

“This is ten percent luck, twenty percent skill, fifteen percent concentrated power of will, five percent pleasure, fifty percent pain, and a hundred percent reason to remember the name!”

Fort Minor – “Remember The Name”

University of Alberta

**Modeling and Development of Soft Sensors with Particle
Filtering Approach**

by

Jing Deng

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

©Jing Deng

Spring 2012

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

To my beloved Dad, Mom and Grandfather, who give me
unconditional love and support through my life

Abstract

Limitations of measurement techniques and increasingly complex chemical process render difficulties in obtaining certain critical process variables. The hardware sensor reading may have an obvious bias compared with the real value. Off-line laboratory analysis with high accuracy can only be obtained every certain period, sometimes even with time delay. Soft sensors are inferential methods that provide real-time estimation for those critical variables. This thesis deals with modeling, on-line calibration and implementation issues that are associated with soft sensor development.

In chemical industries, processes are often designed to perform tasks under various operating conditions. In order to deal with modeling difficulties rendered by multiple operating conditions, the Expectation-Maximization (EM) algorithm is applied to deal with the identification problem of nonlinear parameter varying systems. The existing model is updated using the latest observation data. The particle filter based Bayesian method is proposed in this thesis to synthesize different sources of measurement information. An augmented state is constructed to deal with processes with time delay problem.

Acknowledgements

I am blessed to be a member of Computer Process Control Group at the University of Alberta, where I gained numerous help during my efforts toward my M.Sc degree. Special thanks go to my supervisor Dr. Biao Huang from whom I have benefited a lot for his rigorous research attitude and insightful advice. Without his constant support and guidance, this thesis is impossible to be accomplished. I am grateful to his patience and encouragement at the starting point of my master study and during my past two years' research work. The opportunities to explore applications in different industrial processes are also greatly appreciated.

The help from Computer Process Control Group is precious to me. I would like to thank Yuri Shardt for his instruction and support with the experimental setups of the three-tank system and four-tank system. I would also like to thank Xing Jin for his help and advice whenever I met obstacle during my research time. The help from Ruben Gonzalez with the proof reading of this thesis is also highly appreciated. It was pleasant and memorable experience being able to work with the colleagues in CPC group. I would like thank Fei Qi, Xinguang Shao, Aditya Tulsyan, Yu Zhao, Li Xie, Yijia Zhu, Da Zheng, William Weng and each other group member. I wish you all the best in your future endeavor.

Specially, the financial support from Syncrude Canada Ltd is greatly acknowledged. It was a valuable experience to evaluate and apply the proposed method in this thesis to the oil sands process at Syncrude Canada Ltd. I would like to thank Dr. Fangwei Xu, Dr. Kwanho Lee and Dr. Miao Yu for their guidance and support during the implementation work of the proposed method. The discussion with Shima Khatibisepehr was also an enjoyable experience. In addition, I would like to thank Aris Espejo and Dan Brown for their constant support to my collaborative research project with Syncrude Canada Ltd..

I feel deeply lucky to be surrounded by friendship during the past few years. I would like to thank all my dear friends who gave me love, encouragement and support. Especially Ling Zhang, Ge Li, Yi Sun, Choon Ngan, Weiyuan Jiang and Rui Shen, thanks for standing by my side all the time and I will treasure our friendship for the rest of my life.

Lastly, I thank my beloved parents and grandfather for their unconditional love and support. They have always been open-minded in my pursuit and choices which has shaped me into an independent person. Thank them for trying best to create me a pleasant growing environment, for giving me the freedom in pursuing my dream, and for understanding me not being able to accompany them for these years. Words cannot express my gratitude to them.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	2
1.3	Thesis Outline	3
2	Identification of Nonlinear Parameter Varying Systems with Missing Output Data	4
2.1	Introduction	4
2.2	Problem Statement	6
2.3	Expectation-Maximization Algorithm	8
2.3.1	EM algorithm revisit	8
2.3.2	Formulation of the multiple model parameter estimation based on the EM algorithm	8
2.4	Computation through Particle Filtering	13
2.4.1	Particle filters revisit	13
2.4.2	Particle filters approximation and cautious resampling	14
2.5	Simulations and Pilot-scale Experiment	18
2.5.1	A numerical simulation example	18
2.5.2	Continuous stirred tank reactor	22
2.5.3	Experimental evaluation: a multi-tank system	26
2.6	Discussion and Conclusion	31
3	Inferential Sensor Development	36
3.1	Introduction	36
3.2	Parameter Estimation of the State Space Model	38
3.2.1	Model structure	38
3.2.2	Formulation of the parameter estimation based on the EM algorithm	39
3.2.3	Computation through particle filtering	40
3.3	Bayesian Calibration	43
3.3.1	Formulation for online calibration	43
3.4	Semi-continuous Fermentation Example	45
3.5	Industrial Application	47

3.5.1	Process description	47
3.5.2	Soft sensor development for PSV underflow bitumen content	51
3.5.3	Performance index	57
3.5.4	Online implementation	59
3.6	Conclusion	60
4	Soft Sensor Development for Time Delayed Processes with Bayesian Approach	62
4.1	Introduction	62
4.2	Problem Statement	63
4.3	Bayesian Approach to Deal with Time Delay	64
4.3.1	Problem formulation	64
4.3.2	Formulation for online calibration	66
4.3.3	Calibration through particle filtering	67
4.4	A Numerical Simulation Example	67
4.5	Semi-continuous Fermentation Example	70
4.6	Conclusion	73
5	Conclusions	76
5.1	Summary of this thesis	76
5.2	Recommendations for future work	77

List of Tables

2.1	Estimated parameter a after 150 iterations	21
2.2	Estimated parameter b after 150 iterations	21
2.3	CSTR model parameters and their steady state values	25
2.4	Designed operating point for the experiment	29
3.1	MSE comparison of different measurements	53
3.2	MAE comparison of different measurements	54

List of Figures

2.1	Trajectory of the scheduling variable H	20
2.2	Trajectories of estimated parameter a for each local model when 25% observations are missing.	21
2.3	Trajectories of estimated parameter b for each local model when 25% observations are missing.	22
2.4	Validation of the identified global model against the model training data set.	23
2.5	Cross validation of the identified global model.	23
2.6	Weight of each local model at different operating points	24
2.7	Trajectory of the scheduling variable H	24
2.8	Validation of the identified global model against the model training data set. Blue line represent the real process output and the red line is the simulated output from the identified global model	26
2.9	Cross validation of the identified global model. Blue line represent the real process output and the red line is the simulated output from the identified global model	27
2.10	Weight of each local model at different operating points	27
2.11	Three tank system schematic	28
2.12	Three tank system scheduling variable for the self validation .	29
2.13	Three tank system input-output data (a) water level of the second tank, process output (b) valve position $V2$, process input	31
2.14	Self-validation result. Blue solid line is the collected process data, red dash line is the simulated output of the identified global model	31
2.15	Three tank system input-output data (a) water level of the second tank, process output (b) valve position $V2$, process input	32
2.16	Cross-validation result. Blue solid line is the collected process data, red dash line is the simulated output of the identified global model	32
3.1	The trajectory of the variance of lab analysis	44
3.2	Process inputs. (a): u_1 , the dilution factor h^{-1} ; (b): u_2 , the substrate concentration in the feed.	46

3.3	Estimated parameters. Blue line: trajectory of θ_1 ; Red line: trajectory of θ_3	46
3.4	Comparison of current measurements.	47
3.5	Comparison of different measurements.	48
3.6	45 degree comparison.	48
3.7	Schematic diagram of the Primary Separation Vessel	49
3.8	Comparison between online analyzer and laboratory analysis.	50
3.9	Input data for the PSV unit	52
3.10	Model cross validation result	53
3.11	Trend comparison.	54
3.12	45 degree comparison.	54
3.13	Trend comparison.	55
3.14	45 degree comparison.	55
3.15	Trend comparison.	56
3.16	45 degree comparison.	56
3.17	Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.	57
3.18	Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.	58
3.19	Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.	58
3.20	Distribution of historical lab data. Red line indicates that 95% of the historical lab data is less or equal than 1.1. For proprietary reason, the data has been normalized.	59
3.21	Soft sensor online monitoring result.	60
4.1	Slow sampled lab data without time delay.	64
4.2	(a): Slow sampled lab data without time delay; (b): Slow sampled lab data with 3 sampling time delay.	65
4.3	Fast process measurement and slow measurement with time delay.	68
4.4	Model prediction without Bayesian calibration	69
4.5	Model prediction with Bayesian calibration.	70
4.6	Fast process measurement and slow measurement with time delay.	71
4.7	Process inputs. (a): u_1 , the dilution factor h^{-1} ; (b): u_2 , the substrate concentration in the feed.	72
4.8	Comparison of current measurements.	72
4.9	Comparison of different measurements.	73

4.10 45 degree comparison.	74
------------------------------------	----

Chapter 1

Introduction

1.1 Motivation

Obtaining real-time and accurate measurement of critical process variables is of great importance from the perspectives of both process monitoring and process control. The availabilities of some key variables are often restricted by the limitation of measurement technique or high installation cost. The existing on-line instrument could have large bias compared with the true value and may even encounter technical failure. Sometimes off-line laboratory analysis is performed in accompany with the on-line measurement. Although with higher accuracy, the off-line analysis can only be available every certain period. Therefore it is desired to develop a measurement mechanism which can synthesize various sources of measurements.

Soft sensors are inferential techniques which provide real-time estimation for key process variable by making use of information of readily-available secondary process variables. An important step for soft sensor development is process modeling. Linear modeling techniques such as Ordinary Least Square Regression (OLSR), Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR) have been quite mature over the past few decades. Nonlinear process modeling such as Nonlinear Autoregressive eXogenous (NARX), Artificial Neural Network and Wiener or Hammerstein models have also been widely applied. Nowadays, with the increase of the process complexity, chemical plants are often driven to operate under different operating conditions. Conventional single-model modeling method appears insufficient in describing the processes with multiple operating condition. As a result, researchers have developed various multiple modeling methods to deal with the limitations of conventional single-model base modeling techniques [3], [4] and [17]. These modeling methods adopt linear model structure at each operating point and approximate the nonlinear process by interpolating different local linear models. In this thesis, a nonlinear state space model based multiple modeling method is developed. The identification of nonlin-

ear parameter varying system using particle filter under the framework of the Expectation-Maximization (EM) algorithm is discussed in Chapter 2. Meanwhile, the missing output data problem is also considered, and is dealt with by particle filtering approximation.

Models that are built based on physical principle such as mass balance and energy balance are called first principle models. In practice, some variables are difficult to measure due to the increasing process complexity and the limitation of instrumentation, leading to the unavailability of first principle models. Black box models (data-driven models) which are developed from the historical data have gained a lot of attention during recent years. As a typical model structure, the state space model is considered in detail in Chapter 3 and the parameter estimation of the nonlinear state space model is discussed.

One drawback of data-driven models is that there will always exists model-plant mismatch especially when there is a sudden change in the operating condition. Therefore, it is desired to update the existing model with the newest observation. In Chapter 3, the Bayesian model calibration method is discussed in detail. With the given model, a Bayesian calibration framework with model calibration parameters is constructed. This formulation makes use of different output measurement sources and provides estimation of the process state.

Another challenge that exists in process industries particularly in oil sands industry is that lab measurements have time delay before arriving at the distributed control systems. This problem is called Out Of Sequence Measurement (OOSM) problem [3]. One step or multi steps delay problem has been solved using backward prediction which applied the inverse model of the process to the current state. However, this method is only suitable for linear process models while other solution is needed for nonlinear process models [3], [4]. Challa et al. [5] deals with the OOSM problem using the augmented state Kalman Filter (ASKF) where the uncertainty of the delay was resolved by means of Probabilistic Data Association Filter (PDAF). Chapter 4 of this thesis deals with the infrequently sampled measurement along with time delay problem.

1.2 Contributions

The main contributions of this thesis are listed below:

1. Formulate and solve the identification problem of nonlinear parameter varying system under the framework of the EM algorithm.
2. The missing output problem is addressed by particle filters approximation which is computational efficient.
3. Formulate a Bayesian model calibration strategy which synthesizes different measurement sources.

4. Evaluation of the proposed Bayesian calibration method is performed on a semi-fermenter example as well as an oil sands process.
5. Formulate an augmented state to deal with the time delayed measurement problem.

1.3 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 deals with the identification problem for nonlinear parameter varying system using particle filter under the framework of the expectation-maximization (EM) algorithm. The proposed method is validated through both simulation example and an experiment performed on a pilot-scale setup. Chapter 3 formulates the Bayesian calibration framework which synthesizes both fast-rate sampled and slow-rate sampled measurements to update the model prediction. This approach is validated through a fermenter simulation example and is applied to an oil sands industrial process. In Chapter 4 which deals with the output time delay issue, an augmented state is constructed and the Bayesian calibration proposed in Chapter 3 is applied to estimate the augmented state. The validation is performed based on a numerical simulation and a semi-fermenter example. The thesis is concluded in Chapter 5 which summarizes the work that has been done in this thesis and provides some suggestions for the future research.

This thesis is organized in the paper format. There may be some overlap between the chapters for sake of completeness of each chapter.

Chapter 2

Identification of Nonlinear Parameter Varying Systems with Missing Output Data

¹ This chapter is concerned with identification of nonlinear parameter varying systems using particle filter under the framework of the expectation-maximization (EM) algorithm. In chemical industries, processes are often designed to perform tasks under various operating conditions. In order to circumvent the modeling difficulties rendered by multiple operating conditions and the transitions between different working points, the EM algorithm, which iteratively increases the likelihood function, is applied. Meanwhile the missing output data problem which is common in real industry is also considered in this work. Particle filters are adopted to deal with the computation of expectation functions. The efficiency of the proposed method is illustrated through simulated examples and a pilot-scale experiment.

2.1 Introduction

Over the past few decades, the research of parameter estimation for nonlinear processes has witnessed rapid progress as it plays a key role in the development of mathematical models to describe process behavior. Aspects of parameter estimation and system identification have been discussed extensively in literature [1]. Linear modeling techniques have been quite mature over the past few decades. Nonlinear process modeling such as Nonlinear Autoregressive eXogenous (NARX), Artificial Neural Network and Wiener or Hammerstein models have also been widely applied. However, studies on parameter estimation for nonlinear parameter varying system have been sparse. The inborn

¹Deng, J., Huang, B.. Identification of nonlinear parameter varying systems with missing output data. *Submitted to AICHE Journal*, 2011.

nonlinearity of the chemical processes and the production complexity brought by various working conditions have both increased the estimation difficulties. In general, chemical processes may behave differently when performing different production tasks. This includes the feed raw material property changes, a varying grade in polymer plants, or reaction load changes, etc.

To overcome the limitations of conventional single-model based modeling techniques, researchers have developed various multiple modeling strategies. Shamma et al. [3] first introduced a Linear Parameter Varying (LPV) modeling method, which is featured by its linear structure and varying model parameters. Due to its capability in approximating nonlinear process, the LPV modeling method has drawn growing attention from researchers. An LPV modeling method was put forwarded by Xu et al. [4] in their study of nonlinear MPC. The process is tested around each operating trajectory and the global LPV model is identified by interpolating each linear local models using all available data. Jin and Huang [5] proposed an LPV modeling method under the framework of the expectation and maximization (EM) algorithm, which identified the LPV models using all data points collected from the experiments. Their work only considered the linear input-output ARX model as the local models to approximate a global nonlinear model.

Nonlinear state space model is a general class of models to represent nonlinear dynamic systems. Maximum likelihood estimation of nonlinear parameter invariant state space models has been studied by Schon et al [8]. However, the process often operates over various conditions which render different model parameters. The work conducted in this chapter aims at the identification of parameter varying nonlinear state space models.

On the other hand, missing data or irregularly sampled data is commonly observed in industrial practice. Parameter estimation of nonlinear dynamic models in the presence of missing observation has not been well studied. Missing data could be caused by a sudden mechanical breakdown, hardware sensor failure or data acquisition system malfunction, etc. Another increasing common source for this missing data problem is the integration of communication networks in process control systems and the subsequent potential for data losses and packet dropouts.

Some common approaches in dealing with missing data have been presented and summarized in Khatibisepehr's work [6]. One intuitive way known as case-wise deletion is to simply exclude the records that contain missing values. Its major drawback is that some informative data may also be thrown out in the meantime. In many chemical industries, for example, process variables such as flow rate, temperature, stream density are frequently sampled while the key quality variable like composition which often is of the most interest can only be obtained after hours of laboratory analysis. Arbitrarily removing the records where lab data is not available leads to a loss of useful information contained in fast sampled variables. Another popular treatment of incomplete data set is

called imputation, including mean substitution, regression imputation, multiple imputation, etc. These methods, by their names, replace all missing value with the mean of that variable or the prediction using information gained from other data, which appears to be attractive in the sense that it preserves the complete data size. Nevertheless, as pointed out by Khatibisepehr [6], the variances of the data may be considerably changed with imputation .

The work by Gopaluni [9] is an important step towards identification of parameter invariant nonlinear models with missing observations, where the EM algorithm is adopted for dealing with missing data and hidden state, and the particle filter based smoother is applied for computation of the expectation functions. This chapter extends the work of Gopaluni [9] by considering parameter varying nonlinear systems. In addition, missing output and parameter varying problems in nonlinear state space model parameter estimation are solved simultaneously under the framework EM algorithm. Particle filters are employed for computation of expectation functions. The use of particle filter rather than the smoother significantly reduces the computation load.

The remainder of this chapter is organized as follows: Section 2 states the identification problem of parameter varying nonlinear state space models with missing output data. Section 3 begins with a revisit of the EM algorithm and the derivation of the expression for Q function for nonlinear parameter varying state space models with missing output data is given. Section 4 provides a brief description of particle filters and the detail of evaluating the Q function using particle filters is presented. Numerical simulations as well as an experimental example are illustrated in Section 5 which aim at demonstrating the effectiveness of the proposed method in nonlinear parameter varying system identification with missing output data. Section 6 draws the conclusion based on the results obtained in this chapter.

2.2 Problem Statement

Many industrial processes are often operated in certain “orderly” ways to meet different production objectives. Such orderly ways are also referred as operating trajectory which consists of several predesigned operating points. In this chapter, we use “H” to denote the operating variable according to which the process is operated.

Consider the nonlinear state space model given by

$$x_t = f(x_{t-1}, u_{t-1}, \Theta) + \omega_t \quad (2.1)$$

$$y_t = h(x_t, \Theta) + v_t \quad (2.2)$$

where the system parameters are Θ which are functions of the scheduling variable H such that $\Theta = g(H)$. Assume that J operating points are pre-defined such that at each $H_i, i = 1, 2, \dots, J$, the process has different parameters in

its (nonlinear) model, and each local set of parameters $\theta_i, i = 1, 2, \dots, J$ are to be estimated. x_t, u_t, y_t, ω_t and v_t are state, measured input, measured output, process noise and measurement noise, respectively; ω_t and v_t are independent and identically distributed Gaussian noises with covariance matrices Q and R respectively. The input sequence $\{u_1, \dots, u_T\}$ and the trajectory of scheduling variable $\{H_1, \dots, H_T\}$ are known.

Let X denote the sequence of hidden states $\{x_1, \dots, x_T\}$. The outputs are available at time $\{t_1, \dots, t_\alpha\}$ while missing at time $\{m_1, \dots, m_\beta\}$. $Y_o = \{y_{t_1}, \dots, y_{t_\alpha}\}$ and $Y_m = \{y_{m_1}, \dots, y_{m_\beta}\}$ stand for the corresponding observed outputs set and missing outputs set. It is assumed that the data is missing completely at random (MCAR) [6]. In other words, the probability that data missing mechanism does not depend on any part of the observed data or missing data. The nonlinear model structures in Equation 2.1 and Equation 2.2 are known a priori.

Due to the varying operating condition, a single nonlinear model is not sufficient to represent the process dynamics. Therefore, a global nonlinear model which is a weighted interpolation of each local nonlinear model is adopted as follows

$$x_t = \sum_{j=1}^J \alpha_{tj} x_{tj} \quad (2.3)$$

$$y_t = \sum_{j=1}^J \alpha_{tj} y_{tj} \quad (2.4)$$

An exponential weighting function is employed here to denote the weight for each local model [5]

$$\omega_{tj} = \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \quad (2.5)$$

and the normalized weight α can be derived as

$$\alpha_{tj} = \frac{\omega_{tj}}{\sum_{j=1}^J \omega_{tj}} \quad (2.6)$$

where σ_j represents the validity width for each local model which is bounded by σ_{min} (the lower bound for $\sigma_j, j = 1 : J$) and σ_{max} (the upper bound for $\sigma_j, j = 1 : J$). Hence the parameters of each local state space model, $\theta_j, j = 1 : J$, as well as the model validity $\sigma_j, j = 1 : J$, are of interest. In the following, we will show how to formulate the parameter estimation problem under the scheme of the EM algorithm.

2.3 Expectation-Maximization Algorithm

2.3.1 EM algorithm revisit

Expectation-maximization (EM) algorithm [7] is a well-known maximum likelihood based method, which iterates between two steps, the expectation step and maximization step. The basic principle behind the EM algorithm is that instead of performing a direct optimization of the likelihood of the observed data, which is typically not tractable, one augments the observed data set C_{obs} with missing data set C_{mis} to perform a series of iterative optimizations. In the EM procedure both the complete data log-likelihood, $\log[(C_{obs}, C_{mis}|\Theta)]$ and the conditional predictive distribution, $p(C_{mis}|C_{obs}, \Theta)$, are calculated. Consisting of two steps, namely the expectation step (E-step) and the maximization step (M-step), the EM algorithm proceeds as follows:

Let Θ^k be the current best approximation to the mode of the observed posterior or the best estimated parameters using all available data. With the parameters currently available and data that are observed, the distribution function of the missing observations may be determined. Based on the distribution function the expectation of the complete data with the expectation taking over the missing observation can be derived, which is known as the Q function. The E-step is to compute the Q function which is defined by

$$\begin{aligned} Q(\Theta|\Theta^k) &= E_{C_{mis}|C_{obs}, \Theta^k} \{ \log[p(C_{obs}, C_{mis}|\Theta)] \} \\ &= \int_{C_{mis}} \log[p(C_{obs}, C_{mis}|\Theta)] p(C_{mis}|C_{obs}, \Theta^k) dC_{mis} \end{aligned} \quad (2.7)$$

and the M-step is to maximize the Q function with respect to Θ to obtain

$$\Theta^{k+1} = \arg \max_{\Theta} Q(\Theta|\Theta^k) \quad (2.8)$$

The E-step and M-step iterate until convergence.

2.3.2 Formulation of the multiple model parameter estimation based on the EM algorithm

Consider the state-space model described in Equation 2.1 and Equation 2.2. A hidden variable I_t is introduced to represent the identity of the sub model which takes effect at time t . The observed data set C_{obs} are $Y_o = \{y_{t_1}, \dots, y_{t_\alpha}\}$, $\{u_1, \dots, u_T\}$ and $\{H_1, \dots, H_T\}$, while the hidden states $X = \{x_1, \dots, x_T\}$, the hidden model identity $I = \{I_1, \dots, I_T\}$ and the missing outputs $Y_m = \{y_{m_1}, \dots, y_{m_\beta}\}$ can be viewed as the latent or missing data C_{mis} . Since input sequence $\{u_1, \dots, u_T\}$ are considered known, it will not play a role in the following derivation and will be omitted for simplicity. Let $p(C_{obs}, C_{mis}|\Theta)$ denote the complete likelihood function including both the hidden states and

observations. The Q function is defined as the expectation of the log-likelihood function $\log[p(x_{1:T}, I_{1:T}, H_{1:T}, y_{1:T}|\Theta)]$ with respect to all latent variables or data which is given by

$$\begin{aligned} Q(\Theta|\Theta^k) &= E_{C_{mis}|C_{obs},\Theta^k} \{ \log[p(C_{obs}, C_{mis}|\Theta)] \} \\ &= E_{x_{1:T}, I_{1:T}, Y_m | C_{obs}, \Theta^k} \{ \log[p(Y_o, Y_m, H_{1:T}, x_{1:T}, I_{1:T}|\Theta)] \} \\ &= E_{x_{1:T}, I_{1:T}, Y_m | C_{obs}, \Theta^k} \{ \log[p(y_{1:T}, H_{1:T}, x_{1:T}, I_{1:T}|\Theta)] \} \end{aligned} \quad (2.9)$$

In Equation 2.9, the term $p(y_{1:T}, H_{1:T}, x_{1:T}, I_{1:T}|\Theta)$ which is the joint density function of the full data set can be decomposed using the Bayesian property as

$$\begin{aligned} &p(y_{1:T}, H_{1:T}, x_{1:T}, I_{1:T}|\Theta) \\ &= p(y_{1:T}|H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(H_{1:T}, x_{1:T}, I_{1:T}|\Theta) \\ &= p(y_{1:T}|H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(x_{1:T}|H_{1:T}, I_{1:T}, \Theta)p(H_{1:T}, I_{1:T}|\Theta) \\ &= p(y_{1:T}|H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(x_{1:T}|H_{1:T}, I_{1:T}, \Theta)p(I_{1:T}|H_{1:T}, \Theta) \cdot p(H_{1:T}|\Theta) \end{aligned} \quad (2.10)$$

The first term can be further written as

$$\begin{aligned} &p(y_{1:T}|H_{1:T}, x_{1:T}, I_{1:T}, \Theta) \\ &= p(y_T|y_{1:T-1}, H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(y_{1:T-1}|H_{1:T}, x_{1:T}, I_{1:T}, \Theta) \\ &= p(y_T|y_{1:T-1}, H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(y_{T-1}|y_{1:T-2}, H_{1:T}, x_{1:T}, I_{1:T}, \Theta) \\ &\quad \dots p(y_2|y_1, H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(y_1|H_{1:T}, x_{1:T}, I_{1:T}, \Theta) \\ &= p(y_T|x_T, I_T, \Theta)p(y_{T-1}|x_{T-1}, I_{T-1}, \Theta) \dots p(y_1|x_1, I_1, \Theta) \\ &= \prod_{t=1}^T p(y_t|x_t, \Theta_{I_t}) \end{aligned} \quad (2.11)$$

where in the derivation of Equation 2.11, we have used Markov property and the relation that given the model identity I, the conditional distribution of y is independent of the scheduling variable H . Similarly the second term can be simplified to

$$\begin{aligned} &p(x_{1:T}|H_{1:T}, I_{1:T}, \Theta) \\ &= p(x_T|x_{1:T-1}, H_{1:T}, I_{1:T}, \Theta)p(x_{1:T-1}|H_{1:T}, I_{1:T}, \Theta) \\ &= p(x_T|x_{1:T-1}, H_{1:T}, I_{1:T}, \Theta)p(x_{T-1}|x_{1:T-2}, H_{1:T}, I_{1:T}, \Theta) \\ &\quad \dots p(x_2|x_1, H_{1:T}, x_{1:T}, I_{1:T}, \Theta)p(x_1|H_{1:T}, x_{1:T}, I_{1:T}, \Theta) \\ &= p(x_T|x_{T-1}, I_T, \Theta)p(x_{T-1}|x_{T-2}, I_{T-1}, \Theta) \dots p(x_2|x_1, I_1, \Theta) \\ &\quad \cdot p(x_1|I_1, \Theta) \\ &= p(x_1|\Theta_{I_1}) \prod_{t=2}^T p(x_t|x_{t-1}, \Theta_{I_t}) \end{aligned} \quad (2.12)$$

where in the derivation of Equation 2.12, the Markov property about the state has also been applied. The third term can be derived below:

$$\begin{aligned}
& p(I_{1:T}|H_{1:T}, \Theta) \\
&= p(I_T|I_{1:T-1}, H_{1:T}, \Theta)p(I_{1:T-1}|H_{1:T}, \Theta) \\
&= p(I_T|I_{1:T-1}, H_{1:T}, \Theta)p(I_{T-1}|I_{1:T-2}, H_{1:T}, \Theta) \cdots p(I_1|H_{1:T}, \Theta) \\
&= p(I_T|H_T, \Theta)p(I_{T-1}|H_{T-1}, \Theta) \cdots p(I_1|H_1, \Theta) \\
&= \prod_{t=1}^T p(I_t|H_t, \Theta_t) \tag{2.13}
\end{aligned}$$

Derivation of the third term has used the fact that the distribution of the model identity is completely determined by the scheduling variable H_t . Substituting Equation 2.11, Equation 2.12 and Equation 2.13 into Equation 3.8, the joint density of the likelihood of the full data set can be rewritten as

$$\begin{aligned}
& p(y_{1:T}, H_{1:T}, x_{1:T}, I_{1:T}|\Theta) \\
&= \prod_{t=1}^T (p(y_t|x_t, \Theta_t)p(I_t|H_t, \Theta_t)) \cdot p(x_1|\Theta_{I_1}) \prod_{t=2}^T p(x_t|x_{t-1}, \Theta_{I_t}) \cdot C \tag{2.14}
\end{aligned}$$

where $C = p(H_{1:T}|\Theta)$ is considered as a constant since the trajectory of the scheduling variable $\{H_1 \dots H_T\}$ is known and does not depend on Θ . Furthermore, substituting Equation 3.14 into Equation 2.9, the Q function can be rearranged as

$$\begin{aligned}
& Q(\Theta|\Theta^k) \\
&= E_{x_{1:T}, I_{1:T}, Y_m|C_{obs}, \Theta^k} \left\{ \log \left[\prod_{t=1}^T (p(y_t|x_t, \Theta_t)p(I_t|H_t, \Theta_t)) \cdot p(x_1|\Theta_{I_1}) \right. \right. \\
&\quad \left. \left. \prod_{t=2}^T p(x_t|x_{t-1}, \Theta_{I_t}) \cdot C \right] \right\} \\
&= E_{x_{1:T}, I_{1:T}, Y_m|C_{obs}, \Theta^k} \left\{ \sum_{t=1}^T [\log p(y_t|x_t, \Theta_t) + \log p(I_t|H_t, \Theta_t)] + \log p(x_1|\Theta_{I_1}) \right. \\
&\quad \left. + \sum_{t=2}^T \log p(x_t|x_{t-1}, \Theta_{I_t}) + \log C \right\} \tag{2.15}
\end{aligned}$$

Since the possible set of the operating points are known a priori, the expectation can be taken over the discrete variable I_t first as:

$$\begin{aligned}
Q(\Theta|\Theta^k) &= E_{x_{1:T}, Y_m | C_{obs}, \Theta^k} \left\{ \sum_{t=1}^T \sum_{j=1}^J \log p(y_t | x_t, \Theta_j) \cdot p(I_t = j | C_{obs}, \Theta^k) \right. \\
&\quad + \sum_{t=1}^T \sum_{j=1}^J \log p(I_t = j | \sigma_j, H_t) \cdot p(I_t = j | C_{obs}, \Theta^k) \\
&\quad + \sum_{j=1}^J \log p(x_1 | \Theta_1) \cdot p(I_t = j | C_{obs}, \Theta^k) \\
&\quad + \sum_{t=2}^T \sum_{j=1}^J \log p(x_t | x_{t-1}, \Theta_j) \cdot p(I_t = j | C_{obs}, \Theta^k) \\
&\quad \left. + \sum_{j=1}^J \log C \cdot p(I_t = j | C_{obs}, \Theta^k) \right\} \tag{2.16}
\end{aligned}$$

Then the expectation is further taken over continuous variables states $X(x_{1:T})$ and missing observations Y_m :

$$\begin{aligned}
&= \int_{X, Y_m} \sum_{t=1}^T \sum_{j=1}^J \log p(y_t | x_t, \Theta_j) \cdot p(I_t = j | C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta} | C_{obs}, \Theta^k) \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&\quad + \int_{X, Y_m} \sum_{t=1}^T \sum_{j=1}^J \log p(I_t = j | \sigma_j, H_t) \cdot p(I_t = j | C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta} | C_{obs}, \Theta^k) \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&\quad + \int_{X, Y_m} \sum_{j=1}^J \log p(x_1 | \Theta_1) \cdot p(I_t = j | C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta} | C_{obs}, \Theta^k) \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&\quad + \int_{X, Y_m} \sum_{t=2}^T \sum_{j=1}^J \log p(x_t | x_{t-1}, \Theta_j) \cdot p(I_t = j | C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta} | C_{obs}, \Theta^k) \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&\quad + \int_{X, Y_m} \sum_{j=1}^J \log C \cdot p(I_t = j | C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta} | C_{obs}, \Theta^k) dx_{1:T} dy_{m_1:m_\beta} \tag{2.17}
\end{aligned}$$

Given the previous division of the output into observed and missing subset Y_o and Y_m , the derivation can be continued as

$$\begin{aligned}
&= \sum_{t=t_1}^{t=t_\alpha} \sum_{j=1}^J \int_X \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot \left[\int_{Y_m} p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \right. \\
&\quad \left. dy_{m_1:m_\beta} \right] dx_{1:T} \\
&+ \sum_{t=m_1}^{t=m_\beta} \sum_{j=1}^J \int_{X, Y_m} \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&+ \sum_{t=1}^{t=T} \sum_{j=1}^J \int_X \log p(I_t = j|\sigma_j, H_t) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot \left[\int_{Y_m} p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \right. \\
&\quad \left. dy_{m_1:m_\beta} \right] \cdot dx_{1:T} \\
&+ \sum_{j=1}^J \int_X \log p(x_1|\Theta_1) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot \left[\int_{Y_m} p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \right. \\
&\quad \left. dy_{m_1:m_\beta} \right] dx_{1:T} \\
&+ \sum_{t=2}^{t=T} \sum_{j=1}^J \int_X \log p(x_t|x_{t-1}, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot \left[\int_{Y_m} p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \right. \\
&\quad \left. dy_{m_1:m_\beta} \right] \cdot dx_{1:T} \\
&+ \sum_{j=1}^J \int_X \log C \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot \left[\int_{Y_m} p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) dy_{m_1:m_\beta} \right] dx_{1:T} \\
&= \sum_{t=t_1}^{t=t_\alpha} \sum_{j=1}^J \int_X \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_{1:T}|C_{obs}, \Theta^k) dx_{1:T} \\
&+ \sum_{t=m_1}^{t=m_\beta} \sum_{j=1}^J \int_{X, Y_m} \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k) \cdot \\
&\quad dx_{1:T} dy_{m_1:m_\beta} \\
&+ \sum_{t=1}^{t=T} \sum_{j=1}^J \log p(I_t = j|\sigma_j, H_t) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{j=1}^J \int_X \log p(x_1|\Theta_1) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_1|C_{obs}, \Theta^k) dx_1 \\
&+ \sum_{t=2}^{t=T} \sum_{j=1}^J \int_X \log p(x_t|x_{t-1}, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_{1:T}|C_{obs}, \Theta^k) dx_{1:T} \\
&+ \sum_{j=1}^J \log C \cdot p(I_t = j|C_{obs}, \Theta^k) \tag{2.18}
\end{aligned}$$

The probability of the j^{th} local model taking effect at the t^{th} sampling time $p(I_t = j|C_{obs}, \Theta^k)$ can be calculated as

$$p(I_t = j|C_{obs}, \Theta^k) = \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \quad (2.19)$$

where H_t denotes the measurement of the scheduling variable at time t , H_j is the j^{th} operating point and σ_j represents the validity width of the j^{th} local model.

In order to evaluate the Q function in Equation 2.18, the values of density functions $p(x_{1:T}|C_{obs}, \Theta^k)$ and $p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k)$ are needed. Since direct calculations are intractable, those density functions are to be numerically calculated using particle filter in the next section.

2.4 Computation through Particle Filtering

2.4.1 Particle filters revisit

The basic idea of particle filters is to represent the desired posterior density function by a series of particles with associated weights, i.e. $\{x_t^i, w_t^i\}_{i=1}^N$. Then the density function of the states given the current estimation of parameters Θ^k can be discretely approximated as [12]

$$p(x_t|y_{t_1:t_\beta}, \Theta^k) \approx \sum_{i=1}^N \omega_t^i \delta(x_t - x_t^i) \quad (2.20)$$

where $\delta(\cdot)$ is the Dirac delta function, $t_\beta \leq t$; N is the number of particles; ω_t^i is the normalized weight associated with the i^{th} particle such that $\sum_{i=1}^N \omega_t^i = 1$. Suppose that at time $t - 1$, a set of particles $\{x_{1:t-1}^i\}_{i=1}^N$ are available and we want to obtain N particles which represent the hidden state for time t . Since it is usually difficult to directly draw samples from the true posterior density $p(x_{1:t}|y_{t_1:t_\beta}, \Theta^k)$, the principle of importance sampling [10] is adopted. The idea is to use a so called importance density $q(\cdot)$ from which one can easily draw samples $x_{1:t}^i, i = 1, \dots, N$. Then the posterior is obtained by resampling important sampling. It has been shown that, as long as the support region of the posterior density belongs to that of the importance density, the particle approximation is unbiased [11]. The importance sampling is commonly chosen as the probability of state transition, i.e.

$$q(x_t|y_{t_1:t_\beta}, \Theta^k) = p(x_t|x_{t-1}, \Theta^k) \quad (2.21)$$

With this choice, the weight for each particle can be derived as [12]

$$\omega_t^i \propto \omega_{t-1}^i p(y_t|x_t^i, \Theta^k) \quad (2.22)$$

For time instants $t = m_1, \dots, m_\beta$, when the outputs are not available, draw particles from the importance density $p(x_t|x_{t-1}^i, \Theta^k)$ and keep the weights unchanged, i.e.

$$\omega_t^i = \omega_{t-1}^i \quad (2.23)$$

To avoid the degeneracy problem [12], the importance sampling step is usually followed by a resampling procedure. The idea is to discard the particles with small weights and concentrate on those with large weights. After resampling, each particle's weight will be reset to $\omega_t^i = \frac{1}{N}$.

2.4.2 Particle filters approximation and cautious resampling

The problem brought by brute force resampling is that it reduces the diversity among particles. One solution is to resample the particles only when it is necessary instead of performing it at each step. To be specific, N_{eff} is introduced to represent the effective particle number [13]

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\omega_t^i)^2} \quad (2.24)$$

where ω_t^i is the normalized weight obtained through Equation 2.22. It implies that, as the variance of the weights grows very large, the effective sample size decreases to a small number which indicates a severe degeneracy problem. In practice, one uses resampling to eliminate useless particles only when a severe degeneracy problem occurs, say, N_{eff} falls below the threshold N_{thred} .

Given the current estimation of parameters, the particle filter algorithm is summarized as follows:

Step 1. Initialization. Draw initial N particles $\{x_0^i\}_{i=1}^N$ from the prior density $p(x_0|\Theta^k)$ and set each particle's weight to $\frac{1}{N}$. Set $t=1$.

Step 2. Importance sampling. Generate predicted particles $\{x_t^i\}_{i=1}^N$ from the importance density $p(x_t|x_{t-1}, \Theta^k)$.

Step 3. Assigning weights. Assign the weight to each particle using Equation 3.12 when y_t is available. Otherwise, calculate the weights according to Equation 2.23.

Step 4. Resampling. Compute the number of effective particles using Equation 3.13. If N_{eff} is less than the threshold N_{thred} , then perform resampling and replace the predicted particles in Step 2 with resampled particles. Reset the weights of resampled particles uniformly as $\omega_t^i = \frac{1}{N}$. Otherwise, go to Step 5.

Step 5. Set $t = t + 1$ and repeat Step 2 to Step 4 for $t \leq T$.

Estimation of $p(x_{1:T}|C_{obs}, \Theta^k)$ and $p(x_{1:T}, y_{m_1:m_\beta}|C_{obs}, \Theta^k)$ is a problem of smoothing all states with all available observations. Its computation with the iterative EM algorithm is intensive. With further marginalization of the

states and the missing observations following the approach of Gopaluni [9], the Q function obtained in Equation 2.18 can be rewritten as

$$\begin{aligned}
Q(\Theta|\Theta^k) &= \sum_{t=t_1}^{t=t_\alpha} \sum_{j=1}^J \int_X \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_t|C_{obs}, \Theta^k) dx_t \\
&+ \sum_{t=m_1}^{t=m_\beta} \sum_{j=1}^J \int_{X, Y_m} \log p(y_t|x_t, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_t, y_t|C_{obs}, \Theta^k) \\
&\quad dx_t dy_t \\
&+ \sum_{t=1}^{t=T} \sum_{j=1}^J \log p(I_t = j|\sigma_j, H_t) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{j=1}^J \int_X \log p(x_1|\Theta_1) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_1|C_{obs}, \Theta^k) dx_1 \\
&+ \sum_{t=2}^{t=T} \sum_{j=1}^J \int_X \log p(x_t|x_{t-1}, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \cdot p(x_{t-1:t}|C_{obs}, \Theta^k) \\
&\quad dx_{t-1:t} \\
&+ \sum_{j=1}^J \log C \cdot p(I_t = j|C_{obs}, \Theta^k) \tag{2.25}
\end{aligned}$$

Calculation of $p(x_t|C_{obs}, \Theta^k)$, $p(x_t, y_t|C_{obs}, \Theta^k)$ and $p(x_{t-1}, x_t|C_{obs}, \Theta^k)$ is a smoothing problem, of which the computation cost is very high. A practical solution is to apply recursive state filtering such that $p(x_t|C_{obs}, \Theta^k)$ is recursively approximated by $p(x_t|y_{t_1:t_\beta}, \Theta^k)$ for $t = 1 : T$, $p(x_t, y_t|C_{obs}, \Theta^k)$ is recursively approximated by $p(x_t, y_t|y_{t_1:t_\beta}, \Theta^k)$ for $t = 1 : T$, and $p(x_t, x_{t+1}|C_{obs}, \Theta^k)$ is recursively approximated by $p(x_t, x_{t+1}|y_{t_1:t_\beta}, \Theta^k)$ for $t = 1 : T - 1$, where $t_\beta \leq t$. This solution can significantly reduce the computation complexity and thus make the solution possible in real-time applications.

In Equation 2.25, the density function $p(x_t|C_{obs}, \Theta^k)$ is approximated using particle filters as

$$\begin{aligned}
p(x_t|C_{obs}, \Theta^k) &\approx p(x_t|y_{t_1:t_\beta}, \Theta^k) \\
&= \sum_{i=1}^N \omega_t^i \delta(x_t - x_t^i) \tag{2.26}
\end{aligned}$$

When the observation is missing, the joint density of x_t and y_t is required, which can be derived as

$$\begin{aligned}
p(x_t, y_t|C_{obs}, \Theta^k) &\approx p(x_t, y_t|y_{t_1:t_\beta}, \Theta^k) \\
&= p(y_t|x_t, \Theta^k) p(x_t|y_{t_1:t_\beta}, \Theta^k) \tag{2.27}
\end{aligned}$$

Since y_t is missing, one can replace it with the predicted y_t^i which is the prediction using x_t^i such that

$$y_t^i = h(x_t^i, \Theta^k) \quad (2.28)$$

Therefore,

$$p(x_t, y_t | C_{obs}, \Theta^k) \approx \sum_{i=1}^N \omega_{t|x}^i \delta(x_t - x_t^i) \delta(y_t - y_t^i) \quad (2.29)$$

where

$$\omega_{t|x}^i = \frac{p(y_t^i | x_t^i, \Theta^k) p(x_t^i | y_{t_1:t_\beta}, \Theta^k)}{\sum_{i=1}^N p(y_t^i | x_t^i, \Theta^k) p(x_t^i | y_{t_1:t_\beta}, \Theta^k)} \quad (2.30)$$

Using Equation 2.28,

$$\begin{aligned} p(y_t^i | x_t^i, \Theta^k) &= \frac{1}{\sqrt{2\pi R}} \exp\left(-\frac{(y_t^i - h(x_t^i, \Theta^k))^2}{2R}\right) \\ &= \frac{1}{\sqrt{2\pi R}} \exp\left(-\frac{(h(x_t^i, \Theta^k) - h(x_t^i, \Theta^k))^2}{2R}\right) \\ &= 1 \end{aligned} \quad (2.31)$$

Hence,

$$\begin{aligned} \omega_{t|x}^i &= \frac{p(x_t^i | y_{t_1:t_\beta}, \Theta^k)}{\sum_{i=1}^N p(x_t^i | y_{t_1:t_\beta}, \Theta^k)} \\ &= \omega_t^i \end{aligned} \quad (2.32)$$

As for the joint density function of x_t and x_{t+1} , it can be approximated as

$$\begin{aligned} p(x_t, x_{t+1} | C_{obs}, \Theta^k) &\approx p(x_t, x_{t+1} | y_{t_1:t_\beta}, \Theta^k) \\ &= p(x_{t+1} | x_t, \Theta^k) p(x_t | y_{t_1:t_\beta}, \Theta^k) \\ &= \sum_{i=1}^N \omega_{t|t+1}^i \delta(x_t - x_t^i) \delta(x_{t+1} - x_{t+1}^i) \end{aligned} \quad (2.33)$$

where

$$\omega_{t|t+1}^i = \frac{p(x_{t+1}^i | x_t^i, \Theta^k) p(x_t^i | y_{t_1:t_\beta}, \Theta^k)}{\sum_{i=1}^N p(x_{t+1}^i | x_t^i, \Theta^k) p(x_t^i | y_{t_1:t_\beta}, \Theta^k)} \quad (2.34)$$

Substituting these approximated density functions, the Q function in Equation

2.25 can be finally obtained.

$$\begin{aligned}
Q(\Theta|\Theta^k) &\approx \sum_{t=t_1}^{t=t_\alpha} \sum_{j=1}^J \sum_{i=1}^N \omega_t^i \log p(y_t|x_t^i, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{t=m_1}^{t=m_\beta} \sum_{j=1}^J \sum_{i=1}^N \omega_t^i \log p(y_t^i|x_t^i, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{t=1}^{t=T} \sum_{j=1}^J \log p(I_t = j|\sigma_j, H_t) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{j=1}^J \sum_{i=1}^N \omega_1^i \log p(x_1^i|\Theta_1) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{t=2}^{t=T} \sum_{j=1}^J \sum_{i=1}^N \omega_{t-1|t}^i \log p(x_t^i|x_{t-1}^i, \Theta_j) \cdot p(I_t = j|C_{obs}, \Theta^k) \\
&+ \sum_{j=1}^J \log C \cdot p(I_t = j|C_{obs}, \Theta^k) \tag{2.35}
\end{aligned}$$

With the approximated Q function, the EM algorithm can hence be implemented. In the expectation step, the Q function is evaluated according to Equation 2.35 with the current estimated parameters $\Theta_j^k, j = 1 : J$. In the next maximization step, the new parameters $\Theta_j^{k+1}, j = 1 : J$, are obtained by maximizing the Q function.

To maximize the Q function over parameters Θ , derivative operation is performed with respect to each parameter. Therefore, optimal estimation of system parameters at each iteration can be calculated by equating the derivatives to zero, i.e. $\frac{\partial Q}{\partial \theta_{ji}} = 0$, where θ_{ji} is the i^{th} system parameter for the j^{th} local model.

The EM algorithm is summarized as follows:

Step 1. Initialization. Start with the initial parameters $\Theta_j^0, j = 1 : J$, and set $t=0$.

Step 2. Expectation. At time t , calculate the approximate Q function using 2.35, given the current estimation of the system parameters Θ^k .

Step 3. Maximization. Maximize the approximated Q function and get the new parameters $\Theta_j^{k+1}, j = 1 : J$. Set $k=k+1$.

Step 4. Repeat Step 2 and Step 3 until the converge condition is satisfied, i.e. the change of the estimated parameters between two iterations is less than the tolerance.

The validity for each local model $\sigma_j, j = 1 : J$ also needs to be updated during each iteration. Due to the usage of the exponential function as it is shown in Equation 2.5, an analytical expression is difficult to obtain when maximizing the Q function [5]. The mathematical formulation of the optimization

problem in the search for optimal $\sigma_j, j = 1, 2 \dots J$ values can be expressed as:

$$\begin{aligned} \max_{\sigma_j, j=1,2,\dots,J} & \sum_{t=1}^{t=T} \sum_{i=1}^N \sum_{j=1}^J \omega_t^i \log p(I_t = j | \sigma_j, H_t) \cdot P(I_t = j | C_{obs}, \Theta^k) \\ \text{S.t.} & \sigma_{min} \leq \sigma_j, j = 1, 2 \dots J \leq \sigma_{max} \end{aligned} \quad (2.36)$$

where $\log p(I_t = j | \sigma_j, H_t)$ can be calculated from Equation 2.6. $p(I_t = j | C_{obs}, \Theta^k)$ represents the probability of the data point belonging to i th sub-model at time t .

In this chapter, a constrained nonlinear optimization function named ‘fmin-con’ provided by ‘MATLAB’ is adopted in a search for the optimal value for σ_j at each iteration of the EM algorithm.

Finally, having all the estimated model parameters $\Theta_j, j = 1 : J$ and validity $\sigma_j, j = 1 : J$ for each local model, the global model can be obtained by substituting the estimated parameters into Equation 2.4.

2.5 Simulations and Pilot-scale Experiment

In this section, the proposed approach is evaluated through both numerical simulations as well as experimental verification. Its efficiency in handling missing outputs with less computational cost will be demonstrated. All the simulations were run on a 3.00 GHz CPU with 4 GB RAM PC using MATLAB 2009a.

2.5.1 A numerical simulation example

A first order process with varying system parameters is utilized here to demonstrate the efficiency of the proposed parameter-varying model estimation method. This process was originally used in Zhu and Xu (2008) [14] as an illustrative example. It is described by the following equation:

$$G(s, H) = \frac{K(H)}{\tau(H)s + 1} \quad (2.37)$$

where both the process gain $K(H)$ and the process time constant $\tau(H)$ are nonlinear functions of the scheduling variable H . The specific nonlinear relation is expressed as follows:

$$K(H) = 0.6 + H^2, H \in [1, 4] \quad (2.38)$$

$$\tau(H) = 3 + 0.5H^3, H \in [1, 4] \quad (2.39)$$

By assuming that the observation y_t is a cosine function of the state, this process can be converted to the following nonlinear state space model:

$$\begin{aligned}\dot{x}_t &= a(H)x_t + b(H)u_t + \omega_t \\ y_t &= \cos(x_t) + v_t\end{aligned}\quad (2.40)$$

where

$$\begin{aligned}a(H) &= -\frac{1}{\tau(H)} \\ b(H) &= \frac{K(H)}{\tau(H)}\end{aligned}\quad (2.41)$$

Apparently, over the whole operating range of the process, the gain as well as the time constant changes dramatically and one single process model would hardly capture the dynamics of the process in its complete operating range. In other words, one local model cannot give a good approximation of the process behavior throughout the whole operating trajectory. Therefore, multiple models or a parameter-varying global model is required here in order to describe the behavior of the process under different operating conditions.

It is pre-determined that the process is to be tested at three predesigned local operating points:

$$H_1 = 1, H_2 = 2.25, H_3 = 4 \quad (2.42)$$

When transition from one operating point to the other, scheduling variable H is gradually increased by a fixed interval. Figure 2.1 shows the trajectory of the scheduling variable.

The process input u switches randomly among multiple levels throughout the whole experiment.

$T=250$ measurements are collected from the simulation. Note that the relations between parameters and the scheduling variable expressed in Equation 2.38 and Equation 2.39 are assumed unknown in the following identification process. The proposed multiple model parameter estimation method is then applied. In order to test the algorithm's capability in handling the missing data, different portions of the output data are randomly removed from the model training data set to simulate missing data problem. $N=150$ particles are used for the particle filter computation.

In the expectation step of the EM algorithm, the Q function is calculated according to 2.35, where

$$\log[p(x_t^i|x_{t-1}^i, \Theta_j)] = \log\left[\frac{1}{\sqrt{2\pi}Q_x} \exp\left[-\frac{1}{2} \frac{(x_t^i - a_j x_{t-1}^i - b_j u_{t-1})^2}{Q_x}\right]\right] \quad (2.43)$$

For $t = t_1 : t_\alpha$,

$$\log[p(y_t|x_t^i, \Theta_j)] = \log\left[\frac{1}{\sqrt{2\pi}Q_y} \exp\left[-\frac{1}{2} \frac{(y_t - \cos x_t^i)^2}{Q_y}\right]\right] \quad (2.44)$$

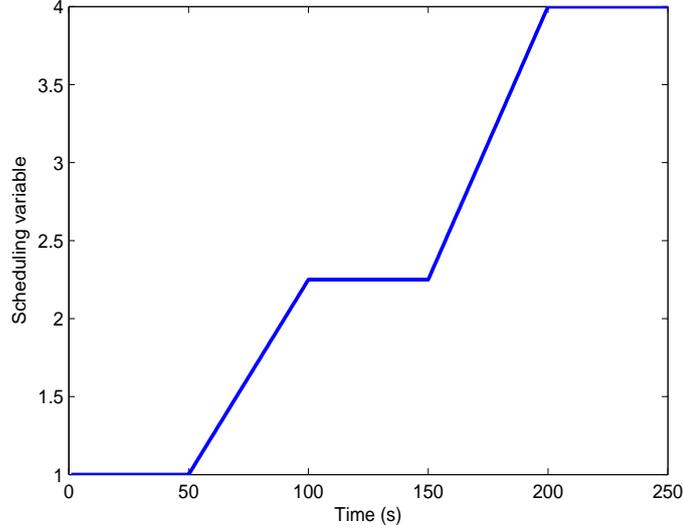


Figure 2.1: Trajectory of the scheduling variable H

For $t = m_1 : m_\beta$,

$$\log[p(y_t|x_t^i, \Theta_j)] = \log\left[\frac{1}{\sqrt{2\pi Q_y}} \exp\left[-\frac{1}{2} \frac{(y_t^i - \cos x_t^i)^2}{Q_y}\right]\right] \quad (2.45)$$

where $y_t^i = \cos x_t^i$.

As for $p(I_t = j|C_{obs}, \Theta^k)$, it can be calculated according to Equation 2.6 as

$$p(I_t = j|C_{obs}, \Theta^k) = \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \quad (2.46)$$

In the maximization step of the EM algorithm, by taking derivative over the Q function and equating it to zero, each individual component of the parameters is hence calculated as

$$a_j^{new} = \frac{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i \cdot \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \cdot (x_t^i x_{t-1}^i - b^{old} x_{t-1}^i u_{t-1})}{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i \cdot \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \cdot (x_{t-1}^i)^2} \quad (2.47)$$

$$b_j^{new} = \frac{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i \cdot \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \cdot (x_t^i u_{t-1} - a^{old} x_{t-1}^i u_{t-1})}{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i \cdot \exp\left(\frac{-(H_t - H_j)^2}{2(\sigma_j)^2}\right) \cdot u_{t-1}^2} \quad (2.48)$$

The trajectories of the estimated parameters for each local model when 25% output data are missing are shown in Figure 2.2 and Figure 2.3. The estimated parameter values after 150 iterations are given in Table 2.1 and Table 2.2.

Table 2.1: Estimated parameter a after 150 iterations

True value	$a_1 = 0.7143$	$a_2 = 0.8850$	$a_3 = 0.9715$
Proportion of missing output	a_1	a_2	a_3
Full data set	0.7258	0.8952	0.9628
25%	0.7074	0.8628	0.9615
50%	0.7021	0.8654	0.9751

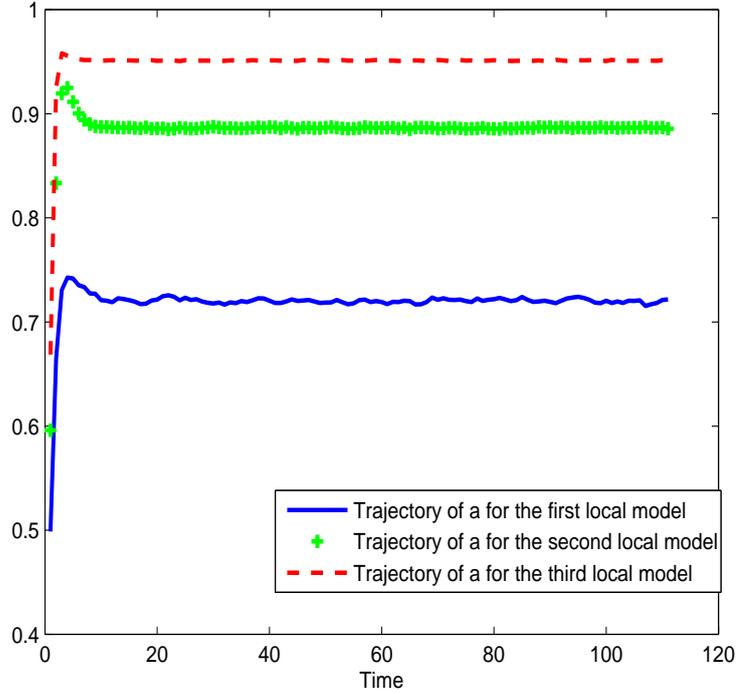


Figure 2.2: Trajectories of estimated parameter a for each local model when 25% observations are missing.

Table 2.2: Estimated parameter b after 150 iterations

True value	$b_1 = 0.4571$	$b_2 = 0.6512$	$b_3 = 0.4743$
Proportion of missing output	b_1	b_2	b_3
Full data set	0.4650	0.6601	0.4798
25%	0.4779	0.6548	0.4534
50%	0.4716	0.6570	0.4681

The comparison result of the identified global model with the true output is displayed in Figure 2.4. Here to better test the validity of the identified model, model validation is conducted under other two different operating points:

$$H_4 = 1.75, \quad H_5 = 3 \quad (2.49)$$

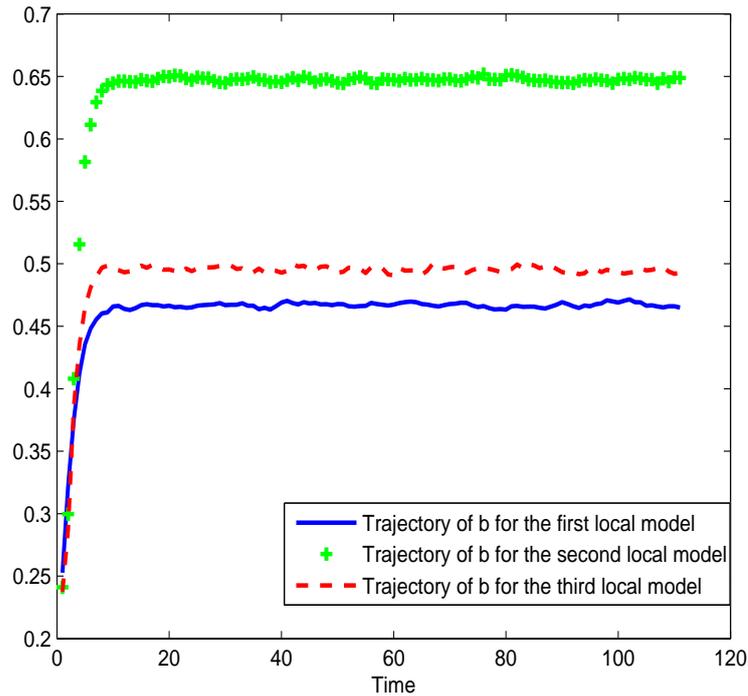


Figure 2.3: Trajectories of estimated parameter b for each local model when 25% observations are missing.

and the comparison of the global model prediction with the true process output is shown in Figure 2.5.

Figure 2.6 provides a weighting map of each local model under different scheduling values. Based on this calculated weighting map as well as Figure 2.4, model predictions can be calculated for all the H values.

Comparison result displayed in Figure 2.4 and Figure 2.5 shows that the identified global model not only can well capture the process dynamics under the training operating conditions, but also perform well in capturing the process dynamics at other operating points that are different from the operating points within the training data. This confirms the effectiveness of the identified global model in approximating the real process dynamics throughout the operating range.

2.5.2 Continuous stirred tank reactor

This model has been utilized as the illustration examples in Gopaluni (2008) [9] and Jin and Huang (2010) [5]. The system is described by the following set

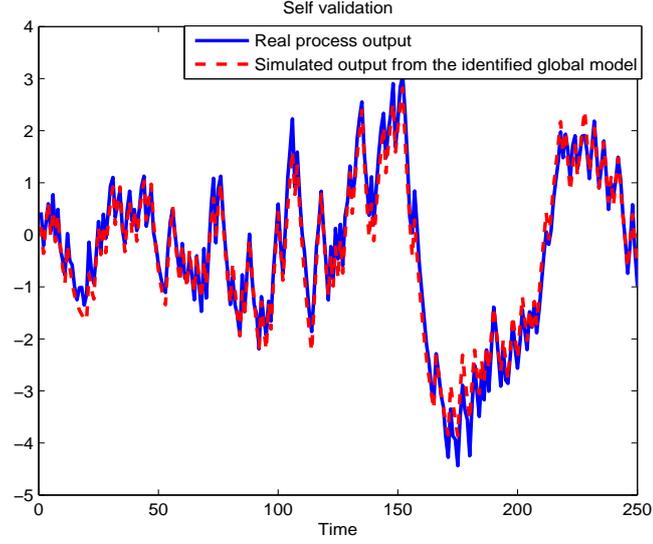


Figure 2.4: Validation of the identified global model against the model training data set.

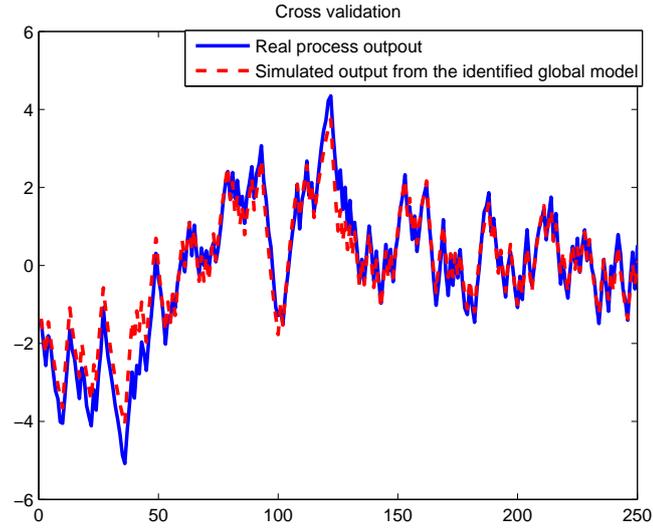


Figure 2.5: Cross validation of the identified global model.

of differential equations:

$$\frac{dC_A(t)}{dt} = \frac{q(t)}{V}(C_{A0}(t) - C_A(t)) - k_0 C_A(t) \exp\left(\frac{-E}{RT(t)}\right) \quad (2.50)$$

$$\begin{aligned} \frac{dT(t)}{dt} = & \frac{q(t)}{V}(T_0(t) - T(t)) - \frac{(\Delta H)k_0 C_A(t)}{\rho C_p} \exp\left(\frac{-E}{RT(t)}\right) \\ & + \frac{\rho_c C_{pc}}{\rho C_p V} q_c(t) \left\{ 1 - \exp\left(\frac{-hA}{q_c(t)\rho C_p}\right) \right\} (T_{c0}(t) - T(t)) \end{aligned} \quad (2.51)$$

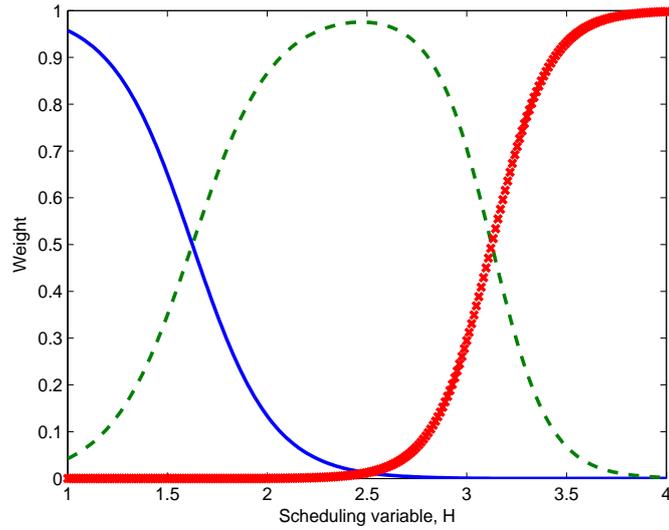


Figure 2.6: Weight of each local model at different operating points

where C_A is the outlet reagent concentration (g/l); T is the reactor temperature (g/l); the inlet flow rate q is the system input. The explanations of the system variables and their corresponding steady state values are given in Table 2.3. We consider the temperature T as an interested output variable. The coolant flow rate q_c has a direct impact on the process dynamic; and it is the scheduling variable H from which the operation condition of the process is determined. The trajectory of the scheduling variable is given in Figure 2.7.

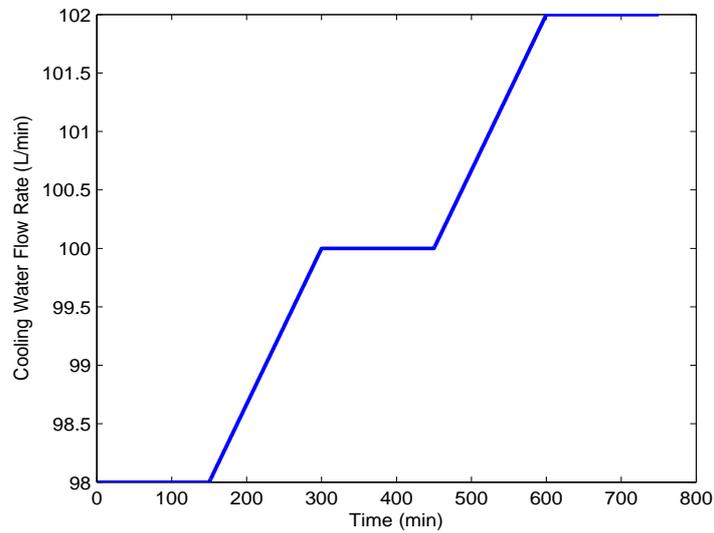


Figure 2.7: Trajectory of the scheduling variable H

Table 2.3: CSTR model parameters and their steady state values

parameters	steady state value
production concentration of Component A, C_A	$output_1$
temperature of the reactor, T	$output_2$
feed Concentration of Component A, C_{A0}	1mol/L
feed temperature, T_0	350.0 K
specific heats, C_p, C_{pc}	1 cal/(g K)
liquid density, ρ, ρ_c	1×10^3 g/L
heat of reaction, ΔH	-2×10^5 cal/mol
activation energy term, E/R	1×10^4 K
reaction rate constant, k_0	$7.2 \times 10^1 \text{min}^{-1}$
heat transfer term, hA	7×10^5 cal/(min K)
the reactor volume, V	100L
inlet coolant temperature, T_{c0}	350.0 K
process flow rate, q	input
coolant flow rate, q_c	Scheduling variable

The system parameters to be estimated are θ_1 and θ_2 which are both functions of the scheduling variable q_c such that

$$\theta_1 = \frac{\rho_c C_{pc}}{\rho C_p} H(t) \quad (2.52)$$

$$\theta_2 = \frac{hA}{H(t)\rho C_p} \quad (2.53)$$

Finally the differential equations can be written in the following form with θ_1 and θ_2 as varying unknown parameters:

$$\frac{dC_A(t)}{dt} = \frac{q(t)}{V} (C_{A0}(t) - C_A(t)) - k_0 C_A(t) \exp\left(\frac{-E}{RT(t)}\right) \quad (2.54)$$

$$\begin{aligned} \frac{dT(t)}{dt} = & \frac{q(t)}{V} (T_0(t) - T(t)) - \frac{(\Delta H)k_0 C_A(t)}{\rho C_p} \exp\left(\frac{-E}{RT(t)}\right) \\ & + \theta_1 \{1 - \exp(-\theta_2)\} (T_{c0}(t) - T(t)) \end{aligned} \quad (2.55)$$

The process is operated at three different operating points $H_1 = 98L/min$; $H_2 = 100L/min$; $H_3 = 102L/min$; during the transition period the dilution factor is increased by a fixed step size and no additional excitation signal is added to the dilution factor. System parameters θ_1 and θ_2 vary with the scheduling variable, resulting in a time varying nonlinear process which cannot be adequately described by a single nonlinear model. The experiment is performed under these three operation conditions and the training data collected with 25% output data randomly erased.

The algorithm is applied to the training data and an approximate global model is obtained afterwards. Once again the explicit relations expressed

in Equation 2.52 and Equation 2.53 are assumed unknown in the following identification process. The result of the model validation for the training data is shown in Figure 2.8.

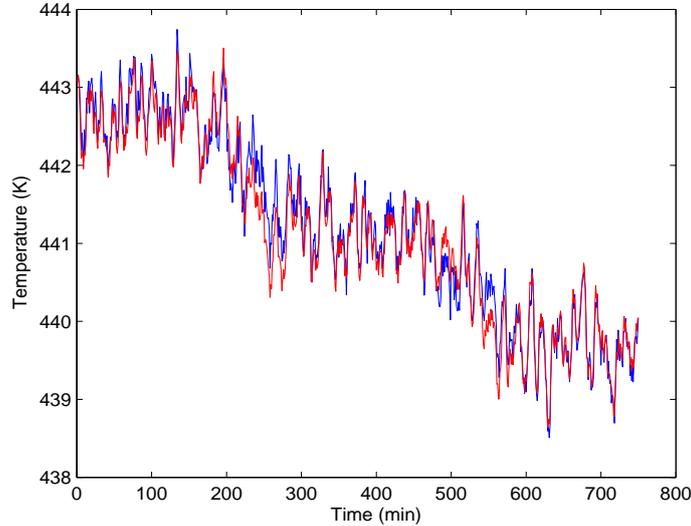


Figure 2.8: Validation of the identified global model against the model training data set. Blue line represent the real process output and the red line is the simulated output from the identified global model

To better test the algorithm, additional simulation is conducted under another operating point $H_1 = 99L/min$. The cross validation is given in Figure 2.9.

Figure 2.10 provides a weighting map of each local model under different scheduling values. Based on this calculated weighting map as well as Figure 2.4, model predictions can be calculated under all the H values within the operating range.

Comparison results displayed in Figure 2.8 and Figure 2.9 show that the identified global model can well capture the process dynamics at both training operating points as well as other operating points that are not included in the training operating points.

2.5.3 Experimental evaluation: a multi-tank system

In this section, an experimental evaluation on a three-tank system is performed to further verify the effectiveness of the proposed algorithm. Figure 2.11 illustrates the simplified process setup of the three-tank system which consists of three tanks placed on top of each other. During the experiment, water is pumped from the bottom supply tank into the top tank and then flows to each tank by gravity. There are three valves, one for each tank, by adjusting

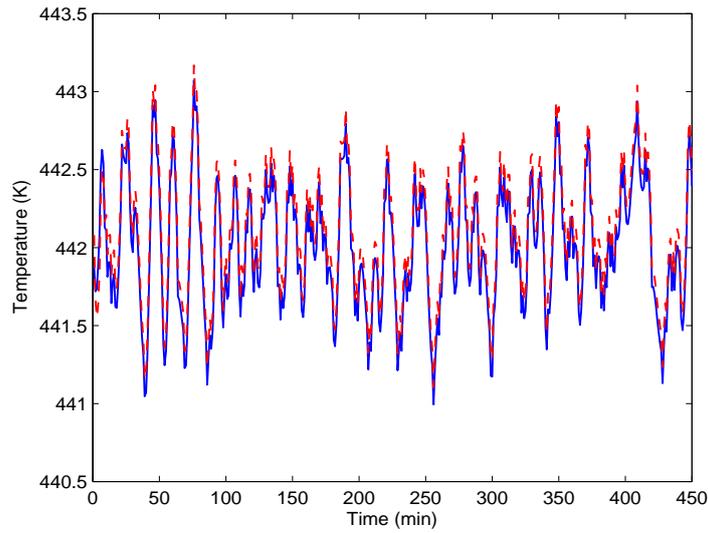


Figure 2.9: Cross validation of the identified global model. Blue line represent the real process output and the red line is the simulated output from the identified global model

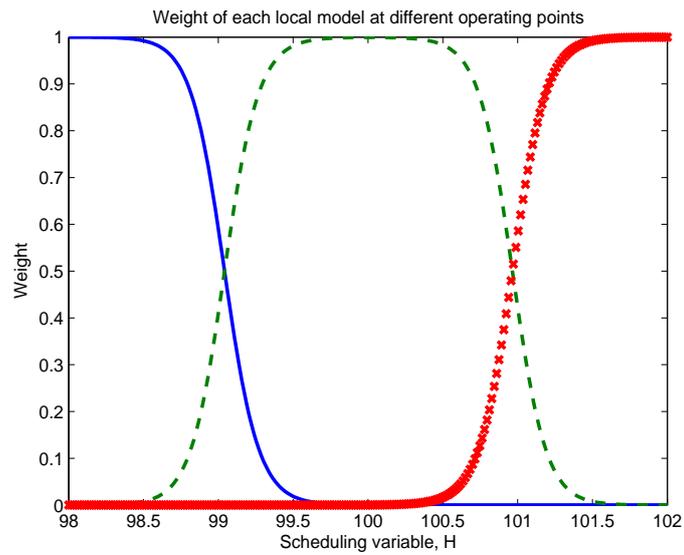


Figure 2.10: Weight of each local model at different operating points

which the outflow rate from each tank can be controlled. The nonlinear model

describing the process dynamics is given by

$$\begin{aligned}\frac{dH_1}{dt} &= \frac{1}{\beta_1}q - \frac{1}{C_1 H_1^{\alpha_1}} C_1 H_1^{\alpha_1} \\ \frac{dH_2}{dt} &= \frac{1}{\beta_2} C_1 H_1^{\alpha_1} - \frac{1}{\beta_2} C_2 H_2^{\alpha_2} \\ \frac{dH_3}{dt} &= \frac{1}{\beta_3} C_2 H_2^{\alpha_2} - \frac{1}{\beta_3} C_3 H_3^{\alpha_3}\end{aligned}\tag{2.56}$$

where

- q is the inlet flow rate into the upper tank;
- H_i is the water level of the i_{th} tank, $i = 1, 2, 3$;
- C_i is the resistance of the output orifice of the i_{th} tank, $i = 1, 2, 3$;
- β_i is the cross sectional area of the i_{th} tank, $i = 1, 2, 3$;
- α_i is the flow coefficient of the i_{th} tank, $i = 1, 2, 3$.

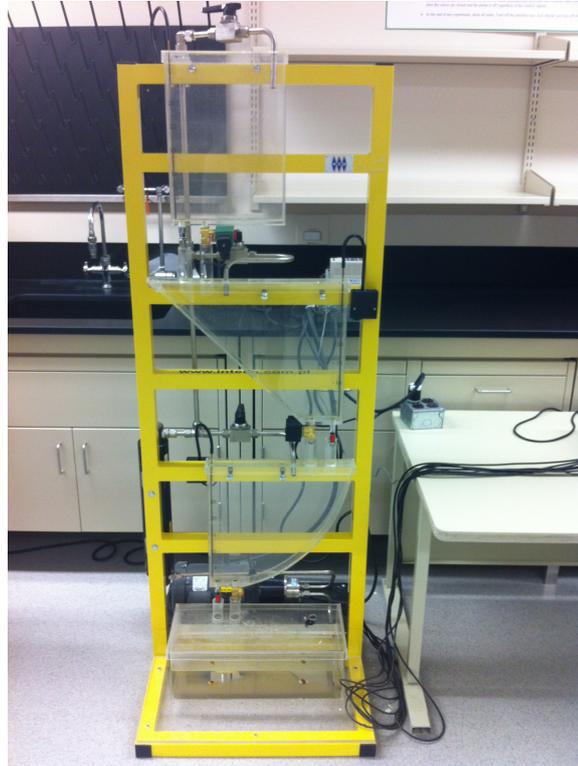


Figure 2.11: Three tank system schematic

The process dynamics of the second tank is of interest in this study. The state space model with the water level in the second tank H_2 as the state is

given by

$$\begin{aligned}\frac{dH_2}{dt} &= \frac{1}{\beta_2}C_1H_1^{\alpha_1} - \frac{1}{\beta_2}C_2H_2^{\alpha_2} + w_t \\ y_t &= x_t + v_t\end{aligned}\tag{2.57}$$

From 2.57, it can be seen that water level of the first tank has a direct impact on the second tank water level. Therefore, H_1 is chosen as the scheduling variable of the system. Three different operating points are selected as shown in Table 2.4. The water level of the top tank H_1 is maintained at the desired value through a PID controller by manipulating the inlet flow rate into the top tank.

Table 2.4: Designed operating point for the experiment

Operating points, $H_{1m}, m = 1, 2, 3, 4$
5 (cm)
10 (cm)
14 (cm)

When transition from one operating point to the other, the water level in the top tank increases at a fixed step over the transition period until reaches the value of the next operating point. The trajectory of the scheduling variable H_1 is shown in Figure 2.12.

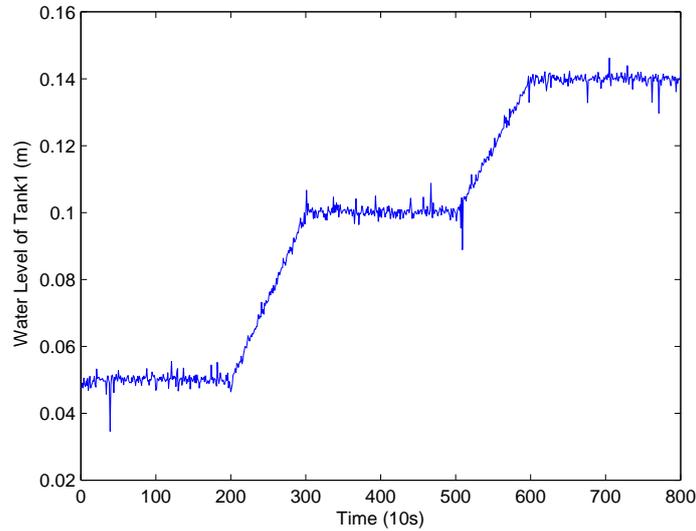


Figure 2.12: Three tank system scheduling variable for the self validation

The value of α_2 is chosen as 0.5 according to the multi-tank system experiment manual. This is an appropriate assumption as the inlet flow rate is fairly

small so that the water dynamic in each tank can be considered as laminar flow. The output orifice resistance C_2 is directly affected by the valve position of the second tank V_2 ; hence C_2 can be approximated as linear function of V_2 such that

$$C_2 \approx \alpha \cdot V_2 \quad (2.58)$$

The sectional area of the second tank is given below

$$\beta_2 = c \cdot w + \frac{H_2}{H_{2max}} \cdot b \cdot w \quad (2.59)$$

where c , w , b , H_{2max} are the geometrical parameters of the second tank with each value given as

$$c = 10cm, \quad w = 3.5cm, \quad b = 34.5cm, \quad H_{2max} = 35cm \quad (2.60)$$

By substituting 2.59 and 2.58 into 2.57, the state equation becomes

$$\frac{dH_2}{dt} = \frac{H_{2max} \cdot C_1 \cdot H_1^{\alpha_1}}{w(cH_{2max} + bH_2)} - \frac{H_{2max} \cdot \alpha \cdot V_2 H_2^{0.5}}{w(cH_{2max} + bH_2)} + w_t \quad (2.61)$$

Therefore, the state equation of H_2 can be rearranged as

$$\frac{dH_2}{dt} = \frac{\theta_1}{cH_{2max} + bH_2} - \frac{\theta_2 \cdot V_2 \cdot H_2^{0.5}}{cH_{2max} + bH_2} \quad (2.62)$$

where θ_1 is function of H_1 such that

$$\theta_1 = \frac{H_{2max} \cdot C_1 \cdot H_1^{\alpha_1}}{w} \quad (2.63)$$

$$\theta_2 = \frac{H_{2max} \cdot \alpha}{w} \quad (2.64)$$

A multiple level random signal is designed for the valve position V_2 as the system input. The process input and output data are given in 2.13.

The relations between parameters and the scheduling variable expressed in Equation 2.63 and Equation 2.64 are assumed unknown in identification. The collected experimental data is then passed through the proposed algorithm with 20% output data randomly removed. The self validation result is given in Figure 2.14.

It can be seen from Figure 2.14 that the estimated global model performs well in the self validation. Another experiment is conducted for the cross validation in order to test the global model's capability in predicting the dynamics of the process at different operating points. In the cross validation experiment, the scheduling variable H_1 is maintained at 8 cm during the whole cross validation experiment. The input signal and the output data for cross validation is shown in Figure 2.15.

Figure 2.16 gives the result of the cross validation which shows a good match between the real process output and the prediction of the identified global model, confirming the effectiveness of the proposed algorithm.

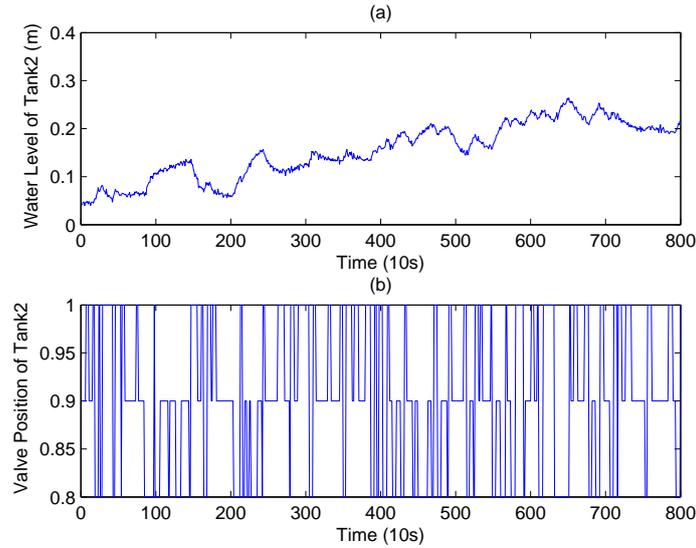


Figure 2.13: Three tank system input-output data (a) water level of the second tank, process output (b) valve position V2, process input

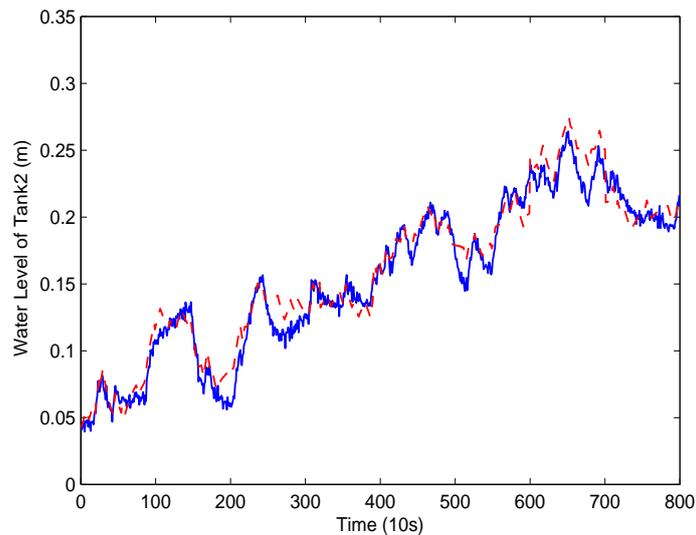


Figure 2.14: Self-validation result. Blue solid line is the collected process data, red dash line is the simulated output of the identified global model

2.6 Discussion and Conclusion

This chapter described a Bayesian approach for identifying parameter-varying nonlinear state space model with missing output data within the framework of the EM algorithm. Particle filter is used for the calculation in the Expectation step. The capability of the proposed algorithm in handling missing observa-

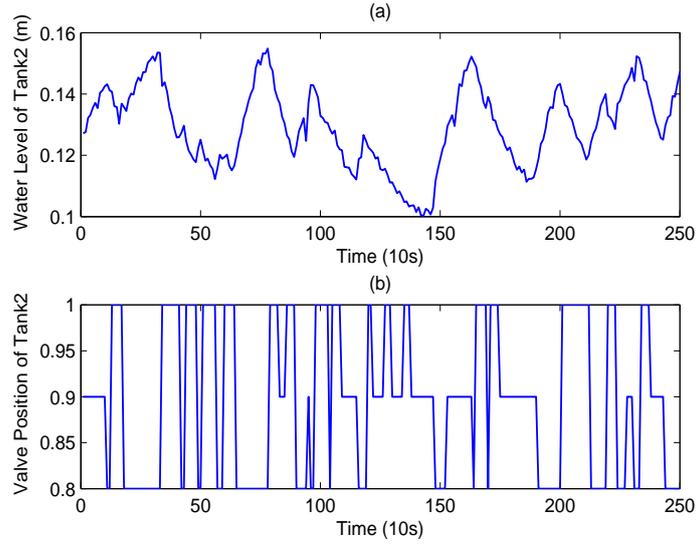


Figure 2.15: Three tank system input-output data (a) water level of the second tank, process output (b) valve position V2, process input

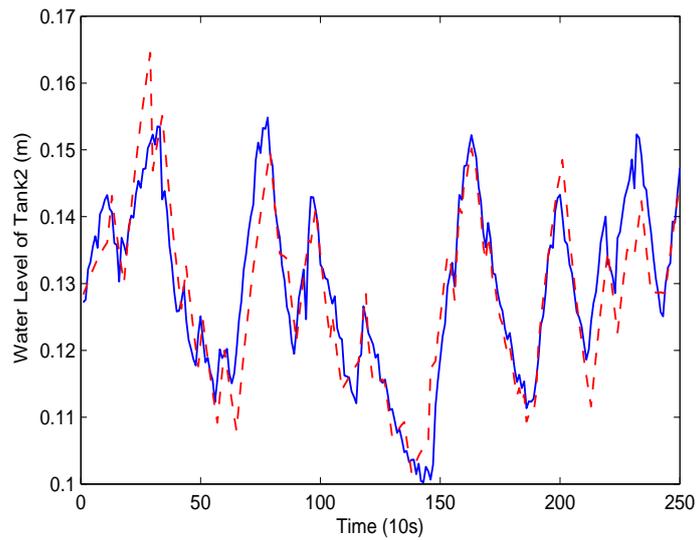


Figure 2.16: Cross-validation result. Blue solid line is the collected process data, red dash line is the simulated output of the identified global model

tions in the presence of varying parameters is demonstrated through numerical examples as well as a pilot-scale experiment.

What have been discussed in this chapter so far mainly focus on the parameter estimation using a complete data set which is usually considered as an off-line analysis. The particle filtering approximation used in this chapter

reduced the computation burden compared with the particle smoothers. The Expectation-Maximization algorithm is an iterative method which takes certain time for the estimation to reach convergence. For example, for the data set with the length of 500 data point, the EM algorithm would take a couple of minutes to reach convergence. When it comes to on-line implementation, it is possible to apply the proposed method for model parameter estimation with proper sampling time and data length.

The model identification is an important step towards process monitoring and process control. Obtaining a fairly accurate model not only help better understand the process behavior, but can also be used for controller design purpose. For example, in the Model Predictive Control (MPC) which is a multivariable control algorithm, an internal dynamic model of the process is a prerequisite. For nonlinear systems with multiple operating points, single model may not be sufficient to describe the process and thus not suitable for controller design. The algorithm proposed in this chapter takes all the operating conditions into account and provides estimation of the global model parameters. Therefore, it is possible to design a MPC controller based on this identified global model and the control signal can be accordingly calculated.

It is doable to extend the proposed algorithm to deal with the parameter identification problem for hybrid systems. In this chapter, the parameter estimation problem is formulated under the framework of the Expectation Maximization (EM) algorithm. In addition to the hidden state $x_{1:T}$ and missing output Y_m , the hidden model identity I is introduced to denote which local model takes effect. As for hybrid system, take the linear piecewise AutoRegressive eXogenous (PWARX) process as an example which is formulated as below:

$$y_k = \begin{cases} \theta_1^T \begin{bmatrix} x_k \\ 1 \end{bmatrix} + e_k, & x_k \in \chi_1 \\ \vdots \\ \theta_M^T \begin{bmatrix} x_k \\ 1 \end{bmatrix} + e_k, & x_k \in \chi_M \end{cases}, k = 1, 2 \dots N \quad (2.65)$$

where N, M represent number of data points collected and number of sub-models respectively, y_k is the output, x_k is the regressor which consists of past input and output, e_k is the Gaussian distributed noise with zero mean and variance σ^2 , θ_i is the parameter vector of the i_{th} sub-model. $Z_k = x_k, y_k, k = 1, 2, \dots, N$ is defined as the observed data set generated from a PWARX system. I is introduced as a 'missing variable' to denote the sub-model identity of each data point. With the defined observed data set and missing data set, the identification problem can be formulated under the framework of the EM algorithm. More discussion with detailed formulation of hybrid PWARX system can be found in Jin and Huang (2010)[17].

Bibliography

- [1] P. K. Pearson, Control Systems, Identification, *Encyclopedia of Physical Science and Technology*, 2004, pp 687-707.
- [2] B. F. Feeny, Nonlinear System Identification, *Encyclopedia of Vibration*, 2004, pp 924-928.
- [3] J. Shamma and M. Athans, Guaranteed Properties of Gain Scheduled Control for Linear Parameter Varying Plants, *Automatica*, 1991, Vol. 27 pp 559-564.
- [4] Z. Xu, J. Zhao and J. Qian, Nonlinear MPC using an Identified LPV Model, *Industrial & Engineering Chemistry Research*, 2009, Vol. 49, pp 3043-3051.
- [5] X. Jin and B. Huang, Multiple Model LPV Approach to Nonlinear Process Identification with EM Algorithm, *Journal of Process Control*, 2010. Vol. 21, pp 182-193.
- [6] S. Khatibisepehr and B. Huang, Dealing with Irregular Data in Soft Sensors: Bayesian Method and Comparative Study, *Industrial & Engineering Chemistry Research*, vol. 22, 2008, pp 8713-8723.
- [7] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM algorithm, *Journal of the Royal Statistical Society, Series B*, vol. 39, 1977, pp 1-38.
- [8] T. B. Schon, A. Wills and B. Ninness, System Identification of Nonlinear State-Space Models, *Submitted to Automatica*, 2009
- [9] R. B. Gopaluni, A Particle Filter Approach to Identification of Nonlinear Process under Missing Observations, *The Canadian Journal of Chemical Engineering*, vol. 86, 2008, pp 1081-1092.
- [10] N. Bergman, Recursive Bayesian estimation: Navigation and tracking applications, *Ph.D Dissertation*, Linkping Univ., Linkping, Sweden, 1999.
- [11] A. Doucet, On sequential Monte Carlo methods for Bayesian filtering, Dept. Eng., Univ. Cambridge, UK, Technical Report, 1998.

- [12] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Transaction on Signal Processing*, vol. 50, 2002, pp 174-188.
- [13] J. S. Liu and R. Chen, Sequential Monte Carlo methods for dynamical systems, *Journal of the American Statistical Association*, vol. 93, 1998, pp 1032-1044.
- [14] Y. Zhu and Z. Xu, A Method of LPV Model Identification for Control, *In Proceeding of the 17th IFAC World Congress, Seoul, Korea*, 2008.
- [15] D. M. Espie and S. Macchietto, The Optimal Design of Dynamic Experiments, *American Institute of Chemical Engineers*, vol. 35, 1989, pp 223-229.
- [16] S. P. Asprey and S. Macchietto, Design Robust Optimal Dynamic Experiments, *Journal of Process Control*, vol. 12, 2002, pp 545-556.
- [17] X. Jin and B. Huang, Robust identification of piecewise/switching autoregressive exogenous process. *American Institute of Chemical Engineers Journal*, 2010, vol 56, pp 1829-1844.

Chapter 3

Inferential Sensor Development

² This chapter deals with the issues associated with the development of black box models as well as model update strategies for soft sensor applications. Key process variables that are hidden (difficult to measure) are commonly encountered due to the limitations of measurement techniques. Even with the appropriate instrumentation, some variables are only available through off-line laboratory analysis, which could have a sampling interval of several hours. Soft sensors are inferences that can provide continuous on-line estimates of hidden variables; this inference is capable of combining real-time measurement with offline lab analysis. Due to the prevalence of plant-model mismatch, it is important to update the model using the latest observation data. In this chapter, the parameters of black box models are estimated using particle filters under the framework of the expectation-maximization (EM) algorithm. A Bayesian methodology for model calibration strategy is formulated. The proposed method for parameter estimation and model updating is illustrated through a semi-fermenter simulation, and application to an oil sands process.

3.1 Introduction

Process industries are seeking to improve the performance of processing facilities in order to increase the productivity as well as reducing environmental footprint through better control. For both process monitoring and control, it is imperative to gather real-time information of the key process variables. Unfortunately, it is common to have key process variables that are *hidden* (difficult or impossible to measure online) such as the concentration of a chemical component. Some expensive on-line instrumentation may be installed to get fast readings for such variables; however, these hardware sensors could easily fail or have significant deviation, leading to missing or inaccurate information.

²Part of this chapter has been published at American Control Conference as Deng, J. and Huang, B.. Bayesian method for identification of constrained nonlinear processes with missing output data. *American Control Conference, San Francisco, 2011.*

In contrast to online measurements, offline analysis is conducted in laboratory, where the technicians collect product samples and perform a series of experimental analysis and calculations. Off-line laboratory analysis is generally more accurate than the on-line instrumentation, but the sampling rate is very slow (ranging from 30 minutes to 24 hours), making it unsuitable for real-time applications.

The applications of soft sensors have been discussed in many publications [1], [6], [2]. Soft sensors or inferential sensors are implemented to make real-time predictions of a measurement that would have otherwise taken much time to obtain. By taking into account of information from relatively fast-sampled variables (e.g. temperature, flowrate, density), soft sensors provide real-time estimates for those difficult-to-measure quality variables.

In the development of a successful soft sensor, a good process model is a prerequisite. The process models can be roughly divided into three categories, namely, the first principle model, the grey-box model and the data-driven model (black-box model). Models that are built based on physical principles such as mass and energy balance are known as first principle models [1]. On the other hand, models that are developed from the historical data without the process knowledge are called data-driven or black-box models. The grey box model structure is based on both insight into the system and experimental data, although in such a framework, peculiarities of what is going on inside the system are not entirely known. In industrial process control practice, the first principle models are usually difficult to obtain due to the complex nature of chemical processes and inability to measure critical variables. Thus, data-driven models and grey-box models have attracted considerable attention and have been extensively applied in industry.

Despite the existence of various model structures, static models and dynamic models are the two types of data-driven models that have been widely studied in literature. Commonly used static modeling techniques are based on regression methods such as Ordinary Least Squares (OLS) regression. The disadvantage of OLS is that it may suffer from numerical problems when collinearity among regression variables exists which is not uncommon in practice. This issue can be addressed by using Principal Component Regression (PCR) and Partial Least Squares (PLS) regression which project the original process variables into a low dimensional space of orthogonal latent variables. In doing this, the variation in both input and output variables are simultaneously explained and the their covariance is maximized.

In contrast with static models, dynamic models take into account the evolution of the process outputs over time. Commonly used dynamic models include State Space models, Autoregressive models (e.g., ARX, ARMAX), Output-Error (OE) models, Box-Jenkins (BJ) models. Aspects of system identification have been discussed extensively in literatures [1]. Among various models, the nonlinear state space model is a general class of models to represent nonlinear

dynamic systems. Therefore it is chosen for soft sensor model development, and parameter estimation of the state space model is studied in this chapter. Recently, particle filters which belong to the family of Sequential Monte Carlo (SMC) [4] methods, have been combined with the EM algorithm [9], [5] for the purpose of parameter estimation. The parameter estimation method reported in [9] approximated the log-likelihood function in the EM algorithm with particle filters and smoothers. Following the point-wise state estimation technique, the smoothed density functions of each state have to be calculated at every iteration in the EM algorithm, which introduces a large computational burden. Based on [9], particle filter approximation is employed in this chapter for computing expectation functions. The use of a particle filter rather than a smoother can significantly reduce the computation load.

The remainder of this chapter is organized as follows: Section 2 introduces the parameter estimation for nonlinear state space models under the framework of the Expectation-Maximization (EM) algorithm. Section 3 presents the model calibration strategy using a Bayesian approach. A simulation example is illustrated in section 4 to demonstrate the effectiveness of the proposed approach. Section 5 presents a soft sensor application in an oil sands process using the proposed approach. Section 6 draws the conclusion based on the results obtained in this chapter.

3.2 Parameter Estimation of the State Space Model

3.2.1 Model structure

Consider the state space model given by

$$x_t = f(x_{t-1}, u_{t-1}, \Theta) + \omega_t \quad (3.1)$$

$$y_t = h(x_t, \Theta) + v_t \quad (3.2)$$

where the system parameters are Θ . x_t, u_t, y_t, ω_t and v_t are state, measured input, measured output, process noise and measurement noise, respectively; ω_t and v_t are independent and identically distributed Gaussian noises with covariance matrices Q , and R respectively. The input sequence $\{u_1, \dots, u_t\}$ is known. Here the parameters in the state space model, Θ , are of interest. Let X denote the sequence of hidden states $\{x_1, \dots, x_T\}$ and $Y_o = \{y_1, \dots, y_T\}$ stand for the corresponding observed outputs. In the following section, we show how to formulate the parameter estimation problem under the framework of the EM algorithm.

3.2.2 Formulation of the parameter estimation based on the EM algorithm

An introduction to the Expectation-Maximization algorithm has been presented in Chapter 2. Consider the state-space model described in Equation (3.1-3.2). The observed output data Y_{obs} are $Y_o = \{y_1, \dots, y_T\}$ while the hidden states $X = \{x_1, \dots, x_T\}$ can be viewed as the latent data Z . Let $p(x_{1:T}, y_{1:T}|\Theta)$ denote the complete likelihood function of the hidden states and observations. The Q function is defined as the expectation of the log-likelihood function $\log[p(x_{1:T}, y_{1:T}|\Theta)]$ which is an integral given by

$$\begin{aligned} Q(\Theta|\Theta^k) &= E_{Z|Y_{obs}, \Theta^k} \{ \log[p(Y_{obs}, Z|\Theta)] \} \\ &= E_{X|Y_o, \Theta^k} \{ \log[p(Y_o, X|\Theta)] \} \\ &= \int_X \log[p(Y_o, X|\Theta)] p(X|Y_o, \Theta^k) dX \\ &= \int_X \log[p(x_{1:T}, y_{1:T}|\Theta)] p(x_{1:T}|y_{1:T}, \Theta^k) dx_{1:T} \end{aligned} \quad (3.3)$$

Following the approach of [9], the Q function can be derived as follows. In Equation 3.3, the first term which is the joint density function of states and outputs can be decomposed using the Markov property as

$$p(x_{1:T}, y_{1:T}|\Theta) = p(x_{1:T}|\Theta) p(y_{1:T}|x_{1:T}, \Theta) \quad (3.4)$$

where

$$\begin{aligned} p(x_{1:T}|\Theta) &= p(x_T|x_{1:T-1}, \Theta) p(x_{1:T-1}|\Theta) \\ &= p(x_T|x_{1:T-1}, \Theta) p(x_{T-1}|x_{1:T-2}, \Theta) \cdots p(x_2|x_1, \Theta) p(x_1|\Theta) \\ &= p(x_T|x_{T-1}, \Theta) p(x_{T-1}|x_{T-2}, \Theta) \cdots p(x_2|x_1, \Theta) p(x_1|\Theta) \\ &= p(x_1|\Theta) \prod_{t=2}^T p(x_t|x_{t-1}, \Theta) \end{aligned} \quad (3.5)$$

$$\begin{aligned} p(y_{1:T}|x_{1:T}, \Theta) &= p(y_T|y_{1:T-1}, x_{1:T}, \Theta) \cdot p(y_{1:T-1}|x_{1:T}, \Theta) \\ &= p(y_T|y_{1:T-1}, x_{1:T}, \Theta) \cdot p(y_{T-1}|y_{1:T-2}, x_{1:T}, \Theta) \cdots p(y_2|y_1, x_{1:T}, \Theta) \\ &= p(y_T|x_T, \Theta) p(y_{T-1}|x_{T-1}, \Theta) \cdots p(y_1|x_1, \Theta) \\ &= \prod_{t=1}^T p(y_t|x_t, \Theta) \end{aligned} \quad (3.6)$$

Substituting Equation 3.5 and 3.6 into Equation 3.4, the joint density function of the states and the outputs is rewritten as

$$p(x_{1:T}, y_{1:T}|\Theta) = p(x_1|\Theta) \prod_{t=2}^T p(x_t|x_{t-1}, \Theta) \cdot \prod_{t=1}^T p(y_t|x_t, \Theta) \quad (3.7)$$

Furthermore, substituting Equation 3.7 into Equation 3.3, the Q function is given by

$$\begin{aligned}
Q(\Theta|\Theta^k) &= \int \log[p(x_{1:T}, y_{1:T}|\Theta)] \cdot p(x_{1:T}|y_{1:T}, \Theta^k) dx_{1:T} \\
&= \int \log[p(x_1|y_{1:T}, \Theta)] \cdot p(x_{1:T}|y_{1:T}, \Theta^k) dx_{1:T} \\
&\quad + \sum_{t=2}^T \int \log[p(x_t|x_{t-1}, \Theta)] p(x_{1:T}|y_{1:T}, \Theta^k) dx_{1:T} \\
&\quad + \sum_{t=1}^T \int \log[p(y_t|x_t, \Theta)] p(x_{1:T}|y_{1:T}, \Theta^k) dx_{1:T} \tag{3.8}
\end{aligned}$$

In order to evaluate the Q function in Equation 3.8, the values of density function $p(x_{1:T}|y_{1:T}, \Theta^k)$ are required. Since direct calculation are intractable, this density function is to be numerically calculated using particle filters in the next section.

3.2.3 Computation through particle filtering

The basic idea of particle filters is to represent the desired posterior density function by a series of particles with associated weights, i.e. $\{x_t^i, w_t^i\}_{i=1}^N$. Then the density function of the states given the current estimation of parameters Θ^k can be discretely approximated as [12]

$$p(x_t|y_{1:T}, \Theta^k) \approx \sum_{i=1}^N \omega_t^i \delta(x_t - x_t^i) \tag{3.9}$$

where $\delta(\cdot)$ is the Dirac delta function, $t_\beta \leq t$; N is the number of particles; ω_t^i is the normalized weight associated with the i^{th} particle such that $\sum_{i=1}^N \omega_t^i = 1$. Suppose that at time $t-1$, a set of particles $\{x_{1:t-1}^i\}_{i=1}^N$ are available and we want to obtain N particles which represent the hidden state for time t . Since it is usually difficult to directly draw samples from the true posterior density $p(x_t|y_{1:T}, \Theta^k)$, the principle of importance sampling [10] is adopted. The idea is to use a so called importance density $q(\cdot)$ from which one can easily draw samples $x_t^i, i = 1, \dots, N$. Then the posterior is obtained by resampling important sampling. It has been shown that, as long as the support region of the posterior density belongs to that of the importance density, the particle approximation is unbiased [11]. The importance sampling is commonly chosen as the probability of state transition, i.e.

$$q(x_t|y_{1:T}, \Theta^k) = p(x_t|x_{t-1}, \Theta^k) \tag{3.10}$$

With this choice, the unnormalized weight for each particle can be derived as [12]

$$\tilde{\omega}_t^i \propto p(y_t|x_t^i, \Theta^k) \tag{3.11}$$

$$\omega_t^i = \frac{\tilde{\omega}_t^i}{\sum_{i=1}^N \tilde{\omega}_t^i} \quad (3.12)$$

To avoid the degeneracy problem [12], the importance sampling step is usually followed by a resampling procedure. The idea is to discard the particles with small weights and concentrate on those with large weights. After resampling, each particle's weight will be reset to $\omega_t^i = \frac{1}{N}$.

The problem brought by brute force resampling is that it reduces the diversity among particles. One solution is to resample the particles only when it is necessary instead of performing it at each step. To be specific, N_{eff} is introduced to represent the effective particle number [13]

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\omega_t^i)^2} \quad (3.13)$$

where ω_t^i is the normalized weight obtained by Equation 3.12. It implies that, as the variance of the weights grows very large, the effective sample size decreases to a small number which indicates a severe degeneracy problem. In practice, one uses resampling to eliminate useless particles only when a severe degeneracy problem occurs, say, N_{eff} falls below the threshold N_{thred} .

Given the current estimation of parameters, the particle filter algorithm is summarized as follows:

Step 1. Initialization. Draw initial N particles $\{x_0^i\}_{i=1}^N$ from the prior density $p(x_0|\Theta^k)$ and set each particle's weight to $\frac{1}{N}$. Set $t=1$.

Step 2. Importance sampling. Generate predicted particles $\{x_t^i\}_{i=1}^N$ from the importance density $p(x_t|x_{t-1}, \Theta^k)$.

Step 3. Assigning weights. Assign the weight to each particle using Equation 3.11 and 3.12.

Step 4. Resampling. Compute the number of effective particles using Equation 3.13. If N_{eff} is less than the threshold N_{thred} , then perform resampling and replace the predicted particles in Step 2 with resampled particles. Reset the weights of resampled particles uniformly as $\omega_t^i = \frac{1}{N}$. Otherwise, go to Step 5.

Step 5. Set $t = t + 1$ and repeat Step 2 to Step 4 for $t \leq T$.

Estimation of $p(x_{1:T}|y_{1:T}, \Theta^k)$ is a problem of smoothing all states with all available observations. Its computation with the iterative EM algorithm is formidable for on-line application. With further marginalization of the states following the approach of [9], the Q function obtained in Equation 3.8 can be

rewritten as

$$\begin{aligned}
Q(\Theta|\Theta^k) &= \int \log[p(x_1|y_{1:T}, \Theta)] \cdot p(x_1|y_{1:T}, \Theta^k) dx_1 \\
&+ \sum_{t=2}^T \int \log[p(x_t|x_{t-1}, \Theta)] p(x_{t-1:t}|y_{1:T}, \Theta^k) dx_{x-1:t} \\
&+ \sum_{t=1}^T \int \log[p(y_t|x_t, \Theta)] p(x_t|y_{1:T}, \Theta^k) dx_t \tag{3.14}
\end{aligned}$$

Calculation of $p(x_t|y_{1:T}, \Theta^k)$ and $p(x_{t-1}, x_t|y_{1:T}, \Theta^k)$ is also a smoothing problem, of which the computation cost is high. A practical solution is to apply recursive state filtering such that $p(x_t|y_{1:T}, \Theta^k)$ is recursively approximated by $p(x_t|y_{t_1:t_\beta}, \Theta^k)$ for $t = 1 : T$, and $p(x_{t-1}, x_t|y_{1:T}, \Theta^k)$ is recursively approximated by $p(x_t, x_{t+1}|y_{t_1:t_\beta}, \Theta^k)$ for $t = 1 : T-1$, where $t_\beta \leq t$. This solution can significantly reduce the computation complexity and thus make the solution possible in real-time applications.

In Equation 3.14, the density function $p(x_t|y_{1:T}, \Theta^k)$ is approximated using particle filters as

$$\begin{aligned}
p(x_t|y_{1:T}, \Theta^k) &\approx p(x_t|y_{t_1:t_\beta}, \Theta^k) \\
&= \sum_{i=1}^N \omega_t^i \delta(x_t - x_t^i) \tag{3.15}
\end{aligned}$$

As for the joint density function of x_t and x_{t+1} , it can be approximated as

$$\begin{aligned}
p(x_t, x_{t+1}|y_{1:T}, \Theta^k) &\approx p(x_t, x_{t+1}|y_{t_1:t_\beta}, u_{1:t}, \Theta^k) \\
&= p(x_{t+1}|x_t, \Theta^k) p(x_t|y_{t_1:t_\beta}, \Theta^k) \\
&= \sum_{i=1}^N \omega_{t|t+1}^i \delta(x_t - x_t^i) \delta(x_{t+1} - x_{t+1}^i) \tag{3.16}
\end{aligned}$$

where

$$\omega_{t|t+1}^i = \frac{p(x_{t+1}^i|x_t^i, \Theta^k) p(x_t^i|y_{t_1:t_\beta}, \Theta^k)}{\sum_{i=1}^N p(x_{t+1}^i|x_t^i, \Theta^k) p(x_t^i|y_{t_1:t_\beta}, \Theta^k)} \tag{3.17}$$

Substituting these approximated density functions, the Q function in Equation 3.14 can be finally obtained.

$$\begin{aligned}
Q(\Theta|\Theta^k) &\approx \sum_{i=1}^N \omega_1^i \log p(x_1^i|\Theta) \\
&+ \sum_{t=2}^T \sum_{i=1}^N \omega_{t-1|t}^i \log p(x_t^i|x_{t-1}^i, \Theta) \\
&+ \sum_{t=1}^T \sum_{i=1}^N \omega_t^i \log p(y_t^i|x_t^i, \Theta) \tag{3.18}
\end{aligned}$$

With the approximated Q function, the EM algorithm can hence be implemented. In the expectation step, the Q function is evaluated according to Equation 3.18 with the current estimated parameters Θ^k . In the next maximization step, the new parameters Θ^{k+1} , are obtained by maximizing the Q function.

To maximize the Q function over parameters Θ , derivative operation is performed with respect to each parameter. Therefore, optimal estimation of system parameters at each iteration can be calculated by equating the derivatives to zero, i.e. $\frac{\partial Q}{\partial \theta_j} = 0$, where θ_j is the j^{th} system parameter.

The EM algorithm is summarized as follows:

Step 1. Initialization. Start with the initial parameters Θ , and set $t=0$.

Step 2. Expectation. At time t , calculate the approximate Q function using Equation 3.18, given the current estimation of the system parameters Θ^k .

Step 3. Maximization. Maximize the approximated Q function and get the new parameters Θ^{k+1} . Set $k=k+1$.

Step 4. Repeat Step 2 and Step 3 until the converge condition is satisfied, i.e. the change of the estimated parameters between two iterations is less than the tolerance.

3.3 Bayesian Calibration

3.3.1 Formulation for online calibration

Since the estimated model from the previous section always has model-plant mismatch, the model needs to be calibrated on-line which synthesizes the measurement information of both fast sampled online analyzer and slow rate laboratory analysis. The calibrations equations are described below::

$$x_t = \rho_t(f(x_{t-1}, u_{t-1})) + b_t + \omega_t \quad (3.19)$$

$$\rho_t = \rho_{t-1} + \omega_t^p \quad (3.20)$$

$$b_t = b_{t-1} + \omega_t^b \quad (3.21)$$

$$b_t^o = b_{t-1}^o + \omega_t^o \quad (3.22)$$

$$y_t^o = x_t + b_t^o + v_t^o \quad (3.23)$$

$$y_t^L = x_t + v_t^L \quad (3.24)$$

where ρ_t is a scaling parameter that brings some flexibility to the model-plant mismatch compensation. A bias term b_t is introduced to correct the systematic error. It is observed that the online analyzer has some deviation from the laboratory analysis, hence another bias term b_t^o is adopted to capture the difference. The calibration procedure is performed using recursive particle filtering as well. x_t , ρ_t , b_t and b_t^o are four states while the online analyzer reading

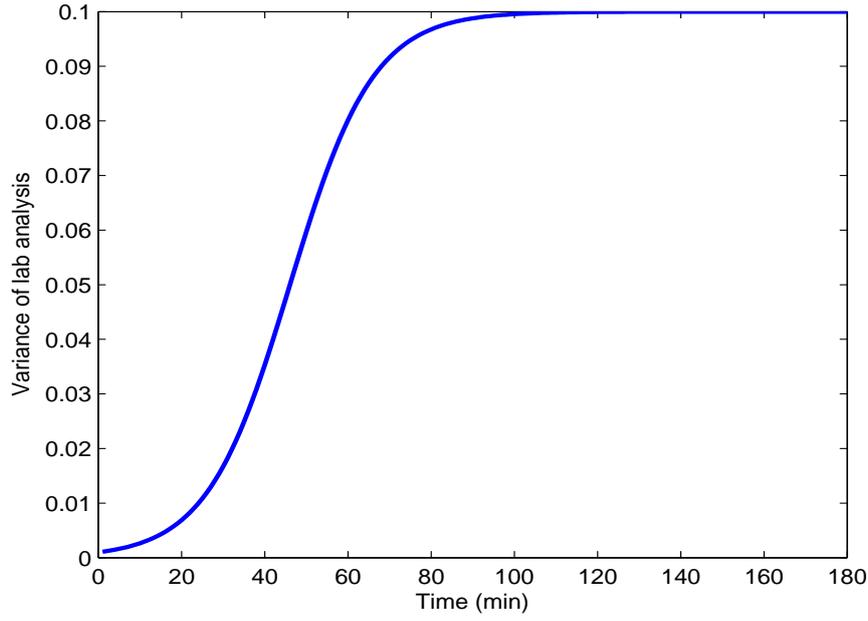


Figure 3.1: The trajectory of the variance of lab analysis

y_t^o and the laboratory analysis y_t^L are two sources of output measurement information.

The particle weight calculation when lab data is available is given below,

$$\omega_t^i = \exp\left(-\frac{(y_t^o - y^{pre})^2}{2R_{o1}} - \frac{(y_t^l - y^{pre})^2}{2R_{l1}}\right) \quad (3.25)$$

When lab data is not available, the weight is calculated as,

$$\omega_t^i = \exp\left(-\frac{(y_t^o - y^{pre})^2}{2R_{o2}} - \frac{(y_t^l - y^{pre})^2}{2R_{l2}(t)}\right) \quad (3.26)$$

where the noise variance is treated as a tuning parameter: R_{o1} is the noise variance for online analyzer when lab data is available; R_{o2} is the noise variance for online analyzer when lab data is not available; R_{l1} is the noise variance for lab data when it is available. When lab data is not available, the previous lab value is used and $R_{l2}(t)$ is the noise variance which is a sigmoidnet function and increases with time.

$$R_{l2}(t) = \frac{1 \times 10^{-3}}{1 \times 10^{-2} + e^{-0.1t}} \quad (3.27)$$

The trajectory of $R_{l2}(t)$ is shown in Figure 3.1.

3.4 Semi-continuous Fermentation Example

A semi-continuous fermentation of baker's yeast is adopted here for illustration purpose of the proposed model estimation and calibration. This example is taken from [16]. Assuming that the biomass growth and substrate consumption follow the Monod-type kinetics, the two inputs two outputs fermentation system is described by the following equations:

$$\frac{dx_1}{dt} = \left(\frac{\theta_1 x_2}{\theta_2 + x_2} - u_1 - \theta_4 \right) x_1 \quad (3.28)$$

$$\frac{dx_2}{dt} = -\frac{\theta_3 x_1 x_2}{\theta_2 + x_2} + u_1 (u_2 - x_2) \quad (3.29)$$

where the output x_1 is the biomass concentration (g/l); output x_2 is the substrate concentration (g/l); input u_1 is the dilution factor h^{-1} ; and the input u_2 is the substrate concentration in the feed (g/l). Both inputs have a sampling rate of 1 minute. $\Delta t = 0.1$ minute is chosen as the discretizing sample time. $\theta_1, \theta_2, \theta_3$ and θ_4 are the system parameters with true values of $\theta_1 = 0.31, \theta_2 = 0.18, \theta_3 = 0.56$, and $\theta_4 = 0.05$. The process is simulated for 2400 minutes and the two inputs are shown in Figure 3.2.

A disturbance d_k is added to the state x_2 in Equation 4.32 such that

$$d_k = 0.5 \cdot \cos\left(\frac{k}{10 \cdot \pi}\right) + 0.2 \cdot n_k \quad (3.30)$$

where n_k is a non-Gaussian noise which is generated from a bimodal distribution such that with 70% of the time it is generated from a Gaussian distribution with a mean value of -0.2 and variance of 0.1^2 , and with 30% of the time it is generated from a Gaussian distribution with a mean value of 0.2 and variance of 0.1^2 . Suppose we are interested in the substrate concentration x_2 whose value is observed by two types of measurements. The first has a fast sampling rate of 1 minute, but is corrupted by a large measurement noise, hence is less accurate. The second measurement is more accurate with smaller noise, however, it is only available every 4 hours. Assuming that two parameters θ_1 and θ_3 is unknown. The parameter estimation algorithm proposed in section 2 is applied to this data set with the less accurate measurement as the output to obtain a nominal model. The initial guess for the two parameters are $\theta_1 = 0.1, \theta_3 = 0.1$. 100 particles are used for the particle filtering approximation. The estimation terminated after near 30 EM iterations and the parameter trajectories are shown in Figure 3.3. The model prediction result with the estimated parameters is shown in Figure 3.4.

It can be seen from Figure 3.3 and Figure 3.4 that both estimated parameters have converged to the neighborhood of the true values. However, the model prediction with the estimated parameters has an obvious mismatch compared to the true output.

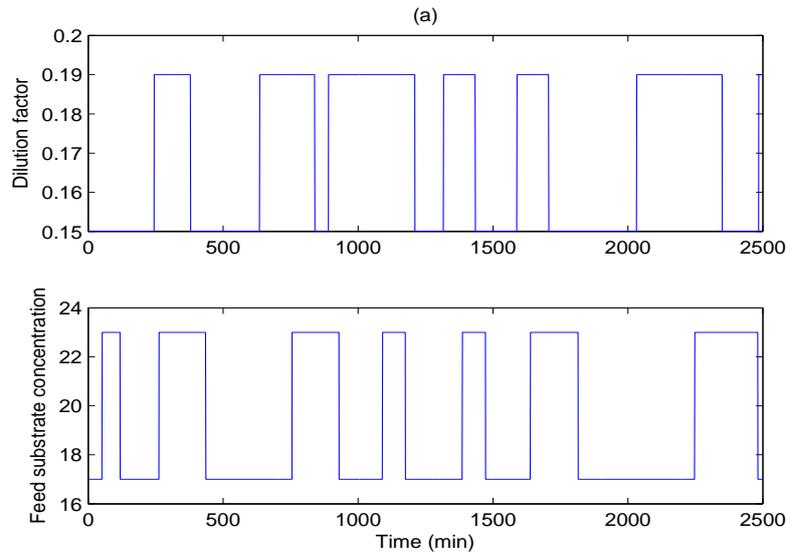


Figure 3.2: Process inputs. (a): u_1 , the dilution factor h^{-1} ; (b): u_2 , the substrate concentration in the feed.

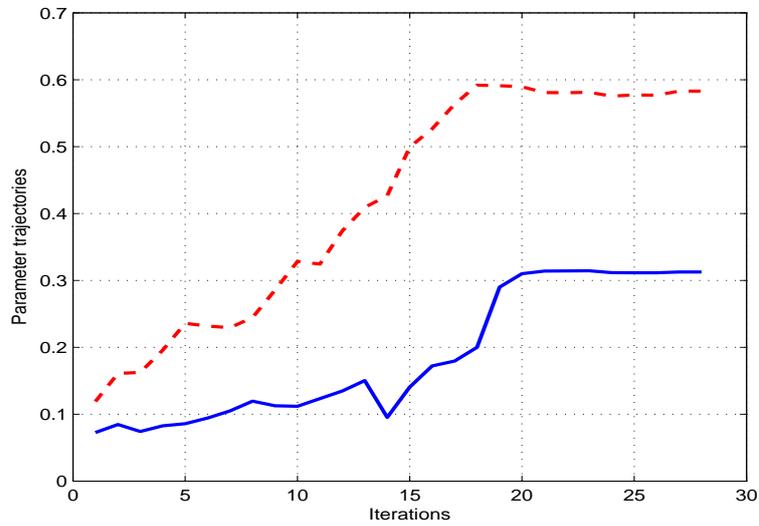


Figure 3.3: Estimated parameters. Blue line: trajectory of θ_1 ; Red line: trajectory of θ_3 .

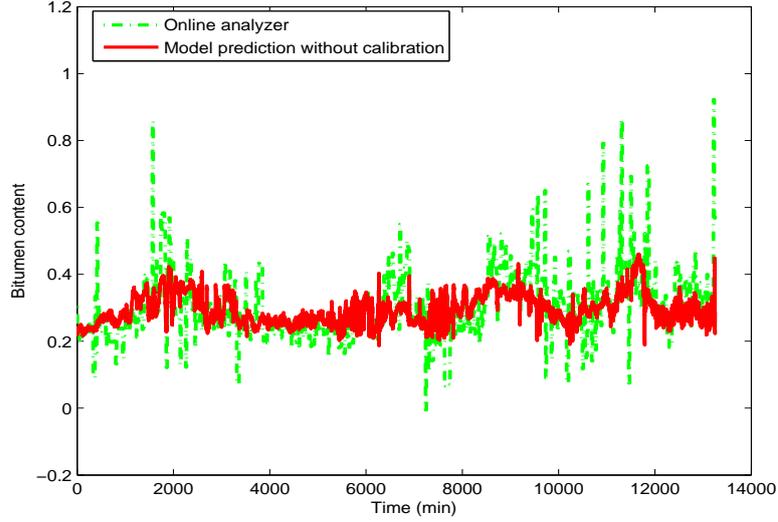


Figure 3.4: Comparison of current measurements.

In order to compensate the difference between the predicted model and the real process output, the proposed Bayesian calibration approach is applied through the following equations:

$$x_t = \rho_t x_t^{pre} + b_t + \omega_t \quad (3.31)$$

$$\rho_t = \rho_{t-1} + \omega_t^\rho \quad (3.32)$$

$$b_t = b_{t-1} + \omega_t^b \quad (3.33)$$

$$b_t^o = b_{t-1}^o + \omega_t^o \quad (3.34)$$

$$y_t^o = x_t + b_t^o + v_t^o \quad (3.35)$$

$$y_t^L = x_t + v_t^L \quad (3.36)$$

where x_t^{pre} is the predicted value for the substrate concentration x_2 from the estimated model. ω_t is the process noise which follows a Gaussian distribution $\omega_t \sim N(0, 0.15^2)$; v_t^o and v_t^L are measurement noises for the fast sampled measurement and slow sampled measurement which are chosen as $N(0, 0.7^2)$ and $N(0, 0.01^2)$. The calibration result is shown in Figure 3.5 and Figure 3.6, from which we can see that the estimation with the Bayesian calibration provides a superior result than the model prediction without calibration.

3.5 Industrial Application

3.5.1 Process description

The Primary Separation Vessel (PSV) is a settling gravity separation unit which is of critical importance in oil sands extraction process. It helps to

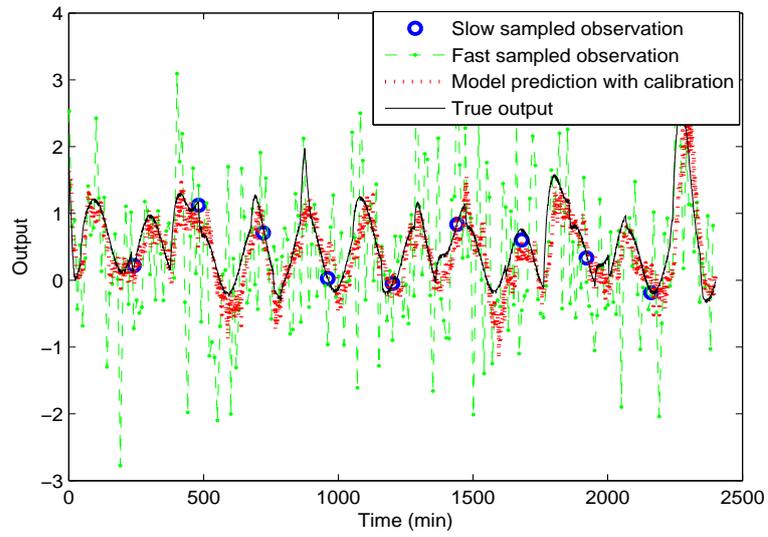


Figure 3.5: Comparison of different measurements.

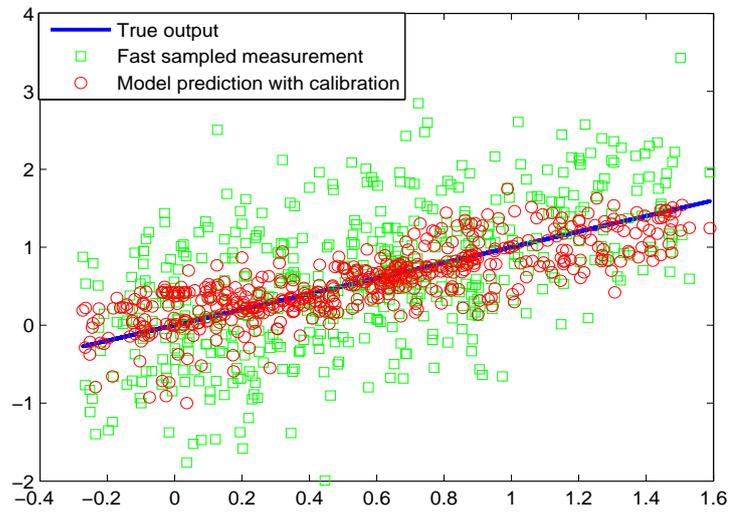


Figure 3.6: 45 degree comparison.

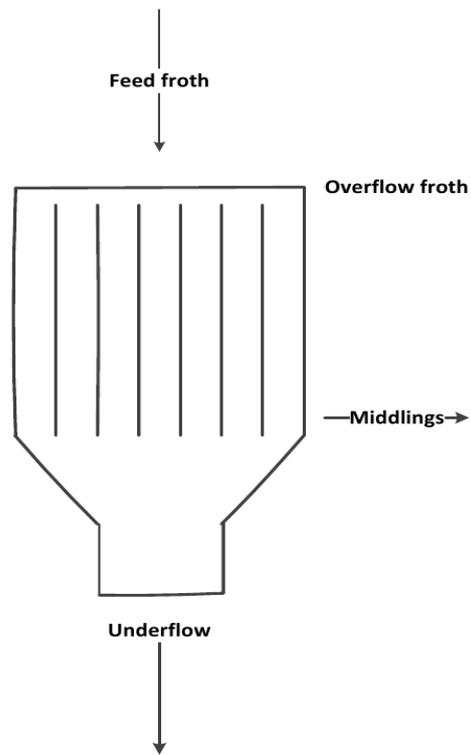


Figure 3.7: Schematic diagram of the Primary Separation Vessel

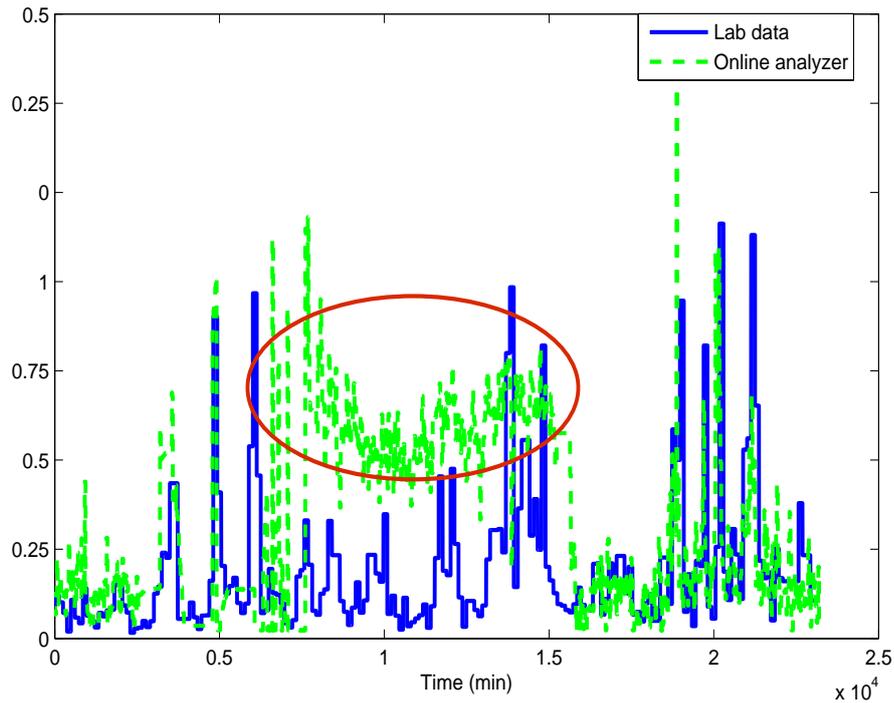


Figure 3.8: Comparison between online analyzer and laboratory analysis.

facilitate bitumen flotation and solids settling. The blended oil sands slurry is fed into the top of the PSV where it is separated into three phases under gravity and mechanical facilitation. The froth phase at the top of the vessel contains the majority of the bitumen, the middling phase contains those hard-to-separate clay particles which are attached to the bitumen, the coarse sand drops to the bottom of the vessel, and is withdrawn in a tailing stream.

The aim of the control strategy of the PSV is to maintain a material balance in the PSV vessel and to keep the underflow stream density at a stable value. The bitumen content in the bottom phase of the PSV is a particularly important variable as it reflects the performance of the separation process. In practice, both hardware instrument (e.g., online analyzer) and laboratory analysis for the underflow bitumen content are available. The online analyzer provides fast sampled measurement, but not accurate enough. In contrast, the off-line laboratory analysis provides more accurate measurement, but is only available every several hours. Figure 3.8 shows the comparison between online analyzer and lab data. It can be seen that in certain periods (red circled), the online analyzer has an obvious bias compared with the lab data. Therefore, obtaining an accurate and real-time measurement of the underflow bitumen content is desired.

3.5.2 Soft sensor development for PSV underflow bitumen content

Being aware that the separation process in the PSV is a continuous one and the underflow stream bitumen content is correlated with the previous value, it is more appropriate to adopt a dynamic model to describe the bitumen content. The laboratory analysis is irregularly sampled every several hours which is not suitable for the dynamic modeling. As an alternative, we are able to collect a sufficient amount of fast rate measurement (online analyzer) during a period when the performance of online analyzer was fairly reasonable. A state space model was developed using the on-line analyzer data. Noticing that the true process is nonlinear, a cosine term is introduced in order to capture the nonlinear dynamics in the process and this function was found to be suitable. Therefore a nonlinear state space model is formulated as follows:

$$\begin{aligned}x_t &= a \cdot x_{t-1} + B^T \cdot u_{t-1} + c \cos(x_{t-1}) + \omega_t \\y_t &= x_t + v_t\end{aligned}\quad (3.37)$$

where the system parameters are $\Theta = [a \ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ c]^T$, and $u_t = [u_{1t} \ u_{2t} \ u_{3t} \ u_{4t} \ u_{5t}]^T$ are the five process inputs including underflow stream density, middling stream density, PSV interface level, middling stream pump flow rate and the underflow stream pump flow rate respectively. Figure 3.9 shows the process input data. For proprietary reason, all data has been normalized. x_t, y_t, ω_t and v_t are state, measured output (online analyzer reading), process noise and measurement noise, respectively; ω_t and v_t are independent and identically distributed Gaussian noises with covariance matrices Q , and R respectively. Here the parameters in the state space model, Θ , are of interest.

The proposed parameter estimation algorithm in section 2 is applied to estimate the parameters Θ in the state space model. In the expectation step of the EM algorithm, the Q function is calculated according to Equation 3.18, where

$$\log[p(x_t|x_{t-1}^i, \Theta)] = \log\left[\frac{1}{\sqrt{2\pi Q}} \exp\left[-\frac{1}{2} \frac{(x_t^i - ax_{t-1}^i - B^T u_{t-1} - A \cos(x_{t-1}^i))^2}{Q}\right]\right]\quad (3.38)$$

$$\log[p(y_t|x_t^i, \Theta)] = \log\left[\frac{1}{\sqrt{2\pi R}} \exp\left[-\frac{1}{2} \frac{(y_t - x_t^i)^2}{R}\right]\right]\quad (3.39)$$

By taking derivative over the Q function and equating it to zero, each individual component of the parameters is hence calculated as

$$a_{new} = \frac{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i (x_t^i x_{t-1}^i - B_{old}^T u_{t-1} x_{t-1}^i - c \cos(x_{t-1}^i))}{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i (x_{t-1}^i)^2}\quad (3.40)$$

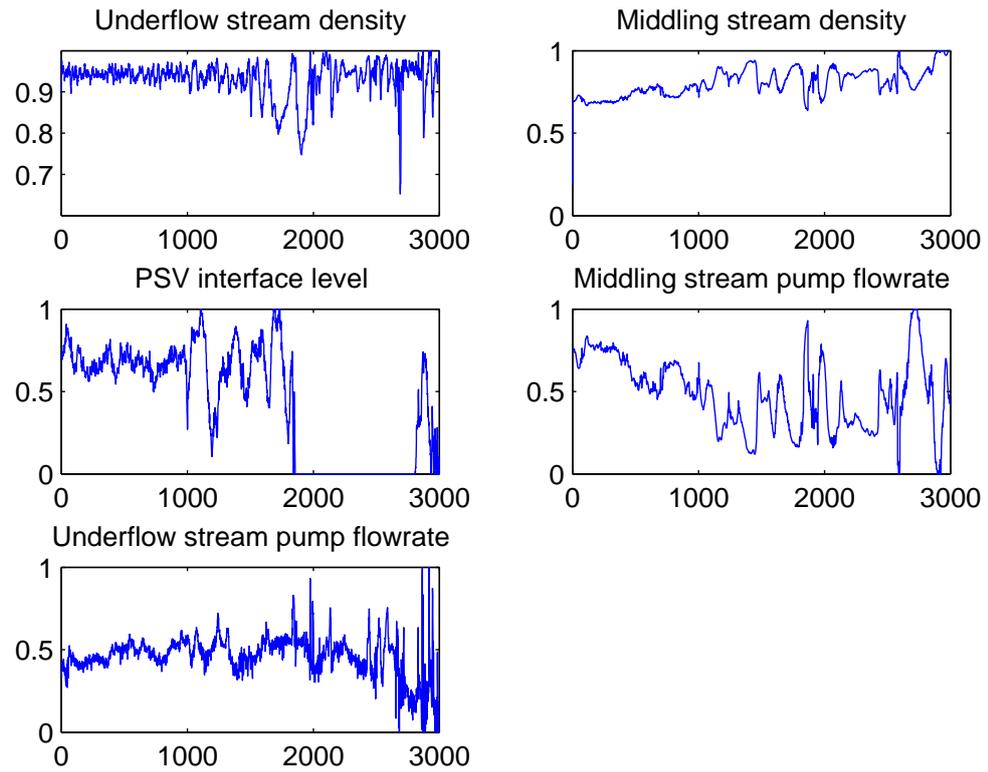


Figure 3.9: Input data for the PSV unit

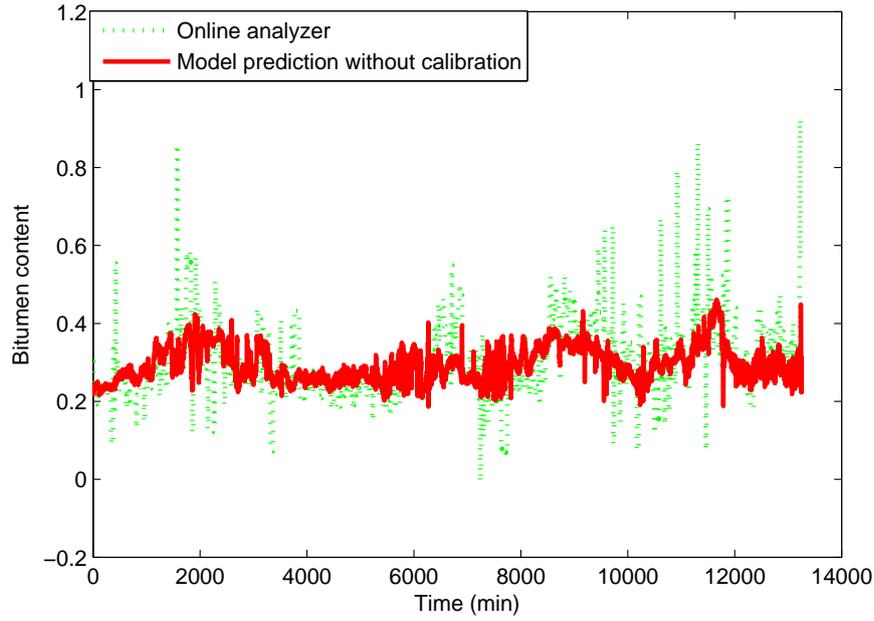


Figure 3.10: Model cross validation result

Table 3.1: MSE comparison of different measurements

	March 2010	December 2009	November 2009
Online analyzer	0.0078	0.0133	0.0425
Soft sensor (EKF)	0.0041	0.029	0.0241
Soft sensor (proposed method)	0.0039	0.008	0.0186

$$b_{j,new} = \frac{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i (x_t^i u_{t-1} - a_{old} u_{t-1} x_{t-1}^i - \sum_{r=1, r \neq j}^5 b_{r,old} u_{r,t-1} - c \cos(x_{t-1}^i))}{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i u_{j,t-1}^2} \quad (3.41)$$

$$c_{new} = \frac{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i (x_t^i - a x_{t-1}^i - B_{old}^T u_{t-1})}{\sum_{t=2}^T \sum_{i=1}^N \omega_t^i (\cos(x_{t-1}^i))^2} \quad (3.42)$$

The proposed parameter estimation algorithm is then applied. 100 particles are used for the particle filtering approximation. The parameter estimation starts from the initial guess of $a = 0.5$, $b_1 = b_2 = b_3 = b_4 = b_5 = 0.1$, $c = 1$. The cross validation result is shown in Figure 3.10.

The algorithm was tested using three different data sets collected in March 2010, December 2009 and November 2009, respectively. To further demonstrate the effectiveness of the proposed method, an comparative study which uses extended kalman filter for online calibration has been conducted.

The performance comparison in terms of MSE and MAE is shown in Table 3.1 and Table 3.2.

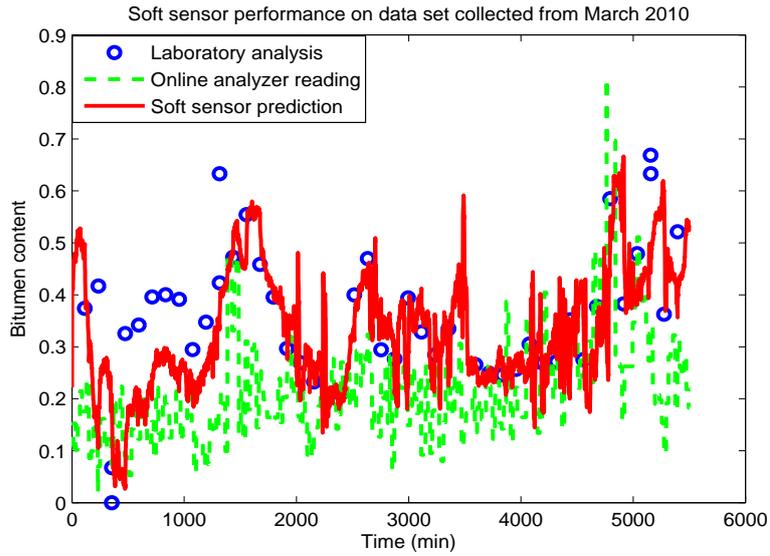


Figure 3.11: Trend comparison.

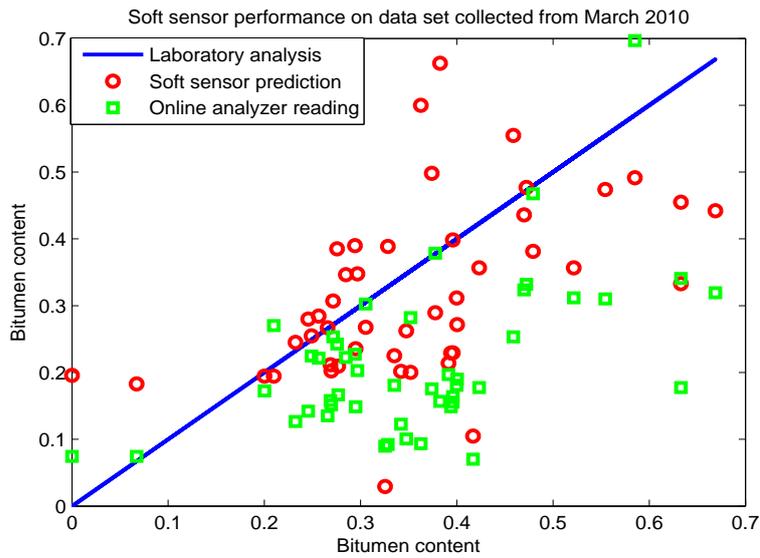


Figure 3.12: 45 degree comparison.

Table 3.2: MAE comparison of different measurements

	March 2010	December 2009	November 2009
Online analyzer	0.0469	0.0888	0.1663
Soft sensor (EKF)	0.0671	0.0688	0.1094
Soft sensor (proposed method)	0.0731	0.0651	0.0928

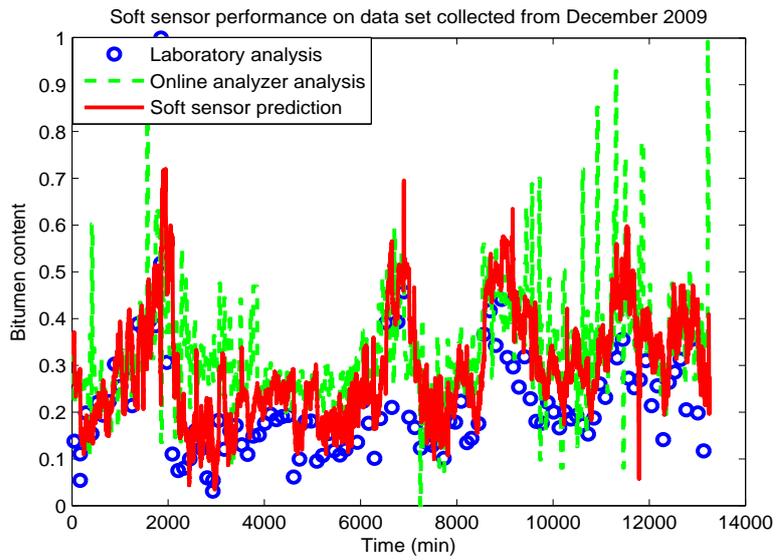


Figure 3.13: Trend comparison.

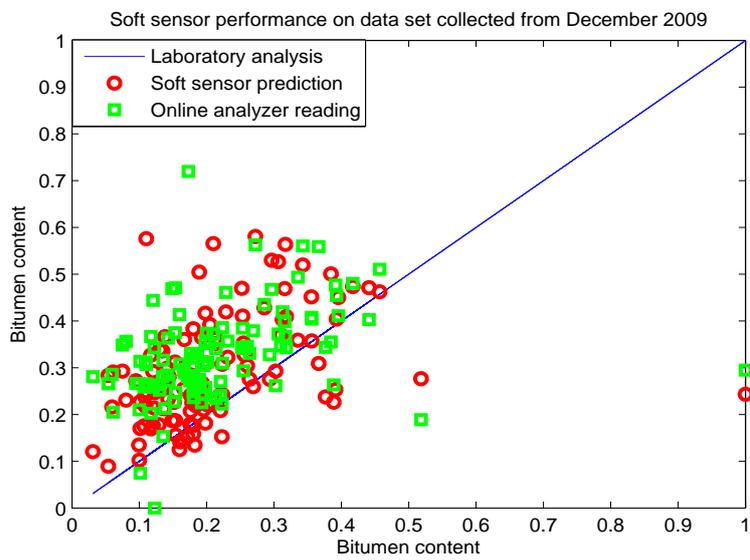


Figure 3.14: 45 degree comparison.

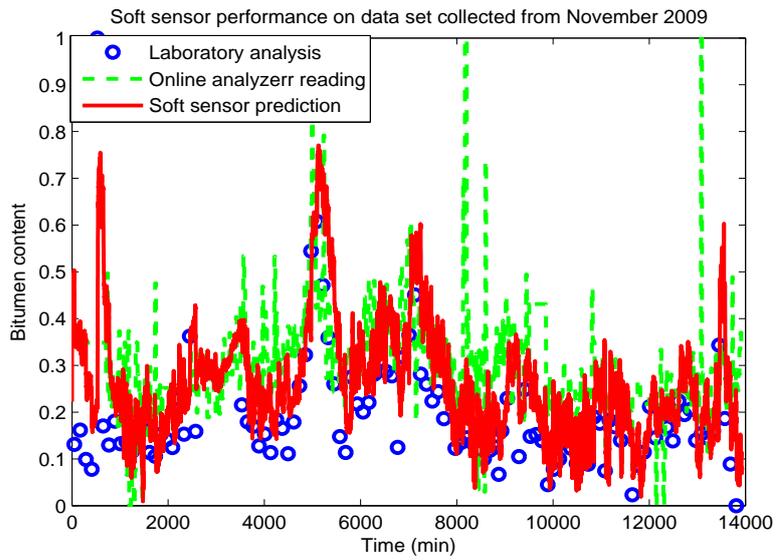


Figure 3.15: Trend comparison.

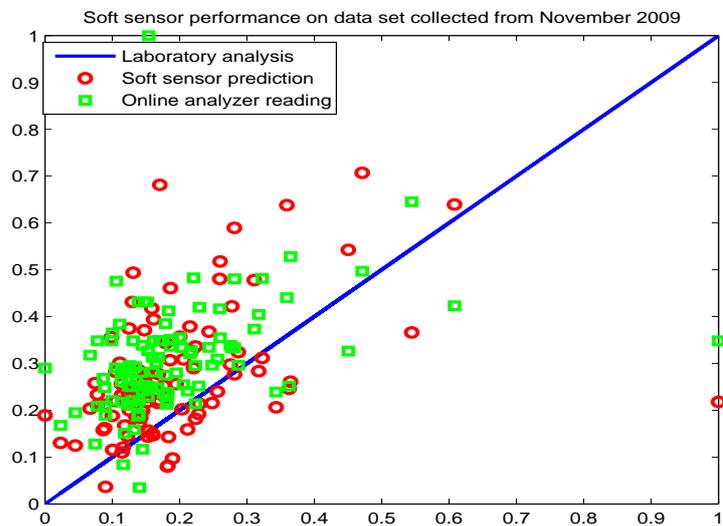


Figure 3.16: 45 degree comparison.

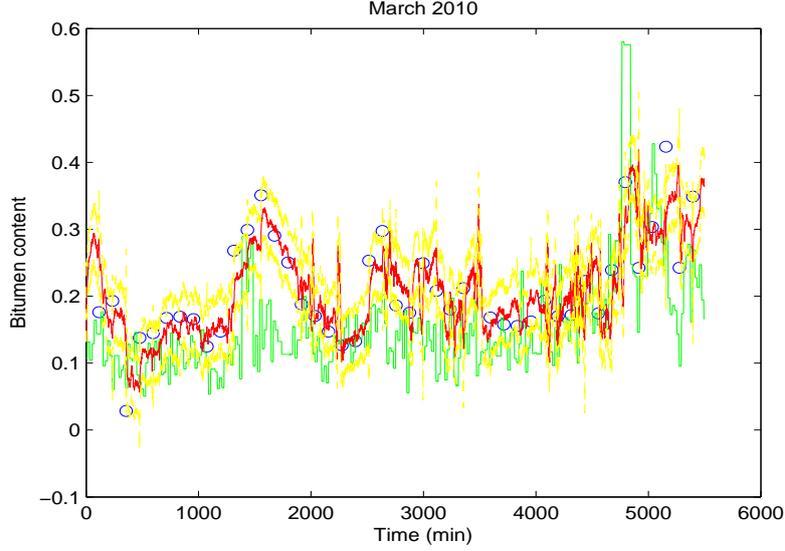


Figure 3.17: Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.

3.5.3 Performance index

When put the inferential sensor online, the performance index is required as an indication of how good the prediction of the soft sensor it is. Since the calibration procedure is conducted using particle filtering, it is convenient to adopt the standard deviation of particles σ_t as the performance index.

At each time stamp, all particles are assigned a weight according to their discrepancy to the output measurement. And the estimation value is the weighted summation of all particles such that

$$y_t^{pre} = \sum_{i=1}^N \omega_t^i x_t^i \quad (3.43)$$

Hence the performance index is then defined as follows

$$\sigma_t = \sqrt{\sum_{i=1}^N \omega_t^i (x_t^i - y_t^{pre})^2} \quad (3.44)$$

The soft sensor prediction with the 99% confidence interval are shown in Figure (3.17), Figure(3.18) and Figure(3.19).

The performance index is calculated to assess the reliability of the soft sensor. Figure 3.20 shows the distribution of the historical lab data which is collected from January 1, 2009 to August 17, 2011. Based on the statistical

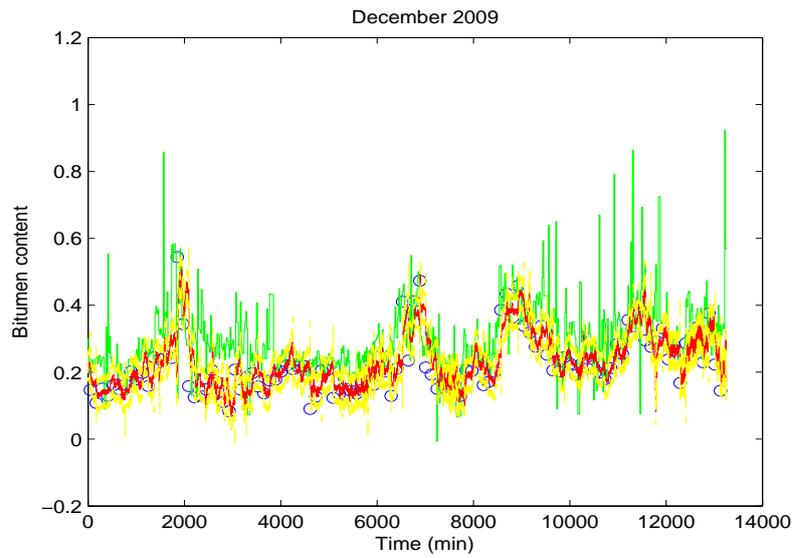


Figure 3.18: Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.

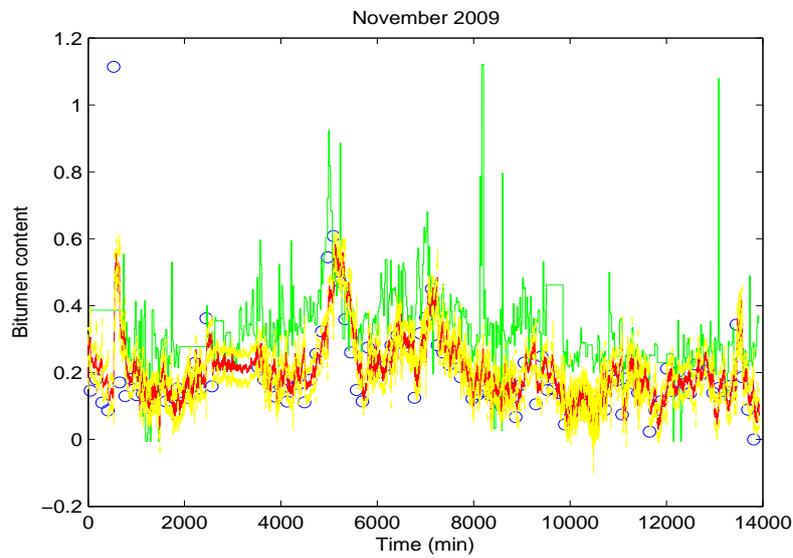


Figure 3.19: Trend comparison. The blue dot is laboratory analysis; the green line is online analyzer reading and the red line is soft sensor prediction; the two yellow dash lines defines the 95% confidence interval.

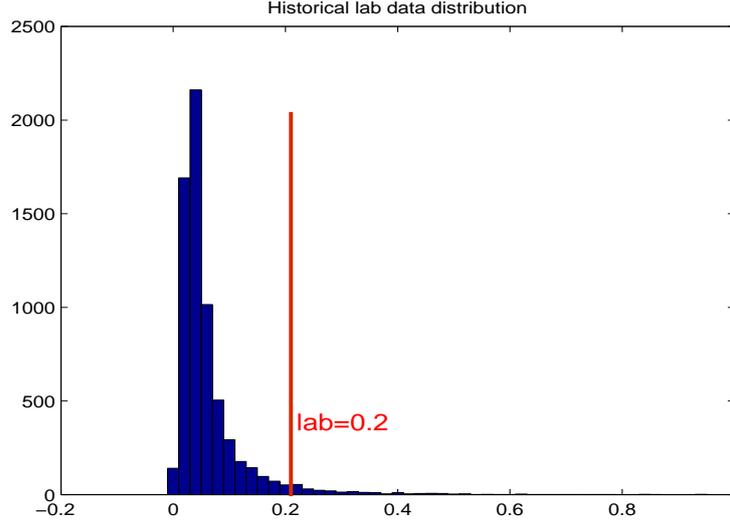


Figure 3.20: Distribution of historical lab data. Red line indicates that 95% of the historical lab data is less or equal than 1.1. For proprietary reason, the data has been normalized.

analysis on the lab data distribution, a threshold of 0.2 can be calculated according to the 95% confidence. Therefore, when the soft sensor reading is within this 95% confidence region (less or equal to 1.1), it is considered as reliable and the performance index is given as 1; when the soft sensor reading falls out of the 95% confidence interval region, its performance is calculated as follows:

$$\begin{aligned}
 & 1; \quad \text{when } 0 < y_{soft} < 1.1 \\
 & 1 - \frac{|y_{soft} - 1.1|}{Lab_{max} - 1.1}; \quad \text{when } y_{soft} < 0 \quad \text{or} \quad y_{soft} > 1.1 \quad (3.45)
 \end{aligned}$$

where Lab_{max} is the maximum value of the historical lab data.

3.5.4 Online implementation

The PSV underflow bitumen content soft sensor is implemented online for monitoring. The implementation procedure is as follows:

Step 1. The process data are real time variables in Distributed Control System (DCS).

Step 2. Selected variables are collected from DCS and sent to MATLAB through OPC server.

Step 3. The soft sensor readings are calculated in MATLAB using the developed algorithm.

Step 4. The soft sensor readings are sent back to DCS through OPC server.

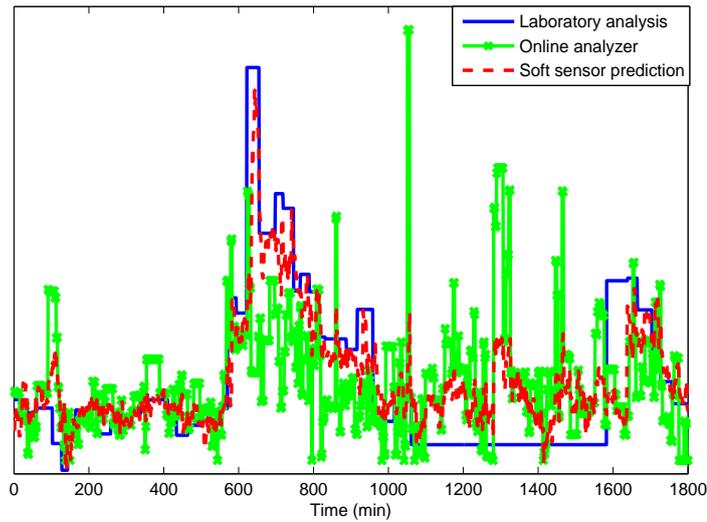


Figure 3.21: Soft sensor online monitoring result.

Step 5. The soft sensor performance is monitored in PI Historian

The online monitoring results is shown in Figure 3.21. It can be seen that the soft sensor gives superior estimation than the online analyzer.

3.6 Conclusion

This chapter deals with identification of nonlinear state space models and model updating issues for soft sensor development. A Bayesian based model calibration scheme is formulated where particle filters are adopted for the online calibration. The proposed approach synthesizes the information from both online analyzer (fast sampled) and laboratory analysis (slow sampled), and provides a improved estimation of the quality variable. The efficiency of the proposed method is illustrated by a semi-fermenter simulation example and has been applied to an oil sands process.

Bibliography

- [1] P. Kadleca, B. Gabrys and S. Strandtb, Data-driven Soft Sensors in the Process Industry. *Computers and Chemicals Engineering*, 2009, vol 33, pp 795-814.
- [2] W. Yan, H. Shao and X. Wang. Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers and Chemicals Engineering*, 2004, vol 28, pp 1489-1498.
- [3] S. D. Grantham, L. H. Ungar, A First Principles Approach to Automation Troubleshooting of Chemical Plants. *Comput. Chem. Eng*, 1990, vol 14, pp 783-798.
- [4] A. Doucet, N. de Freitas and N. Gordon, Sequential Monte Carlo Methods in Practice, *Springer*, New York, 2001.
- [5] J. Kim, Ph.D Thesis, University of Pittsburgh, Pittsburgh, 2005.

Chapter 4

Soft Sensor Development for Time Delayed Processes with Bayesian Approach

This chapter deals with the time delay issues that are associated with soft sensor development. The previous chapter presents a Bayesian approach for soft sensor development which synthesizes multi-rate measurements. However, another challenge to soft sensor applications in many oil sands and chemical industries is that measurement of certain quality variables has a time delay. In this chapter, an augmented state space is constructed to deal with the delay problem which consists of the current state and the previous states. The Bayesian approach is applied to calibrate the soft sensor. The proposed method for states augmentation and Bayesian calibration is illustrated through a numerical simulation and a semi-fermenter simulation example.

4.1 Introduction

In chapter 3, a Bayesian calibration method is developed to synthesize different sources of measurement, both fast online instrument measurement and off-line laboratory analysis. A number of other filtering methods have also been developed to reduce the system noise in order to get a better estimation of the states.

State estimation studied in the previous chapters requires the measurement to be available at the time it is sampled. In other words, it is assumed that the measurement is transmitted to the filter instantly without time delay. However, in reality it is quite possible to have time delay between the instant that the measurement is taken by a sensor and the instant that the measurement is available to the filter [2]. This problem is referred as Out Of Sequence Measurement (OOSM) problem as discussed in the introduction chapter [3]. For example, in the oil sands industry, bitumen concentration in the slurry stream

is an important variable to monitor. The sample stream is collected every two hours by the laboratory technician and the bitumen content is analyzed under a series of experimental steps in the lab which would usually takes probably 30 minutes. Therefore the time that the analyzed result is available is 30 minutes afater the instant that the stream sample is collected.

Several methods have been proposed to deal with OOSM problem. Bar-Shalom [3] used backward prediction to obtain the past state by applying the inverse model of the process model. The method for both one step delay and multi steps delay has been presented in [3] and [4]. However this method is limited to linear process model because it is difficult to get the inverse nonlinear model. Challa et al. [5] used the augmented state Kalman Filter (ASKF) to solve the time delay problem where the uncertainty of the delay is resolved by Probabilistic Data Association Filter (PDAF). In both above scenarios, the measurement is available at every sampling time except for the time delay problem. In practice, however, some measurements are only available every certain period in addition to time delay. Therefore an fusion approach that incorporate all the available sensor information while in the same time it can deal with the time delay issue is needed.

In this chapter, the state is augmented such that both current state and previous states are included. The augmented state is estimated using particle filter. To fuse both fast sampled and slow sampled measurements, the Bayesian calibration approach that is proposed in Chapter 3 is applied.

The remainder of this chapter is organized as follows: Section 2 introduces the model structure that contains the time delay problem. Section 3 presents the model calibration strategy using the Bayesian approach. A numerical simulation example and semi-continuous fermentation example are illustrated in section 4 to demonstrate the effectiveness of the proposed approach. Section 5 draws the conclusion based on the results obtained in this chapter.

4.2 Problem Statement

Consider the following model:

$$y_t = f(x_{t-1}, u_{t-1}, \Theta) + \omega_t \quad (4.1)$$

where the system parameters are Θ . x_t, u_t and ω_t are state, measured input and process noise respectively. In many chemical industries, there are two sources of measurement for the outputs, for example, fast rate online instrument and off-line laboratory analysis that is usually slow sampled. Figure 4.1 illustrates the two sources of measurement. In addition to the low sampling frequency, time delay also exists. Figure 4.2 shows the slow sampled lab data both with and without time delay. The green dot represents the fast sampled measurement which is available every sampling time; the blue dot is slow

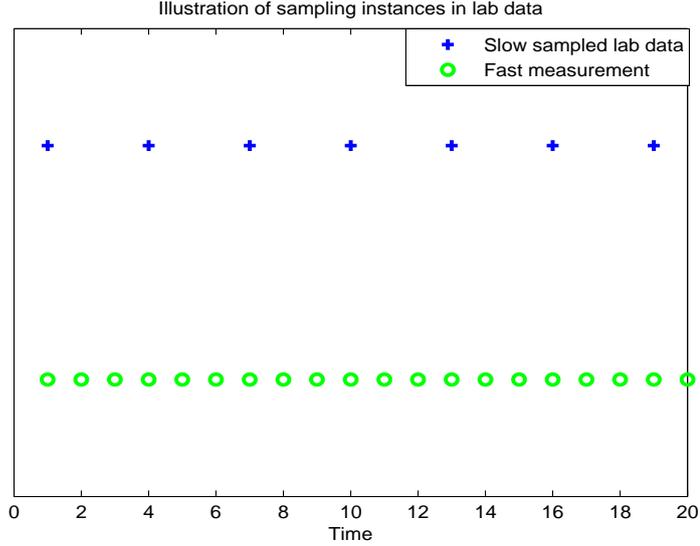


Figure 4.1: Slow sampled lab data without time delay.

sampled lab data. As it can be seen in Figure 4.2a, the lab data is available every 3 sampling time; in Figure 4.2b the slow sampled data is delayed by 3 sampling time.

Let y_t^o denote the fast sampled measurement with lower accuracy, while y_t^L is the measurement with higher accuracy but slower sampling rate; ω_t , v_t^o and v_t^L are independent and identically distributed Gaussian noises with covariance matrices Q , R^o and R^L respectively. The input sequence $\{u_1, \dots, u_t\}$ is known.

4.3 Bayesian Approach to Deal with Time Delay

4.3.1 Problem formulation

Consider the state space model below.

$$x_{t+1} = f(x_{t-1}, u_{t-1}, \Theta) + \omega_t \quad (4.2)$$

$$y_t = h(x_t) + v_t \quad (4.3)$$

To deal with the delayed lab data, an augmented state Z_t which includes both the previous and current states is constructed as follows.

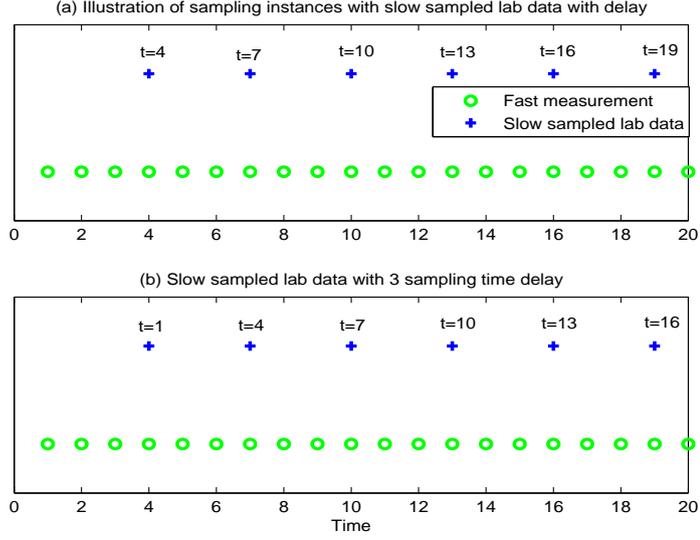


Figure 4.2: (a): Slow sampled lab data without time delay; (b): Slow sampled lab data with 3 sampling time delay.

$$Z_t = \begin{bmatrix} z_t^1 \\ z_t^2 \\ \vdots \\ z_t^{d+1} \end{bmatrix} = \begin{bmatrix} x_{t-d} \\ x_{t-d+1} \\ \vdots \\ x_t \end{bmatrix} \quad (4.4)$$

and

$$Z_{t+1} = \begin{bmatrix} z_{t+1}^1 \\ z_{t+1}^2 \\ \vdots \\ z_{t+1}^{d+1} \end{bmatrix} = \begin{bmatrix} x_{t-d+1} \\ x_{t-d+2} \\ \vdots \\ x_{t+1} \end{bmatrix} \quad (4.5)$$

$$= \begin{bmatrix} f(x_{t-d}, u_{t-d}) + \omega_{t-d} \\ f(x_{t-d+1}, u_{t-d+1}) + \omega_{t-d+1} \\ \vdots \\ f(x_t, u_t) + \omega_t \end{bmatrix} \quad (4.6)$$

$$= \begin{bmatrix} f(z_t^1, u_{t-d}) + \omega_{t-d} \\ f(z_t^2, u_{t-d+1}) + \omega_{t-d+1} \\ \vdots \\ f(z_t^{d+1}, u_t) + \omega_t \end{bmatrix} \quad (4.7)$$

$$= f(Z_t, U_t) + W_t \quad (4.8)$$

$$y_t^o = [0 \ 0 \ \cdots \ 1] Z_t + v_t^o \quad (4.9)$$

$$y_t^L = [1 \ 0 \ \cdots \ 0] Z_t + v_t^L \quad (4.10)$$

where

$$U_t = \begin{bmatrix} u_{t-d} \\ u_{t-d+1} \\ \cdots \\ u_t \end{bmatrix} \quad (4.11)$$

$$W_t = \begin{bmatrix} \omega_{t-d} \\ \omega_{t-d+1} \\ \cdots \\ \omega_t \end{bmatrix} \quad (4.12)$$

4.3.2 Formulation for online calibration

In reality, there is always model-plant mismatch. This model-plant mismatch may arise due to several reasons, for example, due to a misuse of model structure and insufficient training data [6]. In addition, the model built with the training data could provide poor prediction when applied to other data sets. Therefore, the model needs to be online calibrated which synthesizes the measurement information of both fast sampled online analyzer and slow rate laboratory analysis as discussed in the previous chapters. The calibration equations are organized as follows:

$$Z_t = \rho_t(f(Z_{t-1}, U_{t-1}, \Theta)) + b_t + \omega_t \quad (4.13)$$

$$\rho_t = \rho_{t-1} + \omega_t^\rho \quad (4.14)$$

$$b_t = b_{t-1} + \omega_t^b \quad (4.15)$$

$$b_t^o = b_{t-1}^o + \omega_t^o \quad (4.16)$$

$$y_t^o = Z_t + b_t^o + v_t^o \quad (4.17)$$

$$y_t^L = Z_t + v_t^L \quad (4.18)$$

where ρ_t is a scaling parameter that brings some flexibility to the model-plant mismatch compensation. A bias term b_t is introduced to correct the systematic error. It is observed that the online analyzer has deviation from the laboratory analysis, hence another bias term b_t^o is adopted to capture the difference. The calibration procedure is performed using recursive particle filtering. Z_t , ρ_t , b_t and b_t^o are states while the online analyzer reading y_t^o and the laboratory analysis y_t^L are two sources of output measurement information.

4.3.3 Calibration through particle filtering

Particle filter has been discussed in detail in previous chapters and is not repeated here. The algorithm is summarized below.

Given the current estimation of parameters, the particle filter algorithm is summarized as follows:

Step 1. Initialization. Draw initial N particles $\{x_0^i\}_{i=1}^N$ from the prior density $p(x_0|\Theta^k)$ and set each particle's weight to $\frac{1}{N}$. Set $t=1$.

Step 2. Importance sampling. Generate predicted particles $\{x_t^i\}_{i=1}^N$ from the importance density $p(x_t|x_{t-1}, \Theta^k)$.

Step 3. Assigning weights. Assign the weight to each particle using Equation 3.11 and 3.12.

Step 4. Resampling. Compute the number of effective particles using Equation (3.13). If N_{eff} is less than the threshold N_{thred} , then perform resampling and replace the predicted particles in Step 2 with resampled particles. Reset the weights of resampled particles uniformly as $\omega_t^i = \frac{1}{N}$. Otherwise, go to Step 5.

Step 5. Set $t = t + 1$ and repeat Step 2 to Step 4 for $t \leq T$.

The particle weight calculation when lab data is available is given below,

$$\omega_t^i = \exp\left(-\frac{(y_t^o - y^{pre})^2}{2R_{o1}} - \frac{(y_t^l - y^{pre})^2}{2R_{l1}}\right) \quad (4.19)$$

When lab data is not available, the weight is calculated as

$$\omega_t^i = \exp\left(-\frac{(y_t^o - y^{pre})^2}{2R_{o2}} - \frac{(y_t^l - y^{pre})^2}{2R_{l2}(t)}\right) \quad (4.20)$$

where R_{o1} is the noise variance for online analyzer when lab data is available; R_{o2} is the noise variance for online analyzer when lab data is not available; R_{l1} is the noise variance for lab data when it is available. When lab data is not available, the previous lab value is used and $R_{l2}(t)$ is the noise variance which is a sigmoidnet function and increases with time.

4.4 A Numerical Simulation Example

The proposed Bayesian calibration method for time-delayed measurement is illustrated through a numerical simulation example. It is described by the following equation:

$$x_{t+1} = ax_t + bu_t + \omega_t \quad (4.21)$$

$$y_t^o = x_t + v_t^o \quad (4.22)$$

$$y_t^L = x_t + v_t^L \quad (4.23)$$

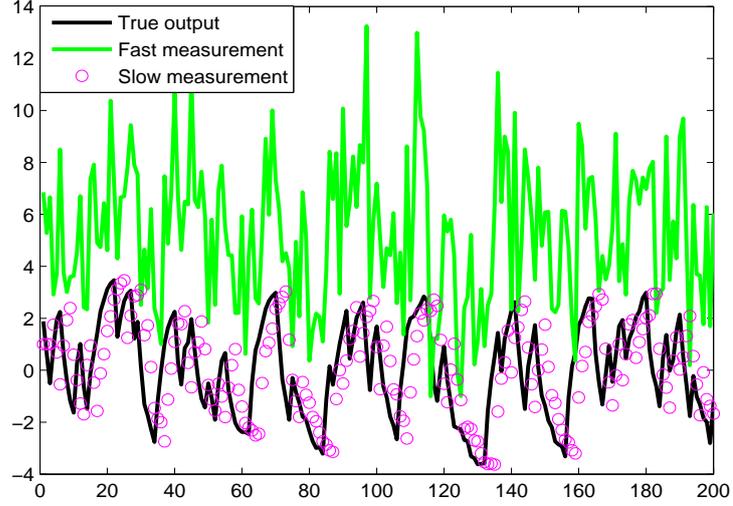


Figure 4.3: Fast process measurement and slow measurement with time delay.

where the system parameter $a = 0.7$ and $b = 1.0$. y_t^o is the fast measurement which is available every sampling time, however is not accurate; y_t^L is the slow sampled measurement with higher accuracy while it can only be obtained every 4 sampling times, and it has a time delay as illustrated by Figure 4.3.

First the system parameters are estimated using the method in Chapter 2 and Chapter 3 based on the collection of fast sampled measurements. The model prediction results based on the estimated parameters are shown in Figure 4.4.

In order to achieve a fast estimation of the states (every sampling time) with higher accuracy, the proposed Bayesian calibration method is applied here. An augmented state Z_t is formulated as

$$Z_t = \begin{bmatrix} z_t^1 \\ z_t^2 \\ z_t^3 \\ z_t^4 \\ z_t^5 \end{bmatrix} = \begin{bmatrix} x_{t-4} \\ x_{t-3} \\ x_{t-2} \\ x_{t-1} \\ x_t \end{bmatrix} \quad (4.24)$$

and

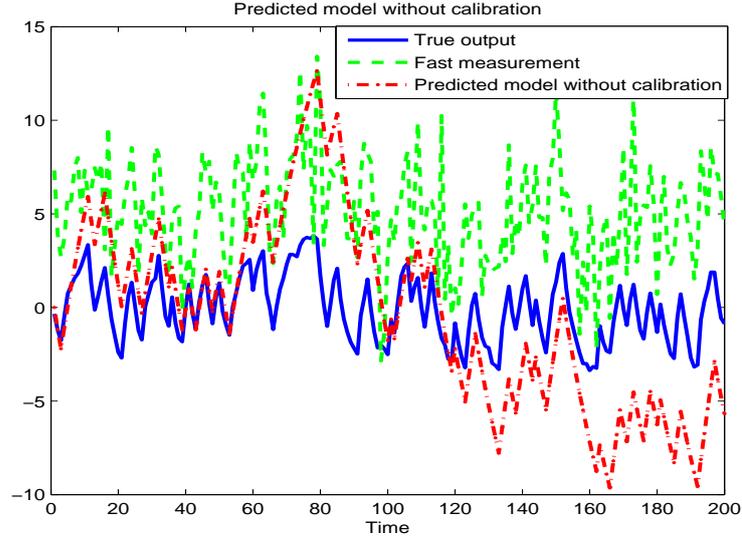


Figure 4.4: Model prediction without Bayesian calibration

$$Z_{t+1} = \begin{bmatrix} z_{t+1}^1 \\ z_{t+1}^2 \\ z_{t+1}^3 \\ z_{t+1}^4 \\ z_{t+1}^5 \end{bmatrix} = \begin{bmatrix} x_{t-3} \\ x_{t-2} \\ x_{t-1} \\ x_t \\ x_{t+1} \end{bmatrix} \quad (4.25)$$

$$= \begin{bmatrix} ax_{t-4} + u_{t-4} + \omega_{t-4} \\ ax_{t-3} + u_{t-3} + \omega_{t-3} \\ ax_{t-2} + u_{t-2} + \omega_{t-2} \\ ax_{t-1} + u_{t-1} + \omega_{t-1} \\ ax_t + u_t + \omega_t \end{bmatrix} \quad (4.26)$$

$$= \begin{bmatrix} az_t^1 + bu_{t-4} + \omega_{t-4} \\ az_t^2 + bu_{t-3} + \omega_{t-3} \\ az_t^3 + bu_{t-2} + \omega_{t-2} \\ az_t^4 + bu_{t-1} + \omega_{t-1} \\ az_t^5 + bu_t + \omega_t \end{bmatrix} \quad (4.27)$$

$$= AZ_t + BU_t + W_t \quad (4.28)$$

$$y_t^o = [0 \ 0 \ 0 \ 0 \ 1] Z_t + v_t^o \quad (4.29)$$

$$y_t^L = [1 \ 0 \ 0 \ 0 \ 0] Z_t + v_t^L \quad (4.30)$$

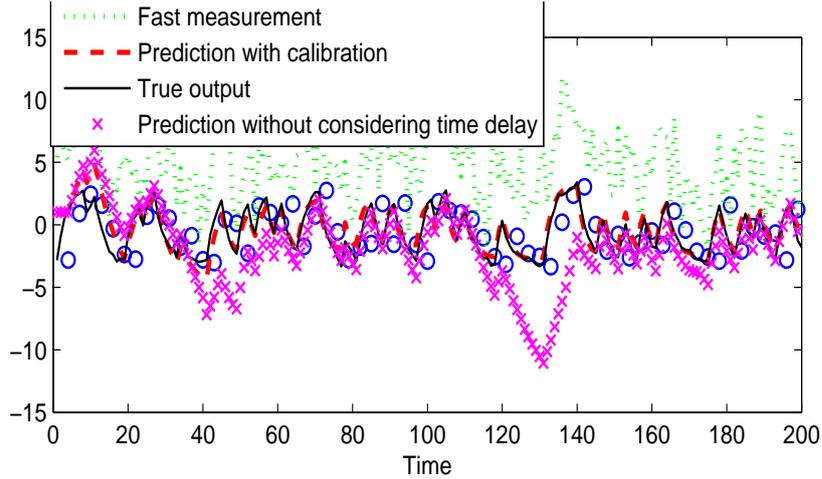


Figure 4.5: Model prediction with Bayesian calibration.

The prediction results with Bayesian calibration are shown in Figure 4.5 and Figure 4.6. It can be seen that the model prediction after Bayesian calibration gives real time estimation of the states while provides better result compared with the existing fast measurement. It also can be seen that the Bayesian calibration method developed in last chapter does not give satisfactory result when there exists a time delay in the measurement.

4.5 Semi-continuous Fermentation Example

A semi-continuous fermentation of baker's yeast is adopted here for illustration purpose. This example has also been used in [16] and last chapter. The two inputs two outputs fermentation system is described by the following equations:

$$\frac{dx_1}{dt} = \left(\frac{\theta_1 x_2}{\theta_2 + x_2} - u_1 - \theta_4 \right) x_1 \quad (4.31)$$

$$\frac{dx_2}{dt} = -\frac{\theta_3 x_1 x_2}{\theta_2 + x_2} + u_1 (u_2 - x_2) \quad (4.32)$$

where the output x_1 is the biomass concentration (g/l); output x_2 is the substrate concentration (g/l); input u_1 is the dilution factor h^{-1} ; and the input u_2 is the substrate concentration in the feed (g/l). Both inputs are with a sampling rate of 1 minute. $\Delta t = 1$ minute is chosen as the discretizing sample time. $\theta_1, \theta_2, \theta_3$ and θ_4 are the system parameters with true values of $\theta_1 = 0.31, \theta_2 = 0.18, \theta_3 = 0.56$, and $\theta_4 = 0.05$. The process is simulated for 500 minutes and the two inputs are shown in Figure 4.7.

A disturbance d_k is added to the state x_2 in Equation 4.32 such that

$$d_k = 0.5 \cdot \cos\left(\frac{k}{10 \cdot \pi}\right) + 0.2 \cdot n_k \quad (4.33)$$

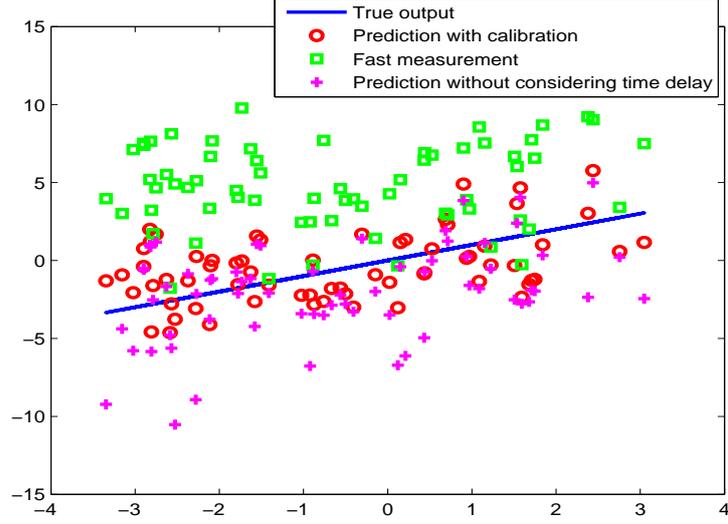


Figure 4.6: Fast process measurement and slow measurement with time delay.

where n_k is a non-Gaussian noise which is generated from a bimodal distribution such that with 70% of the time it is generated from a Gaussian distribution with a mean value of -0.2 and variance of 0.1^2 , and with 30% of the time it is generated from a Gaussian distribution with a mean value of 0.2 and variance of 0.1^2 . Suppose we are interested in the substrate concentration x_2 whose value is observed by two measurements. The first approach has a fast sampling rate of 1 minute, but is corrupted by a large measurement noise, hence is less accurate. The second measurement method is more accurate with smaller noise, however, it is only available every 5 minute and is each lab data is 5 sampling time delayed. Assuming that two parameters θ_1 and θ_3 are unknown. The parameter estimation algorithm proposed in Chapter 2 is applied to this data set with the first less accurate measurement as the output. The initial guess for the two parameters are $\theta_1 = 0.1$, $\theta_3 = 0.1$. 100 particles are used for the particle filtering approximation. The estimation terminated after near 30 EM iterations. The model prediction result with the estimated parameters is shown in Figure 4.8.

It can be seen from Figure 4.8 that, the model prediction with the estimated parameters has an obvious mismatch compared to the true output.

In order to compensate the difference between the predicted model and the real process output, the proposed Bayesian calibration approach is applied through Equation (3.31)-(3.36), where x_t^{pre} is the predicted value for the substrate concentration x_2 from the estimated model; ω_t is the process noise which follows a Gaussian distribution $\omega_t \sim N(0, 0.15^2)$; v_t^o and v_t^L are measurement noises for the fast sampled measurement and slow sampled measurement which are chosen as $N(0, 0.7^2)$ and $N(0, 0.01^2)$. The calibration result is shown in

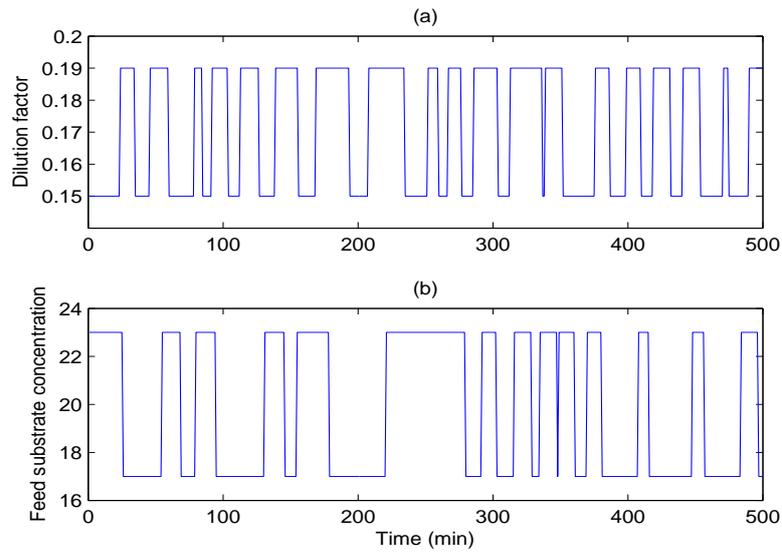


Figure 4.7: Process inputs. (a): u_1 , the dilution factor h^{-1} ; (b): u_2 , the substrate concentration in the feed.

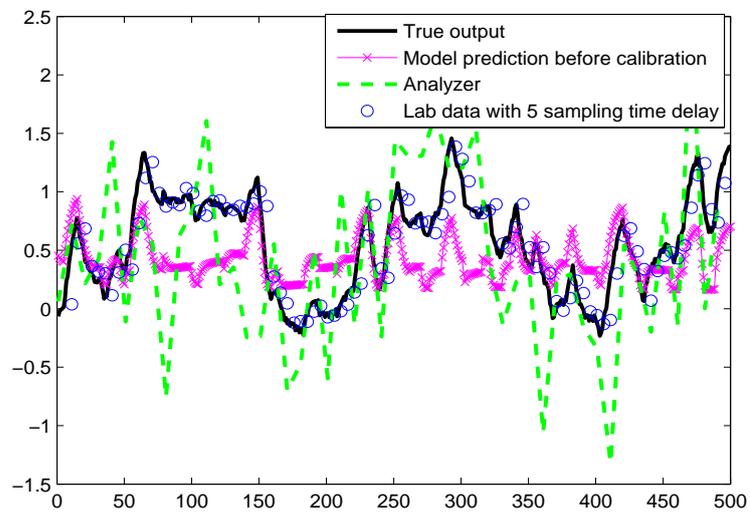


Figure 4.8: Comparison of current measurements.

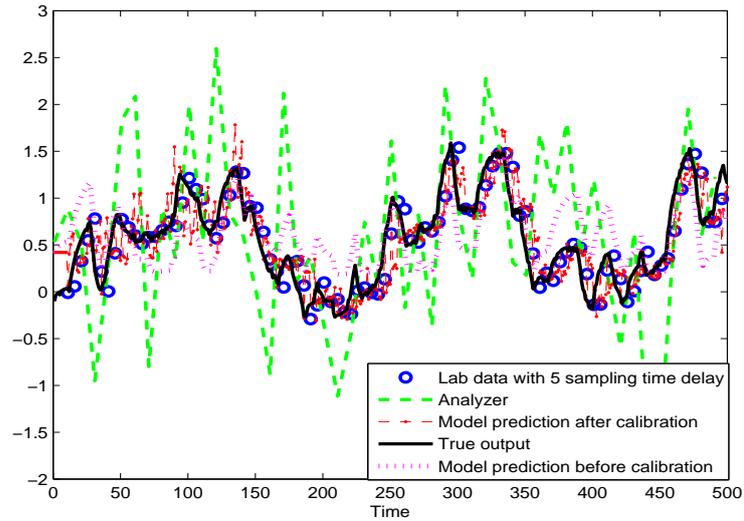


Figure 4.9: Comparison of different measurements.

Figure 4.9 and Figure 4.10, from which we can see that the estimation with the Bayesian calibration provides a better result than the model prediction without calibration.

4.6 Conclusion

The time delayed issue associated with the soft sensor development has been addressed in this chapter. An augmented state which consists of the current state and past states has been constructed to deal with time delay problem. In order to estimate the augmented state, the Bayesian calibration approach is applied which synthesizes both fast sampled measurement and slow sampled measurement with time delay. The proposed method is demonstrated through a numerical simulation example and a semi-fermentor simulation example.

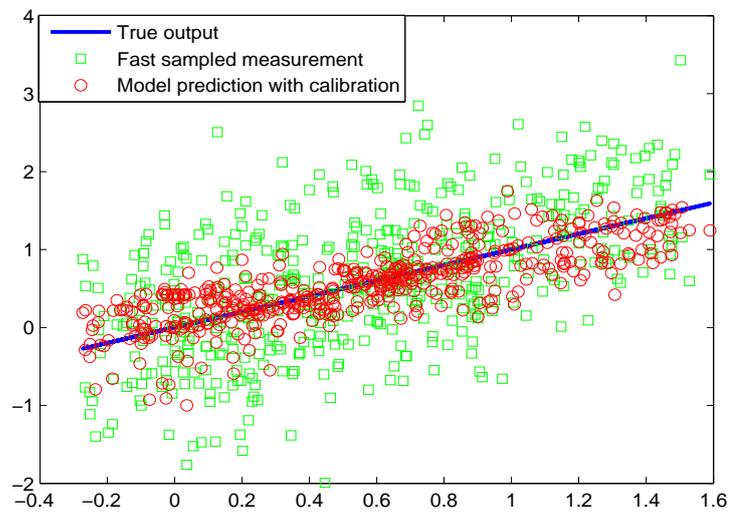


Figure 4.10: 45 degree comparison.

Bibliography

- [1] S. D. Grantham, L. H. Ungar, A First Principles Approach to Automation Troubleshooting of Chemical Plants. *Comput. Chem. Eng*, 1990, vol 14, pp 783-798.
- [2] M. Choi, J. Choi, J. Park and W. Chung, State Estimation with Delayed Measurement Considering Uncertainty of Time Delay, *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp 3987-3992, 2009.
- [3] Y. Bar-Shalom, Update with Out-of-Sequence Measurements in Tracking: Exact Solution, *IEEE Tracsaction on Aerospace and Electronic Systems*, vol. 38, No. 3, pp. 769-778, 2002.
- [4] Y. Bar-Shalom, H. Chen and M. Malick, One-Step Solution for the Multistep Out-of-Sequence-Measurement Problem in Tracking, *IEEE Tracsaction on Aerospace and Electronic Systems*, vol. 40, No. 1, pp. 27-37, 2004.
- [5] S. Challa, R. J. Evans and X. Wang, A Baysian solution and its approximations to out-of-sequence measurement problems, *Information Fusion*, vol. 4, pp. 185-199, 2003.
- [6] X. Shao, B. Huang, J. M. Lee, F. Xu and A. Espejo, Bayesian Method for Multirate Data Synthesis and Model Calibration, *American Institute of Chemical Engineers*, vol. 57, pp. 1514-1525, 2011.

Chapter 5

Conclusions

5.1 Summary of this thesis

This thesis is concerned with the modeling and model calibration issues for soft sensor development. Single-model based modeling techniques have shown limitation in describing processes with multiple operating conditions. Unavailability of certain critical process variables renders difficulties in process monitoring and control. The limitation of conventional hardware sensors and the necessity of measuring important process variables motivate us to explore soft sensor development.

The background material about soft sensor was presented in Chapter 1. Chapter 2 deals with the identification problem of nonlinear parameter varying systems. The local model structure is assumed to be known while the model parameters vary among different operating periods. By employing an exponential weighting function which provides the weight for each local model, a global nonlinear model can be obtained which is a weighted interpolation of each local nonlinear model. The estimation of each local model parameter is performed under the framework of EM algorithm. In the Expectation step of the EM algorithm, particle filter approximation is adopted to calculate the Q function. Meanwhile, the missing output problem is considered and is solved by particle filters approximation. Simulated numerical examples as well as experiment performed on pilot-scale setup are used to demonstrate the capabilities of the proposed identification approach.

In practice, model-plant mismatch always exists, especially for data-driven models which are built based on historical data. Changes of operating conditions could render large mismatch compared with the true value since the available training data only describes a period of process historical behavior. The objective of Chapter 3 is to develop a model updating mechanism by combining different measurement sources. A Bayesian method based model updating strategy is formulated which make uses of both fast-rate sampled and slow-rate sampled measurements. With the given model, the Bayesian

online calibration is performed using particle filter. To illustrate the proposed method, an oil sands soft sensor is studied in Chapter 3. A data-driven model based soft sensor is developed and implemented with the proposed model updating strategy. The online monitoring results prove the superiority of this soft sensor to the existing measurements.

Time delay issue is another challenge in real process. Chapter 4 moves the discussion forward to the soft sensor development of time delayed processes. In Two types of measurements are considered. One is fast sampled with low accuracy; the other measurement is more accurate, however it is slow sampled and couple with time delay. To deal with this problem, an augmented state is constructed which consists of both current state and past states. Such augmented state is estimated using particle filters. Both numerical simulation and a semi-fermenter example are used to confirm the efficiency of the proposed method.

5.2 Recommendations for future work

Throughout the whole thesis, we have been working on the modeling and model calibration issues that are associated with soft sensor development. Several techniques have been adopted such as Expectation-Maximization algorithm, particle filtering, Bayesian online calibration. With the efficiency of those methods being demonstrated in this thesis, there are several open issues and directions which can further extend the application of those techniques.

In the first chapter, it is assumed that the trajectory of the scheduling variable is known. The possibility exists that some data are collected without knowing the identity (from which operating condition) and it would be more practical to formulate a framework which also take part of the trajectory of the scheduling variable as missing.

Further, in the forth chapter which deals with the time delay issue, only small time delay (4 sampling time delay and 5 sampling time dealy) is discussed. It is worth extending the work to longer time delay which is commonly encountered in chemical industries so that the proposed method can be applied to real processes.