution of the hyper-parameter π and a regression parameter θ_1 are shown in the first row, the results after first and third VB updating step are shown in the second and the third row, and the converged distributions are plotted against the actual ones in the last row. Starting from the equally weighted prior, this expectation $\langle \pi \rangle$ has finally converged around the true π^* (which has been re-scaled in yellow bar). The evolution updating procedure for θ_1 can also be found. Starting from a trivial prior distribution (low probability density), the mean of updated Normal distribution has gradually shifted to the real θ_1^*, and its spread of distribution has gradually been reduced.

6.8.2 Validation for System Identification

Following the above preliminary simulation studies, the complete identification procedure is implemented in this section to validate the algorithms for identification of ARMAX with time-variant time delay. In the first validation example, a set of simulated data is generated with an ARMAX model with $n_a = n_b = 2$ and $n_c = 1$, where the time delay is selected from $\{0, 1, 2\}$. As for the two proposed algorithms for time-variant time delay, the upper limit of time delay is chosen as $K = 4$. A comparison of the actual time delay sequence and the estimated sequences is presented in Figure 6.15. It can be observed that the Markov modelling assumption provided a better result for estimating the time delay sequence, where

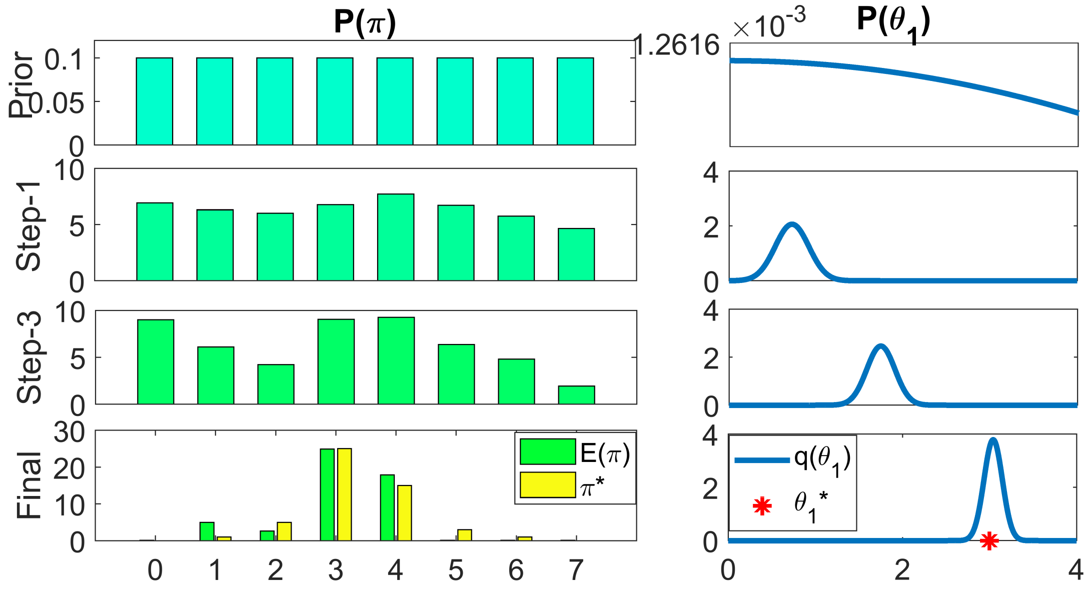


Figure 6.14: Evolutions in the Distribution of π in VB Inference

the dominating time delay value is significant at every time instant. On the other hand, although one can still roughly differentiate several regions from the bottom plot based on the hierarchical modelling, the margins of each region are not as distinctive as the Markov modelling result.

In addition to the comparison between time delay estimates, as the main objective of system identification, the parameter learning progress is plotted along the VI updating steps. Figure 6.17 and Figure 6.16 have shown the parameter estimation result from Markov modelling and hierarchical modelling respectively.

Along with the updating curves (VTD) for variational inference, the 3σ range at each updating step is plotted as an error bar, where σ is obtained from the proposal distribution $q(\theta_{ab})$ in (6.18). It can be observed that within this range, both modelling approaches can capture the real model parameters eventually. In Figure 6.16, a competing estimation (denoted as Existing) result from the existing identification toolbox in MATLAB is also plotted. It can be observed that without considering the time-variant uncertainties of time delay, this method has failed to capture the correct parameters for this example. Similar to Figure 6.15, the Markov model shows its advantages over the hierarchical model with a quicker convergence of parameter estimation.

The proposed algorithm with the Markov model of time-variant time delay shows superior performance in this identification study, indicating that the time delay sequence does have the sequential correlations. In reality, in the process of system identification, the

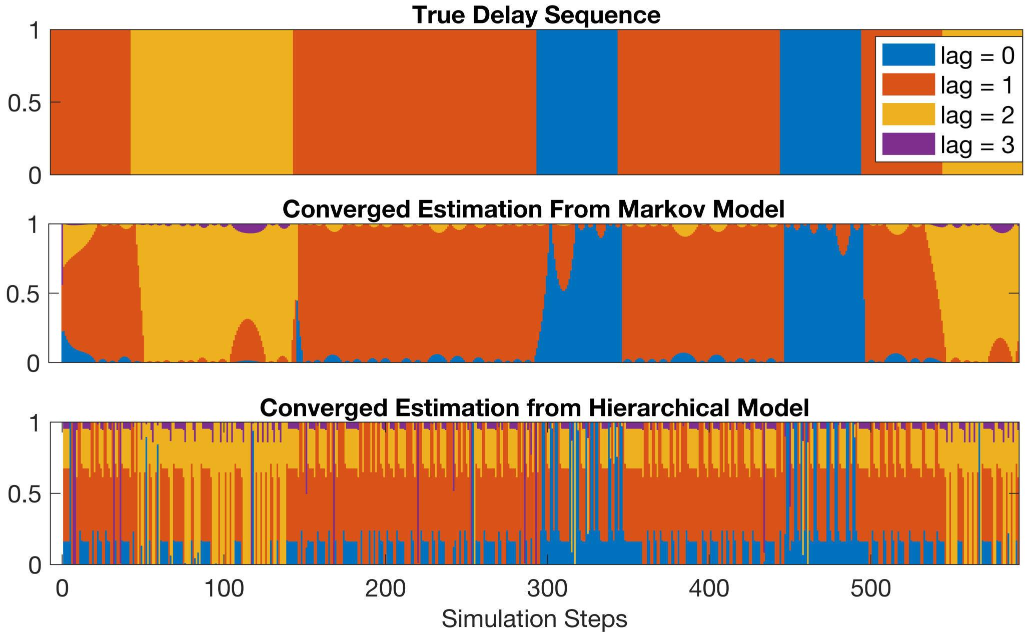


Figure 6.15: Comparison of Delay Estimations in ARMAX Identification

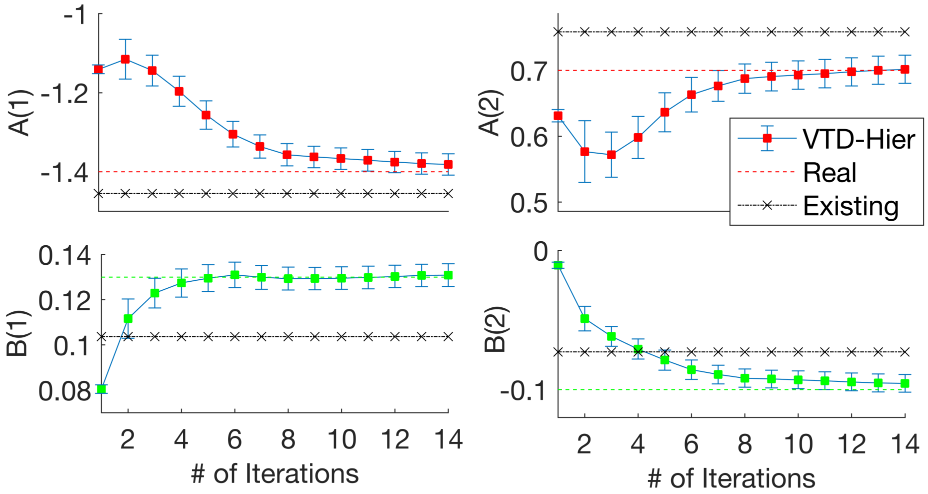


Figure 6.16: Learning Procedure with Independent Time Delays

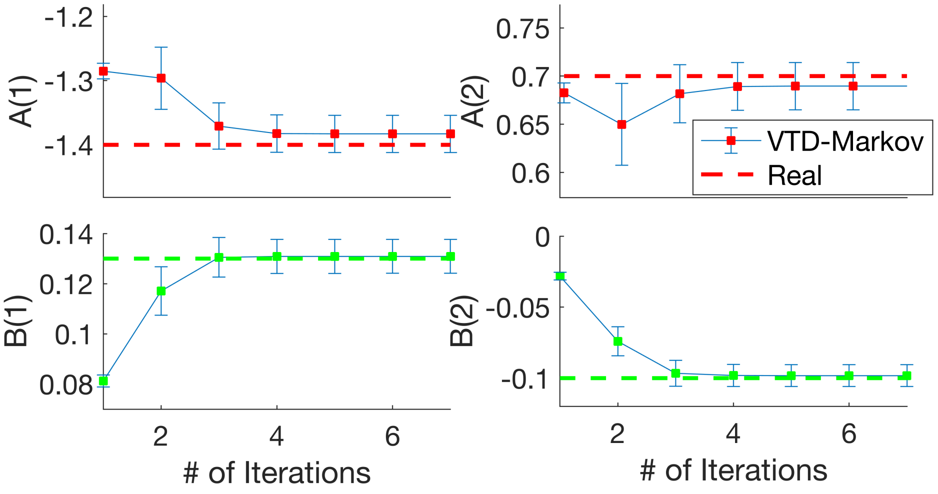


Figure 6.17: Learning Procedure with Dependent Time Delays

input-output pairs are usually sampled at an appropriately selected rate. Under this sampling rate, the time delay sequence may appear to have a sequential dependency in common scenarios. Thus, using a dynamic model to describe $I_{1:T}$ is sufficient, and the prior distribution in (6.51) can provide a reasonable preference. On the other hand, the hierarchical model of random time delay still has its practical usage. In the next validation example, its effectiveness is demonstrated through an industrial application example.

6.8.3 Validation for Inferential Sensor Modelling

In this section, the problem of time-varying time delays from two inferential sensing projects is briefly introduced. Since the main focus is on the demonstration of the proposed algorithm for independent time-variant time delays, only linear and widely applied algorithms are considered, such as the linear regression model (OLS), linear regression with lagged input samples (DOLS), and the principal components regression with lagged input samples (DPCR) [69].

First, the problem is found in the inferential sensor modelling for Once-Through Steam Generators (OTSGs), where the outlet steam quality of individual passes is the model output [115]. Due to the equipment limitation and the capital cost, reliable analyzers cannot be implemented to have on-line measurements for steam quality. The current references of the steam quality are manually sampled once in roughly every 6 hours. The field operators use the sample container to collect the target fluid continuously. Once the container is filled, a conductivity electrode will be placed. After waiting for some time for a stable reading, the analyzed value is recorded and time is stamped. The uncertain delays are originated from: (1) the sampling procedure, since the conductivity reading needs to be stable before recording the sample; (2) the calculation procedure, as both inlet and outlet conductivities are required to obtain steam quality; (3) the human error, the time stamps may be delayed because of operators' shift schedules.

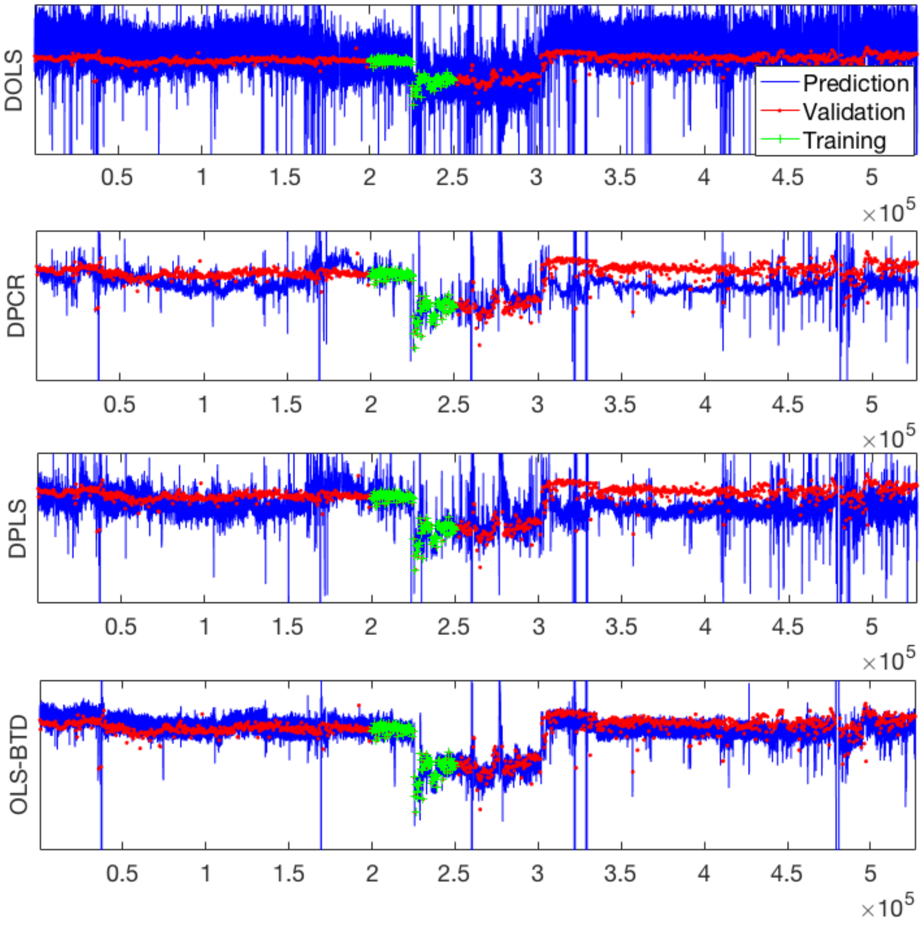


Figure 6.18: Demonstration of Hierarchical Model in Steam Quality Soft Sensor

In this application, the hierarchical model of random time delays is treated as a reinforcing tool to improve conventional models, specifically for the multivariate linear regression model. The distinctive characteristic of target data is the different sampling rate between the inputs and the output. The reference output samples have a significantly slower sampling rate comparing to inputs (which are regularly measured process variables). For example, the steam quality is manually sampled every 6 hours, but flow-rate, temperature, and pressure measurements are available every minute. In such scenarios, the assumption behind hierarchical modelling, namely sequential independence of time delays, becomes quite suitable. In Figure 6.18, the reinforced linear regression model (OLS-BTD) demonstrates the advantage of hierarchical modelling for time-variant time delays. In this comparison, the regular multivariate linear regression model and the two most dominate linear models are used. For a fair competition, these models are provided with augmented inputs; that is, all the lagged inputs are considered in the principal component regression method and the partial least squares method. It can be seen that the proposed model has more effective usage of the lagged inputs, which results in improvements in both accuracy and robustness (less stiff peaks).

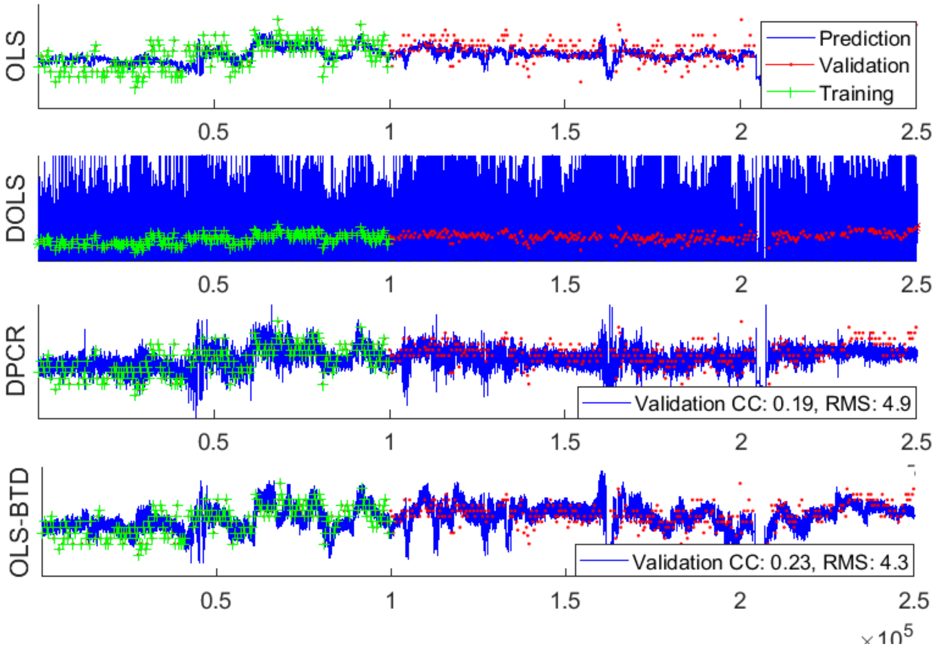


Figure 6.19: Demonstration of Hierarchical Model in Water Content Soft Sensor

The second case is from an oil treating process, where operators usually take manual samples twice per shift. Once the sampled fluid is obtained in the lab, the centrifuge will spin the sample for about 40 minutes to get the reference value (water content). These readings are usually entered at the end of the shift. In practice, the time stamps of the entered data could be either the sampling time or the time when a sample is entered into the database. The delays of the references in this project could be up to one hour and inconsistent due to different operators. Figure 6.19 shows the estimation results on this data set, where "CC" stands for the Pearson's Correlation Coefficient and "RMS" stands for the root mean squared error. Similar to the previous application, the proposed hierarchical modelling approach provided the most accurate validation result. It should be noted that the relatively low “CC” values are because of the quite noisy reference sequence. Although it may not be very significant, the improvement of “CC” is valuable for demonstrating the predictive ability of the models in practice.

6.9 Conclusions

In this study, a probabilistic modelling framework is proposed and applied for ARMAX model identification. Three typical assumptions are considered to describe random time delays. The graphical probability models for these three modelling assumptions are provided in Figure 6.6, Figure 6.9, and Figure 6.10 respectively. where green circles denote random variables, yellow rectangles denote given numbers, and blue diamonds denote the parameters.

Using the approximate Bayesian learning strategies, technical details are provided for each time delay models. During the parameter learning procedure, novel algorithms are proposed and validated for probabilistic ARMAX modelling. Other than the validations of each learning step, the overall performance of the proposed identification framework is validated through both the system identification and the practical inferential sensor modelling. As shown through the examples, the proposed modelling framework is capable of estimating both time-invariant and time-variant time delays, improving existing identification algorithm by considering the random time delays and strengthening existing regression models with more effective use of lagged input samples.

Chapter 7

Concluding Remarks and Future Directions

In this chapter, the concluding remarks are provided for the preceded chapters of this thesis. The connection between these chapters to the central theme of this thesis is explained. Furthermore, possible studies of future research are also discussed.

7.1 Concluding Remarks

The main topic of this thesis is extracting dynamic latent features to improve process data modelling. By formulating this problem under the Bayesian framework, and learning the probability model with the variational Bayesian inference, practicability and advantages of the developed probabilistic methods have been demonstrated via multiple application examples. In particular, better solutions have been demonstrated in the modelling problem for the steam quality and emulsion water content, the state estimation problem of target tracking problems, and the system identification problem with time-variant time delay.

In Chapter 2, the mathematical fundamentals of this thesis are explained. To formulate a decoding format for latent variable modelling, two primary probabilistic models are introduced to describe the dynamic latent features. Along with them, conventional learning algorithms have been briefly reviewed. Motivated by their deficiency in process data analysis, the variational Bayesian inference is introduced as an advanced Bayesian algorithm for feature extraction. To reveal the computational advantages of the variational Bayesian inference, a comparison to the conventional EM algorithm is detailed in each updating step. Additionally, the particle-based algorithms are reviewed to reinforce the variational Bayesian inference for dealing with practical challenges. Typically, they are suitable for solving the non-Gaussian distribution with the constrained variables.

To demonstrate the applicability of dynamic latent features in process data modelling, theoretical contributions are presented in the four chapters respectively.

- The first contribution, as introduced in Chapter 3, provided a Bayesian solution to the probabilistic slow feature analysis. Through the probabilistic modelling approach, the dynamic latent features are formulated by a transition model, which has constrained transition parameters. By adopting a Beta distribution for the prior distribution of transition parameters, the modelling preference on the slow varying trend can be materialized mathematically. Unlike the conventional maximum likelihood estimation, the proposed Bayesian framework can combine both the training data and the modelling preferences for the inference. In the inferencing procedure, the importance sampling techniques are adopted to estimate the constrained transition parameters. After imposing other prior distributions for the observation parameters, the automatic relevance determination can be realized by using a variational lower bound to select the model structure. The proposed algorithm is validated with numerical simulations. Despite the randomness of initial values, the proposed algorithm can provide a consistent estimation result for this feature extraction problem. The proposed algorithm is then applied to an industrial case study for modelling emulsion water content, where the effectiveness is validated through the prediction performance.

- The second contribution, as introduced in Chapter 4, provided a generalization of the hidden Markov model. Instead of using the discrete latent state to represent the separated events, their probabilities are directly modelled as the constrained and continuous latent feature. Thus, the distribution of the original discrete states can be directly used to describe the observation. To extend the transition function for this continuous supporting domain, a Dirichlet distribution is used to describe the constrained transition uncertainty, where the original transition matrix is only used to provide an expected value. The two-dimensional specification with Beta distribution can be used to describe multiple individual features. Through the comparison with the unconstrained state transition model, specific functionalities of the proposed transition parameters have been studied. In particular, the preferred range of these transition parameters is determined by simulating the constrained latent feature, which facilitates the development of Bayesian inference. To integrate the constrained dynamic feature into a feature extraction model, a non-linear observation function is proposed to describe the unconstrained observations with the constrained feature. In the inferencing procedure, novel algorithms have been developed for solving a problem of constrained state estimation and a problem of transition noise estimation. According to the numerical simulations, the proposed algorithms have shown satisfactory results. In the modelling problem of emulsion water content, the proposed structure of the constrained dynamic feature is validated through the case study with sparse reference data.

- The third contribution, as presented in Chapter 5, is the development of the observation function with multiple linear models. While using a single transition function to represent a consistent process dynamic, the observation function is extended with multiple models to reflect switching operations. To prevent unnecessary switching due to disturbance, a heavy-tailed noise distribution is used to make each model more robust to observation outliers. Comparing to learning an explicit hidden Markov model, this strategy has less dependency on the repeatable switching actions. In the inferencing process, the uncertainties of multiple models are considered in learning a unified latent feature. For its online usage, a more general variational filtering algorithm has been developed for solving the multiple model state estimation problem. In an application to the steam quality modelling problem, the proposed feature extraction method has shown to be able to capture the multiple regions in the operating OTSG. In an application to the target tracking problem, the developed state estimator has improved both the effectiveness and efficiency.

- The fourth contribution, as presented in Chapter 6, is the application of the dynamic latent feature to model time-variant time delays in solving the system identification problem. By describing the time delay sequence as a dynamic feature, the corresponding observation function is developed as a likelihood formulation of the ARMAX model. Three possible assumptions are discussed to describe the random time delay: time-invariant, independent time-variant, and dependent time-variant. Regarding the system identification task, the last assumption with a Markov connection is demonstrated to be more useful. In this scenario, the proposed observation function assigned multiple observation terms to one latent state (an instant of time delay). To estimate this unconventional hidden Markov model, an augmented transformation has been developed, and the novel forward-backward algorithm is proposed, which is proved to be able to retain the original Markov transition and preserve the estimation uniqueness. In another industrial case study, the second assumption of sequentially independent time delay is also demonstrated to be useful for the regression analysis. Specifically, if the actual outputs are obtained through manual sampling and lab analysis, the proposed structure can improve the regression results by accommodating the inevitable human errors.

To summarize, this thesis is motivated by the modelling advantage of dynamic latent features in process data analysis, which has been illustrated in Chapter 1. Based on the mathematical foundation shown in Chapter 2, the dynamic latent feature can be established through a transition function and an observation function, for which the inferencing process can be realized with the variational Bayesian inference and the particle-based algorithms. In Chapter 3 and Chapter 4, novel transition functions are developed under the Bayesian framework. In Chapter 5 and Chapter 6, more practical observation functions are studied regarding their impact on extracting latent features. Theoretically, the challenges of constrained parameters, constrained latent features, multiple-model state estimation, and auto-correlated observations have been encountered and addressed in this thesis. Practically, the modelling preferences of process inertia, the assumptions about process limitations, the multiple operating regions, and the time-variant time delays are discussed in this thesis.

7.2 Future Directions

Based on the flexibility of the variational Bayesian inference and the commonly observed process dynamics, there exist many directions that can be proceeded further. For example, from the modelling point of view, further application scenario of the novel transition model proposed in Chapter 4 can be considered for multiple-model cases. From the inferencing point of view, the iterative updating strategy can be further optimized for multiple sets of model parameters, such as optimizing the order of variational updating steps in Chapter 5. Also, the fusion between the particle-based algorithm and the variational Bayesian inference can be extended beyond the proposed "step-wise" integration. In what follows, two more exciting directions are illustrated based on emerging modelling techniques.

7.2.1 Bayesian Co-integration Analysis

A shared assumption made in the models mentioned above is that the latent feature is a stationary process. It means that with the determined transition parameters, the generated stochastic process should have a time-invariant distribution as its stationary solution. For example, with the constraint $a^2 + \rho^{-1} = 1$ in Chapter 3, the standard Normal distribution

$\mathcal{N}(0,1)$ becomes the stationary solution. Generally, the statistical properties of a stationary process are independent of time. For a stationary latent feature, the first two moments are assumed as $\mathbb{E}[s_t] = 0$ and $\mathbb{E}[s_t^2] = 1$. However, in reality, chemical processes can be non-stationary. Possible causes may come from the multiple operating modes, the inevitable equipment ageing, or the possible accumulative deposit. For a better understanding of these processes, analyzing the non-stationary feature is as important as analyzing the stationary feature.

Specifically, the accumulation in chemical processes, such as fouling build-up in the steam generation process, is a very likely situation for the non-stationary feature. Due to its complexity, it is difficult to monitor fouling build-up through hardware sensors. In some obvious cases, the total amount of fouling build-up can be monitored with some simple calculations, such as the pressure drop. However, these indicators are usually affected by operating regions as well. Relying on a single indicator may not give the consistent estimate, and tuning the multiple calculations in real-time may not be practical. However, by extracting the non-stationary feature from the historical data, a more realistic indicator can be developed from multiple process measurements. Figure 7.1 shows an example of the estimated monitoring curve for the fouling build-up, where the non-stationary properties and the co-integration method are used in the feature extraction.

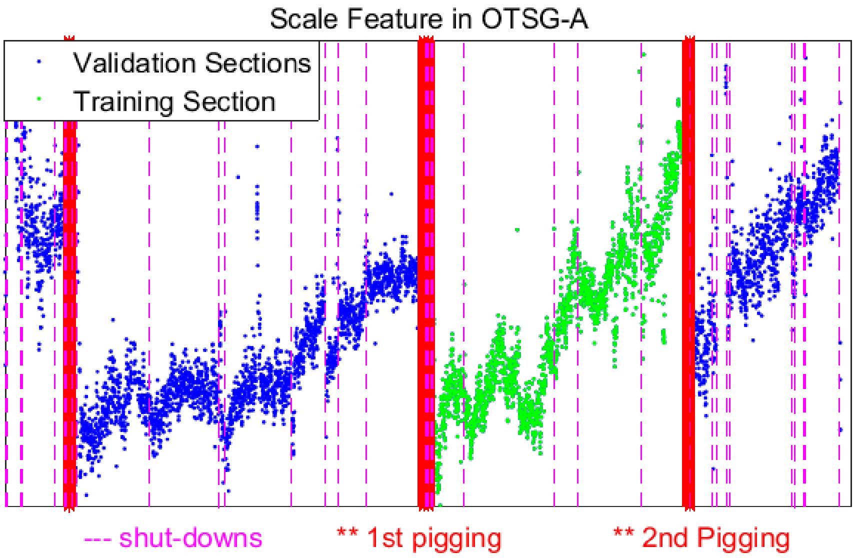


Figure 7.1: Scaling Monitoring in Steam Generation Process

Co-integration analysis, which has been widely applied in the econometrics area, can be applied to investigate the relationship between stationary and non-stationary features. By treating the co-integration analysis as a feature extraction method, a projection matrix can be formulated to learn the stationary feature from the observed data [143]. In the reversed order, the most non-stationary feature can also be formulated as another projection: $S_t = \beta \cdot X_t$. Successful application of co-integration analysis in chemical engineering can be found in [144, 145]. In Figure 7.1, the plotted feature is extracted from the reversed co-integration analysis.

To model this linear feature extraction under the Bayesian framework, the dynamic latent feature can be useful in formulating a probabilistic decoder. A possible approach can start by describing a non-stationary stochastic process by manipulating the transition parameters of the dynamic latent feature, such as in the nominal case:

$$s_t = a \cdot s_{t-1} + w_t, \qquad where : a = 1, \quad w_t \sim \mathcal{N}(0, 1). \tag{7.1}$$

By incorporating the constraint of $a = 1$, the existing transition function with $a \in (0, 1)$ is actually extended with much wider modelling scope. The separated modelling strategies of the stationary and non-stationary work could thus be unified with one feature extraction model. However, the associated estimation challenges can also be increased, where more constraints on other transition parameters may be necessary to have a consistent estimation.

7.2.2 Transition Noises in Recurrent Neural Networks

It can be observed that in the parameter estimation process, the variational learning methods require the posterior of entire latent features $q(S_{1:T})$, but only use it to update the model parameters once. While dealing with a large dataset, such an updating algorithm can be impractical. Moreover, although the variational lower bound can be used to select the final results, each variational updating cycle can still be affected by the initial point. For some heavily loaded algorithms, this trial-and-error strategy becomes extremely computationally expensive.

Meanwhile, as briefly introduced in Chapter 2, the stochastic variational inference has the advantage to overcome the issue of local optima and can be adjusted well for large datasets. More importantly, the combination of an encoding feature learning model and the probabilistic decoding model has provided considerable improvements in computer science [28, 146]. To address the dynamic of a stochastic feature in the hidden layer, temporal connections have been implemented with several considerations [147, 148, 149]. However, to the best of author's knowledge, no literature explicitly discussed the transition noise.

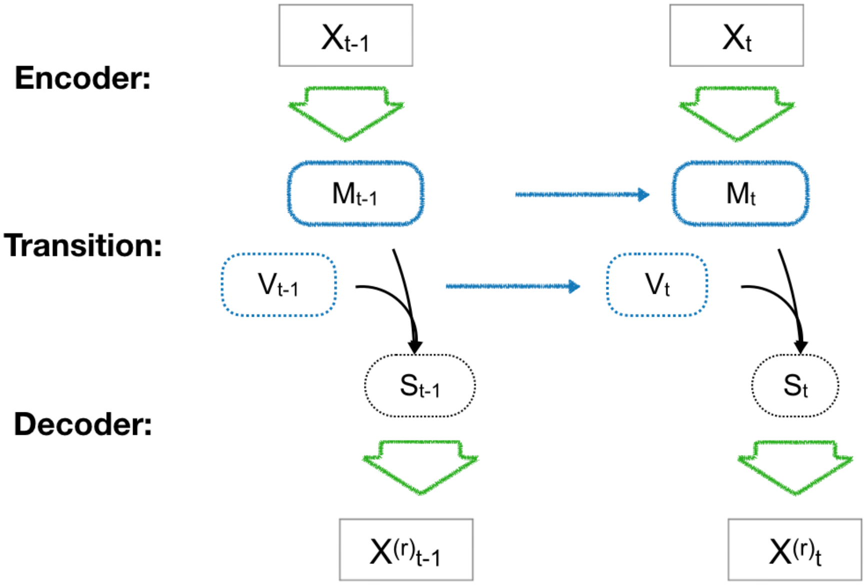


Figure 7.2: Possible Structure for Recurrent Variational Auto-Encoder

Taking the advantages of stochastic variational inference and extend the application of variational auto-encoder from static modelling to dynamic modelling, a recurrent variational auto-encoder can be proposed as shown in Figure 7.2. The encoder function is built to project the observations for the mean and the variance of latent features. Based on the particle-based representation in [28], the distribution of latent state can be represented by the particles that are generated from these parameters. Then, the recovered observation can be obtained from these sampled states. In order to introduce the recurrent properties, the transition function should be built for both mean and variance, where the transition function for the variance is commonly ignored in the existing literature. An explicit model for such a positive random variable not only imposes the auto-correlated uncertainties but also contributes to weighing the recovery cost for the parameter learning objective.

Bibliography

[1] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[2] Leo H Chiang and Richard D Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2):159–178, 2003.

[3] S Joe Qin. Statistical process monitoring: basics and beyond. *Journal of chemometrics*, 17(8-9):480–502, 2003.

[4] Francis J Doyle. Nonlinear inferential control for process applications. *Journal of Process Control*, 8(5):339–353, 1998.

[5] Adilson Jos de Assis and Rubens Maciel Filho. Soft sensors development for on-line bioreactor state estimation. *Computers & Chemical Engineering*, 24(2):1099–1103, 2000.

[6] Yu Miao, Fangwei Xu, Yi Zheng, Biao Huang, John MacGowan, and Aris Espejo. Froth Pipeline Water Content Estimation and Control. *IFAC-PapersOnLine*, 48(8):63–68, 2015.

[7] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt. Data-driven soft sensors in the process industry. *Computers & chemical engineering*, 33(4):795–814, 2009.

[8] Shima Khatibisepehr, Biao Huang, and Swanand Khare. Design of inferential sensors in the process industry: A review of Bayesian methods. *Journal of Process Control*, 23(10):1575–1596, 2013.

[9] Manabu Kano and Koichi Fujiwara. Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications. *Journal of chemical engineering of Japan*, 46(1):1–17, 2013.

[10] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[11] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[12] Alan L Yuille, Peter W Hallinan, and David S Cohen. Feature extraction from faces using deformable templates. *International journal of computer vision*, 8(2):99–111, 1992.

[13] Mark S Nixon and Alberto S Aguado. *Feature extraction & image processing for computer vision*. Academic Press, 2012.

[14] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[15] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[16] Ruomu Tan. *Data-driven Modelling for Process Identification with Flat-topped Gaussian Uncertainty*. PhD Thesis, University of Alberta, 2015.

[17] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[18] Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of vision*, 5(6):9–9, 2005.

[19] Chao Shang, Fan Yang, Xinqing Gao, Xiaolin Huang, Johan AK Suykens, and Dexian Huang. Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. *AIChE Journal*, 61(11):3666–3682, 2015.

[20] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[21] Chunhui Zhao and Biao Huang. A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis. *AIChE Journal*, 64(5):1662–1681, 2018.

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

[23] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[24] Christian F Beckmann and Stephen M Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *Medical Imaging, IEEE Transactions on*, 23(2):137–152, 2004.

[25] Richard Turner and Maneesh Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19(4):1022–1038, 2007.

[26] Weiwu Yan, Huihe Shao, and Xiaofan Wang. Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering*, 28(8):1489–1498, 2004.

[27] Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[29] Adrian Corduneanu and Christopher M Bishop. Variational Bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA, 2001.

[30] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[32] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[33] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[34] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[35] Erik LL Sonnhammer, Gunnar Von Heijne, Anders Krogh, and others. A hidden Markov model for predicting transmembrane helices in protein sequences. In *Ismb*, volume 6, pages 175–182, 1998.

[36] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD Thesis, University of California, Berkeley, 2002.

[37] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

[38] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[39] Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. University of London United Kingdom, 2003.

[40] Simo Srkk. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[41] Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[42] Manabu Kano, Koichi Miyazaki, Shinji Hasebe, and Iori Hashimoto. Inferential control system of distillation compositions using dynamic partial least squares regression. *Journal of Process Control*, 10(2):157–166, 2000.

[43] David Wang, Jun Liu, and Rajagopalan Srinivasan. Data-driven soft sensor approach for quality prediction in a refining process. *Industrial Informatics, IEEE Transactions on*, 6(1):11–17, 2010.

[44] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[45] Manabu Kano, Seunghyun Lee, and Shinji Hasebe. Two-stage subspace identification for softsensor design and disturbance estimation. *Journal of Process Control*, 19(2):179–186, 2009.

[46] Chao Shang, Fan Yang, Xinqing Gao, and Dexian Huang. Extracting latent dynamics from process data for quality prediction and performance assessment via slow feature regression. In *American Control Conference (ACC), 2015*, pages 912–917. IEEE, 2015.

[47] Chao Shang, Biao Huang, Fan Yang, and Dexian Huang. Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling. *AIChE Journal*, 61(12):4126–4139, 2015.

[48] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[49] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[50] David P Wipf and Srikantan S Nagarajan. A new view of automatic relevance determination. In *Advances in neural information processing systems*, pages 1625–1632, 2008.

[51] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[52] Zoubin Ghahramani and Matthew J Beal. Propagation algorithms for variational Bayesian learning. *Advances in neural information processing systems*, pages 507–513, 2001.

[53] Herbert E Rauch, CT Striebel, and F Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[54] Dirk Ostwald, Evgeniya Kirilina, Ludger Starke, and Felix Blankenburg. A tutorial on variational Bayes for latent linear stochastic time-series models. *Journal of Mathematical Psychology*, 60:1–19, 2014.

[55] Jan Beirlant, Edward J Dudewicz, László Györfi, and Edward C Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

[56] Leo H Chiang, Evan L Russell, and Richard D Braatz. Tennessee Eastman Process. In *Fault Detection and Diagnosis in Industrial Systems*, pages 103–112. Springer, 2001.

[57] Christopher V Rao, James B Rawlings, and Jay H Lee. Constrained linear state estimationa moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

[58] Wei He, Yuhao Chen, and Zhao Yin. Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE transactions on cybernetics*, 46(3):620–629, 2016.

[59] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608, 2009.

[60] Tong Zhang, Wenming Zheng, Zhen Cui, Yuan Zong, and Yang Li. Spatial-Temporal Recurrent Neural Network for Emotion Recognition. *IEEE Transactions on Cybernetics*, (99):1–9, 2018.

[61] Xinguang Shao, Biao Huang, and Jong Min Lee. Constrained Bayesian state estimationA comparative study and a new particle filter based approach. *Journal of Process Control*, 20(2):143–157, 2010.

[62] Arjun K Gupta and Saralees Nadarajah. *Handbook of beta distribution and its applications*. CRC press, 2004.

[63] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[64] Dong Dong and Thomas J McAvoy. Nonlinear principal component analysisbased on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1):65–78, 1996.

[65] S Joe Qin. Recursive PLS algorithms for adaptive data modeling. *Computers & Chemical Engineering*, 22(4):503–514, 1998.

[66] S Joe Qin. Neural networks for intelligent sensors and controlpractical issues and some solutions. *Neural Systems for Control*, pages 213–234, 1997.

[67] Sungyong Park and Chonghun Han. A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers & Chemical Engineering*, 24(2):871–877, 2000.

[68] Eliana Zamprogna, Massimiliano Barolo, and Dale E Seborg. Estimating product composition profiles in batch distillation via partial least squares regression. *Control Engineering Practice*, 12(7):917–929, 2004.

[69] Evan L Russell, Leo H Chiang, and Richard D Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 51(1):81–93, 2000.

[70] Pabara-Ebiere Patricia Odiowei and Yi Cao. Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. *IEEE Transactions on Industrial Informatics*, 6(1):36–45, 2010.

[71] Manish Misra, H Henry Yue, S Joe Qin, and Cheng Ling. Multivariate process monitoring and fault diagnosis by multi-scale PCA. *Computers & Chemical Engineering*, 26(9):1281–1293, 2002.

[72] Gang Li, S Joe Qin, and Donghua Zhou. A new method of dynamic latent-variable modeling for process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6438–6445, 2014.

[73] Le Zhou, Gang Li, Zhihuan Song, and S Joe Qin. Autoregressive Dynamic Latent Variable Models for Process Monitoring. 2016.

[74] Vincent Laurain, Marion Gilson, Roland Tth, and Hugues Garnier. Refined instrumental variable methods for identification of LPV BoxJenkins models. *Automatica*, 46(6):959–967, 2010.

[75] Yaojie Lu and Biao Huang. Robust multiple-model LPV approach to nonlinear process identification using mixture t distributions. *Journal of Process Control*, 24(9):1472–1488, 2014.

[76] Minjin Kim, Young-Hak Lee, In-Su Han, and Chonghun Han. Clustering-based hybrid soft sensor for an industrial polypropylene process with grade changeover operation. *Industrial & engineering chemistry research*, 44(2):334–342, 2005.

[77] Yaojie Lu, Biao Huang, and Shima Khatibisepehr. A variational bayesian approach to robust identification of switched arx models. *IEEE transactions on cybernetics*, 46(12):3195–3208, 2016.

[78] Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[79] X Jin and B Huang. Identification of switched Markov autoregressive eXogenous systems with hidden switching state. *Automatica*, 48(2):436–441, 2012.

[80] Nima Sammaknejad, Biao Huang, R Sean Sanders, Yu Miao, Fangwei Xu, and Aris Espejo. Adaptive Soft Sensing and On-line Estimation of the Critical Minimum Velocity with Application to an Oil Sand Primary Separation Vessel. *IFAC-PapersOnLine*, 48(8):211–216, 2015.

[81] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[82] Yujia Zhao, Alireza Fatehi, and Biao Huang. A data-driven hybrid ARX and markov chain modeling approach to process identification with time-varying time delays. *IEEE Transactions on Industrial Electronics*, 64(5):4226–4236, 2017.

[83] G Ackerson and K Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1):10–17, 1970.

[84] Chang-Jin Kim. Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60(1-2):1–22, 1994.

[85] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

[86] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(3):95–108, 1961.

[87] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[88] Yaakov Bar-Shalom, Peter K Willett, and Xin Tian. *Tracking and data fusion.* YBS publishing, 2011.

[89] Yafeng Guo and Biao Huang. Moving horizon estimation for switching nonlinear systems. *Automatica*, 49(11):3270–3281, 2013.

[90] Hajime Akashi and Hiromitsu Kumamoto. Random sampling approach to state estimation in switching environments. *Automatica*, 13(4):429–434, 1977.

[91] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[92] Amir Averbuch, Samuel Itzikowitz, and Tal Kapon. Radar target tracking-Viterbi versus IMM. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):550–563, 1991.

[93] Chaw-Bing Chang and Michael Athans. State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems*, (3):418–425, 1978.

[94] Henk AP Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE transactions on Automatic Control*, 33(8):780–783, 1988.

[95] L Campo, P Mookerjee, and Y Bar-Shalom. State estimation for systems with sojourn-time-dependent Markov model switching. *IEEE Transactions on Automatic Control*, 36(2):238–243, 1991.

[96] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[97] Y Boers and JN Driessen. Interacting multiple model particle filter. *IEE Proceedings-Radar, Sonar and Navigation*, 150(5):344–349, 2003.

[98] Shunyi Zhao, Fei Liu, and Xiaoli Luan. Risk-sensitive filtering for nonlinear Markov jump systems on the basis of particle approximation. *International Journal of Adaptive Control and Signal Processing*, 26(2):158–170, 2012.

[99] Xiao-Rong Li and Yaakov Bar-Shalom. Multiple-model estimation with variable structure. *IEEE Transactions on Automatic control*, 41(4):478–493, 1996.

[100] Leigh A Johnston and Vikram Krishnamurthy. An improvement to the interacting multiple model (IMM) algorithm. *IEEE Transactions on Signal Processing*, 49(12):2909–2923, 2001.

[101] Arnaud Doucet, Andrew Logothetis, and Vikram Krishnamurthy. Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Transactions on Automatic Control*, 45(2):188–202, 2000.

[102] Vesselin P Jilkov and X Rong Li. Online Bayesian estimation of transition probabilities for Markovian jump systems. *IEEE Transactions on signal processing*, 52(6):1620–1630, 2004.

[103] Efim Mazor, Amir Averbuch, Yakov Bar-Shalom, and Joshua Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on aerospace and electronic systems*, 34(1):103–123, 1998.

[104] A. Jasra, C. C. Holmes, and D. A. Stephens. Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science*, 20(1):50–67, 2005.

[105] Mudassir M Rashid, Prashant Mhaskar, and Christopher LE Swartz. Multi-rate modeling and economic model predictive control of the electric arc furnace. *Journal of Process Control*, 40:50–61, 2016.

[106] Alexander T Basilevsky. *Statistical factor analysis and related methods: theory and applications*, volume 418. John Wiley & Sons, 2009.

[107] Peter J Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, pages 799–821, 1973.

[108] Markus Svensn and Christopher M Bishop. Robust Bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2005.

[109] Cdric Archambeau and Michel Verleysen. Robust bayesian clustering. *Neural Networks*, 20(1):129–138, 2007.

[110] Chuanhai Liu and Donald B Rubin. ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, pages 19–39, 1995.

[111] Frank Nielsen and Vincent Garcia. Statistical exponential families: A digest with flash cards. *arXiv preprint arXiv:0911.4863*, 2009.

[112] Matthew D Hoffman and Andrew Gelman. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[113] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[114] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[115] Li Xie, Yu Zhao, Daniel Aziz, Xing Jin, Litao Geng, Errol Goberdhansingh, Fei Qi, and Biao Huang. Soft sensors for online steam quality measurements of OTSGs. *Journal of Process Control*, 23(7):990–1000, 2013.

[116] B KOVACEVIC, Z DUROVIC, and S GLAVASKI. On robust Kalman filtering. *International journal of control*, 56(3):547–562, 1992.

[117] Cl Masreliez and R Martin. Robust Bayesian estimation for the linear model and robustifying the Kalman filter. *IEEE transactions on Automatic Control*, 22(3):361–371, 1977.

[118] Chang Joo Lee, Jung Min Pak, Choon Ki Ahn, Kyung Min Min, Peng Shi, and Myo Taeg Lim. Multi-target FIR tracking algorithm for Markov jump linear systems based on true-target decision-making. *Neurocomputing*, 168:298–307, 2015.

[119] Ayen D Akkaya and Moti L Tiku. Robust estimation in multiple linear regression model with non-Gaussian noise. *Automatica*, 44(2):407–417, 2008.

[120] Weiyu Xu, Er-Wei Bai, and Myung Cho. System identification in the presence of outliers and random noises: A compressed sensing approach. *Automatica*, 50(11):2905–2911, 2014.

[121] Paul Douglas Gilbert. *State space and arma models: An overview of the equivalence*. Bank of Canada, 1993.

[122] Torsten Sderstrm and Petre Stoica. System identification. 1989.

[123] Jean-Pierre Richard. Time-delay systems: an overview of some recent advances and open problems. *automatica*, 39(10):1667–1694, 2003.

[124] Svante Bjrklund. *Experimental evaluation of some cross correlation methods for time-delay estimation in linear systems*. Linkping University Electronic Press, 2003.

[125] Gang Zheng, Andrey Polyakov, and Arie Levant. Delay estimation via sliding mode for nonlinear time-delay systems. *Automatica*, 89:266–273, 2018.

[126] XM Ren, Ahmad B Rad, PT Chan, and Wai Lun Lo. Online identification of continuous-time systems with unknown time delay. *IEEE Transactions on Automatic Control*, 50(9):1418–1422, 2005.

[127] Milena Anguelova and Bernt Wennberg. State elimination and identifiability of the delay parameter for nonlinear time-delay systems. *Automatica*, 44(5):1373–1378, 2008.

[128] G Rao and L Sivakumar. Identification of time-lag systems via Walsh functions. *IEEE Transactions on Automatic Control*, 24(5):806–808, 1979.

[129] Jose Luis Guzmán, Pedro Garcia, Tore Hägglund, Sebastian Dormido, Pedro Albertos, and Manuel Berenguel. Interactive tool for analysis of time-delay systems with dead-time compensators. *Control Engineering Practice*, 16(7):824–835, 2008.

[130] Kenneth L Cooke and Zvi Grossman. Discrete delay, distributed delay and stability switches. *Journal of mathematical analysis and applications*, 86(2):592–627, 1982.

[131] Frederic Mazenc. Stability analysis of time-varying neutral time-delay systems. *IEEE Transactions on Automatic Control*, 60(2):540–546, 2015.

[132] Qing-Long Han. On robust stability of neutral systems with time-varying discrete delay and norm-bounded uncertainty. *Automatica*, 40(6):1087–1092, 2004.

[133] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[134] Daniel Boudreau and Peter Kabal. Joint time-delay estimation and adaptive recursive least squares filtering. *IEEE Transactions on Signal processing*, 41(2):592–601, 1993.

[135] Li Xie, Huizhong Yang, and Biao Huang. FIR model identification of multirate processes with random delays using EM algorithm. *AIChE Journal*, 59(11):4124–4132, 2013.

[136] Yujia Zhao, Alireza Fatehi, and Biao Huang. Robust estimation of ARX models with time varying time delays using variational Bayesian approach. *IEEE transactions on cybernetics*, 48(2):532–542, 2018.

[137] Jonas Mårtensson, Niklas Everitt, and Håkan Hjalmarsson. Covariance analysis in siso linear systems identification. *Automatica*, 77:82–92, 2017.

[138] Yanjun Ma and Biao Huang. Bayesian Learning for Dynamic Feature Extraction with Application in Soft Sensing. *IEEE Transactions on Industrial Electronics*, 2017.

[139] Yanjun Ma, Seraphina Kwak, Lei Fan, and Biao Huang. A variational bayesian approach to modelling with random time-varying time delays. In *2018 Annual American Control Conference (ACC)*, pages 5914–5919. IEEE, 2018.

[140] Tara Baldacchino, Sean R Anderson, and Visakan Kadirkamanathan. Computational system identification for Bayesian NARMAX modelling. *Automatica*, 49(9):2641–2651, 2013.

[141] Fredrik Lindsten, Thomas B Schn, and Michael I Jordan. Bayesian semiparametric Wiener system identification. *Automatica*, 49(7):2053–2063, 2013.

[142] Chi Tim Ng and Harry Joe. Generating random AR (p) and MA (q) Toeplitz correlation matrices. *Journal of Multivariate Analysis*, 101(6):1532–1545, 2010.

[143] Søren Johansen and Katarina Juselius. Maximum likelihood estimation and inference on cointegrationwith applications to the demand for money. *Oxford Bulletin of Economics and statistics*, 52(2):169–210, 1990.

[144] He Sun, Shumei Zhang, Chunhui Zhao, and Furong Gao. A sparse reconstruction strategy for online fault diagnosis in nonstationary processes with no a priori fault information. *Industrial & Engineering Chemistry Research*, 56(24):6993–7008, 2017.

[145] Chunhui Zhao and Biao Huang. A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis. *AIChE Journal*, 64(5):1662–1681, 2018.

[146] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[147] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[148] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[149] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.