DATA-DRIVEN MODELING FOR QUALITY CONTROL IN CHEMICAL PROCESSES

DATA-DRIVEN MODELING FOR QUALITY CONTROL IN CHEMICAL PROCESSES

BY

KUILIN CHEN, B.Eng.

A THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

© Copyright by Kuilin Chen, June 2014

All Rights Reserved

Master of Applied Science (2014) McMaster University

(Chemical Engineering) Hamilton, Ontario, Canada

TITLE:	DATA-DRIVEN MODELING FOR QUALITY CONTROL IN CHEMICAL PROCESSES
AUTHOR:	Kuilin Chen B.Eng., (Automation) East China University of Science and Technology, Shanghai, China
SUPERVISOR:	Dr. Prashant Mhaskar, Dr. Jie Yu
NUMBER OF PAGES:	xviii, 131

Abstract

This thesis considers the problem in data-driven modeling for quality control of complex chemical processes characterized by nonlinearity and time-varying dynamics over a wide range of operating conditions. The critical steps in data-driven modeling for quality control, such as data pretreatment, online inferential model maintenance and quality control mechanism designs, are investigated separately in previous studies, which calls for holistic research on those topics.

In the initial phase of this research, the missing measurement problem in data-driven modeling is addressed by two developed probabilistic partial least squares (PPLS) methods. In the expectation-maximization (EM) based PPLS (EM-PPLS), the convergence properties for model parameters and estimated values of missing measurements are proved in explicit forms, enabling effective and robust application. In parallel, full Bayesian treatment is employed in the Bayesian inference based PPLS (BI-PPLS), to substantially reduce the computational complexity in latent variable selection in the PPLS model. In both proposed PPLS methods, data-driven model generation and missing measurement estimation can be carried out concurrently along with the uncertainty handling in process measurements through a probabilistic strategy.

Next, the degradation problem in the online implementation of the data-driven inferential models is addressed. The discrepancy between the actual model parameters in the data-driven model and filtered model parameters by Kalman filter is calculated

and utilized to derive a model mismatch index, decomposition of which can further lead to identification of leading abnormal model parameters in the data-driven model. With the developed model mismatch index, an online model update strategy is developed to maintain the data-driven inferential model online.

Finally, a predictive controller based on multiple data-driven dynamic and inferential models is designed to control the final product quality in the batch processes. The entire batch is divided into several modeling phases based on the availability of intermittent quality measurements, in which local dynamic models are identified using weighted autoregressive exogenous (ARX) models. Meanwhile, multiway PLS (MPLS) models are also constructed to predict the quality at batch termination given the intermittent quality measurements at the initiation of each modeling phase. The data-driven models are integrated within model predictive control (MPC) framework to control the final product quality in Penicillin fermentation process with the consideration of causality relationships between the future inputs and final product quality.

Acknowledgements

I wish to express my gratitude to my supervisor, Dr. Prashant Mhaskar, for his continued support, advice and guidance throughout this research. It is his suggestions and encouragement that make my experience as a graduate student a positive one. This thesis would certainly not be possible without his vision and direction. I would also like to thank Dr. Jie Yu to give me the opportunity to study at McMaster University and supervise me in my first year study in McMaster.

I wish to acknowledge the financial support from McMaster Advanced Control Consortium (MACC) and the Department of Chemical Engineering at McMaster University. I would like to thank all the professors that have taught me at McMaster University, from whom I have learned a lot of useful knowledge for my research and future work.

Thanks also go to the industrial partner, the Dow Chemical Company, for their support and continued participation in the research project. Special thanks go to Dr. Ivan Castillo and Dr. Leo Chiang for their insights and suggestions.

In addition, I would like to thank fellow students in MACC who have offered their suggestions and assistance. I would also like to appreciate all my friends in McMaster University, whose friendship and moral support are indeed helpful to help me go through the hard time in Canada.

This thesis is dedicated to my parents for their unconditionally financial and emotional support all these years, especially during the difficult periods of my life.

To my dear parents

Notations and abbreviations

Notations

$\mathcal{N}(\cdot)$	Gaussian distribution
$Tr(\cdot)$	Trace operation
$\mathbb{E}[\cdot]$	Expectation
$diag\{\cdot\}$	Diagonal matrix
$\Gamma(\cdot)$	Gamma distribution
$\mathbb{R}^{m\times n}$	m by n Euclidian space
C_i	Contribution of the i-th regression coefficient for model mismatch index
$\mathbf{F}_k$	State transition matrix at the k-th sampling instance
$\mathbf{H}_k$	Observation matrix at the k-th sampling instance
I_k	Model mismatch index
$\mathbf{K}$	Kalman gain
n_x	Lag of process measurements
n_u	Lag of control inputs
$\mathbf{P}$	Loading for regressor matrix in PCA or PLS
$\mathbf{P}^o$	Submatrix of $\mathbf{P}$ corresponding to $\mathbf{x}^o$
$\mathbf{P}^m$	Submatrix of $\mathbf{P}$ corresponding to $\mathbf{x}^m$
$\mathbf{p}_a$	Vector of the a-th column in $\mathbf{P}$

$\mathbf{Q}$	Loading for response matrix in PLS
$\mathbf{Q}^o$	Submatrix of $\mathbf{Q}$ corresponding to $\mathbf{y}^o$
$\mathbf{Q}^m$	Submatrix of $\mathbf{Q}$ corresponding to $\mathbf{y}^m$
$\mathbf{Q}_k$	Covariance of state noise
$\mathbf{q}_a$	Vector of the a-th column in $\mathbf{Q}$
$\mathbf{R}$	Weight in PLS for original regressor matrix
$\mathbf{R}_k$	Covariance of measurement noise
$\bar{\mathbf{R}}$	Rotation matrix for regressor variables
$\bar{\mathbf{R}}^*$	Rotation matrix for response variables
R^2	Coefficient of determination
$\mathbf{r}_a$	Vector of the a-th column in $\mathbf{R}$
$\mathbf{s}$	Vector of process states
$\mathbf{T}$	Latent score matrix
$\mathbf{t}$	Arbitrary sample in $\mathbf{T}$
$\mathbf{t}_{new}$	Score for $\mathbf{x}_{new}$
$\mathbf{u}$	Vector of control action
$\mathbf{v}_k$	Gaussian measurement noise
$\mathbf{W}$	Weight in PLS for deflated regressor matrix
$\mathbf{w}_a$	Vector of the a-th column in $\mathbf{W}$
$\mathbf{w}_k$	Gaussian state noise
w_g^ℓ	Weight of the ℓ-th model in the g-the modeling phase
$\mathbf{X}$	Matrix of process measurements
$\mathbf{X}_{new}$	Matrix of new process measurements
$\mathbf{X}_{old}$	Matrix of old process measurements
$\mathbf{X}_u$	Combined matrix of old and new process measurements

$\tilde{\mathbf{X}}$	Multiway matrix of process measurements
$\tilde{\mathbf{X}}_g$	Multiway matrix of process measurements in the g-th modeling phase
$\mathbf{X}_g$	Matrix of process measurement in the g-th modeling phase
$\mathbf{x}$	Vector of process measurements
$\mathbf{x}^o$	Observed part of $\mathbf{x}$
$\mathbf{x}^m$	Missing part of $\mathbf{x}$
$\hat{\mathbf{x}}$	Predicted process measurements
$\mathbf{x}_{new}$	A sample of $\mathbf{X}_{new}$
$\tilde{\mathbf{x}}_{new}$	Residual of $\mathbf{x}_{new}$
$\bar{\mathbf{x}}$	Vector of lagged process measurements and inputs
$\mathbf{x}_{\text{past}}$	Vector of past batch process measurements
$\mathbf{x}_{\text{future}}$	Vector of future batch process measurements
$\mathbf{Y}$	Matrix of quality measurements
$\hat{\mathbf{Y}}$	Matrix of predicted quality variables
$\mathbf{Y}_{new}$	Matrix of new quality measurements
$\mathbf{Y}_{old}$	Matrix of old quality measurements
$\mathbf{Y}_u$	Combined matrix of old and new quality measurements
$\mathbf{Y}_g$	Matrix of quality measurements at the end of the g-th modeling phase
$\mathbf{Y}_g^0$	Matrix of quality measurements at the beginning of the g-th modeling phase
$\mathbf{y}$	Vector of quality measurements
$\hat{\mathbf{y}}$	Estimated quality measurements
$\mathbf{y}^o$	Observed part of $\mathbf{y}$
$\mathbf{y}^m$	Missing part of $\mathbf{y}$
$\mathbf{Z}$	Stacked matrix of $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{Z}_0$	Matrix of initial conditions in batch processes

$\boldsymbol{\alpha}$	Hyperparameter for precision of loading matrix
β_{PLS}	Regression coefficients in the PLS model
β_{RPLS}	Regression coefficients in the RPLS model
β_k	Regression coefficients at k-th sampling instance
$\hat{\beta}_k$	Prior estimation of regression coefficients in Kalman filter
β_k^{KF}	Posterior estimation of regression coefficients in Kalman filter
$\tilde{\beta}_k$	Residual of regression coefficients
β_u	Updated regression coefficients
μ_x	Mean vector for $\mathbf{X}$
μ_y	Mean vector for $\mathbf{Y}$
$\boldsymbol{\Lambda}$	Stacked matrix for loadings
Λ_T	Covariance of scores
$\Lambda_{\tilde{\beta}}$	Covariance of regression coefficient residuals
$\mathcal{L}$	Log-likelihood function
σ_x	Standard deviation for $\mathbf{x}$
σ_y	Standard deviation for $\mathbf{y}$
λ	Forgetting factor in RPLS
ξ_i	Direction of the i-th regression coefficient
$\boldsymbol{\Psi}$	Weight matrix for quality in MPC
$\boldsymbol{\Phi}$	Weight matrix for control action in MPC

Abbreviations

ANN	Artificial neural networks
ARX	Auto-regressive exogenous
BI-PPLS	Bayesian inference based probabilistic partial least squares

EKF	Extended Kalman filter
EM	Expectation-maximization
EM-PPLS	Expectation-maximization based probabilistic partial least squares
GAMS	General algebraic modeling system
JITL	Just-in-time learning
MAPE	Mean absolute percentage error
MCC	Mid-course correction
MLE	Maximum likelihood estimation
MPLS	Multiway partial least squares
MPC	Model predictive control
OLS	Ordinary least sqaures
PCA	Principal component analysis
PCR	Principal component regression
PPCA	Probabilistic principal component analysis
PLS	Partial least squares
PPLS	Probabilistic partial least squares
SPC	Statistical process control
SPE	Squared prediction error
SVM	Support vector machine
PID	Proportional-integral-derivative
PRBS	Pseudo-random binary signal
RPLS	Recursive partial least squares
RMSE	Root mean squared error

Contents

Abstract **iii**

Acknowledgements **v**

Notations and abbreviations **viii**

1 Introduction **1**

1.1 Motivation and goals 1

1.2 Main contributions 3

1.3 Thesis overview 4

2 Probabilistic partial least squares methods for missing measurement estimation **6**

2.1 Introduction 6

2.2 General framework of probabilistic partial least squares 10

2.3 Expectation-maximization algorithm for probabilistic partial least squares model and its convergence properties 14

2.3.1 Expectation-maximization based probabilistic partial least squares with missing measurements 14

2.3.2 Convergence of expectation-maximization algorithm for probabilistic partial least squares model . . . 19
2.4 Bayesian inference based probabilistic partial least squares with missing measurements . . . 23
2.5 Case study . . . 31
2.5.1 A numerical example . . . 31
2.5.2 Performance comparison . . . 33
2.6 Conclusion . . . 41

3 Data-driven inferential model update in industrial processes 42
3.1 Introduction . . . 42
3.2 Preliminaries . . . 46
3.2.1 Partial least squares . . . 46
3.2.2 Recursive partial least squares . . . 48
3.2.3 Quality relevant process monitoring . . . 50
3.3 Online update of the soft sensor model . . . 50
3.3.1 Detection of mismatch in the soft sensor model by Kalman filter 50
3.3.2 Diagnosis of soft sensor degradation . . . 55
3.3.3 Online model update . . . 56
3.4 Case studies . . . 60
3.4.1 A simulated example . . . 60
3.4.2 An industrial process . . . 69
3.5 Conclusions . . . 79

4 Multiple models based quality control for batch processes 81
4.1 Introduction . . . 81
4.2 Preliminaries . . . 86

4.2.1 Penicillin fermentation process 86
4.2.2 Single Model based Quality Control 89
4.3 Improved Quality Control of Batch Processes 90
4.3.1 Multiple Quality Inferential Models 90
4.3.2 Localized Multiple Discrete Time Dynamic Models 93
4.3.3 Model Predictive Control 96
4.4 Multiple Models based Quality Control of Penicillin Production 100
4.4.1 Inferential and Process Models 101
4.4.2 Closed-loop Simulation Results 108
4.5 Conclusions 113

5 Conclusions and future work **115**

5.1 Conclusions 115
5.2 Future work 118

List of Tables

2.1 Performance of missing measurement estimation by PCA-ALS method 32

2.2 Performance of missing measurement estimation by EM-PPLS method 33

2.3 Performance of missing measurement estimation by BI-PPLS method . 33

2.4 Performance of soft sensor prediction by EM-PPLS method 34

2.5 Performance of soft sensor prediction by BI-PPLS method 34

2.6 CPU time (second) of EM-PPLS and BI-PPLS methods 34

3.1 Comparison of computational results of the simulated example in the online test 63

3.2 Comparison of computational results between RPLS and the proposed method in the online test for the industrial process 72

4.1 Variables in the Penicillin fermentation process 88

4.2 Averages of final product qualities by different control designs 111

List of Figures

2.1 Illustrative diagram of PPLS based missing measurement estimation . . 13

2.2 Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by PCA-ALS method in the case of 10% missing measurements 35

2.3 Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by EM-PPLS method in the case of 10% missing measurements 36

2.4 Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by BI-PPLS method in the case of 10% missing measurements 37

2.5 Time-series plots of the actual output measurements (blue lines) and predicted values (red dots) by EM-PPLS method in the case of 10% missing measurements 39

2.6 Time-series plots of the actual output measurements (blue lines) and predicted values (red dots) by BI-PPLS method in the case of 10% missing measurements 40

3.1 Illustrative diagram of detecting severe model mismatch 58

3.2 Flowchart diagram of the proposed model update method 59

3.3 Online test of the simulated example 64

3.4 Comparison of soft sensor prediction performance for the simulated example . . . 65
3.5 Contribution plot of the model mismatch index for the 214th sampling instance in the simulated example . . . 67
3.6 Contribution plot of the model mismatch index for the 436th sampling instance in the simulated example . . . 68
3.7 Flow sheet of the industrial process . . . 71
3.8 Online test for the industrial process . . . 74
3.9 Comparison of soft sensor prediction performance for the industrial example . . . 75
3.10 Contribution plot of the model mismatch index for the 208th sampling instance in the industrial example . . . 76
3.11 Contribution plot of the model mismatch index for the 538th sampling instance in the industrial example . . . 77
3.12 Comparison of regression coefficients between the proposed approach and RPLS method . . . 79
4.1 Illustrative diagrams for data unfolding and arrangement . . . 91
4.2 Prediction of quality variables of the 1st modeling phase . . . 102
4.3 Prediction of quality variables of the 2nd modeling phase . . . 103
4.4 Prediction of quality variables of the 3rd modeling phase . . . 104
4.5 Prediction of future trajectories in the 1st modeling phase . . . 106
4.6 Prediction of future trajectories in the 2nd modeling phase . . . 107
4.7 Prediction of quality variables in the 3rd modeling phase . . . 108
4.8 Trajectory of control inputs in different control designs . . . 110
4.9 Trajectory of process measurements in different control designs . . . 111
4.10 Comparison of the final product qualities obtained by three control designs 113

Chapter 1

Introduction

1.1 Motivation and goals

Data-driven modeling for quality prediction and control plays an important role in process control, monitoring, optimization towards process safety, stability and product quality, in that accurate first-principle models are not available under some circumstances. Application of data-driven models in process industries usually includes the steps of data collection and pretreatment, model generation and validation, as well as online implementation for quality prediction and control. In the step of data collection and pretreatment, the presence of missing measurements poses a significant challenge on model identification and generation. Traditionally, missing measurements are either ignored or extrapolated without considering cross-variable correlation in data-driven modeling. Those improper ways of handling missing data result in biased data-driven models, which cannot predict product quality accurately. Therefore, it is increasingly important and necessary to develop robust and effective missing data estimation methods by taking the uncertainty in process measurements into account.

After data-driven models are generated, they will be implemented online to predict the difficult-to-measure quality variables in the real-time fashion, known as soft sensors. One of the challenges for a reliable application of data-driven models is that their prediction accuracy deteriorates when processes move to different conditions with new process dynamics. Though some dynamic and recursive modeling approaches are developed for partial least squares (PLS), they are not desirable in industry because the necessity for model update is not considered. To this end, indexes that can monitor the prediction performance of soft sensor models and give guidance about when to update soft sensor models are highly desirable. In parallel, the soft sensor model should be updated online automatically once significant soft sensor model degradation is detected.

Besides continuous processes, batch processes are widely used in polymer, pharmaceutical, bio-chemical and other industries to produce high-value-added products due to their flexibility to manage many different grades and types of products. However, traditional control strategy designed for continuous processes cannot be applied to batch processes directly, due to the special nature of batch processes, such as finite process duration, the absence of equilibrium point as the target point, and nonlinear and time-varying dynamics over a wide range of operating conditions. In addition, control of product quality is even more challenging in batch processes because batch operations suffer from constant changes in raw materials, variations in initial conditions, and disturbances in operating conditions. Moreover, direct compensation for those disturbances on product quality is impractical because the product quality measurements are unavailable during the batch operation and usually taken through off-line laboratory analysis after the completion of the entire batch. As such, effective quality control mechanisms are highly desirable for batch processes to minimize the disturbances on the product quality.

1.2 Main contributions

The first contribution of this thesis is the development of two kinds of probabilistic partial least squares (PPLS) frameworks including expectation-maximization (EM) based PPLS (EM-PPLS) and Bayesian inference based PPLS (BI-PPLS) methods in dealing with stochastic missing measurements in data-driven modeling. Other studies on missing data estimation either estimate the missing data from existing model structures or ignore the multivariate correlation between process and quality variables. The missing measurements in historical data for soft sensor model generation are considered in a probabilistic manner in order to take the process uncertainty into account. Convergence properties of EM-PPLS method are proved in explicit forms so that the robustness of the EM-PPLS method in industrial application is guaranteed. In addition, the full Bayesian framework in BI-PPLS leads to significantly improved computational efficiency by avoiding cross-validation in latent variable selection.

Second, a model mismatch index is proposed to monitor the PLS soft sensor degradation using Kalman filtering. The proposed model mismatch index not only detects the deterioration in data-driven inferential models but also provides the opportunity to identify the leading abnormal regression coefficients that cause the degraded prediction performance via contribution plots. In parallel, a corresponding online model update mechanism is developed to update the data-driven model under necessary scenarios using both historical and new data. The proposed soft sensor model update approach can be applied to industrial processes to maintain the prediction accuracy of soft sensor models online, while saving the effort for model maintenance and keeping the model structure stable.

Last but not the least, a novel multiple models based quality control for the batch

process is developed to control final product quality in batch processes where intermittent quality measurements are available. To this end, multiple dynamic models are developed to describe the process dynamics in each modeling phase, while multiple inferential models are developed to predict the product quality at the end of batch. The dynamic process model and inferential quality model are conjugated to provide a prediction of final product quality at every sampling instance from batch initiation to batch termination. Based on the data-driven models, a model predictive control (MPC) strategy is designed to drive the final product quality towards desirable specifications.

1.3 Thesis overview

The remainder of this thesis is organized as follows.

In Chapter 2, EM-PPLS and BI-PPLS approaches for handling stochastic missing measurements and enabling soft sensor estimations are developed and compared. In the proposed PPLS methods, the expectation-maximization or variational Bayesian inference strategies are designed to recursively update PPLS model parameters and latent scores, which are further employed to estimate missing measurements through iterative learning. Moreover, the convergence properties of EM based PPLS method are investigated and proved in explicit forms. The above two types of PPLS approaches are applied to soft sensor estimation in a simulated example with different test cases under various levels of stochastic missing measurements. It is demonstrated that both EM-PPLS and BI-PPLS methods can robustly estimate missing measurements and further predict quality variables with high accuracy and fidelity. Furthermore, the comparison between the two presented approaches indicates that BI-PPLS outperforms EM-PPLS in terms of more desirable computational efficiency.

In Chapter 3, a Kalman filter based model mismatch index is developed to monitor

the prediction performance of PLS inferential models with the auxiliary of traditional process monitoring indexes, T^2 and SPE. In addition, a contribution plot for the model mismatch index is developed to locate leading abnormal regression coefficients. Then, the data-driven inferential model can be updated through PLS regression by using samples from the offline training set and new process conditions. The proposed online update method is applied to both simulated and industrial examples and the effectiveness of the proposed approach is demonstrated by comparing with traditional recursive partial least squares (RPLS).

In Chapter 4, the problem of quality control of a Penicillin production process is considered. A novel multiple models based quality control for the Penicillin fermentation process is developed to control the final product quality in Penicillin production where intermittent quality measurements are available. To this end, multiple PLS based inferential models are built to predict final product quality by using measurable variable trajectories and intermittent quality measurements. Further, localized multiple linear discrete time models are constructed to characterize the process dynamics in order to forecast the unavailable future trajectories of measurable variables given past measurements and control inputs. The developed multiple inferential and dynamic models are integrated with MPC framework so that causality and nonlinear relationships between the future inputs and outputs are accounted for in predicting the final quality and computing the manipulated input trajectory. Simulation results demonstrate the efficacy of the proposed approach.

Finally, Chapter 5 summaries the main contributions of the thesis and suggests the related future work.

Chapter 2

Probabilistic partial least squares methods for missing measurement estimation[1]

2.1 Introduction

Partial least squares (PLS) method has been applied to different research fields including systems identification, adaptive control, fault detection and diagnosis, parameter estimation and soft sensor prediction (Ding *et al.*, 2009; Yin *et al.*, 2012). Some effort has been attempted to model the relationship between manipulated variables

[1]The results in this chapter have been presented in or submitted to:

- Chen, K., Yu, J., Castillo, I., and Chiang, L.H. (November, 2013). Expectation-maximization and Bayesian inference based probabilistic PLS methods for soft sensor estimation and prediction of industrial processes with stochastic missing measurements. AIChE Annual Meeting. San Francisco, CA, USA.
- Chen, K., and Yu, J. (2014). Expectation-maximization and Bayesian inference based probabilistic partial least squares methods for soft sensor estimation with stochastic missing measurements. Revised to *Automatica*.

and controlled variables of control systems through PLS methods, which can provide predicted trajectories of controlled variables in model predictive control (MPC) applications (Wan *et al.*, 2010; Žáčěková *et al.*, 2011; Yin *et al.*, 2013). In order to handle significant nonlinearity in closed-loop control systems, multiple local PLS models are built based on the fuzzy clustering technique to obtain accurate trajectories of key process variables for model predictive controller so that better control performance can be achieved by the adaptive model-based control strategy than traditional proportional-integral-derivative (PID) controllers (Aumi *et al.*, 2013a). In addition, kernel PLS method is integrated with the decentralized multi-block concept to perform fault detection and identification in large-scale nonlinear systems, which leads to improved fault diagnosis capability than the traditional PLS and kernel PLS methods (Zhang *et al.*, 2010b). Meanwhile, the model structures and theoretical properties of PLS method and its variants are investigated so that their effectiveness are guaranteed for soft sensor estimation, process monitoring, fault detection and diagnosis (Ruscio, 2000; Li *et al.*, 2010). However, the aforementioned work focuses on PLS models without special treatments on stochastic missing measurements.

Random missing data is a challenging problem to deal with in various fields including systems identification, parameter and soft sensor estimation, and fault detection and diagnosis (Rännar *et al.*, 1995; Chen and Chen, 2000). Different types of methods have been developed for missing measurement estimation, including unconditional or conditional mean imputation, maximum likelihood estimation (MLE), expectation-maximization (EM) estimation and Bayesian multiple imputation (Rubin, 1976; Little and Rubin, 1987). For systems identification, the EM algorithm is employed to estimate the parameters in autoregressive-exogenous (ARX) models, which are subjected to missing measurements or switching states (Isaksson, 1993; Jin and Huang, 2012). Meanwhile, a gradient based approach is developed to identify the parameters in the

output error models for input-output systems with scarce outputs and the unavailable outputs are further estimated by using the identified model parameters (Ding *et al.*, 2011). More recently, maximum likelihood estimation for missing measurements is proposed by reformulating various scenarios into eight equivalent formulations within Gaussian process models for systems identification (Hansson and Wallin, 2012). An alternate solution is proposed by designing a robust filter through a linear matrix inequality method so as to handle the missing measurements in a nonlinear time-delayed system, where the overall mean square filtering error is guaranteed to be exponentially stable (Wei *et al.*, 2009). In order to minimize the upper bound of the filtering error covariance, a Riccati-like equation approach based on extended Kalman filter (EKF) is designed for the filtering problem in a class of nonlinear systems with multiple missing measurements characterized by any discrete-time distributions with known probability density functions (Hu *et al.*, 2012). Moreover, a robust estimator can be modified by using the properties of the monotonic function and squeeze rule to estimate the states of time-varying stochastic system with missing measurements (Liang and Zhou, 2011).

In parallel, some effort has been made towards estimating latent scores of PLS model with missing measurements in order to perform process monitoring and soft sensor estimation through nonlinear iterative partial least squares (NIPALS) based single component and multi-dimensional subspace projection methods (Nelson *et al.*, 1996). However, NIPALS algorithm may not be well-suited for data set with large number of missing measurements because it simply ignores missing measurements when computing loading matrices of PLS model and the NIPALS algorithm may not converge with too many missing measurements. Moreover, single component and multi-dimensional subspace projection approaches estimate latent scores of new observations with missing measurements by utilizing an existing PLS model, but may not work properly to build a PLS model from data set with significant portion of random missing measurements.

Later, alternating least squares (ALS) method is developed to handle missing measurements in PCA modeling, which is much more robust and reliable than NIPALS-like algorithms (Kroonenberg and De Leeuw, 1980). In parallel, the idea of probabilistic principal component analysis (PPCA) is proposed for complete data set by integrating conventional PCA model with probabilistic inference strategy to deal with stochastic system uncertainty (Bishop, 1999; Tipping and Bishop, 1999). The developed PPCA approach is employed to estimate missing measurements in the applications of traffic flow volume and gene expression profile with some success (Oba *et al.*, 2003; Qu *et al.*, 2009). Nevertheless, PPCA model does not take into account the quantitative relationships between system input and output variables, both of which are subjected to stochastic missing measurements.

In this study, two kinds of PPLS frameworks for handling stochastic missing measurements and performing soft sensor estimations are developed, which are expectation-maximization based PPLS (EM-PPLS) and Bayesian inference based PPLS (BI-PPLS). In the EM-PPLS method, missing measurements are initially imputed by variable means and the parameter values of EM-PPLS model begin with initial guess. Then, EM algorithm is employed to update latent scores and PPLS model parameters through data-driven iterative learning. Meanwhile, the missing measurements are iteratively re-estimated from the updated latent scores and model parameters after each EM step. Consequently, PPLS modeling and missing measurement estimation can be carried out concurrently along with the uncertainty handling in process measurements through a probabilistic strategy. In addition, the convergence property of the EM algorithm within the PPLS framework is investigated and the explicit projection matrix of the EM step in the PPLS model parameter space is identified. In order to avoid computationally expensive cross-validation based latent variable selection, a Bayesian inference based PPLS model framework is also proposed with prior distributions being defined on

PPLS model parameters. Thus, the optimal dimension of the latent variable subspace in the BI-PPLS framework can be identified by recursively implementing variational Bayesian inference strategy. After each iteratively updating step on PPLS model parameters, missing measurements are re-estimated by utilizing the new statistical expectations of latent scores and model parameters based upon the updated posterior probability distributions. As such, the proposed BI-PPLS method can lead to concurrent solutions of missing measurement estimation and latent variable selection within a Bayesian framework. Both the EM-PPLS and BI-PPLS model frameworks are applied to soft sensor estimation and prediction under significant portion of stochastic missing measurements.

The remainder of this chapter is organized as follows. The general framework of probabilistic PLS is described in Section 2.2. Then the EM based PPLS method under stochastic missing measurements along with its convergence properties is derived and proved in Section 2.3. In Section 2.4, variational Bayesian inference based PPLS method with missing measurements is developed for soft sensor estimation and also compared to EM based PPLS method. Both types of PPLS approaches are applied to a simulated example with different levels of stochastic missing measurements in Section 2.5. Finally, the conclusions of this article are drawn in Section 2.6.

2.2 General framework of probabilistic partial least squares

Factor analysis can be integrated with principal component analysis or partial least squares into a probabilistic PCA or probabilistic PLS framework for soft sensor estimation (Tipping and Bishop, 1999; Li *et al.*, 2011). According to regular PLS

and factor analysis methods, the input data matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_N] \in \mathbb{R}^{D_x \times N}$ consisting of N samples along D_x system input variables and the output matrix $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2\ \ldots\ \mathbf{y}_N] \in \mathbb{R}^{D_y \times N}$ along D_y system output variables can be characterized as a linear combination of factors (latent scores) $\mathbf{T} = [\mathbf{t}_1\ \mathbf{t}_2\ \ldots\ \mathbf{t}_N] \in \mathbb{R}^{D \times N}$ within a D-dimensional latent subspace. Meanwhile, $\mathbf{Z} = [\mathbf{X}\ \mathbf{Y}]^T = [\mathbf{z}_1\ \mathbf{z}_2\ \ldots\ \mathbf{z}_N] \in \mathbb{R}^{D_z \times N}$ is a stacked matrix with the dimension of $D_z = D_x + D_y$. To keep notations uncluttered, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{t}$ are used to denote measurement samples or latent scores in $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{T}$ respectively and they have the following relationships

$$\begin{cases} \mathbf{x} = \mu_x + \mathbf{P}\mathbf{t} + \mathbf{e} \\ \mathbf{y} = \mu_y + \mathbf{Q}\mathbf{t} + \mathbf{f} \end{cases} \tag{2.1}$$

where μ_x and μ_y represent the mean vectors of $\mathbf{X}$ and $\mathbf{Y}$ while $\mathbf{P}$ and $\mathbf{Q}$ are the loading matrices corresponding to $\mathbf{X}$ and $\mathbf{Y}$, respectively. By denoting $\mathcal{N}(\cdot)$ as a Gaussian distribution, $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I})$ and $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\mathbf{I})$ specify the isotropic Gaussian noises of the model with zero means and covariances of $\sigma_x^2\mathbf{I}$ and $\sigma_y^2\mathbf{I}$, respectively. Based on the isotropic Gaussian noise models of $\mathbf{e}$ and $\mathbf{f}$ in conjunction with Eq. (2.1), the conditional probabilities of $\mathbf{x}$ and $\mathbf{y}$ given $\mathbf{t}$ are expressed as (Li *et al.*, 2011)

$$p(\mathbf{x}|\mathbf{t}) = \mathcal{N}(\mathbf{x}|\mathbf{P}\mathbf{t} + \mu_x, \sigma_x^2\mathbf{I}) \tag{2.2}$$

$$p(\mathbf{y}|\mathbf{t}) = \mathcal{N}(\mathbf{y}|\mathbf{Q}\mathbf{t} + \mu_y, \sigma_y^2\mathbf{I}) \tag{2.3}$$

The prior distribution of $\mathbf{t}$ is also assumed to be Gaussian as follows

$$p(\mathbf{t}) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{I}) \tag{2.4}$$

Then the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$ are readily obtained by integrating out $\mathbf{t}$ and are Gaussian as

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t} = \mathcal{N}(\mathbf{x}|\mu_x, \mathbf{PP^T} + \sigma_x^2\mathbf{I}) \tag{2.5}$$

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t})d\mathbf{t} = \mathcal{N}(\mathbf{y}|\mu_x, \mathbf{QQ^T} + \sigma_y^2\mathbf{I}) \tag{2.6}$$

Further, the posterior distributions of $\mathbf{t}$ given $\mathbf{x}$ or $\mathbf{y}$ are obtained by bayesian inference. From Eqs. (2.2) and (2.3), the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ is computed by

$$p(\mathbf{x}, \mathbf{y}|\mathbf{t}, \theta) = \mathcal{N}(\mathbf{x}, \mathbf{y}|\mathbf{m}_{xy}, \mathbf{S}_{xy}) \tag{2.7}$$

where $\mathbf{m}_{xy} = \begin{bmatrix} \mathbf{Pt} + \mu_x \\ \mathbf{Qt} + \mu_y \end{bmatrix}$ is the mean, $\mathbf{S}_{xy} = \begin{bmatrix} \sigma_x^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_y^2\mathbf{I} \end{bmatrix}$ is the covariance and $\theta = \{\mathbf{P}, \mathbf{Q}, \mu_x, \mu_y, \sigma_x^2, \sigma_y^2\}$ represents the set of PPLS model parameters. By using Eqs. (2.4) and (2.7), the posterior distribution of $\mathbf{t}$ upon both $\mathbf{x}$ and $\mathbf{y}$ is expressed as

$$p(\mathbf{t}|\mathbf{x}, \mathbf{y}, \theta) = \mathcal{N}(\mathbf{t}|\mathbf{m}_t, \mathbf{S}_t) \tag{2.8}$$

where $\mathbf{m}_t = \mathbf{S}_t\mathbf{\Lambda}^T\mathbf{S}_{xy}^{-1}(\mathbf{z} - \mu_z)$ is the mean, $\mathbf{S}_t = (\mathbf{I} + \mathbf{\Lambda}^T\mathbf{S}_{xy}^{-1}\mathbf{\Lambda})^{-1}$ is the covariance, $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ denotes a stacked vector combining $\mathbf{x}$ and $\mathbf{y}$, $\mu_z = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$ is a stacked mean vector combining μ_x and μ_y, and $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix}$ represents a stacked matrix combining $\mathbf{P}$ and $\mathbf{Q}$.

Based on the PPLS model, the goal is to predict the values of output variables $\mathbf{y}$ given the input variables $\mathbf{x}$. Hence, the relationship between $\mathbf{x}$ and $\mathbf{y}$ can be modeled

as follows

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|\mathbf{x})d\mathbf{t} = \mathcal{N}(\mathbf{y}|\mathbf{m}, \mathbf{S}) \tag{2.9}$$

where $\mathbf{m} = \mathbf{Q}(\mathbf{P}\mathbf{P}^{\mathbf{T}} + \sigma_x^2\mathbf{I})^{-1}\mathbf{P}^T(\mathbf{x} - \mu_x) + \mu_y$ is the mean and $\mathbf{S} = \mathbf{Q}\sigma_x^2(\mathbf{P}\mathbf{P}^{\mathbf{T}} + \sigma_x^2\mathbf{I})^{-1}\mathbf{Q}^T + \sigma_y^2\mathbf{I}$ is the covariance.

Since the latent scores and model parameters are unknown in the likelihood function in Eq. (2.7), it is difficult to get solutions in closed form for both the latent scores and parameter values of PPLS model simultaneously. Therefore, EM algorithm can be employed to iteratively estimate the latent scores and model parameters from complete measurement data set (Li *et al.*, 2011).

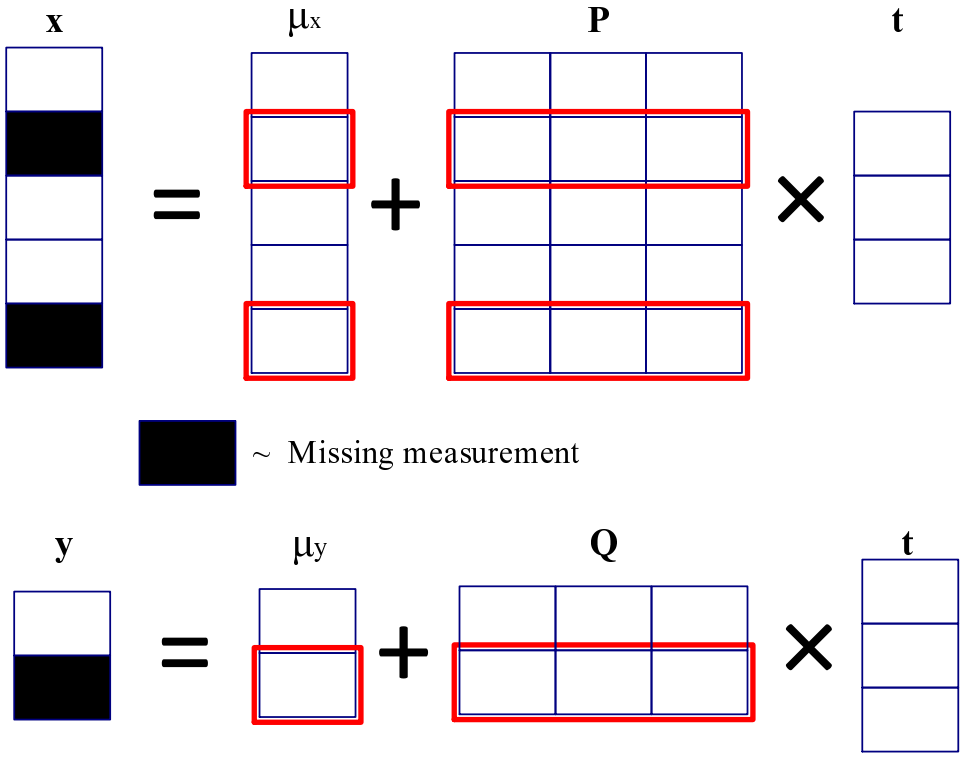

Figure 2.1: Illustrative diagram of PPLS based missing measurement estimation

2.3 Expectation-maximization algorithm for probabilistic partial least squares model and its convergence properties

2.3.1 Expectation-maximization based probabilistic partial least squares with missing measurements

It is necessary to consider why the measurements are missing before developing a valid method to handle them. The missing measurements can be categorized into three types (Little and Rubin, 1987; Walczak and Massart, 2001). (1) The measurements are missing completely at random (MCAR), which means that the probability that a measurement is missing is independent of both observed and missing data. (2) The measurements are missing at random (MAR), which indicates that the probability that a measurement is missing depends on the observed data. In process industry, some variables (quality variables) are MAR because they are not measured every sampling instance due to high measurement cost. (3) The measurements are not missing at random (NMAR), which means that the probability that a measurement is missing depends on the actual value of that missing measurement. For example, some measurement cannot be detected because they are beyond the ranges of sensors. It is necessary to incorporate the actual distribution of missing measurements in order to effectively handle the third type of missing measurements. However, the actual distribution of missing measurements is never available. In this research, we focus on treating the missing measurements which are MCAR or MAR.

The proposed PPLS method is aimed at handling stochastic missing measurements by taking uncertainty into consideration with probabilistic strategy. Without loss of

generality, input and output measurement vectors $\mathbf{x}$ and $\mathbf{y}$ can be partitioned into observed parts $\mathbf{x}^o$ and $\mathbf{y}^o$ and missing parts $\mathbf{x}^m$ and $\mathbf{y}^m$ by introducing permutation matrices $\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R}^o \\ \mathbf{R}^m \end{bmatrix}$ and $\bar{\mathbf{R}}^* = \begin{bmatrix} \mathbf{R}^{*o} \\ \mathbf{R}^{*m} \end{bmatrix}$ such that

$$\begin{bmatrix} \mathbf{x}^o \\ \mathbf{x}^m \end{bmatrix} = \bar{\mathbf{R}}\mathbf{x} = \begin{bmatrix} \mathbf{R}^o \\ \mathbf{R}^m \end{bmatrix} \mathbf{x} \tag{2.10}$$

$$\begin{bmatrix} \mathbf{y}^o \\ \mathbf{y}^m \end{bmatrix} = \bar{\mathbf{R}}^*\mathbf{y} = \begin{bmatrix} \mathbf{R}^{*o} \\ \mathbf{R}^{*m} \end{bmatrix} \mathbf{y} \tag{2.11}$$

It is further defined that

$$\begin{bmatrix} \mathbf{P}^o \\ \mathbf{P}^m \end{bmatrix} = \bar{\mathbf{R}}\mathbf{P} = \begin{bmatrix} \mathbf{R}^o \\ \mathbf{R}^m \end{bmatrix} \mathbf{P} \tag{2.12}$$

$$\begin{bmatrix} \mathbf{Q}^o \\ \mathbf{Q}^m \end{bmatrix} = \bar{\mathbf{R}}^*\mathbf{Q} = \begin{bmatrix} \mathbf{R}^{*o} \\ \mathbf{R}^{*m} \end{bmatrix} \mathbf{Q} \tag{2.13}$$

where $\mathbf{P}^o$ and $\mathbf{P}^m$ represent the submatrices of $\mathbf{P}$ corresponding to $\mathbf{x}^o$ and $\mathbf{x}^m$, respectively. Likewise, $\mathbf{Q}^o$ and $\mathbf{Q}^m$ are the submatrices of $\mathbf{Q}$ corresponding to $\mathbf{y}^o$ and $\mathbf{y}^m$. In order to estimate the latent scores and model parameters by EM algorithm, the random missing measurements are initially imputed by variable means and the values of model parameters are filled by initial guess. Then, E-step and M-step of EM algorithm can be carried out iteratively.

Theorem 1 *The updates of model parameters and missing measurements satisfy the following relationships*

$$\mu_x^{k+1} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n^k - \mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k]) \tag{2.14}$$

$$\mu_y^{k+1} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n^k - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k]) \tag{2.15}$$

$$\mathbf{P}^{k+1} = \left[\sum_{n=1}^{N}(\mathbf{x}_n^k - \mu_x^k)\mathbb{E}[\mathbf{t}_n^k]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\right]^{-1} \tag{2.16}$$

$$\mathbf{Q}^{k+1} = \left[\sum_{n=1}^{N}(\mathbf{y}_n^k - \mu_y^k)\mathbb{E}[\mathbf{t}_n^k]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\right]^{-1} \tag{2.17}$$

$$\begin{aligned}\sigma_x^{2,k+1} =& \frac{1}{D_x N}\sum_{n=1}^{N} Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{P}^{kT}\mathbf{P}^k) \\ &- 2\mathbb{E}[\mathbf{t}_n^k]^T\mathbf{P}^{kT}(\mathbf{x}_n^k - \mu_x^k) + ||\mathbf{x}_n^k - \mu_x^k||^2\end{aligned} \tag{2.18}$$

$$\begin{aligned}\sigma_y^{2,k+1} =& \frac{1}{D_y N}\sum_{n=1}^{N} Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{Q}^{kT}\mathbf{Q}^k) \\ &- 2\mathbb{E}[\mathbf{t}_n^k]^T\mathbf{Q}^{kT}(\mathbf{y}_n^k - \mu_y^k) + ||\mathbf{y}_n^k - \mu_y^k||^2\end{aligned} \tag{2.19}$$

$$\mathbf{x}_n^{k,m} = \mathbf{P}^{k,m}\mathbb{E}[\mathbf{t}_n^k] + \mu_x^{k,m} \tag{2.20}$$

$$\mathbf{y}_n^{k,m} = \mathbf{Q}^{k,m}\mathbb{E}[\mathbf{t}_n^k] + \mu_y^{k,m} \tag{2.21}$$

where k in superscription denotes the number of iteration, $\mathbb{E}[\cdot]$ represents mathematical expectation, $Tr(\cdot)$ denotes matrix trace and $\mu_x^{k,m}$ and $\mu_y^{k,m}$ are the sub-vectors of μ_x^k and μ_y^k corresponding to $\mathbf{x}_n^m$ and $\mathbf{y}_n^m$, respectively. In each iteration, the first and second order statistics of $\mathbf{t}_n$ are computed by $\mathbb{E}[\mathbf{t}_n] = \mathbf{S}_t\mathbf{\Lambda}^T\mathbf{S}_{xy}^{-1}(\mathbf{z}_n - \mu_z)$ and $\mathbb{E}[\mathbf{t}_n\mathbf{t}_n{}^T] = \mathbf{S}_t + \mathbb{E}[\mathbf{t}_n]\mathbb{E}[\mathbf{t}_n]^T$, respectively.

Proof The log-likelihood function is written as

$$\mathcal{L} = \ln p(\mathbf{X}, \mathbf{Y}|\mathbf{T}, \theta) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{t}_n, \theta) \tag{2.22}$$

In the E-step, the $\mathbf{T}$ in the log-likelihood function is replaced by its expectation with respect to the posterior distribution in Eq. (2.8) as follows

$$\begin{aligned}
&\mathbb{E}_{\mathbf{T}\sim p(\mathbf{T}|\mathbf{X},\mathbf{Y},\theta)}[\ln p(\mathbf{X}, \mathbf{Y}|\mathbf{T}, \theta)] = \\
&-\sum_{n=1}^{N}[\frac{D_x}{2}\ln(2\pi\sigma_x^{2,k}) - \frac{1}{\sigma_x^{2,k}}\mathbb{E}[\mathbf{t}_n^k]^T\mathbf{P}^{kT}(\mathbf{x}_n^k - \mu_x^k) \\
&+\frac{1}{2\sigma_x^{2,k}}||\mathbf{x}_n^k - \mu_x^k||^2 + \frac{1}{2\sigma_x^{2,k}}Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{P}^{kT}\mathbf{P}^k) \\
&+\frac{D_y}{2}\ln(2\pi\sigma_y^{2,k}) - \frac{1}{\sigma_y^{2,k}}\mathbb{E}[\mathbf{t}_n^k]^T\mathbf{Q}^{kT}(\mathbf{y}_n^k - \mu_y^k) \\
&+\frac{1}{2\sigma_y^{2,k}}||\mathbf{y}_n^k - \mu_y^k||^2 + \frac{1}{2\sigma_y^{2,k}}Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{Q}^{kT}\mathbf{Q}^k)
\end{aligned}$$

In the M-step, the first-order derivative is computed with respect to each individual component of the parameters so as to maximize the log-likelihood function over model parameters. Therefore, the optimal values of local parameters in each M-step are calculated by setting the derivatives to zero. In the k-th iteration, the partial derivative of the log-likelihood function with respect to μ_x is given by

$$\frac{\partial\mathcal{L}}{\partial\mu_x}|_{\mu_x=\mu_x^k} = \frac{2}{\sigma_x^{2,k}}\sum_{n=1}^{N}(\mathbf{x}_n^k - \mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k]) - \frac{2}{\sigma_x^{2,k}}N\mu_x^k \tag{2.23}$$

As a result, the updating equation for μ_x in Eq. (2.14) can be obtained by setting the above derivative to zero. Similarly, the derivative of the log-likelihood function with respect to μ_y is expressed as

$$\frac{\partial\mathcal{L}}{\partial\mu_y}|_{\mu_y=\mu_y^k} = \frac{2}{\sigma_y^{2,k}}\sum_{n=1}^{N}(\mathbf{y}_n^k - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k]) - \frac{2}{\sigma_y^{2,k}}N\mu_y^k \tag{2.24}$$

Then, the updating equation for μ_y follows Eq. (2.15) by setting the derivative to zero. The derivative of the log-likelihood function with respect to $\mathbf{P}$ is given by

$$\frac{\partial \mathcal{L}}{\partial \mathbf{P}}|_{\mathbf{P}=\mathbf{P}^k} = \sum_{n=1}^{N}(\frac{1}{\sigma_x^{2,k}}(\mathbf{x}_n^k - \mu_x^k)\mathbb{E}[\mathbf{t}_n^k] - \frac{1}{\sigma_x^{2,k}}\mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]) \tag{2.25}$$

Accordingly, solving the above equation that the derivative equals zero leads to the updating equation of $\mathbf{P}$ in Eq. (2.16). Likewise, the derivative of the log-likelihood function with respect to $\mathbf{Q}$ is as follows

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}}|_{\mathbf{Q}=\mathbf{Q}^k} = \sum_{n=1}^{N}(\frac{1}{\sigma_y^{2,k}}(\mathbf{y}_n^k - \mu_y^k)\mathbb{E}[\mathbf{t}_n^k] - \frac{1}{\sigma_y^{2,k}}\mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]) \tag{2.26}$$

Then, the updating equation for $\mathbf{Q}$ in Eq. (2.17) is derived by setting the corresponding derivative to zero. Following this line, the derivative of the log-likelihood function with respect to σ_x^2 is written as follows

$$\frac{\partial \mathcal{L}}{\partial \sigma_x^2}|_{\sigma_x^2=\sigma_x^{2,k}} = \frac{1}{2\sigma_x^{4,k}}\sum_{n=1}^{N}(||\mathbf{x}_n^k-\mu_x^k||^2+Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{P}^{kT}\mathbf{P}^k)-2\mathbb{E}(\mathbf{t}_n^k)\mathbf{P}^{kT}(\mathbf{x}_n^k-\mu_x^k)-\sigma_x^{2,k}D_x) \tag{2.27}$$

Setting the derivative to zero yields the updating equation for σ_x^2 in Eq. (2.18). A similar derivative of the log-likelihood function with respect to σ_y^2 is given by

$$\frac{\partial \mathcal{L}}{\partial \sigma_y^2}|_{\sigma_y^2=\sigma_y^{2,k}} = \frac{1}{2\sigma_y^{4,k}}\sum_{n=1}^{N}(||\mathbf{y}_n^k-\mu_y^k||^2+Tr(\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{kT}]\mathbf{Q}^{kT}\mathbf{Q}^k)-2\mathbb{E}(\mathbf{t}_n^k)\mathbf{Q}^{kT}(\mathbf{y}_n^k-\mu_y^k)-\sigma_y^{2,k}D_y) \tag{2.28}$$

Finally, the updating equation for σ_y^2 in Eq. (2.19) can be obtained in a similar way. As such, the missing measurements can be re-estimated by using the updated latent scores and model parameters from E-step and M-step according to Eqs. (2.20) and (2.21). ∎

In the EM algorithm, the E-step, M-step and re-estimation step are iteratively conducted until the model parameters converge. The re-estimation step for missing measurements is illustrated in Fig. 2.1.

2.3.2 Convergence of expectation-maximization algorithm for probabilistic partial least squares model

The general convergence of EM algorithm for Gaussian mixture models has been studied in literature (Xu and Jordan, 1996). In this subsection, the specific convergence property of EM algorithm within the PPLS model framework is further investigated and proved. First, the EM search direction in the parameter space is derived explicitly by comparing the parameters in two consecutive steps. Then, it can be further demonstrated that the EM search direction has a positive projection on the gradient of the likelihood function, leading to monotonic increase in the likelihood function. Finally, the convergence of EM algorithm for PPLS can be proved based on the nature that the likelihood function is upper bounded.

Theorem 2 *At each iteration of EM algorithm, the updating equations of PPLS model parameters can be derived as follows*

$$\mu_x^{k+1} - \mu_x^k = H_{\mu_x}^k \frac{\partial \mathcal{L}}{\partial \mu_x}|_{\mu_x=\mu_x^k} \tag{2.29}$$

$$\mu_y^{k+1} - \mu_y^k = H_{\mu_y}^k \frac{\partial \mathcal{L}}{\partial \mu_y}|_{\mu_y=\mu_y^k} \tag{2.30}$$

$$(\mathbf{P}^{k+1} - \mathbf{P}^k)\mathbf{C}^k = H_P^k \frac{\partial \mathcal{L}}{\partial \mathbf{P}}|_{\mathbf{P}=\mathbf{P}^k} \tag{2.31}$$

$$(\mathbf{Q}^{k+1} - \mathbf{Q}^k)\mathbf{C}^k = H_Q^k \frac{\partial \mathcal{L}}{\partial \mathbf{Q}}|_{\mathbf{Q}=\mathbf{Q}^k} \tag{2.32}$$

$$\sigma_x^{2,k+1} - \sigma_x^{2,k} = H_{\sigma_x^2}^k \frac{\partial \mathcal{L}}{\partial \sigma_x^2}|_{\sigma_x^2=\sigma_x^{2,k}} \tag{2.33}$$

$$\sigma_y^{2,k+1} - \sigma_y^{2,k} = H_{\sigma_y^2}^k \frac{\partial \mathcal{L}}{\partial \sigma_y^2}|_{\sigma_y^2=\sigma_y^{2,k}} \tag{2.34}$$

where

$$\mathbf{C}^k = \sum_{n=1}^{N} \mathbb{E}[\mathbf{t}_n^k \mathbf{t}_n^{k^T}]$$

$$H_{\mu_x}^k = \frac{\sigma_x^{2,k}}{2N}$$

$$H_{\mu_y}^k = \frac{\sigma_y^{2,k}}{2N}$$

$$H_P^k = \sigma_x^{2,k}$$

$$H_Q^k = \sigma_y^{2,k}$$

$$H_{\sigma_x^2}^k = \frac{\sigma_x^{4,k}}{D_x N}$$

$$H_{\sigma_y^2}^k = \frac{\sigma_y^{4,k}}{D_y N}$$

Proof Multiplying $H_{\mu_x}^k$ with the partial derivative of log-likelihood function as given in Eq. (2.23) and further combining it with Eq. (2.14) can lead to Eq. (2.29) as follows

$$\begin{aligned} H_{\mu_x}^k \frac{\partial \mathcal{L}}{\partial \mu_x}|_{\mu_x=\mu_x^k} &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n^k - \mathbf{P}^k \mathbb{E}[\mathbf{t}_n^k]) - \mu_x^k \\ &= \mu_x^{k+1} - \mu_x^k \end{aligned} \tag{2.35}$$

Similarly, Eq. (2.30) is derived by multiplying $H^k_{\mu_y}$ with Eq. (2.24) and further combining it with Eq. (2.15) as follows

$$\begin{aligned} H^k_{\mu_y}\frac{\partial \mathcal{L}}{\partial \mu_y}|_{\mu_y=\mu_y^k} &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n^k - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k]) - \mu_y^k \\ &= \mu_y^{k+1} - \mu_y^k \end{aligned} \tag{2.36}$$

The same strategy can be used to prove Eqs. (2.31), (2.32), (2.33) and (2.34) as follows

$$\begin{aligned} H^k_{P}\frac{\partial \mathcal{L}}{\partial \mathbf{P}}|_{\mathbf{P}=\mathbf{P}^k} &= \sum_{n=1}^{N}(\mathbf{x}_n^k - \mu_x^k)\mathbb{E}[\mathbf{t}_n^k] - \mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{k^T}] \\ &= (\mathbf{P}^{k+1} - \mathbf{P}^k)\mathbf{C}^k \end{aligned} \tag{2.37}$$

$$\begin{aligned} H^k_{Q}\frac{\partial \mathcal{L}}{\partial \mathbf{Q}}|_{\mathbf{Q}=\mathbf{Q}^k} &= \sum_{n=1}^{N}(\mathbf{y}_n^k - \mu_y^k)\mathbb{E}[\mathbf{t}_n^k] - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k\mathbf{t}_n^{k^T}] \\ &= (\mathbf{Q}^{k+1} - \mathbf{Q}^k)\mathbf{C}^k \end{aligned} \tag{2.38}$$

$$\begin{aligned} &H^k_{\sigma_x^2}\frac{\partial \mathcal{L}}{\partial \sigma_x^2}|_{\sigma_x^2=\sigma_x^{2,k}} \\ &= -\sigma_x^{2,k} + \frac{1}{D_x N}\sum_{i=1}^{N}(Tr(\mathbb{E}[\mathbf{t}_i^k\mathbf{t}_i^{k^T}]\mathbf{P}^{k^T}\mathbf{P}^k) \\ &\quad -2\mathbb{E}[\mathbf{t}_i^k]^T\mathbf{P}^{k^T}(\mathbf{x}_i^k - \mu_x^k) + ||\mathbf{x}_i^k - \mu_x^k||^2) \\ &= \sigma_x^{2,k+1} - \sigma_x^{2,k} \end{aligned} \tag{2.39}$$

$$\begin{aligned} &H^k_{\sigma_y^2}\frac{\partial \mathcal{L}}{\partial \sigma_y^2}|_{\sigma_y^2=\sigma_y^{2,k}} \\ &= -\sigma_y^{2,k} + \frac{1}{D_y N}\sum_{i=1}^{N}Tr(\mathbb{E}[\mathbf{t}_i^k\mathbf{t}_i^{k^T}]\mathbf{Q}^{k^T}\mathbf{Q}^k) \\ &\quad -2\mathbb{E}[\mathbf{t}_i^k]^T\mathbf{Q}^{k^T}(\mathbf{y}_i^k - \mu_y^k) + ||\mathbf{y}_i^k - \mu_y^k||^2 \\ &= \sigma_y^{2,k+1} - \sigma_y^{2,k} \end{aligned} \tag{2.40}$$

This completes the proof of Theorem 2. ∎

Given the conditions in Theorem 2, the following corollary can be further defined.

Corollary 1 *In each iteration of EM algorithm for PPLS model, the search direction* $\vartheta^{k+1} - \vartheta^k$ *for all PPLS model parameters has a positive projection on the gradient* $\frac{\partial \mathcal{L}}{\partial \vartheta}|_{\vartheta=\vartheta^k}$, *where* $\vartheta = \{\mu_x, \mu_y, P, Q, \sigma_x^2, \sigma_y^2\}$.

Proof Based on the trace trick $\mathbf{u}^T\mathbf{V}\mathbf{u} = Tr(\mathbf{u}\mathbf{V}\mathbf{u}^T)$ for any arbitrary vector $\mathbf{u}$ and matrix $\mathbf{V}$, the updates for variances σ_x^2 and σ_y^2 can be reformulated in the quadratic forms as follows

$$\sigma_x^{2,k+1} = \frac{1}{D_x N} Tr\Big(\sum_{n=1}^{N}(\mathbf{x}_n - \mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k] - \mu_x^k)(\mathbf{x}_n - \mathbf{P}^k\mathbb{E}[\mathbf{t}_n^k] - \mu_x^k)^T\Big) \tag{2.41}$$

$$\sigma_y^{2,k+1} = \frac{1}{D_y N} Tr\Big(\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k] - \mu_y^k)(\mathbf{y}_n - \mathbf{Q}^k\mathbb{E}[\mathbf{t}_n^k] - \mu_y^k)^T\Big) \tag{2.42}$$

which automatically satisfy the implicit constraints $\sigma_x^2 > 0$ and $\sigma_y^2 > 0$. Hence, a positive definite matrix $\mathbf{H} = diag\{H_{\mu_x}, H_{\mu_y}, H_P, H_Q, H_{\sigma_x^2}, H_{\sigma_y^2}\}$ can be constructed, where $diag\{\cdot\}$ represents a diagonal matrix. Then, all the updates in Theorem 2 can be expressed as the following general form

$$\vartheta^{k+1} - \vartheta^k = \mathbf{H}\frac{\partial L}{\partial \vartheta}|_{\vartheta=\vartheta^k} \tag{2.43}$$

Due to the positive definiteness of $\mathbf{H}$, it can be obtained that

$$\frac{\partial L}{\partial \vartheta}^T|_{\vartheta=\vartheta^k}(\vartheta^{k+1} - \vartheta^k) = \frac{\partial L}{\partial \vartheta}^T|_{\vartheta=\vartheta^k}\mathbf{H}\frac{\partial L}{\partial \vartheta}|_{\vartheta=\vartheta^k} > 0 \tag{2.44}$$

Therefore, the search direction $\vartheta^{k+1} - \vartheta^k$ has a positive projection on the gradient

$\frac{\partial L}{\partial \vartheta}|_{\vartheta=\vartheta^k}$, which ensures the log-likelihood function value to monotonically increase throughout all EM iterations. Since we have $\sigma_x^2 > 0$ and $\sigma_y^2 > 0$, the log-likelihood function in Eq. (2.22) is upper bounded within the entire parameter space and thus the convergence of EM algorithm in PPLS model framework is guaranteed. ∎

2.4 Bayesian inference based probabilistic partial least squares with missing measurements

Based on the probabilistic reformulations of PLS, a Bayesian framework for PPLS model can be obtained by introducing prior distributions on PPLS model parameters. To avoid computationally intensive cross-validation based latent variable selection, continuous hyperparameters are set in order to identify the appropriate dimension of latent subspace automatically through Bayesian inference strategy. The following hierarchial prior distributions on $\mathbf{\Lambda}$ can be defined in the framework of automatic relevance determination

$$\begin{aligned} p(\mathbf{\Lambda}|\boldsymbol{\alpha}) &= \mathcal{N}(\mathbf{\Lambda}|\mathbf{0}, diag\{\alpha_1^{-1}, \alpha_2^{-1}, \ldots, \alpha_{D_x}^{-1}\}) \\ &= \prod_{i=1}^{D_x} (\frac{\alpha_i}{2\pi})^{\frac{D_x}{2}} \exp\{-\frac{1}{2}\alpha_i ||\Lambda_i||^2\} \end{aligned} \tag{2.45}$$

where each α_i in the vector of hyperparameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{D_x}\}$ controls the precision of the i-th column Λ_i in $\mathbf{\Lambda}$ (Bishop, 1999). By defining $\tau_x = 1/\sigma_x^2$, $\tau_y = 1/\sigma_y^2$ and $\Psi = diag\{\tau_x \mathbf{I}_{D_x \times D_x}, \tau_y \mathbf{I}_{D_y \times D_y}\}$, the prior distributions of the remaining model parameters are specified as follows

$$p(\mu_z|\tau_x, \tau_y) = \mathcal{N}(\mu_z|0, (\beta\Psi)^{-1})$$

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{D_x} \Gamma(\alpha_i|a_\alpha, b_\alpha)$$

$$p(\tau_x) = \Gamma(\tau_x|c_x, d_x)$$

$$p(\tau_y) = \Gamma(\tau_y|c_y, d_y)$$

where $\beta, a_\alpha, b_\alpha, c_x, d_x, c_y$ and d_y denote the parameters of the prior distributions and $\Gamma(\cdot)$ represents a Gamma distribution. Setting the above distribution parameters to small values can lead to broad noninformative prior distributions on PPLS model parameters.

Different from the EM-PPLS method, the model parameters in BI-PPLS are treated as random numbers following some distributions. By setting hierarchical priors on the loading matrix, full bayesian treatment can be applied to not only model parameters but also the hyperparameter $\boldsymbol{\alpha}$. The model parameters are obtained by taking expectation with respect to their corresponding posterior distribution, while the latent variable selection can also be achieved according to the posterior distribution of $\boldsymbol{\alpha}$. When the posterior distribution of any particular α_i centers at a large value, the corresponding Λ_i tends to center closely at zero and that direction in the latent subspace can be switched off.

The posterior probability density function of latent scores $\mathbf{T}$ and parameters $\Theta = \{\boldsymbol{\Lambda}, \boldsymbol{\alpha}, \mu_z, \tau_x, \tau_y\}$ of Bayesian inference based PPLS model given system input and output data $\mathbf{X}$ and $\mathbf{Y}$ is as follows

$$p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y}|\mathbf{T}, \Theta)p(\mathbf{T}, \Theta)}{\iint p(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)d\mathbf{T}d\Theta}. \tag{2.46}$$

Since the evidence part in the denominator requires multiple integration and is computationally intractable, the posterior probability function may not be computed directly.

Thus, variational Bayesian inference is adopted to approximate the true posterior distribution by a proxy distribution $Q(\mathbf{T}, \Theta)$. Based on Gibbs' inequality, the Kullback-Leibler divergence between the true posterior distribution $p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y})$ and the proxy distribution $Q(\mathbf{T}, \Theta)$ follows

$$-\iint Q(\mathbf{T}, \Theta) \ln \frac{p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y})}{Q(\mathbf{T},\Theta)} d\mathbf{T} d\Theta \geq 0 \tag{2.47}$$

where the equality holds only if $Q(\mathbf{T}, \Theta) = p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y})$. Though it is difficult to calculate $p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y})$, the following function can be maximized to find the optimal approximation for the true posterior probability density function

$$F = \iint Q(\mathbf{T}, \Theta) \ln \frac{p(\mathbf{X}, \mathbf{Y}|\mathbf{T}, \Theta) p(\mathbf{T}, \Theta)}{Q(\mathbf{T}, \Theta)} d\mathbf{T} d\Theta \tag{2.48}$$

The corresponding proxy distribution can be constructed as

$$Q(\mathbf{T}, \Theta) = Q(\mathbf{T}) Q(\mathbf{\Lambda}) Q(\boldsymbol{\alpha}) Q(\mu_z) Q(\tau_x) Q(\tau_y) \tag{2.49}$$

where

$$Q(\mathbf{T}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{t}_n | m_{\mathbf{t}_n}, \Sigma_{\mathbf{t}_n}) \tag{2.50}$$

$$Q(\mu_z) = \mathcal{N}(\mu_z | m_{\mu_z}, \Sigma_{\mu_z}) \tag{2.51}$$

$$Q(\mathbf{\Lambda}) = \prod_{d=1}^{D_z} \mathcal{N}(\Lambda_d | m_{\Lambda_d}, \Sigma_{\Lambda_d}) \tag{2.52}$$

$$Q(\boldsymbol{\alpha}) = \prod_{i=1}^{D_x} \Gamma(\alpha_i | \tilde{a_\alpha}, \tilde{b_{\alpha_i}}) \tag{2.53}$$

$$Q(\tau_x) = \Gamma(\tau_x | \tilde{c_x}, \tilde{d_x}) \tag{2.54}$$

$$Q(\tau_y) = \Gamma(\tau_y | \tilde{c_y}, \tilde{d_y}) \tag{2.55}$$

with $m_{\mathbf{t}_n}$, $\Sigma_{\mathbf{t}_n}$, m_{μ_z}, Σ_{μ_z}, m_{Λ_d}, Σ_{Λ_d}, $\tilde{a_\alpha}$, $\tilde{b_{\alpha_i}}$, $\tilde{c_x}$, $\tilde{d_x}$, $\tilde{c_y}$ and $\tilde{d_y}$ being the parameters of the proxy distributions to be estimated. Instead of considering model parameters as unknown but fixed values in EM based PPLS method, the model parameters in Bayesian inference based PPLS method are randomly selected from proxy distributions. Thus, the parameters in proxy distributions are iterated through variational Bayesian inference strategy in BI-PPLS approach. The parameters of proxy distributions and missing measurements can be updated iteratively after the initial guess.

Theorem 3 *The missing measurements and parameters of proxy distributions can be updated as follows*

$$m_{\mathbf{t}_n} = \Sigma_{\mathbf{t}_n}\mathbb{E}[\mathbf{\Lambda}^T]\begin{bmatrix} \frac{\tilde{c_x}}{\tilde{d_x}}\mathbf{I}_{D_x\times D_x} & \mathbf{0} \\ \mathbf{0} & \frac{\tilde{c_y}}{\tilde{d_y}}\mathbf{I}_{D_y\times D_y}\end{bmatrix}(\mathbf{z}_n - m_{\mu_z}) \tag{2.56}$$

$$\Sigma_{\mathbf{t}_n} = \left(\mathbf{I} + \mathbb{E}[\mathbf{\Lambda}]^T\begin{bmatrix} \frac{\tilde{c_x}}{\tilde{d_x}}\mathbf{I}_{D_x\times D_x} & \mathbf{0} \\ \mathbf{0} & \frac{\tilde{c_y}}{\tilde{d_y}}\mathbf{I}_{D_y\times D_y}\end{bmatrix}\mathbb{E}[\mathbf{\Lambda}]\right)^{-1} \tag{2.57}$$

$$m_{\mu_z} = \begin{bmatrix} \frac{\tilde{c_x}}{\tilde{d_x}}\mathbf{I}_{D_x\times D_x} & \mathbf{0} \\ \mathbf{0} & \frac{\tilde{c_y}}{\tilde{d_y}}\mathbf{I}_{D_y\times D_y}\end{bmatrix}\Sigma_{\mu_z}\sum_{n=1}^{N}(\mathbf{z}_n - \mathbb{E}[\mathbf{\Lambda}]m_{\mathbf{t}_n}) \tag{2.58}$$

$$\Sigma_{\mu_z} = \begin{bmatrix} (\beta + N\frac{\tilde{c_x}}{\tilde{d_x}})^{-1}\mathbf{I}_{D_x\times D_x} & \mathbf{0} \\ \mathbf{0} & (\beta + N\frac{\tilde{c_y}}{\tilde{d_y}})^{-1}\mathbf{I}_{D_y\times D_y}\end{bmatrix} \tag{2.59}$$

$$m_{\Lambda_d} = \begin{cases} \frac{\tilde{c_x}}{\tilde{d_x}}\Sigma_{\Lambda_d}\sum_{n=1}^{N} m_{\mathbf{t}_n}(z_{n(d)} - m_{\mu_z(d)}) \\ (if\ 1 \le d \le D_x) \\ \frac{\tilde{c_y}}{\tilde{d_y}}\Sigma_{\Lambda_d}\sum_{n=1}^{N} m_{\mathbf{t}_n}(z_{n(d)} - m_{\mu_z(d)}) \\ (if\ D_x + 1 \le d \le D_z) \end{cases} \tag{2.60}$$

$$\Sigma_{\Lambda_d} = \begin{cases} (diag(\mathbb{E}[\boldsymbol{\alpha}]) + \frac{\tilde{c_x}}{\tilde{d_x}} \sum_{n=1}^{N} \mathbb{E}[\mathbf{t}_n \mathbf{t}_n^T])^{-1} \\ (if\ 1 \leq d \leq D_x) \\ (diag(\mathbb{E}[\boldsymbol{\alpha}]) + \frac{\tilde{c_y}}{\tilde{d_y}} \sum_{n=1}^{N} \mathbb{E}[\mathbf{t}_n \mathbf{t}_n^T])^{-1} \\ (if\ D_x + 1 \leq d \leq D_z) \end{cases} \tag{2.61}$$

$$\tilde{a_\alpha} = a_\alpha + \frac{1}{2} D_x \tag{2.62}$$

$$\tilde{b_{\alpha_i}} = b_\alpha + \frac{E[||\Lambda_i||^2]}{2} \tag{2.63}$$

$$\tilde{c_x} = c_x + \frac{N D_x}{2} \tag{2.64}$$

$$\begin{aligned} &\tilde{d_x} = d_x + \\ &\frac{1}{2} \sum_{n=1}^{N} Tr((\mathbf{x}_n - \mathbb{E}[\mathbf{P}]\mathbb{E}[\mathbf{t}_n] \\ &-\mathbb{E}[\mu_x])(\mathbf{x}_n - \mathbb{E}[\mathbf{P}]\mathbb{E}[\mathbf{t}_n] - \mathbb{E}[\mu_x])^T) \end{aligned} \tag{2.65}$$

$$\tilde{c_y} = c_y + \frac{N D_y}{2} \tag{2.66}$$

$$\begin{aligned} &\tilde{d_y} = d_y + \\ &\frac{1}{2} \sum_{n=1}^{N} Tr((\mathbf{y}_n - \mathbb{E}[\mathbf{Q}]\mathbb{E}[\mathbf{t}_n] \\ &-\mathbb{E}[\mu_y])(\mathbf{y}_n - \mathbb{E}[\mathbf{Q}]\mathbb{E}[\mathbf{t}_n] - \mathbb{E}[\mu_y])^T) \end{aligned} \tag{2.67}$$

$$\boldsymbol{\Lambda}_{new} = \int \boldsymbol{\Lambda} Q(\boldsymbol{\Lambda}) d\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{P}_{new} \\ \mathbf{Q}_{new} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{-1} \begin{bmatrix} \mathbf{P}^o_{new} \\ \mathbf{P}^m_{new} \end{bmatrix} \\ \mathbf{R}^{*-1} \begin{bmatrix} \mathbf{Q}^o_{new} \\ \mathbf{Q}^m_{new} \end{bmatrix} \end{bmatrix} \tag{2.68}$$

$$\mathbf{t}_{n,new} = \int \mathbf{t}_n Q(\mathbf{t}_n) d\mathbf{t}_n \tag{2.69}$$

$$\mu_{z,new} = \int \mu_z Q(\mu_z) d\mu_z = \begin{bmatrix} \mu_{x,new} \\ \mu_{y,new} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{-1} \begin{bmatrix} \mu^o_{x,new} \\ \mu^m_{x,new} \end{bmatrix} \\ \mathbf{R}^{*-1} \begin{bmatrix} \mu^o_{y,new} \\ \mu^m_{y,new} \end{bmatrix} \end{bmatrix} \tag{2.70}$$

$$\mathbf{x}^m_{n,new} = \mathbf{P}^m_{new} \mathbf{t}_{n,new} + \mu^m_{x,new} \tag{2.71}$$

$$\mathbf{y}^m_{n,new} = \mathbf{Q}^m_{new} \mathbf{t}_{n,new} + \mu^m_{y,new} \tag{2.72}$$

where Λ_d *denotes a column vector corresponding to the d-th row of* $\mathbf{\Lambda}$, $z_{n(d)}$ *is the d-th element of* $\mathbf{z}_n$, $m_{\mu_z(d)}$ *is the d-th element of* m_{μ_z}, $\mathbf{\Lambda}_{new}$, $\mathbf{P}_{new}$, $\mathbf{Q}_{new}$, $\mathbf{t}_{n,new}$, $\mu_{z,new}$, $\mu_{x,new}$ *and* $\mu_{y,new}$ *are the updated expectations of* $\mathbf{\Lambda}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{t}_n$, μ_z, μ_x, μ_y, *respectively, and* $\mathbf{x}^m_{n,new}$ *and* $\mathbf{y}^m_{n,new}$ *are the updated estimations for missing measurements. Similarly,* $\mathbf{P}_{new}$, $\mathbf{Q}_{new}$, $\mu_{x,new}$ *and* $\mu_{y,new}$ *are divided into* $\mathbf{P}^o_{new}$, $\mathbf{Q}^o_{new}$, $\mu^o_{x,new}$ *and* $\mu^o_{y,new}$ *corresponding to the observed parts of* $\mathbf{x}_n$ *and* $\mathbf{y}_n$, *respectively, and* $\mathbf{P}^m_{new}$, $\mathbf{Q}^m_{new}$, $\mu^m_{x,new}$ *and* $\mu^m_{y,new}$ *corresponding to the missing parts of* $\mathbf{x}_n$ *and* $\mathbf{y}_n$, *respectively. It should be noted that the expectations in above updating equations are taken with respect to the corresponding proxy distributions. In addition, the expectations of* $\mathbf{P}$ *and* $\mathbf{Q}$ *are computed by using the expectation of* $\mathbf{\Lambda}$ *given* $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{P}^T & \mathbf{Q}^T \end{bmatrix}^T$.

Proof Eq. (2.47) can be rewritten as

$$\begin{aligned} &- \iint Q(\mathbf{T}, \Theta) \ln \frac{p(\mathbf{T}, \Theta | \mathbf{X}, \mathbf{Y})}{Q(\mathbf{T}, \Theta)} d\mathbf{T} d\Theta = \\ &\ln p(\mathbf{X}, \mathbf{Y}) - \iint Q(\mathbf{T}, \Theta) \ln \frac{p(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)}{Q(\mathbf{T}, \Theta)} d\mathbf{T} d\Theta \end{aligned} \tag{2.73}$$

where the first term on the right side can be regarded as a constant while the second term is identical as F in Eq. (2.48). The objective is to find the optimal approximate proxy distribution for the computationally intractable posterior distribution by

minimizing the Kullback-Leibler divergence between $Q(\mathbf{T}, \Theta)$ and $p(\mathbf{T}, \Theta|\mathbf{X}, \mathbf{Y})$. Since $\ln p(\mathbf{X}, \mathbf{Y})$ is a constant, the optimization problem is equivalent to maximizing F in Eq. (2.48). Therefore, variational Bayesian inference is employed to compute the optimal proxy distribution.

The optimization of F with respect to each element in $Q(\mathbf{T}, \Theta)$ is conducted every time while keeping the other elements constant. First, the proxy distribution of PPLS latent scores is optimized under the condition that all parameters remain constant. Based on the factorization in Eq. (2.49), Eq. (2.48) can be rewritten as

$$\begin{aligned} & \iint Q(\mathbf{T})Q(\Theta)(\ln p(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta) - \ln Q(\mathbf{T})Q(\Theta))d\mathbf{T}d\Theta \\ = & \int Q(\mathbf{T}) \ln \frac{p_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)}{Q(\mathbf{T})} d\mathbf{T} - \int Q(\Theta) \ln Q(\Theta) d\Theta \end{aligned} \tag{2.74}$$

where $\ln p_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta) = \mathbb{E}_{\Theta \sim Q(\Theta)}[\ln p(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)]$ and $Q(\Theta) = Q(\mathbf{\Lambda})Q(\boldsymbol{\alpha})Q(\mu_z)Q(\tau_x)Q(\tau_y)$.

The above equation is equivalent to the summation of $-\int Q(\Theta) \ln Q(\Theta) d\Theta$, the negative Kullback-Leibler divergence between $Q(\mathbf{T})$ and $p_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)$ and a constant. Therefore, maximizing Eq. (2.48) with respect to $Q(\mathbf{T})$ is equivalent to minimizing the Kullback-Leibler divergence between $Q(\mathbf{T})$ and $p_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)$. The optimal solution can be obtained from Gibbs' inequality so that $Q(\mathbf{T}) = p_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}, \mathbf{T}, \Theta)$. As a result, the parameters in proxy distribution $Q(\mathbf{T})$ are updated according to Eqs. (2.56) and (2.57). Then, the proxy distributions of PPLS model parameters are updated while keeping $Q(\mathbf{T})$ as constant. Given the large number of model parameters, the proxy distributions corresponding to different parameters are updated one by one. By denoting Θ_j as the particular parameter to be updated and Θ_{-j} as the set of model

parameters excluding Θ_j, Eq. (2.48) can be reformulated as

$$\begin{aligned} & \iiint Q(\Theta_j)Q(\Theta_{-j})Q(\mathbf{T})(\ln p(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta) \\ & - \ln Q(\Theta_j)Q(\Theta_{-j})Q(\mathbf{T}))d\mathbf{T}d\Theta_j d\Theta_{-j} \\ = & \int Q(\Theta_j)\ln\frac{p_{\Theta_j}(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta)}{Q(\Theta_j)}d\Theta_j \\ & - \int Q(\mathbf{T})\ln Q(\mathbf{T})d\mathbf{T} - \int Q(\Theta_{-j})\ln Q(\Theta_{-j})d\Theta_{-j} \end{aligned} \tag{2.75}$$

where

$$\begin{aligned} & p_{\Theta_j}(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta) \\ & = \mathbb{E}_{(\Theta_{-j}\sim Q(\Theta_{-j}),\mathbf{T}\sim Q(\mathbf{T}))}\{\ln p(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta)\} \end{aligned}$$

While performing optimization with respect to $Q(\Theta_j)$, Eq. (2.48) is reorganized as the negative Kullback-Leibler divergence between $Q(\Theta_j)$ and $p_{\Theta_j}(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta)$ and a constant. Thus, maximizing Eq. (2.48) with respect to $Q(\Theta_j)$ is equivalent to minimizing the Kullback-Leibler divergence between $Q(\Theta_j)$ and $p_{\Theta_j}(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta)$. Likewise, the solution can be derived from Gibbs' inequality so that $Q(\Theta_j) = p_{\Theta_j}(\mathbf{X},\mathbf{Y},\mathbf{T},\Theta)$. Consequently, the updating equations for model parameters in proxy distributions in Eqs. (2.58) to (2.67) can be obtained. ∎

Based on Theorem 3, the missing measurements can be re-estimated by the new expectations of latent scores and model parameters derived from the updated proxy distributions. The iterative updates of latent scores, model parameters and missing measurements should terminate if the value of F does not increase any more.

All distributions used in BI-PPLS belong to the exponential family so that the posterior distribution of interest also belongs to the exponential family. In this case, the variational Bayes algorithm is guaranteed to converge to the global optimum since the exponential family is convex (Wang and Titterington, 2004; Bishop, 2006).

2.5 Case study

2.5.1 A numerical example

A simulated example is designed to examine and compare the performance of EM-PPLS and BI-PPLS methods for missing measurement estimation and soft sensor prediction through Monte Carlo simulations. In addition, the proposed methods are compared with the PCA-ALS algorithm, which is a reliable missing measurement estimation method in multivariate statistical modeling (Kroonenberg and De Leeuw, 1980). (In this study, we use the built-in functions in MATLAB when using PCA-ALS algorithms for missing measurement estimation.) An input-output system with random noises is defined as follows

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 5.5 \\ 4.0 \\ 4.5 \\ 4.0 \\ 2.5 \end{bmatrix} + \begin{bmatrix} -0.25 & -0.08 & -0.28 \\ -0.33 & 0.80 & 0.20 \\ -0.45 & 0.40 & 0.60 \\ -0.65 & 0.32 & 0.24 \\ 0.40 & 0.34 & 0.45 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + \mathbf{e}^*$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 3.0 \\ 3.5 \end{bmatrix} + \begin{bmatrix} -0.22 & -0.38 & 0.52 \\ 0.35 & 0.62 & -0.83 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + \mathbf{f}^*$$

where $\mathbf{x}^* = [x_1\ x_2\ x_3\ x_4\ x_5]^T$ are input variables, $\mathbf{y}^* = [y_1\ y_2]^T$ are output variables corresponding to quality attributes, $\mathbf{t}^* = [t_1, t_2, t_3]^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ represent latent scores, $\mathbf{e}^* \sim \mathcal{N}(\mathbf{0}, 0.1^2\mathbf{I})$ and $\mathbf{f}^* \sim \mathcal{N}(\mathbf{0}, 0.1^2\mathbf{I})$ are measurement noises on input and output variables, respectively. In order to build the PPLS model, 500 training samples are generated. Then 10%, 20% and 30% of the data are randomly selected to be the missing

measurements in three different test cases. Further, EM-PPLS and BI-PPLS methods are used to build the model for estimating missing measurements and predicting output quality variables.

In order to evaluate the performance of two types of PPLS methods, the following root mean square error (RMSE) and mean absolute percentage error (MAPE) indices are used

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(z_n - \hat{z}_n)^2} \tag{2.76}$$

$$\text{MAPE} = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{z_n - \hat{z}_n}{z_n}\right| \times 100\% \tag{2.77}$$

where z_n denotes the actual measurement value, $\hat{z}_n$ represents the corresponding estimated value and N is the number of measurements. In addition, the computational programs are run on a computer with Intel® Core™ Duo E8400 processor (3.00 GHz) and 4GB RAM and the CPU times are used to assess the computational efficiency.

Table 2.1: Performance of missing measurement estimation by PCA-ALS method

	10% Missing		20% Missing		30% Missing	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
x_1	0.2674	4.16%	0.2792	4.21%	0.4912	7.02%
x_2	0.5788	14.02%	0.5888	14.21%	0.9132	18.43%
x_3	0.4630	9.21%	0.3514	6.64%	0.6969	12.96%
x_4	0.4804	10.78%	0.5501	11.54%	0.5617	12.24%
x_5	0.1858	11.17%	0.6429	12.49%	0.8920	17.21%
Avg.	0.4767	10.99%	0.4914	11.17%	0.7318	16.12%

Table 2.2: Performance of missing measurement estimation by EM-PPLS method

	10% Missing		20% Missing		30% Missing	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
x_1	0.1511	1.98%	0.1371	1.90%	0.1555	2.13%
x_2	0.1468	3.37%	0.1702	3.46%	0.1851	3.58%
x_3	0.1200	2.16%	0.1420	2.47%	0.1994	3.48%
x_4	0.1477	3.00%	0.1580	3.22%	0.1951	3.83%
x_5	0.1858	3.93%	0.1958	4.04%	0.2773	5.33%
Avg.	0.1517	3.37%	0.1616	3.43%	0.2058	4.30%

Table 2.3: Performance of missing measurement estimation by BI-PPLS method

	10% Missing		20% Missing		30% Missing	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
x_1	0.1506	1.97%	0.1366	1.89%	0.1539	2.09%
x_2	0.1466	3.37%	0.1699	3.45%	0.1836	3.56%
x_3	0.1199	2.15%	0.1411	2.47%	0.1974	3.46%
x_4	0.1473	3.09%	0.1569	3.21%	0.1925	3.76%
x_5	0.1849	3.92%	0.1944	4.01%	0.2753	5.29%
Avg.	0.1512	3.37%	0.1607	3.42%	0.2039	4.26%

2.5.2 Performance comparison

The test results of the simulated example are shown in Tables 2.1 to 2.6. It is obvious that the the PCA-ALS achieves much worse results than the proposed PPLS methods because the non-probabilistic PCA-ALS method does not consider the uncertainty in process measurements. As shown in Fig. 2.2, the reconstructed missing measurements by the PCA-ALS method do not coincide well with the actual measurements. One can easily see that both EM-PPLS and BI-PPLS method can achieve fairly accurate missing measurement estimation and the difference in terms of estimation accuracy between two types of PPLS models is negligible. For instance, the MAPE values of

Table 2.4: Performance of soft sensor prediction by EM-PPLS method

	10% Missing		20% Missing		30% Missing	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
y_1	0.0905	2.59%	0.0944	2.62%	0.1058	2.76%
y_2	0.0821	2.07%	0.1108	2.65%	0.1382	3.04%

Table 2.5: Performance of soft sensor prediction by BI-PPLS method

	10% Missing		20% Missing		30% Missing	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
y_1	0.0906	2.59%	0.0966	2.65%	0.1057	2.74%
y_2	0.0824	2.08%	0.1086	2.57%	0.1332	2.81%

missing measurement estimations for all input variables by BI-PPLS approach are 3.37%, 3.42% and 4.26% for the data sets with 10%, 20% and 30% stochastic missing measurements, while the corresponding MAPE values by EM-PPLS method are 3.37%, 3.43% and 4.30%. Meanwhile, it can be observed that the estimation accuracy of missing measurements in all input variables slightly degrades with the increasing percentage of missing measurements within the data set for both EM-PPLS and BI-PPLS methods. Therefore, both EM-PPLS and BI-PPLS methods are able to estimate significant percentages of stochastic missing measurements robustly. For illustration purpose, the actual and the reconstructed missing measurements by EM-PPLS and

Table 2.6: CPU time (second) of EM-PPLS and BI-PPLS methods

	10% Missing	20% Missing	30% Missing
EM-PPLS	53.26	53.65	54.70
BI-PPLS	12.21	12.36	13.78

BI-PPLS methods are depicted in Figs. 2.3 and 2.4 in the case of 10% missing data.

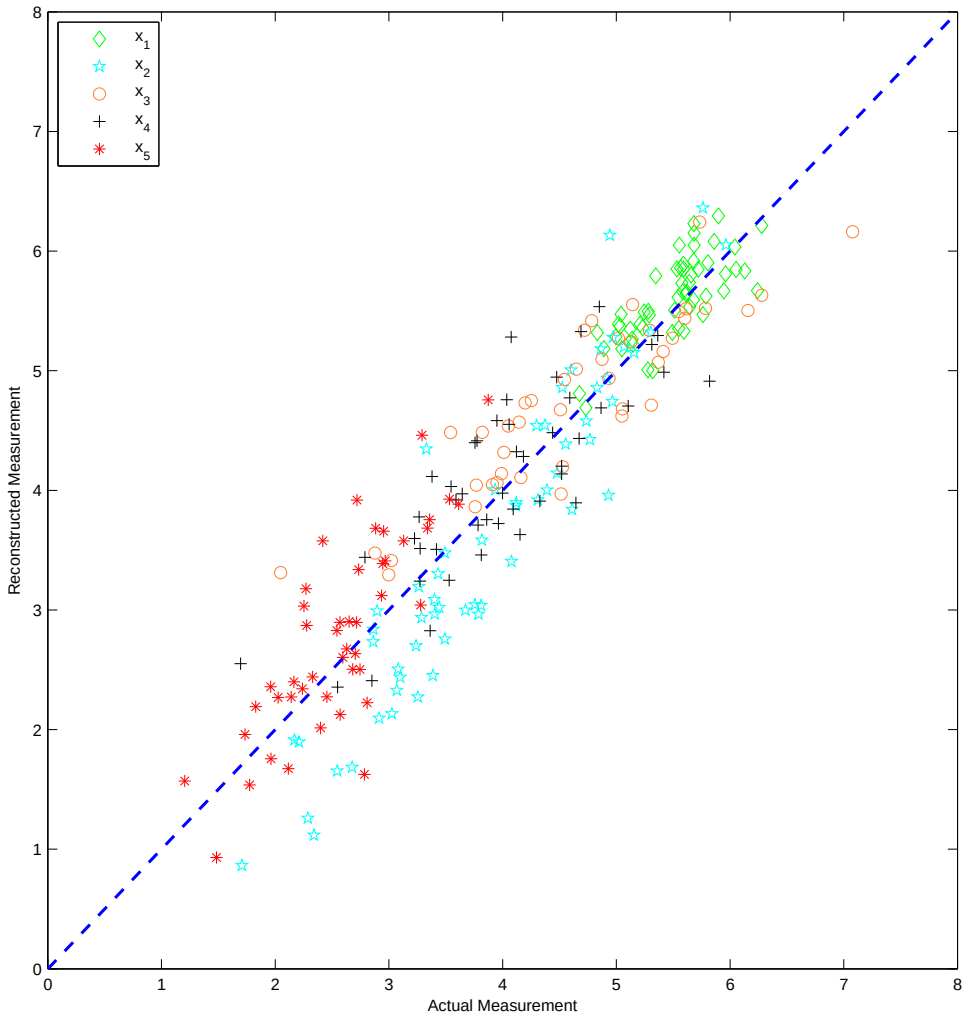

Figure 2.2: Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by PCA-ALS method in the case of 10% missing measurements

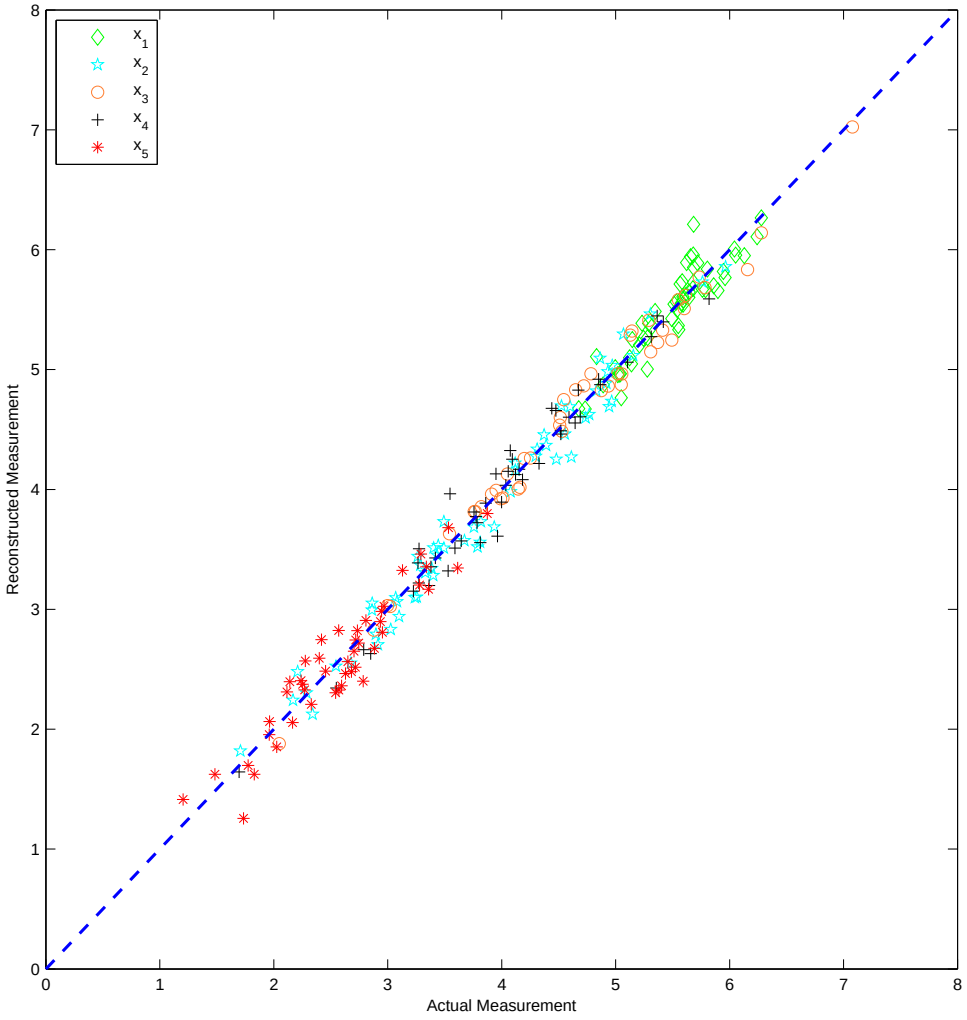

Figure 2.3: Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by EM-PPLS method in the case of 10% missing measurements

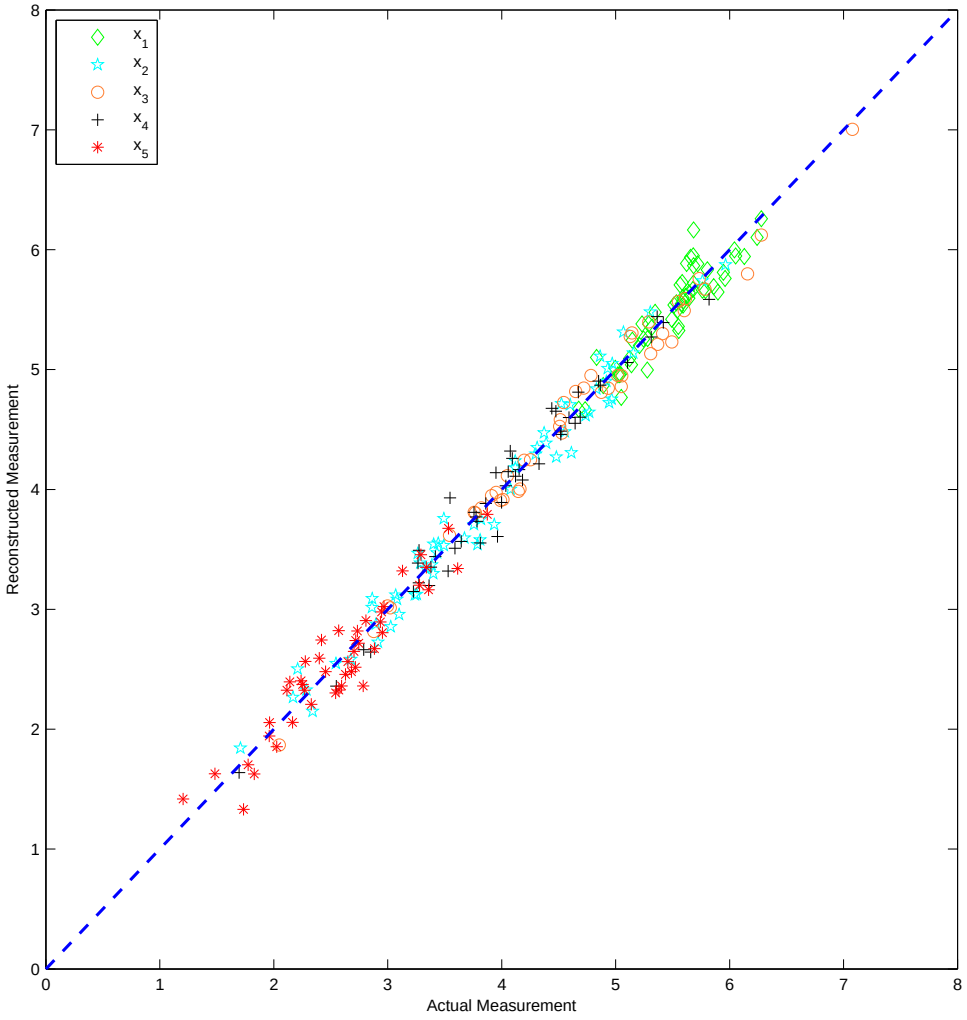

Figure 2.4: Actual input measurements (blue lines) vs. estimated missing measurements (red stars) by BI-PPLS method in the case of 10% missing measurements

The soft sensor model based quality prediction performance is further examined on the EM-PPLS and BI-PPLS methods. As shown in Tables 2.4 and 2.5, high prediction accuracy can be obtained from both PPLS methods on different output variables, although BI-PPLS approach performs slightly better than EM-PPLS method. For instance, the MAPE values of quality variable predictions for y_2 by BI-PPLS approach is 2.08%, 2.57% and 2.81% for test cases containing 10%, 20% and 30% stochastic

missing measurements, while those of EM-PPLS method are 2.07%, 2.65% and 3.04%, respectively. For illustration, the time-series plots of the actual and predicted values of output variables by EM-PPLS and BI-PPLS methods are shown in Figs. 2.5 and 2.6 for the test case with 10% stochastic missing measurements. Similarly as observed on missing measurement estimation, the prediction errors on quality variables do slightly increase with the higher percentage of random missing measurements. Hence, it is demonstrated that both EM-PPLS and BI-PPLS methods can predict quality variables accurately and robustly given significant percentage of missing measurements.

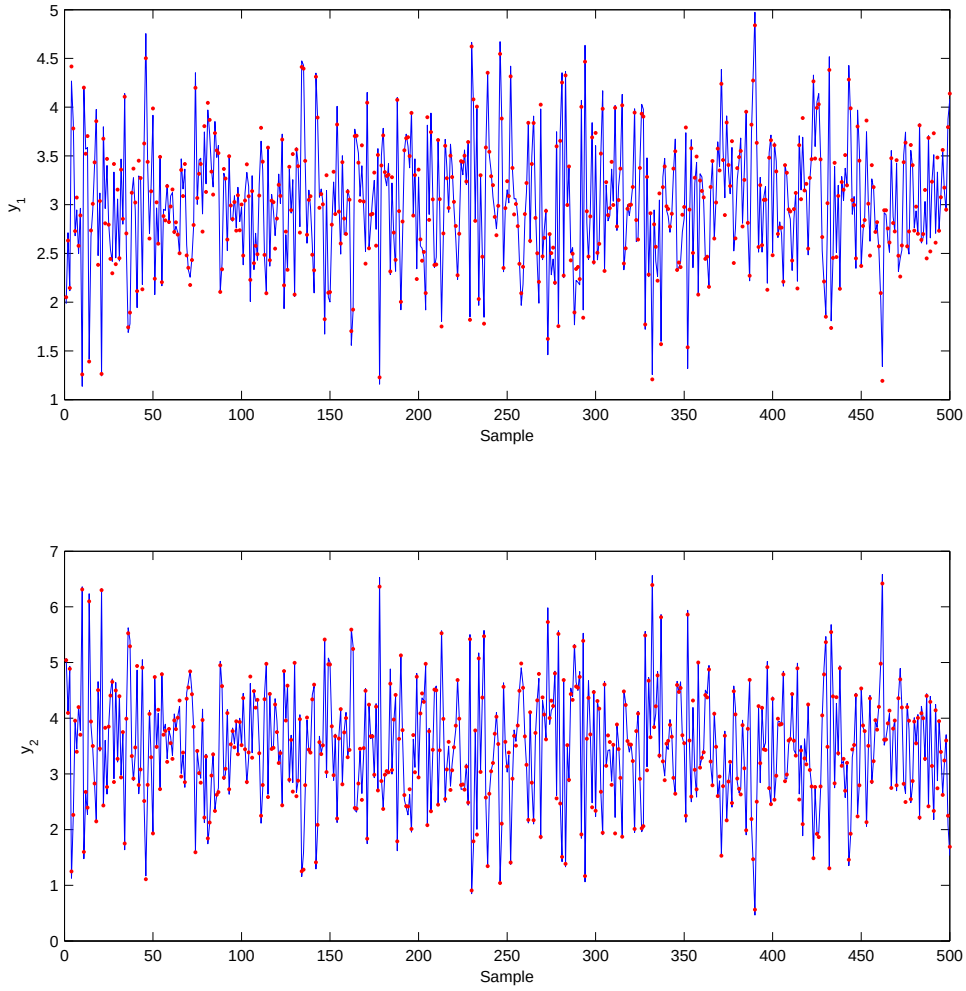

Figure 2.5: Time-series plots of the actual output measurements (blue lines) and predicted values (red dots) by EM-PPLS method in the case of 10% missing measurements

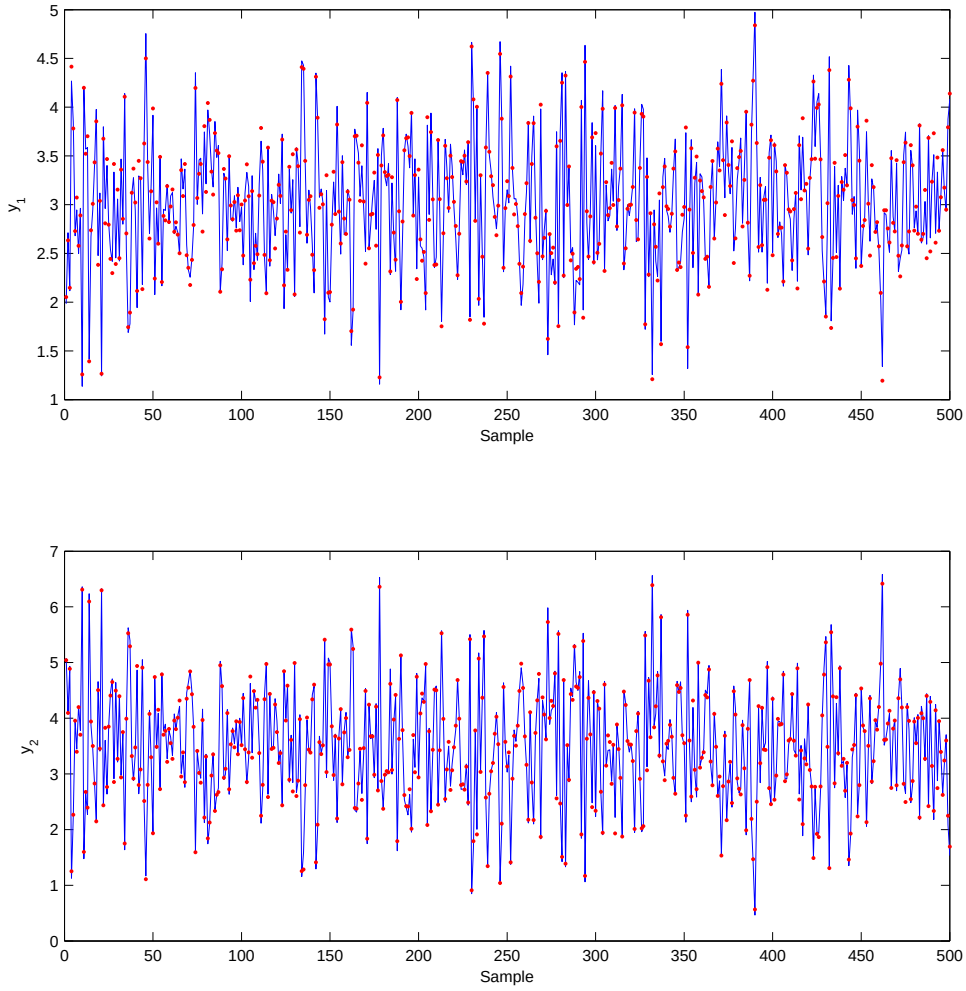

Figure 2.6: Time-series plots of the actual output measurements (blue lines) and predicted values (red dots) by BI-PPLS method in the case of 10% missing measurements

Though BI-PPLS and EM-PPLS approaches result in similar accuracy on missing measurement estimation and quality prediction, BI-PPLS method significantly outperforms EM-PPLS method in terms of computational efficiency. It should be noted that the CPU times for EM-PPLS method as shown in Table 2.6 are the sum of CPU times of cross validations while selecting the dimension of latent subspace. The significantly increased computational time of EM-PPLS method is due to the fact that the optimal

dimension of latent subspace in EM-PPLS model is obtained through cross-validation. In contrast, the latent variable selection in BI-PPLS model is embedded in Bayesian inference strategy with the hierarchial prior distributions and thus the computationally intensive cross validation is not needed in BI-PPLS method. Therefore, BI-PPLS approach is advantageous over EM-PPLS method in terms of computational efficiency, especially for large data set with high dimension.

2.6 Conclusion

In this chapter, expectation-maximization and Bayesian inference based PPLS methods are proposed for soft sensor modeling under stochastic missing measurements. The convergence properties of EM based PPLS method are investigated and proved in explicit forms. In both EM-PPLS and BI-PPLS methods, the random missing measurements are initially imputed by variable means and then iteratively re-estimated from the updated model parameters and latent scores of PPLS model. In order to improve the computational efficiency of selecting optimal latent variables for PPLS model, prior distributions are introduced on model parameters in the presented BI-PPLS method so that the computationally expensive cross validation can be avoided. Both EM-PPLS and BI-PPLS methods are applied to a simulated example with accurate soft sensor prediction and superior performance on missing measurement estimation over the traditional PCA-ALS method. In addition, the comparison of computational time between EM-PPLS and BI-PPLS methods demonstrates the superior performance of Bayesian inference framework for PPLS model in terms of higher computational efficiency. The developed methods can be applied to deal with process data with significant number of missing measurements.

Chapter 3

Data-driven inferential model update in industrial processes[1]

3.1 Introduction

In process industry, accurate and reliable measurement and prediction of quality variables play an important role in process control, monitoring, optimization towards process safety, stability and product quality (Sharmin *et al.*, 2006; Zhang *et al.*, 2010a). Soft sensors are widely used to predict quality variables that are difficult to measure online by using the real-time process data collected by sensors instrumented in processes (Bosca and Fissore, 2011; Yu, 2012c). With the development of soft sensors, the quality variables can be computed by virtual computer programs with the online process measurements instead of hardware analyzers (Fortuna *et al.*, 2007; Lin *et al.*, 2007).

[1]The results in this chapter have been submitted to:

- Chen, K., Yu, J., Castillo, I., and Chiang, L.H. (2014). Soft sensor model update in industrial processes. Submitted to *Chemometrics and Intelligent Laboratory Systems*.

Traditional soft sensors are usually based on first principle models as well as Kalman filter and observers (de Assis and Fiho, 2000; Heineken *et al.*, 2007; Mangold, 2012). Nevertheless, model based soft sensors require in-depth process knowledge and significant effort for model development. Meanwhile, data-driven soft sensors are also developed, which rely on process knowledge and measurement data (Ündey *et al.*, 2003; Dufour *et al.*, 2005; Facco *et al.*, 2009). The well-known multivariate statistical methods such as principal component regression (PCR) and partial least squares (PLS) gain some success in building linear inferential models for state and quality estimations from high-dimensional data with collinearity (Kano and Nakagawa, 2008; Kadlec *et al.*, 2009). Later, machine learning methods including artificial neural networks (ANN) and support vector machines (SVM) are employed for nonlinear state estimations and quality predictions (Gonzaga *et al.*, 2009; Yu, 2012a). The data-driven soft sensors attract increasing attention because the data-driven models are based on real process measurements, leading to more reliable prediction performance. However, the prediction accuracy of soft sensors tends to degrade after a period of their online operation due to process fouling, abrasion of mechanical components, drifted operating conditions, process faults, changes in external environment and others (Kadlec *et al.*, 2011). The degradation of soft sensor models may result in a series of issues in process operation with undesirable quality of final products. Therefore, solutions for degradation of soft sensor models are highly desirable in industrial practice.

Various techniques are developed for online adaptation of soft sensor models to cope with the issue of degradation in soft sensors. Block-wise moving window techniques are employed to update the soft sensor model sequentially by retraining the model periodically when a given number of new data samples are collected, such as fast moving window principal component analysis and moving window kernel principal component analysis (Wang *et al.*, 2005; Liu *et al.*, 2009). More recently, a PLS based

local learning algorithm is developed to construct an adaptive soft sensor model by using the data in a moving window with different process states (Kadlec and Gabrys, 2011). Nevertheless, the effectiveness of moving window based methods is based on the assumptions that the window size and the intervals between updates are set correctly and the process dynamics do not change within the span of one moving window. If the assumptions do not hold, it is very likely that the soft sensors adapt to noise or have very weak adaptation capability. Meanwhile, the recursive partial least squares (RPLS) model is developed by updating the model structure recursively at each sampling instance when the new process and quality measurements are available (Helland *et al.*, 1992; Mu *et al.*, 2006). In addition, recursive adaptation methods for PLS are further modified by using the the new process data in either sample-wise or block-wise manner to update the current soft sensor model in the form of covariance matrices (Dayal and MacGregor, 1997; Qin, 1998). In order to enhance the online prediction capability of soft sensors for nonlinear processes, the adaptive least squares support vector machine (LSSVM) method is developed by recursively updating the kernel matrix of LSSVM (Tang *et al.*, 2010). Compared with moving window techniques, the computational efficiency of recursive adaptation methods is much higher because only covariance or kernel matrix is updated in model adaptation. Nevertheless, choosing an appropriate forgetting factor for previous models is not a trivial task in recursive adaptation methods. Some effort has been made to employ just-in-time learning (JITL) strategy to construct a local model based on a number of nearest neighbors of the test sample (query data) for adaptive predictions (Cheng and Chiu, 2004). The local modeling capability of JITL is improved by selecting local training samples based on correlation between process variables (Fujiwara *et al.*, 2009). However, both distance and correlation based JITL methods cannot model the brand new process dynamics between process and quality variables if corresponding data are not stored in the

database. More recently, soft sensors based on the time difference of process and quality variables are developed to reduce the degradation of soft sensor models caused by process drift without reconstructing the model (Kaneko and Funatsu, 2011b,a, 2013b). Though the time difference models are maintenance-free, choosing appropriate difference intervals to make soft sensors adapt to process drift remains a challenging issue. In addition, time difference models, as well as moving window, recursive adaptation and JITL methods, cannot handle abrupt changes in process dynamics between process and quality variables. Meanwhile, mixture model based soft sensors are developed to model processes running at several substantially different conditions by constructing a local soft sensor model for each operating condition or phase (Yu, 2012d,b; Yu *et al.*, 2013). However, these methods are undesirable in practice because it is very difficult to collect process data from all possible conditions to train the mixture model based soft sensors.

The prediction performance of soft sensor models should be evaluated to check whether degradation of soft sensor models happens or not. Usually, the soft sensor models are recognized as problematic if the prediction errors of quality variables are above a threshold (Kaneko and Funatsu, 2013a). In industrial practice, a PLS soft sensor model will not be updated as long as the model gives accurate predictions for quality variables. Therefore, the adaptive online modeling methods that update soft sensor models sample-wise or block-wise are not desirable in industry. However, it is difficult to obtain a reliable threshold for prediction errors due to the the strong uncertainty in industrial processes. In addition, simply monitoring prediction errors cannot precisely capture some problems in soft sensor predictions. For example, the one-sided offset between actual and predicted quality variables cannot be detected if the offset is within the normal range of prediction errors. As such, some novel indexes should be developed to detect the mismatch between the current soft sensor model and

the process dynamics.

A novel model mismatch index is developed to monitor the performance of PLS soft sensor models in industrial application. Based on virtually estimated regression parameters, the Kalman filter (KF) based model mismatch index is developed and employed to detect the mismatch between the current soft sensor model and process dynamics. In order to avoid biased update for soft sensor models under faulty operations, the decision of model update can be made based on the model mismatch index with the assistance of traditional T^2 and SPE process monitoring indexes. In addition, a contribution plot for the model mismatch index is developed in order to locate the leading abnormal regression coefficients for soft sensor degradation. After updating the model is decided, the regression coefficients can be updated through PLS regression using samples in the training set and the current process conditions. Consequently, the proposed method can handle changes in process dynamics and lead to accurate quality predictions under various process conditions.

The remainder of this chapter is organized as follows. Section 3.2 gives preliminaries about PLS regression. Then the proposed online update method is developed in Section 3.3. The effectiveness of the proposed method is demonstrated in Section 3.4 through simulation of synthetic and industrial examples. Finally, concluding remarks are drawn in Section 3.5.

3.2 Preliminaries

3.2.1 Partial least squares

In this study, the case of a single quality output is considered (PLS1). Given a regressor matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ consisting of N samples with D selected process variables per

sample, and the response matrix $\mathbf{Y} \in \mathbb{R}^{N\times 1}$ of quality outputs, PLS projects $\mathbf{X}$ and $\mathbf{Y}$ onto a lower dimensional subspace defined by a number of latent variables $[\mathbf{t}_1, \ldots, \mathbf{t}_A]$ as follows:

$$\begin{cases} \mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{F} \end{cases} \tag{3.1}$$

where $\mathbf{T} \in \mathbb{R}^{N\times A}$ (A is the the number of latent variables) is the score matrix representing the projections of the variables on the subspace, $\mathbf{P} \in \mathbb{R}^{D\times A}$ represents the loading matrix for $\mathbf{X}$, and $\mathbf{Q} \in \mathbb{R}^{1\times A}$ defines the loading matrix for $\mathbf{Y}$ (Dayal and Macgregor, 1997; Li *et al.*, 2010). $\mathbf{E}$ and $\mathbf{F}$ denote the modeling residuals. Both $\mathbf{X}$ and $\mathbf{Y}$ matrices are usually scaled to zero mean and unit variance. The projection matrices in PLS are calculated in an iterative way by solving the following optimization problem:

$$\begin{aligned} &\max \mathbf{w}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{q}_a \\ &\text{s.t. } ||\mathbf{w}_a|| = 1, ||\mathbf{q}_a|| = 1 \end{aligned}$$

where $\mathbf{w}_a$ and $\mathbf{q}_a$ are loading vectors for $\mathbf{X}_a$ and $\mathbf{Y}_a$, respectively. Denoting $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_A]$, $\mathbf{T}$ cannot be calculated directly from $\mathbf{X}$ using $\mathbf{W}$ because $\mathbf{X}$ is deflated in each iteration. Instead, $\mathbf{T}$ can be computed from $\mathbf{X}$ directly as follows:

$$\mathbf{T} = \mathbf{X}\mathbf{R} \tag{3.2}$$

where $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_A]$. Each column of $\mathbf{R}$ can be computed in a recursive manner as follows:

$$\begin{aligned} &\mathbf{r}_1 = \mathbf{w}_1 \\ &\mathbf{r}_a = \mathbf{w}_a - \mathbf{p}_1^T \mathbf{w}_a \mathbf{r}_1 - \mathbf{p}_2^T \mathbf{w}_a \mathbf{r}_2 - \ldots - \mathbf{p}_{a-1}^T \mathbf{w}_a \mathbf{r}_{a-1}, \quad (a > 1) \end{aligned} \tag{3.3}$$

where $\mathbf{p}_a$ is the column vector in $\mathbf{P}$. Based on Eqs. (3.1) and (3.2), the PLS regression coefficients β_{PLS} between $\mathbf{X}$ and $\mathbf{Y}$ are given by:

$$\beta_{PLS} = \mathbf{R}\mathbf{Q}^T \tag{3.4}$$

In addition, the number of latent variables for PLS regression is usually determined by cross-validation in order to achieve the optimal prediction performance.

3.2.2 Recursive partial least squares

Recursive partial least squares (RPLS) is one of the most commonly used methods for adaptive online process modeling, especially when the process variables are highly correlated (Qin, 1998; Wang *et al.*, 2003). In RPLS, the old data $\mathbf{X}$ and $\mathbf{Y}$ can be discounted by updating the covariance matrices as new data become available

$$(\mathbf{X}^T\mathbf{X})_k = \lambda(\mathbf{X}^T\mathbf{X})_{k-1} + \mathbf{x}_k^T\mathbf{x}_k \tag{3.5}$$

$$(\mathbf{X}^T\mathbf{Y})_k = \lambda(\mathbf{X}^T\mathbf{Y})_{k-1} + \mathbf{x}_k^T\mathbf{y}_k \tag{3.6}$$

where $\mathbf{x}_k$ and $\mathbf{y}_k$ are the new process and quality variables observed at sampling instance k, $(\mathbf{X}^T\mathbf{X})_k$ and $(\mathbf{X}^T\mathbf{Y})_k$ are the updated covariance matrices at time k. λ $(0 < \lambda \leq 1)$ is a forgetting factor that discounts the old data in each sampling instance and $\lambda = 1$ means no discounting of the old data. An improved kernel algorithm is employed to compute the model parameters in RPLS in a very efficient way (Dayal and MacGregor, 1997). First, the covariance matrices are updated by Eqs. (3.5) and (3.6) and let $(\mathbf{X}^T\mathbf{X})_a = (\mathbf{X}^T\mathbf{X})_k$ and $(\mathbf{X}^T\mathbf{Y})_a = (\mathbf{X}^T\mathbf{Y})_k$ when $a = 1$. Then, $\mathbf{w}_a$ can be computed as follows:

$$\mathbf{w}_a = (\mathbf{X}^T\mathbf{Y})_a\mathbf{q}_a \tag{3.7}$$

$$\mathbf{w}_a = \frac{\mathbf{w}_a}{||\mathbf{w}_a||} \tag{3.8}$$

where $\mathbf{q}_a$ is the eigenvector corresponding to the largest eigenvalue of $(\mathbf{Y}^T\mathbf{X}^T\mathbf{X}\mathbf{Y})_a$. Further, $\mathbf{r}_a$ can be computed by:

$$\begin{aligned} &\mathbf{r}_1 = \mathbf{w}_1 \\ &\mathbf{r}_a = \mathbf{w}_a - \mathbf{p}_1^T\mathbf{w}_a\mathbf{r}_1 - \mathbf{p}_2^T\mathbf{w}_a\mathbf{r}_2 - \ldots - \mathbf{p}_{a-1}^T\mathbf{w}_a\mathbf{r}_{a-1}, \quad (a > 1) \end{aligned} \tag{3.9}$$

Compute the loading vectors $\mathbf{p}_a$ and $\mathbf{q}_a$ using $\mathbf{r}_a$ as follows:

$$\mathbf{p}_a^T = \frac{\mathbf{r}_a^T(\mathbf{X}^T\mathbf{X})_a}{\mathbf{r}_a^T(\mathbf{X}^T\mathbf{X})_a\mathbf{r}_a} \tag{3.10}$$

$$\mathbf{q}_a^T = \frac{\mathbf{r}_a^T(\mathbf{X}^T\mathbf{Y})_a}{\mathbf{r}_a^T(\mathbf{X}^T\mathbf{X})_a\mathbf{r}_a} \tag{3.11}$$

Deflate the covariance matrix for the next step decomposition:

$$(\mathbf{X}^T\mathbf{Y})_{a+1} = (\mathbf{X}^T\mathbf{Y})_a - \mathbf{p}_a\mathbf{q}_a^T(\mathbf{r}_a^T(\mathbf{X}^T\mathbf{X})_a\mathbf{r}_a) \tag{3.12}$$

The latent vector computation continues until a reaches the maximum number of latent variables, A. Finally, the regression coefficients for RPLS are given by:

$$\beta_{RPLS} = \mathbf{R}\mathbf{Q}^T \tag{3.13}$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_A]$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_A]$.

3.2.3 Quality relevant process monitoring

One of the advantages of using PLS in industry is that it leads to quality relevant process monitoring. The PLS score $\mathbf{t}_{new}$ and residual $\tilde{\mathbf{x}}_{new}$ for a new sample $\mathbf{x}_{new}$ can be calculated as follows:

$$\mathbf{t}_{new} = \mathbf{R}^T \mathbf{x}_{new} \tag{3.14}$$

$$\tilde{\mathbf{x}}_{new} = (\mathbf{I} - \mathbf{P}\mathbf{R}^T)\mathbf{x}_{new} \tag{3.15}$$

Then, the process monitoring indexes T^2 and SPE are given by:

$$T^2 = \mathbf{t}_{new}^T \Lambda_T \mathbf{t}_{new} \tag{3.16}$$

$$\text{SPE} = ||\tilde{\mathbf{x}}_{new}||^2 \tag{3.17}$$

where Λ_T is the sample covariance of the PLS score (MacGregor *et al.*, 1994). The corresponding control limits for the two process monitoring indexes can also be obtained given the significance levels (MacGregor *et al.*, 1994; Choi and Lee, 2005). A fault is detected if either index exceeds the control limit.

3.3 Online update of the soft sensor model

3.3.1 Detection of mismatch in the soft sensor model by Kalman filter

In literature, the decision for soft sensor model update is based on prediction errors. However, a single abnormally large prediction error does not necessarily indicate mismatch between the current soft sensor model and actual process dynamics because of the uncertainty in processes. Since the process is subjected to noises and unknown

disturbances, the regression parameters may need adjustment to adapt to the current process conditions. As such, the current optimal regression parameters can be treated as hidden states that can be estimated by Kalman filter. If the difference between the filtered and actual regression parameters is within a certain range, the soft sensor model can still characterize the process dynamics well and model update is not necessary. Otherwise, the mismatch between the soft sensor model and current process dynamics is severe and the soft sensor model needs update. Note that the regression parameters estimated by Kalman filter are virtually computed for detection of soft sensor model mismatch and the model updating mechanism of PLS models will be discussed in Section 3.3.3. The state space model for recursive Kalman estimation can be formulated as follows:

$$\beta_k = \mathbf{F}_k \beta_{k-1} + \mathbf{w}_{k-1} \tag{3.18}$$

$$\mathbf{y}_k = \mathbf{H}_k \beta_k + \mathbf{v}_k \tag{3.19}$$

where β_k is the regression parameter at time k, $\mathbf{F}_k$ is the state transition matrix, $\mathbf{w}_k$ is the Gaussian state noise, $\mathbf{y}_k$ is the quality measurement at time k, $\mathbf{H}_k$ is the observation matrix and $\mathbf{v}_k$ is the Gaussian measurement noise. In this study, the regression parameters in the next sampling instance are assumed to be similar with the regression parameters in the last sampling instance in addition with some disturbance so that $\mathbf{F}_k = \mathbf{I}$ in Eq. (3.18). Based on the PLS model, $\mathbf{H}_k$ in the measurement equation is given by $\mathbf{x}_k^T$, where $\mathbf{x}_k$ is the process measurement at time k. With the initial guess for the state being β_o (β_o is the regression coefficients of the current soft sensor model), the Kalman filter can perform a recursive estimation for the state (Rutan, 1990; Chen, 2003). A prior state estimation based on Eq. (3.18) can be expressed as:

$$\hat{\beta}_k = \mathbf{F}_k \beta_{k-1}^{KF} \tag{3.20}$$

where β_{k-1}^{KF} is the filtered regression parameters in the last step. Then the prior covariance $\mathbf{P}_{k|k-1}$ of the state is updated as follows:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{3.21}$$

where $\mathbf{Q}_k$ is the covariance of state noise and $\mathbf{P}_{k-1|k-1}$ is the posterior covariance of the state in the last step. The residual $\tilde{\mathbf{y}}_k$ is computed by:

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\beta}_k \tag{3.22}$$

Then the posterior state is given by:

$$\beta_k^{KF} = \hat{\beta}_k + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{3.23}$$

where $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ is the optimal Kalman gain and $\mathbf{R}_k$ is the covariance of measurement noise. Finally, the posterior covariance $\mathbf{P}_{k|k}$ of the state is updated as follows:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{3.24}$$

Based on the recursive update procedure of Kalman filter, it can be observed that the filtered regression parameters are affected by the quality prediction residual computed in Eq. (3.22). (In this study, the quality measurement $\mathbf{y}_k$ is initially validated through statistical analysis and quality control in lab analysis and assumed to be correct.) If the current soft sensor model can predict the quality variable well, the variation of filtered regression parameters in the next time step will be small. On the other hand, the change of filtered regression parameters in the next time step can be significant based on Eq. (3.23), if the prediction residual of the current soft sensor model is

large. Therefore, the filtered regression parameters tend to be quite different from the actual regression parameters β_o when the soft sensor model cannot characterize the relationship between process and quality variables well. Given the filtered regression parameters β_k^{KF} by Kalman filter at time k, the residual between the estimated and actual regression parameters $\tilde{\beta}_k$ is given by:

$$\tilde{\beta}_k = \beta_k^{KF} - \beta_o \tag{3.25}$$

Then, the model mismatch index is computed as follows:

$$I_k = \tilde{\beta}_k^T \Lambda_{\tilde{\beta}}^{-1} \tilde{\beta}_k \tag{3.26}$$

where $\Lambda_{\tilde{\beta}}$ is the sample covariance of $\tilde{\beta}_k$. With the developed model mismatch index, it is also necessary to obtain a control limit for it in order to decide whether the soft sensor model mismatch is severe or not. The probability density of the model mismatch index can be estimated through kernel density estimation using the computed model mismatch index in the training set as follows:

$$\hat{p}(I) = \frac{1}{Nh} \sum_{k=1}^{N} \kappa(\frac{I - I_k}{h}) \tag{3.27}$$

where $\hat{p}(I)$ denotes the estimated probability density, N is the number of samples in the training set, $h > 0$ is a smoothing parameter called bandwidth and $\kappa(\cdot)$ represents a Gaussian kernel function (Bowman and Azzalini, 1997). For a given significance level, the control limit for the model mismatch index can be obtained using the estimated probability density. In the test set, the model mismatch index can be calculated every sampling instance when a quality measurement is available. Maximum tolerated alarms are predefined to quantify the necessity of model update and an illustrative

diagram is shown in Fig. 3.1. Once the number of model mismatch index values that are consecutively above the control limit reach the maximum tolerated alarms, model update shall be performed to recover the mismatch between the current PLS soft sensor model and actual process dynamics. However, it is recognized that the Kalman filter based model mismatch index alone is not sufficient for making the decision of soft sensor model update because abnormal conditions in the process can also result in performance degradation of the soft sensor model. Therefore, T^2 and SPE indexes are employed to monitor the abnormality in the process. The process is in an abnormal condition if either index exceeds the control limit and the soft sensor model update cannot be performed even in the case that the model mismatch index is above the control limit consecutively. Meanwhile, actions should be taken to address the detected process faults at first. If the model mismatch index is still above the control limit after the process is recovered from the abnormal condition, the soft sensor model will be updated. To summarize, only under the condition that the model mismatch is severe and the process is under normal condition will the soft sensor model be updated.

Remark 1 *Dynamic and recursive modeling approaches are not desirable in some industrial applications because the soft sensor model is updated at each sampling instance or each block of sampling instances without considering the necessity of model update. As long as the soft sensor model provides accurate quality predictions, updating the soft sensor model is unnecessary effort in industry. In addition, it is not safe to update the soft sensor model at each sampling instance or each block of sampling instances because the online measurements from an industrial process usually contain outliers and faulty samples, which may corrupt the model. The most desirable way in industry is to update the model when it needs update because it saves effort and leads to stable model structure. The proposed Kalman filter based model mismatch index is capable of*

monitoring the performance of the soft sensor model and provides guidance about when to update the model.

3.3.2 Diagnosis of soft sensor degradation

After the mismatch between the soft sensor model and the actual process dynamics is detected, the model mismatch index can be further investigated to locate the specific variables that cause the degradation of the soft sensor model. Contribution plots for the model mismatch index can be developed based on the idea that the regression coefficients with the largest contributions to the model mismatch index are most likely the ones leading to soft sensor degradation. It is noticed that the model mismatch index given in Eq. (3.26) has the quadratic form. Therefore, the model mismatch index can be reformulated as follows:

$$\begin{aligned} I_k &= ||\Lambda_{\tilde{\beta}}^{-\frac{1}{2}} \tilde{\beta}_k||^2 \\ &= \sum_{i=1}^{D+1} (\xi_i^T \Lambda_{\tilde{\beta}}^{-\frac{1}{2}} \tilde{\beta}_k)^2 \end{aligned} \tag{3.28}$$

where ξ_i is the i-th column of the identity matrix representing the direction of β_{i-1} (β_0 is the bias term in the regression coefficients $\beta_k = \begin{bmatrix} \beta_0 & \beta_1 & \dots & \beta_D \end{bmatrix}^T$). For example, the direction of β_3 is given by $\xi_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}^T$. As such, the contribution for the model mismatch index is given by:

$$C_i = (\xi_i^T \Lambda_{\tilde{\beta}}^{-\frac{1}{2}} \tilde{\beta}_k)^2 \tag{3.29}$$

The contributions of the model mismatch index help to locate the leading abnormal regression coefficients that cause the degraded performance of the soft sensor model.

3.3.3 Online model update

When it is time to update the soft sensor model based on the criteria described in Section 3.3.1, M process measurements $\mathbf{X}_{new} \in \mathbb{R}^{M \times D}$, and quality measurements $\mathbf{Y}_{new} \in \mathbb{R}^{M \times 1}$ in the new condition are collected and combined with M samples in the training data to update the regression coefficients ($M \ll N$). In order to make sure that the samples from the training data, $\mathbf{X}_{old} \in \mathbb{R}^{M \times D}$ and $\mathbf{Y}_{old} \in \mathbb{R}^{M \times 1}$, represent the distribution of the whole training data set, those samples can be obtained through equidistant sampling of the whole training data set. In parallel, $\mathbf{X}_{new}$ is automatically validated online through the PLS monitoring model while $\mathbf{Y}_{old}$ is validated through the quality control procedure in lab analysis. Given the combined input process data matrix $\mathbf{X}_u = [\mathbf{X}_{old}^T \quad \mathbf{X}_{new}^T]^T$ and output data matrix $\mathbf{Y}_u = [\mathbf{Y}_{old}^T \quad \mathbf{Y}_{new}^T]^T$, the linear regression model in the new process condition can be described as follows:

$$\mathbf{Y}_u = \mathbf{X}_u \beta_u + \mathbf{E}' \tag{3.30}$$

where β_u is the updated regression coefficient vector computed via PLS regression and $\mathbf{E}'$ denotes the prediction residual. Then, the quality will be predicted through the updated soft sensor model.

Remark 2 *It is very common that a linear soft sensor model (e.g. PLS) is employed to predict the quality variables while the actual relationship between process and quality variables is nonlinear. Therefore, the linear model represents an approximation of the actual relationship between* $\mathbf{X}$ *and* $\mathbf{Y}$ *within a certain condition defined by the training data. In reality, it is unlikely to collect samples from all possible scenarios and process conditions to train the soft sensor model, especially if the plant is in the startup stage. Even if the training data are composed of samples from all possible process conditions,*

a single static linear model is not capable of characterizing the quality variable in an essentially nonlinear process. Therefore, online model update is necessary when the process condition shifts and a new linear model is constructed in the new condition. Under some circumstances, nonlinear type PLS models are employed to characterize essentially nonlinear processes and Kalman filter is no longer appropriate. In that case, unscented Kalman filter or particle filter can be potentially utilized to monitor the performance of the nonlinear PLS soft sensor model.

Remark 3 *After updating the soft sensor model is decided, various methods can be employed to calculate the new regression parameters, such as ordinary least squares (OLS), Bayesian regression and time difference methods (Kaneko and Funatsu, 2011a, 2013b). PLS regression method is employed to compute the updated regression coefficients in this study because it also provides the opportunity for updating the weight and loading matrices for quality relevant process monitoring. The original PLS model is not sufficient for process monitoring if the mean and covariance structures of the process and quality variables are changed due to disturbances or set point changes. However, the online update of the PLS monitoring model is another complex issue that is beyond the scope of this article. It is assumed that the set points of the process variables in PLS are not changed in this work. Future research will be extended to co-updating of soft sensor and process monitoring models.*

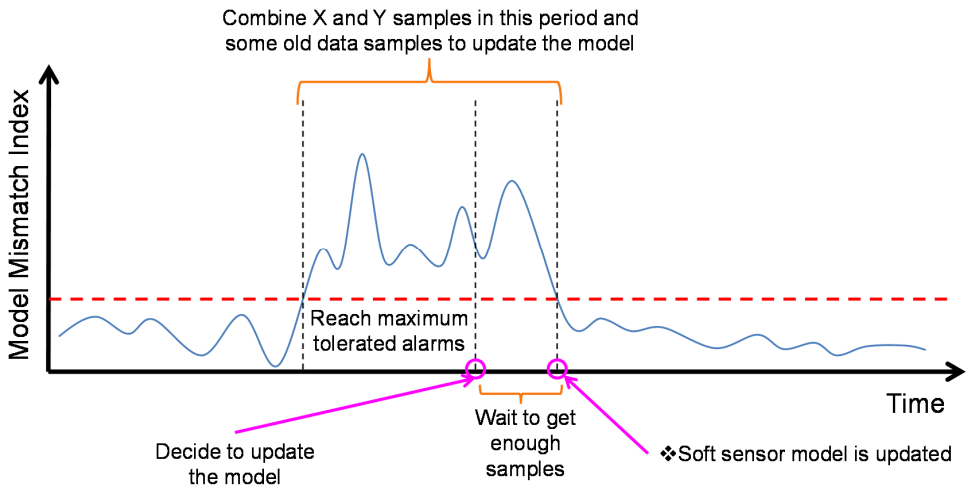

Figure 3.1: Illustrative diagram of detecting severe model mismatch

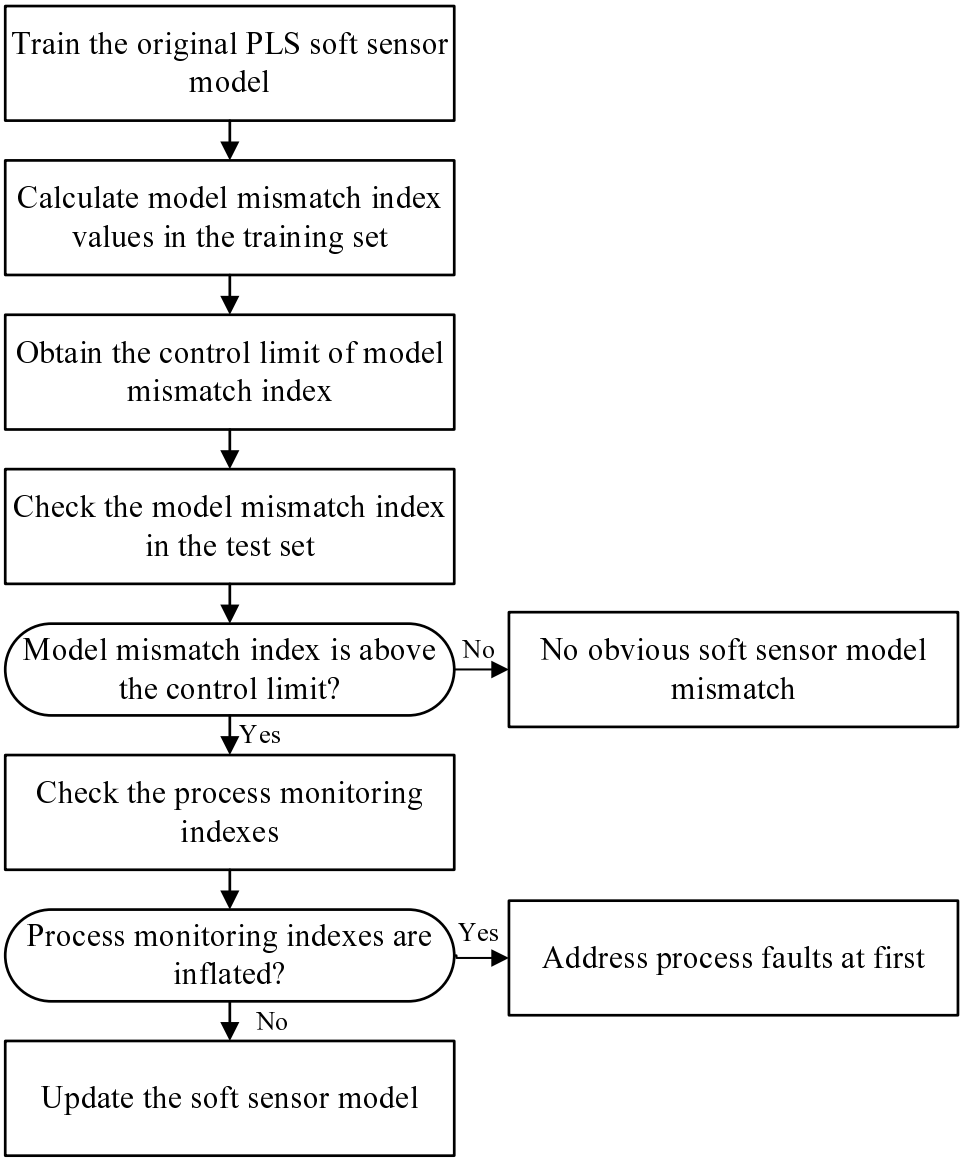

Figure 3.2: Flowchart diagram of the proposed model update method

The step-by-step procedure of the proposed soft sensor model update method is summarized below and the corresponding flow chart diagram is shown in Fig. 3.2.

1. Construct the original PLS model from the offline training data

2. Compute the model mismatch index in the training set and calculate the corresponding control limit;

3. Check the model mismatch index for online samples;

4. If the process monitoring indexes are inflated due to process faults, address process faults at first;

5. If the process monitoring indexes are normal and model mismatch index is above the control limit consecutively, perform model update;

6. Collect a number of process and quality measurements in the new condition and combine them with samples from training data after it is decided to update the model;

7. Compute the updated regression parameters via PLS regression;

8. Obtain quality predictions by using the updated soft sensor model;

9. Employ the updated regression parameters as the base line for future model update.

3.4 Case studies

3.4.1 A simulated example

A simulated system is designed to demonstrate the benefit of the proposed soft sensor model mismatch detection and model update methods. The simulated data are

generated from the following multivariate linear system:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 1 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix} \tag{3.31}$$

$$y = \begin{bmatrix} 1 & 1.5 & -2 & 2.5 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + e_y \tag{3.32}$$

where $s_1 \sim \mathcal{N}(4,1)$, $s_2 \sim \mathcal{N}(3,0.7)$, $s_3 \sim \mathcal{N}(1,0.9)$, $s_4 \sim \mathcal{N}(1,1)$, $s_5 \sim \mathcal{N}(0,0.2)$ and $s_6 \sim \mathcal{N}(0,0.3)$ denote Gaussian distributed data sources, $e_1 - e_6$ are white noises for process variables with deviations of 0.1 and e_y represents the quality measurement noise with the deviation of 1. The objective is to build a soft sensor model to predict the quality variable y by using some of the process variables, x_1, x_2, x_3, x_4, x_5 and x_6. 100 samples are generated based on Eqs. (3.31) and (3.32) for PLS model training. In addition, 600 samples that consist of different process conditions and scenarios are generated for online test of the proposed method. The 1st-200th samples in the online test data are generated based on Eqs. (3.31) and (3.32), which represent the same scenario in the training set. The 201st-400th samples in the online test data are driven

by the process dynamics as follows:

$$y = \begin{bmatrix} 1.1 & 1.4 & -0.5 & 2.6 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + e_y \tag{3.33}$$

where the relationship between process and quality variables is different from that in Eq. (3.32). The last 200 samples in the online test data follow a new relationship given by:

$$y = \begin{bmatrix} 1.2 & 3 & -0.6 & 2.4 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + e_y \tag{3.34}$$

Moreover, an abnormal step change of 9 units happens for x_4 between the 401st and the 420th samples. Meanwhile, the mean of process variables between the 201st to the 600th samples shifts by 0.5 because it is reasonable for the process variables to shift a little when the process condition changes. Note that the magnitude of shift is within the detectability range of the process monitoring model. In addition, only x_1, x_2, x_3 and x_4 are selected in the PLS soft sensor model based on variable importance of projection (VIP) values (Chong and Jun, 2005; Mehmood *et al.*, 2012). Incorporating part of process variables in the soft sensor model is an intentional design to make

this simulated example analogous to an industrial case, because industrial soft sensor models usually do not include all process variables. In this simulated example, the control limit for the model mismatch index is calculated given the significance level of 0.05. In addition, 15 samples from the historical training data and 15 samples from the current process condition are used in the model update.

Three different performance indices are employed to evaluate the prediction accuracy from different aspects, including root mean square error (RMSE), mean absolute percentage error (MAPE) and the coefficient of determination R^2 defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N_t}\sum_{n=1}^{N_t}(y_{tn} - \hat{y}_{tn})^2} \tag{3.35}$$

$$\text{MAPE} = \frac{1}{N_t}\sum_{n=1}^{N_t}\left|\frac{y_{tn} - \hat{y}_{tn}}{y_{tn}}\right| \times 100\% \tag{3.36}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N_t}(y_{tn} - \hat{y}_{tn})^2}{\sum_{n=1}^{N_t}(y_{tn} - \bar{y})^2} \tag{3.37}$$

where N_t is the number of samples, y_{tn} and $\hat{y}_{tn}$ are the actual and predicted values of each quality sample, and $\bar{y}$ is the mean of actual quality measurements.

Table 3.1: Comparison of computational results of the simulated example in the online test

	RMSE	MAPE	R^2
Without Update	8.9104	33.59%	-0.3889
With Update	2.5736	9.62%	0.8842

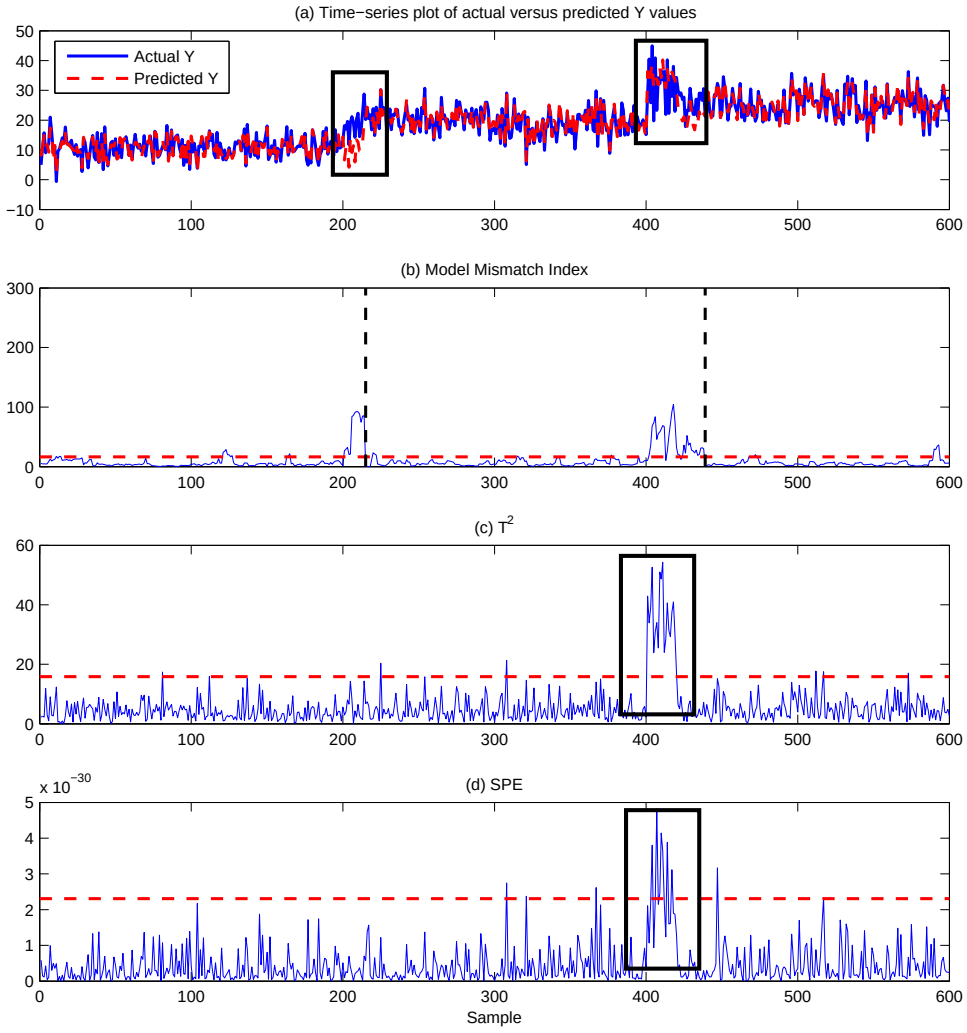

Figure 3.3: Online test of the simulated example

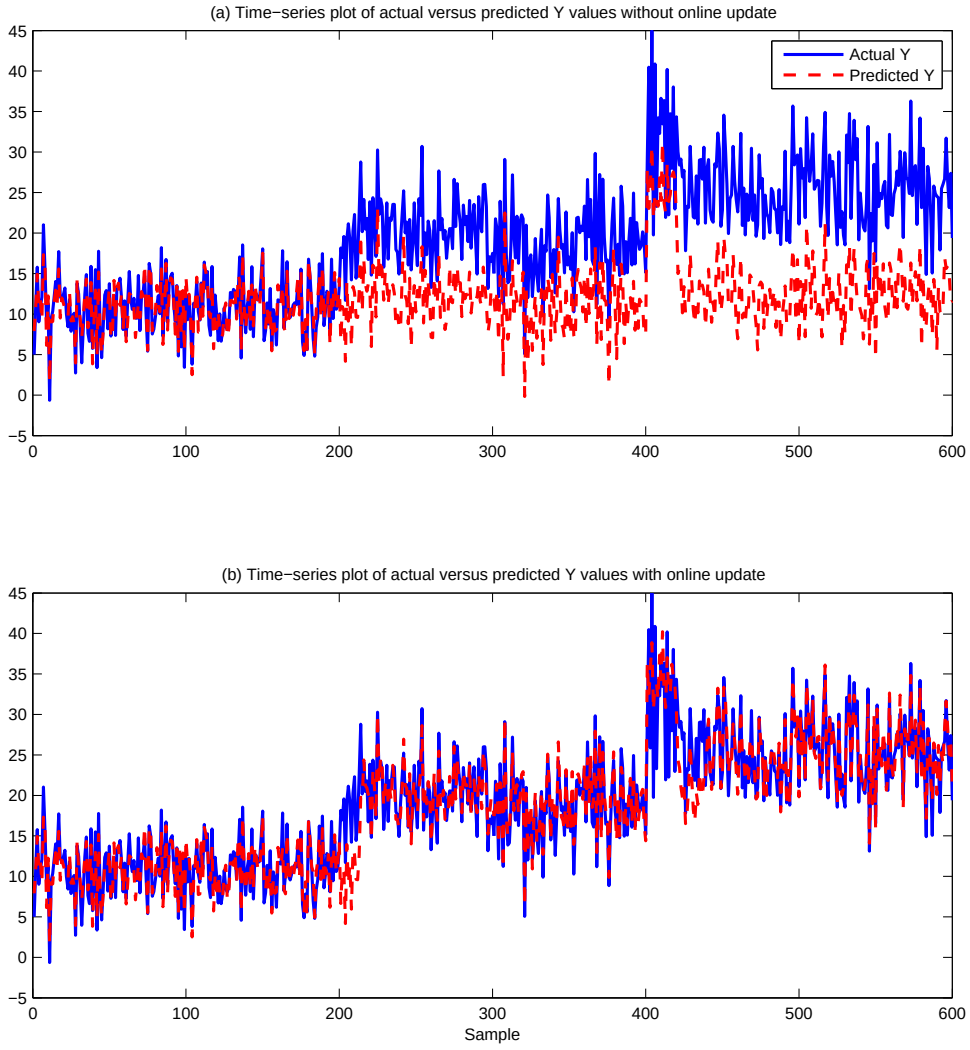

Figure 3.4: Comparison of soft sensor prediction performance for the simulated example

Fig. 3.3 shows that the soft sensor model is updated twice in the online test and the two updates are made after the shifted process dynamics is detected. In Fig. 3.3 (a), the soft sensor cannot predict the actual quality values well between the 201st and the 215th sampling instances since the relationship between process and quality variables is altered at the 201st sampling instance according to the design of the simulated example. Meanwhile, the mismatch between the soft sensor model and the current

process dynamics is captured by the model mismatch index and the index exceeds the control limit, as shown in Fig. 3.3 (b). Then, the soft sensor model is updated based on the criteria described in Section 3.3.3 and the updated model can characterize the quality variable well. Figs. 3.3 (c) and (d) highlight that the fault is captured by T^2 and SPE indexes. In parallel, the prediction performance of the soft sensor model deteriorates again after the 401st sampling instance due to the change in the process dynamics but the model is not updated until the 438th sampling instance in order to avoid adapting the model into abnormal conditions. After the second model update, the PLS model is adapted to accommodate the new conditions in the process and it leads to accurate quality predictions. The quantitative results of quality predictions by the soft sensor models with and without online update are shown in Table 3.1. It is demonstrated that the RMSE and MAPE values of predictions without model update are as high as 8.9104 and 33.59% and R^2 value is as low as -0.3889. The negative R^2 value means that the predictions from the PLS model without online update are biased. In contrast, the predictions of the PLS soft sensor model with online update lead to the satisfactory RMSE value of 2.5726, MAPE value of 9.62% and R^2 value of 0.8842. Correspondingly, the predicted values by the updated soft sensor model can follow the actual trend of the quality variable closely with mitigated prediction residuals while the predictions by the original soft sensor model cannot estimate actual quality values appropriately, as shown in Fig. 3.4. The improved soft sensor prediction accuracy in the new process conditions can be attributed to the accurate detection of the mismatch in the soft sensor model by the Kalman filter.

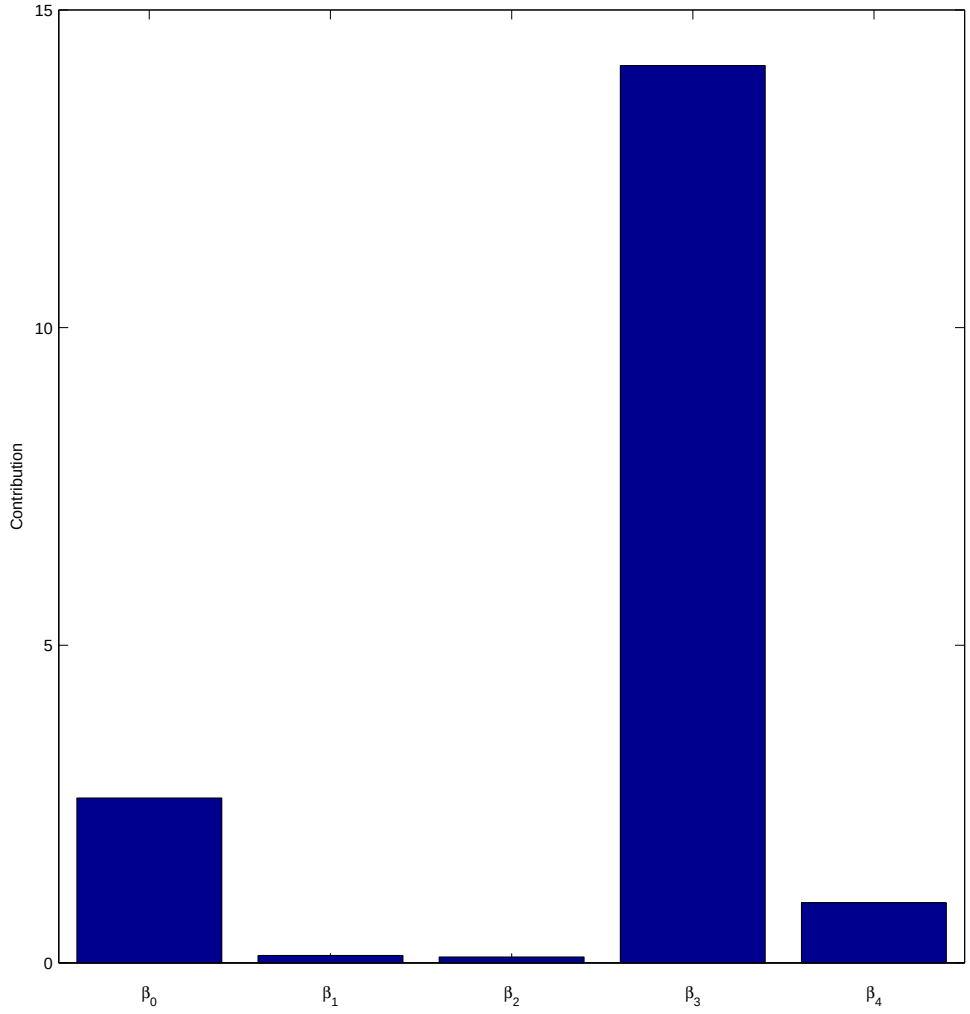

Figure 3.5: Contribution plot of the model mismatch index for the 214th sampling instance in the simulated example

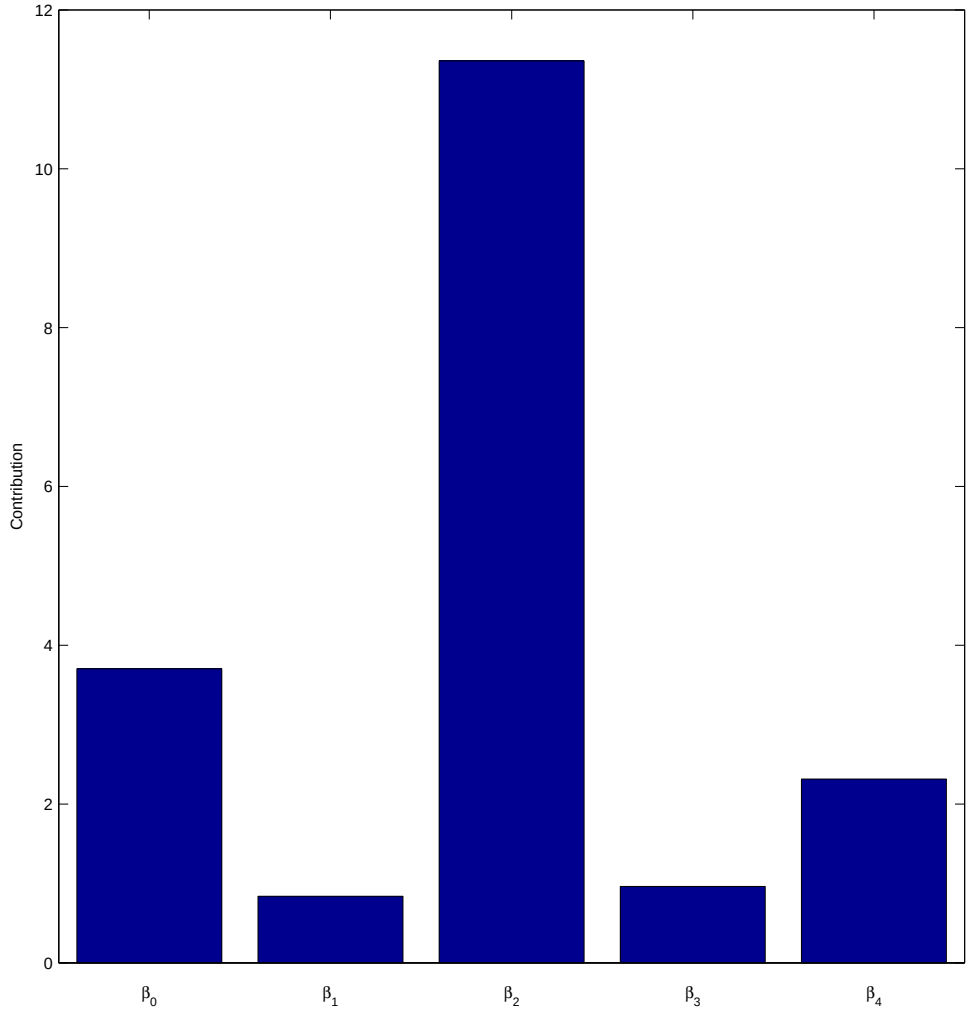

Figure 3.6: Contribution plot of the model mismatch index for the 436th sampling instance in the simulated example

The model mismatch index can be further analyzed to locate the regression coefficients with significant changes. The contribution plot of the model mismatch index for the 214th sampling instance is depicted in Fig. 3.5. Note that the contribution of β_3 is the most significant among all regression coefficients, indicating that the change of the relationship between x_3 and y leads to the degraded performance of the soft sensor model. By comparing Eqs. (3.32) and (3.33), it is obvious that the regression

coefficient for x_3 has the most significant change. Consequently, the contribution plot of the model mismatch index correctly identifies the root cause of the soft sensor degradation before the first model update. In addition, the contribution plot of the model mismatch index for the 436th sampling instance shown in Fig. 3.6 indicates that β_2 is the leading abnormal regression coefficient for the deteriorated prediction performance before the second model update. It can be seen that the regression coefficient for x_2 has the most substantial change among all regression coefficients by comparing Eqs. (3.33) and (3.34). Therefore, the diagnosis results coincide well with the root cause of the soft sensor degradation.

3.4.2 An industrial process

A simplified flow sheet diagram of the industrial process is shown in Fig. 3.7. The industrial process has three unit operations and 61 measurable process variables. Units A, B and C are distillation columns, in which temperatures, pressures, flows and calculated variables are measured and stored. This industrial process has one quality variable, which is associated with the final product of Unit C. The process variables are measured online every hour while the quality variable is measured through off-line laboratory analysis every six hours approximately. The main objective is to build and maintain a reliable soft sensor model by using some of the 61 process measurements to predict the quality variable. In addition, only 4 variables are selected in the soft sensor model based on engineering requirements. Variable selection is necessary in this case because it is found that a relatively reliable soft sensor model can be generated from limited training samples using a small number of key process variables. Considering this is a new plant with limited historical data samples, we employ only 4 process variables in the soft sensor model. (In practice, we found that the soft sensor

model with 4 process variables performs much better than the one with 61 variables.) Given sufficient data samples, the soft sensor model may be built on all 61 variables, which can be investigated in the future work. 735 samples collected from January 2011 to February 2012 are used for training the soft sensor model while another 750 samples collected from February 2012 to May 2013 are employed as online test to demonstrate the effectiveness of the proposed online update method. Note that the raw data contains many missing measurements, which are pretreated by the BI-PPLS method developed in Chapter 2. The process undergoes different operating conditions with various disturbances during the 29 months. The training data does not include all possible scenarios in the online test so that online update for the soft sensor model is necessary. The control limit of the model mismatch index is computed for a given significance level of 0.05. In the model update, 20 samples from the historical training data and 20 samples from the current process condition are employed to update the regression coefficients.

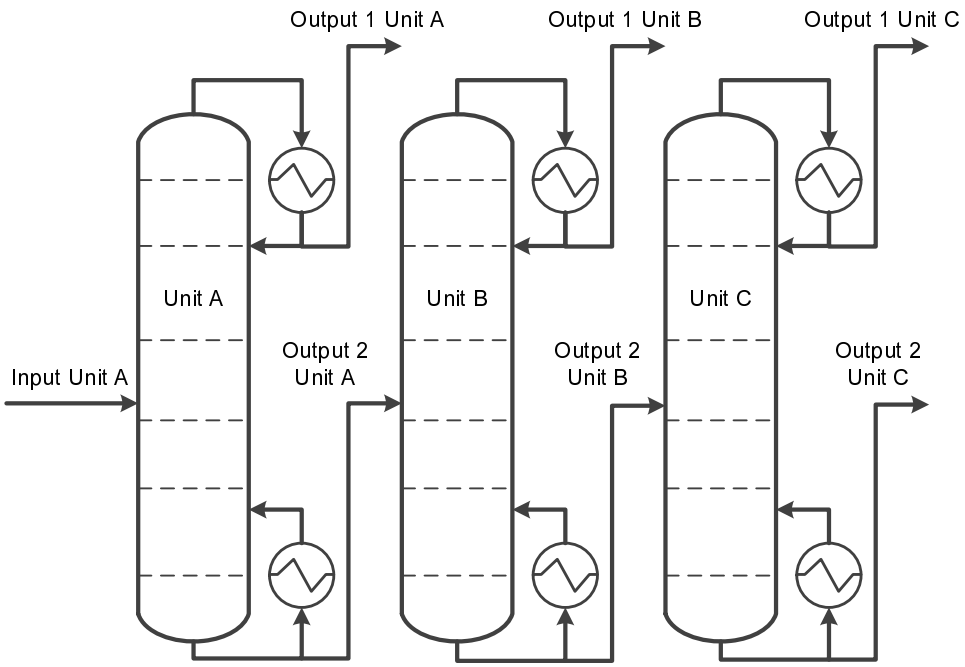

Figure 3.7: Flow sheet of the industrial process

The PLS soft sensor model is built upon Variables 25, 42, 51 and 52 based on engineering knowledge and optimization procedure. As shown in Fig. 3.8, the PLS model is updated 10 times in the online test and the predicted quality values of the updated PLS model follow the actual quality values properly. It can be observed in Fig. 3.8 that an abnormal condition happens between the 140th and the 160th sampling instances, which is indicated by the T^2 and SPE indexes. Though the model mismatch index also exceeds the control limit between the 140th and the 160th sampling instances, the soft sensor model is not updated during that period because it is an abnormal condition. The predictions by the original PLS model and the PLS model with online update are depicted in Fig. 3.9. It is obvious that the quality predictions from the original soft sensor model cannot well characterize the trend in the quality variable, while the predicted values by the soft sensor model with online update follow the actual trend of

the quality variable closely. A strong offset between the actual quality values and the predicted ones by the original soft sensor model from the 1st to the 320th sampling instances is observed, indicating that the soft sensor model is not well suited to predict the quality variable under the current process condition. The prediction accuracy indexes are also listed in Table 3.2, where the RMSE and MAPE values of predictions without model update are as high as 0.7499 and 15.78% and R^2 value is as low as 0.6110. In comparison, the soft sensor with online update leads to more satisfactory prediction results, with the RMSE and MAPE values being 0.6443 and 12.86% and R^2 value being 0.7093. Such comparison demonstrates that the proposed update method can make the soft sensor model adaptive with improved prediction accuracy.

Table 3.2: Comparison of computational results between RPLS and the proposed method in the online test for the industrial process

Method	RMSE	MAPE	R^2
No Update	0.7499	15.78%	0.6110
The proposed method	0.6443	12.86%	0.7093
RPLS ($\lambda = 0.9$)	0.6297	11.57%	0.7220

The cause for the deterioration of the industrial soft sensor model is also analyzed by the contribution plot of the model mismatch index. Though the actual relationship between process and quality variables are unknown, the regression coefficients computed via PLS can be compared to validate the diagnosis results by the contribution plot of the model mismatch index. Fig. 3.10 shows that β_0 and β_2 have the leading contributions for the model mismatch index of the 208th sampling instance. Given that the soft sensor model is updated for the fourth time at the 209th sampling instance, the regression coefficients (for the mean centered and scaled data) before and after the fourth model update are compared. The regression coefficients before the fourth model

update are [0.1882 −0.3459 0.4258 −0.4257 −0.2227], while the regression coefficients after the fourth model update are [0.3172 −0.3435 0.4774 −0.4020 −0.2660]. β_0 and β_2 have obvious changes, which are readily diagnosed by the contribution plot in Fig. 3.10. The change of β_0 can be caused by the shift of the mean in process or quality variables. The shift of the mean is still within a reasonable range since the T^2 and SPE do not exceed their control limits. Moreover, the leading abnormal regression coefficients for the eighth model update are shown in Fig. 3.11 through contributions of the model mismatch index at the 538th sampling instance. The regression coefficients before the eighth model update are [0.0847 −0.3010 0.5086 −0.4299 −0.1089], while the regression coefficients after the eighth model update are [0.0190 −0.2756 0.6069 −0.4435 −0.2420]. Both the contribution plot and the comparison of regression coefficients point to β_0 and β_4, which are the regression coefficients with significant changes.

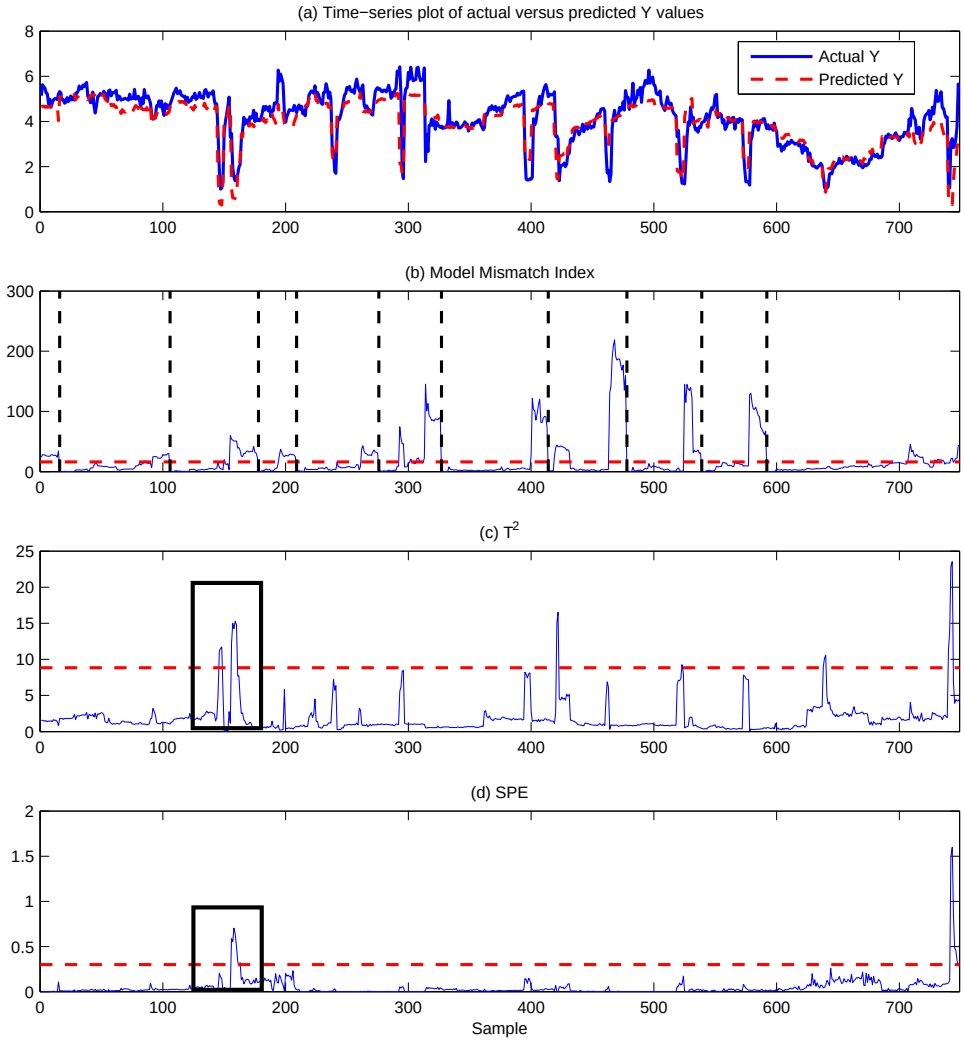

Figure 3.8: Online test for the industrial process

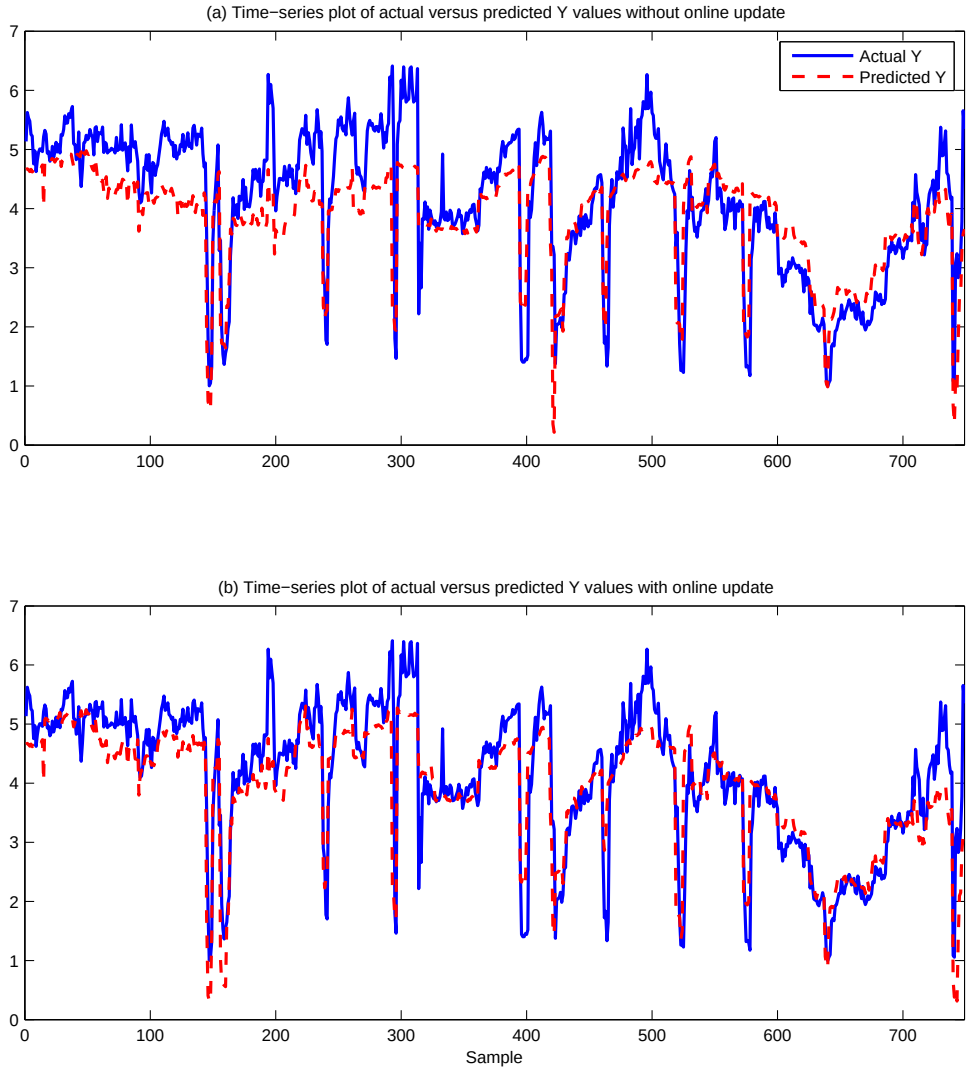

Figure 3.9: Comparison of soft sensor prediction performance for the industrial example

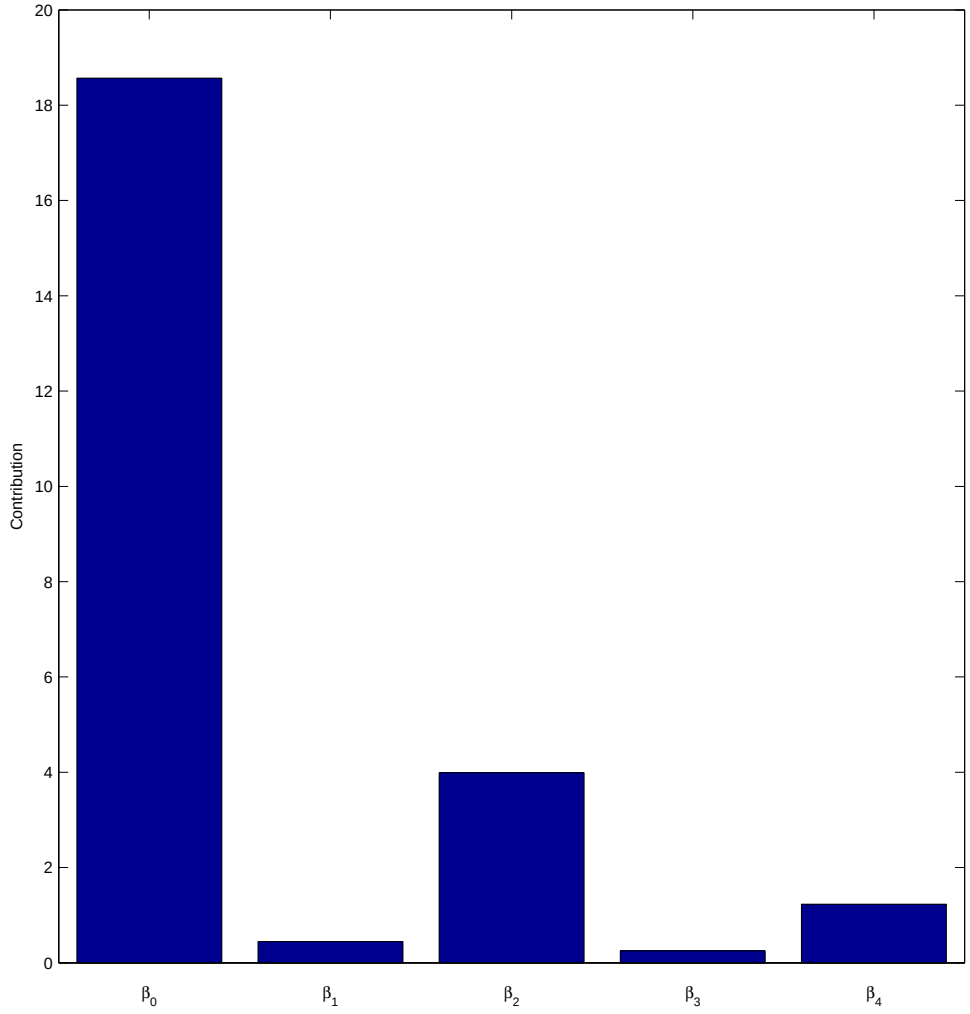

Figure 3.10: Contribution plot of the model mismatch index for the 208th sampling instance in the industrial example

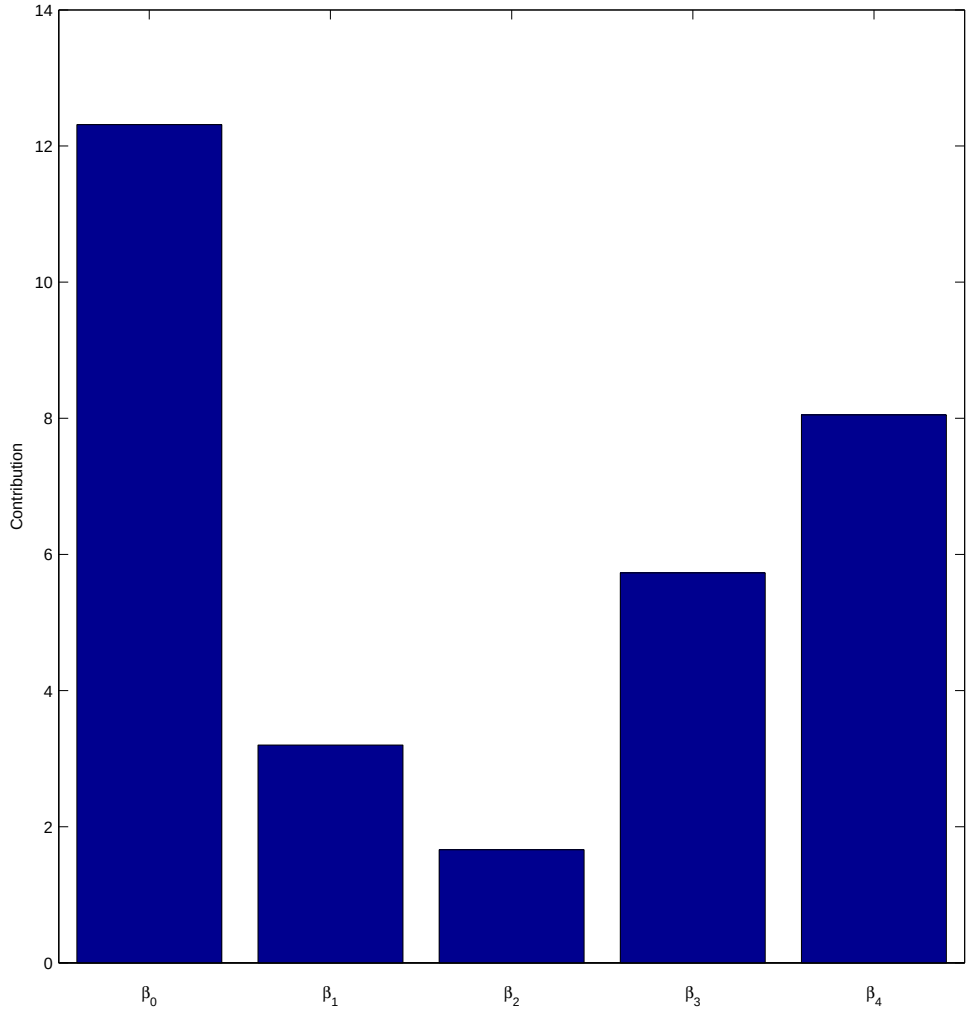

Figure 3.11: Contribution plot of the model mismatch index for the 538th sampling instance in the industrial example

The PLS model with the proposed online update method is compared against the RPLS model with the optimal forgetting factor $\lambda = 0.9$, which is determined through grid search. Although the RPLS model with $\lambda = 0.9$ achieves slightly better performance than the proposed method, the oscillation in its regression coefficients in Fig. 3.12 is strong. The RPLS method updates regression coefficients at each sampling instance, while the majority of updates in RPLS are not necessary. Further, it is not

reasonable for the signs of regression coefficients (besides the bias term) in RPLS to change during the evolution of process. If a process variable is positively correlated to the quality variable based on statistical analysis of historical data, they should not be negatively correlated for several samples in the online operation. Hence, the regression coefficients of RPLS lack consistency and interpretability. In contrast, the signs of regression coefficients in the proposed method never change, which leads to a stable model structure and better interpretability. Consequently, the proposed method is the most desirable in industrial practice.

The soft sensor degradation problem arises due to various reasons, such as process fouling, abrasion of mechanical components, drifted operating conditions, process faults, changes in external environment and others. If the soft sensor degradation problem is caused by those reasons, the prediction accuracy of the data-driven inferential model will decrease slowly and gradually. However, it is observed that the prediction accuracy of the PLS model drops significantly and abruptly sometimes in the industrial process, which means an abrupt change in the relationship between process and quality variables. Further analysis of the data indicates that the abrupt changes are caused by the set point changes in some unselected process variables. Since the industrial process is nonlinear, set point changes of some unselected variables drive the process to different operating regions, which in turn changes the relationship between selected process variables and product quality. The proposed model mismatch index detects the abrupt changes between process and quality variables through recursive Kalman filtering.

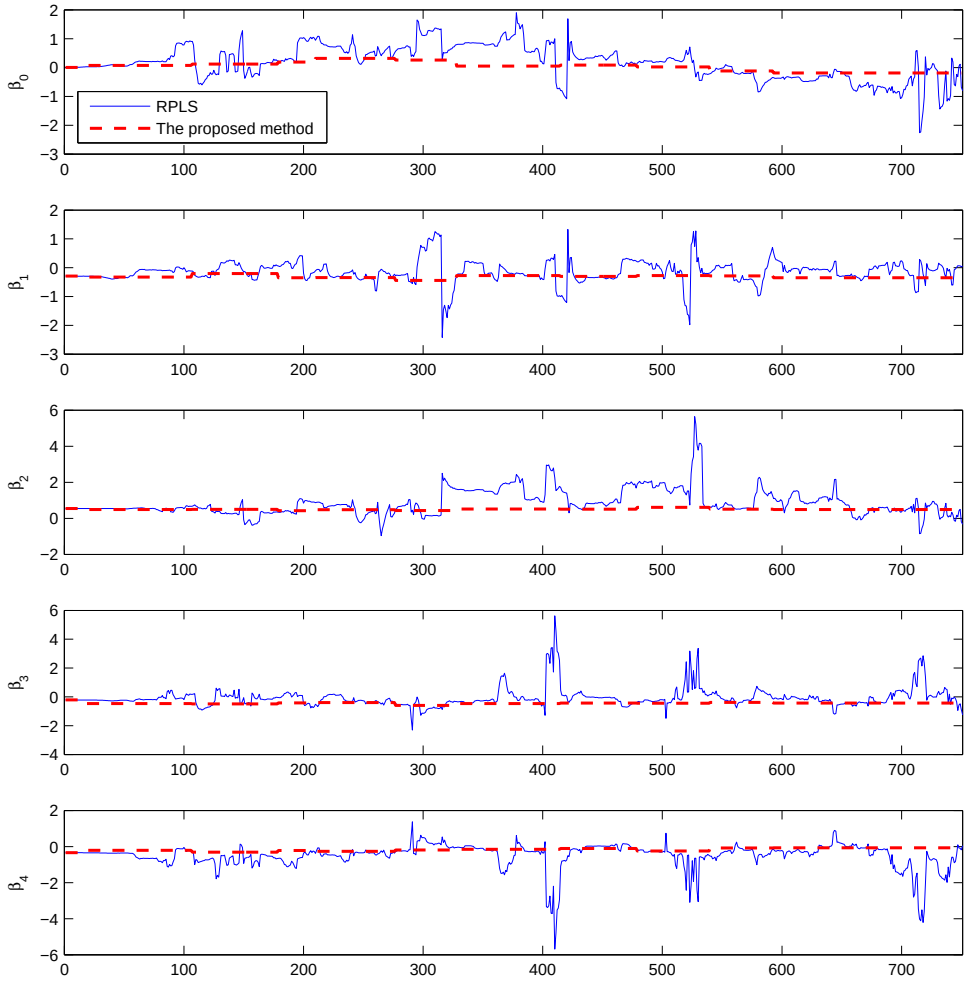

Figure 3.12: Comparison of regression coefficients between the proposed approach and RPLS method

3.5 Conclusions

A Kalman filter based soft sensor model mismatch index and a corresponding contribution plot for diagnosis are developed in this study. Aimed at detecting the soft sensor degradation in industrial processes, the current optimal regression parameters are treated as hidden states and virtually estimated by the Kalman filter in order to

capture the soft senor model mismatch. Based on the statistical properties of the model update index in the training set, the control limit denoting the red line for severe model mismatch for the soft sensor model is obtained through kernel density estimation. Integrated with the process monitoring indexes, the Kalman filter based model mismatch index is employed to determine when to update the soft sensor model. In addition, a contribution plot is also developed to locate the leading abnormal regression coefficients for soft sensor degradation. With the decision to update the soft sensor model, the new regression parameters are computed by using samples in the training set and new conditions. Consequently, the soft sensor model is updated only when significant degradation happens and the prediction performance of the soft sensor model can be improved by the proposed method.

The presented approach is applied to simulated and industrial examples and compared against the RPLS method. The computational results indicate that the proposed index can capture the model mismatch of the soft sensor model in the early stage and the online updated soft sensor model leads to accurate quality predictions. In addition, the comparison with the RPLS method demonstrates that the proposed approach is more desirable in industrial applications.

In industry, the process monitoring model may also be updated online when the mean and covariance structures of process and quality variables are changed. Our current research focus is on online update of the soft sensor model. Future work may extend to co-updating of soft senor and process monitoring models.

Chapter 4

Multiple models based quality control for batch processes[1]

4.1 Introduction

Penicillin production plays an important role in pharmaceuticals and bio-chemicals industries for the production of low-volume but high-value-added secondary metabolites for antibiotics. In order to form the target product, it is common to grow the cells in a batch culture followed by a fed-batch operation to facilitate the synthesis of the antibiotic (Birol *et al.*, 2002). The main advantage of batch operation in Penicillin production is consistently producing products of the desired quality. The Penicillin production, owing to its batch nature, is finite duration process with unique characteristics, such as the absence of equilibrium point as the target point, and nonlinear and time-varying dynamics over a wide range of operating conditions, that preclude

[1]The results in this chapter have been submitted to:

- Chen, K., Corbett, B., Giordano, J., and Mhaskar, P. (2014). Multiple Models based Quality Control of Penicillin Production. Submitted to *Journal of Process Control*.

the direct application of control strategies designed for continuous systems.

In contrast to continuous operation, the primary control objective in batch processes is to reach a specified product quality by batch termination. However, direct control to the specified quality is impractical in most cases because the product quality measurements are unavailable during the batch operation and usually taken through off-line laboratory analysis after the completion of the entire batch. Motivated by the increased demands for consistent production of high-quality products, several control strategies have been developed, including offline batch-to-batch control and online within-batch control.

Batch-to-batch control strategies are designed to bring the final product quality closer to the specified value for the upcoming batch by charging the reactor with a recipe and implementing predetermined input trajectories, which are either optimized offline, determined through data-mining, or historically yielded on-spec product, and include some sort of a correction based on the previous batch. One kind of batch-to-batch control strategy drives the process towards a specified optimum batch-wise by updating the model parameters and then re-computing the batch input trajectories (and/or batch recipe) or directly updating the process variable trajectories using an optimization-based algorithm (Srinivasan *et al.*, 2003a,b; Welz *et al.*, 2008). In parallel, another kind of batch-to-batch strategy exploits the repetitive nature of batch systems by using the error in the final quality from the last batch to update the process variable trajectories and/or initial conditions in the iterative learning control (ILC) framework (Flores-Cerrillo and MacGregor, 2005; Lee *et al.*, 2000; Farasat and Huang, 2012). Batch-to-batch control represents an entirely offline and open-loop operation policy and lacks any real-time feedback mechanism for rejecting disturbances encountered during batch evolution and in the initial conditions, leading to negative impacts on the reproducibility of product quality.

Real-time within-batch control approaches are developed to overcome the shortcomings in batch-to-batch control. Within-batch control approaches can be broadly divided into trajectory tracking and inferential quality control approaches. Trajectory tracking is utilized where the variance in the initial conditions for the batch is very little so that reference trajectories for the measurable process variables can be tracked to reliably meet the specified quality. Tracking is achieved using classical control designs (Cruickshank *et al.*, 2000) or advanced control designs, such as differential geometric (Clarke-Pringle and MacGregor, 1997; Kravaris and Soroush, 1990) or predictive (Shi *et al.*, 2005, 2006; Golshan *et al.*, 2009, 2010; Aumi and Mhaskar, 2009, 2011) controllers, which are capable of compensating for the effects of nonlinearity and tracking set-points over a wide operating range. Since some variables to be tracked cannot be measured online, nonlinear filtering techniques, including unscented Kalman filter and particle filter, are applied to estimate those variables in the online fashion (Shao *et al.*, 2010; Zhao *et al.*, 2013). However, it is not guaranteed to meet to the desired quality even with perfect tracking if there is significant variation in the initial conditions from batch to batch in that disturbances encountered during the new batch could alter the relationship between the product quality and the trajectories of the process variables. Thus, implementing the same reference trajectories batch-to-batch is not guaranteed to consistently produce on-spec product.

More recently, direct quality control for batch processes is developed through quality inferential models, such as principal component analysis (PCA) or partial least squares (PLS) regression (Wold *et al.*, 2002). For batch processes, the model development for the majority of statistical process control (SPC) applications for batch processes begins with the batch-wise unfolding of multiway batch data (Wold *et al.*, 1987). These data-driven modeling methods are particularly well-suited for batch systems because they can exploit existing batch databases and also provide an opportunity to update the

model with the new data after the completion of each batch (Aumi *et al.*, 2013b). In order to identify a reliable data-driven model, the process must be excited in some fashion so that the trajectories of online measurements of the key process variables can contain meaningful information about the process states. In batch process operation, due to the inherent process nonlinearity and moving setpoint, such excitation is often naturally present, allowing a reliable model to be identified.

The unfolded data is regressed onto a matrix of final quality measurements to obtain a multiway PLS (MPLS) inferential model (Flores-Cerrillo and MacGregor, 2003) that is capable of predicting the final quality prior to batch completion. For batches with multiple phases or stages with distinct dynamics, multiple phase-specific (and transition) models can also be constructed (Ündey and Çinar, 2002; Ng and Srinivasan, 2009). During the batch evolution, the final quality can be predicted at every sampling instant or predetermined decision points and appropriate remedial action can be taken to correct the batch if the prediction exceeds the control limits. The corrective action can be obtained systematically through a mid-course correction (MCC) control strategy, which inverts the quality model to compute the future input trajectories that recover the batch (Flores-Cerrillo and MacGregor, 2002). Since it requires model inversion, the effectiveness of an MCC approach is particularly dependent on the quality model with strong generalization capability, and hence demands richer training data that spans a wider operating range and exhibits more input variation compared to modeling for SPC (Wang and Srinivasan, 2009).

An important issue that arises in SPC and MCC approaches is that future measurements that are required to predict the quality are not available. More specifically, the multiway analysis based inferential model calls for the entire batch trajectory to predict the quality of the batch. However, measurements are only available up to the current sampling instant during the batch operation, and the future data required

to predict the final quality is not measured yet, which is treated as a missing data problem. Many methods utilize missing data algorithms available for latent variable methods, which work on the assumption that the correlation structure between the collected measurements and future measurements for the new batch is the same as in the training data (Flores-Cerrillo and MacGregor, 2002, 2004, 2005). However, the causal relationship between the control inputs and the process trajectory are ignored in these latent variable based missing data estimation methods. When the inferential model is used in a control design, the need to consider the nonlinear casual relationship between the future input-output behavior is obvious. Therefore, in recent results, a dynamic model is constructed to predict the future trajectories of process outputs for the MPLS inferential model so that the computed control inputs account for the process dynamics (Aumi *et al.*, 2013a,b).

However, in the Penicillin process, the quality variables are also available intermittently, instead of only at the end, and existing approaches, including those that use a causal model (Aumi *et al.*, 2013a,b) do not utilize the improved information available through intermittent quality measurements. In addition, one global data-driven dynamic process model may not well characterize the essentially nonlinear dynamics in batch processes. Motivated by these considerations, this work addresses the problem of quality control for Penicillin production by developing a framework that utilizes the increased availability of quality measurements. To this end, the whole batch process is divided into several modeling phases based on the available intermediate and final product qualities. Then, each modeling phase is characterized by a localized multiple linear discrete time model, with a corresponding MPLS quality inferential model to predict the final product quality using the process measurement trajectory and intermittent quality measurements. The developed multiple inferential and dynamic models

are integrated with MPC framework so that causality and nonlinear relationships between the future inputs and outputs are accounted for in predicting the final quality and computing the control inputs. The remainder of this chapter is organized as follows. First, the Penicillin fermentation process and the key concepts for single model based quality control are described and reviewed. Next, the details of a multiple models based predictive controller that is designed to drive a batch process to a desired product quality by batch termination are developed. The efficacy of the control design is then demonstrated via simulations of the Penicillin fermentation batch process system. Finally, we summarize our results.

4.2 Preliminaries

In this section, first, the Penicillin fermentation process is described, followed by the introduction for the single model based quality control (Aumi *et al.*, 2013a,b).

4.2.1 Penicillin fermentation process

The Penicillin fermentation process is characterized by nonlinear process dynamics and multi-phase characteristics. The Penicillin fermentation process begins with a cultivation phase to grow the microorganisms and maximize the cell density. The process soon switches to a fed-batch phase to accelerate the synthesis of Penicillin while the growth rate of cell is kept at a slow rate. The glucose and air are continuously fed into the fermenter to supply the raw materials of cell culture and maintain the necessary oxygen consumption for microorganisms. A detailed mechanistic model has been developed (Birol *et al.*, 2002), which describes the state transition dynamics of the Penicillin fermentation process and is described by the following set of ordinary

differential equations:

$$\begin{cases} \frac{dB_c}{dt} = \mu B_c - \frac{B_c}{V}\frac{dV}{dt} \\ \frac{dP_c}{dt} = \mu_{pp} B_c - K P_c - \frac{P_c}{V}\frac{dV}{dt} \\ \frac{dS_c}{dt} = -\frac{\mu}{Y_{b/s}} B_c - \frac{\mu_{pp}}{Y_{p/s}} B_c - m_b B_c + \frac{F s_f}{V} - \frac{S_c}{V}\frac{dV}{dt} \\ \frac{dO_d}{dt} = -\frac{\mu}{Y_{b/o}} B_c - \frac{\mu_{pp}}{Y_{p/o}} B_c - m_o B_c + K_{la}(O_{max} - O_d) - \frac{O_d}{V}\frac{dV}{dt} \\ \frac{dV}{dt} = F + F_{a/b} - F_{loss} \\ \frac{dCO_2}{dt} = \alpha_1 \frac{dB_c}{dt} + \alpha_2 B_c + \alpha_3 \end{cases} \tag{4.1}$$

where B_c, P_c, S_c, O_d, V and CO_2 denote biomass concentration, Penicillin concentration, substrate concentration, dissolve oxygen, fermenter volume, and carbon dioxide concentration, respectively. μ, μ_{pp}, F, $F_{a/b}$, F_{loss} and s_f represent specific growth rate, specific Penicillin production rate, the feed flow rate of glucose, the effect of acid/base on volume, evaporative loss and feed substrate concentration, respectively. In addition, K, $Y_{b/s}$, $Y_{p/s}$, $Y_{b/o}$, $Y_{p/o}$, K_{la}, O_{max}, α_1, α_2 and α_3 are constant parameters. With the glucose feed flow rate, aeration feed rate, agitator power and cooling water flow rate being control inputs, the batch process described in Equation 4.1 can be further summarized as follows

$$\begin{aligned} &\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{u}) \\ &\mathbf{x} = \mathbf{g}(\mathbf{s}, \mathbf{u}) + \mathbf{v} \\ &\mathbf{y} = \mathbf{h}(\mathbf{s}, \mathbf{u}) \\ &t \in [t_0, t_{\mathrm{f}}], \mathbf{u}(\cdot) \in \mathcal{U}, \mathbf{s}(t_0) = \mathbf{s}_0 \end{aligned} \tag{4.2}$$

where the vectors, $\mathbf{s} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^p$, and $\mathbf{y} \in \mathbb{R}^q$ denote the state variables, measurable process variables, and quality variables, respectively. Measurements of $\mathbf{x}$ are assumed to be available at every sampling instant whereas the elements of $\mathbf{y}$ are only measured intermittently during the batch. $t = t_0$ and $t = t_\mathrm{f}$ are the initial and termination times, respectively. The vector $\mathbf{v} \in \mathbb{R}^p$ represents zero-mean, normally distributed measurement noise. The vector $\mathbf{u} \in \mathbb{R}^m$ consists of constrained manipulated inputs, taking values in a nonempty, convex set, $\mathcal{U} \triangleq \{\mathbf{u} \mid \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\} \subset \mathbb{R}^m$ where $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ define the minimum and maximum (respectively) allowable input values.

Table 4.1 lists the state, measurable, manipulated and quality variables (Birol *et al.*, 2002). The states are assumed to be measured only once at the initial sampling instance. All batches are assumed to last 300h, with a sampling interval of 0.5h. The product quality of Penicillin is defined by the concentration of glucose, biomass and Penicillin, which are unmeasured state variables in the process (Birol *et al.*, 2002). In keeping with real process operation, it is assumed that that these quality variables can be measured at 100h, 200h and 300h, which divides the fermentation process into three modeling phases. The control objective in this work is to achieve desired end-of-batch qualities. It should be noted that the proposed approach does not require a model of the form of Equation 4.1 to be available; the form is only used to make the description of the state, measured and input variables clear.

Table 4.1: Variables in the Penicillin fermentation process

State	Measurable	Manipulated	Quality
glucose conc.	jacket temp.	feed flow rate	glucose conc.
dissolved oxygen	dissolved oxygen	aeration flow rate	biomass conc.
biomass conc.	reactor temperature	agitator power	Penicillin conc.
Penicillin conc.	carbon dioxide conc.	cooling water flow rate	
fermenter volume	fermenter volume		
carbon dioxide conc.	feed temp.		

4.2.2 Single Model based Quality Control

A MPLS based inferential quality model is built to predict the end-of-batch quality through multiway analysis. For a typical batch process, the training data of measurement process variables are expressed in the form of three-dimensional matrix $\tilde{\mathbf{X}}$ $(B \times J \times K)$, where B denotes the number of batches, J represents the number of measurement variables and K is the number of sampling instances. First, the three-dimensional matrix $\tilde{\mathbf{X}}$ is transformed to a two dimensional matrix $\mathbf{X}$ $(B \times JK)$ through batch-wise unfolding such that each of its vertical slices is arranged side-by-side (Wold *et al.*, 1987). Next, the initial conditions matrix, $\mathbf{Z}_0$ $(B \times M)$, is concatenated to the unfolded matrix, forming a regressor matrix, $\begin{bmatrix}\mathbf{Z}_0 & \mathbf{X}\end{bmatrix}$. The unfolded matrix can be regressed onto the quality data matrix using PLS regression, yielding a model that relates the initial and process conditions to the quality characteristics as follows

$$\hat{\mathbf{Y}} = \begin{bmatrix}\mathbf{Z}_0 & \mathbf{X}\end{bmatrix} \beta_{PLS} \tag{4.3}$$

where $\hat{\mathbf{Y}}$ is the predicted end-of-batch quality and β_{PLS} is a $(M + JK) \times q$ matrix of the quality model coefficients (Aumi *et al.*, 2013a,b).

Then a data-driven modeling approach is developed to predict the future output behavior for the inferential model (Aumi and Mhaskar, 2011; Aumi *et al.*, 2013b). Mathematically, the model for the process outputs takes the form of a weighted combination of L linear dynamic discrete time models as shown below in Equation 4.5.

$$\hat{\mathbf{x}}(k) = \sum_{\ell=1}^{L} w_\ell(k) \hat{\boldsymbol{\beta}}_\ell \bar{\mathbf{x}}(k) \tag{4.4}$$

$$= \sum_{\ell=1}^{L} w_\ell(k) \hat{\boldsymbol{\beta}}_\ell \left[\mathbf{x}^T(k-1), \cdots, \mathbf{x}^T(k-n_x), \mathbf{u}^T(k-1), \cdots, \mathbf{u}^T(k-n_u)\right]^T \tag{4.5}$$

where $\hat{\mathbf{x}}(k)$ is the predicted measurements at sampling instance k, $\mathbf{x}^T(k-n_x)$ represents the measurements at sample instance $k-n_x$, $\mathbf{u}^T(k-n_u)$ denotes the control inputs at sampling instance $k-n_u$, $w_\ell(k)$ is model ℓ's weight at sampling instant k, $\hat{\boldsymbol{\beta}}_\ell$ defines the ℓ-th local model, and $\bar{\mathbf{x}}(k)$ is a vector of lagged inputs and outputs. The scalars, n_x and n_u, denote the number of lags in the outputs and inputs, respectively. Each model can be identified through least-squares estimation given the specified lag structure (Box *et al.*, 1994).

4.3 Improved Quality Control of Batch Processes

4.3.1 Multiple Quality Inferential Models

In order to utilize intermediate quality measurements and improve the final product quality, multiple quality inferential models are constructed to predict intermediate and final product qualities.

Consider batch of the Penicillin process in which $g = 1, 2, \ldots, G(= 3)$ qualities are measured at sampling instances $k = k_1, \ldots, k_G$ ($k_G = K$). Usually, the number of available quality measurements is much smaller than that of measurable variables ($G \ll K$). Without loss of generality, constructing the quality inferential model for the g-th quality is described here. Firstly, process measurements, including reactor temperature, dissolved oxygen, carbon dioxide concentration and fermenter volume, from sampling instance 1 to k_g are extracted from the three-dimensional matrix $\tilde{\mathbf{X}}$ ($B \times J \times K$) to form a data matrix $\tilde{\mathbf{X}}_g$ ($B \times J \times (K-k_g+1)$), which is further converted to a two-dimensional matrix $\mathbf{X}_g$ ($B \times J(K-k_g+1)$) through batch wise unfolding. Then, the initial quality measurement for the current phase $\mathbf{Y}_g^0$ ($B \times M$) containing the initial conditions of glucose concentration, dissolved oxygen, biomass concentration, Penicillin

concentration, fermenter volume and carbon dioxide concentration, is concatenated to the unfolded matrix to form a regressor matrix $\begin{bmatrix} \mathbf{Y}_g^0 & \mathbf{X}_g \end{bmatrix}$. In addition, the final quality measurements can be stacked in a response matrix $\mathbf{Y}_G$ $(B \times q)$. An illustration diagram for data unfolding and arrangement is shown in Figure 4.1.

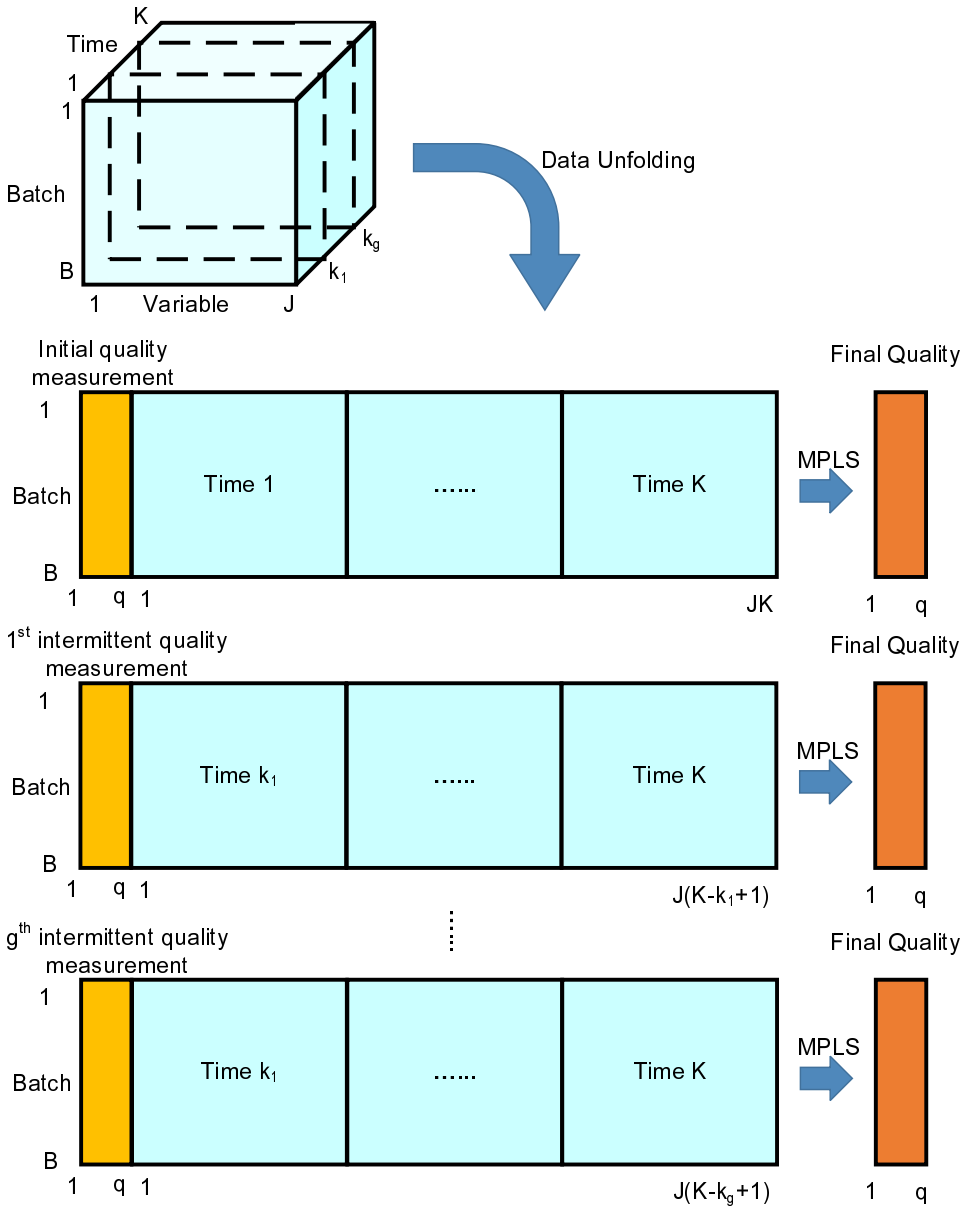

Figure 4.1: Illustrative diagrams for data unfolding and arrangement

Since the number of variables is larger than that of observations in the regressor matrix and variables in the regressor matrix are (auto/cross) correlated, the use of ordinary least squares (OLS) regression leads to parameter estimates with large variances due to covariance of the regressor matrix being ill-conditioned. One way to handle this problem is through partial least squares (PLS) regression. PLS projects the regressor and response matrices, $\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}$ and $\mathbf{Y}_G$, onto a lower dimensional subspace defined by a small number of latent variables $[\mathbf{t}_1, \ldots, \mathbf{t}_{A_g}]$ (A_g is the the number of latent variables) as follows

$$\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix} = \mathbf{T}_g\mathbf{P}_g^T + \mathbf{E}_g \tag{4.6}$$

$$\mathbf{Y}_G = \mathbf{T}_g\mathbf{Q}_g^T + \mathbf{F}_g \tag{4.7}$$

where $\mathbf{T}_g = [\mathbf{t}_1, \ldots, \mathbf{t}_{A_g}]$ is the score matrix representing the projections of the variables on the subspace, $\mathbf{P}_g = [\mathbf{p}_1, \ldots, \mathbf{p}_{A_g}]$ represents the loading matrix for $\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}$, and $\mathbf{Q}_g = [\mathbf{Q}_1, \ldots, \mathbf{Q}_{A_g}]$ defines the loading matrix for $\mathbf{Y}_G$. $\mathbf{E}_g$ and $\mathbf{F}_g$ denote the modeling residual for $\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}$ and $\mathbf{Y}_G$, respectively. In addition, the score matrix $\mathbf{T}_g$ is computed from $\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}$ as follows

$$\mathbf{T}_g = \begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}\mathbf{R}_g \tag{4.8}$$

where $\mathbf{R}_g$ is a weight matrix. Therefore a linear model that relates the initial and process conditions to the quality characteristics can be described as

$$\mathbf{Y}_G = \begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}\beta_{PLS_g} + \mathbf{F}_g \tag{4.9}$$

where $\beta_{PLS_g} = \mathbf{R}_g\mathbf{Q}_g^T$ is the matrix of regression coefficients in the g-th quality inferential model. In PLS algorithms, the subspace orientation and scores for both matrices are determined simultaneously to maximize the correlation between $\begin{bmatrix}\mathbf{Y}_g^0 & \mathbf{X}_g\end{bmatrix}$ and $\mathbf{Y}_G$ through the matrix decompositions and therefore obtain the optimal fit for their relationship.

Remark 4 *For a* new *batch, at sampling instant k, process variable trajectories are only available up to k. More specifically, the process outputs are available up to sampling instant k and the inputs are available up to $k-1$. As a result, the vector required to make the state estimation at batch termination is incomplete. There are ways to eliminate this problem in monitoring applications (e.g., by using multiple models (Wang and Srinivasan, 2009), lookup-tables (Patel* et al.*, 2011), or a different unfolding scheme (Wold* et al.*, 1987)); however, when using the model for* control*, the prediction of the future behavior for a given input is a necessity. Rather than eliminating the need for future data, we recognize the causal nature of the inputs in determining the future trajectory and in turn the quality.*

4.3.2 Localized Multiple Discrete Time Dynamic Models

As mentioned in Remark 4, the prediction of future behavior for a given input is necessary in order to infer the quality. Nevertheless, a single global model may not sufficiently characterize the inherently nonlinear dynamics in the Penicillin fermentation process. The entire batch can be divided in to several modeling phases with the availability of intermediate quality measurements so that localized multiple models can be constructed to predict future trajectories of reactor temperature, dissolved oxygen, carbon dioxide concentration and fermenter volume given control inputs of glucose feed flow rate, aeration feed rate, agitator power and cooling water flow rate within certain

phases. The predicted future trajectories of process variables can be further utilized to forecast the intermediate or final product quality by the corresponding quality inferential model.

Without loss of generality, the localized models to characterize the dynamics of outputs given inputs between sampling instance k_{g-1} and k_g takes the form of weighted combination of L_g linear dynamic models as follows

$$\hat{\mathbf{x}}(k) = \sum_{\ell=1}^{L_g} w_g^{\ell}(k)\hat{\boldsymbol{\beta}}_g^{\ell}\bar{\mathbf{x}}_g(k) \tag{4.10}$$

$$= \sum_{\ell=1}^{L_g} w_g^{\ell}(k)\hat{\boldsymbol{\beta}}_g^{\ell}\left[\mathbf{x}^T(k-1),\cdots,\mathbf{x}^T(k-n_x),\mathbf{u}^T(k-1),\cdots,\mathbf{u}^T(k-n_u)\right]^T \tag{4.11}$$

where $\hat{\mathbf{y}}(k)$ is the predicted measurements at sampling instance k, $\mathbf{x}^T(k-n_x)$ represents the measurements at sample instance $k-n_x$, $\mathbf{u}^T(k-n_u)$ denotes the control inputs at sampling instance $k-n_u$, $w_g^{\ell}(k)$ is model ℓ's weight at sampling instant k, $\hat{\boldsymbol{\beta}}_g^{\ell}$ defines regression coefficients in the ℓ-th local model, and $\bar{\mathbf{x}}_g(k)$ is a vector of lagged inputs and outputs. The scalars, n_y and n_u, denote the number of lags in the outputs and inputs, respectively. Equation 4.11 can be re-written in the vector form

$$\hat{\mathbf{x}}(k) = \hat{\boldsymbol{\beta}}_g\mathbf{h}_g(k) \tag{4.12}$$

by using the following definitions

$$\hat{\boldsymbol{\beta}}_g \triangleq \left[\hat{\boldsymbol{\beta}}_g^1 \quad \cdots \quad \hat{\boldsymbol{\beta}}_g^{\ell} \quad \cdots \quad \hat{\boldsymbol{\beta}}_g^{L_g}\right]$$

$$\mathbf{h}_g(k) \triangleq \left[w_g^1(k)\bar{\mathbf{x}}_g'(k) \quad \cdots \quad w_g^{\ell}(k)\bar{\mathbf{x}}_g'(k) \quad \cdots \quad w_g^{L_g}(k)\bar{\mathbf{x}}_g'(k)\right]'$$

The model identification procedure consists of an initial clustering step followed by

solving a linear regression problem. In the first step, for a given lag structure, a matrix, $\bar{\mathbf{X}}_g$, corresponding to $\bar{\mathbf{x}}_g(k)$, is generated by sorting the plant data sample-wise and then, $\bar{\mathbf{X}}_g$ is clustered into L_g clusters using the fuzzy c-means clustering method (Aumi *et al.*, 2013b). Each cluster represents a region in the $\bar{\mathbf{X}}_g$ space where an associated model has the highest degree of validity, and in this way, the cluster's center point represents the linearization point of its associated model. Using the center points of clusters, the weights $w_g^\ell(\cdot)$ for the training data can be computed prior to the model coefficients. It is intuitive that the model weights $w_\ell^g(\cdot)$ should depend on the current values of the states and inputs since they define the system dynamics through Equation 4.2. In other words, the linear dynamic models should be weighted according to the current process conditions. In the absence of state measurements, the vector of lagged outputs and inputs $\bar{\mathbf{x}}_g(k)$ can be used to infer the current process conditions, and each model's weight can be assigned based on the proximity of the operating conditions to its center point. For instance, denoting model ℓ's center point as $\mathbf{c}_g^\ell$, its weight should be inversely proportional to the squared distance between $\bar{\mathbf{x}}_g(k)$ and $\mathbf{c}_g^\ell$ as follows

$$w_g^\ell(k) \propto \|\bar{\mathbf{x}}_g(k) - \mathbf{c}_g^\ell\|^{-2}$$

Normalizing this expression over all clusters yields the following normalized weighting function

$$w_g^\ell(k) = \frac{\|\bar{\mathbf{x}}_g(k) - \mathbf{c}_g^\ell\|^{-2}}{\sum_{i=1}^{L_g} \|\bar{\mathbf{x}}_g(k) - \mathbf{c}_g^i\|^{-2}}$$

The number of clusters is an important parameter in this approach. Well-defined criteria (based on the cluster geometry) to iteratively refine the number of clusters are available (Frigui and Krishnapuram, 1997). Additionally, to evaluate the goodness of

the final fuzzy partitions, many validation measures have also been proposed with the most popular being the Xie-Beni index (Xie and Beni, 1991), which is a ratio of the total within-cluster variance to the separation of the cluster centers (and therefore should be minimal for the best partition). In this work, we picked the number of clusters (iteratively) based on how well an independent validation data set was predicted. Thus, there was a balancing of the number of clusters and prediction error. Consequently, the $\mathbf{h}_g(k)$ vector in Equation 4.12 is completely specified for the training data. Thus, a regressor matrix corresponding to $\mathbf{h}_g(k)$ can be constructed, and the regression coefficients $\hat{\boldsymbol{\beta}}_g^{\ell}$ in the ℓ-th local linear model in the g-th modeling phase can be computed by using OLS linear regression.

4.3.3 Model Predictive Control

In this section, the localized models developed in the previous subsection is used in conjunction with multiple inferential quality models designing the MPC for the Penicillin fermentation process. The multiple quality inferential models capture the time-cumulative effects of the trajectories of reactor temperature, dissolved oxygen, carbon dioxide concentration and fermenter volume within certain modeling phases on the intermediate and final concentrations of glucose, biomass and Penicillin, while the localized models for the measurable process variables take the causality and nonlinear relationship between the inputs and outputs into account. The benefit from this approach is the ability to account for the direct connection between the control action and intermediate and final product qualities.

With the identified multiple quality inferential models and dynamic process models, the MPC optimization problem is solved to compute the control action with the

objective of achieving a desired final product quality $\mathbf{y}_G^{\text{des}}$ as follows

$$\min_{\mathbf{u}(k)\in\mathcal{U}} \left(\hat{\mathbf{y}}_G - \mathbf{y}_G^{\text{des}}\right)' \mathbf{\Psi} \left(\hat{\mathbf{y}}_G - \mathbf{y}_G^{\text{des}}\right) + \sum_{i=k}^{k_G} \Delta\mathbf{u}'(i)\mathbf{\Phi}\Delta\mathbf{u}(i) \tag{4.13}$$

$$\text{s.t.: } \hat{\mathbf{x}}(k) = \mathbf{x}(t) \tag{4.14}$$

$$\hat{\mathbf{x}}(k) = \sum_{\ell=1}^{L_g} w_g^{\ell}(k)\hat{\boldsymbol{\beta}}_g^{\ell}\bar{\mathbf{x}}_g(k) \tag{4.15}$$

$$\mathbf{x}_{\text{future}} = \begin{bmatrix} \hat{\mathbf{x}}^T(k+1) & \cdots & \hat{\mathbf{x}}^T(k_g-1) & \hat{\mathbf{x}}^T(k_g) \end{bmatrix} \tag{4.16}$$

$$\hat{\mathbf{y}}_G = \begin{bmatrix} \mathbf{y}_g^{0T} & \mathbf{x}_{\text{past}} & \mathbf{x}_{\text{future}} \end{bmatrix} \beta_{PLS_g} \tag{4.17}$$

$$k_{g-1} < k \leq k_g \tag{4.18}$$

In this optimization problem, the objective function consists of a term for minimizing the discrepancy between the target product quality and the predicted quality $\hat{\mathbf{y}}_G$, and a move suppression factor. Each term's relative importance is traded-off using the positive-definite weighting matrices, $\mathbf{\Psi}$ and $\mathbf{\Phi}$. Equation 4.14 is the MPC initialization at the current plant conditions and Equation 4.15 represents the prediction of the future process variables in the current modeling phase using the localized models given the current input trajectory obtained from the optimizer. The predicted process outputs are stored appropriately in the row vector $\mathbf{x}_{\text{future}}$, which is further concatenated with a vector of the initial quality measurement of current modeling phase $\mathbf{y}_g^0$ and previous plant outputs $\mathbf{x}_{\text{past}}$ to predict the quality at batch termination through Equation 4.17. In addition, the specific models to be used at current sampling instance are determined through Equation 4.18.

At sampling instance k, the vector required to predict the quality is incomplete; the inputs and outputs beyond k are unknown. However, when the model is embedded in the MPC optimization problem (wherein the prediction horizon extends to the end of

the current modeling phase), a candidate input trajectory exists in the optimizer that can be used to predict the future outputs up to batch termination. In our work, the future outputs are predicted by the localized models. For a candidate input trajectory, the product quality at the end of current modeling phase can be thus predicted, allowing direct control to a desired quality.

Remark 5 *A distinguishing feature of this MPC design is that it employs an input-output model for predicting the future process outputs given a particular candidate input trajectory. The majority of control designs that have utilized a PLS-based inferential model instead use missing data estimation algorithms, which essentially invert the* linear *PLS model based on the data collected up to the current sampling instant by assuming that the future behavior maintains the same correlation structure as previous batches. This leads to an inherent mismatch in the sense that the correlation in the past batch data that typically uses existing PI controller is different from that in the new batch data which employs a different control algorithm, ignoring the input-output behavior between control inputs and process outputs.*

In contrast, it is recognized in the proposed method that the future trajectories depend on state and input trajectories up to the current point as well as future input moves. Therefore, the only part that needs to be estimated is the the set of future control moves, which can be computed by the controller. By utilizing an appropriate input-output model that links the future inputs to the future process outputs and in turn to the quality, the controller then computes the set of input moves that would yield the desired quality. The problem of unequal batch lengths can also be readily handled by the proposed quality prediction approach by virtue of using a dedicated (nonlinear) model for predicting the future batch behavior. In particular, all the batch data can be used to build the dynamic model, while the quality model can be built using a common time

from the end of batch for all batches (i.e., using the batch time of the shortest batch, if not significantly shorter than the other batches).

Remark 6 *Other control designs evolve quality models at each sampling instant or a selected number of pre-determined decision points by utilizing measurements only up to a given time in order to eliminate the unavailable future measurement problem completely. These models are designed to forecast the quality without the future batch trajectories and inherently rely on the assumption that the same control action is implemented for the rest of the batch. As such, they are not well suited for use in a new control design aimed at computing the control action to yield the desired quality though such methods may be good to predict the quality under an existing controller.*

Remark 7 *The proposed framework addresses the problem where quality measurements are available but with significantly less frequency than the process measurements. This warrants (and make it possible) to build as many quality models as the quality measurements. The key idea behind building the quality models is as follows- It is well understood that quality variables can only be functions of the system states. A model that estimates quality from process variable trajectories has to essentially estimate the states from the process measurements trajectory and also determine the relationship between the states and the quality variables. In building multiple quality models, as more quality measurements become available, they serve as corrections to the state estimation and quality model. In the case where the frequency of quality measurements is significantly close (or equal) to the process measurements, one would need to employ a different approach. In particular, the quality measurements would have to concatenated with the other process measurements in the dynamic model, with the number of dynamic models dictated by the prediction ability. In other words, one could completely do away with separate quality models, and include the quality prediction as part of the dynamics*

model- of course that would likely alter the number of lags and principal components needed to be retained in the dynamic models to fulfil the state estimation and quality modeling requirements.

Remark 8 *Since it is realized that quality variables can only be functions of the process states, the latent scores in Equation 4.7 must be some linear combinations of process states. As such, Equation 4.6 can be interpreted as estimating states from process measurement trajectories, and the loading matrix in Equation 4.6 should also contain useful information about the process dynamics. Though outside the scope of the present work, the state estimation and quality prediction can also be dealt with separately. First, the process state and dynamics can be identified through projection of past system outputs and control inputs into a latent space. Then, a (linear or nonlinear) model, or known dependence of the quality on system states can be utilized to predict and control quality variables.*

4.4 Multiple Models based Quality Control of Penicillin Production

In this section, we demonstrate the efficacy of the proposed MPC design through closed-loop simulations of the Penicillin fermentation process. First, we develop multiple data-driven models for the measurable process variables and product quality for the Penicillin fermentation process. Finally, the proposed MPC design is implemented with those developed models and compared against trajectory tracking PI control and single quality model based MPC.

4.4.1 Inferential and Process Models

To develop data-driven models for the quality and process variables, a batch database of the form in Figure 4.1 is first generated. To this end, the deterministic model is simulated 60 times from different initial conditions with random measurement noises (10 batches are reserved as the validation data set). In generating the training data, a set of reference profiles of fermenter volume, dissolved oxygen, carbon dioxide and jacket temperature are tracked using feed flow rate, aeration flow rate, agitator power and cooling water flow rate (respectively) via three tightly tuned PI controllers. The manipulated variables will move in order to compensate the difference between the actual and nominal measurable variables caused by various initial conditions and measurement noises.

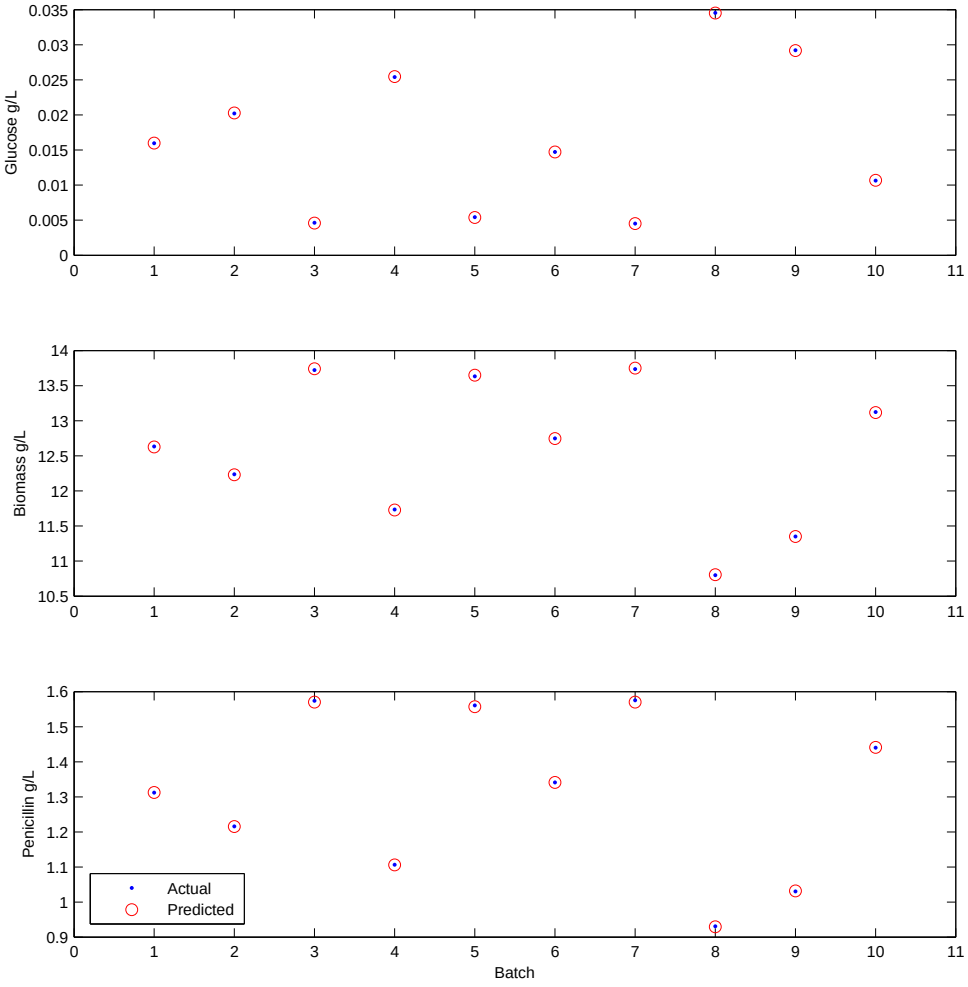

Figure 4.2: Prediction of quality variables of the 1st modeling phase

The training database consisted of measurements of the states at the initial time, measurable variables at every sampling instant, and the qualities at 100h, 200h and 300h. Among all the measurable variables, reactor temperature, dissolved oxygen, carbon dioxide concentration and fermenter volume are employed to predict the quality variables, glucose, biomass and Penicillin concentrations based on engineering knowledge. Three quality inferential models are constructed to predict the intermediate and final product qualities through multiway PLS regression. The numbers of latent

variables in those three quality inferential models are 25, 18 and 14, respectively, determined by cross-validation to minimize the root mean square error (RMSE). In Figures 4.2, 4.3 and 4.4, the qualities predicted by the inferential models for the 10 validation batches are displayed along with the actual qualities. Note that the total number of columns in the regressor matrices in three quality inferential models range from 811 to 2411; thus, a small number of latent variables represent a significant reduction in the dimensionality of the process. Overall, the inferential quality models predict the final qualities with high fidelity.

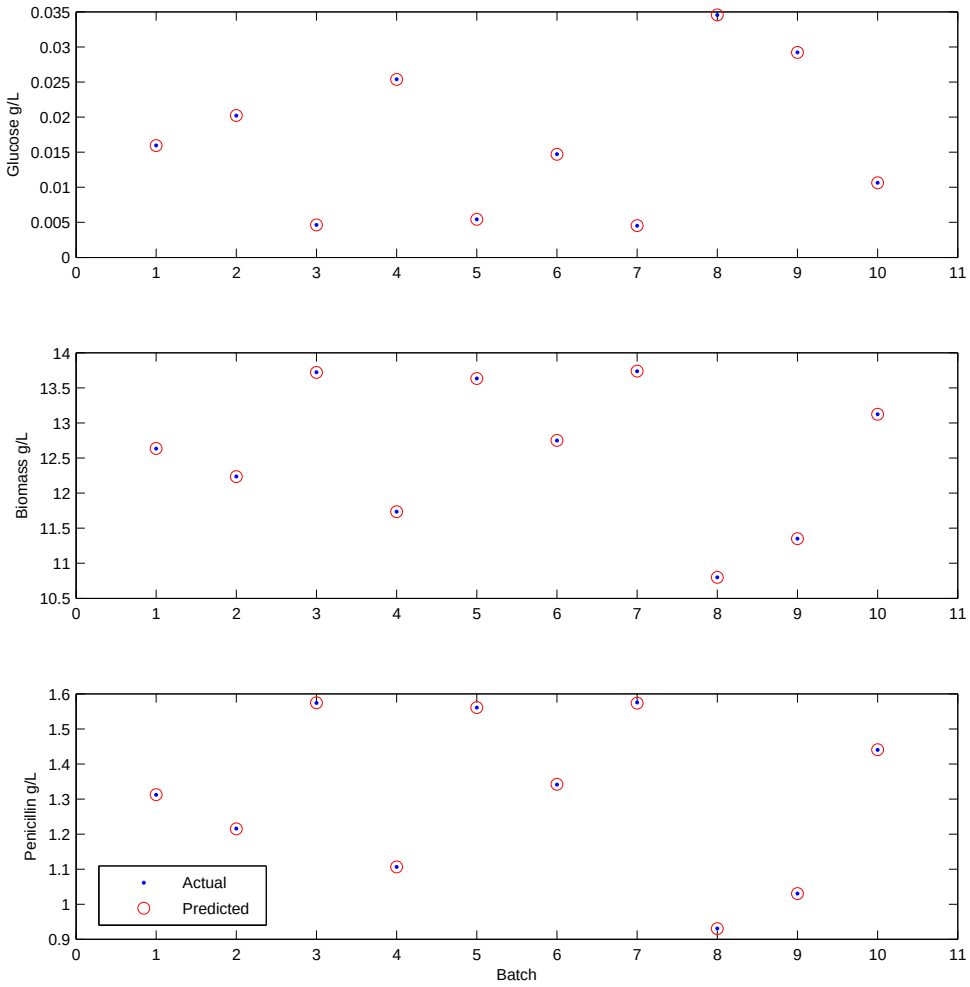

Figure 4.3: Prediction of quality variables of the 2nd modeling phase

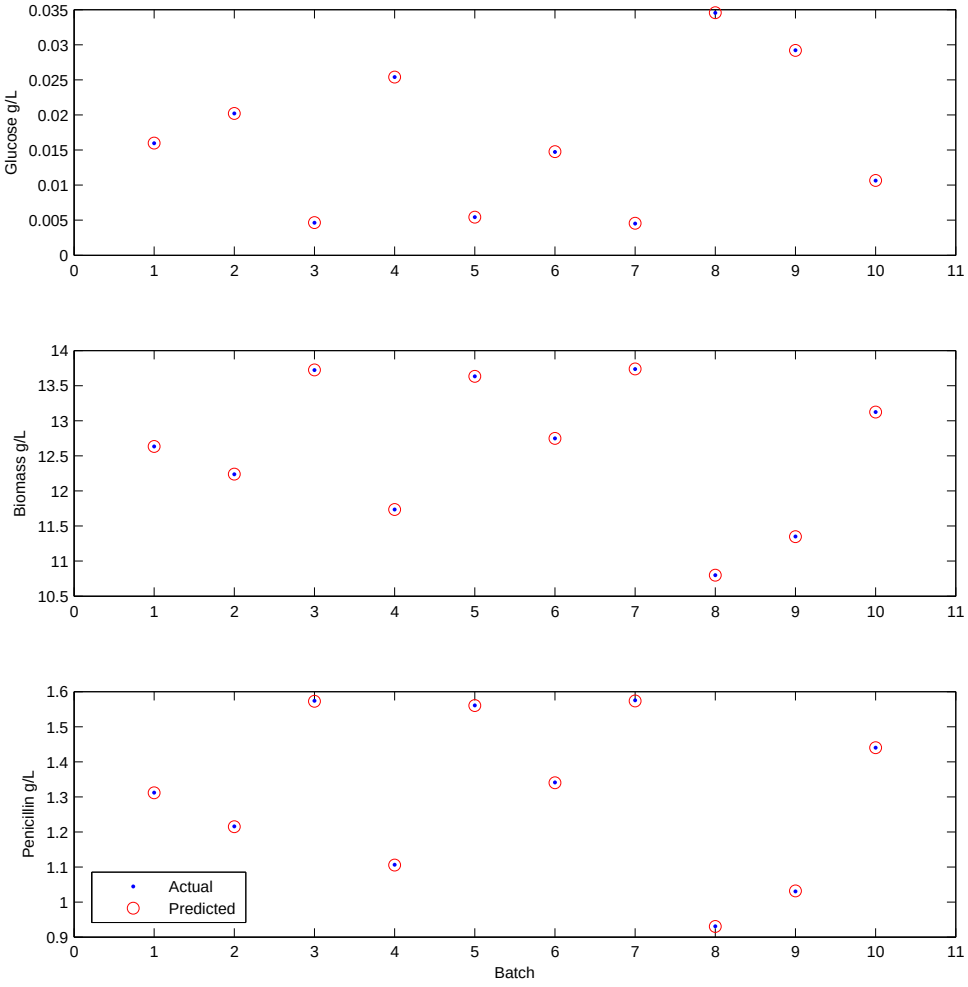

Figure 4.4: Prediction of quality variables of the 3rd modeling phase

Next, localized dynamic models are developed for reactor temperature, dissolved oxygen, carbon dioxide concentration and fermenter volume given the manipulated variables, feed flow rate, aeration flow rate, agitator power and cooling water flow rate since these measurable variables must be predicted as part of the quality prediction. To cluster the database and compute the resulting local linear models, the following parameters are required to be specified: the lag structure (n_x and n_u) and the number of clusters L. The model identification procedure is as follows. The $\bar{\mathbf{X}}$ matrix is

constructed based on a given lag structure and clustered for a range of clusters, $L = \{1, 2, \ldots, 20\}$. For each cluster choice and a lag structure, a linear discrete time model is identified. This is repeated for all possible lag structures with a lag range of $0 - 2$ for each variable. Cross-validation is employed to minimize the RMSE of the predicted trajectories in order to select the best model structure of the localized models. The number of lags is found to be 1 (i.e. $n_x = n_u = 1$) and the number of clusters L is 3, 4 and 5 for the first, second and third model phases, respectively. One explanation for requiring only one lag is the assumption of the same lag structure for all the local models within one modeling phase (note that this assumption can be readily relaxed if needed). With this assumption, using all first order models minimizes the possibility of over-fitting, and in this case, yields the lowest RMSE values. In Figures 4.5, 4.6 and 4.7, the predicted outputs from the localized models are compared with the actual outputs of the nonlinear process for a batch in the validation data set. Overall, the developed localized models are capable of characterizing the major nonlinearities and lead to relatively reliable predictions in all modeling phases.

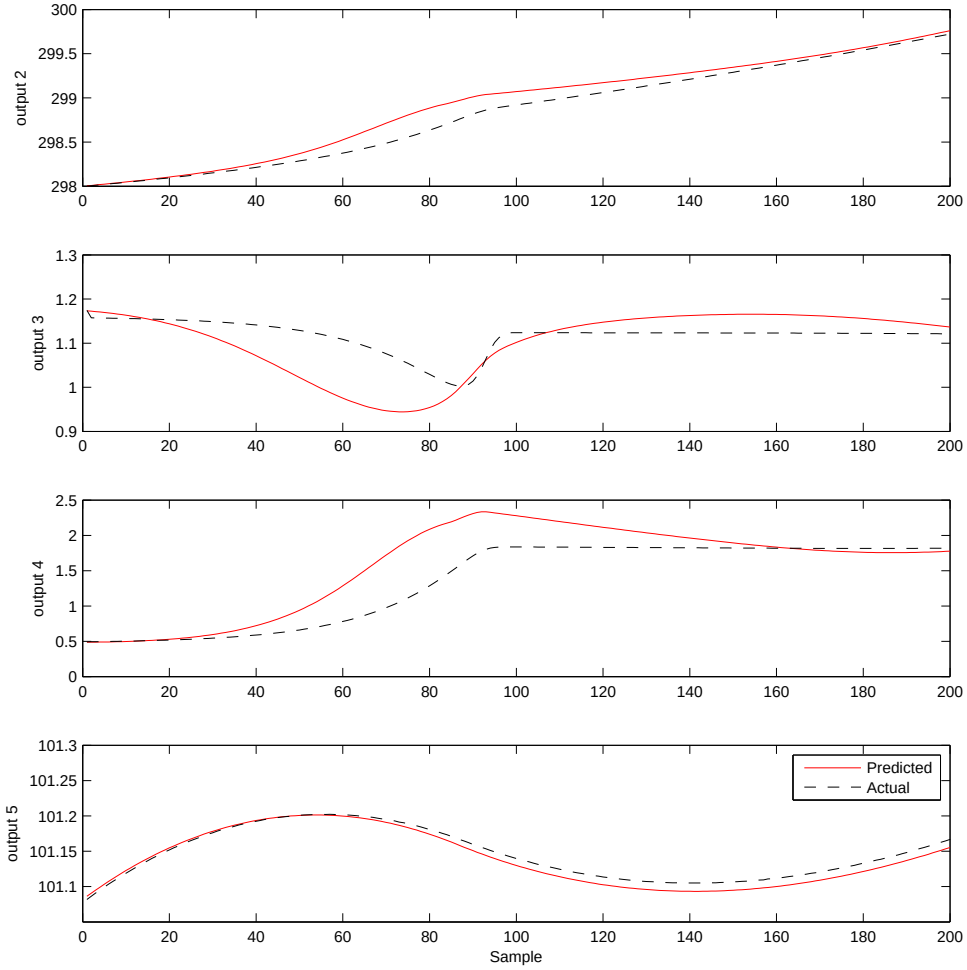

Figure 4.5: Prediction of future trajectories in the 1st modeling phase

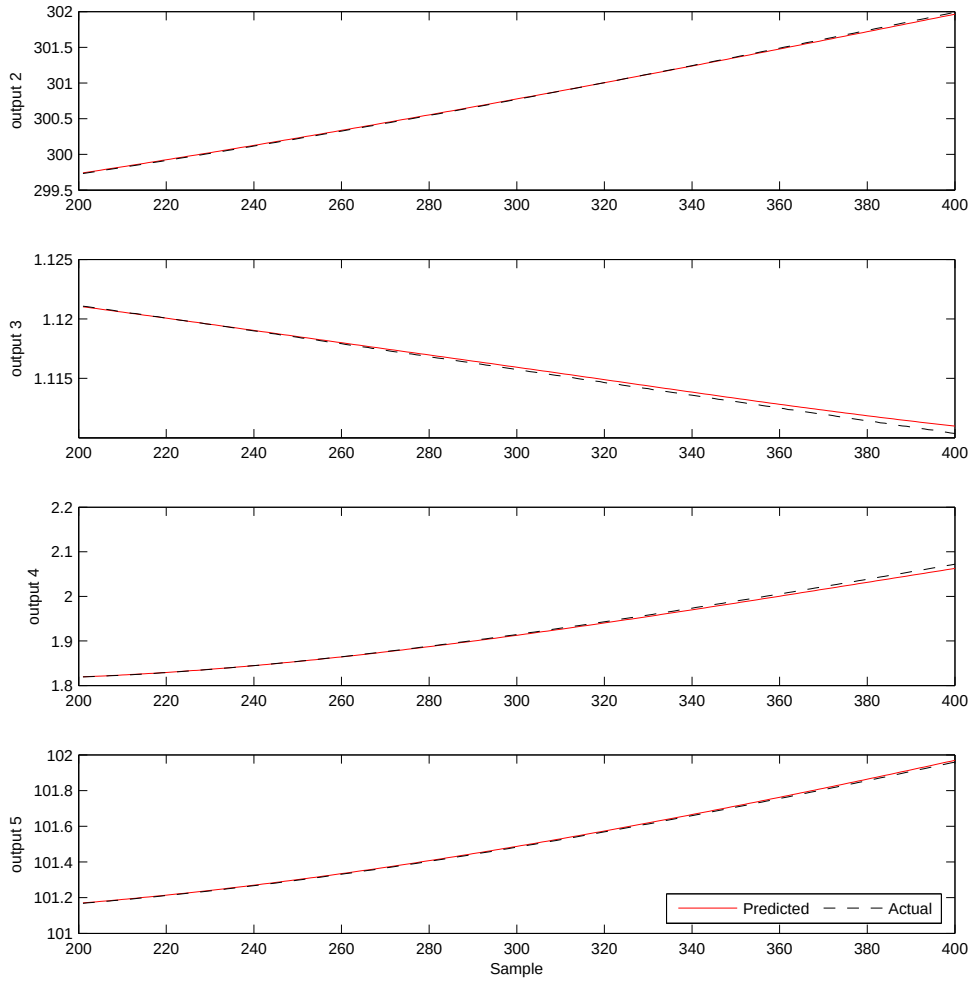

Figure 4.6: Prediction of future trajectories in the 2nd modeling phase

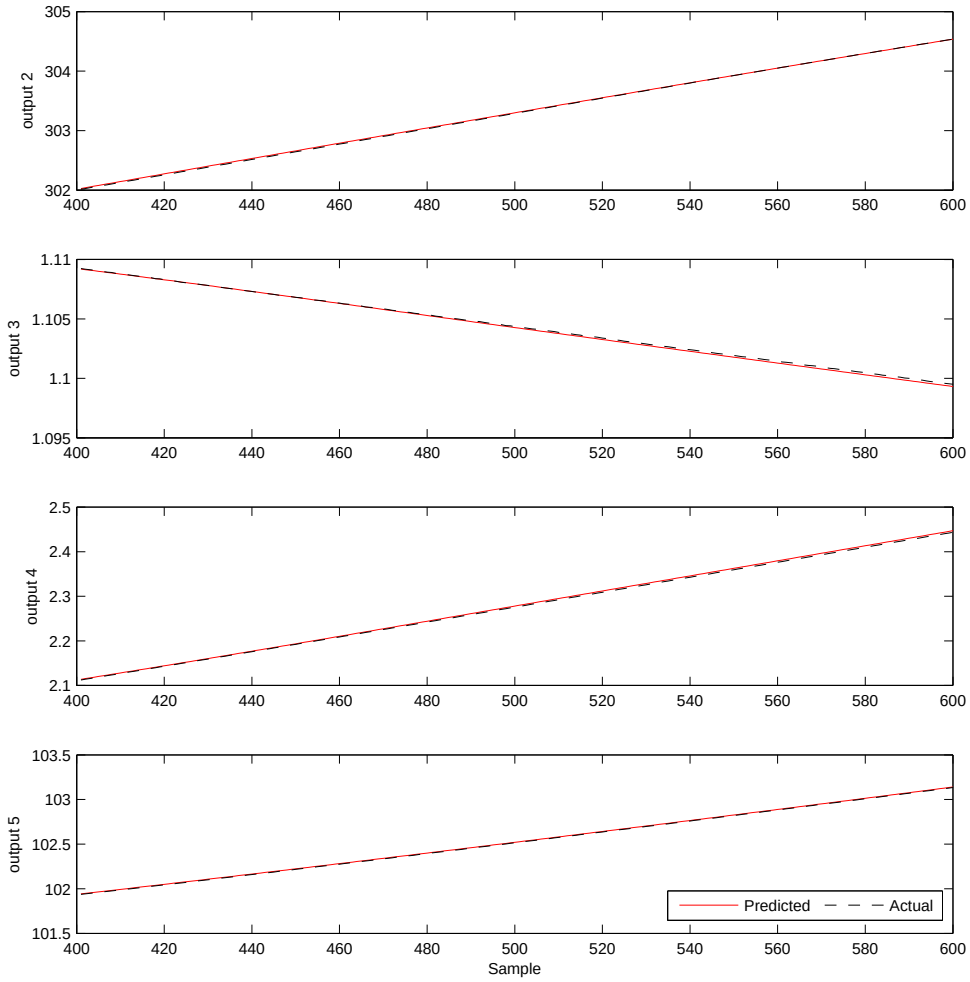

Figure 4.7: Prediction of quality variables in the 3rd modeling phase

4.4.2 Closed-loop Simulation Results

Using the models developed in the previous section, in this section, the multiple models based predictive controller can be implemented and its control performance is compared against PI control and single model based MPC. For these simulations, 30 new batches that are not in the training or validation data sets are generated and tested. The reference trajectories for the PI controllers are from the nominal batch, and the loop-pairings and tunings are kept consistent with the database generation procedure. In

solving the MPC optimization problem, the initial guess for the input trajectories is set to the nominal trajectories at $k = 0$ and the tail of the solution at the previous sampling instant for all subsequent sampling instants. The computation time reported by the MATLAB for the predictive controller at $k = 0$ is 32.83 seconds on an Intel Dual Core machine using GAMS with CONOPT as the optimization software. The computation times for all successive sampling times are lower due in part to the shrinking horizon nature of the optimization problem. The trajectory of control inputs and process measurements of the Penicillin process by different control designs are shown in Figures 4.8 and 4.9.

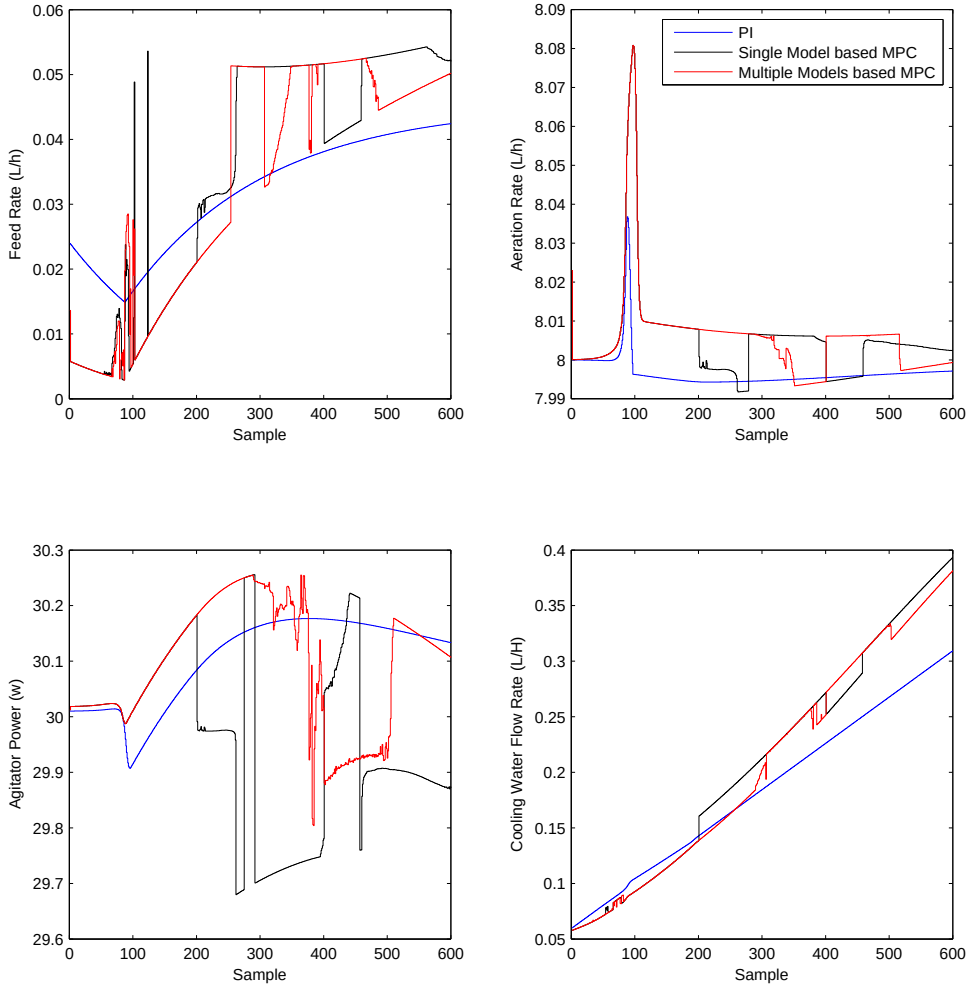

Figure 4.8: Trajectory of control inputs in different control designs

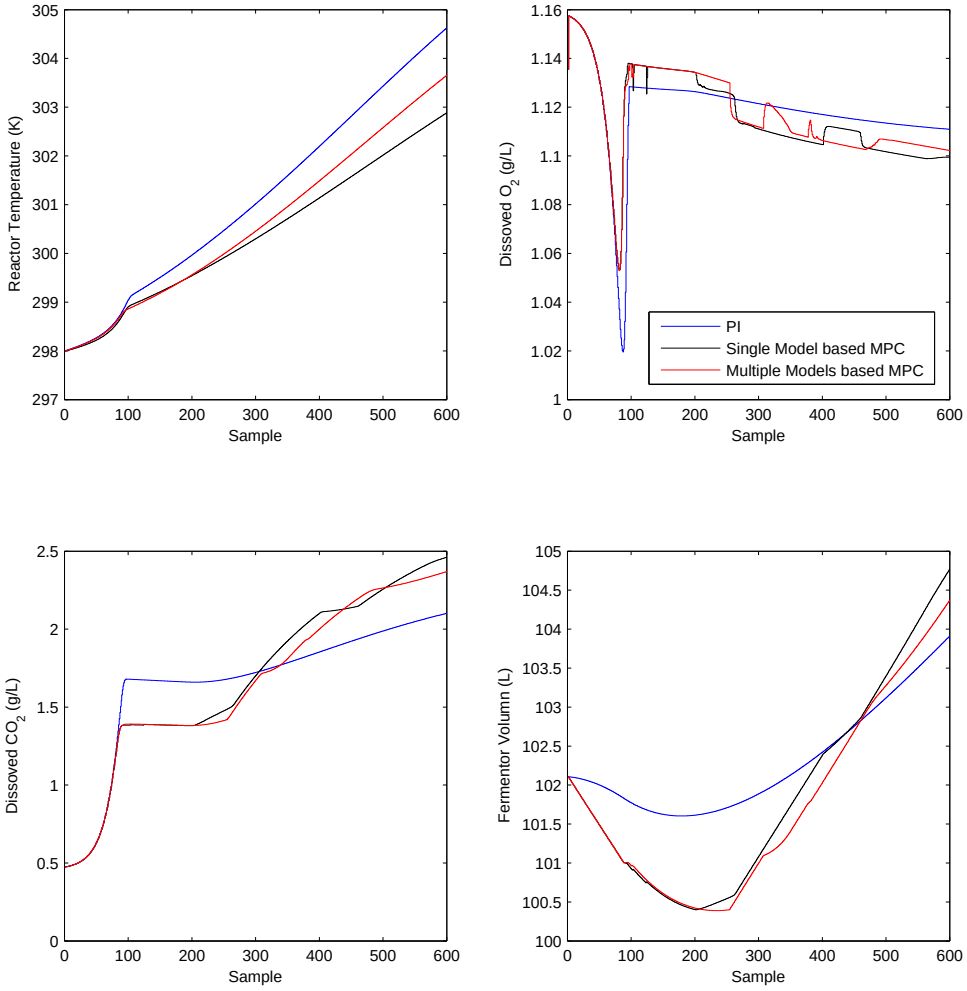

Figure 4.9: Trajectory of process measurements in different control designs

Table 4.2: Averages of final product qualities by different control designs

Approach	Biomass (g/L)	Penicillin (g/L)
PI Control	11.3844	1.0422
Single Model based MPC	12.5734	1.2354
Multiple Models based MPC	13.0567	1.2807

In Figure 4.10, the final qualities yielded from the proposed control design are compared with those from trajectory tracking and single model based MPC designs. It can be readily observed that the deviations of the final product qualities obtained by single model based MPC and the proposed multiple models based MPC are much smaller than that of the PI control design. It indicates that the predictive controller leads to more reliable control performance than the trajectory tracking design. As shown in Table 4.2, the proposed multiple models based MPC design leads to the best average final product qualities for 30 test batches. The best control performance of the presented multiple models based MPC can be first attributed to the localized dynamic models in that the localized dynamic models can well characterize the nonlinear process dynamics in each modeling phase. Moreover, the Penicillin fermentation process can be driven to the desired quality through phase-to-phase MPC optimization in the multiple models based MPC so that scale of the optimization problem is significantly reduced compared with the single model based MPC. Therefore, the proposed multiple models based MPC achieves the most desirable control performance.

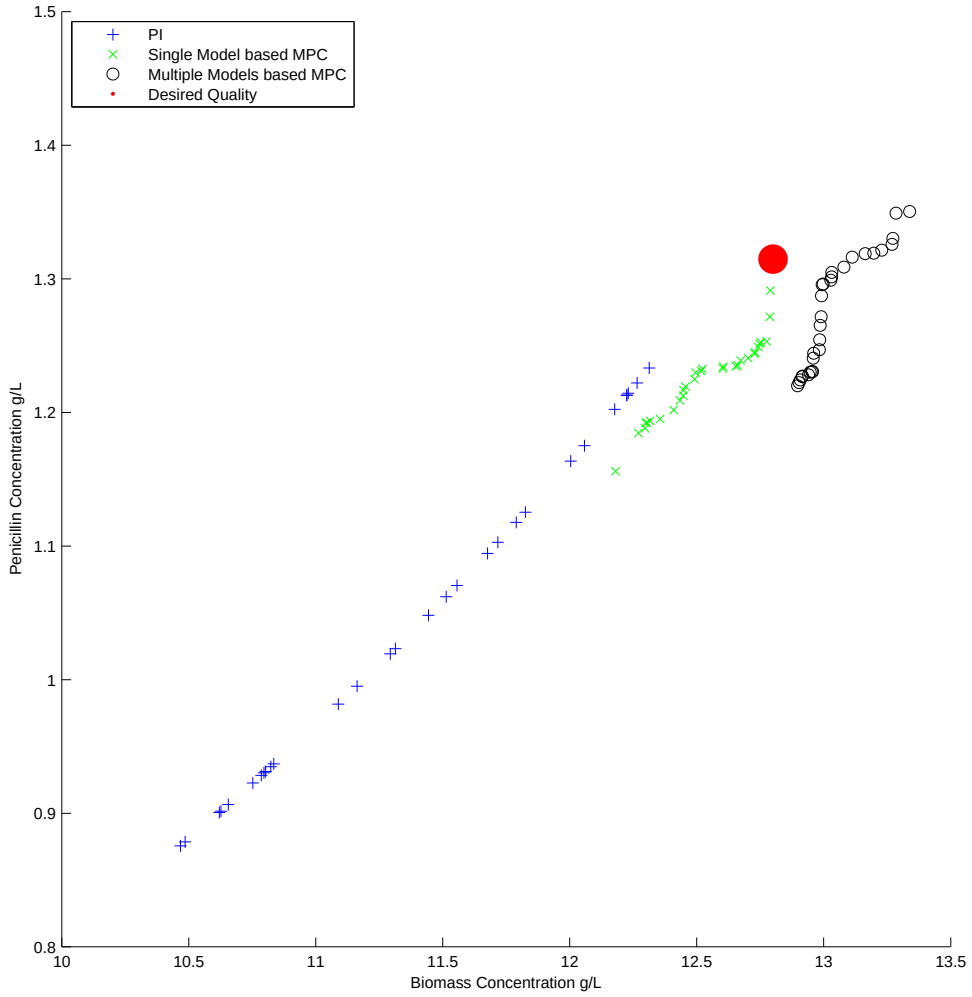

Figure 4.10: Comparison of the final product qualities obtained by three control designs

4.5 Conclusions

In this chapter, a multiple quality model based control design is proposed for the Penicillin production process to drive the batch to a specified end-of-batch product quality. The MPC design integrates multiple PLS models to relate the process measurements with intermittent quality measurements to the final product quality and localized models to predict future process trajectories up to the termination of a modeling phase.

The presented approach leads to effective control action by accounting for the causality and nonlinear relationships between the future input and process measurements trajectory through the multiple data-driven models. The proposed multiple models based MPC results in significant improvement on final product quality in application to the highly nonlinear batch fermentation process, with comparison to trajectory tacking control and single model based MPC.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this work, we address the data-driven modeling for quality control in complex chemical processes. For the initial phase of this research, covered in Chapter 2, expectation-maximization and Bayesian inference based PPLS methods are proposed for data-driven modeling under stochastic missing measurements. In the EM-PPLS method, missing measurements are initially imputed by variable means and the parameter values of EM-PPLS model are filled with initial guess. Then, EM algorithm is employed to update latent scores and PPLS model parameter values through data-driven iterative learning. Meanwhile, the missing measurements are re-estimated by using the updated latent scores and model parameters after each EM step. Consequently, PPLS modeling and missing data estimation can be iterated simultaneously along with the uncertainty handling in process measurements through a probabilistic strategy. However, the selection of latent variables in EM-PPLS still involves computationally expensive cross-validation. To solve this issue, prior distributions are introduced on the parameters of BI-PPLS model so that the parameters along with the optimal size of

latent space in BI-PPLS can be identified through the recursive variational Bayesian inference. After each updating step for model parameters in BI-PPLS, missing measurements are also re-estimated by utilizing the new expectations of latent score and model parameters based on the updated posterior distributions. Therefore, the proposed BI-PPLS method can lead to a concurrent solution for missing data imputation and latent variable selection within a Bayesian framework. The proposed PPLS methods are applied to a simulated example with stochastic missing measurements. The soft sensor modeling results indicate that both approaches can robustly estimate missing measurement and predict quality variables with satisfactory accuracy. In comparison, BI-PPLS outperforms EM-PPLS in terms of significantly higher computational efficiency. The data-driven modeling methods for quality prediction and control are developed for the purpose of industrial application. The developed PPLS methods can potentially be incorporated in multivariate statistical modeling software with graphical user interface, so that engineers can easily use the advanced PPLS methods to handle the missing measurements in training data without knowing the details of sophisticated statistical inference.

With the missing data estimated by the methods in Chapter 2, data-driven inferential models can be generated and implemented online. In Chapter 3, a novel model mismatch index is developed to monitor the performance of PLS soft sensor models in industrial applications. Based on estimated regression parameters, the Kalman filter based model mismatch index is developed and employed to detect the mismatch between the current data-driven model and process dynamics. The predictions of the data-driven inferential model are recognized as uncertain if the model mismatch index overpasses the predefined control limit. In order to avoid a biased update for soft sensor models under faulty operations, the decision of the model update can be made based on the model mismatch index with the assistance of traditional process monitoring

indexes. Furthermore, a contribution plot for the model mismatch index is developed in order to isolate the leading abnormal regression coefficients that are causing the degradation of the data-driven model. After updating the model, the regression coefficients are estimated through PLS regression using data of the training set and the current process conditions. Consequently, the proposed method can handle changes in process dynamics and lead to accurate quality predictions under various process conditions. The proposed approach is validated using simulated and industrial case studies, indicating that the calculated index can capture the model mismatch of the soft sensor model earlier, which in turn results in more accurate predictions. In addition, the comparison with the recursive partial least squares (RPLS) method demonstrates that the proposed approach leads to stable model structures, which is more desirable in industrial applications. The developed online model maintenance method is based on Kalman filter and PLS, which are mature technologies with many years' industrial application. Therefore, the online model maintenance method can be easily adopted by process control engineers in process industry.

Finally, in Chapter 4, a multiple quality model based control design is proposed for the Penicillin production process to drive the batch to a specified end-of-batch product quality. The MPC design integrates multiple PLS models to predict the final product quality using intermittent quality measurements at different modeling phases and localized models to predict future process trajectories up to the termination of a modeling phase. The presented approach leads to effective control action by accounting for the causality and nonlinear relationships between the future input and process measurements trajectory through the multiple data-driven models. The proposed multiple models based MPC results in significant improvement on final product quality in application to the a highly nonlinear batch fermentation process, with comparison to trajectory tacking control and single model based MPC. The ARX and PLS modeling

techniques utilized in the developed multiple models based MPC are widely used in industrial application with reliable performance, so the proposed control design can be applied to industrial processes without technology difficulty.

5.2 Future work

In this section, a few topics of future research are suggested.

- The missing measurement problem is considered in a liner model framework. However, a liner model structure may not be sufficient to characterize the data collected from essentially nonlinear processes. To enhance the nonlinear modeling capability of PPLS, kernel functions can be introduced and combined with the existing PPLS algorithms.

- The proposed soft sensor performance monitoring method in Chapter 3 is designed for linear PLS models. The real challenge in industrial practice will be how to handle any inherent process nonlinearity. Particularly for chemical processes like polymer process or other batch processes, process nonlinearity often cannot be ignored. If nonlinear type of PLS model is used, then Kalman will no longer be appropriate in that Kalman filter only handles linear Gaussian systems. Under that circumstances, unscented Kalman filter or particle filter can potentially be employed to monitoring the soft sensor prediction performance.

- Note that the process trajectories during some operating phases in the batch are irrelevant to the final product quality. Including irrelevant information in the predictive controller may deteriorate the performance of quality control. Instead, it is recognized that the final product quality only depends on the final state of the process. As such, a state space model that describes the dynamic evolution

of the batch process shall be identified through subspace identification and the final state of the batch can be predicted at every sampling instance by iterating the state and measurement equations in the state space model. Meanwhile, a soft sensor model can be constructed to infer the end-of-batch quality from the estimated final state. Compared with widely used multiway analysis methods in batch processes, estimating quality from final state leads to better interpretability. Another distinguishing feature of such design is that irrelevant information in the process trajectory will not affect the estimation and the quality control performance may be improved.

Bibliography

Aumi, S. and Mhaskar, P. (2009). Safe-steering of batch processes. *AIChE J.*, **55**, 2861–2872.

Aumi, S. and Mhaskar, P. (2011). Integrating data-based modelling and nonlinear control tools for batch process control. In *Proc. of the American Control Conf.*, pages 2534–2539, San Francisco, CA.

Aumi, S., Corbett, B., Mhaskar, P., and Clarke-Pringle, T. (2013a). Data-based modeling and control of nylon-6, 6 batch polymerization. *IEEE Trans. Control Syst. Technol.*, **21**, 94–106.

Aumi, S., Corbett, B., Clarke-Pringle, T., and Mhaskar, P. (2013b). Data-driven model predictive quality control of batch processes. *AIChE J.*, **59**, 2105–2119.

Birol, G., Ündey, C., and Çinar, A. (2002). A modular simulation package for fed-batch fermentation: penicillin production. *Comput. Chem. Eng.*, **26**, 1553–1565.

Bishop, C. (1999). Variational principal components. In *Ninth International Conference on Artificial Neural Networks, ICANN99*, pages 509–514, Edinburgh, Scotland.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA.

Bosca, S. and Fissore, D. (2011). Design and validation of an innovative soft-sensor for pharmaceuticals freeze-drying monitoring. *Chem. Eng. Sci.*, **66**, 5127–5136.

Bowman, A. W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, New York, USA.

Box, G., Jenkins, G., and Reinsel, G. (1994). *Time Series Analysis: Forecasting and Control*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Chen, J.-M. and Chen, B.-S. (2000). System parameter estimation with input/output noisy data and missing measurements. *IEEE Trans. Signal Process.*, **6**, 1546–1558.

Chen, Z. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, **182**, 1–69.

Cheng, C. and Chiu, M.-S. (2004). A new data-based methodology for nonlinear process modeling. *Chem. Eng. Sci.*, **59**, 2801–2810.

Choi, S. W. and Lee, I.-B. (2005). Multiblock pls-based localized process diagnosis. *Journal of Process Control*, **15**, 295–306.

Chong, I.-G. and Jun, C.-H. (2005). Performance of some variable selection methods when multicollinearity is present. *Chemometrics Intell. Lab. Syst.*, **78**, 103–112.

Clarke-Pringle, T. and MacGregor, J. F. (1997). Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors. *Comput. Chem. Eng.*, **21**, 1395–1409.

Cruickshank, S. M., Daugulis, A. J., and McLellan, P. J. (2000). Dynamic modeling and optimal fed-batch feeding strategies for a two-phase partitioning bioreactor. *Biotech. and Bioeng*, **67**, 224–233.

Dayal, B. and Macgregor, J. (1997). Improved PLS algorithms. *J. Chemometrics*, **11**, 73–85.

Dayal, B. and MacGregor, J. (1997). Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *J. Proc. Cont.*, **7**(3), 169–179.

de Assis, A. and Fiho, R. (2000). Soft sensors development for on-line bioreactor state estimation. *Comput. Chem. Eng.*, **24**, 1099–1103.

Ding, F., Liu, G., and Liu, X. (2011). Parameter estimation with scarce measurements. *Automatica*, **47**, 1646–1655.

Ding, S., Yin, S., Peng, K., Hao, H., and Shen, B. (2009). A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *IEEE Trans. Ind. Informat.*, **9**, 2239–2247.

Dufour, P., Bhartiya, S., Dhurjati, P., and Doyle III, F. (2005). Neural network-based software sensor: training set design and application to a continuous pulp digester. *Control Eng. Pract.*, **13**, 135–143.

Facco, P., Doplicher, F., Bezzo, F., and Barolo, M. (2009). Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process. *J. Proc. Cont.*, **19**, 520–529.

Farasat, E. and Huang, B. (2012). Deterministic vs. stochastic performance assessment of iterative learning control for batch processes. *AIChE J.*, **59**, 457–464.

Flores-Cerrillo, J. and MacGregor, J. F. (2002). Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Ind. Eng. Chem. Res.*, **41**, 1805–1814.

Flores-Cerrillo, J. and MacGregor, J. F. (2003). Within-batch and batch-to-batch inferential-adaptive control of semibatch reactors: A partial least squares approach. *Ind. Eng. Chem. Res.*, **42**, 3334–3345.

Flores-Cerrillo, J. and MacGregor, J. F. (2004). Control of batch product quality by trajectory manipulation using latent variable models. *J. Proc. Cont.*, **14**, 539 – 553.

Flores-Cerrillo, J. and MacGregor, J. F. (2005). Latent variable MPC for trajectory tracking in batch processes. *J. Proc. Cont.*, **15**, 651–663.

Fortuna, L., Graziani, S., Rizzo, A., and Xibilia, M. G. (2007). *Soft sensors for monitoring and control of industrial processes*. Springer, London, UK.

Frigui, H. and Krishnapuram, R. (1997). Clustering by competitive agglomeration. *Pattern Recognit.*, **30**, 1109–1119.

Fujiwara, K., Kano, M., Hasebe, S., and Takinami, A. (2009). Soft-sensor development using correlation-based just-in-time modeling. *AIChE J.*, **55**, 1754–1765.

Golshan, M., MacGregor, J. F., Bruwer, M.-J., and Mhaskar, P. (2009). Latent variable MPC for trajectory tracking in batch processes: Role of the model structure. In *Proc. of the American Control Conference*, pages 4779–4784.

Golshan, M., MacGregor, J. F., Bruwer, M.-J., and Mhaskar, P. (2010). Latent variable model predictive control (LV-MPC) for trajectory tracking in batch processes. *J. Proc. Cont.*, **20**, 538–550.

Gonzaga, J., Meleiro, L., Kianga, C., and Filho, R. (2009). ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Comput. Chem. Eng.*, **33**, 43–49.

Hansson, A. and Wallin, R. (2012). Maximum likelihood estimation of Gaussian models with missing data-Eight equivalent formulations. *Automatica*, **48**, 1955–1962.

Heineken, W., Flockerzi, D., Steyer, C., Voigt, A., and Sundmacher, K. (2007). Nonlinear dynamics of continuous precipitation reactors: A model based analysis. *Chem. Eng. Sci.*, **62**, 4896–4902.

Helland, K., Berntsen, H. E., Borgen, O. S., and Martens, H. (1992). Recursive algorithm for partial least squares regression. *Chemometrics Intell. Lab. Syst.*, **14**, 129–137.

Hu, J., Wang, Z., Gao, H., and Stergioulas, L. (2012). Extended Kalman filtering with stochastic nonlinearities and multiple missing measurements. *Automatica*, **48**, 2007–2015.

Isaksson, A. (1993). Identification of ARX-models subject to missing data. *IEEE Trans. Autom. Control.*, **38**, 813–819.

Jin, X. and Huang, B. (2012). Identification of switched markov autoregressive exogenous systems with hidden switching state. *Automatica*, **48**, 436–441.

Kadlec, P. and Gabrys, B. (2011). Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE J.*, **57**(5), 1288–1301.

Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Comput. Chem. Eng.*, **33**, 795–814.

Kadlec, P., Grbić, R., and Gabrys, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Comput. Chem. Eng.*, **35**, 1–24.

Kaneko, H. and Funatsu, K. (2011a). Maintenance-free soft sensor models with time difference of process variables. *Chemometrics Intell. Lab. Syst.*, **107**, 312–317.

Kaneko, H. and Funatsu, K. (2011b). A soft sensor method based on values predicted from multiple intervals of time difference for improvement and estimation of prediction accuracy. *Chemometrics Intell. Lab. Syst.*, **109**, 197–206.

Kaneko, H. and Funatsu, K. (2013a). Adaptive soft sensor model using online support vector regression with time variable and discussion of appropriate hyperparameter settings and window size. *Comput. Chem. Eng.*, **58**, 288–297.

Kaneko, H. and Funatsu, K. (2013b). Discussion on time difference models and intervals of time difference for application of soft sensors. *Ind. Eng. Chem. Res.*, **52**, 1322–1334.

Kano, M. and Nakagawa, Y. (2008). Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Comput. Chem. Eng.*, **32**, 12–24.

Kravaris, C. and Soroush, M. (1990). Synthesis of multivariable nonlinear controllers by input/output linearization. *AIChE J.*, **36**(2), 249–264.

Kroonenberg, P. M. and De Leeuw, J. (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, **45**(1), 69–97.

Lee, J. H., Lee, K. S., and Kim, W. C. (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, **36**, 641 – 657.

Li, G., Qin, S., and Zhou, D. (2010). Geometric properties of partial least squares for process monitoring. *Automatica*, **46**, 204–210.

Li, S., Gao, J., Nyagilo, J., and Dave, D. (2011). Probabilistic partial least square regression: A robust model for quantitative analysis of Raman spectroscopy data. In *2011 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 526–531, Atlanta, Georgia, USA.

Liang, H. and Zhou, T. (2011). Robust state estimation for uncertain discrete-time stochastic systems with missing measurements. *Automatica*, **47**, 1520–1524.

Lin, B., Recke, B., Knudsen, J., and Jorgensen, S. (2007). A systematic approach for soft sensor development. *Comput. Chem. Eng.*, **31**, 419–425.

Little, R. and Rubin, D. (1987). *Statistical analysis with missing data*. John Wiley and Sons, Inc, New York, USA.

Liu, X., Kruger, U., Littler, T., Xie, L., and Wang, S. (2009). Moving window kernel PCA for adaptive monitoring of nonlinear processes. *Chemometrics Intell. Lab. Syst.*, **96**, 132–143.

MacGregor, J. F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock PLS methods. *AIChE J.*, **40**, 826–828.

Mangold, M. (2012). Use of a Kalman filter to reconstruct particle size distributions from FBRM measurements. *Chem. Eng. Sci.*, **70**, 99–108.

Mehmood, T., Liland, K. H., Snipen, L., and Sæbø, S. (2012). A review of variable selection methods in partial least squares regression. *Chemometrics Intell. Lab. Syst.*, **118**, 62–69.

Mu, S., Zeng, Y., Liu, R., Wu, P., Su, H., and Chu, J. (2006). Online dual updating with recursive PLS model and its application in predicting crystal size of purified terephthalic acid (PTA) process. *J. Proc. Cont.*, **16**, 557–566.

Nelson, P., Taylor, P., and MacGregor, J. (1996). Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics Intell. Lab. Syst.*, **35**, 45–65.

Ng, Y. S. and Srinivasan, R. (2009). An adjoined multi-model approach for monitoring batch and transient operations. *Comput. Chem. Eng.*, **33**, 887–902.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., and Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, **19**, 2088–2096.

Patel, S., Yelchuru, R., Ryaliand, S., and Gudi, R. D. (2011). Discriminatory learning based performance monitoring of batch processes. In *Proc. of the American Control Conf.*, pages 2552–2557.

Qin, S. J. (1998). Recursive PLS algorithms for adaptive data modeling. *Comput. Chem. Eng.*, **22**, 503–514.

Qu, L., Li, L., Zhang, Y., and Hu, J. (2009). PPCA-based missing data imputation for traffic flow volume: a systematical approach. *IEEE Trans. Itell. Transp. Syst.*, **10**, 512–522.

Rännar, S., Geladi, P., Lindgren, F., and Wold, S. (1995). A PLS kernel algorithm for data sets with many variables and few objects. Part II: Cross-validation, missing data and examples. *J. Chemometrics*, **9**, 459–470.

Rubin, D. (1976). Inference and missing data. *Biometrika*, **63**, 581–592.

Ruscio, D. (2000). A weighted view on the partial least-squares algorithm. *Automatica*, **36**, 831–850.

Rutan, S. C. (1990). Recursive parameter estimation. *J. Chemometrics*, **4**, 103–121.

Shao, X., Huang, B., and Lee, J. M. (2010). Constrained bayesian state estimation–a comparative study and a new particle filter based approach. *J. Proc. Cont.*, **20**, 143–157.

Sharmin, R., Sundararaj, U., Shah, S., Griend, L. V., and Sun, Y.-J. (2006). Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant. *Chem. Eng. Sci.*, **61**, 6372–6384.

Shi, D., Mhaskar, P., El-Farra, N., and Christofides, P. (2005). Predictive control of crystal size distribution in protein crystallization. *Nanotechnol.*, **16**, 562–574.

Shi, D., El-Farra, N. H., Li, M., Mhaskar, P., and Christofides, P. D. (2006). Predictive control of particle size distribution in particulate processes. *Chem. Eng. Sci.*, **61**, 268–281.

Srinivasan, B., Palanki, S., and Bonvin, D. (2003a). Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Comput. Chem. Eng.*, **27**(1), 1–26.

Srinivasan, B., Bonvin, D., Visser, E., and Palanki, S. (2003b). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Comput. Chem. Eng.*, **27**(1), 27–44.

Tang, C., Yang, C., Gui, W., and He, J. (2010). Online prediction model based on adaptive recursive least squares support vector machine for silicon-manganese alloy composition. In *2010 International Conference on Intelligent System Design and Engineering Application (ISDEA)*, pages 1044–1048, Changsha, China.

Tipping, M. and Bishop, C. (1999). Probabilistic principal component analysis. *J. R. Stat. Soc. B*, **61**, 611–622.

Ündey, C. and Çinar, A. (2002). Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Syst. Mag.*, **22**, 1361–1375.

Ündey, C., Ertunç, S., and Çinar, A. (2003). Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis. *Ind. Eng. Chem. Res.*, **42**, 4645–4658.

Walczak, B. and Massart, D. (2001). Dealing with missing data: Part II. *Chemometrics Intell. Lab. Syst.*, **58**(1), 29–42.

Wan, J., Marjanovic, O., and Barry, L. (2010). Control of batch end-product quality using partial least squares-based models. In *UKACC International Conference on Control 2010*, pages 1–6, Coventry, UK.

Wang, B. and Titterington, D. (2004). Convergence and asymptotic normality of variational bayesian approximations for exponential family models with missing values. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 577–584.

Wang, D. and Srinivasan, R. (2009). Multi-model based real-time final product quality control strategy for batch processes. *Comput. Chem. Eng.*, **33**, 992–1003.

Wang, X., Kruger, U., and Lennox, B. (2003). Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Eng. Pract.*, **11**, 613–632.

Wang, X., Kruger, U., and Irwin, G. W. (2005). Process monitoring approach using fast moving window PCA. *Ind. Eng. Chem. Res.*, **44**, 5691–5702.

Wei, G., Wang, Z., and Shu, H. (2009). Robust filtering with stochastic nonlinearities and multiple missing measurements. *Automatica*, **45**, 836–841.

Welz, C., Srinivasan, B., and Bonvin, D. (2008). Measurement-based optimization of batch processes: Meeting terminal constraints on-line via trajectory following. *J. Proc. Cont.*, **18**, 375–382.

Wold, S., Geladi, P., Esbensen, K., and Öhman, J. (1987). Multi-way principal components-and PLS-analysis. *J. Chemom.*, **1**, 41–56.

Wold, S., Sjöström, M., and Eriksson, L. (2002). PLS-regression: a basic tool of chemometrics. *Chemometrics Intell. Lab. Syst.*, **58**, 109–130.

Xie, X. L. and Beni, G. (1991). A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**, 841–847.

Xu, L. and Jordan, M. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Comput.*, **8**, 129–151.

Yin, S., Ding, S., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *J. Proc. Cont.*, **22**, 1567–1581.

Yin, S., Luo, H., and Ding, S. (2013). Real-time implementation of fault-tolerant control systems with performance optimization. *IEEE Trans. Ind. Electron.*, **61**, 2402–2411.

Yu, J. (2012a). A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Comput. Chem. Eng.*, **41**, 134–144.

Yu, J. (2012b). Multiway gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes. *Ind. Eng. Chem. Res.*, **51**, 13227–13237.

Yu, J. (2012c). A nonlinear kernel Gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes. *Chem. Eng. Sci.*, **68**, 506–519.

Yu, J. (2012d). Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach. *Chem. Eng. Sci.*, **82**, 22–30.

Yu, J., Chen, K., and Rashid, M. (2013). A Bayesian model averaging based multi-kernel Gaussian process regression framework for nonlinear state estimation and quality prediction of multiphase batch processes with transient dynamics and uncertainty. *Chem. Eng. Sci.*, **93**, 96–109.

Žáčěková, E., Prívara, S., and Váně, Z. (2011). Model predictive control relevant identification using partial least squares for building modeling. In *Australian Control Conference (AUCC), 2011*, pages 422–427, Melbourne, Australia.

Zhang, Y., Teng, Y., and Zhang, Y. (2010a). Complex process quality prediction using modified kernel partial least squares. *Chem. Eng. Sci.*, **65**, 2153–2158.

Zhang, Y., Zhou, H., Qin, S., and Chai, T. (2010b). Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *IEEE Trans Ind. Informat.*, **6**, 3–10.

Zhao, Z., Huang, B., and Liu, F. (2013). Bayesian method for state estimation of batch process with missing data. *Comput. Chem. Eng.*, **53**, 14–24.