Advanced Monitoring and Soft Sensor Development with Application to Industrial Processes

by

Hector Joaquin Galicia Escobar

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 07, 2012

Keywords: Process Monitoring, Fault Detection, Fault Diagnosis, Soft Sensor, Online Adaptation

Approved by

Dr. Jin Wang, Chair, Associate Professor of Chemical Engineering
Dr. Qinghua He, Associate Professor of Chemical Engineering, Tuskegee University
Dr. Harry T. Cullinan, Professor of Chemical Engineering
Dr. Mario R. Eden, Professor of Chemical Engineering

Abstract

This dissertation presents the research performed to develop innovative multivariate process monitoring and soft sensor approaches for industrial processes.

In the operation of modern industrial processes, it is attempted to improve product quality and yield in shorter periods of time while observing tighter environmental regulations and safety guidelines. Furthermore, these goals are tried to be accomplished facing a severe global competition and shorter product life cycles. The most reliable approach to achieve these goals is the development of first principle models (e.g. based on dynamic mass and energy balances) to describe the physics, chemistry, and the different phenomena occurring in the process. However, because industrial processes are generally quite complex to model and are characterized by significant inherent nonlinearities, a rigorous theoretical modeling approach is often impractical, requiring a great amount of effort. An alternative approach to address these aspects is using process data. In most modern industries, a large amount of variables are measured and stored automatically by the governing control system (e.g. DCS). The stored data corresponds to different measurement instruments that are used for closed-loop control or merely as indicators of different conditions in process units. The process data can be utilized to address several aspects. These include monitoring the process over time in order to detect special events (e.g. faults) and assign causes for them. Furthermore, the data can be used to predict key process variables that are difficult to measure online (e.g. quality) based on easy-to-measure process variables.

In the first part of this work (PART I), a new multivariate method to monitor continuous and batch processes is developed based on the statistics pattern analysis (SPA) framework. The SPA framework is developed to address challenges present in industrial data such as nonlinearities and multi-modal distributions, which are characteristics of process data that normally affect traditional multivariate monitoring methods. In addition, different unique characteristics of the SPA framework are discussed in detail. Also presented are comparisons of the monitoring performance based on SPA with the monitoring performance of different linear and nonlinear multivariate monitoring methods. The comparison results clearly demonstrate the superiority of the SPA-based monitoring in key aspects such as detecting and diagnosing different faults. The research findings indicate that the SPA-based monitoring is a promising alternative for monitoring industrial processes, especially where nonlinearities and multi-modal distributions are present.

In the second part of this work (PART II), the idea of obtaining quality estimates of key process variables using multivariate projection methods is explored. In many industrial processes, the primary product variable(s) are not measured online or not measured frequently, but are required for feedback control. To address these challenges, there has been increased interest toward developing data-driven software sensors (or soft sensors) using secondary measurements (e.g. variables easy-to-measure) based on multivariate regression techniques. The prediction of these key process variables not only provides a way to have real time measurements, but also aids in improving quality and preventing undesirable operating conditions.

In this work, the properties and characteristics of dynamic partial least squares (DPLS) based soft sensors are studied in detail. In addition, a reduced-order DPLS (RO-DPLS) soft sensor approach is proposed to address the limitation of the traditional DPLS soft sensor when applied to model processes with large transport delay. The RO-DPLS soft sensor is also extended to its adaptive version to cope with processes where frequent process changes occur. Also presented are alternatives to address practical issues such as data scaling and the presence of outliers off-line and online. The application of the proposed soft sensor is illustrated with various simulated and industrial case studies of a continuous Kamyr digester. In both simulation and industrial case studies, the proposed RO-DPLS soft sensor shows that it can provide quality estimates of key process variables in the presence of different disturbances, data outliers, and time-varying conditions.

Acknowledgments

This dissertation would not have been possible without the help of many people. First and foremost, I'd like to thank my advisor, Dr. Jin Wang, whose encouragement, guidance and support from the initial to the final level enabled me to develop a deeper understanding. Her example of hard work and dedication have definitively made me a better researcher. I would also like to thank my research committee members, Dr. Qinghua He, Dr. Mario R. Eden, and Dr. Harry T. Cullinan for their words of encouragement and their valuable comments which have helped me complete this dissertation.

I would like to show my gratitude to Dr. Qinghua He. His contributions to develop my understanding of different aspects of my research field have been fundamental and definitively beyond the call of duty. Special thanks are given to Dr. Christopher B. Roberts, Dr. Mark E. Byrne and Dr. Mario R. Eden who have always provided advice, whether it was scholarly or personal. I'd also like to thank Dr. John Hung for his classes and teaching style which have had tremendous impact in my career and in my development as researcher. Thanks are also due to current and past members of the Wang's Research Group, specially Meng Liang, Min Kim and Xiu Wang, I wish you the best in your future careers.
For their financial support, I would like to thank the National Science Foundation and the Alabama Center for Paper and Bioresource Engineering.

On the non-technical side, I owe my deepest gratitude to my parents Joaquin Galicia and Carmelina Escobar for teaching me that a good attitude and effort are a fundamental part of a happy life. To my sister Melina, I will never forget you and will love you forever.

Last, but not least, I would like to thank my beautiful wife Arianna for everything she is and does. You are the love of my life and my inspiration “Hasta El Final”.

Thank you God for showing me the path of light through the heart of my wife, my family, my friends, my professors, and the Auburn Family, I trust You with all my heart.

Hector J. Galicia Escobar
Auburn, Alabama
April 09, 2012

Table of Contents

List of Figures

List of Tables

List of Abbreviations

$(\bullet)^T$	transpose of a vector or matrix
$(\bullet)^{-1}$	inverse of a matrix
$(\hat{\bullet})$	prediction
α	significance level
$\mathbf{C}$	$\in \mathbb{R}^{m \times p}$ coefficient matrix in PLS
$\mathbf{\Lambda}$	diagonal matrix of the a retained eigenvalues in PCA
λ	forgetting factor
$\langle \mathbf{x}, \mathbf{y} \rangle$	inner product of vectors $\mathbf{x}$ and $\mathbf{y}$
τ_D	derivative time
τ_I	integral time
θ	process time delay
a	number of retained principal components in PCA/PLS
D_p	monitoring index for the principal component subspace in SPA
D_r	monitoring index for the residual subspace in SPA
$E\left\{\bullet\right\}$	expected value of a random variable
f	future horizon
K_c	proportional gain

l_a	time lag between measurements for DPLS and auto-/cross-correlation statistics
m	number of inputs in $\mathbf{X}$
n	number of samples in $\mathbf{X}$ and $\mathbf{Y}$
p	number of outputs in $\mathbf{Y}$
T^2	Hotelling's T^2 statistic
V	$\in \Re^{m\times n}$ noise matrix
w	window size
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C^s}$, $\mathbf{C^c}$	system matrices for a LTI system
$\mathbf{K}$	the steady-state Kalman filter gain
$\mathbf{P}$	$\in \Re^{m\times a}$ loading matrix for $\mathbf{X}$
$\mathbf{R}$	$\in \mathbb{R}^{n\times p}$ residual matrix of $\mathbf{Y}$
$\mathbf{T}$	$\in \Re^{n\times a}$ score matrix for $\mathbf{X}$
$\mathbf{X}$	$\in \Re^{m\times n}$ original process matrix
$\mathbf{x}(k)$	sample vector for inputs
$\mathbf{U}$	$\in \mathbb{R}^{n\times a}$ score matrix for $\mathbf{Y}$
$\mathbf{B}$	$\in \mathbb{R}^{a\times a}$ diagonal matrix of inner model coefficients in PLS
$\mathbf{Q}$	$\in \mathbb{R}^{p\times a}$ loading matrix for $\mathbf{Y}$
$\mathbf{Y}$	$\in \mathbb{R}^{n\times p}$ matrix of primary measurements
AA	Active Alkali
ANN	Artificial Neural Networks

BRPLS	Block-Wise Recursive Partial Least Squares
DCS	Distributed Control System
DL	Dissolved Lignin
DPA	Dynamic Principal Component Analysis
DPLS	Dynamic Partial Least Squares
DS	Total Dissolved Solids
EA	Effective Alkali
FF-BRPLS	Forgetting Factor Block-Wise Recursive Partial Least Squares
HOS	Higher-Order Statistics
IC	Independent Component
ICA	Independent Component Analysis
KPCA	Kernel Principal Component Analysis
LTI	Linear Time Invariant
MLR	Multiple Linear Regression
MPC	Model Predictive Controller
MPCA	Multiway Partial Least Squares
MPE	Mean Prediction Error
MPLS	Multiway Principal Component Analysis
MSPE	Mean Squared Prediction Error
MSPM	Multivariate Statistical Process Monitoring

MW-BRPLS	Moving Window Block-Wise Recursive Partial Least Squares
NIPALS	Nonlinear Iterative Partial Least Squares
PCA	Principal Component Analysis
PCR	Principal Component Regression
PCS	Principal Component Subspace
PID	Proportional-Integral-Derivative Controller
PLS	Partial Least Squares
RO-DPLS	Reduced-order Dynamic Partial Least Squares
RPLS	Recursive Partial Least Squares
RS	Residual Subspace
SP	Statistic Pattern
SPA	Statistics Pattern Analysis
SPE	Squared Prediction Error statistic
SPM	Statistical Process Monitoring
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TEP	Tennessee Eastman Process

Chapter 1

Introduction and Dissertation Outline

Modern chemical plants normally have a large variety of sensors dedicated solely to the task of monitoring equipments and process conditions. As a result, a large number of variables are automatically measured and stored by the governing control system. This has led to the collection of vast amounts of data at different rates (e.g. seconds, minutes or hours). In the last few decades, researchers have dedicated considerable efforts to develop and improve ways to analyze the data. One of the most active research areas in recent decades have been the development of Statistical Process Monitoring (SPM) methods [1–11]. Traditional data-based monitoring methods include, the Shewhart [1], cumulative sum (CUSUM) [3], and exponentially weighted moving average (EWMA) control charts [4]. These methods are categorized as "univariate" since they only look at the behavior of a single variable at a time. Univariate approaches are utilized to monitor key process variables in order to distinguish between *common causes* of variation and *special* (or *assignable*) *causes*[1, 12]. The first type of variation refers to variations that affect the process all the time and are unavoidable within the current process [6]. Special causes, on the other hand, are those producing deviations that can potentially upset the process and deviate it from the desired operating point. It is well known that univariate monitoring approaches can fail [2], as these methods ignore the correlation among different variables that produce an effect in a certain key process variable (e.g quality). To overcome the difficulty of analyzing simultaneously hundreds of process variables collected at different rates, Multivariate Statistical Process Monitoring (MSPM) techniques, which make use of the multivariate space, have been extensively studied by the research community [13–15], and provide superior performance compared to the univariate

SPM control charts. MSPM techniques are designed to extract the main underlying phenomena that are driving the process at a certain time from the process data. Most of the MSPM techniques rely on the properties of chemometrics techniques. Wise et al. [16], define chemometrics as the science of relating measurements made on a chemical system to the state of the system via application of mathematical or statistical methods. In essence, chemometrics techniques make use of the available data to develop an empirical model that allows estimation of properties of interest such as quality or yield. The success of MSPM based on chemometrics techniques relies on the fact that multivariate statistical models extract the correlations among variables due to physical and chemical phenomena in an empirical way. Chemometrics techniques such as principal component analysis (PCA) [2, 13, 17, 18] and partial least squares (PLS) [19–25] are commonly applied. These methods are particularly suited to analyzing large sets of correlated data. This is achieved by finding linear combinations of the original variables and projecting them in a lower-dimensional space. This reduced space contains most of the variance observed originally and accounts for correlations among different variables. Some pioneering work using PCA-based monitoring is presented in the work of Jackson [14], Jackson and Mudholkar [15], Jackson [2], Kresta et al. [26], and more recently Wise et al. [27]. The application of MSPM monitoring methods have been reported successful in chemical and petrochemical industries [10, 11, 28–30] for batch and continuous processes. In industry, one of the major benefits of MSPM monitoring methods is associated with the detection and diagnosis of faults. By detecting failing sensors and process upsets, MSPC techniques complement the role of automatic process control (APC) to improve safety and process efficiency. It is important to point out that the goal of MSPM approaches is different from automatic process control. APC is applied to reduce the variability of key process and product variables. This is achieved by compensating from the observable behavior of disturbances in the process and product variables via manipulation of less important variables (e.g. manipulated variables) [6, 31]. MSPM methods are an additional layer that is applied on top of the process and its automatic control system. It is the goal of any monitoring approach to confirm that the process remains in the desired operating region, otherwise detect and diagnose the process behavior that indicates the occurrence of a special event.

Although MSPM methods such as PCA-based monitoring methods have been extremely successful in many applications, their monitoring performance can deteriorate depending on the characteristics of the process. There are two main reasons for the deterioration in performance. First, PCA is a second-order method, i.e., the mean and the variance-covariance of the process variables are only considered to characterize the process. As a result, PCA lacks the capability of providing higher order representations for non-Gaussian data [32], which is common in industrial data due to nonlinear characteristics of processes and the effect of feedback control. In addition, the theoretical development to determine the thresholds for the traditional monitoring indices in PCA (e.g. Hotelling's T^2 and SPE) is based on a multivariate distribution assumption. Therefore, when the scores (or latent variables) are non-Gaussian distributed due to process nonlinearity or other reasons, the utilization of the traditional monitoring indices may lead to a less satisfactory performance [32, 33].

In the first part of this work, a novel process monitoring framework termed statistics pattern analysis (SPA) is developed. The advantages for monitoring industrial processes are discussed, and its performance is comprehensively compared with traditional linear and nonlinear monitoring methods. In addition, the SPA framework is extended to fault diagnosis based on contribution plots [5, 6, 9, 10, 34–36] to cope with the difficulties of traditional methods for isolating the root cause of faults.

It is well known that regular PCA is only concerned with finding linear combinations of the original variables contained in a matrix $\mathbf{X}$ that explain most of the variability contained originally. In several cases, including a group of primary measurements of greater importance (e.g. quality or yield) could help indicating how the process is behaving. Unfortunately, in the majority of cases, these variables are not measured as frequently due to several reasons (e.g. no hardware sensors, online analyzers etc.). To address the challenge, soft sensors (or inferential estimators) have been developed. Soft sensors make use of the information contained in the matrix of process variables $\mathbf{X}$ to estimate the primary measurements $\mathbf{Y}$. Soft sensors can be developed in the form of a Luenberger observer [37] or a Kalman filter [38] using a first principles dynamic model of the process. However, because chemical processes are generally quite complex to model and are characterized by significant inherent nonlinearities, a rigorous theoretical modeling approach is often impractical, requiring a great amount of effort [39]. As a consequence, data-driven soft sensors have been developed and studied extensively. Multiple linear regression (MLR) is the most basic form of model to design a soft sensor. However, due to correlations in process variables, size of data set, and measurement noise, MLR methods lead to imprecise estimations and poor prediction [26, 40]. Data-driven soft sensors based on multivariate regression techniques have been developed to overcome the dimensionality, measurement noise and collinearity problems [8, 9, 28, 39, 41–49]. Because data-driven soft sensors are easy to develop and to implement online, they are potentially more attractive than stochastic filters or deterministic observers [39]. Artificial neural networks (ANN), principal component regression (PCR) or principal component analysis (PCA), and partial least squares (PLS) are widely used regression techniques, and their successful application to the development of soft sensors has been reported for different processes [9, 47, 50–54]. Dynamic PCA (DPCA) and Dynamic PLS (DPLS), the extensions of the conventional PCA and PLS algorithms for dynamic modeling, have been developed to cope with the dynamic nature of processes. In these methods, lagged measurements are augmented to the original input data matrices to cope with dynamic processes [29, 55]. Among the different soft sensor approaches, DPLS based soft sensors have been reported successful in different type of processes [44, 48, 56–58]. However, despite many successful applications, one limitation associated with DPLS is that it is usually viewed as an ad hoc method. Additionally, there is a lack of theoretical understanding on whether it can capture process dynamics adequately and whether it can provide unbiased estimate for both open and closed-loop data. Another problem encountered when performing DPLS is that the addition of a relatively large number of lagged values in the input data matrix causes a substantial increase in the dimension of the matrix. This implies that the computational load required to the development of multivariate regression becomes considerably expensive, and the regression procedure gets much lengthier. This is particularly true for spline PLS and ANN PLS, which are based on computationally more intensive algorithms. As a result, the model becomes more cumbersome to manipulate due to the presence of a higher number of parameters [46]. Furthermore, as it can be seen later in this work, when process residence time is long, large number of lagged measurements and more training data are needed for DPLS to capture the process dynamics, and to provide satisfactory soft sensor performance. To address these issues, in the second part of this work, a theoretical proof is derived to show that DPLS-based soft sensors are dynamic estimators that can adequately capture process dynamics, and are able to provide unbiased estimates of primary variables for both open loop and closed-loop data. In addition, a reduced order DPLS (RO-DPLS) approach with optimally selected lagged measurements of secondary variables is proposed (e.g. variables that are easy-to-measure). The RO-DPLS soft sensor not only reduces model size and improves prediction, but also significantly increases prediction horizon. The RO-DPLS soft sensor is also extended to its adaptive version to cope with processes with frequent process changes, addressing practical issues such as data scaling and the presence of outliers off-line and online.

To summarize, this dissertation focuses on developing advance data-driven monitoring and soft sensor strategies that can cope different industrial processes and their characteristics.

PART I : ADVANCED PROCESS MONITORING

Chapter 2 presents a review of the different data-driven techniques for process monitoring compared in this work. These methods include principal component analysis (PCA), dynamic PCA, kernel PCA (KPCA), independent component analysis (ICA). In this chapter, the main characteristics of the different algorithms and methods for fault detection and diagnosis are discussed.

In Chapter 3, a novel process monitoring framework termed statistics pattern analysis (SPA) is introduced [32, 59–62]. The major difference of SPA and the other multivariate statistics process monitoring methods is that in SPA, instead of monitoring the process variables directly, the statistics of the process variables are monitored instead. In this chapter, the SPA framework characteristics for fault detection are discussed. In addition, the framework is extended to fault diagnosis based on contribution plots. This chapter contains some illustrative examples that show in detail how and why the SPA framework works. Finally, the main characteristics and advantages of the SPA framework are discussed.

In Chapter 4, the potential of the SPA framework for fault detection and diagnosis is evaluated by using two case studies. In the first case study, a nonlinear system is simulated to examine the performance and properties of the statistics pattern analysis framework (SPA). In addition, the SPA framework performance is compared with traditionally applied linear methods for process monitoring. In this chapter, it is demonstrated that SPA is able to handle nonlinear processes, and is able to detect and diagnose subtle changes in the dynamics of the system. In this case study, compared to the traditional PCA and DPCA methods, SPA provides a more accurate, efficient, and reliable detection and diagnosis of faults. Two different cases with different size of training samples are also studied. The study indicates that SPA can also perform well with a limited amount of data. However, for the case where data are limited, a window overlapping is necessary. If the window overlapping is large, it causes the samples to be no longer independent from each other, which causes an increment in the false alarm rates for normal operating data. For the case where enough data are available (i.e., slight window-overlapping to no window overlapping), SPA not only is able to provide high detection rates and a more clear identification of a fault, but it also reduces the false alarms for normal operating data considerably and shows limited fault detection delay. In the second case study, the SPA-based monitoring is tested at a plant-wide level.

The Tennessee Eastman process is used to evaluate the performance of the developed SPA-based fault detection and diagnosis methods. In this case study, the performance of SPA is compared with the two commonly applied linear monitoring methods (e.g. PCA and DPCA) and two nonlinear monitoring methods (e.g. ICA and KPCA). This case study shows that in general, all five methods work well in detecting and diagnosing most of faults. However, for some faults that are difficult to detect and/or diagnose by PCA, DPCA, KPCA and ICA based methods, SPA based method provides superior performance. In addition, because the SPA-based method breaks down the contribution of a fault to different variable statistics, it provides extra information in addition to identifying the major fault-contributing variable(s). For example, the SPA-based contribution plots tell us whether the fault is due to a change in variable mean or variance, which is a tremendous advantage for online process monitoring. The excellent performance observed for fault detection and diagnosis, together with the less intensive computational load required implementing SPA makes the framework a promising alternative for monitoring industrial processes, especially when nonlinearities are prevalent.

PART II : SOFT SENSOR DEVELOPMENT

In industrial processes, several process variables that indicate the quality of the product or yield are not measured frequently. This issue not only prevents the establishment of feedback control strategies, but also forces the exclusion of these variables from the process monitoring strategy. As a consequence, the performance of industrial processes is deteriorated. To address the challenge, soft sensors (or inferential estimators) have been developed. Among the different methods to develop soft sensors, the focus is given to dynamic partial least square (DPLS) based soft sensors. DPLS based soft sensors have been successfully applied to provide quality estimates of primary variables (e.g. quality or yield) in several processes. As mentioned before, despite many successful applications, one limitation associated with DPLS is that it is usually viewed as an ad hoc method, as there is a lack of theoretical understanding on whether it can capture process dynamics adequately, and whether it can provide unbiased estimates for both open and closed-loop data. Another problem encountered when performing DPLS is that the addition of a relatively large number of lagged values in the input data matrix causes a substantial increase in the dimension of the matrix. This implies that the computational load required to develop the multivariate regression becomes considerably heavier, and the regression procedure gets much lengthier. As a result, the model becomes more cumbersome to manipulate due to the presence of a higher number of parameters. In addition, when process residence time is long, large number of lagged measurements and more training data are needed for DPLS to capture the process dynamics and to provide satisfactory soft sensor performance.

Given the importance of PLS in the development of this work, in Chapter 5, a brief review of the PLS method, its adaptive versions, and the data scaling schemes that accompany the different versions of a PLS model is presented.

In Chapter 7, a rigorous proof is derived to show that a DPLS-based soft sensor is a dynamic estimator that can adequately capture process dynamics, and provides unbiased estimates of the primary variable for both open loop and closed-loop data [63–66]. Furthermore, a reduced order DPLS (RO-DPLS) approach with optimally selected lagged measurements of the secondary variables is proposed. The RO-DPLS soft sensor not only reduces model size and improves prediction, but also significantly increases prediction horizon (i.e., provides future predictions). The performance of the proposed RO-DPLS is demonstrated using both a simulated single-vessel continuous digester and an industrial Kamyr digester. The simulation results confirm that the DPLS and RO-DPLS soft sensors can capture process dynamics adequately for both open loop and closed-loop operations. In addition, it is shown that the RO-DPLS soft sensor performs better than the traditional DPLS approach, especially when training data are limited. The industrial case study also confirms that RO-DPLS provides an effective and practical solution to estimate the Kappa number (i.e, the key indicator of the extent or reaction) in a continuous digester.

Real industrial processes are usually time-varying. For example, in the operation of a continuous digester, cooking temperature, the wood type utilized and other conditions vary with time. Therefore the soft sensor model developed off-line should be updated to cope with these changes. To address this difficulty, different adaptive schemes to update the soft sensor online are investigated in Chapter 7. In this chapter, the developed RO-DPLS soft sensor is extended to its online adaptive version [67–70]. The properties of four different model update schemes are examined. These adaptation techniques include, recursive PLS (RPLS), block-wise RPLS (BRPLS), moving window BRPLS (MW-BRPLS), and forgetting factor BRPLS (FF-BRPLS). In addition, the data scaling methods that accompany the different updating schemes are studied. Extensive experiments were performed with one simulated case study and two industrial case studies to investigate the properties of different model update schemes. Results of the study presented indicate that for processes that experience constant changes and various disturbances such as the digester process, recursive RO-DPLS with RPLS model update scheme generally performs the best with frequent update. The effectiveness of the recursive RO-DPLS soft sensor is demonstrated through a digester closed-loop control case study, where a PID controller is used to reject a major process disturbance i.e., a wood type change, and other common process disturbances. By comparing the closed-loop performance between measurement feedback and soft sensor prediction feedback, it is shown that the closed-loop control performance can be significantly improved if the soft sensor prediction is fed back to the PID controller. The improvement is mainly due to the reduced overall process delay as well as more frequent feedback enabled by the reduced order model and the high frequency Kappa number predictions.

In the development of PLS-based soft sensors, outlier detection plays a critical role, especially for recursive soft sensors that are updated online. In Chapter 8, multivariate approaches for both off-line outlier detection (for initial soft sensor model building) and online outlier detection (for soft sensor model recursive update) are proposed [71, 72]. Specifically, for off-line outlier detection leverage and y-studentized residuals are combined; while for online outlier detection, the squared prediction error indices for **X** and **Y** are used to monitor the independent variable and dependent variable space, respectively. For online outlier detection, to differentiate the outliers caused by erroneous reading from those caused by process changes, a Bayesian supervisory approach is proposed to further analyze and classify the identified outliers [71, 72]. Both simulated and industrial case studies demonstrate the superior performance of the soft sensor with Bayesian supervisory approach for outlier detection and classification.

In the final Chapter 9, the major contributions of this work are summarized and suggestions on future research directions are provided.

Chapter 2

Multivariate Process Monitoring

As discussed in the previous chapter, process monitoring techniques are not only key in determining equipment malfunctions and instrument failures, but also are fundamental in ensuring process safety, product quality, and process efficiency. As large amounts of variables are measured and stored automatically by governing control systems, multivariate statistical monitoring methods have become increasingly common in process industry. Multivariate process monitoring techniques are based in different mathematical algorithms. In addition, the way to detect and diagnose faults vary from one approach to the other. In this chapter, the main characteristics of some relevant multivariate process monitoring techniques are reviewed. This chapter reviews process monitoring techniques such as PCA and dynamic PCA, together with a brief review of nonlinear monitoring techniques such as Kernel PCA (KPCA) and Independent Component Analysis (ICA). The complete treatment of these process monitoring methods is out of the scope of this dissertation. However, enough background information is included so that the reader can follow the development of these process monitoring techniques, and to understand the development of Part I of this work.

2.1 Principal Component Analysis

Given the conceptual importance of PCA in the development of process monitoring schemes, in this chapter, the PCA method and related strategies for fault detection and diagnosis based on PCA are reviewed.

2.1.1 Motivation

Highly automated control systems in industrial sites are able to collect thousands of process measurements every day. The data captures the governing phenomena occurring in the process. However, the m variables that are selected to represent any given process are normally highly correlated and noisy. In addition, the number of variables that adequately represent a process can be very large. These factors can make the analysis of process data very difficult. The central idea of PCA is reducing the dimensionality of the original correlated process data while retaining as much as possible the variation present originally. This is achieved by projecting the process data into a lower-dimensional space, where a new set of uncorrelated variables, the principal components, contain most of the variation of the original variables in the first few principal components [73].

2.1.2 The PCA Method

Let $\mathbf{X} \in \Re^{\mathbf{m\times n}}$ denote the original process data matrix scaled to zero mean and unit variance with n samples (rows) and m process variables (columns). PCA decomposes the process data matrix $\mathbf{X}$, by either nonlinear iterative partial least squares (NIPALS) [74] or singular value decomposition (SVD) [75] $\mathbf{X}$, as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{X}} = \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{T}}\tilde{\mathbf{P}}^T = \left[\mathbf{T}\tilde{\mathbf{T}}\right]\left[\mathbf{P}\tilde{\mathbf{P}}\right]^T \tag{2.1}$$

where $\mathbf{T} \in \Re^{\mathbf{n\times a}}$ and $\mathbf{P} \in \Re^{\mathbf{m\times a}}$ are known as the score and loading matrices, respectively. The PCA projection reduces the original set of m variables to a principal components. The residual matrix $\tilde{\mathbf{X}}$ can be further decomposed into the product of residual scores $\tilde{\mathbf{T}}$ and residual loading $\tilde{\mathbf{P}}$ matrices. The column space of $\mathbf{P}$ is the principal component subspace (PCS) with dimension a. The column space of $\tilde{P}$ is the residual subspace (RS) with dimensions m-a. The PCA projection reduces the original set of m variables to a principal components. The decomposition is made such that $[\mathbf{T}\tilde{\mathbf{T}}]$ s orthogonal and $[\mathbf{P}\tilde{\mathbf{P}}]$ is orthonormal. The columns of $\mathbf{P}$ are the eigenvectors of the correlation matrix of the $\mathbf{X}$ variables associated with the a largest eigenvalues and forms a new orthogonal basis for the space spanned by $\mathbf{X}$. The remaining columns of $\mathbf{P}$ (e.g $\tilde{\mathbf{P}}$) correspond to the residual eigenvectors [7]. The p_i loading vectors (the columns of $\mathbf{P}$), often referred as principal components, are linear combinations of the original variables that explain the dominant directions of variability of the information contained in the original process data matrix $\mathbf{X}$ [73, 76]. The score vectors $\mathbf{t_i}$ are projections of the process variables contained in $\mathbf{X}$ onto the new basis vector $\mathbf{p_i}$.

2.1.3 Fault Detection

Once the PCA model has been determined based on historical data, new multivariate observations in a sample vector $\mathbf{x}$ can be referenced against this model. This is achieved by projecting these observations onto the plane defined by the PCA loading vectors to obtain their scores and residuals [6]. Traditionally, the projections or scores are plotted to reveal the relationships of the variables forming a new multivariate observation $\mathbf{x}$. In addition, to reveal data patterns, to identify clusters, and to determine the influence of different multivariate observations, two-dimensional plots of the most significant scores related to the dominant directions of variability (e.g. dominant eigenvectors) are used. In addition, an evaluation of the magnitudes of the loadings can serve as an indication of the relationships between the process variables contained in $\mathbf{X}$. For fault detection, Hotelling's T^2 [77] and SPE charts [15] are commonly used. The SPE index measures variability that breaks the normal process correlation [8], in other words, measures the variation in the residual subspace (RS), and is defined as follows:

$$SPE = \|\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i}\|^2 = \|\tilde{\mathbf{x}}\|^2 = \left\|\left(\mathbf{I} - \mathbf{P}\mathbf{P}^T\right)\mathbf{x}\right\|^2 \tag{2.2}$$

where $\hat{\mathbf{x}}_\mathbf{i}$ is the prediction of the multivariate observation $\mathbf{x}$. Assuming that $\mathbf{x}$ follows a multivariate normal distribution, Jackson and Mudholkar [15] developed a confidence limit for SPE. The process is considered normal if SPE $\leq \delta_\alpha^2$, where δ_α^2 denotes the upper control limit for SPE with a significance level α. An alternative way of determining the confidence limits for the SPE statistic is presented in [78]. The SPE statistic provides a way to test if the process data are outside the normal operating region. The SPE statistic is normally complemented by the use of the Hotelling's T^2 [77] statistic. The T^2 index measures the distance to the origin in the principal component subspace (PCS), i.e., provides an indication of abnormal variability in the space defined by the PCS.

$$T^2 = \mathbf{x}^T \mathbf{P} \mathbf{\Lambda}^{-1} \mathbf{P}^T \mathbf{x} \tag{2.3}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of the a largest eigenvalues of $\mathbf{X}\mathbf{X}^T$. The process is considered normal if $T^2 \leq T_\alpha^2$, where the upper control limit can be determined using the F-distribution [17]. It is important to point out that SPE and T^2 limits can also be determined empirically using validation data [32], which is often the case for industrial applications. Although T^2 and SPE indices are used in conjunction for process monitoring, as noted in [8], their role in process monitoring is asymmetric, i.e., both measure different situations in the process. The normal region defined by the T^2 statistic is usually much larger (e.g. capture normal process variations) than the one defined by the SPE (e.g. noise and unexplained variability). In general, a sample that only violates the upper control limit defined by the T^2 statistic, could indicate a fault but also a process change (e.g. away from the origin of the PCS), which is not a fault. On the other hand, the SPE statistic defines a tighter region of normal operation, and a sample that breaks the correlation structure defined in the model can easily exceed the SPE control limit. Thus giving a more reliable indication of the presence of a fault. This causes the SPE indexes to have lower chances of false alarms (i.e., type I errors) and missed detection rates (e.g. type II errors) compared to the T^2 [8]. Evidently, if a given set of process variables (e.g. quality variables) have a stationary nature (e.g. vary around an operating point), the use of the T^2 statistic is preferred [8, 17].

2.1.4 Fault Diagnosis

One of the most important aspects in process monitoring is fault diagnosis. As described before, traditionally plots of the scores and residuals together with the monitoring of the time series of T^2 indices and SPE indices are utilized for fault detection. However, with these tools, it is difficult to determine the source of the problem after it has been detected. In fault diagnosis, one is interested in pinpointing the variable or group of variables that are most related to the particular fault. After diagnosis, the actual problem can be eliminated and improvements in quality and yield can be observed. One of the most utilized solution for fault diagnosis based on multivariate data analysis techniques is contribution plots [5, 6, 9, 10, 34–36]. Contribution plots provide a way to connect the information provided by the monitoring index in the principal (e.g. T^2) or the residual (e.g. SPE) subspace to the actual process variables. In these plots, the contribution of the different variables to the given monitoring index is visualized. In this way, variable(s) that contribute the most to the fault are highlighted and related problems can be solved in a more oriented way. The contributions of each process variable to the SPE statistic at instant i are the m (e.g. the number of variables in the input space) elements of the residual $x_i - \hat{x}_i$ squared [34]. These m elements are plotted as bars to find the contributions of the variables to the SPE index at instant i. For T^2, the contributions of the different variables can be calculated in different ways [34, 79, 80]. Following [34], the contribution of the T^2 statistic is determined by rewriting the scores $\mathbf{t}$ as a sum of weighted data, with the loadings of each principal component direction used as weights.

$$\mathbf{t}_{id} = \sum_{j=1}^{m} (\mathbf{x}_{ij}\mathbf{p}_{jd}) \tag{2.4}$$

where p_{jd} is the p loading for variable j in dimension d (e.g. d goes from 1 to a, the number of retained principal components) for the m process variables included in the input space. The m elements represent the contribution of each variable to the score t_{id}. As with the contributions to SPE, a chart with bars of the m contributions to the t_{id} is utilized for fault diagnosis.

2.2 Dynamic Principal Component Analysis

In PCA, variables are assumed to be uncorrelated in time [29]. In other words, the observations at sampling instant k are assumed to be independent from observations in the past, i.e., at sampling instant $k - l_a$, where l_a denotes the time lag between measurements. As noted in [81], this assumption holds for processes where the sampling times are long. However, for dynamic processes with fast sampling, a method needs to take into account the serial correlations present in the data. In these type of processes, the process variables will vary around the operating point region based on their dynamics. This in turn will make the values of the variables at the current time, to depend on information of the past measurements of the same variable and other related variables (e.g., auto-correlation and cross-correlation). These characteristics of industrial processes often times make static PCA to fail. To cope with these aspects, dynamic extensions of PCA to monitor different types of processes have been proposed [29, 82–85]. In dynamic PCA (DPCA), static PCA is extended to apply for monitoring multivariate dynamic systems. This is achieved by developing a multivariate time series model of the data before the application of PCA. In a dynamic system, the current values of the process variables $\mathbf{X(k)}$ will depend on the past values (e.g. $\mathbf{X(k-1)}$, where k is the current sample index). Typical cases of dynamic models include, multivariate autoregressive (AR), multivariate autoregressive with exogenous inputs (ARX), AR or ARX time series models for processes with moving average (ARMA) [86]. The common factor of these methods is the augmentation of lagged data to the process variables to capture the dynamic characteristics of the process. In DPCA, both the dimensionality reduction and model extraction are done in a single step [29]. Two parameters are needed to be defined in DPCA, the number of components to retain, and the number of time lags l_a needed to characterize the dynamic of the process. Once these two parameters are defined, PCA is applied to the dynamic version of matrix $\mathbf{X}$ (e.g. $\mathbf{X} = [\mathbf{X(k)}\ \mathbf{X(k-1)}\ \mathbf{X(k-l_a)}]$. Typically, l_a is set to 1 or 2 representing a first and second order dynamic system, respectively [29, 81]. The order can be increased to have better approximations of nonlinear relationships. Similar to PCA, after the principal components are determined based on the dynamic version of the matrix $\mathbf{X}$, the Hotelling's T^2 statistics and SPE indexes are used for fault detection. For fault diagnosis, contribution plots of the T^2 and SPE statistics are typically utilized. This is achieved by summing up the contributions to T^2 and SPE of the variables at the different time lags [81]. In general, for processes where serial correlations are present, DPCA observes better performance than regular PCA [29, 81].

2.3 Kernel Principal Component Analysis

2.3.1 Motivation

PCA-based monitoring methods have been extremely successful in many applications. However, as PCA only considers the mean and the variance-covariance of variables to characterize the process, and it assumes a linear relationship among the process variables, it lacks the capability to provide higher order representations for non-Gaussian data. Complex chemical and biological processes are nonlinear in nature. As a consequence, the process data inherits a nonlinear behavior. The nonlinearity present in the process variables causes PCA to perform poorly [87]. Kernel PCA (KPCA) has emerged as a solution to address this problem [11, 88–93].

2.3.2 The KPCA Method

In KPCA the input data (e.g. $x_1, x_2, ..., x_N \in \mathbb{R}^m$) is first mapped into a higher-dimensional space via a nonlinear function, i.e., $\boldsymbol{\phi} : \mathbb{R}^m \rightarrow \mathscr{F}$. After the higher-dimensional nonlinear mapping, the variables in the feature space are more likely to vary linearly [94]. The higher-dimensional linear space is also known as feature space ($\mathscr{F}$) [11, 88, 89]. In KPCA, PCA is done to find eigenvectors $\mathbf{v}$ and eigenvalues λ of the covariance matrix $\mathbf{C}$ in the feature space $\mathscr{F}$

$$\mathbf{C} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\phi}(\mathbf{x_j})\boldsymbol{\phi}(\mathbf{x_j})^T \tag{2.5}$$

where the input process data $\mathbf{x_k} \in \mathbb{R}^m$, $k = 1, ..., N$ is zero mean and unit variance and $\boldsymbol{\phi}(\cdot)$ is a nonlinear mapping function that projects the input data to $\mathscr{F}$. To find the principal components, the eigenvalue problem has to be solved in $\mathscr{F}$

$$\lambda \mathbf{v} = \mathbf{C}\mathbf{v} \tag{2.6}$$

In Eq. (2.6), all solutions $\mathbf{v}$ with $\lambda \neq 0$ must lie in the span of $\boldsymbol{\phi}(\mathbf{x_1}), ...,$ $\boldsymbol{\phi}(\mathbf{x_N})$ with coefficients α_i $(i = 1 = 1, ..., N)$ such that [11, 88, 89]

$$\mathbf{v} = \sum_{i=1}^{N} \alpha_i \boldsymbol{\phi}(\mathbf{x_i}) \tag{2.7}$$

Let $\langle \mathbf{a}, \mathbf{b} \rangle$ denote the dot product between vectors $\mathbf{a}$ and $\mathbf{b}$, by defining a $\mathbf{K} \in \mathbb{R}^{N \times N}$ matrix with elements $[K]_{ij} = \langle \boldsymbol{\phi}(\mathbf{x_i})\ ,\ \boldsymbol{\phi}(\mathbf{x_j}) \rangle$, the eigenvalue decomposition in the feature space is simplified as follows, see [11, 88–90] for a detailed derivation.

$$\lambda \boldsymbol{\alpha} = \frac{1}{N}\mathbf{K}\boldsymbol{\alpha} \tag{2.8}$$

where $\boldsymbol{\alpha} = [\alpha_1 \alpha_2\ \cdots\ \alpha_N]^T$ are the eigenvectors with eigenvalues $\lambda_1\ \geq\ \lambda_2\ \geq\ \cdots\ \lambda_N$. As in regular PCA, the dimensionality of the problem in the feature space can be reduced by selecting only the first few a eigenvectors. The KPCA procedure requires the matrix $\mathbf{K}$ to be mean centered (e.g. $\overline{\mathbf{K}}$). The mean centered matrix $\overline{\mathbf{K}}$ is obtained as follows [11, 88, 89]

$$\overline{\mathbf{K}} = \mathbf{K} - \mathbf{KE} - \mathbf{EK} + \mathbf{EKE} \tag{2.9}$$

where

$$\mathbf{E} \quad = \quad \frac{1}{N}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{N\times N} \qquad (2.10)$$

To satisfy the eigenvalue normality constraint (e.g. $\|\mathbf{v}\|^{\mathbf{2}} = 1$), the coefficients α should be normalized during the KPCA decomposition (e.g. $\|\boldsymbol{\alpha}\|^2 = \frac{1}{N\lambda}$). The score $\mathbf{t}$ of a test vector $\mathbf{x}$ is obtained by projecting $\boldsymbol{\phi}(\mathbf{x})$ into the space defined by the principal components $\mathbf{v}_k$ in $\mathscr{F}$ [11, 88]

$$t_k = \langle \mathbf{v_k}, \boldsymbol{\phi}(\mathbf{x})\rangle = \sum_{i=1}^{N} \alpha_i^k \langle \boldsymbol{\phi}(\mathbf{x_i}), \boldsymbol{\phi}(\mathbf{x})\rangle \qquad (2.11)$$

where $k = 1, ..., a$ retained eigenvectors. In KPCA, *kernel functions* are introduced to compute the dot products in the feature space to solve the eigenvalue problem of Eq. (2.8) in the form of $k\langle \mathbf{a}, \mathbf{b}\rangle = \langle \boldsymbol{\phi}(\mathbf{a}), \boldsymbol{\phi}(\mathbf{b})\rangle$ [11, 89]. Some commonly applied kernel functions are listed below

- Linear kernel

$$k(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b} \qquad (2.12)$$

- Polynomial kernel

$$k(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T\mathbf{b} + d_1)^{d_2} \qquad (2.13)$$

- Gaussian kernel

$$k(\mathbf{a}, \mathbf{b}) = exp\left(-0.5\frac{\|\mathbf{a}-\mathbf{b}\|^2}{d_3^2}\right) \qquad (2.14)$$

- Sigmoid kernel

$$k(\mathbf{a}, \mathbf{b}) = tanh(d_4(\mathbf{a}^T\mathbf{b}) + d_5) \qquad (2.15)$$

where the d_i parameters ($i = 1, ..., 5$) are empirically defined to achieve a given mapping $\boldsymbol{\phi}$ and feature space $\mathscr{F}$ [11].

2.3.3 Fault Detection

Similar to PCA, for fault detection, the T^2 and SPE statistics are utilized as defined below [11, 88, 90, 91]

$$T^2 = [t_1, ..., t_l]\Lambda^{-1}[t_1, ..., t_l]^T \tag{2.16}$$

$$SPE = \sum_{j=1}^{N} \mathbf{t}_j^2 - \sum_{j=1}^{l} \mathbf{t}_j^2 \tag{2.17}$$

where t_k is calculated by Eq. (2.11) and Λ is a diagonal matrix containing the eigenvalues of the retained dimensions a after applying PCA in the feature space. As in PCA, the confidence limits of the monitoring indexes can be determined theoretically via F-distribution [95] and χ^2 distribution [6], respectively, or empirically using validation data. Once a KPCA model is obtained, the kernel $k_{te} \in \mathbb{R}^{1\times N}$ of a test vector $\mathbf{x_{te}} \in \mathbb{R}^m$ can be determined using the training data x_j by $[k_i]_j = [k_{te}(\mathbf{x}_{te}, \mathbf{x}_j)]$ [11]. Before projecting the test vector into the space spanned by the retained eigenvectors, the k_{te} kernel needs to be mean centered $\overline{k}_{te} = \mathbf{k}_{te} - \mathbf{E}_1\mathbf{K} - \mathbf{k}_{te}\mathbf{E} + \mathbf{E}_1\mathbf{K}\mathbf{E}$, where $\mathbf{E}_1 = \frac{1}{N}[1, ..., 1] \in \mathbb{R}^{1\times N}$ [11, 88]. The projection of the test vector is calculated by

$$t_k = \sum_{i=1}^{N} \alpha_i^k \overline{k}_{te}(\mathbf{x}_i, \mathbf{x}_{te}) \tag{2.18}$$

2.3.4 Fault Diagnosis

In regular PCA, the identification of the process variables that are most related to a particular fault is traditionally done by utilizing contribution plots of the monitoring indices T^2 and SPE. However, for KPCA this task is not straightforward. This is a consequence of the nonlinear mapping of the input space performed prior applying PCA in the feature space. In this work, the approach proposed by Choi et al. [11] based on the sensor reconstruction approach described in [96] is used for fault detection. In this approach, a fault index is defined as a function of two type of squared errors (SPE) [90, 96]:

$$\zeta_i = \frac{\left\|\tilde{\mathbf{x}}^i - \hat{\tilde{\mathbf{x}}}^i\right\|^2}{\left\|\mathbf{x}^i - \hat{\mathbf{x}}^i\right\|^2} \tag{2.19}$$

where $\mathbf{x}$ is a multivariate observation from the input space, $\hat{\mathbf{x}}$ is the estimate of $\mathbf{x}$, $\tilde{\mathbf{x}}^i$ is identical to $\mathbf{x}$ except for the i^{th} entry which is replaced by its reconstructed value, and $\hat{\tilde{\mathbf{x}}}^i$ is its estimate. In other words, the denominator in Eq. (2.19) represents the SPE of the input variable space, while the numerator represents the SPE of the reconstructed $\tilde{\mathbf{x}}$ in the input space. In this approach, the trend of the fault index ζ_i for the different variables is monitored to diagnose variables that contribute the most to the fault. In this approach, when a faulty variable is reconstructed, the corresponding SPE will reduce or not increase considerably. Thus a significant reduction in the fault index ζ_i of the i^{th} variable is observed, which indicates a variable with an abnormal pattern. In other words, if a fault is detected, the denominator in Eq. (2.19) increases, while the numerator for a reconstructed faulty variable decreases or not increases considerably compared to the numerator of Eq. (2.19) when the faulty variable is not reconstructed. In this work, robust reconstruction is achieved by using the algorithm proposed by Mika et al., [92, 93]. To achieve an optimal reconstructed value $\mathbf{z}$, the kernel function is restricted to the Gaussian form of Eq. (2.14), and the following iterative approach is implemented to compute $\mathbf{z}$ [90]

$$\mathbf{z}(t) = \frac{\sum_{i=1}^{N} \beta_i K(\mathbf{x}_i), \mathbf{z}(t-1))\mathbf{x}_i}{\sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{z}(t-1)))} \tag{2.20}$$

where $\mathbf{z}(0) = \mathbf{x}$ and $\beta_i = \sum_{k=1}^{l} t_k \alpha_i^k$. In this way an estimate $\mathbf{z}$ of the testing vector $\mathbf{x}$ is obtained by reconstructing its projection into the principal component subspace.

For online implementation of KPCA, there are several factors to be considered. First, the dimension of the feature space need to be determined. When the training data are large, the computation time for computing the matrix $\mathbf{K}$ increases significantly (e.g. $\mathbf{K} \in \mathbb{R}^{N \times N}$).

Second, the number of retained eigenvectors after performing PCA in the hyperplane needs to be determined. Typically, a cross-validation approach is used [90]. In addition, the parameters d_i involved in the computation of the kernel matrix $\mathbf{K}$ need to be determined. These parameters, for instance d_3 for the Gaussian kernel, not only play an important role in the nonlinear mapping, but also in the monitoring and diagnosis performance. For a Gaussian kernel, typically d_3 in Eq. (2.14) is set to be a product of an ad-hoc constant times the number of variables used in the monitoring approach (e.g. $5m$ or $10m$, where m is the number or variables in the input space) [11, 90].

2.4 Independent Component Analysis

2.4.1 Motivation

The presence of unmeasured disturbances, correlations and noise hides the valuable factors that govern industrial processes. Revealing these hidden factors can be used to better characterize a process. In PCA, this is achieved by projecting the input data onto a lower-dimensional space that captures most of the variance and accounts for correlations among variables. Since the objective of PCA is to decorrelate the scores (e.g. projections) and only consider up to second-order statistics to characterize a process, it lacks the capability of represent processes where nonlinearities are prevalent (e.g. non-Gaussian distribution of input variables). Independent component analysis (ICA) is a recently developed technique [97–102] that by satisfying formal statistical independence and implicitly using higher-order statistics, it decomposes the process data into linear combinations of statistically independent components (ICs) that contain the main governing characteristics of the process. Applications of ICA have been reported in several fields of engineering, some examples are signal processing, machine vibration analysis, radio communication, infrared optical source separation etc. [98, 103].

2.4.2 The ICA Method

In ICA, x_i zero-mean measured variables $(i = 1, ..., m)$ are expressed as a linear combination of s_j statistically independent components $(j = 1, ..., d \leq m)$. Several algorithms have been proposed to estimate independent components [104, 105]. In this work, the algorithm developed in [102] is briefly reviewed. Let $\mathbf{x} = [x_1, ..., x_m]^T$ denote a column vector of observations of process variables, and $\mathbf{s} = [s_1, ..., s_d]^T$ a column vector of independent components, to be related by a mixing matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ of coefficients a_{ij} (e.g. $x_i = a_{i1}s_1 + a_{i2}s_2 + ... + a_{id}s_d$) $\forall\ i$, as follows

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2.21}$$

The model of Eq. (2.21) is referred as ICA model [102]. In Eq. (2.21), both the mixing matrix $\mathbf{A}$ and and independent components $\mathbf{s}$ must be estimated based on the vector of observations $\mathbf{x}$. The starting point of the estimation process is based on the assumption of statistically independence of the the components s_i [102], i.e, the joint probability function can be factorized in individual probability functions. For ICA to be possible, the independent components are restricted to be non-Gaussian [102], since a joint probability function of Gaussian, uncorrelated variables would not provide enough information to identify dominant directions of the mixing matrix $\mathbf{A}$. Assuming that $d = m$ the mixing matrix $\mathbf{A}$ is square. After the estimation of $\mathbf{A}$, its inverse ($W = A^{-1}$, the demixing matrix) can be computed, and the independent components can be obtained by [102, 105]:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{2.22}$$

Before ICA is applied, sphering (or whitening) is performed to remove correlation and cross-correlation between the random input variables. Whitening can be achieved by performing regular PCA on the correlation matrix $\mathbf{R}$, where $\mathbf{R} = E\left\{\mathbf{x}\mathbf{x}^T\right\}$, with vector $\mathbf{x}$ zero mean and unit variance, and $E\{\bullet\}$ denotes the expected value. The decomposition of $\mathbf{R}$ by singular value decomposition (SVD) is as follows

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{2.23}$$

where $\mathbf{U}$ is an orthogonal matrix of the eigenvectors corresponding to the diagonal matrix $\mathbf{\Lambda}$ of eigenvalues of $\mathbf{R}$. A whitening transformation is achieved by

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{x} \tag{2.24}$$

where $\mathbf{\Lambda}^{-1/2} = diag(\lambda_1^{-1/2}, ..., \lambda_d^{-1/2})$. After whitening, the mixing matrix $\mathbf{A}$ is transformed to a new one $\mathbf{B}$ with better properties

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{x} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{A}\mathbf{s} = \mathbf{B}\mathbf{s} \tag{2.25}$$

where $\mathbf{B}$ is an orthogonal matrix (e.g. $E\left\{\mathbf{z}\mathbf{z}^T\right\} = \mathbf{B}E\left\{\mathbf{s}\mathbf{s}^T\right\}\mathbf{B}^T = \mathbf{B}\mathbf{B}^T = I$, $E\left\{\mathbf{s}\mathbf{s}^T\right\} = \mathbf{I}$) [33, 102, 106]. Thus, the estimation problem is reduced from determining d^2 parameters of matrix $\mathbf{A}$ to $\frac{d(d-1)}{2}$ parameters of the orthogonal matrix $\mathbf{B}$ [102]. The estimates of $\mathbf{s}$ (e.g. $\hat{\mathbf{s}}$) are determined from Eq. (2.25) by

$$\hat{\mathbf{s}} = \mathbf{B}^T\mathbf{z} = \mathbf{B}^T\Lambda^{-1/2}\mathbf{U}^T\mathbf{x} \tag{2.26}$$

The relationship of the demixing matrix $\mathbf{W}$ and $\mathbf{B}$ is obtained from Eq. (2.22) and Eq. (2.26)

$$\mathbf{W} = \mathbf{B}^T\Lambda^{-1/2}\mathbf{U}^T \tag{2.27}$$

A robust and reliable way of determining the ICs is achieved through the use of the FastICA algorithm [100–102, 105]. The key of the algorithm is to find a vector b_i (e.g. a column of $\mathbf{B}$ that maximizes non-Gaussianity of $\hat{s}_i = (\mathbf{b}_i\mathbf{z})$). In [102], it is shown that a measure of non-Gaussianity is a direct indicator of statistical independence. A typical way of measuring departures from Gaussianity is the use of the absolute value of the fourth-order cumulant or kurtosis. As kurtosis is affected by outliers, other measures of non-Gaussianity have been developed. Non-Gaussianity indicators based on the differential entropy such as negentropy are normally utilized. The entropy of a random variable gives an indication of unpredictability and lack of structure [102]. Thus Gaussian variables have the largest entropy among all random variables of equal variance [102]. Negentropy J is defined as follows

$$J(y) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{2.28}$$

where H indicates the differential entropy, $\mathbf{y}_{gauss}$ is a Gaussian random variable of the same variance as $\mathbf{y}$. The estimation of J is difficult to determine in practice. As a consequence, approximations of J have been developed.

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2 \tag{2.29}$$

where G is a non-quadratic function, y is assumed to be zero mean and unit variance, and v is a Gaussian variable of zero mean and unit variance. To obtain robust approximations of negentropy, the following contrast functions are suggested [102, 105]

$$G_1(u) = \frac{1}{a_1} log\, cosh(a_1 u) \tag{2.30}$$

$$G_2(u) = exp\left(-\frac{a_2 u^2}{2}\right) \tag{2.31}$$

where $1 \leq a_1 \leq 2$ and $a_2 \approx 1$. In this work, G_1 is the contrast function selected as indicator of non-Gaussianity for the FastICA algorithm. Based on this approximation, the FastICA algorithm calculates the columns of matrix $\mathbf{B}$, which allows the identification of the corresponding ICs (e.g. Eq. (2.26)). The details of the algorithm can be found in [100–102, 105]. A Matlab$^{\text{TM}}$ implementation of the FastICA algorithm is available at http://www.cis.hut.fi/projects/ica/fastica/.

2.4.3 Fault Detection

The application of process monitoring based on ICA have been studied by several investigators [33, 103, 106–113]. In this work, the approach found in [106] is followed. For fault detection, under the assumption that the number of variables (m) is equal to the number of independent components (d), the demixing matrix $\mathbf{W}$ and estimates of independent components $\hat{\mathbf{S}}$ for normal operation are found by the FastICA algorithm. In addition, the matrices $\mathbf{B}$, $\Lambda^{-1/2}\mathbf{U}^T$, and $\mathbf{A}$ are obtained from the whitening step. In ICA, the ordering of independent components is not determined by the variance of the variables as in PCA [107]. Several criteria exist to order the independent components. These include, extent of non-Gaussianity, L_∞ *norm*, Euclidean norm, and mean squared error of reconstruction [114–117]. In [106], the Euclidean norm approach is selected to order the columns of the demixing matrix $\mathbf{W}$ and to sort the ICs. In addition, a SCREE plot of the norm of the Euclidean norm of the columns of $\mathbf{W}$ versus the number of IC is used to reduce the dimensionality of the problem, i.e., select only the columns of $\mathbf{W}$ with the most dominant Euclidean norm. The selected columns of $\mathbf{W}$ are assumed to have the greatest effect in the variation of the estimates of the independent components ($\hat{\mathbf{S}}$). Thus the matrix $\mathbf{W}$ is partitioned into two, the dominant part $\mathbf{W}_d$ (i.e. selected rows of $\mathbf{W}$) and reduced part $\mathbf{W}_e$ (i.e., remaining columns of $\mathbf{W}$). Using the selected indices, the equivalent is done for matrix $\mathbf{B}$ to generate matrices $\mathbf{B}_d$ and $\mathbf{B}_e$. For a new multivariate observation $\mathbf{x}_{new}$, the new dominant and reduced independent data vectors can be computed after the whitening transformation, i.e., $\hat{\mathbf{s}}_{new,d} = \mathbf{W}_d\mathbf{x}_{new}$ and $\hat{\mathbf{s}}_{new,e} = \mathbf{W}_e\mathbf{x}_{new}$, respectively.

In [106], three different statistics for fault detection are defined

$$I^2 = \hat{\mathbf{s}}_{new,d}^T\hat{\mathbf{s}}_{new,d} \tag{2.32}$$

$$I_e^2 = \hat{\mathbf{s}}_{new,e}^T \hat{\mathbf{s}}_{new,e} \tag{2.33}$$

$$SPE = (\mathbf{x}_{new} - \hat{\mathbf{x}}_{new})^T (\mathbf{x}_{new} - \hat{\mathbf{x}}_{new}) \tag{2.34}$$

where $\hat{\mathbf{x}} = (\Lambda^{-1/2}\mathbf{U}^T)^{-1}\mathbf{B}_d\mathbf{W}_d\mathbf{x}_{new}$. In Eq. (2.32), I^2 gives an indication of the variability observed in the dominant independent component subspace. I_e^2 in Eq. (2.33), accounts for errors in the selection of the dominant part of the independent component subspace [106]. SPE in Eq. (2.34) accounts for the monitoring of the unmodeled part of the common cause of variation in the input data. In [107], the thresholds for the different monitoring statistics are determined based on a kernel density estimation to cope with the non-Gaussianity of the ICs. For the performance comparisons in subsequent chapters and to provide a fair comparison of different methods, the thresholds are determined empirically using validation data.

2.4.4 Fault Diagnosis

For fault diagnosis (e.g. to reveal the variables that are most related to the fault), Lee et al. [107] develop contribution plots for the different monitoring statistics, i.e., Equations (2.32) - (2.34). The variable contributions of $\mathbf{x_{new}}$ for I^2, I_e^2, and SPE are determined as follows

$$\mathbf{x}_{cd} = \frac{\mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}}_{new,d}}{||\mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}}_{new,d}||} \, ||\hat{\mathbf{s}}_{new,d}|| \tag{2.35}$$

$$\mathbf{x}_{ce} = \frac{\mathbf{Q}^{-1}\mathbf{B}_e\hat{\mathbf{s}}_{new,e}}{||\mathbf{Q}^{-1}\mathbf{B}_e\hat{\mathbf{s}}_{new,e}||} \, ||\hat{\mathbf{s}}_{new,e}|| \tag{2.36}$$

$$\mathbf{x}_{c,spe} = \mathbf{e} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new} \tag{2.37}$$

where $\mathbf{Q} = (\Lambda^{-1/2}\mathbf{U}^T)$. The contribution plots resulting from the application of Equations (2.35) - (2.37) can have negative and positive contributions. As with the contribution plots of regular PCA, the magnitude of the contribution indicates what variable or group of variables are most related to the fault.

Chapter 3

Statistics Pattern Analysis: A Novel Process Monitoring Framework

In this section, the development of a novel process monitoring method is presented. Statistics pattern analysis (SPA) is a new multivariate process monitoring method recently developed [32, 59–61]. In this dissertation, the SPA framework properties for fault detection and diagnosis are developed, and its application is illustrated.

3.1 Motivation

The SPA framework was originated to cope with the challenges in semiconductor manufacturing [59], i.e., a batch process. Traditional methods for monitoring batch processes, including multiway PCA (MPCA) and multiway PLS [6, 10, 28, 80, 118, 119], are faced with the difficulty of handling the characteristics of batch data, such as unequal batch and/or step length, unsynchronized or misaligned batch trajectory, and multi-modal batch trajectory distribution. These characteristics make traditional process monitoring methods to fail or need additional preprocessing steps. These steps are usually done off-line and are difficult to automate [59]. Similar to continuous processes, batch process monitoring methods make use of multivariate data analysis techniques to extract the correlation among different variables and to reduce the dimensionality of the problem (e.g. MPCA). However, the characteristics of certain processes, including the ones present in semiconductor manufacturing, make the data to show significant deviations from a multivariate normal distribution. As discussed previously, these characteristics make PCA or PLS monitoring performance to deteriorate. In addition, including several preprocessing steps increases the difficulty for establishing and maintaining these models. For example, in a high-mix semiconductor fabrication site, the number of models is in the range of several thousands to tens of thousands [59, 120]. This occurs due to the diversity of products manufactured in semiconductor fabrication site, where a single machine can produce different type of products. As a consequence, model building and maintenance are much more labor intensive than in a typical chemical plant. In [59], the negative effect of adding pre-processing steps is illustrated by comparing the monitoring performance of traditional monitoring methods (e.g. MPCA and MPLS), as a function of the pre-processing steps. The study confirms that traditional methods tend to deteriorate their effectiveness as the number of pre-processing steps increase.

3.2 The SPA Method

Unlike traditional process monitoring methods that monitor the different process variables to perform fault detection and diagnosis, the SPA-based monitoring makes use of the "statistics" of the process variables in the batch to perform fault detection and diagnosis. In this way, the method not only eliminates the need of all data preprocessing steps in monitoring batch processes, but also provides superior performance [59]. Figure 3.1 and 3.2 illustrate the steps involved for batch process monitoring for MPCA (e.g. traditional method) and SPA, respectively. In SPA, a set of statistics patterns (SPs), which capture the characteristics of the process variables in a batch, such as mean, and variance-covariance are used as an input to perform fault detection. In SPA, a statistic pattern (SP) is defined as a collection of different statistics calculated from a batch. Since batches are processed in the same equipment, the governing physical/chemical mechanisms, such as mass transport, kinetics and thermodynamics are the same. The SPs of "in-control" batches will follow a certain pattern, whereas abnormal SPs would exhibit some deviation from the normal behavior. As illustrated in Figure 3.2, after assembling a matrix of SPs, the second step, called dissimilarity quantification step, is performed. In this step, dissimilarities among SPs are measured and confidence limits are determined to distinguish between abnormal and normal batches. For a new batch, if its dissimilarity index is below the confidence limit, it is considered normal;

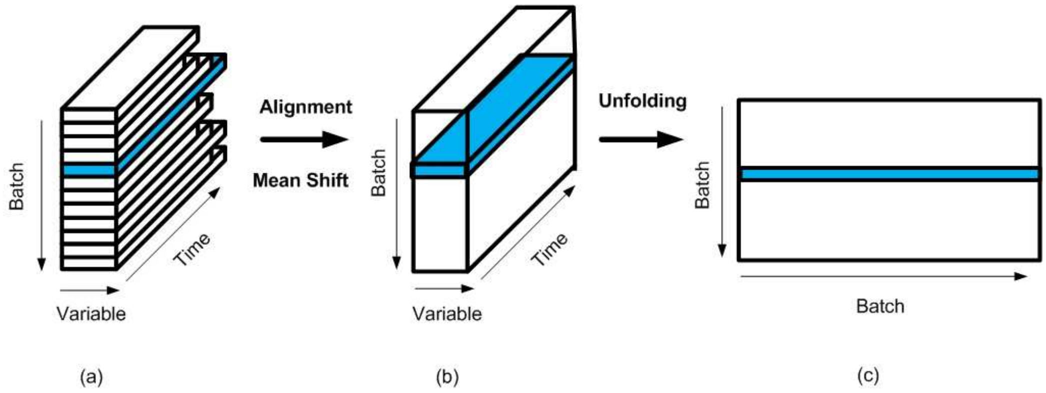


Figure 3.1: Illustration of steps involved in multivay PCA (MPCA). (a) Original batch records with unequal length; (b) 3-dimensional array after trajectory alignment and mean shifting; (c) Batch information for MPCA after unfolding (e.g. a matrix).

otherwise, it is categorized as faulty. Continuous rather than batch processes are common in chemical and petrochemical industries. As a consequence, the SPA framework was extended to monitor continuous processes [32, 32, 60, 61]. For a continuous process, the definition of statistic pattern (SP) varies. A SP for a continuous process is a collection of various statistics calculated from a window (or segment) of the process variables [32]. These statistics capture the characteristics of each individual variable (such as mean, variance, kurtosis and skewness), the interactions among different variables (such as correlation), as well as process dynamics (such as auto-correlation and cross-correlation). After the SPs are computed from the training data, the dissimilarities among the training SPs are quantified to determine an upper control limit of the detection index [32]. Figure 3.3 illustrates the two steps in SPA for monitoring continuous processes.

The following sections provide a brief summary of the characteristics and elements constituting the steps involved in SPA-based monitoring.

3.3 Fault Detection

In this section, the main characteristics of the two steps for fault detection in SPA are described.

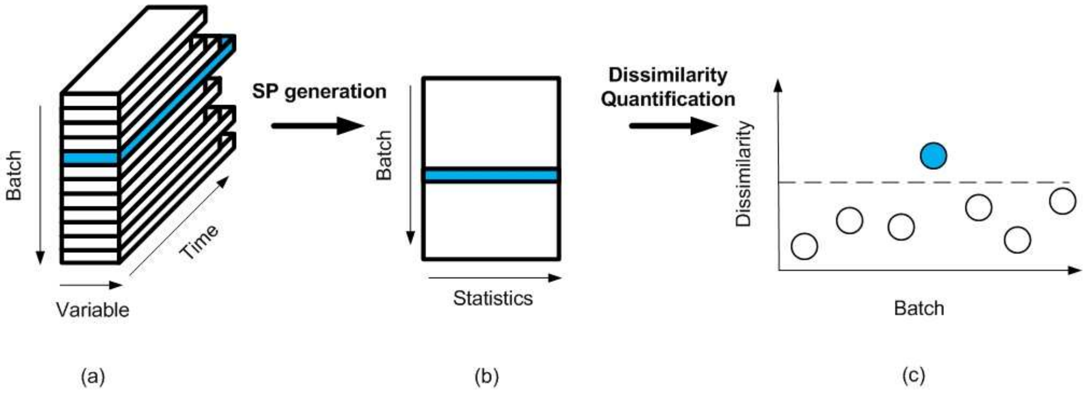


Figure 3.2: Schematic of the SPA framework for fault detection in batch processes. (a) Original batch records with unequal length; (b) SP generation (e.g. various batch statistics); (c) fault detection by dissimilarity quantification.

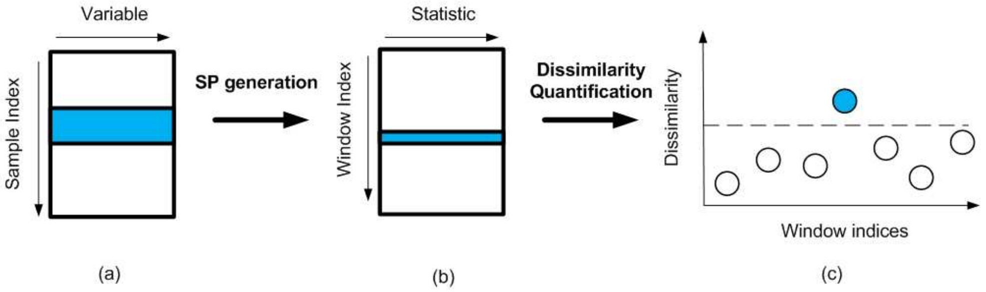


Figure 3.3: Schematic of the window-based SPA method for monitoring continuous processes: (a) original process data; (b) computed statistics patterns; (c) fault detection by dissimilarity quantification. A block or a window of process variables shaded in (a) is used to generate an SP shaded in (b). SPs are used to generate a point of dissimilarity measure (e.g. shaded in (c)) to perform fault detection.

3.3.1 Statistic patterns generation

Let the matrix $\mathbf{X}$ represent the collection of information of process variables (e.g. from a window for continuous processes or from a batch for batch processes), as shown below:

$$\mathbf{X} = [\mathbf{x}_1 \; \mathbf{x}_2 \; \mathbf{x}_3 \; \cdots \; \mathbf{x}_m]$$

$$X = \begin{bmatrix} \mathbf{x}_1(k-w+1) & \mathbf{x}_2(k-w+1) & \mathbf{x}_3(k-w+1) & \cdots & \mathbf{x}_m(k-w+1) \\ \mathbf{x}_1(k-w+2) & \mathbf{x}_2(k-w+2) & \mathbf{x}_3(k-w+2) & \cdots & \mathbf{x}_m(k-w+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1(k) & \mathbf{x}_2(k) & \mathbf{x}_3(k) & \cdots & \mathbf{x}_m(k) \end{bmatrix} \tag{3.1}$$

where w represents the window width for continuous processes or available samples in a batch, k is the sample index of the last measurement (most recent or last sample in a batch) in the window (or the batch). As previously discussed, a SP consists of different statistics calculated from the group of data. Generally, statistics can be lumped in three different categories:

$$\mathbf{S} = [\ \boldsymbol{\mu} \mid \boldsymbol{\Sigma} \mid \boldsymbol{\Xi}\ ] \tag{3.2}$$

$\boldsymbol{\mu}$ in Equation (3.2) denotes the first order statistics, i.e., variable means, whose elements are calculated from the data contained in the window (or batch). $\boldsymbol{\Sigma}$ in Equation (3.2) denotes the second-order statistics, which include variance, correlation, auto-correlation, and cross-correlation of different variables. $\boldsymbol{\Xi}$ in Equation (3.2) denotes the higher-order statistics, including skewness, kurtosis, and other higher-order moments. Higher-order statistics (HOS) measure the extent of nonlinearity and quantify the non-Gaussianity of the probability distribution of the process variables [32, 121]. Specifically, skewness measures the asymmetry and kurtosis measures the "peakedness" of the process variable distribution. HOS' for variables that follow a normal distribution have a value of zero. After matrix $\mathbf{S}$ has been formed, the final SP is generated by vectorizing $\mathbf{S}$ into a row vector. Note that by the symmetry of groups of statistics such as covariance (e.g. symmetric matrix), the dimension of the SP can be reduced. Once the training SPs have been lumped into a matrix, the dissimilarity quantification step can be performed.

3.3.2 Dissimilarity Quantification

After the SP matrix is obtained for the training data, a modeling stage is followed. This model is used to perform a quantification of dissimilarities of training and testing SPs. Dissimilarities can be quantified by measuring distances or angles between objects. In the case of PCA, distance-based metrics such as T^2 and SPE are used. In SPA, PCA is chosen to craft a model relating the different SPs in the training data (e.g. normal operating data). Note that the framework is not limited to the use of PCA as this method is just one way to determine dissimilarities among different SPs. [122]. The major difference between the traditional PCA-based and the proposed SPA-based fault detection methods is that PCA monitors the process variables, while SPA monitors various statistics of the process variables. In other words, in PCA singular value decomposition (SVD) is applied to the original process variables to build the model for normal operation, and the new measurements of the process variables are projected onto the PCA model to perform fault detection; while in SPA, SVD is applied to various statistics of the process variables to build the model, and the statistics calculated using the new measurements are projected onto the model to perform fault detection. In this way, different statistics that capture the different characteristics of the process can be selected to build the model for normal process operation, and various higher-order statistics can be utilized explicitly. As PCA is used to craft a model for normal operation, fault detection indices such as T^2 and SPE are also applicable. To distinguish the PCA-based indices from the SPA-based indices, in this work, D_p and D_r indices are defined to represent the T^2 and SPE indices monitoring the principal component subspace and residual subspace, respectively. As in PCA, the process is considered within the normal operating region if both dissimilarities indices are below their thresholds calculated at the α significance level. Before modeling, the SPs from training and testing data have to be normalized (zero mean and unit variance), to avoid the influence in the model of differences in scaling (i.e. units of measurement) and operating point differences (i.e. different mean values) among variables. In addition, the thresholds for D_p and D_r can be obtained using analytical expressions based on normality assumptions [8, 15], or based on an empirical method using the training data [123, 124].

3.3.3 Why SPA works?

In SPA, PCA is applied to quantify the dissimilarities among statistic patterns (SPs). As a result, the fundamental assumptions of PCA have to be satisfied, i.e., multivariate Gaussian distribution and independent samples. The major claim in SPA is that although the process variables may show deviations from a multivariate Gaussian distribution, the statistics of these process variables satisfy a multivariate Gaussian distribution much better than the process variables, and their projections in the principal component subspace, i.e., the scores. In other words, the SPs follow a multivariate Gaussian distribution asymptotically. This enables SPA to provide superior performance compared to applying PCA directly to the input data [59]. This fact can be proved by the classical central limit theorem (CLT), which states that "whatever the distribution of the independent variables ξ_v - subject to certain very general conditions - the sum $\xi = \xi_1 + \cdots + \xi_n$ is asymptotically Gaussian $(m,\ \sigma)$ [125], where

$$m = m_1 + m_2 + \cdots + m_n \tag{3.3}$$

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2 \tag{3.4}$$

where m_v and σ_v are the mean and standard deviation of σ_v". Among the many extensions of the classical CLT, extensions relevant to this work are provided in [126, 127], where the requirements of the independency among different random variables is relaxed. In the following section, the SPA framework and its properties are illustrated to elucidate the previous discussions.

3.4 Illustrative Example

In this section an illustrative example of the application of SPA is presented. As discussed previously, the major difference and advantage of SPA is the use of the statistics of the process variables rather than the process variables themselves for process monitoring. The success of SPA is based from the fact that although the process variables may have a non-Gaussian distribution, due to the central limit theorem (CLT), the statistics of the process variables follow a Gaussian distribution asymptotically. This feature enables the dissimilarity quantification based on PCA to truly capture the main characteristics of the process variables, thus providing a better monitoring performance. In this section the application and success of SPA is elucidated by using a simple illustrative example.

In this example, the process consist of two variables behaving according to the following nonlinear relationship

$$x_2 = x_1^2 + e \tag{3.5}$$

where x_1 is a random variable taken from a uniform distribution with values between $[-0.5\ 0.5]$ and $e \in N(0, 0.02^2)$. A normal data set consisting of $15,000$ samples is simulated, $10,000$ samples are used for training, and the rest of the samples are used for validation. Two data sets of $5,000$ samples each, corresponding to two groups of faults (A and B) are simulated as follows:

Fault A

$$x_1 = e_1 \tag{3.6}$$

$$x_2 = 0.15 + e_2 \tag{3.7}$$

Fault B

$$x_1 = -0.6 + e_{1_a} \tag{3.8}$$

$$x_2 = 0.1 + e_{2_a} \tag{3.9}$$

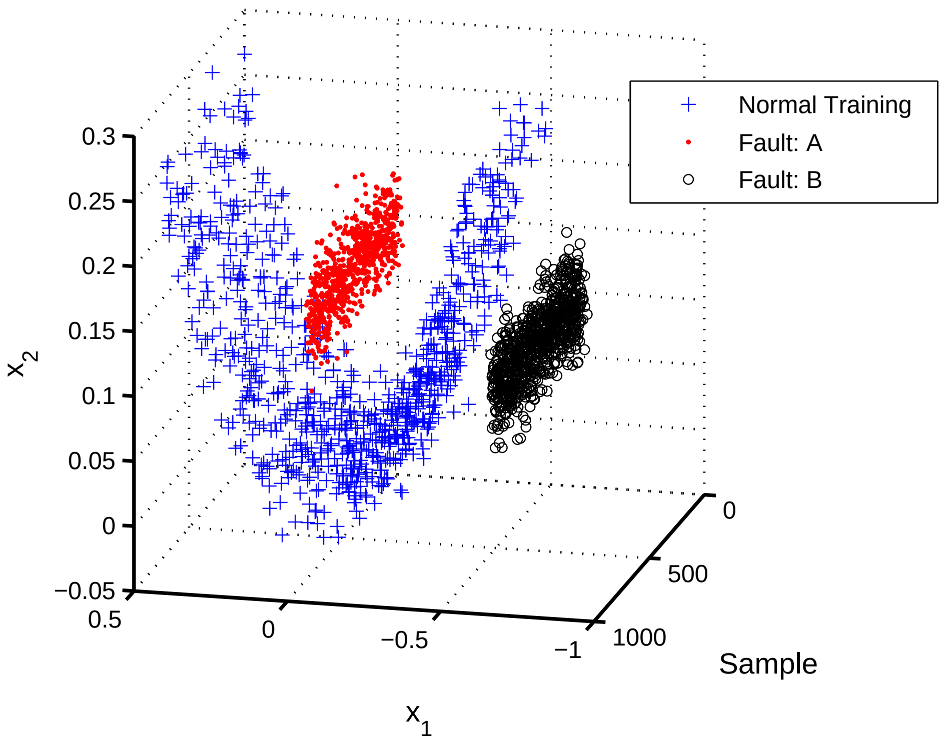


Figure 3.4: Scatter plot of the process variables for the nonlinear illustrative example

where e_1 and e_{1_a} are two different white noise realizations taken from a normal distribution with variance $\sigma^2 = 0.02^2$. Similarly, e_2 and e_{2_a} are two different white noise realizations with variance $\sigma^2 = 0.01^2$. For each fault data set, the first 100 samples are normal samples, and the remaining samples $(4,900)$ are affected by the fault. Figure 3.4 shows the scatter plot of some samples taken from the different data sets. It is clear from Figure 3.4 that the two faults do not correspond to the behavior of the normal operating data. For comparison, the PCA and SPA methods are used to detect the faults A and B. For PCA, two principal components are selected to capture the variability of the process. This causes only the T^2 index to be applicable for fault detection (e.g. to monitor the principal component subspace). For the SPA method, a nonoverlapping window width of 50 samples is used for training and validating. For testing, (i.e., the faulty data) after the first window is assembled, the window is shifted by one sample to reduce the detection delay. Since this process is static and there is not a strong correlation between x_1 and x_2, no correlation

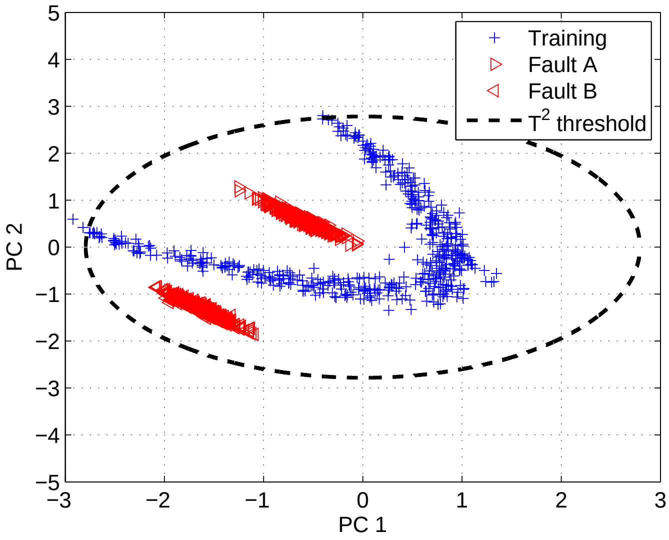
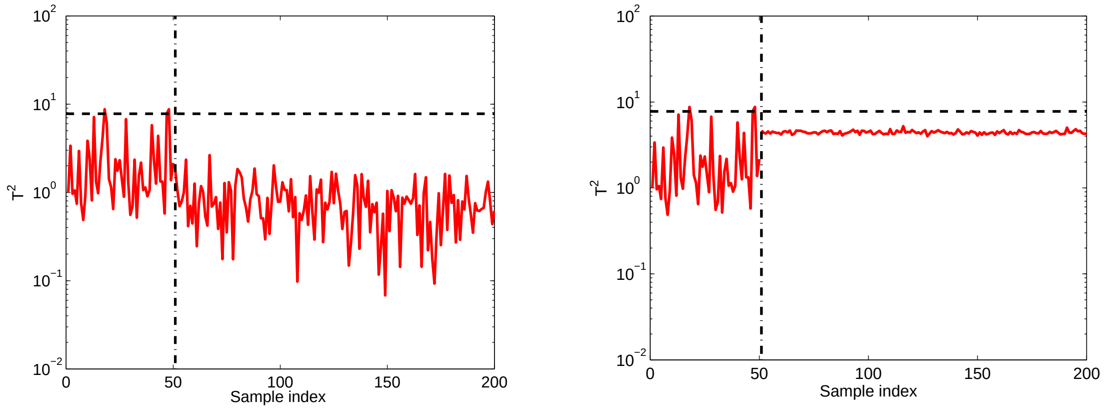


Figure 3.5: Scores of normal and faulty samples for PCA with the T^2 threshold calculated at the 99% confidence level (dashed-line).

coefficients, auto- or cross- correlation coefficients are used. The statistics that are selected to characterize the process are mean, standard deviation, skewness, and kurtosis (i.e. 8 statistics). Based on cross-validation, four principal components are selected to capture the main characteristics of the process. For PCA, Figure 3.5 shows that the scores corresponding to the faults A and B are located well within the normal operation region defined by the T^2 index. Therefore it is expected that this fault cannot be detected by the T^2 index. This is illustrated in Figure 3.6, where the T^2 indices fail to detect both faults. The PCA method failure to detect these faults occurs as a result of the non-Gaussian multivariate distribution of the process data, which is caused by their nonlinear relationship. This is illustrated in Figure 3.7, where the multivariate distribution of x_1 and x_2 is evidently non-Gaussian. The nonlinearity is then inherited by the process variables projected onto the model defined by the principal directions. As previously discussed, due to the central limit theorem, the statistics of the process variables follow a normal distribution asymptotically. Figure 3.8 shows the multivariate distribution of pairs of different statistics (e.g. mean x_1 - mean x_2). Both, the density histogram and the intensity map of the different pairs of statistics



(a)

(b)

Figure 3.6: T^2 of testing data. (a) Fault A; (b) Fault B. The dashed line indicates the threshold on the T^2 at the 99% confidence level. The dashed-dotted line indicates the fault onset.

corroborate the central limit theorem, with all of the pairs showing a multivariate distribution very close to a pure multivariate Gaussian distribution. It is important to point out that as the number of SPs increases (i.e., the number of samples), the multivariate distribution of statistics approach a multivariate Gaussian distribution asymptotically, i.e., they would eventually completely resemble a multivariate Gaussian distribution. By using the SP matrix for monitoring instead of the matrix of process variables for this nonlinear example, the main requirements of PCA, i.e., multivariate Gaussian distribution and independent scores are satisfied. As a consequence, the scores obtained by applying PCA to the SP matrix for normal and faulty data (Figure 3.9) observe a defined separation among each other, with the scores corresponding to faulty data located outside of the normal operating region defined by the D_p threshold. Consequently, both faults are expected to be detected by SPA. This illustrated in Figure 3.10, where both faults are detected by the D_p and D_r indexes in SPA.

3.5 Fault Diagnosis

After a fault is detected by one or more fault detection indices exceeding their control limits, it is desirable to perform fault diagnosis to identify the root cause of the fault. For

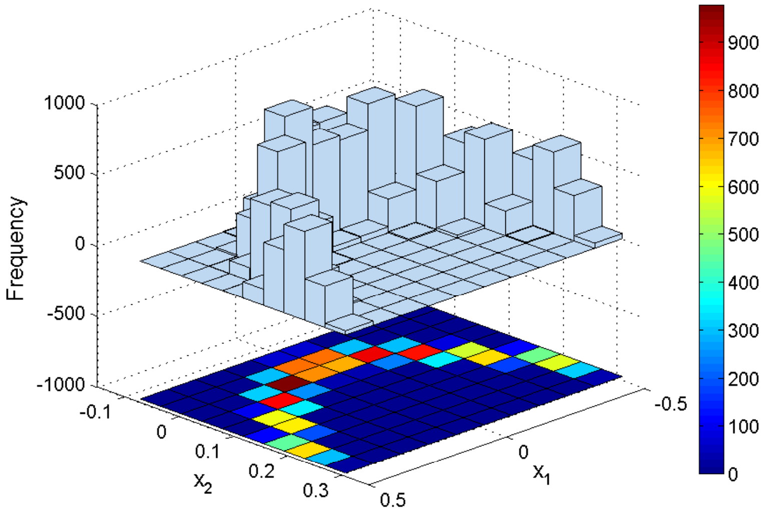


Figure 3.7: Density histogram and intensity map for the multivariate distribution of x_1 and x_2.

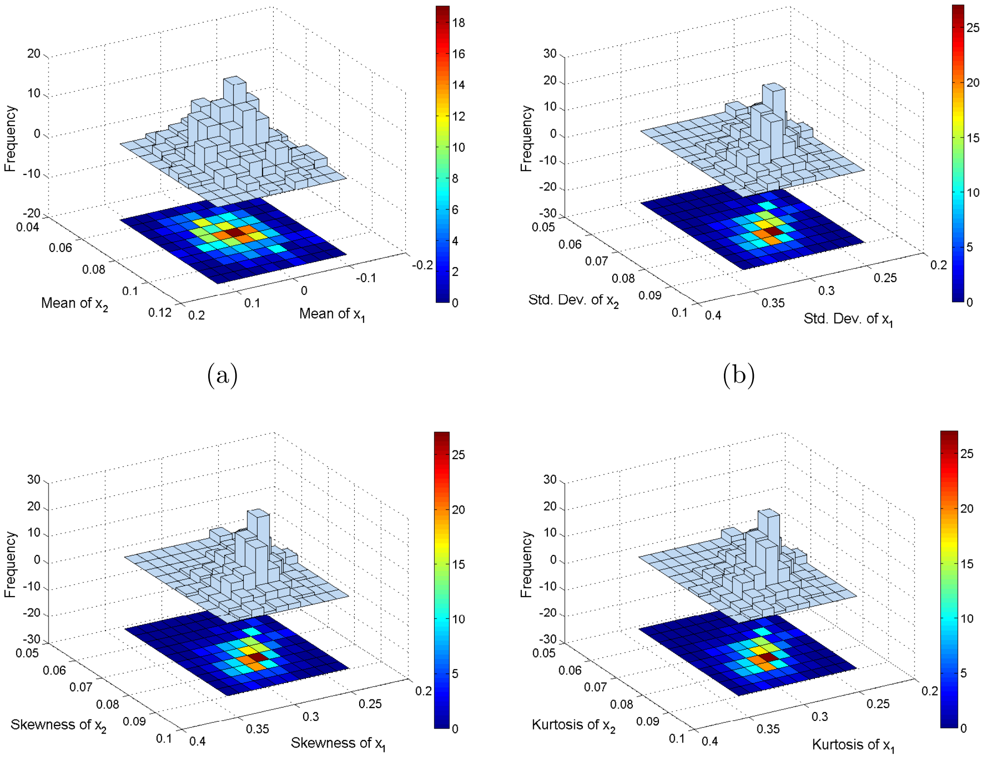
PCA-based fault detection methods, contribution plot is the most commonly applied fault diagnosis method, which is based on the assumption that the variables with the largest contributions to the fault detection index are most likely the faulty variables. Because in the SPA-based fault detection method, PCA is applied to quantify the dissimilarity among different statistics patterns, here contribution plots for D_p and D_r are constructed to perform fault diagnosis.

3.5.1 Contributions to D_p and D_r

Let $\mathbf{X}$ denote the SP matrix with n samples (rows) and m statistics (columns). After auto-scaling to zero mean and unit variance, the matrix $\mathbf{X}$ is decomposed as in Equation (2.1). For fault detection on a new multivariate observation $\mathbf{x}$, two indices are used: D_p and D_r as defined below

$$D_p = (\mathbf{x}^T\mathbf{P}\mathbf{\Lambda}\mathbf{P}^T\mathbf{x}) \tag{3.10}$$

$$D_r = (\|\tilde{\mathbf{x}}\|^2 = (\mathbf{x}^T\tilde{\mathbf{P}}\tilde{\mathbf{P}}^T\mathbf{x}) \tag{3.11}$$



(a)

(b)

(c)

(d)

Figure 3.8: Density histogram intensity map for different pairs of statistics for normal operating data. (a) Mean of x_1 and mean of x_2; (b) Std. deviation of x_1 and Std. deviation of x_2; (c) Skewness of x_1 and Skewness of x_2; (c) Kurtosis of x_1 and Kurtosis of x_2.

where $\mathbf{\Lambda}$ is the diagonal matrix of the l largest eigenvalues of $\mathbf{XX}^T$, $\tilde{\mathbf{P}}$ is the residual loading matrix. The D_p statistic is a measure of the process variation in the principal component subspace, while the D_r statistic indicates how well each SP conforms with the model. The contribution of the i^{th} statistics to D_p and D_r are the following:

$$C_i^p = (\boldsymbol{\xi}_i^T(\mathbf{P\Lambda P}^T)^{1/2}\mathbf{x})^2 \tag{3.12}$$

$$C_i^r = (\boldsymbol{\xi}_i^T\tilde{\mathbf{P}}\tilde{\mathbf{P}}^{\mathbf{T}}\mathbf{x})^{\mathbf{2}} \tag{3.13}$$

where $\boldsymbol{\xi}_i^T$ is the i^{th} column of the m-dimensional identity matrix.

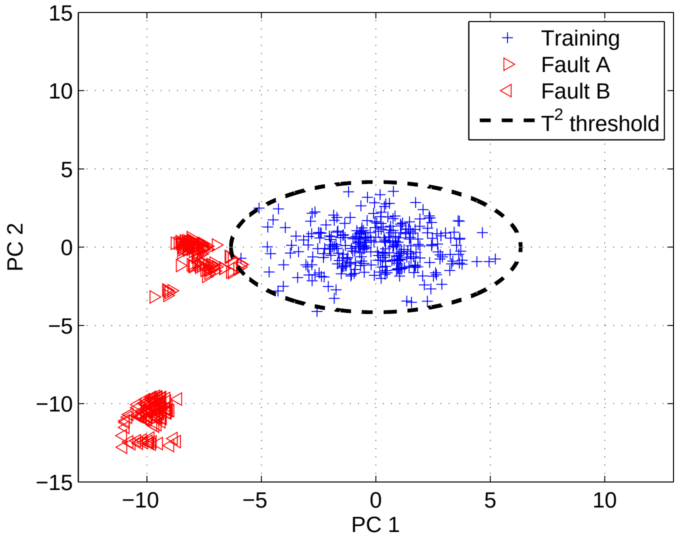


Figure 3.9: Scores of normal and faulty samples for SPA with the D_p threshold calculated at the 99% confidence level (dashed-line).

3.5.2 Hierarchical Contribution Approach

The contributions can be determined based on a single point, or based on a group of points. A visual inspection of the contributions highlights the variables that are more related to the particular fault. In SPA, a hierarchical contribution approach is utilized. First, the major group of statistics contributing to the fault are identified (e.g. mean, variance etc.). Second, the contribution is breakdown into single elements, i.e., governing statistic of different variables. In this way, not only SPA can isolate the variable or variables that are most related to the fault, but it yields an indication of the nature of the fault (e.g. a fault due to mean or standard deviation change). This unique feature makes contribution plots based on SPA to be a more effective and informative way to tackle problems related to the roots of faults online, e.g., instrument drift, noise, bias etc.

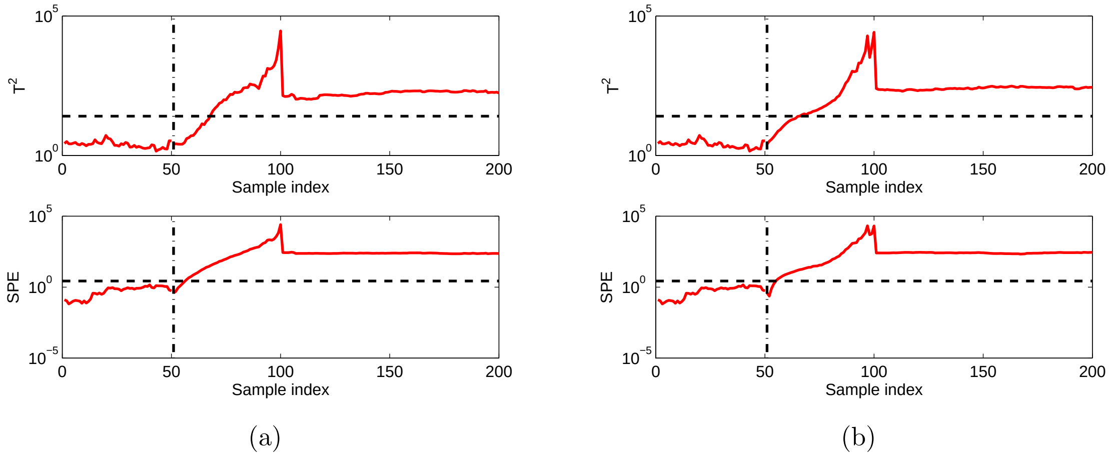


Figure 3.10: D_p and D_r indices of testing data. (a) Fault A; (b) Fault B. The dashed line indicates the threshold on the D_p and D_r indices calculated at the 99% confidence level. The dashed-dotted line indicates the fault onset.

3.6 Conclusion and Discussion

The idea of generating higher order statistics (HOS) for fault detection is not new, and other existent methods such as artificial neural networks, kernel principal component analysis, and independent component analysis all make use of HOS' implicitly. However, Statistics Pattern Analysis (SPA) is the first method available that not only utilizes regular statistics and HOS to better characterize the process, but it is the first method to monitor these statistics "explicitly". As discussed in this chapter, SPA not only can be applied to overcome the problems of manufacturing sites where mainly batch processes are utilized, but it also can be applied to continuous processes, which are common in petrochemical and chemical industries. As a result of the central limit theorem (CLT), the statistic patterns (SPs) are guaranteed to approach a Gaussian distribution asymptotically, or at least comply with a Gaussian distribution much better than the original variables due to the averaging effect. The better properties offered by the SPs enhances the capabilities of the already powerful PCA method at the dissimilarity quantification step. In addition, as the statistics of the process variables are more likely to capture the main characteristics of faults, the hierarchical contribution plot approach for fault detection is able not only to isolate the possible root of the fault, but to indicate the nature of the fault (e.g. mean or variance change), which is a huge advantage for online implementation. The material presented in this chapter highlights the premise of the SPA framework for process monitoring, especially for industrial processes where nonlinearities are prevalent. In the following chapter, more of the properties of SPA applied for fault detection and diagnosis will be studied, and its performance evaluated by the application to different case studies.

Chapter 4

Comprehensive Comparison of Linear and Nonlinear Monitoring Methods

4.1 Introduction

In this chapter, the performance of the SPA-based fault detection and diagnosis method is compared with some representative linear and nonlinear multivariate process monitoring methods using two case studies.

As discussed previously, the increasing demand for safer and more reliable systems in modern process operations leads to the rapid development of process monitoring techniques. By promptly detecting process upsets, equipment malfunctions, and other special events, online process monitoring not only plays an important role in ensuring process safety, but also improves process efficiency and product quality. Generally speaking, there are two main categories in process monitoring methods: univariate methods and multivariate methods. In univariate methods such as Shewhart, cumulative sum and exponentially weighted moving average charts [1, 3, 4], each variable (usually product quality) is monitored separately to identify the special causes of variation (i.e., faults) from natural or common causes of variation. In multivariate methods such as the principal component analysis (PCA) and partial least squares (PLS) methods [5, 6, 8, 9, 16, 26, 28], multiple variables are monitored together to detect faults that violate normal process variable correlations.

With a large amount of variables measured and stored automatically by distributed control systems (DCS), multivariate statistical monitoring methods have become increasingly common in process industry. Specifically, PCA-based process monitoring methods have gained wide application in chemical and petrochemical industries. PCA-based monitoring methods can easily handle high dimensional, noisy and highly correlated data generated from industrial processes, and provide superior performance compared to univariate methods.

In addition, the PCA-based process monitoring methods are attractive because they only require a good historical data set of normal operation, which are easily available for the computer-controlled industrial processes.

Although PCA-based monitoring methods have been very successful in many applications, there are cases where they do not perform well. Two of the possible reasons are the following. First, PCA only considers the mean and variance-covariance of the process data, and lacks the capability of providing higher-order representation for non-Gaussian data. Second, the control limits of Hotelling's T^2 and the squared prediction error (SPE) charts are developed based on the assumption that the latent variables follow a multivariate Gaussian distribution. When the latent variables are non-Gaussian distributed due to the process nonlinearity or other reasons, using Hotelling's T^2 and SPE may be misleading [33, 128].

To obtain better monitoring performance for the processes where PCA does not work well, several alternative approaches have been developed. Luo et al. [84] apply discrete wavelet analysis (DWA) to decompose the process measurements and then apply PCA to the wavelet coefficients at different detail levels to monitor the process; Kano et al. [129] define a dissimilarity index to monitor the distribution of the process time series data and to detect changes for the correlations among process variables that cannot be detected by Hotelling's T^2 and SPE charts; more recently, several multivariate monitoring methods based on independent component analysis (ICA) have been proposed, where ICA decomposes the process data in to linear combinations of statistically independent components [11, 33, 103]. Among these approaches, Kano's approach is a second-order method similar to PCA, while DW- and ICA-based fault detection methods involve higher-order statistics implicitly.

In Chapter 2, a new multivariate statistical process monitoring framework, named Statistics Pattern Analysis (SPA) [32, 59] is introduced. The major difference between the PCA-based and SPA-based fault detection methods is that PCA monitors process variables, while SPA monitors the statistics of the process variables. In other words, SPA examines the variance-covariance of the process variables statistics (e.g. mean, variance, autocorrelation, cross-correlation etc.) to perform fault detection. In SPA, different statistics that capture the different characteristics of the process can be selected to build the model for normal process operation, and various higher-order statistics can be utilized explicitly. Fault detection methods based on the SPA framework have been shown to provide superior monitoring performance for several batch and continuous processes compared to the PCA and dynamic PCA based methods [32, 59].

In this chapter, the potential of the SPA framework for fault diagnosis of continuous processes is explored. Specifically, the variable contributions from the fault detection indices generated by SPA are derived; then contribution plots are constructed to perform fault diagnosis. Because the SPA based fault detection method for continuous processes is a window-based approach, the SPA based contribution plot is an averaged contribution plot for all samples in the window. In the first case of study, the SPA framework performance is compared to linear monitoring methods such as PCA and DPCA using a nonlinear quadruple-tank system. In addition, as the SPA-based monitoring for continuous processes is a window-based approach, and there might be cases where samples from process variables are limited, in this chapter, the effect of window overlapping for calculating the SPs is presented.

In the second case study, the challenging Tennessee Eastman Process (TEP) [130] is used to demonstrate the performance of the SPA-based fault detection and diagnosis method, which is compared with linear process monitoring methods such as PCA, dynamic PCA (DPCA), together with nonlinear process monitoring methods such as kernel PCA (KPCA), and independent component analysis (ICA). In this case study, the main characteristics of SPA are discussed. In addition, a comprehensive analysis of the important aspects for fault detection and diagnosis are elucidated and discussed.

4.2 Quadruple Tank

In this section, a simulated quadruple-tank process is used to demonstrate the performance of the proposed SPA method and investigate its properties. In addition, the monitoring performance of SPA is compared with the traditional PCA and DPCA methods. The quadruple-tank process was originally developed by Johansson [131] as a novel multivariate laboratory process. This process consists of four interconnected water tanks, two pumps, and associated valves. A schematic diagram of the process is shown in Figure 4.1. The inputs are the voltages supplied to the pumps, v_1 and v_2, and the outputs are the water levels $h_1 - h_4$. The flow to each tank is adjusted using the associated valves γ_1 and γ_2. A nonlinear model is derived based on mass balances and Bernoulli's law.

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 \kappa_1}{A_1} v_1 \tag{4.1}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 \kappa_2}{A_2} v_2 \tag{4.2}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)\,\kappa_2}{A_3} v_2 \tag{4.3}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)\,\kappa_1}{A_4} v_1 \tag{4.4}$$

$$\tag{4.5}$$

For tank i, A_i is the cross-section of the tank, a_i is the cross section of the outlet hole, and h_i is the water level. The voltage applied to pump i is v_i and the corresponding flow is $\kappa_i v_i$. The parameters γ_1 and $\gamma_2 \in (0, 1)$ are determined from how the valves are set before an experiment. The water flow rate to tank 1 (i.e., f_1) is $\gamma_1 \kappa_1 v_1$ and to tank 4 (i.e., f_4) is $(1 - \gamma_1)\kappa_1 v_1$. The flow rates for tanks $2 - 3$ are obtained similarly. The acceleration of gravity is denoted by g. The parameter values of this process are given in Table 4.1.

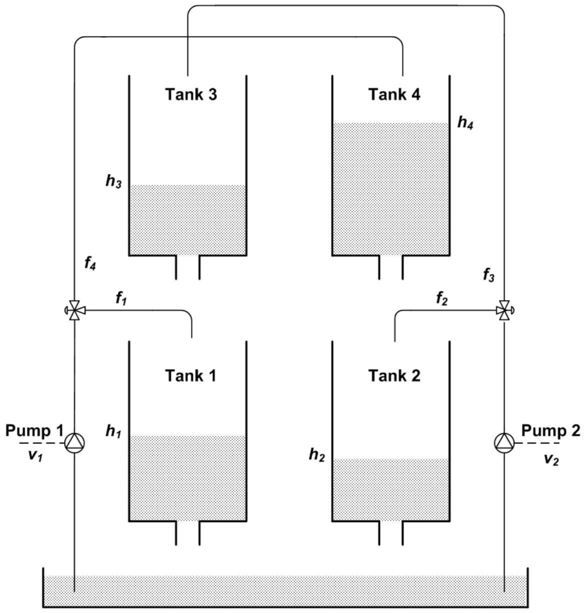


Figure 4.1: Schematic diagram of the quadruple-tank process

Table 4.1: Simulation parameters for the quadruple tank case study*

Parameter	Units	Value
$A_1; A_3$	cm^2	28
$A_2; A_4$	cm^2	32
$a_1; a_3$	cm^2	0.0071
$a_2; a_4$	cm^2	0.0057
$\kappa_1; \kappa_2$	$cm^3V^{-1}s^{-1}$	3.33
g	cm/s^2	9.81

*After Johansson [131]

4.2.1 Simulation Settings

The data are generated by Eqs. (4.1) - (4.4) where γ_i and v_i are corrupted by independently Gaussian white noise with zero mean and standard deviation of 0.01 and 0.05, respectively, which are about $1-2\%$ of their upper limit or steady-state value. Measured h_i is corrupted by Gaussian distributed white noise with zero mean and standard deviation of 0.1 and measured γ_i and v_i contain the same level of noise as the input γ_i and v_i, respectively. Three normal data sets are generated with different random seeds with a sampling frequency of 10 seconds. They are denoted as training set, validation set and testing set, respectively. The training set contains 240 hours of normal data and is used to build the process model; the validation set contains 15 hours of normal data and is used to determine the thresholds of the monitoring indexes, i.e., D_p and D_r for SPA, T^2 and SPE for PCA and DPCA. Finally, the testing set contains 15 hours of normal data, and is used to test the false alarm rate of different monitoring methods. Two faults, a sensor fault and a tank leakage fault are examined in this work. For the sensor fault, a bias of $\Delta h_4 = 0.3$ is added to sensor measurement h_4. Figure 4.2 shows the time series data of the process variables, water levels h_i and flow rates f_i $(i = 1, ..., 4)$ corresponding to the sensor fault in h_4. It is clear from Figure 4.2 that this fault is very difficult to detect by the naked eye due to its small magnitude. For the leakage fault, a leakage in tank 1 is introduced by assuming that there is a small hole at

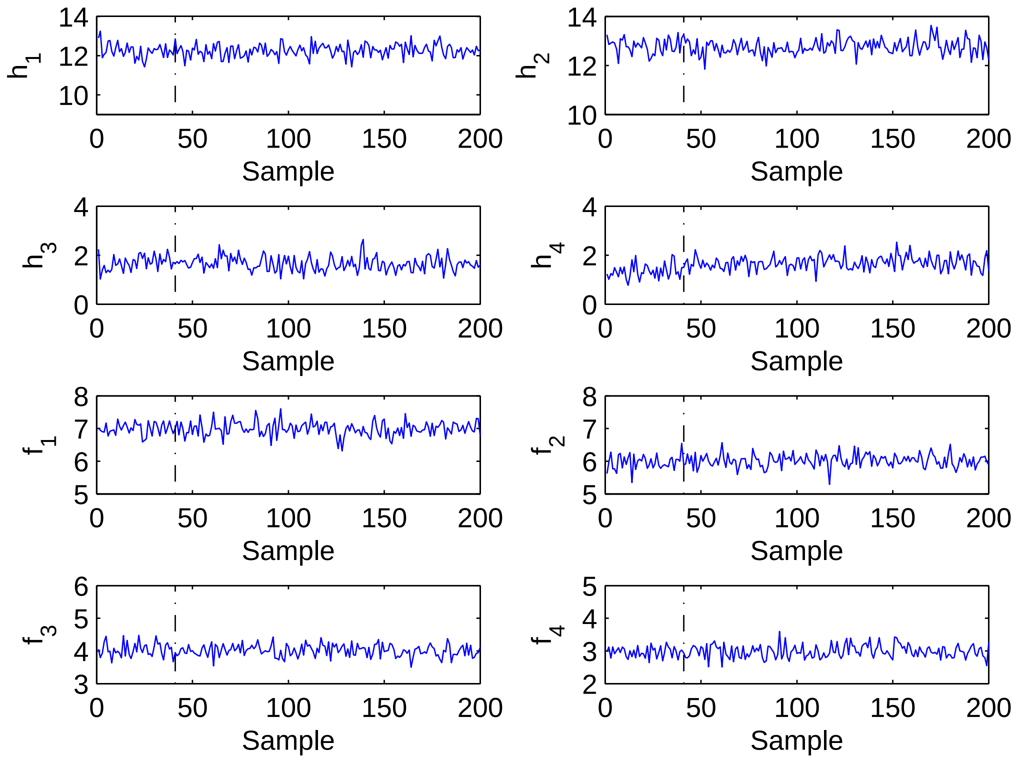


Figure 4.2: Time series of process variables for the sensor fault in h_4

the bottom of the tank with cross-section $a_{leak} = 0.005\ cm^2$. The mass balance for tank 1 changes from Eqn. (4.1) to the following.

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1\kappa_1}{A_1}v_1 - \frac{a_{leak}}{A_1}\sqrt{2gh_1} \tag{4.6}$$

where the last term corresponds to the leakage of tank 1. Mass balance equations for the other tanks do not change. Figure 4.3 shows the time series data of the process variables, water levels h_i and flow rates f_i $(i = 1, ..., 4)$ corresponding to the the leakage in tank 1. For both faults we start the simulation with normal operation, and then introduce the fault after 0.5 hours. Totally, 15 hours of data are generated.

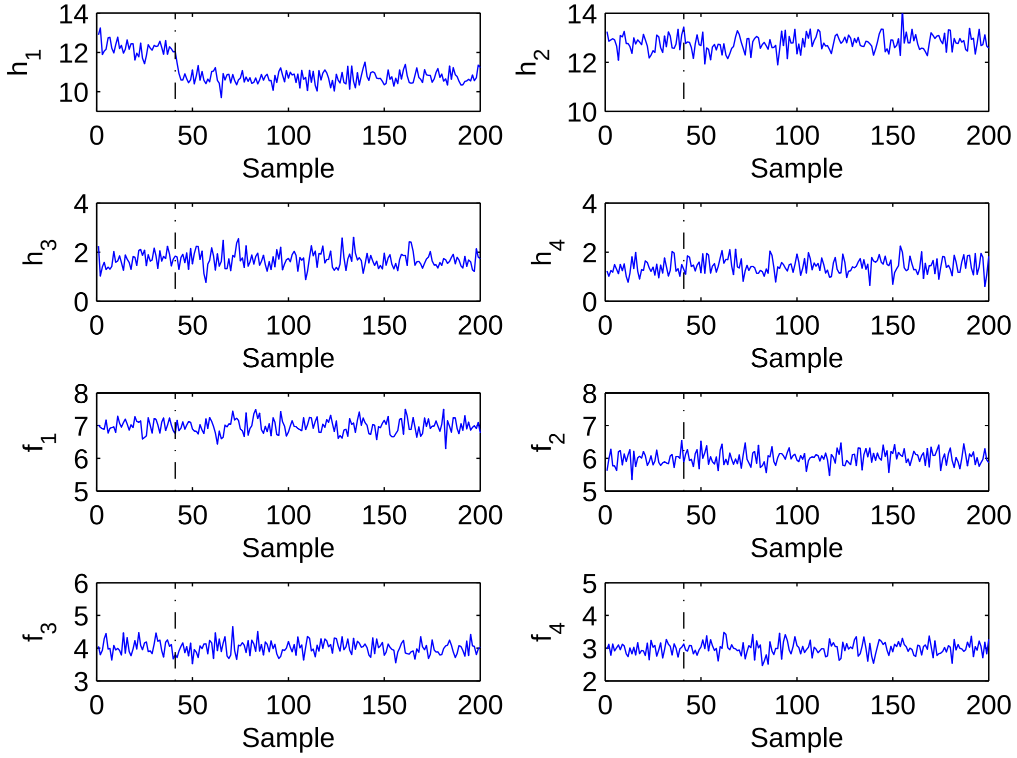


Figure 4.3: Time series of process variables for the leakage fault in h_1

4.3 The Effect of Window Overlapping

One of the major requirements of the application of SPA is the availability of enough data in order to have enough training SPs compared to the number of statistics included to characterized the process. For continuous processes, if the data are limited, the number of windows available to calculate the different statistics is reduced, and the determination of the different coefficients of the SPA model might be affected. To circumvent this problem, a window overlapping approach is utilized, i.e., the window to calculate the statistics is shifted by a certain percentage of the total size of the window. Depending of the amount of overlapping, the number of SPs increases, thus the ratio of number of statistics to number of samples available decreases, and the determination of the different coefficients relating the different statistics is more reliable. Although valid, following this approach rises one question, what is the effect of using a large window overlapping, considering that one of the major assumptions of PCA is the availability of independent samples, and a large window overlapping could violate this assumption?

In this work, to test the effect of window overlapping, two scenarios are compared: in case I, 3 hours of training data are used to build the model, and in case II, all 240 hours of training data are used to build the model. For all three methods, the number of PC's is determined through cross-validation. The thresholds are determined through empirical method using validation data. Specifically, a 99% confidence upper control limit is determined as the index value below which 99% of the calibration samples are located. Standard contribution plots [34] are used for fault diagnosis. To reduce the effect of process measurement noise on the contribution plot, the average contribution plot from 100 samples is examined to evaluate the fault diagnosis performance of different methods.

4.3.1 Fault 1: The Sensor Bias

PCA, DPCA and SPA are applied to perform fault detection. The model settings for all three methods are presented in Table 4.2. For SPA method, the window width to calculate

Table 4.2: Model settings for the quadruple tank case study

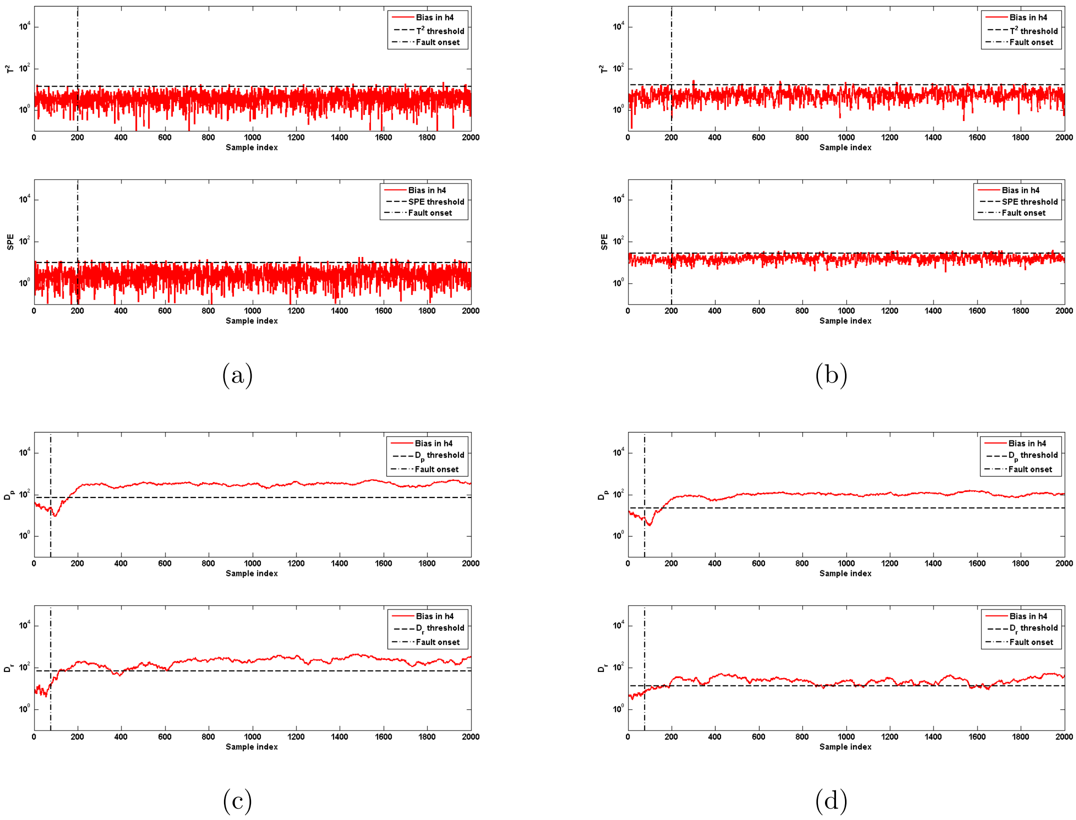
	Method	Variables	PC's	Lags	Window Width	Window Overlap
Case I	PCA	8	4	–	–	–
	DPCA	24	6	2	–	–
	SPA	18	11	2	125	60%
Case II	PCA	8	4	–	–	–
	DPCA	24	6	2	–	–
	SPA	18	9	2	125	0%

the SPs is chosen to be 125 samples. Due to the limited training samples in case I, 60% of overlap is allowed between two adjacent windows. In case II, with enough training samples, the window overlap is removed to eliminate any correlation among different training SPs, which is expected to improve process monitoring performance. Figure 4.4 presents the fault detection results from different methods. Because detection results from PCA and DPCA are similar for case I and case II, only results from case II are presented. For SPA, detection results for both cases are presented because there are some noticeable differences between them. The summary of fault detection rates, detection delays and false alarms rates for both cases is presented in Table 4.3.

Table 4.3: Fault detection for bias in h_4

		PCA		PCA		SPA	
		T^2	SPE	T^2	SPE	D_p	D_r
Case I	Detection rate (%)	1.6	2.6	1.1	3.5	98.5	99.8
	False alarm rate (%)	1.4	1.0	1.1	0.9	10.0	6.7
	Detection delay (min.)	16.5	0.17	16.3	0.17	13.8	6.16
Case II	Detection rate (%)	1.5	2.6	1.3	4.7	98.1	98.6
	False alarm rate (%)	0.7	0.7	1.3	1.1	0.5	2.4
	Detection delay (min.)	16.7	0.5	16.3	0.17	15.8	12.3

Table 4.3 indicates that the sensor bias is very difficult to detect by PCA and DPCA methods, with detection rates lower than 5% for both cases. This is mainly due to the relatively small bias in the level sensor h_4. On the other hand, this fault can be easily



(a)

(b)

(c)

(d)

Figure 4.4: Fault detection performance. (a) PCA case II, (b) DPCA case II, (c) SPA case I, (d) SPA case II.

detected by SPA for both cases. However, for case I, SPA has a higher false alarm rate (10% for D_p and 6.7% for D_r). This is because window overlapping in training data is implemented due to the limited training samples. The overlapping causes the training SPs no longer be independent from each other; therefore the model may not adequately capture the process characteristics, which results in higher false alarms rate. In case II, when more SPs are included to craft the model and no window overlapping is present, not only the detection rate is high, but the number of false alarms reduces significantly. The effect of highly correlated SPs and their influence in the increment of false alarms rate is illustrated in Figure 4.5. In Figure 4.5, the scores of the first two principal components of highly correlated and uncorrelated SPs for normal training and testing are shown. As observed in Figure 4.5(a), the scores of the normal training data cannot fully describe the normal operating region. This causes the scores of normal testing data to form a cluster that occupies a different region in the plane defined by the first two principal components. This occurs as a consequence of the high overlapping of windows for calculating the SPs (e.g. 95%), which causes an increase in the false alarm rate. In contrast, Figure 4.5(b) show that the scores of normal operating data, where no overlapping is used to calculate the statistics of the process variables form an ellipse that describes better the normal operating region. This causes the scores for normal testing data to be well within that region. As a result, the percentage of false alarm rates is reduced to the established significance level, as verified in in the results presented in Table 4.3. Figure 4.6 compares the monitoring results of SPA on

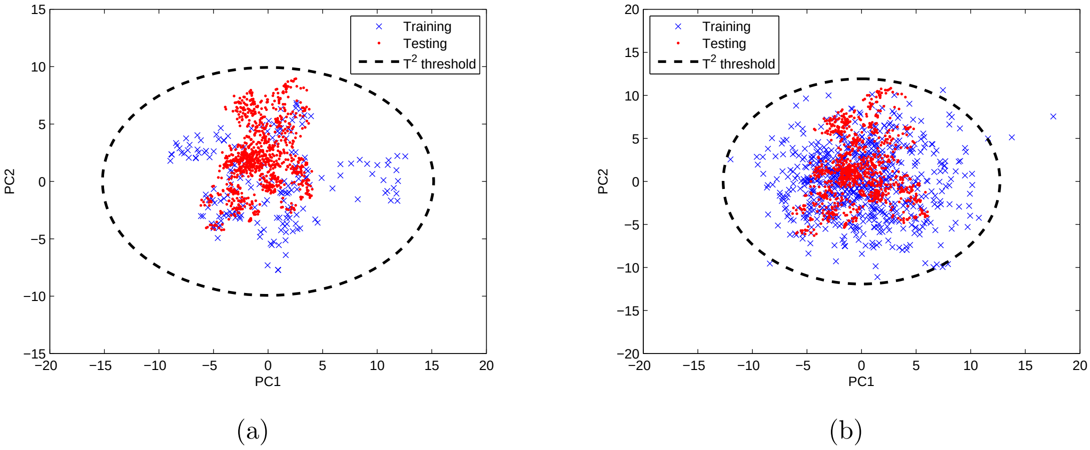


Figure 4.5: Score plot of the first two principal components for the SPA method. (a) Highly correlated SPs (95% window overlapping); (b) Uncorrelated SPs (i.e., without window overlapping). The dashed-line indicates the control limit of the Hotelling's T^2 calculated at the 99% confidencel level.

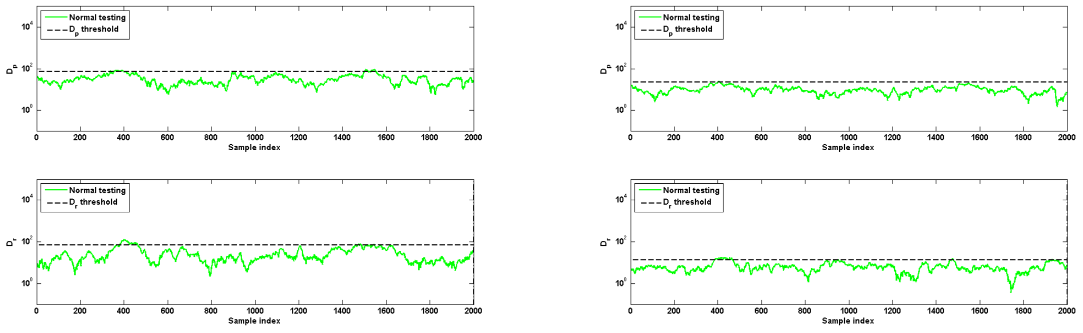
the testing set. Clearly, in case II with independent training samples, the SPA model is able to capture the process characteristics adequately, and the false alarms rate is compatible with the confidence limits (99%) that is used to determine the threshold. Note that, in order to show the details, only segments of the process data are plotted for each method.

Contribution plots are generated to diagnose the fault for the three different methods for case II. The average contribution plots for the three methods are presented in Figure 4.7.

Figures 4.7(a) and 4.7(b) show the contribution based on PCA-SPE and PCA-T^2, respectively. PCA is able to identify h_4 as the root cause of the fault. However, the other level measurements also show significant contribution to the fault, which makes pinpointing the root cause of the fault difficult. On the other hand, DPCA in Figure 4.7(c) shows that is able to clearly identify the fault by SPE. However it fails to correctly identify the fault by T^2, as shown in Figure 4.7(d). For SPA, Figures 4.7(e) and 4.7(f) show that both indexes (i.e. D_r and D_p) are able to clearly identify the root cause of the fault (e.g. a mean shift of level h_4).

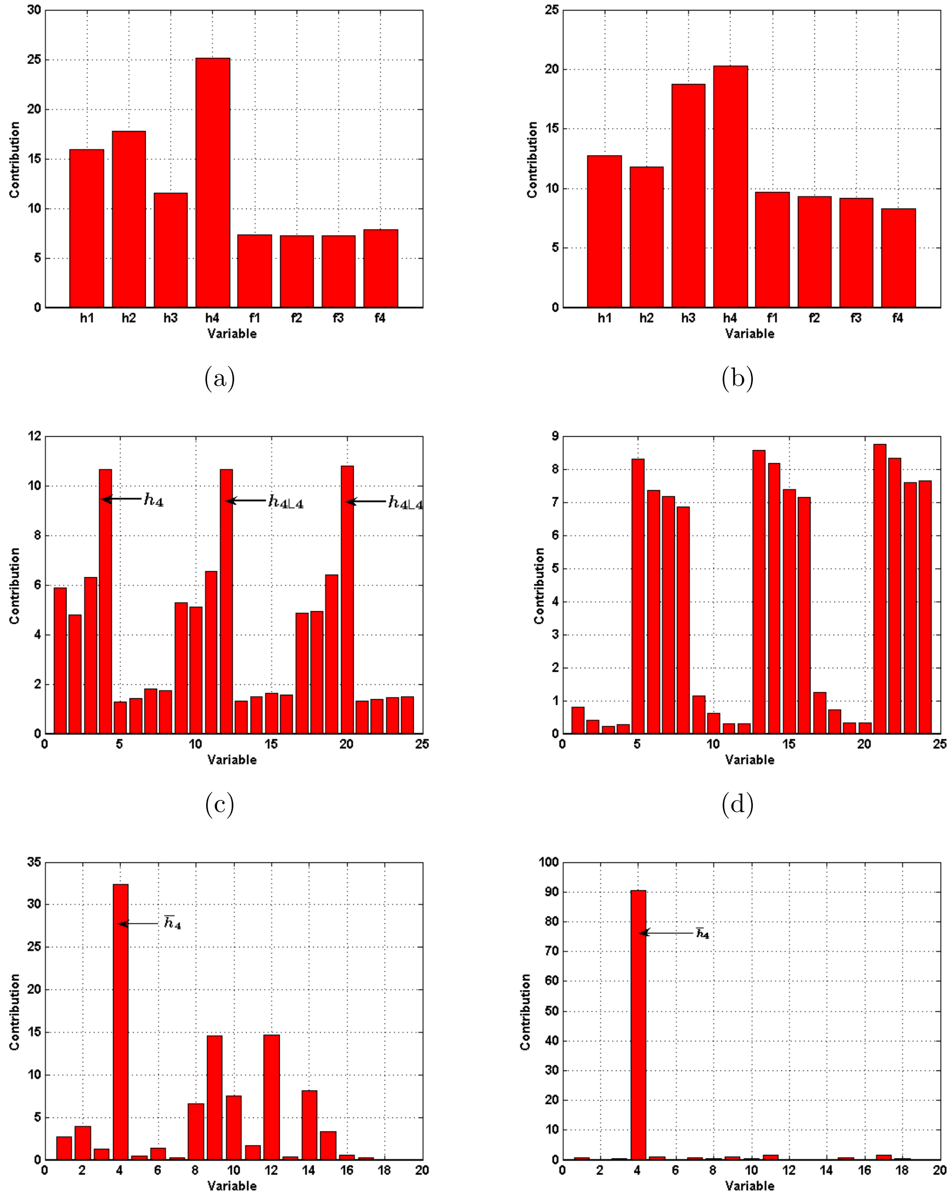
4.3.2 Fault 2: Leakage in Tank 1

The model settings for fault detection are the same as for the previous fault. Table 4.4 summarizes the fault detection rates and detection delays for each case. Figure 4.8 shows the detection details for different methods. For this fault, both PCA and DPCA are able to detect the fault, with PCA showing detection rates ranging from 60% – 73% and DPCA being able to detect the fault almost entirely by SPE index. The SPA method was able to detect the fault successfully with detection rates close to 100% for both cases. The average SPE contribution plots for the three methods are presented in Figure 4.9. Figures 4.9(a) and 4.9(b) display the contribution based on PCA-SPE and PCA-T^2, respectively. PCA is able to identify h_1 as the root cause of the fault in both indices. However, h_3 is also identified



(a) (b)

Figure 4.6: D_p and D_r indexes for normal testing data. (a) SPA case II, (b) SPA case II.



(a)

(b)

(c)

(d)

(e)

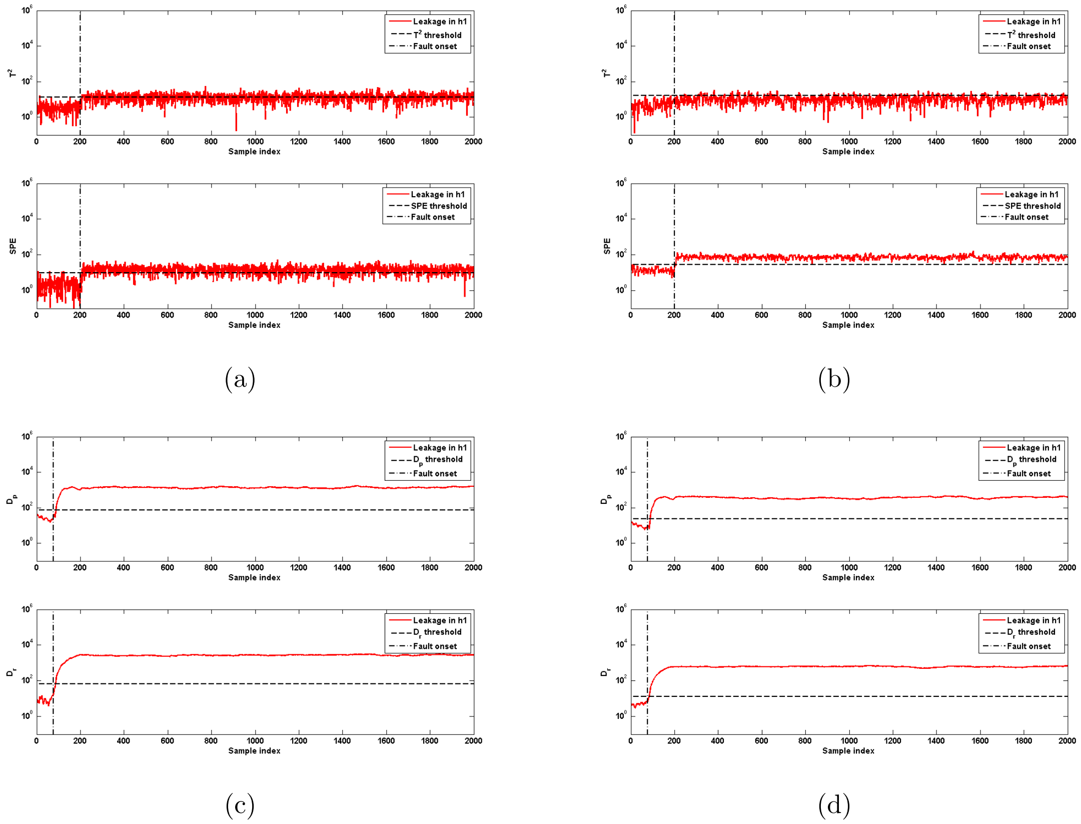
(f)

Figure 4.7: Contribution plots based on (a) PCA SPE; (b) PCA T^2; (c) DPCA SPE; (d) DPCA T^2; (e) SPA D_r; (f) SPA D_p for case II with bias in h_4. Where h_i and $h_i - Li$ denote the mean of h_i and a lagged measurement of h_i, respectively.

as one major contributor to the fault in the contribution to SPE which could lead to more difficult diagnosis. On the other hand, DPCA and SPA in Figures 4.9(c); 4.9(d), 4.9(e), 4.9(f), indicate that both methods are able to clearly diagnose the fault by identifying a mean shift of level h_1 for SPA (e.g. $\overline{h}_1$) and a fault in level h_1 for DPCA. However, the diagnosis of

Table 4.4: Fault detection for leakage in tank 1

		PCA		PCA		SPA	
		T^2	SPE	T^2	SPE	D_p	D_r
Case I	Detection rate (%)	60.6	72.9	48.4	99.7	99.8	99.8
	Detection delay (min.)	0.17	1.3	0.17	0.33	2.3	2.0
Case II	Detection rate (%)	49.2	75.3	16.0	99.0	99.7	99.8
	Detection delay (min.)	1.3	1.3	4.5	0.17	2.5	1.7



(a)

(b)

(c)

(d)

Figure 4.8: Fault detection performance. (a) PCA case II, (b) DPCA case II, (c) SPA case I, (d) SPA case II.

the nature of the fault is straightforward for SPA since the contribution plot of the fault indicates that an operating point shift in h_1 has occurred.

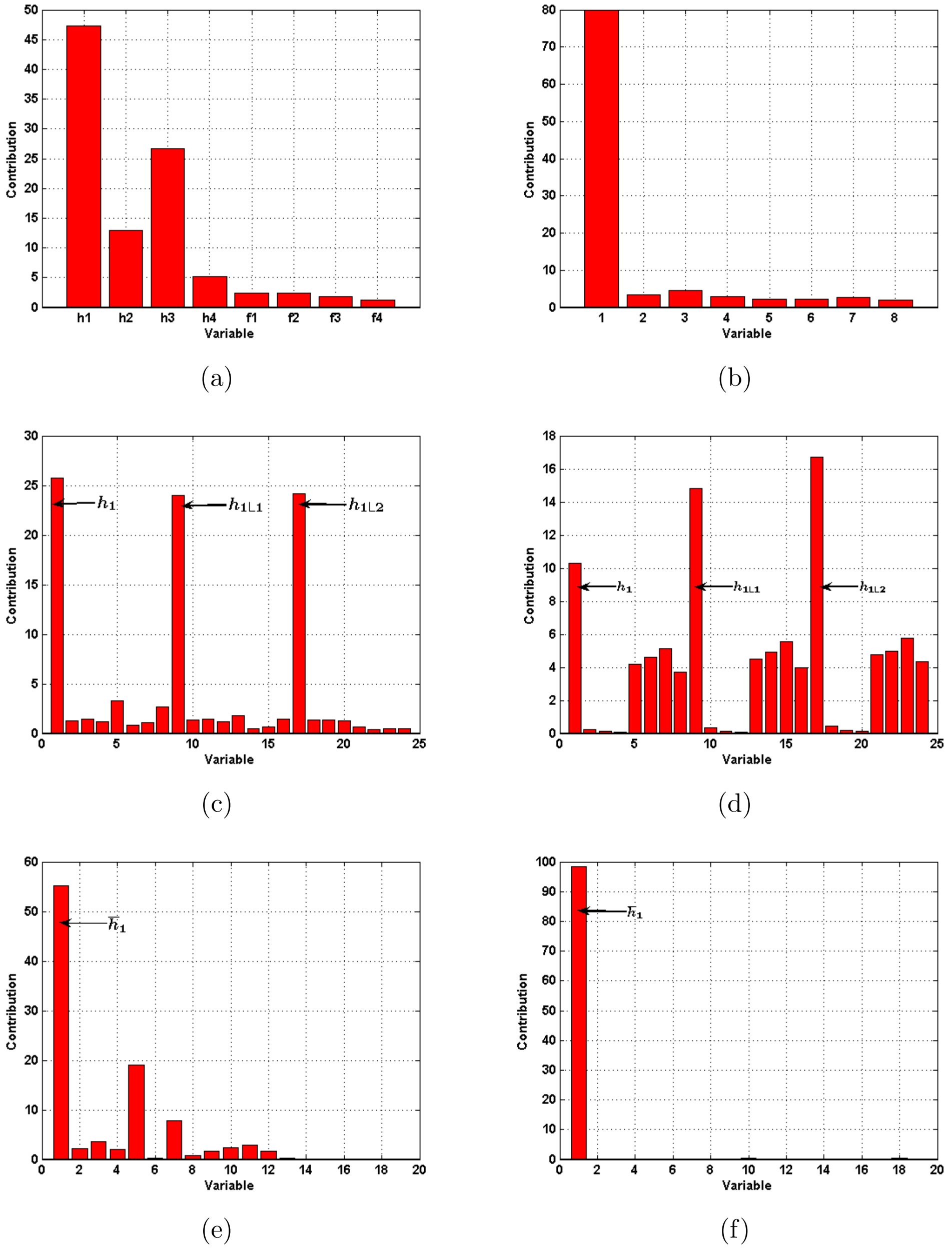


Figure 4.9: Contribution plots based on (a) PCA SPE; (b) PCA T^2; (c) DPCA SPE; (d) DPCA T^2; (e) SPA D_r; (f) SPA D_p for case II with bias in h_4. Where $\bar{h}_i$ and $h_i - Li$ denote the mean of h_i and a lagged measurement of h_i, respectively.

4.4 Tennessee Eastman Process

The TEP process simulator has been widely used by the process monitoring community as a realistic example to compare various monitoring and control approaches [29, 124, 129]. The process consists of five major unit operations: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor, and a product stripper. Four reactants A,C,D and E plus the inert B are fed to the reactor to generate products G and H, as well as byproduct F through two exothermic reactions. Figure 4.10 shows a process flow diagram of the TEP. The simulator was originally developed by Downs and Vogel [130] and different control strategies were implemented in different modified versions [132–135]. In this work, Ricker's simulator is used to generate normal and faulty data. The simulator is able to simulate normal process operation together with 20 faulty conditions. Table 4.5 shows a list of the 20 process disturbances and their type.

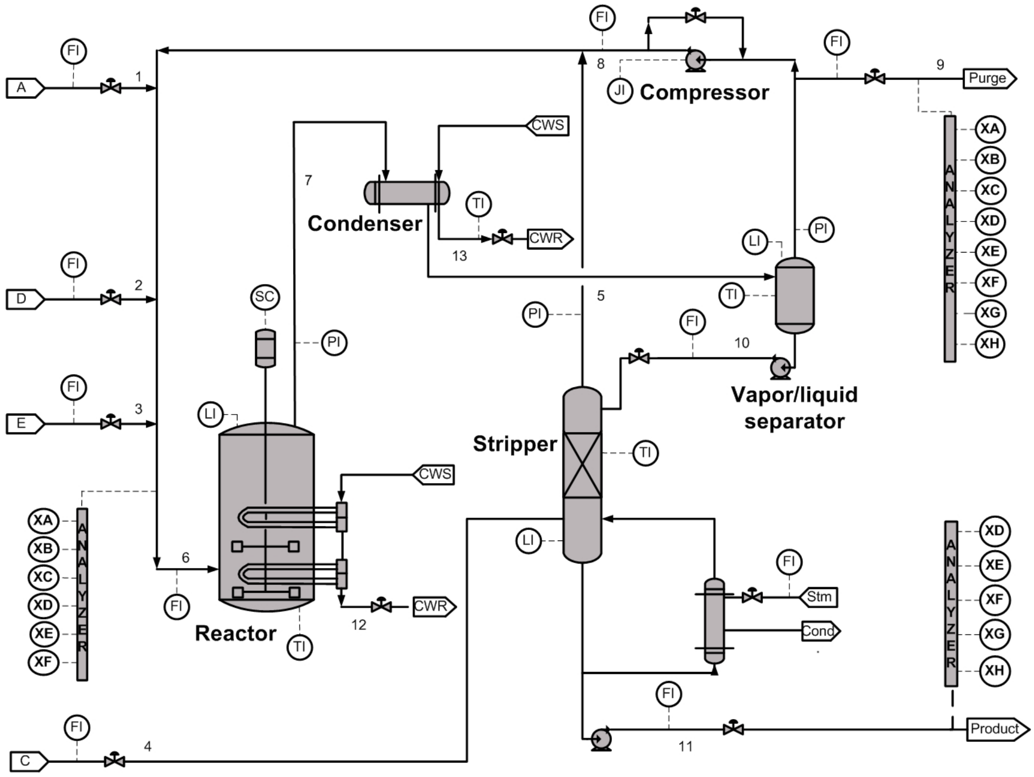


Figure 4.10: Flow diagram of the Tennessee Eastman processes

Table 4.5: Disturbances in the TEP

no.	Description	Type
1	A/C feed ratio, B composition constant (stream 4)	Step
2	B composition, A/C ratio constant (stream 4)	Step
3	D feed temperature (stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (stream 1)	Step
7	C header pressure loss-reduced availability (stream 4)	Step
8	A,B,C feed composition (stream 4)	Random variation
9	D feed temperature (stream 2)	Random variation
10	C feed temperature (stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16-20	Unknown	Unknown

4.4.1 Simulation Settings

Following [133], the data are collected at a sampling interval of 3 min. The process data include 11 manipulated variables, 22 continuous process measurements, and 19 composition measurements which are sampled less frequently. Similar to [33], 22 continuous process measurements and 9 manipulated variables listed in Table 4.6 are used to monitor the process. The other two manipulated variables were fixed to constants in Ricker's decentralized control scheme, and therefore are not included in the monitoring. Details about the process variables can be found in [130, 135]. A normal training set of 800 hr of normal data is used for training the model, 50 hr of normal data for false alarm rate calculation, and each fault consists of 50 hr with the first 10 hr being normal.

4.4.2 Monitoring Settings

The monitoring settings of the different methods are listed in Table 4.7. The settings for PCA, DPCA and SPA are similar to the settings in [32]. For SPA, the statistics included in

Table 4.6: Monitored variables in TEP

no.	Variable description	no.	Variable description
	Process Measurements		
1	A feed (stream 1)	12	Product separator level
2	D feed (stream 2)	13	Product separator pressure
3	E feed (stream 3)	14	Product separator underflow (stream 10)
4	A and C feed (stream 4)	15	Stripper level
5	Recycle flow(stream 8)	16	Stripper pressure
6	Recycle feed rate (stream 6)	17	Stripper underflow (stream 11)
7	Reactor pressure	18	Stripper temperature
8	Reactor level	19	Stripper steam flow
9	Reactor temperature	20	Compressor work
10	Purge rate (stream 9)	21	Reactor cooling water outlet temperature
11	Product separator temperature	22	Separator cooling water temperature
	Manipulated Variables		
23	D feed flow (stream 2)	28	Separator pot liquid flow (stream 10)
24	E feed flow (stream 3)	29	Stripper liquid product flow (stream 11)
25	A feed flow (stream 1)	30	Reactor cooling water flow
26	A and C feed flow (stream 4)	31	Condenser cooling water flow
27	Purge valve (stream 9)		

the statistic pattern (SP) include the mean, the standard deviation, correlation coefficients, cross-correlation and auto-correlation coefficients. A correlation coefficient (r_{ij}) is retained if $|r_{i,j}| > 0.3$ for more than 50% of the training SPs. For auto-correlation (r_i^d) and cross-correlation ($r_{i,j}^d$) coefficients, it is required that their values are $\left|r_i^d\right| > 0.3$ $\left|r_{i,j}^d\right| > 0.3$ for more than 50% of the training SPs, where d denotes the time lag ($d = 2$) between variables. For KPCA and ICA, the settings were found after a thorough search. For KPCA, a gaussian kernel function, i.e., $k(\mathbf{x}, \mathbf{y}) = \mathbf{exp}\left(-\frac{\|\mathbf{x}-\mathbf{y}\|}{\mathbf{c}}\right)$ is used to compute the kernel matrix ($\mathbf{K}$), where $c = \beta m \sigma^2$. The parameters $\beta = 8$, $\sigma = 0.9$ were found by optimizing the detection performance, and $m = 31$ is the number of process variables. In addition, the number of components retained after applying PCA in the feature space, and number of components for testing are determined by optimizing the detection performance. For ICA, the FastICA [102] algorithm described in Chapter 2 is used to find the independent components. A contrast function to approximate negentropy equal to $G_1(u) = \frac{1}{a_1} log\, cosh(a_1 u)$, where $a_1 = 1$ is used.

The number of independent components for testing is determined used the Euclidean norm approach [106]. For all methods, an empirical approach is used to determine the upper control limits so that different methods can be compared based on the same confidence level [32].

Table 4.7: Settings of different fault detection methods

Methods	Variables	PCs (ICs)	lags	Window width	Window shifting step
PCA	31	9	-	-	-
DPCA	90	20	2	-	-
ICA	31	6	-	-	-
KPCA	60	26	-	-	-
SPA	257	6	2	100	50

4.5 Fault detection

In this section, the most important aspects in fault detection are discussed based on the results obtained in monitoring the 20 faults in the TEP benchmark problem.

4.5.1 Detection Rate

The detection rates of the five methods for all faults are listed in Table 4.8. It is worth noting that the results in Table 4.8 for PCA, DPCA and SPA are different from the results reported in [32]. This is due to the fact that different data sets were used. In [32], the data set was generated by the control strategy developed in [133], where the plant-wide control structure recommended in [134] was implemented to generate the closed-loop normal and faulty process data. While in the current work, the TEP simulator with decentralized control system developed by Ricker [135] was implemented to generate the closed-loop normal and faulty process data. By visually comparing the two control strategies under normal operation condition, it was noticed that the process variation under the decentralized control is much smaller than that under the plant-wide control. Therefore, the normal process model with the decentralized control defines a tighter region of normal operation than that with the plant-wide control, which makes the model more sensitive and therefore more effective to fault detection. This is true no matter what fault detection is used. As a result, although similar settings were used, the fault detection rates of all the three methods are higher for almost all faults in this work than those in reported in [32], with PCA and DPCA improved the most. From Table 4.8, it is evident that all five methods are effective in detecting most of the faults. It is worth noting that faults 3, 9 and 15 have been suggested to be difficult to detect when the plant-wide control strategy is implemented. In this work, these results were confirmed since these faults are also difficult to detect by any of the five methods when the decentralized control strategy is implemented. Therefore, these three faults are not listed in Table 4.8. In addition, fault 16 cannot be detected by any of the five methods when the decentralized control strategy is implemented. After visually inspecting the four difficult cases, the reason for these faults not being detected is that the disturbances were completely rejected without introducing noticeable changes to any process variables. For the rest of the 16 faults, PCA, DPCA, KPCA and ICA have difficulty in detecting faults 5 and 12 with detection rates lower than 55% in most of the cases, which are depicted in Figures 4.11 - 4.13. On the other hand, SPA based method is able to detect all 16 faults with all detection rates higher than 90%. Additionally, for the faults that cannot be completely detected by PCA, DPCA, KPCA and ICA methods, such as fault 18, SPA is more effective, this is illustrated in Figure 4.13.

4.5.2 False Alarm Rate

In any process monitoring approach, it is attempted to reduce the false alarm rates to the desired significance level (e.g. consistent samples labeled as faulty samples). In this work, to ensure a fair comparison, the thresholds for monitoring are determined empirically based on the values of the monitoring indexes in normal validation data. The false alarm rates of all the methods for training and testing normal data are listed in Table 4.9. A

Table 4.8: Fault detection rates (percentage) of PCA, DPCA, KPCA, DPCA, and SPA for the TEP

	PCA		DPCA		KPCA		ICA			SPA	
fault	T^2	SPE	T^2	SPE	T^2	SPE	I_d^2	I_e^2	SPE	D_p	D_r
1	99.9	100	100	100	100	99.7	100	100	99.9	99.3	99.5
2	99.5	99.2	99.6	99.4	99.6	98.6	99.4	99.4	99.1	99.0	98.8
4	100	100	100	100	100	100	100	100	100	99.8	99.9
5	3.6	0.6	3.6	1.5	3.0	0.4	0.6	2.1	3.7	0.0	93.4
6	100	100	100	100	99.3	100	100	100	100	99.8	99.4
7	100	100	100	100	100	100	100	100	100	99.9	99.9
8	98.9	98.1	99.0	98.0	98.9	95.6	98.8	98.8	98.4	96.9	98.4
10	83.7	93.1	93.0	96.2	97.3	69.6	95.6	96.4	92.7	98.1	98.5
11	94.9	98.2	98.9	99.7	99.1	81.4	94.1	98.9	98.3	99.0	99.5
12	44.2	14.6	59.1	39.1	54.9	4.5	22.9	44.9	42.1	93.8	94.6
13	98.7	98.7	99.0	99.1	99.1	98.7	98.6	99.1	98.9	97.9	98.3
14	99.1	100	100	100	100	92.7	100	100	100	99.5	99.8
16	4.4	0.5	4.1	0.4	2.6	0.4	0.6	1.8	3.7	0.0	0.0
17	97.2	99.2	99.2	99.6	99.5	88.0	99.2	99.4	99.12	98.4	99.0
18	71.3	76.5	73.0	85.7	86.6	64.5	78.1	81.6	75.1	93.6	96.0
19	98.0	98.5	99.5	99.0	99.4	94.9	98.1	99.3	98.9	96.5	98.9
20	98.9	98.6	99.0	99.0	98.9	99.6	98.1	99.3	98.9	97.9	98.5

detailed analysis of the results shown in Table 4.9 indicates that the thresholds determined empirically with 99% confidence level are reasonable and consistent, because the false alarm rates are not far from 1% for all the methods.

Table 4.9: False alarms rate (percentage) of PCA, DPCA, KPCA, ICA, and SPA for the TEP

	PCA		DPCA		KPCA		ICA			SPA	
Data	T^2	SPE	T^2	SPE	T^2	SPE	I_d^2	I_e^2	SPE	D_p	D_r
Training	2.1	0.4	2.4	0.7	0.6	1.7	0.6	1.2	1.6	1.0	1.0
Testing	1.5	0.3	1.4	0.7	1.1	1.0	0.6	0.3	0.8	0.0	2.6

4.5.3 Detection Delay

It is worth noting that although detection delay is usually associated with window-based approaches, there is only minor detection delay associated with SPA method. This is because the detection limit is much tighter in SPA [59], since the variance of the variable statistics is only $\frac{1}{n}$ of the process variable (n is the window width). This is illustrated in Table 4.10, where the detection delay associated with SPA is not significant compared to the detection delay observed for the rest of the methods. Table 4.10 shows that there is a slight increment of detection delay for faults 5 and 12, which is related to the nature of the fault and the type of control strategy implemented. From Table 4.8, it is observed that SPA is the only method to detect fault 5. Fault 5 corresponds to a step change of an unmeasured disturbance (i.e., the condenser cooling water inlet temperature). A detailed analysis of the process data indicates that this fault is rejected by the control strategy fairly well. However, as it will be discussed in the fault diagnosis section, it eventually affects the mean of variable 31 (i.e., the condenser cooling water flow), which is only detected by SPA. As shown in Figure 4.11, after this fault propagates enough through the system, the detection of SPA gives a clear indication of the fault. For fault 12, a random variation of the condenser cooling water inlet temperature affects the system. Again, the control strategy rejects the disturbance fairly well, and although there is a slight delay of SPA detecting this fault, SPA is the only method that after this fault propagates enough, it gives a clear indication that the fault occurred. This is illustrated in Figure 4.12, where the detection performance of D_p and D_r in SPA does not observe a chattering effect (i.e., drastic up's and downs) around the monitoring threshold, as the monitoring indexes of the other methods. For fault 16, as this fault is not detected by any of the methods, the associated delay time is obviously large. For the rest of the faults, the delay associated with SPA varies from 1 to 5 samples (1 sample is equivalent to 3 minutes), which is not significant if one considers the benefits of having a clear indication that a fault has occurred, as illustrated in Figures 4.11 - 4.13. This behavior is also observed for the rest of the detectable faults.

Table 4.10: Fault detection delay (samples) of PCA, DPCA, KPCA, DPCA, and SPA for the TEP

	PCA		DPCA		KPCA		ICA			SPA	
fault	T^2	SPE	T^2	SPE	T^2	SPE	I_d^2	I_e^2	SPE	D_p	D_r
1	2	1	1	1	1	3	2	1	1	7	5
2	5	7	4	4	4	11	4	3	7	9	11
4	1	1	1	1	1	1	1	1	1	3	2
5	> 70	> 70	> 70	> 70	> 70	> 70	> 70	> 70	> 70	> 70	54
6	1	1	1	1	1	1	1	1	1	3	2
7	1	1	1	1	1	1	11	1	1	2	2
8	10	13	9	9	10	28	> 70	10	13	19	14
10	12	8	10	6	7	14	3	7	9	16	13
11	3	2	1	1	2	3	21	2	2	9	5
12	19	15	10	40	15	> 70	11	14	15	49	44
13	11	11	9	8	8	11	1	9	10	18	15
14	1	1	1	1	1	2	198	1	17	5	3
16	> 70	> 70	> 70	> 70	> 70	> 70	5	> 70	> 70	> 70	> 70
17	8	7	6	4	5	8	29	6	7	14	9
18	39	31	41	22	25	33	5	30	38	52	33
19	7	8	3	6	4	11	8	4	6	29	10
20	10	11	9	8	10	3	1	7	8	18	13

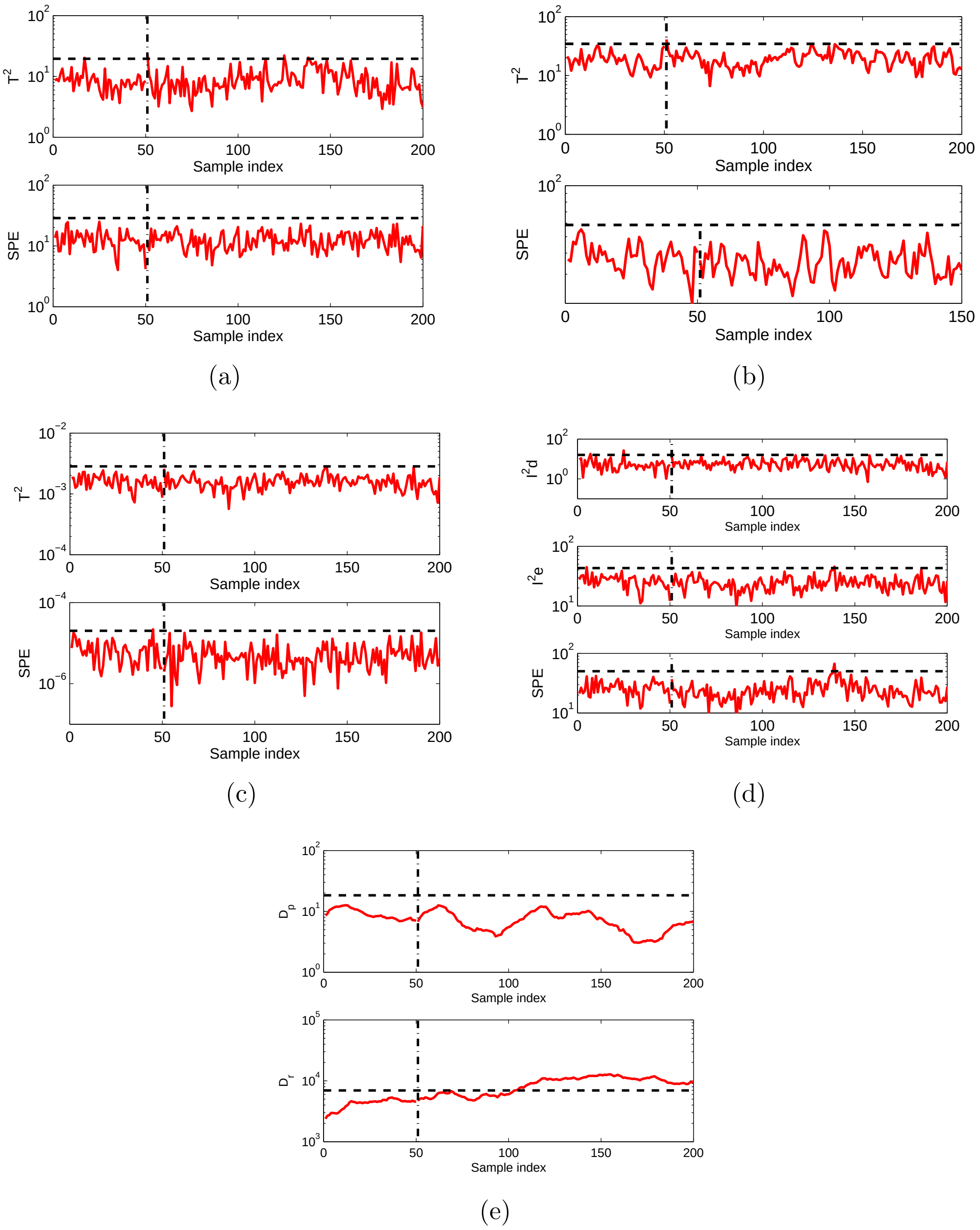
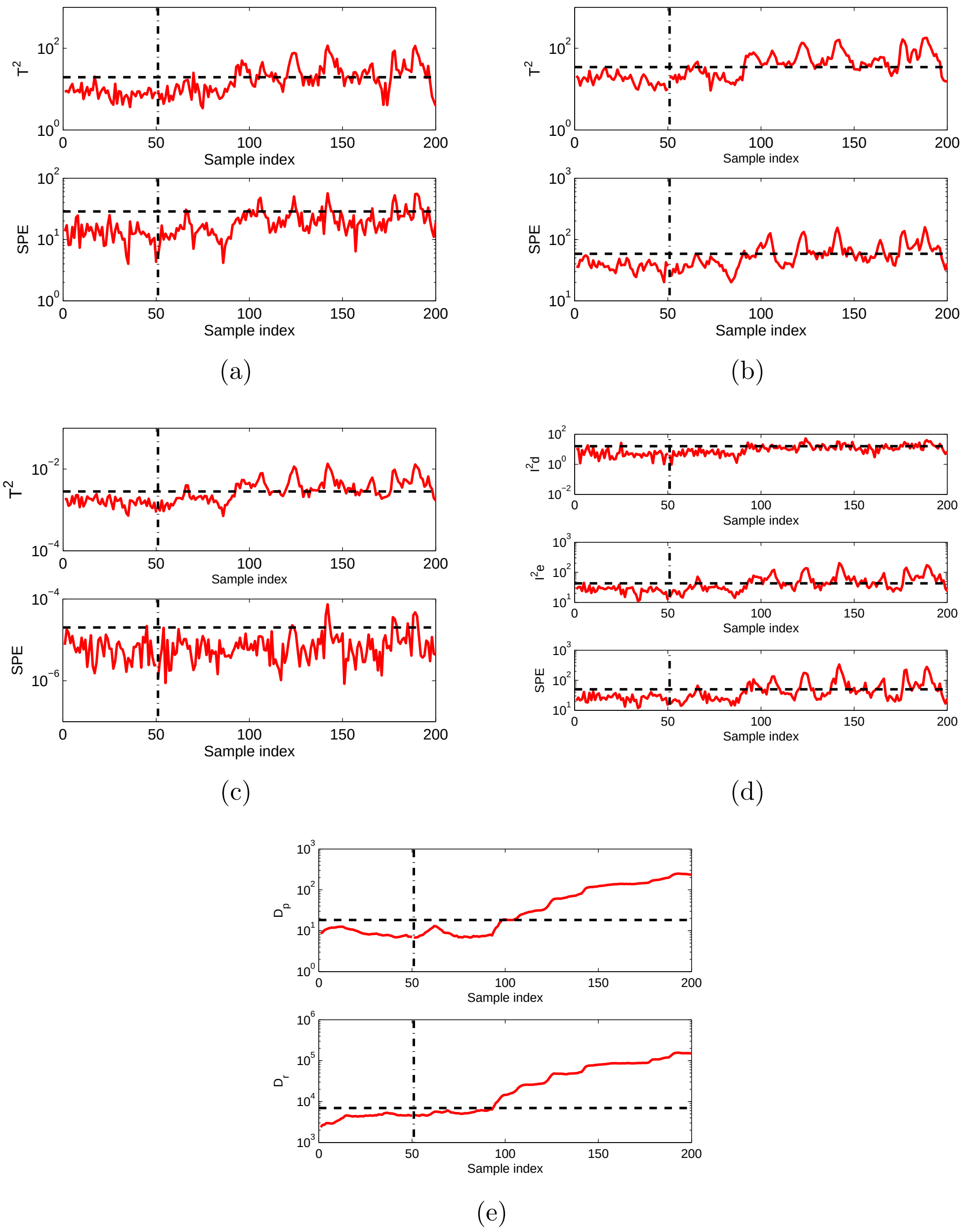


Figure 4.11: Detection performance of PCA, DPCA, KPCA, ICA and SPA for fault 5. (a) PCA; (b) DPCA; (c) KPCA; (d) ICA; (e) SPA. The dash-dotted lines indicate the fault onsets and dashed lines the index thresholds.



(a)

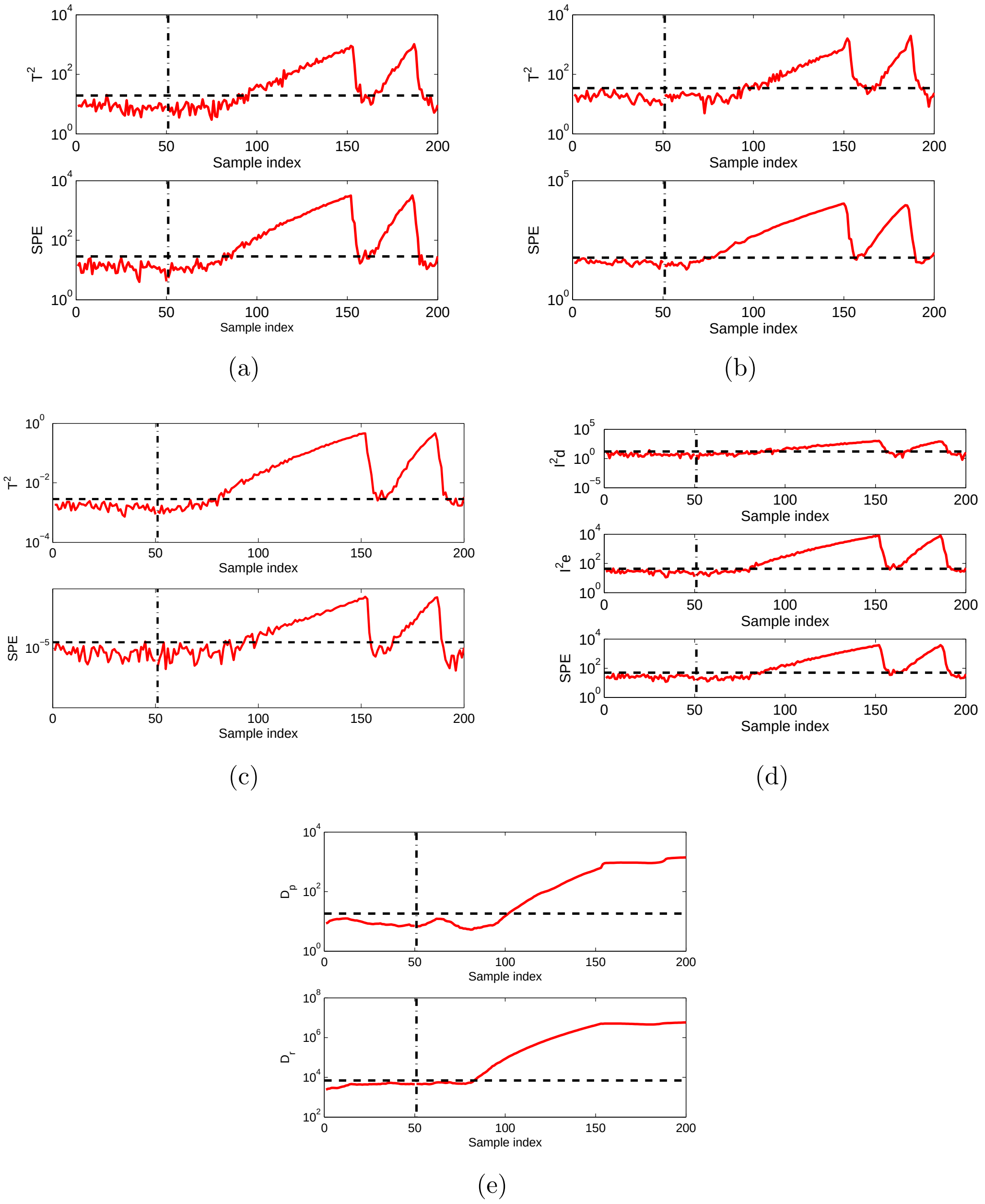
(b)

(c)

(d)

(e)

Figure 4.12: Detection performance of PCA, DPCA, KPCA, ICA and SPA for fault 12. (a) PCA; (b) DPCA; (c) KPCA; (d) ICA; (e) SPA. The dash-dotted lines indicate the fault onsets and dashed lines the index thresholds.



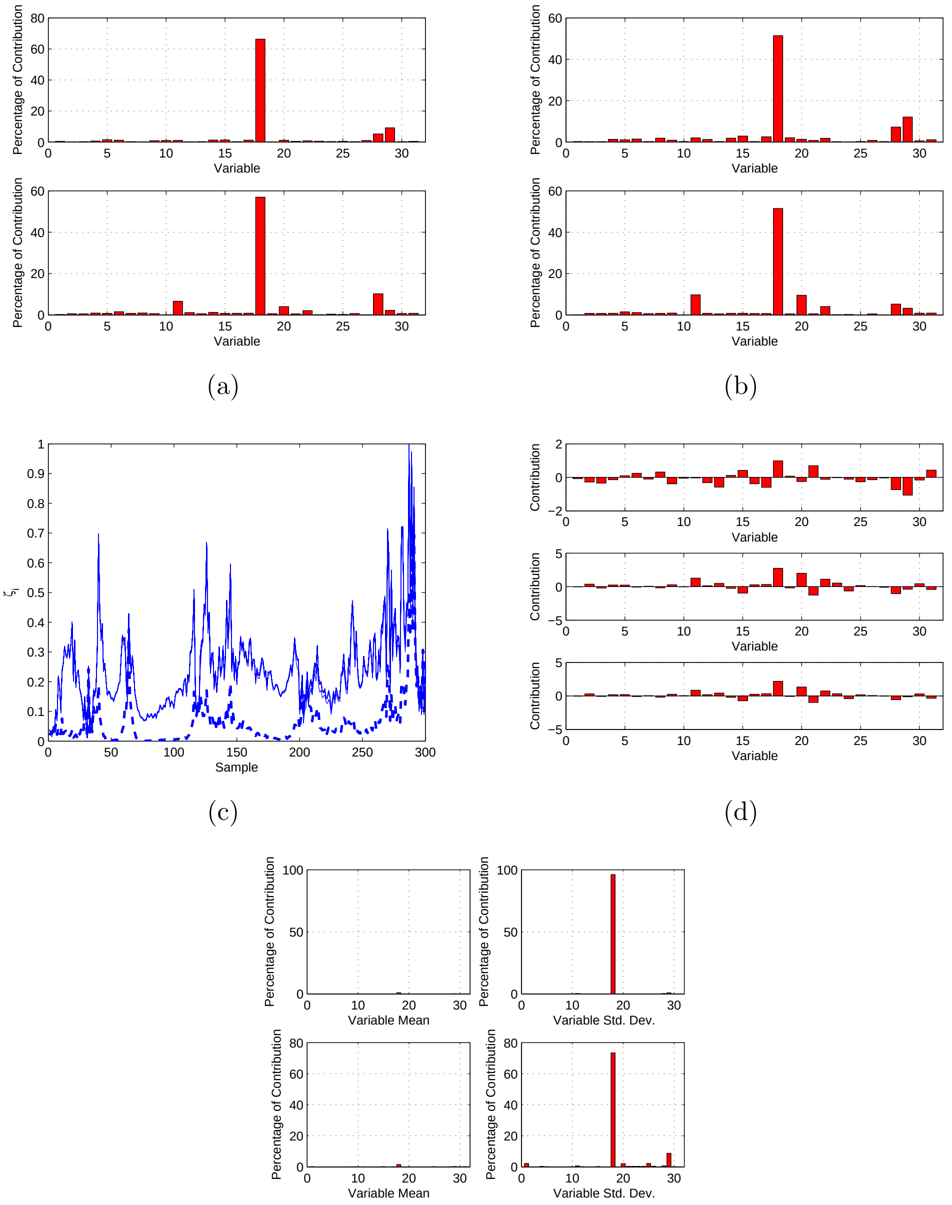
(a) (b) (c) (d) (e)

Figure 4.13: Detection performance of PCA, DPCA, KPCA, ICA and SPA for fault 18. (a) PCA; (b) DPCA; (c) KPCA; (d) ICA; (e) SPA. The dash-dotted lines indicate the fault onsets and dashed lines the index thresholds.

4.6 Fault diagnosis

In this subsection, the proposed SPA based fault diagnosis approach is applied to TEP to identify the root cause of each fault. As a comparison, PCA, DPCA, KPCA, ICA and SPA are also applied to diagnose the faults. For the sake of clarity, the diagnoses of all faults are not discussed in detail except the three illustrative examples given below. It is important to mention that, in general, PCA, DPCA, KPCA, ICA and SPA methods are effective in pinpointing the major fault-contributing process variables for most of the faults. For the case of KPCA diagnosis, the method was found to be quite sensitive to the monitoring settings, i.e. a good set of detection settings might not provide a good diagnosis and vice versa. For the contribution plots based on SPA, two subplots are used to show the contributions from variable mean and standard deviation. The contributions from other statistics such as auto- and cross-correlations are small and are not shown. The example of fault 10 is shown in Figure 4.14(a), where random variation is introduced to C feed temperature (stream 4 to the stripper in Figure 4.10). Figure 4.14(a) shows that PCA, DPCA, KPCA, ICA and SPA methods correctly identify that the variable 18 (stripper temperature) is the major fault-contributing variable. In the case of KPCA, a significant drop for variable 18 is observed in the reconstruction index (e.g. ζ_i), thus indicating that variable 18 is the root cause of the fault (e.g. dashed line in Figure 4.14(c)). Compared to the other diagnosis, the SPA based diagnosis via contribution plots provide extra information by identifying that it was the variance, not the mean, of the variable 18 that contribute to this fault.

There are cases where SPA does a better job in fault detection than the rest of the methods. In these cases, SPA also does a better job in fault diagnosis. One such example is fault 5 shown in Figure 4.15, where a step change occurred in condenser cooling water inlet temperature. Table 4.8 shows that PCA, DPCA, KPCA and ICA cannot detect this fault. Therefore, it is expected that they cannot diagnose the fault either, which is verified by Figures 4.15(a), 4.15(b), 4.15(d) where a range of process variables across different units all contribute to this fault. In the case of KPCA, Figure 4.15(c) shows that variable 31



(a)

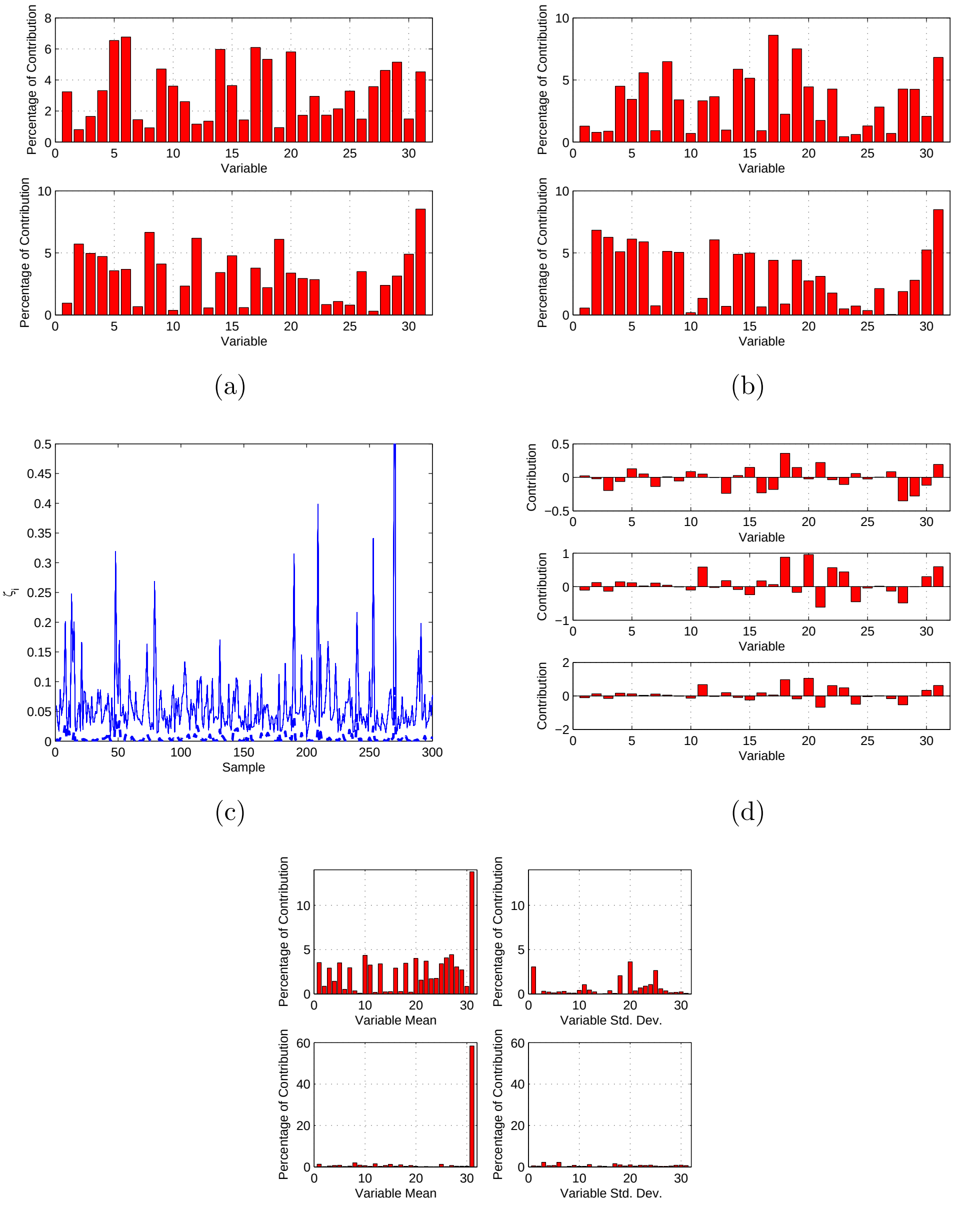
(b)

(c)

(d)

(e)

Figure 4.14: Diagnosis of fault 10. (a) PCA (top T^2, bottom SPE); (b) DPCA (top T^2, bottom SPE); (c) KPCA (reconstruction index ζ_i); (d) ICA(top I_d^2, middle I_d^2 bottom SPE SPE); (e) SPA (top row D_p, bottom row D_r).



(a)

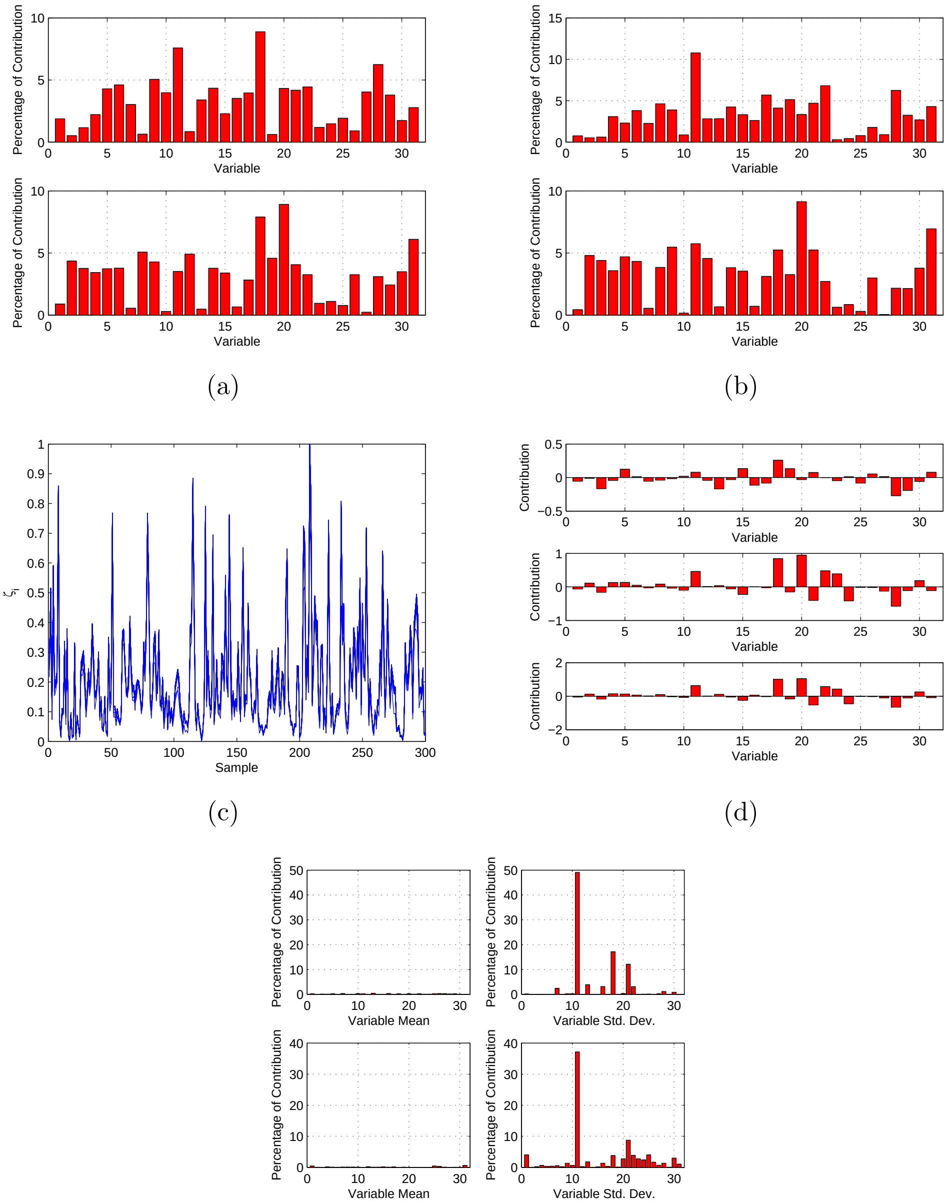
(b)

(c)

(d)

(e)

Figure 4.15: Diagnosis of fault 5. (a) PCA (top T^2, bottom SPE); (b) DPCA (top T^2, bottom SPE); (c) KPCA (reconstruction index ζ_i); (d) ICA(top I_d^2, middle I_d^2 bottom SPE SPE); (e) SPA (top row D_p, bottom row D_r).



(a)

(b)

(c)

(d)

(e)

Figure 4.16: Diagnosis of fault 12. (a) PCA (top T^2, bottom SPE); (b) DPCA (top T^2, bottom SPE); (c) KPCA (reconstruction index ζ_i); (d) ICA(top I_d^2, middle I_d^2 bottom SPE SPE); (e) SPA (top row D_p, bottom row D_r).

observes a significant drop in the reconstruction index for this variable. However, to obtain this diagnosis performance, a different set of diagnosis settings (e.g. width of the gaussian kernel c) was used, which is an adverse factor for online implementation since it introduces more tunable parameters. On the other hand, SPA was able to isolate the fault root cause to the manipulated variable 31, the condenser cooling water flow rate. Since the cooling water inlet temperature is not measured, the manipulated variable 31 is the most related variable. In addition, SPA was able to indicate that a mean change, not variance, in variable 31 is the root cause, which is directly related to the step change in the condenser cooling water inlet temperature.

In the final example of fault 12 (random variation in condenser cooling water inlet temperature) shown in Figure 4.16, the purpose is to illustrate that a successful fault detection does not necessarily lead to a correct fault diagnosis. From Table 4.8, all the five methods were able to detect the fault. However, from Figure 4.16, PCA, DPCA, KPCA and ICA were not able to isolate the root cause of the fault. SPA was able to relate the fault to the separator temperature (variable 11) right after the condenser. Again, since the cooling water temperature or the condenser temperature was not measured and this random variation would not trigger the manipulated cooling water flow rate to change, the downstream separator right after the condenser is the closest to the actual fault location. In other words, the random variation in condenser cooling water inlet temperature (not measured) will immediately lead to random variation in the separator temperature (measured, variable 11). It is also worth noting that SPA correctly identified that the standard deviation, not the mean, of variable 11 is the major fault contributor.

4.7 Conclusion and Discussion

In the first case study, a nonlinear system (quadruple-tank) is simulated to examine the performance of the SPA method. The example demonstrates that SPA is able to handle nonlinear processes, and to detect and diagnose subtle changes in the dynamics of a system.

Additionally, SPA is able to detect and diagnose both faults more efficiently that the PCA and DPCA methods. The two different cases with different size of training samples indicate that SPA can also perform well with a limited amount of data. However, for the case where data are limited, a window overlapping is necessary. The window overlapping causes the samples to be no longer independent from each other which causes an increment in the false alarm rates for normal operating data. For the case where enough data are available, SPA not only is able to provide high detection rates and a more clear identification of a fault, but it also reduces the false alarms for normal operating data considerably. It is worth noting that the cost associated with using more SPs is that a large amount of data are needed for model building, as the number of SPs is only $1/w$ of the number of original training samples, where w is the size of the window. However, for most industrial processes this limitation can be overcome since large amounts of historic data are available. Additionally, because of the window-based approach, there is limited detection delay associated with the SPA method. In fact, the detection delay is comparable to that the other monitoring methods, because the window from SPA can be shifted by one sample during the online monitoring once the model is build.

In the second case study, the potential of the SPA framework in fault detection and diagnosis at a plant-wide level was explored. The Tennessee Eastman process is used to evaluate the performance of the developed fault detection and diagnosis method based on the SPA framework, and compared with the two commonly applied linear monitoring methods (e.g. PCA and DPCA) and nonlinear monitoring methods (e.g. ICA and KPCA). The case study shows that in general all five methods work well in detecting and diagnosing most of faults. However, for some faults that are difficult to detect and/or diagnose by PCA, DPCA, KPCA and ICA based methods, SPA based method provides superior performance. As the SPA-based process monitoring for continuous processes is a window-based approach, detection delay is associated with its application. In this case study, it was shown that the detection delay associated with SPA is not significant, as the window can be shifted by one online.

In addition, the detection of faults is much more clear than other methods and does not observe drastic changes in the monitoring index around the threshold. For fault diagnosis, because the SPA-based method breaks down the contribution of a fault to different variable statistics, it provides extra information in addition to identifying the major fault-contributing variable(s). For example, the SPA-based contribution plots tell us whether the fault is due to a change in variable mean or variance. It also should be noted that in general SPA requires more training data to build a reliable model due to the computation of variable statistics. However, this is not a big issue because most modern processes are equipped with DCS systems and therefore are data rich.

Chapter 5

Partial Least Squares Algorithms

Given the importance of partial least squares (PLS) and recursive PLS in the development of the second part of this work, in this chapter, the PLS algorithm and different recursive model update approaches are briefly discussed. Specifically, the main characteristics of regular recursive PLS (RPLS), block-wise RPLS (BRPLS), moving window block-wise RPLS (MW-BRPLS), and forgetting factor block-wise RPLS (FF-BRPLS) are reviewed. In addition, the data scaling methods that accompany the different algorithms are developed and discussed.

5.1 PLS Regression

PLS regression have arisen as a useful alternative for analyzing systems of secondary measurements **X** with strongly collinear dependency, highly correlated, noisy, and of high dimensionality, typical characteristics of data from industrial processes [19, 22, 24, 25, 51, 136]. In PLS, the interest is using the variables in the **X** matrix (i.e., easy-to-measure) to predict the behavior of one or several primary variables **Y** (e.g. quality or yield), which is one of the most common type of problem in science and engineering [24]. Multiple linear regression (MLR) is the most basic form of model to achieve this goal. However, due to correlations in process variables, size of data set, and measurement noise, MLR methods lead to imprecise estimations and poor prediction [26, 40]. In PLS , the properties of the NIPALS [24] algorithm are utilized to let the score matrix **T** (e.g. projections of the **X** matrix onto the plane defined by the characteristic directions of variability) to represent the original data matrix **X**. The scores **T** of **X** are calculated in such a way that they are good predictors of **Y**. In other words, the decompositions of the matrix **X** and matrix **Y** are done in such a way as to maximize the covariance between them [136]. In this way, the first few retained latent variables contain enough information to characterize $\mathbf{X}$, and at the same time they contain useful predictive data, providing both, a dimensionality reduction and numerical robustness compared to MLR [136]. The PLS algorithm, as developed by Wold et al. [24] is briefly reviewed below. Details of the algorithm and its properties can be found in [19, 24, 25].

Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{n \times p}$ denote the input and output data matrices; where n is the number of samples, m the number of independent variables, and p the number of dependent variables.

Assume that $\mathbf{X}$ and $\mathbf{Y}$ are linearly related by:

$$\mathbf{Y} = \mathbf{XC} + \mathbf{V} \tag{5.1}$$

where $\mathbf{C}$ is the coefficient matrix and $\mathbf{V}$ is the noise matrix with appropriate dimensions. The PLS regression method derives the linear model by first decomposing data matrices $\mathbf{X}$ and $\mathbf{Y}$ into bilinear terms:

$$\mathbf{X} = \mathbf{TP^T} + \mathbf{E} \tag{5.2}$$

$$\mathbf{Y} = \mathbf{UQ^T} + \mathbf{F} \tag{5.3}$$

where $\mathbf{T} \in \mathbb{R}^{n \times a}$ and $\mathbf{U} \in \mathbb{R}^{n \times a}$ are the score matrices of $\mathbf{X}$ and $\mathbf{Y}$ respectively, and a is the number of retained PLS components. Matrix $\mathbf{P} \in \mathbb{R}^{m \times a}$ and $\mathbf{Q} \in \mathbb{R}^{p \times a}$ are known as the loading matrices. Matrices $\mathbf{E}$ and $\mathbf{F}$ are the corresponding residual matrices with appropriate dimensions. Equations (5.2) and (5.3) formulate the PLS outer model. The final goal of PLS is to describe as much variation of $\mathbf{Y}$ as possible and at the same time get a useful relation between $\mathbf{X}$ and $\mathbf{Y}$ [19]. This is done by relating the scores of $\mathbf{X}$ and $\mathbf{Y}$ with a linear inner model:

$$\mathbf{U} = \mathbf{TB} + \mathbf{R} \tag{5.4}$$

where $\mathbf{B} \in \mathbb{R}^{a\times a}$ is a diagonal matrix of inner model coefficients that are obtained by minimizing the residual $\mathbf{R}$.
The estimate $\widehat{\mathbf{Y}}$ of $\mathbf{Y}$ is obtained consequently:

$$\widehat{\mathbf{Y}} = \mathbf{TBQ}^T + \mathbf{F} \tag{5.5}$$

For convenience, in this work the following notation is used to represent the modeling procedure that obtains the PLS model $\{\mathbf{T}, \mathbf{P}, \mathbf{U}, \mathbf{Q}, \mathbf{B}\}$ from the pair of data matrices $\{\mathbf{X}, \mathbf{Y}\}$

$$\{\mathbf{X}, \mathbf{Y}\} \overset{PLS}{\rightarrow} \{\mathbf{T}, \mathbf{P}, \mathbf{U}, \mathbf{Q}, \mathbf{B}\} \tag{5.6}$$

Examples of the application of PLS regression for soft sensor development are given in the subsequent chapters.

5.2 Recursive PLS

In traditional PLS, a static model is built using a training data set, which is later used for predicting future observations. If either new information is introduced or the conditions reflected in the training data change, a new model needs to be trained using both the old and the new training samples to adequately capture the characteristics of the process [137]. If this approach is followed continuously, the amount of training data for update will become an issue very quickly. Helland et al. [137] proposed a recursive PLS (RPLS) algorithm that efficiently overcomes the issue of data build up. In their work, it is proposed that instead of using the old data and the new data to rebuild a new PLS model, the old PLS model derived from the old data is augmented with the new data to obtain an updated PLS model. However, Helland's RPLS approach suffers from numerical instability for rank deficient data. This issue was later addressed by Qin in [138]. In this work, Qin's approaches are implemented to update the RO-DPLS model when new data becomes available. In this and the following sections, Qin's approaches are briefly reviewed.

If $\{\mathbf{X}, \mathbf{Y}\}$ is used to denote the old data, and $\{\mathbf{X_1}, \mathbf{Y_1}\}$ is used to denote the new data, it was shown in [138] that performing PLS on a data pair:

$$\left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{X_1} \end{bmatrix}, \begin{bmatrix} \mathbf{Y} \\ \mathbf{Y_1} \end{bmatrix} \right\}$$

results in the same regression model as performing PLS on the data pair:

$$\left\{ \begin{bmatrix} \mathbf{P^T} \\ \mathbf{X_1} \end{bmatrix}, \begin{bmatrix} \mathbf{BQ^T} \\ \mathbf{Y_1} \end{bmatrix} \right\} \tag{5.7}$$

provided that the residual matrix $\mathbf{E}$ of $\mathbf{X}$ in Equation (5.2) is essentially zero. If n and n_1 are used to denote the number of samples in the old data set and the new data set respectively, the RPLS approach updates the model using a PLS run size of $(a + n_1)$, while the regular PLS updates the model with a run size of $(n+n_1)$, where a is the number of PLS components used to replace the input matrix $\mathbf{X}$. It is clear that RPLS is much more efficient than the regular PLS if $n >> a$, which is typical in PLS applications [138].

5.3 Block-wise RPLS

To further improve computation efficiency, Qin [138] also introduced the block-wise RPLS (BRPLS), where a PLS sub-model is first built using the new data pair $\{\mathbf{X_1}, \mathbf{Y_1}\}$, i.e., $\{\mathbf{X_1}, \mathbf{Y_1}\} \overset{PLS}{\to} \{\mathbf{T_1}, \mathbf{P_1}, \mathbf{U_1}, \mathbf{Q_1}, \mathbf{B_1}\}$; then it is combined with the existing model to derive the updated PLS model. It was shown in [138] that performing PLS on the data pair:

$$\left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{X_1} \end{bmatrix}, \begin{bmatrix} \mathbf{Y} \\ \mathbf{Y_1} \end{bmatrix} \right\}$$

results in the same regression model as performing PLS on the data pair:

$$\left\{ \begin{bmatrix} \mathbf{P^T} \\ \mathbf{P_1^T} \end{bmatrix}, \begin{bmatrix} \mathbf{BQ^T} \\ \mathbf{B_1 Q_1{}^T} \end{bmatrix} \right\} \tag{5.8}$$

provided that the residual matrices of $\mathbf{X}$ and $\mathbf{X_1}$ are essentially zero, i.e., $\mathbf{E} = \mathbf{0}$ and $\mathbf{E}_1 = 0$.

5.4 Block-wise RPLS with Moving Window

To adequately adapt process changes, it is desirable to exclude older data from the model by incorporating a moving window approach. Given $\{\mathbf{X_i}, \mathbf{Y_i}\} \overset{PLS}{\rightarrow} \{\mathbf{T_i}, \mathbf{P_i}, \mathbf{U_i}, \mathbf{B_i}, \mathbf{Q_i}\}$, following [138], the MW-BRPLS performs PLS regression on the following pair of matrices:

$$\left\{ \begin{bmatrix} \mathbf{P_i^T} \\ \mathbf{P_{i-1}^T} \\ \vdots \\ \mathbf{P_{i-w+1}^T} \end{bmatrix}, \begin{bmatrix} \mathbf{B_i Q_i^T} \\ \mathbf{B_{i-1} Q_{i-1}^T} \\ \vdots \\ \mathbf{B_{i-w+1} Q_{i-w+1}^T} \end{bmatrix} \right\} \tag{5.9}$$

When the new data pair $\{\mathbf{X_{i+1}}, \mathbf{Y_{i+1}}\}$ becomes available, a PLS sub-model is first derived:

$$\{\mathbf{X_{i+1}}, \mathbf{Y_{i+1}}\} \overset{PLS}{\rightarrow} \{\mathbf{T_{i+1}}, \mathbf{P_{i+1}}, \mathbf{U_{i+1}}, \mathbf{B_{i+1}}, \mathbf{Q_{i+1}}\}$$

then, $\mathbf{P}_{i+1}^T$ and $\mathbf{B_{i+1} Q_{i+1}^T}$ are augmented into the top row of the matrices in Equation 5.9 and the bottom row is dropped out. The window size w controls how fast the old data are excluded from the model. A narrow window makes the model adapt faster to the new data.

5.5 Block-wise RPLS with Forgetting Factor

An alternative approach for online adaptation is to use a forgetting factor. Following [138], the PLS model at step i is obtained by performing PLS on the following data pair:

$$\left\{ \begin{bmatrix} \mathbf{P_i^T} \\ \lambda \mathbf{P_{ic}^T} \end{bmatrix}, \begin{bmatrix} \mathbf{B_i Q_i^T} \\ \lambda \mathbf{B_{ic} Q_{ic}^T} \end{bmatrix} \right\} \tag{5.10}$$

where the subscript ic denotes the combined model obtained at step $i - 1$, and λ is the corresponding forgetting factor. A λ close to zero forgets the old data faster.

5.6 Online Data Scaling

The PLS and RPLS algorithms discussed before are derived with the assumption that the data pair $\{\mathbf{X}, \mathbf{Y}\}$ is scaled to zero mean and unit variance, so that different variables will equally influence the regression model. In addition, the linear relationship between $\mathbf{X}$ and $\mathbf{Y}$ extracted from the process data can be easily distorted by measurement noise, the presence of outliers, and/or autocorrelated measurements. Therefore, data preprocessing is an important step in building a data-driven PLS soft sensor. The commonly applied data preprocessing steps include data scaling, outlier detection, and data denoising [139]. For off-line model building, the data preprocessing is usually an iterative process, i.e., the preprocessing steps such as outlier detection are repeated several times until the data obtained can be used for model training, testing, and validation. Currently, data preprocessing, especially outlier detection, is the step that requires significant amount of manual work and expert knowledge about the underlying process characteristics [140]. For online applications, data preprocessing becomes more challenging because it is generally impossible to repeat the data preprocessing steps with plant expert inputs as in off-line model building.

The most commonly applied scaling method is normalization or autoscaling, i.e., removing variable means and scaling variables to unit variance. For off-line PLS model building, the mean and the variance of the n_0 training samples are calculated by:

$$\overline{x}_0 = \frac{1}{n_0}\sum_{i=1}^{n_0} x_i \tag{5.11}$$

$$\sigma_0^2 = \frac{1}{n_0 - 1}\sum_{i=1}^{n_0} (x_i - \overline{x}_0)^2 \tag{5.12}$$

Because many processes are time varying, the means and variances of the process variables usually change over time, correspondingly the means and variances should be updated as well. In this work, three different preprocessing approaches that accompany the different model update approaches are considered.

1. *Recursive scaling and centering (RSC)*:

   For each variable

$$\overline{x}_n = \frac{n-1}{n}\overline{x}_{n-1} + \frac{1}{n}x_n \tag{5.13}$$

$$\sigma_n^2 = \frac{n-2}{n-1}\sigma_{n-1}^2 + \frac{1}{n-1}\left(x_n - \overline{x}_n\right)^2 + \left(\overline{x}_n - \overline{x}_{n-1}\right)^2 \tag{5.14}$$

   where n is the total number of the old and new data points; $\overline{x}_n$ and σ_n are the updated mean and standard deviation. The detailed derivation of Equations (5.13) and (5.14) is given in Appendix C. Note that the commonly applied formula for variance update is the following [139]:

$$\sigma_n^2 = \frac{n-2}{n-1}\sigma_{n-1}^2 + \frac{1}{n-1}\left(x_n - \overline{x}_n\right)^2 \tag{5.15}$$

   which is not exactly correct. However, by comparing Equation (5.14) with Equation (5.15), the only difference is the term $(\overline{x}_n - \overline{x}_{n-1})^2$. This term can be neglected when n is large, because the ratio of this term to the second term in Equation (5.14) can be calculated as

$$\frac{(\overline{x}_n - \overline{x}_{n-1})^2}{\frac{1}{n-1}(x_n - \overline{x}_n)^2} = \frac{1}{n-1} \tag{5.16}$$

   which approaches zero when n is large. Therefore, Equation (5.15) can be used as a good approximation to update the variance for large n. Detailed derivation of Equation (5.16) is given in Appendix D. It is worth noting that one drawback with this approach is that n keeps increasing as more data becomes available and the influence of the new data point diminishes with increased n.

   For block-wise algorithms, the variable mean and variance can be updated using the following equations

$$\overline{x}_n = \frac{n-n_1}{n}\overline{x}_{n-n_1} + \frac{n_1}{n}\overline{x}_{n,B} \tag{5.17}$$

$$\sigma_n^2 = \frac{n-n_1-1}{n-1}\sigma_{n-n_1}^2 + \frac{n_1-1}{n-1}\sigma_{n,B}^2 + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1} - \overline{x}_{n,B}\right)^2 \tag{5.18}$$

where n_1 is the number of samples in the new data block; $\overline{x}_{n,B}$ and $\sigma^2_{n,B}$ are the new data block mean and variance. The detailed derivation of Equations (5.17) and (5.18) is given in Appendix E.

2. *Moving window scaling and centering (MWSC)*:

   In this case, since the oldest measurement in the window is removed with the newest one added, the total number of measurements stays constant. One can recalculate the mean or variance for the measurements in the window, or use the following approximation.

$$\overline{x}_n = \frac{w-1}{w}\overline{x}_{n-1} + \frac{1}{w}x_n \tag{5.19}$$

$$\sigma^2_n = \frac{w-2}{w-1}\sigma^2_{n-1} + \frac{1}{w-1}\left(x_n - \overline{x}_n\right)^2 \tag{5.20}$$

   where w is the window width which stays constant. In MWSC, the effect of the new sample will not diminish as more data becomes available.

   For block-wise algorithms with block size n_1, the approximation becomes:

$$\overline{x}_n = \frac{w-1}{w}\overline{x}_{n-n_1} + \frac{1}{w}\overline{x}_{n,B} \tag{5.21}$$

$$\sigma^2_n = \frac{(w-1)n_1-1}{wn_1-1}\sigma^2_{n-n_1} + \frac{n_1-1}{wn_1-1}\sigma^2_{n,B} \tag{5.22}$$

   where, $\overline{x}_{n,B}$ and $\sigma^2_{n,B}$ are the new data block mean and variance.

3. *Forgetting factor scaling and centering (FFSC)*:

   For each variable:

$$\overline{x}_n = \alpha\overline{x}_{n-1} + (1-\alpha)\,x_n \tag{5.23}$$

$$\sigma^2_n = \alpha\sigma^2_{n-1} + (1-\alpha)\left(x_n - \overline{x}_n\right)^2 \tag{5.24}$$

For block-wise algorithms:

$$\overline{x}_n = \alpha \overline{x}_{n-n_1} + (1-\alpha)\, \overline{x}_{n,B} \tag{5.25}$$

$$\sigma_n^2 = \alpha \sigma_{n-n_1}^2 + (1-\alpha)\, \sigma_{n,B}^2 \tag{5.26}$$

In the following chapters, the properties of PLS are used to develop different software sensors, and the properties of the different methods reviewed in this chapter are discussed in detail.

Chapter 6

A Reduced-Order Dynamic PLS Soft Sensor Approach

6.1 Introduction

Pulping process, which converts wood chips into pulp by displacing lignin from cellulose fibers, is one of the most important operations in a pulp and paper mill [141]. Yearly US wood pulp production is around 52 million tons and the pulp and paper industry is an important sector of the US manufacturing [142, 143]. Kraft pulping is the most commonly used chemical pulping process, which usually utilizes a continuous Kamyr digester. A Kamyr digester is a complex vertical plug flow reactor where the wood chips react with an aqueous solution of sodium hydroxide and sodium sulfide, also known as white liquor, at elevated temperatures. Fig. 6.1 shows the schematic diagram of a single-vessel Kamyr digester with three distinct zones. Pre-steamed wood chips and white liquor enter the impregnation zone from the top of the digester, and chemicals in white liquor diffuse into the pores of the wood chips in this zone. Subsequently in the cook zone, the temperature is raised through the upper and lower heated circulation and most of the delignification reactions occur. Finally in the wash zone, the countercurrent flow of dilute wash liquor quenches the reactions and removes the unreacted chemicals. It is well known that the performance of the pulping process is of paramount importance to maximize the pulp quality and yield, reduce the overall operating cost including energy usage, and minimize the adverse environmental impacts of pulp mills [142]. The most important quality variable in pulping process is the Kappa number, which represents the residual lignin content of the pulp [142]. It is desired to minimize the variations of the Kappa number in the pulp product. For a continuous digester, process variables are measured at different frequencies. Primary measurement, *i.e.*, the pulp Kappa

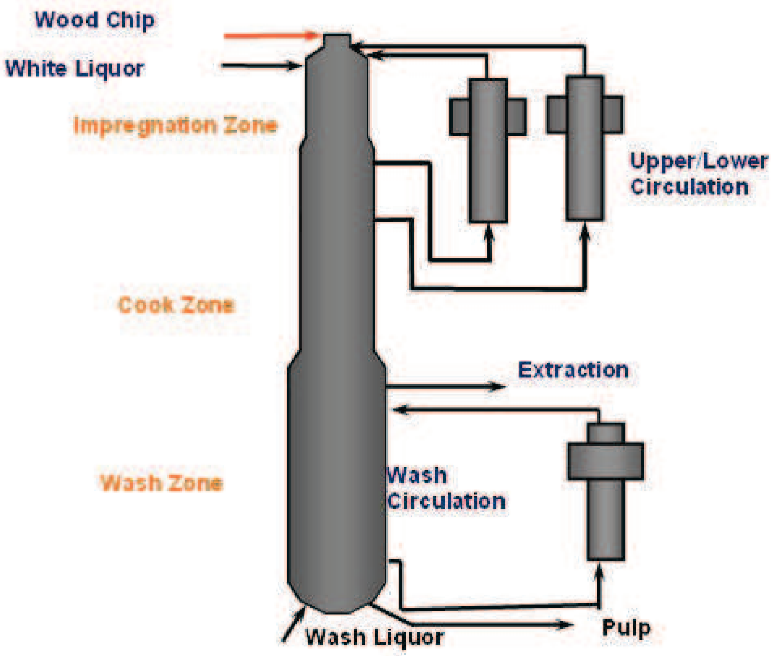


Figure 6.1: Schematic of a high-yield single-vessel Kamyr digester

number, is measured at a low frequency (*e.g.*, one sample every 2 hrs with 30 mins measurement delay); while secondary measurements are measured at a higher frequency (*e.g.*, one per 6 to 10 minutes) which include white liquor flow rate, wood chip flow rate, the temperature and composition of upper circulation, lower circulation and extraction [144]. In order to describe the dynamics of a continuous digester and develop effective control methods, first principles or physical models have been developed to simulate the pulping process by approximating the digester with a series of continuous stirred tank reactors (CSTRs) [145, 146]. However, control of a continuous digester is challenging due to the following characteristics of the pulping process: (1) long residence time or delay; (2) lack of real-time measurement of the extent of reaction (i.e., Kappa number); (3) complex nonlinear dynamic behavior; (4) unmeasured stochastic disturbances. Some of these challenges are also present in other industrial processes such as distillation columns [46, 48].

To address the challenge that the primary variable (*i.e.*, the Kappa number) is not measured online but is required for feedback control, soft sensors (or inferential estimators) have been developed in digester control to estimate the Kappa numbers based on the secondary measurements [147–153]. Soft sensors can be developed in the form of a Luenberger observer [37] or a Kalman filter [38] using a first principles dynamic model of the process. However, because chemical processes are generally quite complex to model and are characterized by significant inherent nonlinearities, a rigorous theoretical modeling approach is often impractical, requiring a great amount of effort [39]. Consequently, there has been an increasing interest toward the development of data-driven soft sensors using multivariate regression techniques recently [8, 9, 28, 39, 41–49]. Because data-driven soft sensors are easy to develop and to implement online, they are potentially more attractive than stochastic filters or deterministic observers [39]. Artificial neural networks (ANN), principal component regression (PCR) or principal component analysis (PCA), and partial least squares (PLS) are widely used regression techniques, and their successful application to the development of soft sensors has been reported for different processes [9, 47, 50–54]. Dynamic PCA (DPCA) and Dynamic PLS (DPLS), the extensions of the conventional PCA and PLS algorithms for dynamic modeling, were studied through the augmentation of the original input data matrices with lagged measurements [29, 55]. Recursive PCA/PLS algorithms have also been proposed to deal with time-varying characteristics of industrial processes [138, 154, 155]. Because the DPLS soft sensor is the focus of this work, some representative work in this area is briefly reviewed.

Lakshminarayanan et al. [44] proposed using DPLS to the identification and control of multivariate systems. In [44], the linear and nonlinear DPLS models were integrated into the model predictive control (MPC) framework. Kano et al. [48] compared steady-state, static and dynamic PLS-based soft sensors and found that dynamic soft sensor DPLS could greatly improve the estimation accuracy. Kano and others [48] also applied the DPLS soft sensor to estimate the product compositions of a multicomponent distillation column from online measured process variables such as tray temperatures, reflux flow rate, reboiler heat duty and pressure. Zamprogna et al. [46] exploited DPCA and DPLS soft sensors to estimate the product composition profiles in a simulated batch distillation process using temperature measurements. It was also shown in [46] that linear DPLS performed at least as good as nonlinear DPLS algorithms (polynomial, spline or ANN DPLS) in the batch distillation process case study. More recently, DPLS soft sensors have been applied to several industrial processes, including a specialty chemical batch process [56], a splitter and a crude columns [156], a dearomatization process [57] and a cement kiln system [58].

Despite many successful applications, one limitation associated with DPLS is that it is usually viewed as an ad hoc method as there is a lack of theoretical understanding on whether it can capture process dynamics adequately and whether it can provide unbiased estimate for both open and closed-loop data. Another problem encountered when performing DPLS is that the addition of a relatively large number of lagged values in the input data matrix causes a substantial increase in the dimension of the matrix. This implies that the computational load required to the development of the multivariate regression becomes considerably heavier, and the regression procedure gets much lengthier. This is particularly true for spline PLS and ANN PLS, which are based on computationally more intensive algorithms. As a result, the model becomes more cumbersome to manipulate due to the presence of a higher number of parameters [46]. As can be seen later in this work, when process residence time is long, which is true for all continuous digesters, large number of lagged measurements and more training data are needed for DPLS to capture the process dynamics and to provide satisfactory soft sensor performance.

To address these issues, in this work, a rigorous proof is derived to show that: (1) DPLS-based soft sensor is a dynamic estimator that can adequately capture process dynamics; (2) DPLS-based soft sensor can provide unbiased estimate of the primary variable for both open loop and closed-loop data. Next, a reduced order DPLS (RO-DPLS) approach with optimally selected lagged measurements of the secondary variables is proposed. The RO-DPLS soft sensor not only reduces model size and improves prediction but also significantly increases prediction horizon [63–66].

6.2 The Relationship of DPLS Soft Sensor and Subspace Identification

It has been widely assumed and accepted that DPLS-based soft sensors can capture the processes dynamics, and DPLS-based soft sensor approaches have been used rather commonly in industrial applications. The basic idea behind the DPLS soft sensor, i.e., using augmented measurements (or "lagged data") to capture process dynamics, is not new and has been widely used in system identification, such as subspace identification where extended state-space model is used to derive how to extract system matrices. In addition, this idea has been used in process monitoring where dynamic PCA is used to remove the time dependence of the self-correlated and cross-correlated dynamic process data. However, to the best of the authors knowledge, there has not been any rigorous proof to show how the process dynamics is captured by the static linear regression employed in the DPLS-based soft sensor. Therefore, compared to Luenberger observer or Kalman filter, DPLS has been commonly viewed as an ad hoc approach which uses lead-lag elements determined by cross-validation to handle process dynamics [152]. In addition, it has been assumed that DPLS soft sensor works for closed-loop data and successful applications exist, but there has not been any proof that in theory DPLS-based soft sensor can provide unbiased estimate for closed-loop data. In this work, a rigorous theoretical analysis is provided to show that for linear time invariant (LTI) systems the DPLS soft sensor can adequately capture process dynamics and provide unbiased estimate of the primary variable for both open loop and closed-loop data.

In the following, the DPLS soft sensor formulation is derived using the subspace identification (subspace ID) framework. Then, based on the derived soft sensor formulation, it is shown that unbiased estimates of the primary variable can be obtained from DPLS soft sensor under both open loop and closed-loop operations.

In subspace ID, a stochastic linear system can be represented in the following innovation form, which is utilized in several subspace ID algorithms such as MOESP [157] and

SIMPCA [158].

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}\mathbf{e}_k^s \\ \mathbf{y}_k^s &= \mathbf{C}^s\mathbf{x}_k + \mathbf{e}_k^s \\ \mathbf{y}_k^c &= \mathbf{C}^c\mathbf{x}_k + \mathbf{e}_k^c \end{aligned} \tag{6.1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^m$ are the system state and input; $\mathbf{y}_k^c \in \mathbb{R}^q$ and $\mathbf{y}_k^s \in \mathbb{R}^l$ are the primary and secondary outputs; the innovations $\mathbf{e}_k^s \in \mathbb{R}^l$ and $\mathbf{e}_k^c \in \mathbb{R}^q$ are white noise; $\mathbf{A},\ \mathbf{B},\ \mathbf{C^s},\ \mathbf{C^c}$ are system matrices with appropriate dimensions; $\mathbf{K}$ is the steady-state Kalman filter gain. The system represented in (6.1) is assumed to be minimal in the sense that $(\mathbf{A},\ \mathbf{C^s})$ is observable and $(\mathbf{A},\ [\mathbf{B},\ \mathbf{K}])$ is controllable; in addition, the eigenvalues of $\mathbf{A} - \mathbf{K}\mathbf{C^s}$ are assumed to be strictly inside the unit circle. In other words, it is assumed that the secondary variables provide enough information to describe the dynamics of the overall system. This assumption is reasonable, because if the primary variables provide information that the secondary variables do not contain, one cannot expect to estimate the primary variables accurately using the secondary variables. Without loss of generality, the following linear relationship between the two innovation terms is assumed: $\mathbf{e}_k^c = \mathbf{F}\mathbf{e}_k^s$, which can be justified with the assumption that $(\mathbf{A},\ \mathbf{C^s})$ is observable as shown in Appendix A.

In order to derive the dynamic soft sensor formulation, the innovation form is transformed into its predictor form, as used in [159].

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_K\mathbf{x}_k + \mathbf{B}_K\mathbf{z}_k \\ \mathbf{y}_k^s &= \mathbf{C}^s\mathbf{x}_k + \mathbf{e}_k \\ \mathbf{y}_k^c &= \mathbf{C}^c\mathbf{x}_k + \mathbf{e}_k^c \end{aligned} \tag{6.2}$$

where $\mathbf{z}_k = \begin{bmatrix} \mathbf{u}_k^T, & (\mathbf{y}_k^s)^T \end{bmatrix}$ is the augmented process input and secondary output; $\mathbf{A}_K = \mathbf{A} - \mathbf{K}\mathbf{C}^s$ and $\mathbf{B}_K = [\ \mathbf{B},\ \ \mathbf{K}\ ]$ are predictor matrices. The predictor form has the numerical advantage for identifying both stable and unstable processes because $\mathbf{A}_K = \mathbf{A} - \mathbf{K}\mathbf{C}^s$ is guaranteed to be stable even though the process matrix $\mathbf{A}$ may be unstable, while the innovation form may lead to ill-conditioning for unstable processes [159].

By iterating the state equation in Eqn.(6.2), it is straightforward to derive the extended state equation,

$$\mathbf{x}_k = \mathbf{L}_p^* \mathbf{z}_p(k) + \mathbf{A}_K^p \mathbf{x}_{k-p} \approx \mathbf{L}_p^* \mathbf{z}_p(k) \text{ for large } p \tag{6.3}$$

where $\mathbf{L}_p^* = \begin{bmatrix} \mathbf{B}_K & \mathbf{A}_K\mathbf{B}_K & \cdots & \mathbf{A}_K^{p-1}\mathbf{B}_K \end{bmatrix}$, the subscript p denotes the past horizon, and the past data $\mathbf{z}_p$ is defined as

$$\mathbf{z}_p(k) = \begin{bmatrix} \mathbf{z}_{k-1} \\ \mathbf{z}_{k-2} \\ \vdots \\ \mathbf{z}_{k-p} \end{bmatrix} \tag{6.4}$$

Similarly, based on the state-space description in Eqn.(6.2), the extended output equation can be obtained

$$\mathbf{y}_f^s(k) = \Gamma_f^{*,s} \left(\mathbf{L}_p^* \mathbf{z}_p(k) + \mathbf{A}_K^p \mathbf{x}_{k-p} \right) + \mathbf{G}_f^{*,s} \mathbf{z}_{f-1}(k) + \mathbf{e}_f(k) \tag{6.5}$$

$$\mathbf{y}_f^c(k) = \Gamma_f^{*,c} \left(\mathbf{L}_p^* \mathbf{z}_p(k) + \mathbf{A}_K^p \mathbf{x}_{k-p} \right) + \mathbf{G}_f^{*,c} \mathbf{z}_{f-1}(k) + \mathbf{e}_f^c(k) \tag{6.6}$$

where the subscript f denotes the future horizon; $\Gamma_f^{*,i}$ and $\mathbf{G}_f^{*,i}$ $(i = \{s, c\})$ are the extended observability matrix and Toeplitz matrix corresponding to the secondary and primary outputs; superscript $*$ indicates that these matrices are constructed with predictor matrices instead of the original system matrices:

$$\Gamma_f^{*,i} = \begin{bmatrix} \mathbf{C}^i \\ \mathbf{C}^i\mathbf{A}_K \\ \vdots \\ \mathbf{C}^i\mathbf{A}_k^{f-1} \end{bmatrix}$$

$$\mathbf{G}_f^{*,i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}^i\mathbf{B}_K & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}^i\mathbf{A}_K^{f-2}\mathbf{B}_K & \mathbf{C}^i\mathbf{A}_K^{f-3}\mathbf{B}_K & \cdots & \mathbf{C}^i\mathbf{B}_K \end{bmatrix}$$

the future data $\mathbf{y}_f^i(k)$ $(i = \{s, c\})$ and $\mathbf{z}_{f-1}(k)$ are defined as the following:

$$\mathbf{y}_f^i(k) = \begin{bmatrix} \mathbf{y}_k^i \\ \mathbf{y}_{k+1}^i \\ \vdots \\ \mathbf{y}_{k+f-1}^i \end{bmatrix} \quad \mathbf{z}_{f-1}(k) = \begin{bmatrix} \mathbf{z}_k \\ \mathbf{z}_{k+1} \\ \vdots \\ \mathbf{z}_{k+f-2} \end{bmatrix}$$

When p is large, Eqns.(6.5) and (6.6) can be reduced to

$$\mathbf{y}_f^s(k) = \Gamma_f^{*,s}\mathbf{L}_p^*\mathbf{z}_p(k) + \mathbf{G}_f^{*,s}\mathbf{z}_{f-1}(k) + \mathbf{e}_f(k) \tag{6.7}$$

$$\mathbf{y}_f^c(k) = \Gamma_f^{*,c}\mathbf{L}_p^*\mathbf{z}_p(k) + \mathbf{G}_f^{*,c}\mathbf{z}_{f-1}(k) + \mathbf{e}_f^c(k) \tag{6.8}$$

It is worth noting that Eqn.(6.7) or (6.8) serves as the foundation for subspace ID and usually $p \geqslant f > n$ is required to facilitate the estimation of the system matrices [158].

Next the main conclusion on the DPLS soft sensor is stated in the following theorem:

Theorem 1. *For a linear time invariant (LTI) system as shown in Eqn.(6.1), there exists a linear regression, with the primary output as the response and past input and secondary output as the regressor, that can provide unbiased estimate of primary output for both open loop and closed-loop operations.*

Proof. First, note that Eqn.(6.8) not only captures the process dynamics, but also holds for both open loop and closed-loop cases, as it is the extended state-space representation of the LTI system defined by Eqn. (6.1).

Eqn.(6.8) can be reformulated into

$$\mathbf{y}_f^c(k) = \begin{bmatrix} \Gamma_f^{*,c}\mathbf{L}_p^* & \mathbf{G}_f^{*,c} \end{bmatrix} \begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_{f-1}(k) \end{bmatrix} + \mathbf{e}_f^c(k) \tag{6.9}$$

which can serve as a soft sensor model to predict $\mathbf{y}_f^c(k)$ from $\mathbf{z}_p(k)$ and $\mathbf{z}_{f-1}(k)$. Eqn.(6.9) indicates that if and only if the future innovation $\mathbf{e}_f^c(k)$ is uncorrelated to the regressor variable $\begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_{f-1}(k) \end{bmatrix}$, i.e.,

$$E\left\{ \mathbf{e}_f^c(k) \begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_{f-1}(k) \end{bmatrix}^T \right\} = 0 \tag{6.10}$$

an unbiased prediction of $\mathbf{y}_f^c(k)$ can be obtained from the model given in Eqn. (6.9), *i.e.*,

$$\hat{\mathbf{y}}_f^c(k) = \begin{bmatrix} \boldsymbol{\Gamma}_f^{*,c}\mathbf{L}_p^* & \mathbf{G}_f^{*,c} \end{bmatrix} \begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_{f-1}(k) \end{bmatrix} \tag{6.11}$$

Note that condition (6.10) always holds for open loop cases. Because the future innovation is white, without any feedback, the future innovation $\mathbf{e}_f^c$ is independent from both past data $\mathbf{z}_p$ and future data $\mathbf{z}_{f-1}$. However, condition (6.10) does not hold for closed-loop cases when $f > 1$ [160–164]. This is because under feedback control, the future input will be correlated to past innovation, i.e. $E\left\{(\mathbf{e}_k^c)^T \mathbf{u}_{k+i}\right\} \neq 0$, for all positive integer $i \in I_+$. As a result, $E\left\{\mathbf{e}_f^c(k)\mathbf{z}_{f-1}(k)^T\right\} \neq 0$, although $E\left\{\mathbf{e}_f^c(k)\mathbf{z}_p(k)^T\right\} = 0$ as the future innovation is still independent from past data. The detail of the matrix $E\left\{ \mathbf{e}_f^c(k) \begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_{f-1}(k) \end{bmatrix}^T \right\}$ is given in Appendix B, which shows that the lower triangular part of the matrix is non-zero.

When $f = 1$, $\mathbf{z}_{f-1}(k)$ is eliminated and Eqn.(6.9) reduces to

$$\mathbf{y}_k^c = \Gamma_f^{*,c}\mathbf{L}_p^*\mathbf{z}_p(k) + \mathbf{e}_k^c \tag{6.12}$$

As the current innovation $\mathbf{e}_k^c$ is uncorrelated to past data for both open loop and closed-loop, i.e., $E[\mathbf{e}_k^c(\mathbf{z}_p)^T] = 0$, the soft sensor model obtained based on Eqn.(6.12) provides unbiased estimate of $\mathbf{y}_k^c$, which is the one-step-ahead prediction, no matter if the primary variable is used for feedback control or not. It is worth noting that in Eqn.(6.12) only process inputs ($\mathbf{u}$) and secondary output ($\mathbf{y}^s$) are included in the regressor. □

To address the ill-conditioning problem caused by the correlated regressors (both between the same variables at different time instants, and between different variables), partial least squares (PLS) can be applied to estimate $\left(\Gamma_f^{*,c}\mathbf{L}_p^*\right)$ for the soft sensor model, which is the commonly used DPLS soft sensor because of the way $\mathbf{z}_p(k)$ is constructed (see Eqn.(6.4)). The above analysis clearly shows that both the subspace ID and DPLS soft sensor are derived based on the same extended state-space model (*i.e.*, Eqn.(6.6)), while the DPLS soft sensor is a special case with $f = 1$. Therefore, the DPLS soft sensor is theoretically a dynamic estimator that gives unbiased estimate for both open loop and closed-loop data.

Remark. It is worth noting that the purpose of the soft sensor model in this work is to predict the Kappa number, instead of estimating the system matrices A, B and C^s. Therefore, the system does not have to be persistently excited; instead, only the observable part of the process needs to be captured so that under a similar operation condition, the Kappa number can be estimated from the secondary variables accurately. The purpose of the Theorem is to show that by applying DPLS, one is able to construct a soft sensor that can adequately capture the process dynamics under the given operation condition and provide unbiased estimate of the primary variable, provided that enough secondary variables are used in the regression.

Remark. Ku et al [29] extended PCA to DPCA in order to monitor a dynamic process by using the augmented measurements of process variables as the new variables, which is also the basic idea behind DPLS. The fundamental reason that such augmentation can enable static PCA and PLS to capture process dynamics can be explained by the extended state-space model, which is the foundation of subspace identification. This was also pointed out in [29]. However, in [29], the emphasis is on process monitoring, and on providing a heuristic explanation on how monitoring lagged data can remove the time dependence of the self-correlated and cross-correlated dynamic process data, rather than providing a generalized proof/derivation on how process dynamics is captured by a static linear relationship between lagged variables. In this work, one major contribution is to derive an exact static linear regression between the augmented secondary variables and primary variables based on the state-space representation of an LTI system, and to show that such model provides unbiased prediction of the primary variables under both open loop and closed-loop operations.

Remark. During the stage of model building, the measurements of both primary and secondary variables that are collected at the same frequency are needed to estimate the model, i.e., $\Gamma_f^{*,c}\mathbf{L}_p^*$. Note that high frequency measurements of the primary variables is not required. In this work, the secondary variable measurements are down-sampled to obtain the same frequency of both primary and secondary variables for the industrial case study. During model prediction/application phase, the secondary variables are not down-sampled and the Kappa numbers are predicted at the higher frequency as that of the secondary variables. If zero or first order hold were implemented for the Kappa number, only the Kappa numbers that were actually measured should be used for model building.

Remark. In the traditional subspace ID methods such as MOESP [157] and N4SID [165], the innovation form instead of the observation form is used. The extended state-space form using Hankel data matrix is given below:

$$\mathbf{Y}_f = \mathbf{\Gamma}_f\mathbf{X}_k + \mathbf{H}_f\mathbf{U}_f + \mathbf{E}_f \tag{6.13}$$

One of the first steps in both methods is to eliminate the contribution from future input ($\mathbf{U}_f$) to the future output ($\mathbf{Y}_f$). In MOESP, it is done by projecting out the future input through post multiplying Eqn. (6.13) with the orthogonal complement of future input, i.e., $\Pi_{\mathbf{U}_f}^T = \mathbf{I} - \mathbf{U}_f^T(\mathbf{U}_f\mathbf{U}_f^T)^{-1}\mathbf{U}_f$; in N4SID, it uses an oblique projection to remove the contribution from both the future input ($\mathbf{U}_f$) and innovation ($\mathbf{E}_f$), by projecting Eqn. (6.13) onto the past data along the future input. In order to obtain unbiased estimate of the system matrices, both methods require that the future innovation term ($\mathbf{E}_f$) is uncorrelated with future input ($\mathbf{U}_f$), i.e., $\lim_{n\to\infty} \frac{1}{n} \left\{ \mathbf{E}_f \mathbf{U}_f^T \right\} = 0$, which is the fundamental reason that these traditional methods only work for open loop cases, as this condition is true for open loop only. For closed-loop data, due to feedback, the future input will be correlated to past innovation. Similar to what we have shown in Eqn. (B.2), the lower triangular part of the matrix $\lim_{n\to\infty} \frac{1}{n} \left\{ \mathbf{E}_f \mathbf{U}_f^T \right\}$ is non-zero due to feedback control. Note that for subspace ID, in order to estimate system matrices, it is usually required that the future horizon $f > n \geq 1$. Therefore, the traditional subspace ID methods cannot provide unbiased estimates using closed-loop data [163].

Remark. In practice, most processes are time-varying instead of LTI, which is difficult to address under the subspace ID framework. However, this can be addressed by adaptive or recursive update of the soft sensor. In other words, a sequence of LTI systems is used to approximate a time-varying and possibly nonlinear system. The details on how to effectively update the soft sensor are presented in Chapter 7.

6.3 Reduced Order DPLS Soft Sensor

For some chemical processes, the process orders are high due to transport delay, strong process dynamics, and nonlinearity. For example, for pulp digesters, $n = 25$ and $n = 50$ have been reported in [152] and [166] respectively. In addition, in order for the contribution from the initial state to be negligible, the past horizon is usually much larger than the process order. Therefore, a large number of regressor variables (including their lagged values) would be included in the soft sensor if it is constructed as the traditional DPLS soft sensor following Eqn.(6.12). The potential issue with this is that model building and maintenance would be more difficult and the model tends to be slow in response to process changes. In addition, DPLS can only provide one-step-ahead predictions.

To address these issues, in this work, a reduced order DPLS (RO-DPLS) approach is proposed. This approach minimizes the number of regressor variables by taking the process characteristics into account. For digester control problem, one reason for the high system order is the long transport delay. If the delay time can be identified and eliminated by time shifting the measurements, the system order and the past horizon can be effectively reduced. In digester control, because the wood chips flow from the top to the bottom of the digester, the delays associated with the measurements collected at different locations are different, *e.g.*, the delay associated with the upper circulation would be different from the delay associated with the extraction. To illustrate this point, the extended Purdue model [145, 146] is implemented, which simulates the pulping process by approximating the digester with a series of continuous stirred tank reactors (CSTRs). As shown in Figure 6.2, after a disturbance in wood chip composition was introduced at 0 hour, the effect of the disturbance shows up in the Kappa numbers with different delays at different locations indicated by the indices of the CSTRs. Fig. 6.2 indicates that among all the variables included in the past data vector $\mathbf{z}_p$, only the ones corresponding to their associated delays contribute the most to the final Kappa number. Therefore, instead of using all elements in $\mathbf{z}_p(k)$ to predict the pulp Kappa number, based on the principle of parsimony, only those that contribute the most will be selected to build the dynamic sensor. To illustrate this point, by denoting $\mathbf{u}_k = [\ u_1(k) \ \ \cdots \ \ u_m(k)\ ]^T$, and $\mathbf{y}_k^s = [\ y_1(k) \ \ \cdots \ \ y_l(k)\ ]^T$, the past data $\mathbf{z}_p$ is reorganized into the following format and compare it to the regressor with selected variables, denoted as $\mathbf{z}_p^*$.

$$\mathbf{z}_p(k) = \begin{bmatrix} u_1(k-1) \\ \vdots \\ u_1(k-p) \\ u_2(k-1) \\ \vdots \\ u_2(k-p) \\ \vdots \\ u_m(k-1) \\ \vdots \\ u_m(k-p) \\ y_1(k-1) \\ \vdots \\ y_1(k-p) \\ \vdots \\ y_l(k-1) \\ \vdots \\ y_l(k-p) \end{bmatrix} \quad \mathbf{z}_p^*(k) = \begin{bmatrix} u_1(k-d_{u_1}) \\ \vdots \\ u_1(k-d_{u_1}-p') \\ u_2(k-d_{u_2}) \\ \vdots \\ u_2(k-d_{u_2}-p') \\ \vdots \\ u_m(k-d_{u_m}) \\ \vdots \\ u_m(k-d_{u_m}-p') \\ y_1(k-d_{y_1}) \\ \vdots \\ y_1(k-d_{y_1}-p') \\ \vdots \\ y_l(k-d_{y_m}) \\ \vdots \\ y_l(k-d_{y_1}-p') \end{bmatrix}$$

where d_{u_i} denotes the delay difference between the i^{th} input and the product Kappa number, and d_{y_j} denotes the delay difference between the j^{th} secondary output and the product Kappa number. Let $d_{max} = \max\{d_{u_i},\ d_{y_j}\}$, $d_{min} = \min\{d_{u_i},\ d_{y_j}\}$ $(i \in [1,\ \cdots,\ m],\ j \in [1,\ \cdots,\ l])$. It is clear that $p \geq d_{max} + p'$, and in most cases $p \gg p'$. Therefore, by incorporating the process delay information, RO-DPLS effectively reduces the dimension of the regressor. Another big advantage associated with RO-DPLS is that d_{min}-step-ahead prediction can be obtained directly instead of the one-step-ahead prediction by regular DPLS as given in Eqn.(6.12). It is worth noting that multiple-step-ahead prediction depends on process characteristics. Only one-step-ahead prediction is possible if $d_{min} = 1$. However, for most applications in practice that require soft sensor approaches to estimate process quality variables, such as digester and distillation column, $d_{min} \gg 1$. It is worth noting that because the goal of a soft sensor is

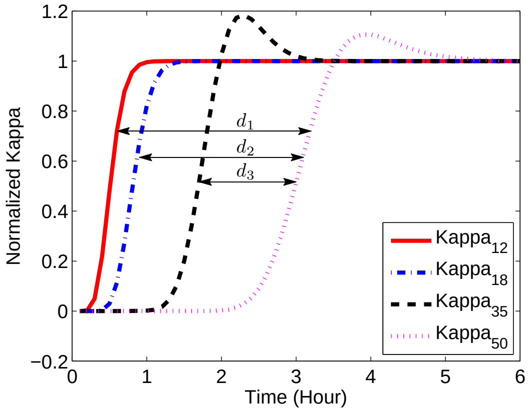


Figure 6.2: Different delay associated with variables at different location. Subscripts in the legend are the indices of the CSTRs used to simulate the digester where 1 and 50 correspond to the top and bottom of the digester respectively. Refer to Section 6.5 for the simulator details.

to obtain an accurate estimation of the primary output instead of system matrices, the past horizon p' associated with each variable can be different, which provides additional flexibility compared to subspace ID methods. In addition, it is possible to reduce the sampling points within the past horizon so as to further reduce the dimension of the regressor.

6.4 Implementation of the RO-DPLS Soft Sensor

The implementation of the RO-DPLS soft sensor is illustrated in Figure 6.3, which consists of five steps: variable selection, variable delay estimation, variable past horizon estimation, regressor and regressand matrices construction and application of PLS to the constructed matrices. As the last two steps are common to all DPLS based soft sensors, below the details on the first three steps are discussed.

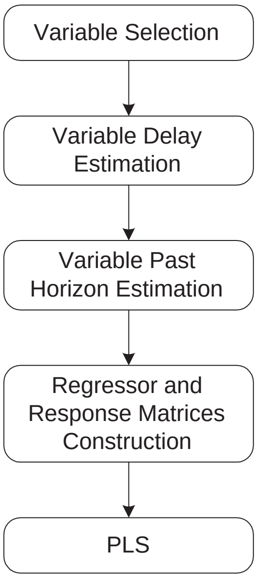

Variable Selection
Variable Delay Estimation
Variable Past Horizon Estimation
Regressor and Response Matrices Construction
PLS

Figure 6.3: Flow diagram of the proposed RO-DPLS approach

6.4.1 Regressor Variable Selection

Similar to DPLS, one potential issue with the RO-DPLS soft sensor approach is the selection of the secondary variables. Inappropriate selection of secondary variables as soft sensor inputs may lead to numerical problems, such as singularity and over-parameterization, or may significantly reduce the estimation accuracy [39, 48]. It is well known that the satisfactory performance of soft sensors is likely to be achieved if only those secondary variables that are most sensitive to the primary variables are employed. This can be done based on process knowledge [167], iterative selection method [168], sensitivity analysis [169–171] and others [39, 152]. A brief review of these methods are given below.

Joseph and Brosilow [168] proposed an iterative method to select the optimal set of secondary variables to be used as soft sensor inputs for a distillation column. The method is based on the addition of secondary variables to the optimal set, one at a time. The procedure is repeated until satisfactory estimation accuracy is obtained, or until all variables have been included. Tolliver and McCume [169] suggested that the optimal temperature locations (secondary measurements) in a distillation column be determined by evaluating the column sensitivity to the material balance. Another soft sensor input variable selection approach is based on the application of singular value decomposition (SVD) to the sensitivity gain matrix [170, 171]. Amirthalingam and Lee suggested that variables selection should be based on the achievable nominal performance [152]. In their approach, a closed-loop performance index is established and the achievable values for all candidate variables are computed and ranked. More recently, Zamprogna et al. proposed a variable selection method based on a PCA sensitivity analysis [39]. All these methods have their pros and cons. In this work, Joseph and Brosilow's approach is adopted due to its simplicity. Because process knowledge helps to remove most of the insignificant variables, the iterative procedure is not time consuming.

6.4.2 Variable Delay Estimation

One key step of the proposed RO-DPLS soft sensor is the estimate of the delays associated with different variables. For the continuous digester, because wood chips flow from top to bottom, the transport delay associated with the measurements collected at different locations of the digester are different. By identifying the delay time associated with each one of the streams in the digester, the system order can be effectively reduced, therefore the past horizon required to predict the Kappa number is reduced.

To estimate the delay time associated with each stream of the digester (*i.e.*, upper heater, lower heater, extraction), the major source of disturbance in the continuous pulping process is considered – wood composition variations. For the simulated case, a step change can be introduced in the wood composition and the delays of the secondary variables for the different streams are estimated using the methods reported in [172, 173] by approximating the responses with systems of second order plus time delay (SOPTD). For industrial digesters, the variable delays can be estimated based on the wood chip flow rate and the physical dimensions of the digesters, *e.g.*, the diameters and heights of different zones and physical locations of different sensors. Similar variable delay estimation strategy has been reported in [174].

6.4.3 Variable Past Horizon Estimation

To extract the dynamic relations between the primary output and the secondary output, one has to determine the necessary past horizon (p for DPLS or p' for RO-DPLS) . For linear systems, this can be determined based on the order of the dynamic system. For nonlinear systems, past horizon could be higher in order to get a better linear approximation of the actual nonlinear relationships [29]. Two different approaches can be adopted to select the past horizon. One approach is based on [29], where PCA is performed on the data matrix $\mathbf{z}_p(k)$ or $\mathbf{z}_p^*(k)$, so that an implicit multivariate autoregressive (AR) (ARX model if the process inputs are included) is extracted directly from the data. The Q statistic obtained from PCA is then the squared prediction error of the ARX model. If enough past horizon, p or p', is included in the data matrix, the Q statistic is statistically independent from one time instant to the next. Another approach is based on [124] where autoregressive models with different p or p' values to the training data are fitted. The past horizon is selected to be the p or p' value that minimizes the small sample AIC criterion [175]. In this work, the second approach is used.

6.5 Application to a Simulated Continuous Digester

In this section, the performance of the proposed RO-DPLS soft sensor approach will be evaluated using a high-yield digester simulated with the extended Purdue model, and compared with the traditional DPLS approach.

6.5.1 Simulator

The extended Purdue model developed in [146] was implemented as the “plant” for simulation purposes. The single-vessel high-yield digester as shown in Fig. 6.1 is approximated by 50 CSTRs in series, which results in 950 nonlinear ordinary differential equations (ODEs). For this case study, the following process variables are employed:

- ***Output*** The primary output is the product Kappa number; the secondary outputs include the effective alkali (EA), dissolved lignin (DL) and free liquor temperature (T) of upper recirculation, lower recirculation and extraction flow. Both primary and secondary outputs are measured every 6 minutes during the simulation. White noises were added to both primary and secondary measurements during all plant simulation with variances listed in Table 6.1. Note that only EA and DL were used as regressor variables in [152], but we also include T because our experience shows that temperature contributions cannot be neglected in the DPLS and RO-DPLS approaches. Also note that besides the variables mentioned above, there are many other variables that are usually measured in practice, including flow rates of different streams, concentrations of other liquid phase compositions, etc.. However, for soft sensor construction in this work, their contributions are negligible.

Table 6.1: Variances of the white measurement noise for the primary and secondary outputs

Output	Variance
EA	0.015
DL	0.015
T	0.05
Kappa #	0.05

- ***Manipulated input*** Although upper heater temperature, lower heater temperature and white liquor flow rate can be adjusted in the real process, the lower heater temperature was chosen as the sole manipulated variable based on the results in [152].

- ***Unmeasured disturbance*** Three different types of unmeasured disturbances are considered in this work, namely entering white liquor EA concentration, upper and

Table 6.2: Selected regressor variables

Variable	Description	Location/Stream	CSTR index
EA_{12}	Free liquor effective alkali concentration	Upper cooking	12
DL_{12}	Free liquor dissolved lignin concentration	Upper cooking	12
T_{12}	Free liquor temperature	Upper cooking	12
T_U	Upper heater exit temperature	Upper heater exit	-
EA_{18}	Free liquor effective alkali concentration	Lower cooking	18
DL_{18}	Free liquor dissolved lignin concentration	Lower cooking	18
T_{18}	Free liquor temperature	Lower cooking	18
T_L	Lower heater exit temperature	Lower heater exit	-
EA_{35}	Free liquor effective alkali concentration	Extraction	35
DL_{35}	Free liquor dissolved lignin concentration	Extraction	35
T_{35}	Free liquor temperature	Extraction	35

lower heater temperatures, and wood compositions. The disturbances can be introduced to the simulator either separately or simultaneously, and both cases are studied in this work.

6.5.2 Soft Sensors Setup

Following the procedures presented in Section 6.4, for both DPLS and RO-DPLS, the same set of variables are selected to build the soft sensors, which are listed in Table 6.2. The RO-DPLS parameters associated with the selected variables, including delay difference from the product Kappa number (d_i) and past horizon (p'_i), are listed in Table 6.3. For DPLS, the past horizon p was chosen to be 34, which is the optimized number based on the AIC criterion. For both DPLS and RO-DPLS, the number of principal components (PC's) are determined by 4-contiguous-block cross-validation. In this work, the same settings are applied to both open loop and closed-loop case studies.

6.5.3 Open Loop Case Studies

Open loop simulations were conducted first. During the simulations, the liquor measurements and Kappa number measurement were collected at the same frequency, *i.e.*, every

Table 6.3: Dynamic soft sensor setup

Variable	p_i'	d_i
EA_{12}	3	17
DL_{12}	3	17
T_{12}	3	17
T_U	3	17
EA_{18}	5	16
DL_{18}	5	16
T_{18}	5	16
T_L	5	16
EA_{35}	6	14
DL_{35}	6	14
T_{35}	5	14

Table 6.4: Variances of the white noise for the unmeasured disturbances

Disturbance	Variance
ρ_{s_1}	0.0003
ρ_{s_2}	0.0007
ρ_{s_3}	0.007
ρ_{s_4}	0.0001
ρ_{s_5}	0.002

6 minutes. To evaluate the performance of RO-DPLS, four different types of disturbances are introduced:

Case 1: Integrated white noise with variance 0.0009 in entering white liquor EA concentration.

Case 2: Integrated white noise with variance 0.005 in upper and lower heater temperatures.

Case 3: Integrated white noises with variances listed in Table 6.4 in the five wood compositions ($\rho_{s_i}, i = 1, 2, ..., 5$) at the inlet.

Case 4: The combination of the above three cases.

Note that the parameters listed in Tables 6.1 and 6.4 are the same as those in [166]. For each case, 80 hours (800 samples) of simulated data were used for training; 700 hours (7000 samples) of the simulated data were used for testing. To assess the performance of the different soft sensors, three statistics are used, namely maximum relative estimated error (ϵ_{max}), mean prediction error (MPE) and mean squared prediction error (MSPE) as defined below.

$$\epsilon_{max} = max \left| \left(\frac{y_i - \hat{y}_i}{y_i} \right) \right| \tag{6.14}$$

$$MPE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i) \tag{6.15}$$

$$MSPE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{6.16}$$

where y_i and $\hat{y}_i$ are the measured and predicted Kappa numbers at the digester exit (*i.e.*, the 50^{th} CSTR) respectively.

The performances of the DPLS and RO-DPLS soft sensors for the four different cases are summarized in Table 6.5. The results indicate that both methods can satisfactorily predict the Kappa numbers, which is also confirmed by Fig. 6.4 where the predictions of the both methods are compared to the measurements for Case 4. In other words, the case studies demonstrate our theoretical result that DPLS can adequately capture process dynamics if enough past horizon is included in the model. It can also be seen that the proposed RO-DPLS soft sensor performed better than DPLS in all cases. RO-DPLS performed particularly better than DPLS in the combined disturbance case (*i.e.*, Case 4), which is probably more realistic in an actual pulp digester. More importantly, RO-DPLS predicts the Kappa number 14 steps ahead, while DPLS only provides one-step-ahead predictions. It is worth noting that if the number of the training samples is increased by 10 times to 800 hours (8000 samples), the performance of DPLS would be similar to that of RO-DPLS. However, the requirement of large training data makes model building and maintenance more difficult. For example, the industrial digester data used in this work were collected over a year. After data preprocessing (i.e., eliminating data during digester down time, sensor failure, and zero/first order hold values), only 1550 samples were left, which is equivalent to 155 hours of simulation. Also, the model built based on large training data would be slow to capture process changes when implemented online with the corresponding recursive algorithm. In addition, it does not change the fact that RO-DPLS prediction is 13 steps ahead of DPLS, which is beneficial for controller design. The aspects of recursive implementations and controller design are revised in Chapter 7.

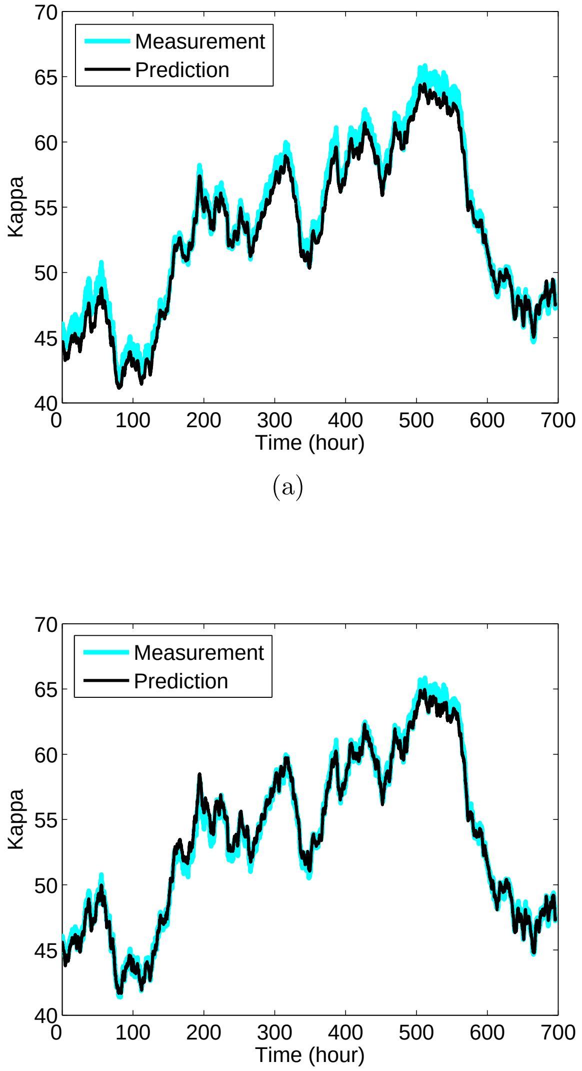
Table 6.5: Comparison of DPLS and RO-DPLS: open loop cases

Case		Case 1	Case 2	Case 3	Case 4
ϵ_{max}	DPLS	0.9315	1.1044	1.5982	2.1831
	RO-DPLS	0.9292	1.0211	1.2271	1.4879
	% reduction	0.3%	7.5%	23.2%	31.8%
MPE	DPLS	-0.1305	-0.0747	-0.3629	-0.6147
	RO-DPLS	-0.1094	0.0092	-0.1045	-.0595
	% reduction	16.2%	87.7%	71.2%	90.3%
MSPE	DPLS	0.07493	0.08208	0.1991	0.6307
	RO-DPLS	0.06892	0.06840	0.08640	0.1924
	% reduction	8.0%	16.7%	56.6%	69.5%

6.5.4 Closed-Loop Case Study

In this subsection, it is demonstrated that both DPLS and the proposed RO-DPLS soft sensors work not only for open loop data but also for closed-loop data as discussed in Section 6.2. In addition, it is shown that RO-DPLS performs better than the traditional DPLS soft sensor.

Previous studies [148, 176] have shown that with real-time feedback of the pulp Kappa number measurements, a well tuned PID controller can provide satisfactory control performance in rejecting wood feed variances. In this closed-loop case study, the lower heater temperature is selected as the manipulated input as in [152], and the rest of the 10 variables in Table 6.2 are secondary outputs. The plant model used to design the controller is a SOPTD model identified from a step response in wood composition using the method reported in [172].



(a)

(b)

Figure 6.4: Soft sensor performance comparison - open loop: (a) DPLS; (b) RO-DPLS

Based on the SOPTD model approximation, a PID controller is designed with the transfer function shown below, and the parameters based on the IMC-based controller tuning relations [167] are summarized in Table 6.6.

$$G\left(s\right)=\frac{K_c e^{-\delta\, s}}{\tau^2 s^2+2\,\zeta\tau+1} \tag{6.17}$$

Table 6.6: Controller tuning parameters

K_c	τ_I	τ_D
-0.0997	22.24	54.00

In this case study, 39 hours (390 samples) of simulated data were used for training; 95 hours (950 samples) of the simulated data were used for testing. White noise with variance 0.0002 was added to the Kappa measurements. Similar to the open loop cases, ϵ_{max}, MPE, and MSPE are used to evaluate the performance of DPLS and RO-DPLS, and the result is summarized in Table 6.7. Figure 6.5 shows the comparison of the predictions of both methods to the measurements. It can be seen from Figure 6.5 that DPLS and RO-DPLS are indeed able to capture process dynamics when the process is under closed-loop control, which is consistent with our theoretical finding in Section 6.2. In addition, both Table 6.7 and Figure 6.5 indicate that RO-DPLS outperforms DPLS. Similar to the open loop cases, if the number of the training samples is increased by 10 fold, the performance of DPLS would be similar to RO-DPLS. Even so, RO-DPLS provides 14-step-ahead prediction horizon that the traditional DPLS cannot offer, which would allow us to tune the controller to be more aggressive for better control. It is worth noting that in this case study, it is assumed that real-time Kappa number is available for feedback control, which is not the case in practice. The integration of the RO-DPLS soft sensor into digester control is currently under investigation.

It is worth noting that in this work the soft sensors were tuned (*e.g.*, variable selection, past horizon, number of PC, etc) to minimize MSPE because it is what matters in practice.

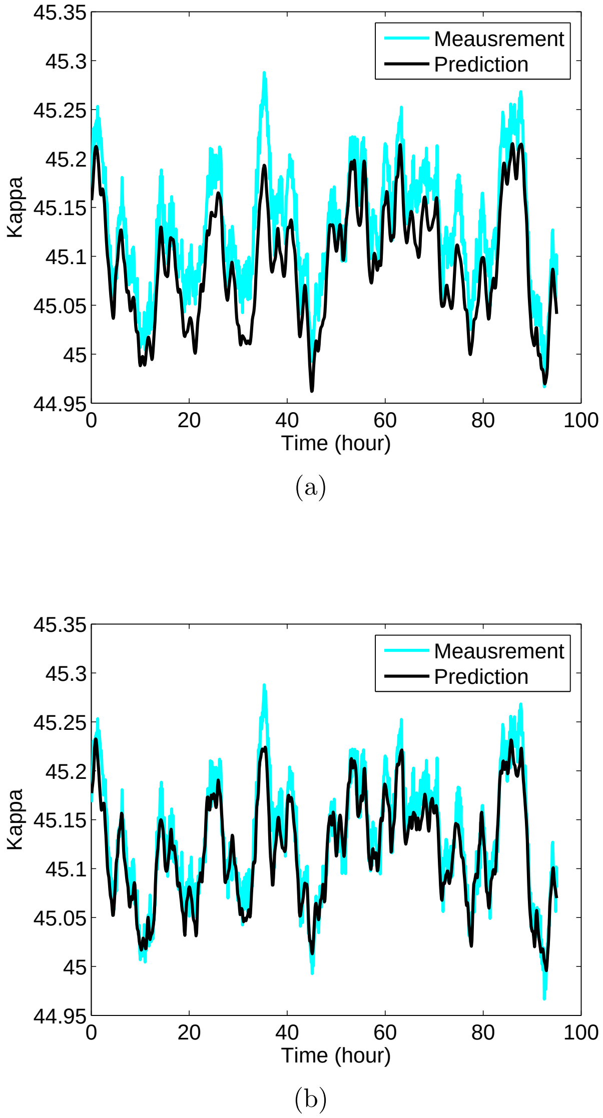
Our experiences have shown that if the soft sensors were tuned to minimize MPE, higher MSPE will be resulted.

Table 6.7: Comparison of DPLS and RO-DPLS: closed-loop case

ϵ_{max}	DPLS	0.1062
	RO-DPLS	0.0832
	% reduction	21.7%
MPE	DPLS	0.0387
	RO-DPLS	0.0225
	% reduction	41.9%
MSPE	DPLS	1.90×10^{-3}
	RO-DPLS	0.523×10^{-3}
	% reduction	72.5%

6.6 Application to an Industrial Kamyr Digester

In this section we apply the RO-DPLS soft sensor to predict the Kappa number of a single-vessel Kamyr digester using data from a paper mill located at Mahrt, Alabama run by MeadWestvaco Corporation. The digester has a DCS and a Duralyzer-NIR digester analyzer system [177, 178] to measure secondary measurements such as EA, DL, total dissolved solids (TS) and active alkali (AA) for the different zones of the digester. The performance of RO-DPLS is compared with that of DPLS. The soft sensors were trained using 1100 samples and tested using 450 samples. Based on Joseph and Brosilow's approach, totally 16 secondary outputs were selected as regressor variables, which are listed in Table 6.8, together with the associated delay and past horizon used in building the RO-DPLS model. As discussed in Sec. 6.3, it is not required that all variables the have the same past horizon. It is just in this case that a past horizon of 3 seems to work for all variables. Fine tuning the past horizon for each variable would improve the performance for some cases, but the improvement is incremental. For DPLS, the past horizon is set to be 22 based on the AIC criterion. The same set of regressor variables are used for both DPLS and RO-DPLS soft sensors. The



(a)

(b)

Figure 6.5: Soft sensor performance comparison - closed-loop: (a) DPLS; (b) RO-DPLS

number of PC's to be retained was determined by 4-block cross-validation. Table 6.9 gives the summary of the performance of the different soft sensors in terms of MPE and MSPE. ϵ_{max} is not included as a performance measure due to the significant measurement noise and possibly outliers in the lab Kappa measurements as indicated by the occasional spikes shown in Figure 6.6, where the soft sensor predictions are compared to the lab measurements. Both Figure 6.6 and Table 6.9 show that RO-DPLS predicts the trend of Kappa number changes better than DPLS. Another key aspect is that the RO-DPLS soft sensor offers

Table 6.8: RO-DPLS model parameters

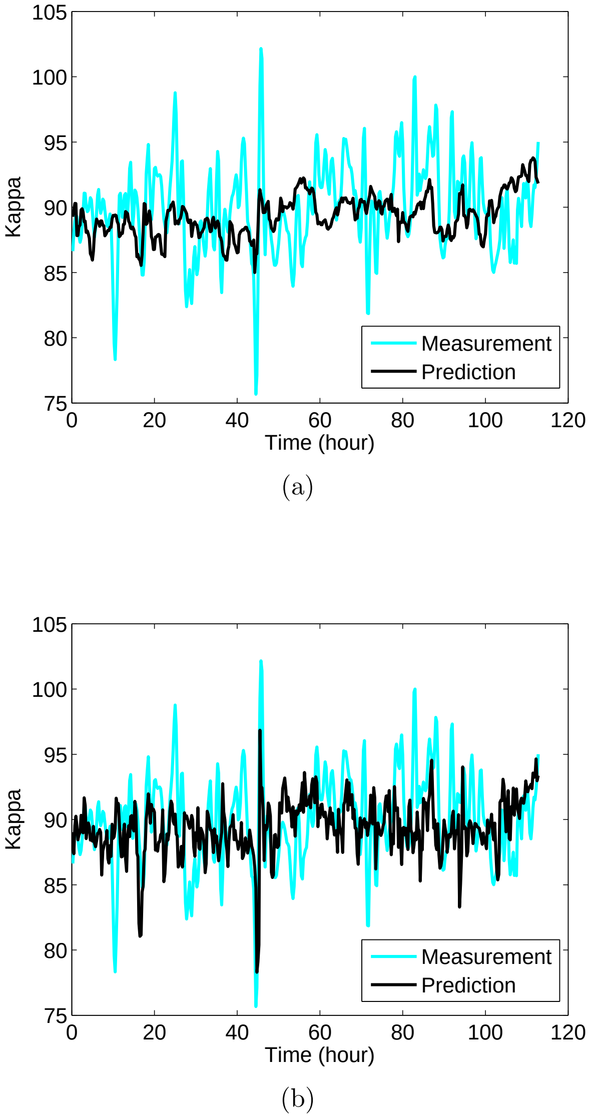
Variable	d_i	p'_i
Top Circulation Temperature	10	3
Combined Heater Temperature	10	3
Upper Cooking [Na_2S]	9	3
Upper Cooking [Na_2CO_3]	9	3
Upper Cooking [Lignin]	9	3
Upper Cooking [Total EA]	9	3
Lower Cooking [Lignin]	8	3
Lower Cooking [Na_2CO_3]	8	3
Lower Cooking [Na_2S]	8	3
Lower Cooking [Total EA]	8	3
Extraction Temperature	6	3
Extraction [Na_2CO_3]	6	3
Extraction [Lignin]	6	3
Extraction [Na_2S]	6	3
Extraction [Total EA]	6	3
Extraction [% Solids]	6	3

Table 6.9: Comparison of soft sensor performance in the industrial case

MPE	DPLS	1.0703
	RO-DPLS	0.6267
	% reduction	41.5%
MSPE	DPLS	14.6199
	RO-DPLS	12.6850
	% reduction	13.2%

a prediction horizon of 1.5 hours, that as mentioned earlier can be combined with more aggressive controller settings to obtain better control of the Kappa number. In addition, the DPLS soft sensor requires a much longer past horizon (22 samples, or 5.5 hr) compared to that of RO-DPLS (3 samples, or 0.75 hr). The possible reason is that a LTI system is used to approximate a nonlinear system with large transport delay. To capture the large delay using an approximated linear system, the resulted linear system usually has a very high system order (n); in addition, the past horizon p is usually required to be much larger than the system order n in order to ignore the effect of initial state. Therefore, the resulted

DPLS soft sensor requires a much longer past horizon. This is consistent with the findings in [152], where the identified linear model for a continuous Kamyr digester using subspace ID has the system order of $n = 40$. Even for the reduced order model in [166], the system order is 25 for a continuous Kamyr digester. All of these advantages offered by the RO-DPLS soft sensor makes the proposed approach a viable option for online implementation.



(a)

(b)

Figure 6.6: Soft sensor performance comparison - industrial case: (a) DPLS; (b) RO-DPLS

6.7 Conclusion and Discussion

In this chapter, a theoretical analysis on the DPLS soft sensor is provided. By deriving the DPLS soft sensor formulation using the subspace ID framework, it is shown that the DPLS soft sensor is theoretically a dynamic estimator that can adequately capture the process dynamics provided that the past horizon is long enough. In addition, by setting the future horizon $f = 1$, the DPLS soft sensor works for both open loop and closed-loop data, which is different from some subspace ID methods.

To reduce the number of lagged variables that are required to be included in the regressor matrix for the traditional DPLS soft sensor, a reduced-order DPLS approach is proposed. By explicitly estimating the transport delay involved in the process, time-shifted variables are used as regressor variables to reduce the system order, therefore reducing the required past horizon for prediction. In this way, not only a reduced-order model that provides superior performance can be obtained, but also multiple-step-ahead prediction instead of one-step-ahead prediction are achieved.

The performance of the proposed RO-DPLS is demonstrated using both a simulated single-vessel continuous digester and an industrial Kamyr digester at MeadWestVaco Corp. The simulation results confirm that the DPLS and RO-DPLS soft sensors can capture process dynamics adequately for both open loop and closed-loop operations. In addition, it was shown that the RO-DPLS soft sensor performs better than the traditional DPLS approach, especially when training data are limited. The industrial case study also confirms that RO-DPLS provides an effective and practical solution to estimate the product Kappa number for continuous digester.

It is worth noting that the application of the proposed RO-DPLS approach is not limited to pulp digesters. Instead, it can be applied whenever DPLS is applicable. RO-DPLS will be advantageous when significant delays exist between the available secondary output and the primary output to be estimated.

Finally, as pointed out earlier, real industrial processes are usually time-varying. For example, when the operation point such as cooking temperature changes, or when the wood type changes, the soft sensor model should change as well. To address this difficulty, different adaptive schemes to update the soft sensor online are investigated in the next chapter.

Chapter 7

Recursive Reduced-Order Dynamic PLS Soft Sensor

7.1 Introduction

In many industrial processes such as distillation columns and pulping digesters, the primary product variables that are required for feedback control are either not measured online or not measured frequently. For example, for the Kamyr pulp digesters, the Kappa number, a key indicator of the extent of reaction, is usually measured in a laboratory once per hour with half an hour to one hour measurement delay. To address this challenge, many data-driven soft sensors have been developed and implemented in process industry. Due to the growing popularity, demonstrated usefulness and huge potential, soft sensor development has attracted many research interests in the past two decades [39, 44, 46, 47, 179, 180]. The most commonly applied modeling techniques in data-driven soft sensors are principal component analysis (PCA), partial least squares (PLS), artificial neural networks (ANN) and support vector machine (SVM) [51, 52, 54, 57, 181–183]. Kadlec et al. [139, 140] provide a comprehensive review on the characteristics of the data generated from the industrial processes and different data-driven soft sensor modeling techniques. A more systematic and thorough discussion on the design procedures of soft sensors and their applications for solving industrial problems can be found in [49].

Among different data-driven soft sensor approaches, the dynamic partial least squares (DPLS) soft sensor approach has been applied to many industrial processes due to its simplicity and available adaptive versions. In the previous chapter [63], a theoretical analysis on the DPLS soft sensor modeling approach is provided, and it was proved that for a linear time invariant (LTI) system, the DPLS soft sensor is a dynamic estimator that can adequately capture process dynamics provided that enough lagged variables are used to build the model [63]. In addition, a reduced-order DPLS (RO-DPLS) soft sensor approach was developed to address some limitations of the traditional DPLS soft sensor when applied to processes with large transport delays. Case studies of both simulated and industrial Kamyr digesters demonstrate the improved performance of RO-DPLS soft sensor compared to the traditional DPLS soft sensor.

In [63], only a static RO-DPLS soft sensor was considered. However, industrial processes often experience time-varying changes, such as variations in process input material, process fouling and catalytic decaying. As a result, the performance of an off-line developed soft sensor often deteriorates. Under these circumstances, it is desirable to update the soft sensor model with the new process data to reflect the process changes. Various adaptation techniques have been published to update DPLS soft sensors online. Helland et al. [137] introduced a recursive PLS (RPLS) regression algorithm which updates the model based on new data without increasing the size of data matrices. Qin [138] further proposed a blockwise RPLS, as well as a moving window and forgetting factor adaptation schemes. Dayal and MacGregor [154] proposed a computationally efficient RPLS by using a kernel algorithm [184] for the PLS computation rather than the NIPALS method [24]. However, existing work on recursive PLS schemes mainly focus on investigating their theoretical properties, as well as designing schemes to reduce computational load. To the best of our knowledge, no result has been published on the practical aspects of these schemes, such as how to choose the updating frequency and how to select the number of PLS components in order to optimize the model performance.

In this chapter, the previously proposed RO-DPLS soft sensor [63] is extended to its online adaptive version in order to track process changes. Several model update schemes as well as different data scaling methods are examined with the emphasis on gaining understanding on the practical aspects of the different recursive schemes. Simulated cases studies and two data sets collected from an industrial Kamyr digester are used to compare different approaches [67–70]. The rest of the chapter is organized as follows: Section 7.2 provides some background information for the RO-DPLS soft sensor. Section 7.3 compares the performance of RO-DPLS with different model updating schemes and data scaling methods reviewed in Chapter 5 using simulated and real industrial data. In Section 7.4, the effectiveness of the recursive RO-DPLS soft sensor is also demonstrated using a digester closed-loop control case study. Finally, conclusions and discussions are presented in Section 7.5.

7.2 Brief Review of the RO-DPLS Soft Sensor

For some chemical processes such as Kraft pulping, the process orders are very high due to transport delay, strong process dynamics and/or nonlinearity. For such processes, if a traditional DPLS approach is used, a large number of regressor variables (including their lagged values) would be included in the soft sensor to capture the process dynamics adequately. The regressor variables in the traditional DPLS approach denoted by $\mathbf{z}_p(k)$ are shown in Equation (7.1).

$$\begin{aligned}\mathbf{z}_p(k) \;=\;& [u_1(k-1),\ \cdots,\ u_1(k-p),\ \cdots,\ u_m(k-1),\ \cdots,\ u_m(k-p),\\ & v_1(k-1),\ \cdots,\ v_1(k-p),\ \cdots,\ v_l(k-1),\ \cdots,\ v_l(k-p)]^T \end{aligned} \tag{7.1}$$

where u_i $(i = 1 \cdots m)$ are process input variables, p is the window of lagged variables which is usually (much) larger than the process order, and v_i $(i = 1 \cdots l)$ are secondary output variables. The potential issue with this traditional DPLS approach is that model building requires large amount of training samples due to large number of regressor variables, which in turn slows down the model's response to process changes. To address this, in the previous chapter, the RO-DPLS approach is proposed which significantly reduces the number of regressor variables by taking the process characteristics into account. For processes with large transport delay, if the delay time can be identified and eliminated by time shifting the measurements, the system order can be effectively reduced, and much fewer lagged variables can be used to build the soft sensor without sacrificing the prediction performance. In the

RO-DPLS soft sensor, instead of including all the variables that are used in the traditional DPLS approach, based on the principle of parsimony, only the variables that contribute the most to the primary output variables are selected to build the dynamic soft sensor. The regressor variables in the RO-DPLS approach, denoted by $\mathbf{z}_p^*(k)$ are shown in Equation (7.2).

$$\begin{aligned} \mathbf{z}_p^*(k) \;=\; & [u_1(k-d_{u_1}),\ \cdots,\ u_1(k-d_{u_1}-p'),\ \cdots, \\ & u_m(k-d_{u_m}),\ \cdots,\ u_m(k-d_{u_m}-p'), \\ & v_1(k-d_{v_1}),\ \cdots,\ v_1(k-d_{v_1}-p'),\ \cdots, \\ & v_l(k-d_{v_m}),\ \cdots,\ v_l(k-d_{v_1}-p')]^T \end{aligned} \tag{7.2}$$

where d_{u_i} denotes the delay difference between the i^{th} input and the primary output, and d_{v_j} denotes the delay difference between the j^{th} secondary output and the primary output. Let $d_{max} = \max\{d_{u_i}\ d_{v_j}\}$, $d_{min} = \min\{d_{u_i},\ d_{v_j}\}, (i \in [1,\ \cdots,\ m],\ j \in [1,\ \cdots,\ l])$. It is clear that $p \geq d_{max} + p'$, and in most cases $p \gg p'$. Therefore, by incorporating the process delay information, RO-DPLS effectively reduces the dimension of the regressor. Another important advantage of RO-DPLS is that d_{min}-step-ahead prediction is obtained compared to one-step-ahead prediction from the traditional DPLS.

7.3 Comparison of the RO-DPLS Soft Sensor Performance and Practical Aspects

In this section, the previously developed RO-DPLS soft sensor is extended to its recursive version by applying the different model updating schemes and data scaling methods discussed in Chapter 5. In this section, the impact of two practical aspects on different recursive PLS schemes is investigated: model update frequency and the way to determine the number of PLS components for prediction. A simulated case study and two industrial case studies of a single vessel high yield Kamyr digester are used to compare the performance of RO-DPLS with different model updating schemes and the corresponding data scaling methods.

Table 7.1: Standard deviations of the white noise for the unmeasured disturbances

Disturbance	σ
ρ_{s_1}	0.1732
ρ_{s_2}	0.2646
ρ_{s_3}	0.8367
ρ_{s_4}	0.1000
ρ_{s_5}	0.4472
EA	0.0300
HS	0.0089

7.3.1 Description of the Data Sets and Case Studies

Simulated data set

The extended Purdue model [146] is implemented to simulate a single vessel high yield Kamyr digester. The digester is approximated by 50 CSTRs in series. Open loop simulation is conducted to generate the data set. During the simulation, the liquor measurements and Kappa number measurements are collected at the same frequency, i.e., one sample every 6 minutes. Integrated white noise disturbances ($d_{k+1} = d_k + w_k$) are introduced to white liquor compositions of effective alkali (EA), active hydrosulfide (HS), and five wood compositions ($\rho_{s_i}, i = 1, 2, ..., 5$) at the inlet. The standard deviations of the white noise (w_k) for the unmeasured disturbances are listed in Table 7.1. In addition, measurement noises are added to both primary and secondary variables during all plant simulation with standard deviations listed in Table 7.2. 300 hours of data were simulated for training and 250 hours for testing. The RO-DPLS soft sensor is set up as in [63]. The secondary variables that are selected as regressor variables are listed in Table 7.3, and the delay (d_i) and past horizon (p_i) of each regressor variable are listed in Table 7.4.

Industrial data sets

Two industrial data sets were collected from a Kamyr digester at a pulp mill located in Mahrt, Alabama run by MeadWestvaco Corp. The digester has a DCS system and a

Table 7.2: Standard deviations of the white measurement noise for the primary and secondary outputs

Output	σ
EA	0.1225
DL	0.1225
T	0.1225
Kappa #	0.2236

Table 7.3: Selected regressor variables for the simulated case study

Variable	Description	Location/Stream	CSTR index
EA_{12}	Free liquor EA concentration	Upper cooking	12
DL_{12}	Free liquor DL concentration	Upper cooking	12
T_{12}	Free liquor temperature	Upper cooking	12
T_U	Upper heater exit temperature	Upper heater exit	-
EA_{18}	Free liquor EA concentration	Lower cooking	18
DL_{18}	Free liquor DL concentration	Lower cooking	18
T_{18}	Free liquor temperature	Lower cooking	18
T_L	Lower heater exit temperature	Lower heater exit	-
EA_{35}	Free liquor EA concentration	Extraction	35
DL_{35}	Free liquor DL concentration	Extraction	35
T_{35}	Free liquor temperature	Extraction	35

Duralyzer-NIR digester analyzer [177, 178] to measure secondary measurements such as effective alkali (EA), total dissolved solids (DS), dissolved lingin (DL), and active alkali (AA) for different streams of the digester. The first data set collected in 2006 contains 1570 samples with Kappa measurements (data set I). The second data set collected in 2010 contains 350 samples (data set II). Several commonly applied basic steps were taken to preprocess the data sets, which include the removal of zero and first order hold readings, as well as the removal of evident outliers (e.g. negative readings for flowrate). To avoid distorting the information contained in the process data, no filtering technique (e.g. low pass filter) was used because the presence of non-white noise could affect the performance of the filtering step.

For industrial case study I, 1100 sample points from data set I are used as the training data

Table 7.4: RO-DPLS model parameters for the simulated case study

Variable	d_i	p_i
EA_{12}	17	3
DL_{12}	17	3
T_{12}	17	3
T_U	17	3
EA_{18}	16	5
DL_{18}	16	5
T_{18}	16	5
T_L	16	5
EA_{35}	14	6
DL_{35}	14	6
T_{35}	14	5

to build the initial model and 420 sample points from data set I as the testing data for online update. For industrial case study II, as in the previous case, 1100 sample points from data set I are used as training data, but 300 samples from the data set II are used as the testing data for online update. Clearly, case study II presents a significantly more challenging problem as training and testing data sets are collected about 4 years apart. The soft sensor setup for the industrial case is the same as that reported in [63]. Totally 10 secondary variables are selected as the regressor variables, which are listed in Table 7.5 together with the associated delay (d_i) and past horizon (p_i).

7.3.2 RO-DPLS Model Update Performance Comparison

To eliminate the effect of outliers on different RO-DPLS model update schemes, multivariate off-line outlier detection is performed to remove outliers from the industrial data sets. In this way, a fair comparison of different model update schemes can be provided. Specifically, the leverage and studentized residual of each sample is computed in order to analyze the independent and dependent variable spaces and detect the outliers. To assess the performance of the RO-DPLS model update schemes, the mean squared error (MSE)

Table 7.5: RO-DPLS model parameters for the industrial case studies

Variable	d_i	p_i
Top Circulation Temperature	3	10
Combined Heater Temperature	10	3
Upper Cooking [Na_2S]	9	3
Upper Cooking [Na_2CO_3]	9	3
Upper Cooking [Lignin]	9	3
Upper Cooking [Total EA]	9	3
Lower Cooking [Lignin]	8	3
Lower Cooking [Na_2CO_3]	8	3
Lower Cooking [Na_2S]	8	3
Lower Cooking [Total EA]	8	3
Extraction Temperature	6	3
Extraction [Na_2CO_3]	6	3
Extraction [Lignin]	6	3
Extraction [Na_2S]	6	3
Extraction [Total EA]	6	3
Extraction [% Solids]	6	3

and mean error (ME) are utilized. Their definitions are given below.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{7.3}$$

$$ME = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) \tag{7.4}$$

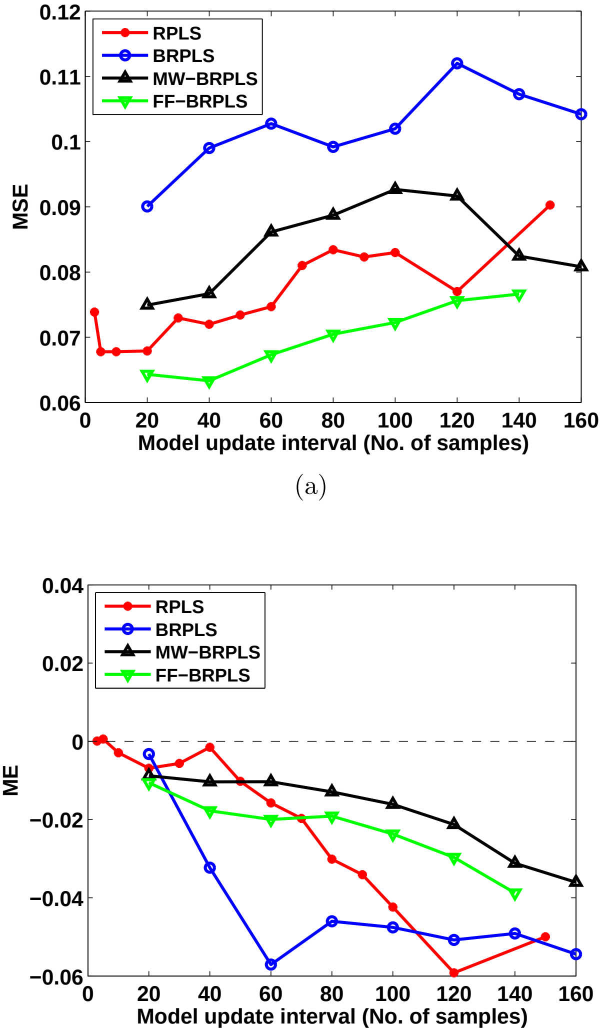
where y_i and $\hat{y}_i$ are the measured and predicted Kappa numbers at the digester exit, respectively. Because each model update scheme has certain tuning parameters associated with it, extensive experiments have been done to test the different settings of each scheme in order to gain better understanding on their properties. In this comparison, the effect of two factors on model performance is examined in detail: model update frequency and how to select the number of PLS components for prediction.

The effect of model update frequency

How often the soft sensor model should be updated in order to achieve the optimal performance depends on the process and the model update scheme. For all model update schemes, a search is done over a wide range of updating intervals by changing the number of new samples used for model update. The comparison results for the three case studies are presented in Figures 7.1 - 7.3 and will be discussed in detail in the following paragraphs. Note that for FF-BRPLS, besides the model update interval, the choice of forgetting factor also plays an important role in determining model prediction performance. To address this, a two-dimensional search is conducted, and in Figures 7.1 - 7.3 only the effect of the model updating frequency under the optimal forgetting factor is presented. Figure 7.4 shows the effect of forgetting factor on FF-BRPLS with a fixed model update interval (80 samples) for all three case studies.

a) *Simulated case study*

For the simulation case study, Figure 7.1 shows that the RO-DPLS soft sensor performs better when the model is updated more frequently, no matter what recursive model update algorithm is used. Specifically, updating the model more frequently not only reduces MSE, but also reduces bias (ME) in the prediction error. This is mainly due to the strong time-varying characteristics of the process caused by the disturbances in the wood composition as well as the white liquor composition. It is worth noting that although Figure 7.1(a) indicates RO-DPLS with FF-BRPLS updating scheme provides the smallest MSE, it does not necessarily mean that it provides the best prediction performance as its estimates are more biased than some other updating schemes such as RPLS when the model is updated relatively frequently. To illustrate this point, the best prediction performances of different updating schemes are compared in Figures 7.5(a) (RPLS and FF-BRPLS) and 7.5(b) (BRPLS and MW-BRPLS). Both figures show that the RO-DPLS soft sensors with different model update schemes provide



(a)

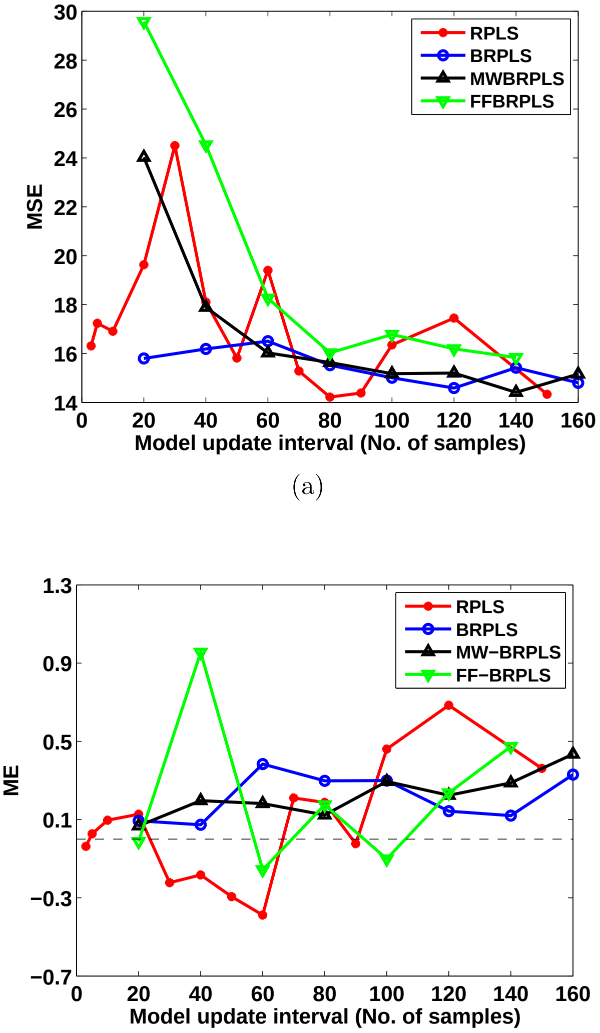
(b)

Figure 7.1: Effect of different update intervals in four model update schemes applied to a simulated case study. (a)MSE ; (b) ME.

good prediction performance and track the process changes well. In addition, Figure 7.5(a) shows that although both FF-BRPLS and RPLS updating schemes provide good prediction, the RPLS scheme tracks the process trend better than FF-BRPLS. Figure 7.5(b) shows that MW-BRPLS tracks the process trend better than BRPLS, but the predictions are noisier.

b) *Industrial case studies*

As shown in Figures 7.2 and 7.3, the results of RO-DPLS with different model updating

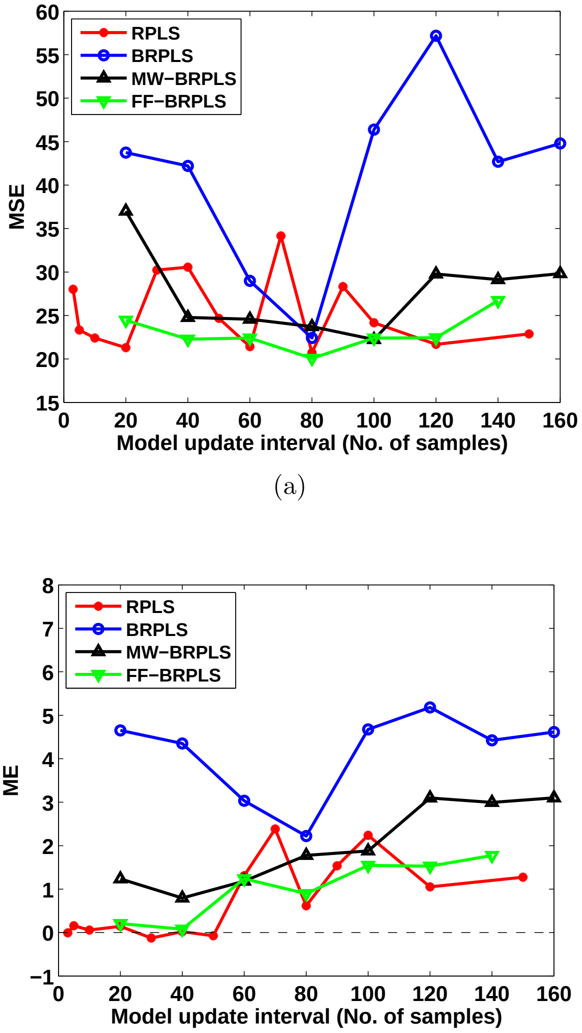


(a)

(b)

Figure 7.2: Effect of different update intervals in four model update schemes applied to industrial case study I. (a) MSE ; (b) ME .

schemes are mixed for the industrial case studies. There is no single method that consistently outperforms others for different model update intervals. This is possibly due to the fact that in the industrial data sets, the Kappa number measurements contain significant amount of noises caused by different technicians who handle the samples during different shifts; in addition, there are limited testing samples in both data sets. Because of the same reasons, MSE may not be an accurate indicator of the soft sensor performance, and ME seems to be a better performance index than MSE. To

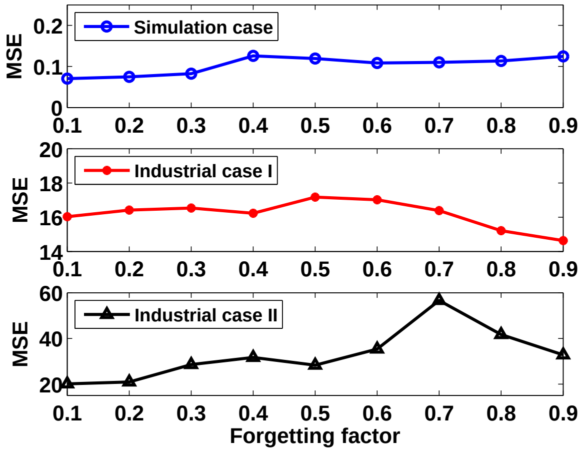


(a)

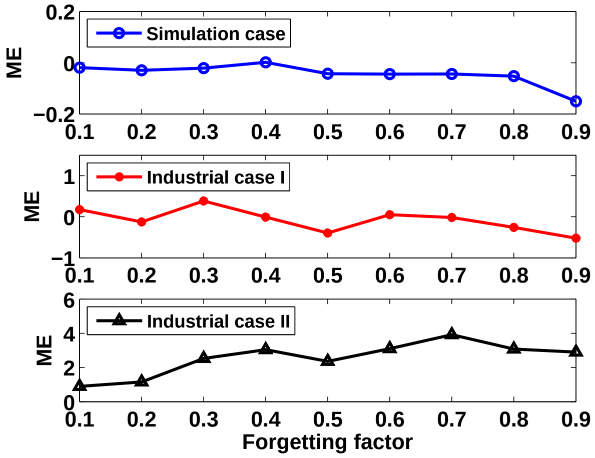
(b)

Figure 7.3: Effect of different update intervals in four model update schemes applied to industrial case study II. (a) MSE ; (b) ME .

illustrate this point, the prediction performance of two selected settings are compared in detail for each method. For industrial case study I, the specific settings and performance indices of each recursive model update scheme are listed in Table 7.6 and their prediction performance are plotted in Figure 7.6. Both Table 7.6 and Figure 7.6 show that when the RO-DPLS soft sensor is updated more frequently, it tracks the process variation better, although it may have a slightly higher MSE. When the soft sensor is updated less frequently, the model prediction tends to ignore the large process



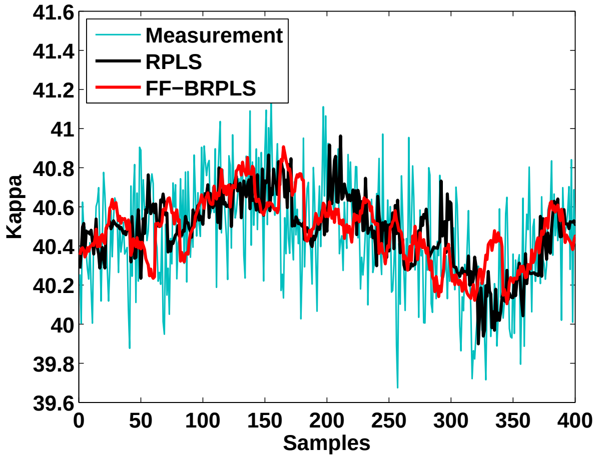
(a)



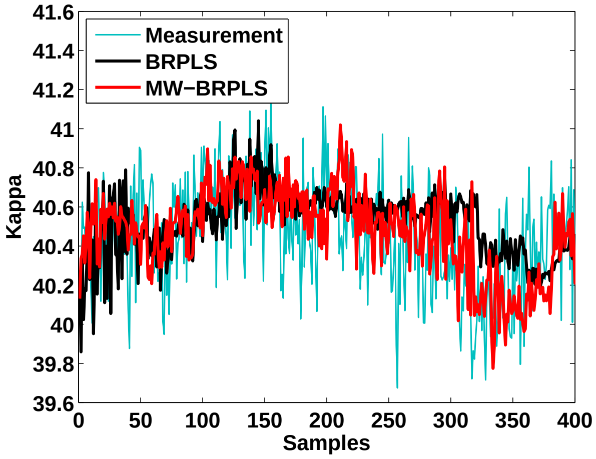
(b)

Figure 7.4: Effect of different forgetting factor on soft sensor performance for a fixed update interval in three case studies. (a) MSE ; (b) ME.

variations and stays closer to the mean of the primary variable. These are true for all updating schemes and are most clearly demonstrated by the RPLS updating scheme (i.e., Figure 7.6(a)). Therefore, ME is a better performance index than MSE in this specific industrial case study. For industrial case study II, Figure 7.7 compares the prediction performances of the RO-DPLS soft sensor with two selected settings for each model updating scheme. The specific settings and corresponding performance indices are listed in Table 7.7. Compared to the industrial case I, the prediction performances



(a)



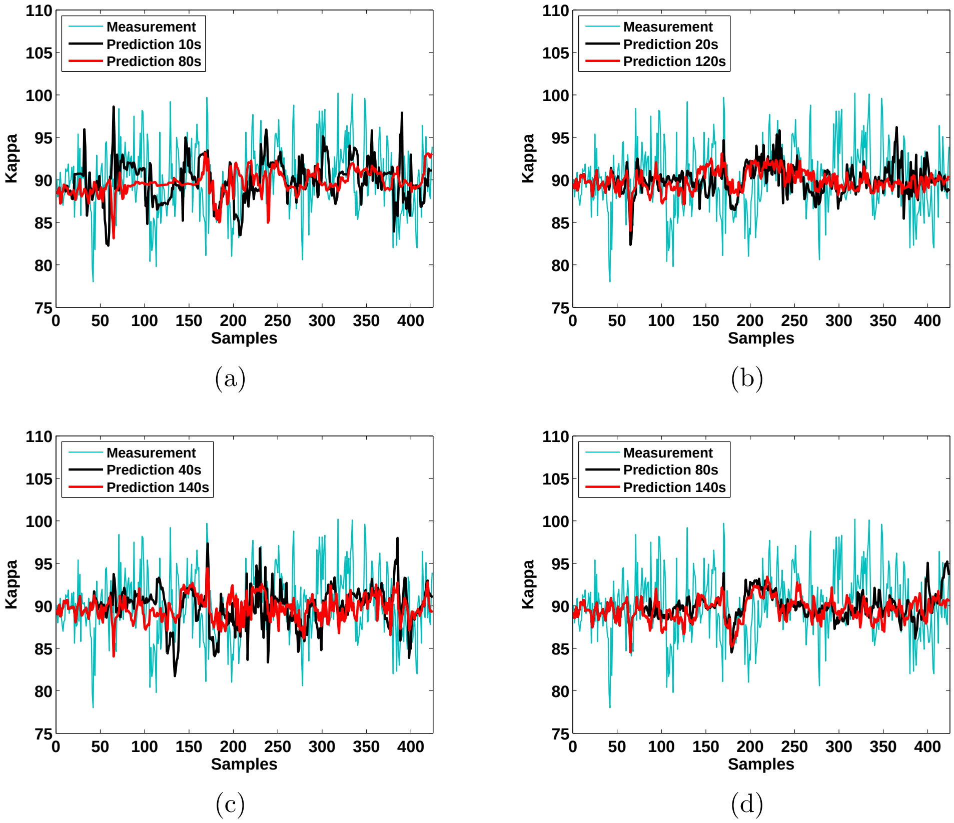
(b)

Figure 7.5: Prediction performance of different model update schemes applied to a simulated case study. RPLS vs. FF-BRPLS (a); BRPLS vs. MW-BRPLS (b).

of the RO-DPLS soft sensor with different updating schemes are noticeably worse. This is expected given that the initial model is the same initial model of the case study I, which is built with the data collected in 2006, while the testing data of the case study II were collected in 2010. In addition, because ME provides better indication of soft sensor performance than MSE in this case study, Table 7.7 shows that RO-DPLS soft sensor with RPLS updating scheme provides the best prediction performance when updated relatively frequently (i.e., every 10 samples or about every 2.5 hours). This is mainly due to the fact that in the industrial case study II, the initial 2006 model does not provide a good description of the 2010 testing data, and the soft sensor performance relies heavily on how the model is updated. Because the information contained in the new measurements is quite different from the initial model, it makes sense to use the data themselves directly (i.e., the RPLS updating scheme) to update the model instead of using the sub-PLS model extracted form the new data (i.e., the rest of the updating schemes), as the sub-model will more or less remove some information from the data.

c) *Discussion*

The extensive experiments performed, show that for the digester process the RO-DPLS



(a)

(b)

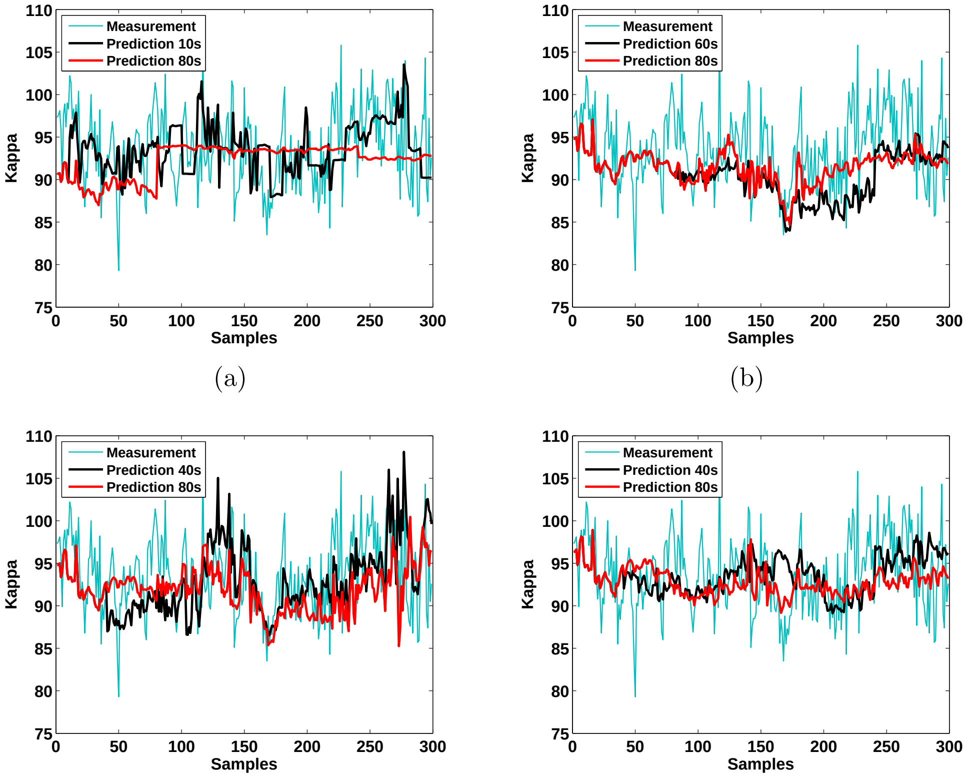
(c)

(d)

Figure 7.6: Prediction performance of selected cases for different model update schemes applied to industrial case study I. RPLS (a); BRPLS (b); MW-BRPLS (c); FF-BRPLS with $\lambda = 0.1$ (d).

Table 7.6: Settings and performance indices of RO-DPLS with different model update schemes for industrial case study I

Updating scheme	Scaling method	Update interval	MSE	ME
RPLS	RSC	10 samples	16.9133	0.0968
		80 samples	14.2192	0.1877
BRPLS	RSC	20 samples	15.7986	0.0934
		120 samples	14.5871	0.1429
MW-BRPLS	MWSC	40 samples	17.8934	0.1968
		140 samples	14.4135	0.2875
FF-BRPLS	FFSC	80 samples	16.0281	0.1753
		140 samples	15.8370	0.4732



(a)

(b)

(c)

(d)

Figure 7.7: Prediction performance of selected cases for different model update schemes applied to industrial case study II. RPLS (a), BRPLS (b), MW-BRPLS (c), and FF-BRPLS with $\lambda = 0.1$ (d).

soft sensor with regular RPLS model updating scheme usually provides the best prediction performance when the model is updated frequently (e.g., with 10 sample interval),

Table 7.7: Settings and performance indices of RO-DPLS with different model update schemes for industrial case study II

Updating scheme	Scaling method	Update interval	MSE	ME
RPLS	RSC	10 samples	22.4115	0.0592
		80 samples	20.7392	0.6181
BRPLS	RSC	60 samples	28.9921	3.0328
		80 samples	22.4171	2.2200
MW-BRPLS	MWSC	40 samples	24.7838	0.7956
		80 samples	23.7232	1.7756
FF-BRPLS	FFSC	40 samples	22.2816	0.0787
		80 samples	20.0813	0.8975

which is mainly due to the volatile nature of the process. In addition, generally speaking for processes that experience considerable variations, BRPLS model updating scheme may not be a good choice. The benefits associated with BRPLS, i.e., computational and storage efficiency, become more clear when the block size is large. However, large block size usually results in sluggish tracking of process changes and may deteriorate soft sensor performance for the processes that show significant time-varying behavior. A question may arise that if it has been proven that both regular RPLS and BRPLS are equivalent to the regular PLS, why their performances differ from each other noticeably in these case studies. The answer is that the condition for the equivalency to hold, i.e., the PLS model for the historical data and the PLS sub-model for the new data have to capture the variability in $\mathbf{X}$ and $\mathbf{X_1}$ data blocks completely, is generally not satisfied, especially for industrial data. To elaborate this point, the data pairs used to build the model of the three algorithms are listed below.

- Regular PLS: $\left\{ \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{Y} \\ \mathbf{Y}_1 \end{bmatrix} \right\}$
- RPLS: $\left\{ \begin{bmatrix} \mathbf{P}^T \\ \mathbf{X}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{B}\mathbf{Q}^T \\ \mathbf{Y}_1 \end{bmatrix} \right\}$

- BRPLS: $\left\{ \begin{bmatrix} \mathbf{P}^T \\ \mathbf{P}_1^T \end{bmatrix}, \begin{bmatrix} \mathbf{BQ}^T \\ \mathbf{B}_1\mathbf{Q}_1{}^T \end{bmatrix} \right\}$

One way to interpret the RPLS algorithm is that $\mathbf{P}^T$ and $\mathbf{BQ}^T$ can be viewed as the representative samples for the historical data $\mathbf{X}$ and $\mathbf{Y}$, as they are the set of orthogonal bases that span the principal subspace of data block $\mathbf{X}$ and $\mathbf{Y}$, respectively. Similarly, for BRPLS, $\mathbf{P}_1^T$ and $\mathbf{B}_1\mathbf{Q}_1^T$ can be viewed as the representative samples for the new data block. When RPLS and BRPLS are implemented, the number of PLS components is often determined through cross-validation, and is usually much smaller than the number of samples in the data sets. Therefore, part of the information contained in the original data will be lost when $\mathbf{P}^T$ and $\mathbf{BQ}^T$ (or $\mathbf{P}_1^T$ and $\mathbf{B}_1\mathbf{Q}_1^T$) are used to replace the data blocks. By choosing the size of the new data block carefully, the PLS sub-model (i.e., $\mathbf{P}_1^T$ and $\mathbf{B}_1\mathbf{Q}_1^T$) can adequately capture the variability in the new data. However, this may be difficult to achieve when there are sudden changes in the process, as shown in the industrial case study II. When the size of the new data block is too small, the PLS sub-model may not adequately capture the variability of the new process condition, as the variability in the residual space of the PLS sub-model is completely discarded. On the other hand, if the size of the new data block is too big, the delay associated with data accumulation may deteriorate the soft sensor performance. Therefore, for the cases where the process is not sampled frequently, RPLS is a more reliable choice compared to BRPLS.

d) *A modified data scaling approach*

It has been recognized that data scaling affects the soft sensor performance, especially when there exists operation point changes. One disadvantage associated with the commonly used data scaling method for RPLS, i.e., Equations (5.13) and (5.14), is that when n (i.e., the number of samples) is large, the mean and the variance do not change much, no matter how the new data differs from the old data, as the weighting assigned to the new data becomes negligible for large n. To address this limitation, a modified data scaling method is proposed by replacing n with N, where N is a fixed integer. In this way fixed weightings are assigned to the previous data and the new data.

The effect of the number of PLS components retained for prediction

For model prediction, Qin [138] recommended that the number of components to be determined through cross-validation for various recursive PLS algorithms. However, in the same work, it was also pointed out that "An interesting issue that deserves further study is how to perform cross-validation online to update the number of components needed for prediction". In this subsection, the performance of the RO-DPLS soft sensor with different ways of selecting the number of components for prediction is compared and our recommendations are provided. The performances of the RO-DPLS soft sensor with five different component selection methods are compared using the simulated data.

a) Number of components is determined through leave-one-out cross-validation with maximum allowed number of components as 10.

b) Number of components is determined through block cross-validation, with maximum allowed number of components as 10, i.e., all training data are divided into four blocks and three of them are used to build the model while the fourth one is used for cross-validation.

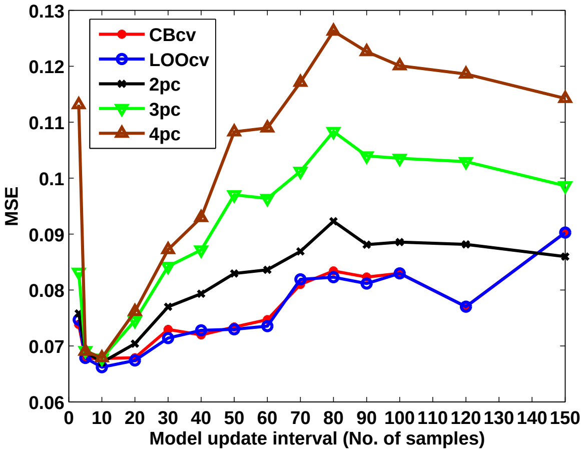
c) Number of components for prediction is fixed to two.

d) Number of components for prediction is fixed to three.

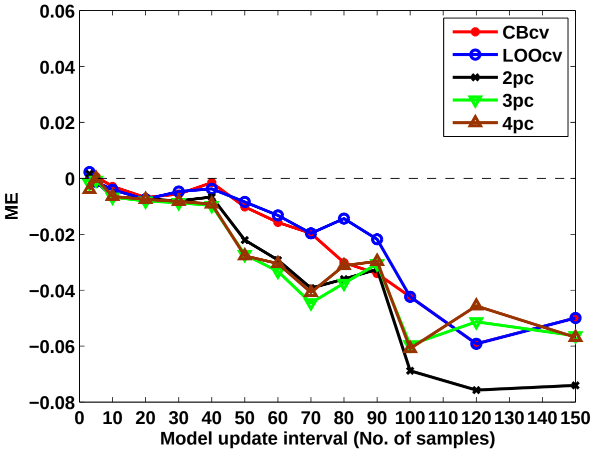
e) Number of components for prediction is fixed to four.

In this comparison, only the results on the RO-DPLS with regular RPLS update scheme are presented. The results on other model update schemes are similar to those on RPLS.

Figure 7.8 shows the MSE and ME for the five cases over a wide range of model updating intervals. Figure 7.8 indicates that the two cross-validation approaches perform better than the approaches with fixed number of components, especially when the soft sensor model is not updated frequently. However, around the optimal updating intervals, i.e., model update every 10 samples, there is not much difference among the different component selection approaches in terms of MSE and ME. To gain a better understanding on how cross-validation



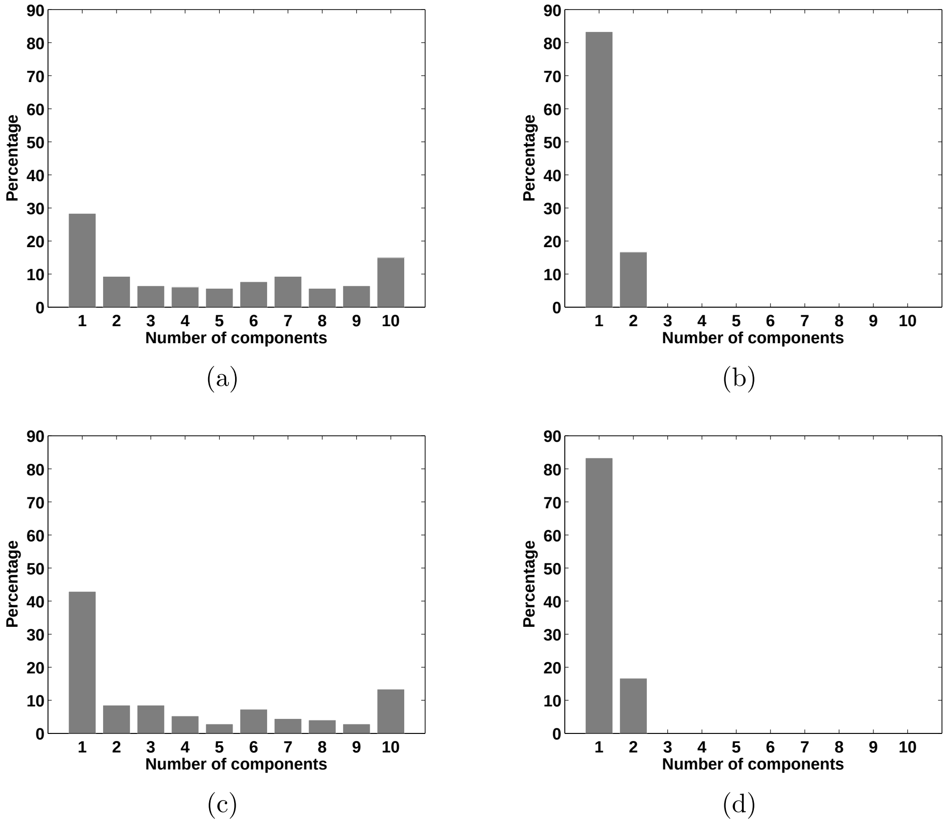
(a)



(b)

Figure 7.8: Simulated case study, MSE (a) and ME (b) of RO-DPLS prediction with fixed or cross-validation selected components for a range of updating intervals.

approaches perform, the number of components determined through cross-validation for each



(a)

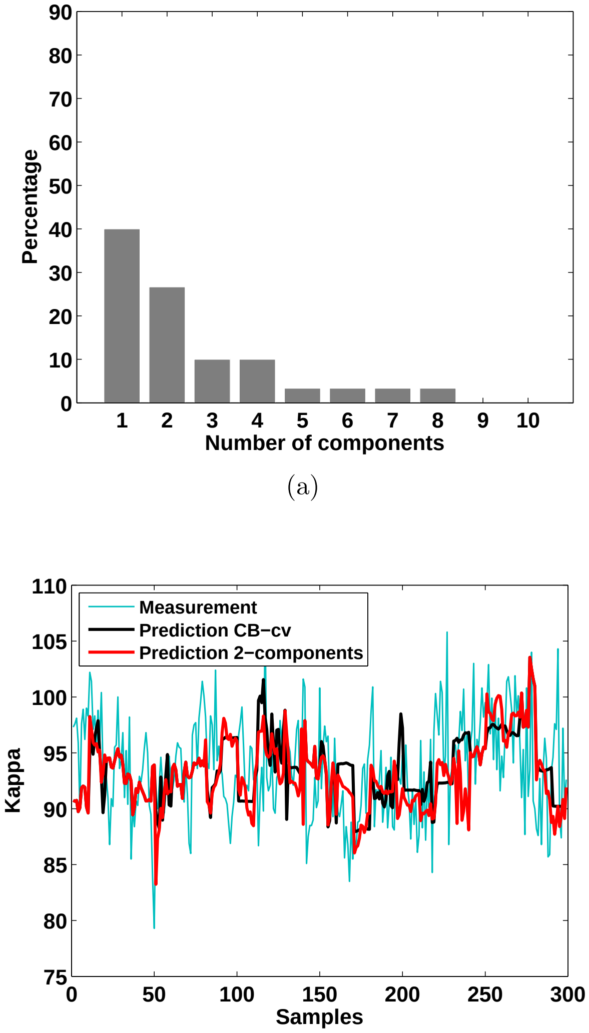
(b)

(c)

(d)

Figure 7.9: Histogram of the number of components determined through cross-validation in the simulated case study for two different cross-validation methods. (a) four contiguous blocks cross-validation with model updated every 10 samples; (b) four contiguous blocks cross-validation with model updated every 100 samples; (c) leave-one-out cross-validation with model updated every 10 samples; (d) leave-one-out cross-validation with model updated every 100 samples.

update is monitored over the entire testing period. Two representative cases are examined, model update every 10 samples and model update every 100 samples. Figure 7.9 shows the histogram of the number of components determined through cross-validation approaches. From Figure 7.9, it is observed that when the model is updated every 10 samples, the number of components for prediction determined through two cross-validation approaches have a wide distribution from 1 to 10. However, when 100 samples are used for update, the selected number of components becomes either 1 or 2. One possible reason for such difference is the large variation presented in the process. When fewer samples are used to update the model, the disturbances and noises contained in the new measurements have a bigger impact on cross-validation. Consequently, the number of components determined through cross-validation shows a wide distribution. Because the dominant process variability is not likely to change so dramatically with in a short period of time, the great variability in the number of components for prediction determined through cross-validation many not represent the process correctly. Therefore, when the soft sensor is updated every 10 samples, using the approach of fix number of components is preferred. On the other hand, when more samples (e.g., 100 samples) are used to update the model, the process may experience a sequence of changes during the update interval. Consequently, using more components will provide better description for one segment of the data, but it may yield poorer prediction for another segment of the data, as the process may have already changed. In contrast, if fewer components are selected, the model will only capture the most dominant directions of process variation which are not likely to change over a period of time. This is why only one or two components are selected for optimal prediction when 100 samples are used to update the model. For the simulated case study and industrial case study I, when the model is updated every 10 samples, there is not much difference in the model prediction performance between the cross-validation approaches and the fixed number of components approach. However, for the industrial case study II, even though the MSE for both approaches are similar, their detailed prediction performances look different. In Figure 7.10(a), the histogram of the number of components determined through cross-validation is shown and Figure 7.10(b) compares the detailed prediction performance of the fix number of components (i.e., 2 components) and cross-validation approach. Figure 7.10(a) shows that for 40% of the testing period, only one component is selected, this is due to the extremely volatile nature of this industrial process. Figure 7.10(b) indicates that due to the limited amount of process variability captured by one component, several segments of the predictions are nearly constant. In contrast, when the number of components is fixed at 2, there is no segment with nearly constant predictions. This example further supports the choice of the fixed number of components approach for model prediction for the processes of volatile nature which require a frequent model update.



(a)

(b)

Figure 7.10: Histogram of the number of components determined through cross-validation (a) and comparison of prediction performance for fixed number of components vs. cross-validation approach (b) for industrial case study II.

7.4 Application of the Recursive RO-DPLS Soft Sensor to Digester Control

In this section, the recursive version of the RO-DPLS soft sensor with RPLS updating scheme is applied to control the simulated Kamyr digester. Two control scenarios are compared: PID control with Kappa number measurement feedback (CL-1), and PID control with the soft sensor predicted Kappa number feedback (CL-2). In this case study, a challenging control problem is considered: rejecting the disturbance of wood type change. It is worth noting that wood type change (from softwood to hardwood and vice versa) is a major disturbance in pulping process, and usually results in off-spec product during the transition [185].

7.4.1 Simulation Setup

The Purdue model introduced in section 7.3.1 is implemented to simulate the Kamyr digester. The simulation starts with softwood as feeding material. Ten hours after the process reaches steady-state, the feed is switched from softwood to hardwood, i.e., a step disturbance. The corresponding reaction kinetics parameters and stoichiometric coefficients for the different types of wood can be found in [146]. The parameters of the integrated white noise are the same as those listed in Table 6.4. The secondary variables are sampled every 6 minutes, and the primary variable (i.e., the Kappa number) is sampled every 30 minutes plus 30 minutes measurement delay. For soft sensor update, the RO-DPLS with regular RPLS update scheme is implemented and the soft sensor model update interval is 10 samples of the Kappa measurement.

7.4.2 Controller Design

Similar to the work presented in [176], the transfer function model for the Kamyr digester is identified through step response. The transfer function uses a second order plus time delay (SOPTD) to approximate the dynamics between the input (lower heater temperature) and the output (the Kappa number); i.e.,

$$G(s) = \frac{K_p e^{-\theta_p s}}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{7.5}$$

Table 7.8: PID controller settings

Case	Feedback scheme	K_c	τ_I(min)	τ_D(min)
CL-1	measurement feedback	-0.0796	54.21	22.27
CL-2	soft sensor prediction feedback	-0.1114	54.21	22.27

with $K_p = -2.4$, $\tau = 34.7$ min, and $\zeta = 0.78$. Here $\theta_p = 112.3$ min represents the transport delay associated with the pulping process. When the measurements are fed back to control the process, measurement delay has to be considered, and the overall transfer function becomes

$$G(s) = \frac{K_p e^{-(\theta_p+\theta_m)s}}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{7.6}$$

where $\theta_m = 30$ minutes is the measurement delay associated with the Kappa number. Based on the identified SOPTD model, the PID controller settings are determined following the IMC tuning relationships [167]. To make the comparison consistent, the same closed-loop time constant is used to tune the two controllers, where τ_c is set to be $\theta_p + \theta_m$, which is a relatively conservative setting. For Kappa number measurement feedback, the overall delay time is $\theta = \theta_p + \theta_m = 142.3$ minutes, and the feedback frequency is every 30 minutes. For soft sensor estimated Kappa number feedback, because the soft sensor predicts the future Kappa number with future horizon ($\theta_f = 42$ minutes), the overall delay $\theta = \theta_p - \theta_f = 70.3$ minutes. In addition, because the soft sensor provides the Kappa number every 6 minutes, the feedback frequency is every 6 minutes. The same tuning rule is applied to both control scenarios. The corresponding PID controller tuning parameters are listed in Table 7.8.

7.4.3 Comparison of Closed-Loop Performance

In this section, the performance of the recursive RO-DPLS soft sensor is examined, then the closed-loop performances for the two PID controllers are compared. Figure 7.11 shows the Kappa number measurement vs. soft sensor prediction in the open loop operation after a wood type change disturbance is introduced. Figure 7.11 indicates that with recursive update, the RO-DPLS soft sensor can track the major process change very well.

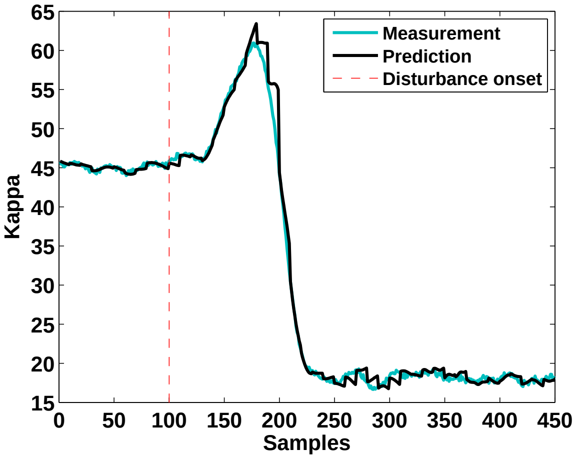


Figure 7.11: Open loop response to a wood type disturbance and soft sensor prediction

Figure 7.12 shows the closed-loop performance for the two control scenarios. By comparing the two closed-loop responses, it is clear that feeding back the predicted future Kappa number can significantly improve the closed-loop performance by considerably reducing settling time. This improvement is mainly achieved by the significantly reduced overall process delay (from $\theta_p + \theta_m$ to $\theta_p - \theta_f$), which allows a much more aggressive controller tuning. In addition, the improvement is due to the more frequent feedback which is also enabled by the soft sensor. It is worth noting that the controller performances depicted in Figure 7.12 present a fair comparison but are not optimized, which can be improved through optimizing control tuning parameters. However, the conclusion that the closed-loop control with soft sensor prediction feedback greatly outperforms the closed-loop control with measurement feedback will not change under any fair comparison.

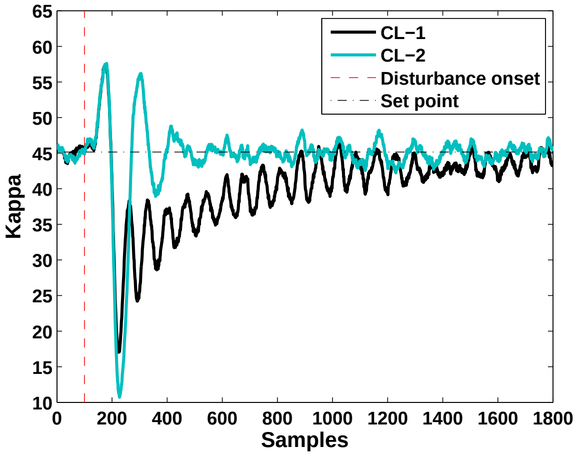


Figure 7.12: Comparison of closed-loop response for CL-1 (Kappa measurement feedback) and CL-2 (soft sensor prediction feedback)

7.5 Conclusion and Discussion

In this chapter, the previously developed RO-DPLS soft sensor is extended to its online adaptive version. Four different model update schemes are examined, namely RPLS, BR-PLS, MW-BRPLS, and FF-BRPLS, together with their corresponding data scaling methods. Extensive experiments were performed with one simulated case study and two industrial case studies to investigate the properties of different model update schemes.

Results show that for the processes that experience constant changes and various disturbances such as the digester process, recursive RO-DPLS with RPLS model update scheme generally performs the best with frequent update. This is expected because for processes that are constantly changing, frequent update provides better tracking of the process changes. Also, for frequent model update, it is preferred to use the new process data directly to update the soft sensor model (i.e., the RPLS scheme), instead of using the sub-PLS model extracted from the new data as in the BRPLS scheme. This is because the sub-PLS model built based on the new process data with limited sample size may lose important process information if it is not in the principal subspace defined by the directions of dominant process variation during that short period. In addition, although the cross-validation method is usually recommended to determine the number of components for model prediction, it is found that when the soft sensor model is updated frequently, it would be better to fix the number of components for prediction. This is because when the number of samples for model update is small, the results from the cross-validation method may be overly sensitive to process disturbance and results in dramatic changes in the number of selected components, which may negatively affect prediction performance.

The effectiveness of the recursive RO-DPLS soft sensor with RPLS update scheme is demonstrated through a digester closed-loop control case study where a PID controller is used to reject a major process disturbance, wood type change, and other common process disturbances. By comparing the closed-loop performance between measurement feedback and soft sensor prediction feedback, it is shown that the closed-loop control performance can be significantly improved if the soft sensor prediction is fed back to the PID controller. The improvement is mainly due to the reduced overall process delay as well as more frequent feedback enabled by the reduced order model and the high frequency Kappa number predictions.

Finally, it is worth noting that outlier detection plays a critical role in PLS-based soft sensor, especially for recursive soft sensors that are updated online. In this chapter, the industrial data are pre-processed to remove all outliers before applying soft sensors scaling, as the focus is to investigate the different recursive update schemes and data scaling methods. However, for industrial application, online outlier detection is an imperative component for recursive data-driven soft sensor development and will be addressed in the next chapter.

Chapter 8

Bayesian Supervision of Outliers for a Recursive Soft Sensor Update

8.1 Introduction

In many industrial processes such as distillation columns and pulping digesters, the primary product variables that are required for feedback control are either not measured online or not measured frequently. To address this challenge, many data-driven soft sensors have been developed and implemented in process industry (see comprehensive reviews in [140], [49] and references cited therein). Various adaptation techniques have been published to update data-driven soft sensors online, and Kadlec et al. [139] provide a comprehensive review on the adaptation mechanisms for data-driven soft sensors. In Chapter 6, a reduced-order dynamic PLS (RO-DPLS) soft sensor was developed to address some limitations of the traditional DPLS soft sensor when applied to processes with large transport delays [63]. By taking the process characteristics into account, RO-DPLS soft sensor can significantly reduce the number of regressor variables and improve prediction performance. More recently, the RO-DPLS soft sensor was extended to its online adaptation version in order to track process changes [67], as described in Chapter 7. Since the focus in [67] was to investigate the properties of different recursive updating schemes and data scaling methods, the industrial datasets were preprocessed to remove all outliers before subjecting them to different experiments. However, it should be noted that the PLS algorithms are sensitive to outliers in the dataset [186]. Therefore, outlier detection and handling plays a critical role in the development of the PLS based soft sensors, and there exist extensive studies on outlier detection for off-line model building [187]. Typical approaches are based on statistics of historical data such as the 3σ outlier detection method [188] and the more robust Hampel identifier method [189]. In addition, multivariate outlier detection methods have also been proposed, such as the PCA-based Jolliffe parameter [190]. Although many methods have been published, outlier detection remains a challenging problem. Therefore, it is often recommended to accompany any outlier detection method with a graphical inspection of the residual space and model parameters to eliminate any possible outlier masking effect (i.e., outliers are classified as consistent samples) and outlier swamping effect (i.e., consistent samples are classified as outliers) [191]. For online adaptation of soft sensor models, if erroneous readings are used to update the soft sensor model, future predictions from the updated model may deteriorate significantly. In addition, online outlier detection is even more challenging since while outliers could be erroneous readings, they could also be normal samples of new process states. It is worth mentioning that an alternative approach to reduce the effect of outliers on soft sensor performance is to use the robust versions of PLS algorithms [186, 192–195]. However, the robust versions of PLS algorithms usually involve much more complicated computation, and may have various limitations, such as applicable to one-dimensional response variable, not resistant to leverage point, and not applicable to high dimensional regressors. More importantly, unlike the conventional PLS algorithm which has various recursive versions available for online updating, currently there are no recursive versions available for the robust PLS algorithms. Therefore, they are not desirable for online model updating, and in this work the conventional PLS algorithm is considered only. In this chapter, multivariate monitoring methods are developed for both off-line and online outlier detection. In addition, to differentiate the samples that represent a process change from those of erroneous readings, a Bayesian supervisory approach to further analyze and classify the detected outliers is proposed [71, 72].

8.2 Outlier Detection Methods

In this section, the proposed outlier detection methods for off-line and online detection are described in detail.

8.2.1 Off-line Outlier Detection for Initial Model Building

A combination of automatic outlier detection, via leverage and y-studentized residuals, together with a visual inspection of the behavior of the different variables (e.g. zero-order hold and evident outliers) is proposed to detect outliers off-line. Leverage of an observation is a concept developed in the ordinary regression theory [196], which defines the influence that a given sample will have on a model and is related to the Hotelling's T^2 statistic [191]. The leverage in terms of the $\mathbf{T}$ scores [197] is computed as follows:

$$h_i = \frac{1}{n} + \sum_{j=1}^{a} \frac{\mathbf{t_{i,j}^2}}{\mathbf{t_j^T t_j}} \tag{8.1}$$

where n is the total number of samples, $\mathbf{t_{i,a}}$ is the vector of scores for object i, and a is the number of principal components retained. A sample is classified as an outlier if its leverage $h_i > \beta(1+a)/n$, where β is a constant (usually 2 or 3). In this work, $\beta = 3$ is chosen as this setting provides a balanced performance between desired sensitivity and specificity. For a given sample, the studentized residual is an indication of the lack of fit of the y-value, which is defined as the following [197].

$$r_i = \frac{f_i}{s(1-h_i)^{1/2}} \tag{8.2}$$

where $f_i = y_i - \hat{y}_i$ is the residual of the dependent variable, $\hat{y}_i$ is the i^{th} prediction of the dependent variable provided by the soft sensor model, and $\frac{1}{n-a-1}\sum_{i=1}^{n} f_i^2$ is the sample estimate variance of the residual. To detect outliers, the studentized residuals are usually compared to a normal distribution N(0,1) [191]. In this work, if $|f_i| > 3$, the sample is classified as an outlier.

8.2.2 Online Outlier Detection for Recursive Model Update

For online outlier detection, the SPE$_x$ and SPE$_y$ (squared prediction error for $\mathbf{X}$ and $\mathbf{Y}$) indices [6, 8] are used to monitor the independent variable and dependent variable space, respectively. It is worth noting that Hotelling's T^2 index is not used to detect outliers, although it is commonly used in conjunction with SPE index for fault detection. Our choice of SPE over T^2 is mainly due to their different roles in process monitoring. It has been suggested that for the samples whose T^2 indices exceed the threshold but not the SPE indices, in many cases they correspond to process operation changes instead of outliers [8]. Our own experiences also confirm that when a real outlier occurs, it is usually identified by both the T^2 index and SPE index. Therefore, in this chapter, the SPE index is used alone to detect outliers in the independent and dependent variable space. Specifically, if the SPE index (i.e., SPE$_x$ or SPE$_y$) violates its corresponding control limit, the sample is declared an outlier and should be analyzed further. SPE$_x$ and SPE$_y$ for a new sample to be used for soft sensor update are defined as:

$$SPE_x = \sum_{i=1}^{m} (\mathbf{x_{new,i}} - \hat{\mathbf{x}}_{\mathbf{new,i}})^2 \tag{8.3}$$

$$SPE_y = \sum_{i=1}^{p} (\mathbf{y_{new,i}} - \hat{\mathbf{y}}_{\mathbf{new,i}})^2 \tag{8.4}$$

where m is the number of independent variables, p the number of dependent variables, $\mathbf{x}_{new,i}$ and $\mathbf{y}_{new,i}$ are the new samples of independent and dependent variables, and $\hat{\mathbf{x}}_{new,i}$ and $\hat{\mathbf{y}}_{new,i}$ are the corresponding predictions. The thresholds for SPE$_x$ and SPE$_y$ can be determined based on the theorems developed by Box [198] using the training data, or they can be determined empirically using the training or validating data under normal operating conditions [123, 124].

8.3 Bayesian Supervisory Approach for Online Outlier Classification

For online outlier detection, once a new sample is identified as an outlier, it is necessary to determine whether it corresponds to an erroneous reading, or it represents a new process state. In this work, a Bayesian supervisory approach is proposed to perform this task. It should be noted that this task is less difficult for off-line data, because the data collected after the outlier(s) are available to help make the decision. In the proposed Bayesian supervisory approach, once an outlier is detected by the SPE indices, we wait for few more measurements to become available and apply Bayesian inference to make the decision. The basic assumption is that if an outlier is due to erroneous readings, the increase in the SPE indices will not be sustainable and will result in an impulse or short step disturbance in the time series of SPE indices. On the other hand, if an outlier is caused by a process change, the following samples will all deviate from the previous model and will result in a sustained step or ramp disturbance in the time series of SPE indices. Therefore, when an outlier is identified, it is tried to classify whether the change in the SPE index belongs to an impulse/short step or a ramp/step disturbance in order to determine whether the outlier is due to an erroneous reading or a process change. In this work, the classification is achieved through a Bayesian approach. By definition, Bayes' Theorem is a simple mathematical formula used for calculating conditional probabilities. Simply put, it gives the probability of a random event A occurring given that we know a related event B occurred. This probability is denoted as $P(A|B)$ and is called the posterior probability, since it is computed after all other information on A and B is known. Using Bayes' Theorem, the posterior probability can be computed as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{8.5}$$

In our case, event B corresponds to the measurements collected after the outlier is detected and event A corresponds to a specific disturbance type. In the proposed approach, instead of using the values of SPE indices directly which are often affected by the stochastic nature of the process, the indexes values are transformed into a more robust statistic-based probability description using Bayesian statistics. In this way, different processes with different dynamic characteristics can be analyzed using a unified statistical framework.

8.3.1 Bayesian Outlier Classification Procedure

Figure 8.1 shows the schematic diagram of the proposed Bayesian outlier classification procedure, which is a modification of the previously developed Bayesian approach for detection and classification of different disturbances in the semiconductor processes [199]. The classification algorithm is triggered by the identification of an outlier through SPE indices, and a brief description is provided below.

1. Denote the time index of the outlier as k; construct the pre- and post-change windows around the outlier k. The pre-change window contains a few samples SPE indices prior to the identified outlier; while the post-change window contains the SPE indices after (and including) the outlier. In this work, the width of the pre-change window is fixed to 5 samples, while the width of the post-change window varies depending on the assumed type of disturbance.

2. Wait until sample k+1 is available, then perform hypothesis testing using Bayes Theorem to determine whether the disturbance is an impulse.

3. If the hypothesis is rejected, we wait for more future samples to determine whether the sample is part of a short step disturbance (with duration 2, 3 or 4).

4. If all previous hypotheses are rejected, it is concluded that a real process change has occurred.

It is worth noting that the posterior probabilities in the post-change window form different patterns depending on the pre-assumed disturbance type and observation values in the post-change window. Instead of putting a threshold on a single posterior probability, which is a univariate method [200], the proposed Bayesian approach compares the pattern of the posterior probabilities in the post-change window to the predefined patterns in order to classify the type of disturbance. This pattern matching approach is a multivariate method, which is more robust compared to the univariate method, and greatly improves classification

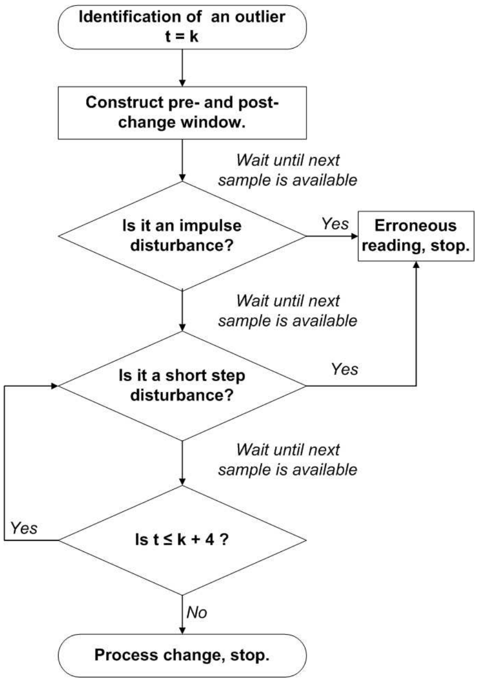

Identification of an outlier
t = k
Construct pre- and post-change window.
Wait until next sample is available
Is it an impulse disturbance?
Yes
Erroneous reading, stop.
Wait until next sample is available
Is it a short step disturbance?
Yes
Wait until next sample is available
Yes
Is t ≤ k + 4 ?
No
Process change, stop.

Figure 8.1: Bayesian disturbance classification procedure

accuracy and reduces classification delay. In addition, it makes the classification results not sensitive to the a priori probability (in this work, the a priori probability is set to 0.5). Detailed description of different patterns of posterior probability in the post-change window and pattern matching procedure can be found [199]. In addition, by specifying different post-change window widths for different disturbances, the classification delay is minimized. In other words, the classification decision is made when the minimum required information becomes available.

8.4 Soft Sensor Recursive Update with Outlier Detection and Classification

In this section, the proposed outlier detection and classification methods are integrated into the previously developed RO-DPLS soft sensor [63, 67] for online recursive update. Based on the results of a comprehensive comparison of different recursive PLS update schemes [67], regular recursive PLS updating scheme is chosen to update the RO-DPLS soft sensor model. Both simulated case study and industrial case studies of the single vessel

Kamyr digester are used to demonstrate the performance of the proposed approach. In this section, the performance of the RO-DPLS soft sensor is compared for four scenarios.

a) Static model without update;

b) Recursive update without outlier detection, i.e., all samples are used to update the model;

c) Recursive update with outlier detection, i.e., all outliers identified by SPE indices are excluded from model update;

d) Recursive update with Bayesian supervisory approach for outlier detection, i.e., erroneous readings are excluded from model update, while process changes are used for model update;

Simulated case study

In this subsection, the simulated Kamyr digester is used to illustrate how the Bayesian supervisory approach works. The extended Purdue model [146] is implemented to simulate a single vessel high yield Kamyr digester. The RO-DPLS soft sensor set up can be found in Chapter 6. In this case study, a very challenging problem is considered: tracking the disturbance of a wood type change. It is worth noting that wood type change (from softwood to hardwood and vice versa) is a major disturbance in pulping processes, and usually results in off-spec product during the transition. Both single and consecutive multiple outliers (i.e. impulses and short steps) are added to the process, and the Kappa number measurements are plotted in Figure 8.2. From Figure 8.2, it is seen that the dramatic change in Kappa number during the transition period is due to the wood type change, and the samples during the transition should be used to update the model; while the changes that occur at samples 10 (impulse), 45 (short step with duration 3) and 65 (short step with duration 2), are introduced outliers and should not be used for model update. The soft sensor predicted Kappa number values are also plotted in Figure 8.2, with the corresponding mean squared prediction error

(MSE) and mean prediction error (ME) given in Table 8.1. Both Figure 8.2 and Table 8.1 demonstrate the important roles of outlier detection and classification, and their impact on soft sensor performance. It is shown that outlier detection alone may even deteriorate the performance of a soft sensor if process changes were treated the same as erroneous measurements. On the other hand, if the proposed Bayesian outlier classification mechanism is integrated into outlier detection, the soft sensor can be made more robust to erroneous measurements and at the same time is able to track process changes. Figures 8.3(a) and 8.3(b) show the SPE$_x$ and SPE$_y$ indices for the Bayesian supervisory approach, together with the classified outliers. Figure 8.3 shows that SPE$_x$ and SPE$_y$ indices can promptly identify the outliers caused by both erroneous reading and process change, and the Bayesian

Table 8.1: Performance of different soft sensors for the simulated case study

Soft Sensor	MSE	ME
(a)	405.4800	-10.8817
(b)	1.1476	-0.1695
(c)	120.6689	-3.7614
(d)	0.8505	-0.1526

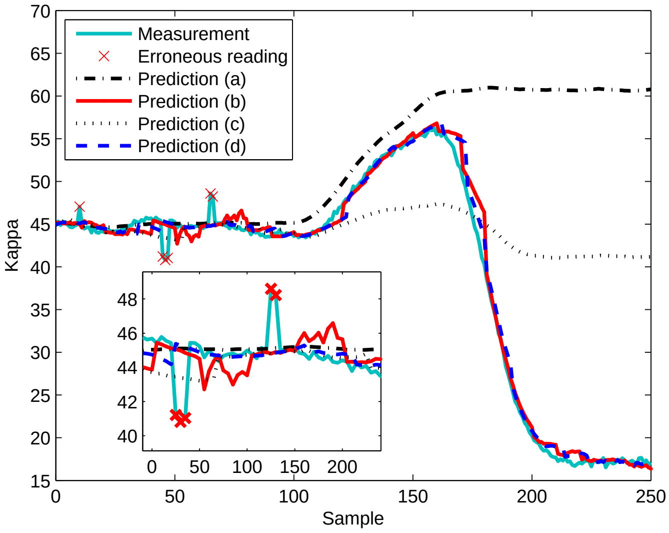
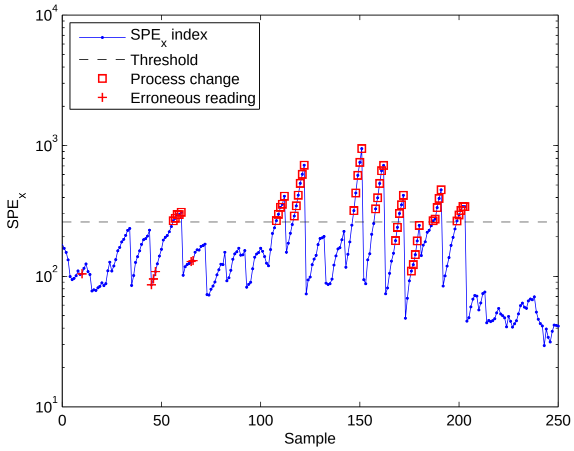
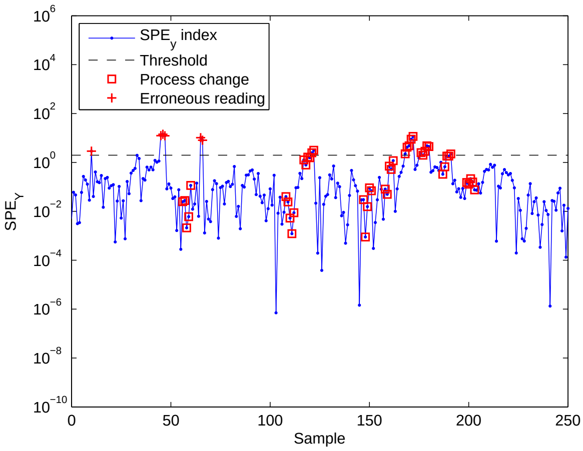


Figure 8.2: Prediction comparison of different approaches applied to a simulated case study

supervisory approach is effective in classifying the identified outliers. Without the Bayesian supervisory approach, all identified outliers will be excluded from updating the model, which results in poor prediction performance of approach (c), similar to the static model. For approach (b) where all new samples are used for model update, the negative impact of using erroneous readings for model update are illustrated more clearly with the insert in Figure 8.2.



(a)



(b)

Figure 8.3: SPE indices for the simulated case study. (a) SPE_x; (b) SPE_y

Industrial case study

In the industrial case study, the process data were collected from a Kamyr digester at a pulp mill located in Mahrt, Alabama run by MeadWestvaco Corp. The training data were collected in 2006 which contain 1100 samples, while the testing data for online update were collected in 2010 which contain 300 samples. Clearly, this case study presents a challenging problem as training and testing data sets were collected about 4 years apart. The soft sensor setup for the industrial case is the same as that reported in Chapter 6. Figure 8.4 plots a segment of the testing data to illustrate the prediction performance of different soft sensors and compare them with the process measurements, and Table 8.2 lists the MSE and ME of different approaches for the whole testing data set. Figures 8.5(a) and 8.5(a) show the SPE_x and SPE_y indices for the Bayesian supervisory approach, together with the classified outliers. It should be noted that for this case study the soft sensor that updates recursively with outlier detection (but without classification) performs exactly the same as the static soft sensor. This is due to the big difference between the training data and testing data, which causes all new data to be classified as outliers. From this case study, it is clear that

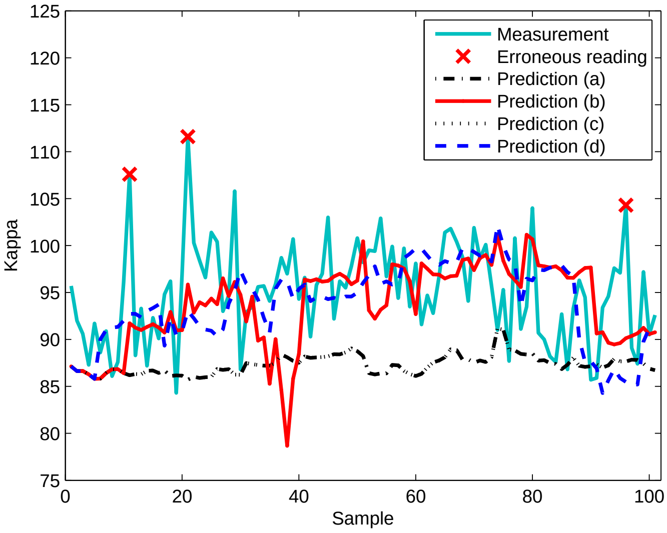
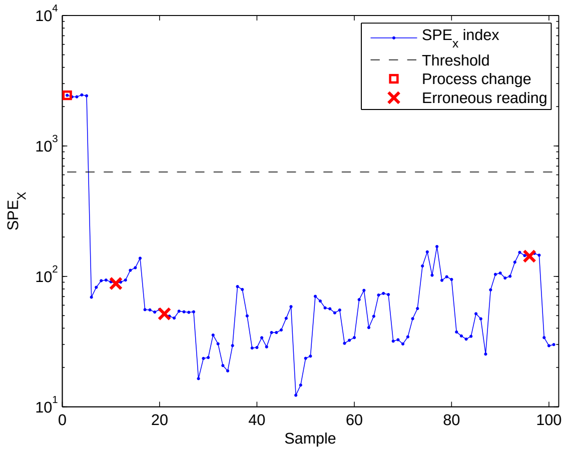


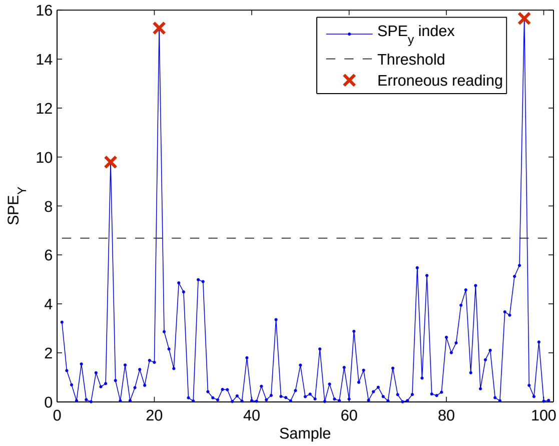
Figure 8.4: Comparison of predictions of different approaches for the industrial case study

Table 8.2: Performance of different soft sensors for the industrial case study

Soft Sensor	MSE	ME
(a)	75.3031	7.3024
(b)	34.8632	1.0535
(c)	75.3031	7.3024
(d)	29.4228	0.7468



(a)



(b)

Figure 8.5: SPE indices for the industrial case study. (a) SPE_x; (b) SPE_y

the Bayesian supervisory approach is effective and robust in determining whether an outlier is caused by erroneous reading or process change.

8.5 Conclusion and Discussion

Outlier detection and handling plays a critical role in data-driven soft sensor development. In this chapter, multivariate approaches for both off-line outlier detection (for initial soft sensor model building) and online outlier detection (for soft sensor model recursive update) are proposed. Specifically, for off-line outlier detection leverage and y-studentized residuals are combined; while for online outlier detection, the squared prediction error indices for **X** and **Y** are used to monitor the independent variable and dependent variable space, respectively. For online outlier detection, to differentiate the outliers caused by erroneous reading from those caused by process changes, a Bayesian supervisory approach is proposed to further analyze and classify the identified outliers. The classification of outliers is achieved through matching the pattern generated by the posterior probabilities in the post-change window to pre-defined patterns corresponding to different disturbances. Both simulated and industrial case studies show that by updating the soft sensor model and correctly analyzing the identified outliers, the soft sensor enhances its prediction performance qualitatively and quantitatively.

Chapter 9

Conclusions and Proposed Future Work

9.1 Summary of Contributions

The overall goal of this work is to advance the state-of-the-art of industrial process monitoring and soft sensor design techniques.

9.1.1 Process Monitoring

As industrial processes become more complex, advance monitoring techniques play a fundamental role in assessing the process performance. By promptly detecting process upsets and/or instrumentation faults, process monitoring techniques not only aid in improving product quality and process efficiency, but also guarantee a safe operation. In this work, the potential for fault detection and diagnosis of a new state-of-the-art process monitoring framework is studied. The statistics pattern analysis framework, by utilizing the statistics of the process instead of the process variables themselves, not only improves the detection performance compared with traditional linear process monitoring methods, but also outperforms nonlinear extensions of PCA-based monitoring such as Kernel PCA (KPCA) and independent component analysis (ICA). Furthermore, due to the flexible nature of the framework, different statistics can be incorporated to the statistic pattern (SP) to capture the dominant process characteristics such as nonlinearity, non-Gaussianity and dynamics. Additionally, in SPA, the diagnosis of the root cause of different faults is improved not only by providing a clear indication of the contribution of the variable that is most related to the given fault, but indicating the nature of the type of fault (e.g. mean or variance change).

In this work, the properties of the SPA-based monitoring of continuous processes are studied using two different case studies. With the quadruple-tank case study, the SPA-based process monitoring showed that it can correctly detect and diagnose faults of a nonlinear process that are difficult to detect visually or by using traditional monitoring approaches. In addition, in this case study SPA showed that it can offer excellent fault detection rates in the event of limited samples in the training data set. Evidently, due to the calculation of different statistics, when the training data are limited, window overlapping is necessary to have a representative number of SPs. In the study, an increase in the false alarms rate is observed as the percentage in overlapping increases. This effect is caused by the lack of independence if the samples (e.g. SPs). The study showed that for a case with an overlapping of around 60%, SPA still observes a high detection performance and a reasonable false alarm rate. On the other hand, when the number of training samples is significant, SPA is able to detect and diagnose faults in a more reliable manner. Type I errors in normal testing data (e.g. false alarms) are close to the expected confidence level. In addition, both SPA-based fault detection and diagnosis is significantly better than traditional monitoring methods.

In the TEP case study, the SPA framework was tested to monitor a more complex process. The good performance observed in the quadruple-tank case study is confirmed with the more complicated TEP process. In this case study, the performance of SPA was compared to other linear and nonlinear monitoring methods. By monitoring the statistics of the process variables instead of the process variables themselves, SPA was able to outperform the detection and diagnosis performance of the other monitoring methods. Furthermore, even that detection delay is associated with window-based approaches like SPA, it this study, it was found that after the assembly of the validating window, the detection delays of SPA compared to other methods are not significant. A closer look at the detection performance indicates that SPA-based monitoring can offer a more consistent and reliable detection compared to the other methods, which are important characteristics to consider in an online implementation. In regards of the computation load, the complexity of SPA compared with the two linear methods (e.g. PCA and DPCA) is not significant, and much less expensive than the observed in the nonlinear case studies evaluated in this study. For fault diagnosis, the contributions to the different monitoring indices in SPA show a clear and reliable indication of the root cause of different faults. Additionally, the hierarchical contribution approach provides the framework with extra information about the nature of the faults (e.g. mean or variance change). As a summary, the research findings indicate that the SPA-based monitoring is a promising alternative for monitoring processes especially where nonlinearities and multi-modal distributions are present.

9.1.2 Soft Sensor Development

In modern process operation, a massive amount of data from secondary measurements (e.g. easy-to-measure) are readily available at high frequencies, but data indicating product quality and/or yield are scarce and often times sampled at low frequency (e.g. hard to measure). As a result, developing data-driven software sensors (or soft sensors) using secondary measurements (e.g. variables easy to measure) to predict primary measurements based on multivariate regression techniques is gaining wide application in industry. The prediction of these key process variables not only provides a way to have real time measurements, but also aids in improving quality and preventing undesirable operating conditions. In this work, the dynamic partial least squares (DPLS) soft sensor is studied in depth due to its successful application in industry. Despite its successful applications, before this work, there was a lack of theoretical understanding on the properties of the DPLS soft sensor. One of the major contributions of this dissertation is the development of a theoretical understanding of why a DPLS-based soft sensor can adequately capture process dynamics and provides unbiased estimates under closed-loop operation. The theoretical analysis confirms that a DPLS soft sensor is a dynamic estimator that gives unbiased estimate of primary measurements for both open loop and closed-loop data. In addition, to overcome the difficulties related with processes with large transport delays, a reduced-order DPLS (RO-DPLS) soft sensor approach is proposed.

Real industrial processes have time-varying behavior and soft sensors developed off-line are prone to fail as time progresses, thus indicating that an updating strategy is necessary. To address this issue, the RO-DPLS soft sensor is also extended to its adaptive version by exploring the properties and the characteristics of different recursive update schemes. In addition, practical issues such as data scaling and the presence of outliers off-line and online are studied.

It is well known that introducing outliers at the design and update stage of any data-driven approach will deteriorate its performance. The task of detecting outliers off-line is relatively easier compared to do it online. One of the major reasons is that the data and expert knowledge are present to make a decision. For online outlier detection, the resources are limited as expert inputs are not available to base the decision. Furthermore, outliers online can be part of two groups; erroneous readings and process change. Outliers that are part of a group of erroneous readings need to be identified and discarded from the update procedure. On the other hand, outliers that are part of a process change are necessary, and need to be kept to update the soft sensor model. In this aspect, the contribution of this work is significant. Based on multivariate techniques for outlier detection online (e.g. SPE), once an outlier is detected online, the monitoring indices are transformed into probabilities of the occurrence of a given type of fault (e.g. impulse, short step etc.). In this work, a Bayesian-based approach is developed to further analyze the detected outliers and the probabilities obtained from the SPE indices. These probabilities are compared with patterns of probabilities corresponding to different type of events. For example, with the Bayesian-based supervisory approach, SPE indices corresponding to samples that are part of impulses and short steps can be categorized as outliers. On the other hand, if the pattern of probabilities defines a sustained change (e.g. process change), those identified outliers are not discarded, and the multivariate observations of these secondary measurements are used to update the soft sensor model. In both, simulation and industrial cases, the recursive RO-DPLS soft sensor with outlier detection and classification approach observed enhanced prediction capabilities if compared with the static RO-DPLS and with recursive-only RO-DPLS. As a summary, the RO-DPLS has been equipped with the necessary tools for handling challenging cases where a prediction of a key indicator of production performance is required.

9.2 Suggestions for Future Work

Future research directions which deserve further investigation are summarized in this section.

9.2.1 Process Monitoring

- As discussed in this work, the SPA-based fault detection and diagnosis is a promising alternative for monitoring industrial processes, especially when nonlinearities are prevalent. One of the major requirements of the framework is the availability of data to obtain a number of training SPs (e.g. observations) that is statistically significant compare to the number of statistics included in the SP (e.g. variables). Nowadays, processes can be categorized as data rich, and thus this is not a big limitation. However, to address this requirement, the following strategy to monitor continuous processes only is proposed (for batch processes the advantages provided by SPA are far superior than other methods). First, after the collection of the process variables that will be part of the monitoring strategy, a PCA decomposition is performed to obtain both, scores and loadings representing the process. Second, a subset of scores (e.g. defined by the retained number of principal components) is used to characterize the process. In other words, although nonlinearities might be present and obviously inherited by the scores in PCA, redundancies and measurement noise are discarded in the last few components, and only the first few scores are retained to represent the process. Third, instead of calculating the statistics of the process variables to perform SPA, which require the availability of a good amount of data, the statistics of the first few retained scores are determined to perform SPA in the latent variable space. At the time of the preparation of this document, preliminary results indicate that although PCA fails to detect faults for some cases (e.g. especially when nonlinearities are prevalent), the statistics of the projections (or scores) contain enough information to correctly define normal and faulty operating conditions. Thus, enhancing the detection power of regular PCA. In addition, since the number of retained scores a is normally much lower than the number of available variables m, the number of observations needed to obtain the matrix of SPs is much less (e.g. data to calculate different statistics). One challenge associated with this approach is to develop a way to diagnose the variable that is most related to the fault. In other words, since two PCA models are used (e.g. for reducing the dimensionality, and for SPA), a method to back-calculate or estimate the contribution of the input variables based on the contributions of the different statistics to a fault in the latent variable space needs to be defined.

- A more thorough study of different properties of SPA need to be performed. In this work, the effect of adding/removing different statistics, the effect of window-overlap and fault diagnosis issues were addressed. More insights need to be developed to generate rules of thumb to define groups of statistics that need to be included as pat of the SP, based on the known characteristics of the process, which is important from the point of view of online implementation. An additional factor is the study of providing weights to different groups of statistics. Preliminary results indicate that not only defining the right set of statistics enhances both, detection and diagnosis performance, but also the weights of different statistics can contribute to achieve these two goals.

- In this work, the SPA framework has been shown to be a promising alternative for monitoring industrial processes, especially where nonlinearities and multi-modal distributions are present. Additionally, the framework has shown good results even in the event of window overlapping. Although, depending on the amount of overlapping, an increase in the false alarm rates is observed. Closer evaluation of these results raises one question, Is the overlapping the only source of the increment of the false alarms rates?. In this regard, our experience indicate that other factors are involved and can be studied to overcome this issue. First, it has been observed that although cross-validation is a reasonable way to determine the number of principal components to use as representative of the input space, it can provide a number of PCs that only improves fault detection, but does not offer a balance of reduction of Type I and Type II errors. In this regard, more testing needs to be done to evaluate this effect. In addition, for the sake of a fair comparison, the empirical way of determining the SPs is utilized in this work. However, it is possible that developing a way to determining the thresholds for monitoring, taking into account the amount of overlapping, can aid balancing the fault detection performance and the reduction of the false alarm rates of normal testing and validating data.

9.2.2 Soft Sensor Development

- As PCA, the SPA framework can be extended to the development of soft sensors. In other words, the statistics of secondary measurements in a batch or a window can be utilized to predict key product quality variables. In this aspect, preliminary results show that SPA-based soft sensors can effectively outperform the prediction power offered by soft sensors based on PCA and PLS [201]. A preliminary study with data from a semiconductor fabrication site have already shown promising results.

- As discussed previously, the recursive RO-DPLS soft sensor with outlier detection and classification approach has shown impressive results in tracking the variability of kappa number for the digester case study. In this regard, additional testing needs to be done to expand the outlier detection and classification approach, to handle the update of the thresholds of monitoring instead of a static threshold. Preliminary results in this matter indicate that providing an update of the monitoring threshold can cope with issues of transitioning the SPE indexes to a different operating region, and still provide consistent outlier detection close to the given significance level.

- Although in this work some efforts were done in regards to making use of the prediction provided by the RO-DPLS soft sensor for control, an extended comparison of the benefits associated with it can be done. In other words, different control strategies such as Smith predictor with PID and model predictive control (MPC) can be implemented to highlight further the benefits associated with the future predictions provided by the RO-DPLS soft sensor.

Bibliography

[1] Walter A. Shewhart. *Economic Control of Quality of Manufactured Product.* Van Nostrand, Princeton, 1931.

[2] J. Edward Jackson. Principal components and factor analysis: Part I- principal components. *Journal of Quality Technology*, 12:201–213, 1980.

[3] Douglas M. Hawkins and David H. Olwell. *Cumulative sum charts and charting for quality improvement.* Springer, 1998.

[4] J. Stuart Hunter. Exponentially weigthed moving average. *Journal of Quality Technology*, 18:203–210, 1986.

[5] John F. MacGregor, Christiane Jaeckle, Costas Kiparissides, and M. Koutoudi. Process monitoring and diagnosis by multiblock PLS methods. *AIChE Journal*, 40(5):826–838, 1994.

[6] J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414, 1995.

[7] Q. Peter He, S. Joe Qin, and Jin Wang. A new fault diagnosis method using fault directions in Fisher discriminant analysis. *AIChE Journal*, 51(2):555–571, 2005.

[8] S. Joe Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502, 2003.

[9] Theodora Kourti, Paul Nomikos, and John F. MacGregor. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *Journal of Process Control*, 5(4):277–284, 1995.

[10] Theodora Kourti, Jennifer Lee, and John F. Macgregor. Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering*, 20(Supplement 1):S745–S750, 1996.

[11] Changkyu Lee, Sang Wook Choi, and In-Beum Lee. Sensor fault identification based on time-lagged PCA in dynamic processes. *Chemometrics and Intelligent Laboratory Systems*, 70(2):165–178, 2004.

[12] W.E. Deming. *Some Theory of Sampling.* John Wiley, New York, 1950.

[13] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[14] J. Edward Jackson. Quality control methods for several related variables. *Technometrics*, 1(4):359–377, 1959.

[15] J. Edward Jackson and Govind S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

[16] Barry M. Wise and Neal B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6):329–348, 1996.

[17] J. Edward Jackson. *A User's guide to Principal Components.* John Wiley and Sons, Inc., 1991.

[18] Sergio Valle, Weihua Li, and S. Joe Qin. Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research*, 38(11):4389–4401, 1999.

[19] Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[20] P. Geladi. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2(4):231–246, 1988.

[21] Bhupinder. S. Dayal and John F. MacGregor. Improved PLS algorithms. *Journal of Chemometrics*, 11(1):73–85, 1997.

[22] Nouna Kettaneh, Anders Berglund, and Svante Wold. PCA and PLS with very large data sets. *Computational Statistics and Data Analysis*, 48(1):69–85, 2005.

[23] Philip R. C. Nelson, Paul A. Taylor, and John F. MacGregor. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems*, 35(1):45–65, 1996.

[24] Svante Wold, Michael Sjstrm, and Lennart Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001.

[25] Agnar Hskuldsson. PLS regression methods. *Journal of Chemometrics*, 2(3):211–228, 1988.

[26] Kresta J. V., MacGregor J. F., and Marlin T. E. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.

[27] Davis B. Ricker N.L. Kowalski B.R. Wise B.M., Veltkamp D.J. Principal component analysis for monitoring the West Valley liquid fed ceramic melter. *Waste Management'88 Proc.*, pages 811–818, 1988.

[28] Paul Nomikos and John F. MacGregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40(8):1361–1375, 1994.

[29] Wenfu Ku, Robert H. Storer, and Christos Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196, 1995.

[30] Y. Rotem, A. Wachs, and D. R. Lewin. Ethylene compressor monitoring using model-based PCA. *AIChE Journal*, 46(9):1825–1836, 2000.

[31] J.J. Downs and J.E. Doss. Present status and future needs- A view from north american industry. In Y. Arkun and H. Ray, editors, *AIChE Symposium Proceedings of the Fourth International Conference of Chemical Process Control*, volume P-67, Padre Island, TX, 1991. AIChE publication.

[32] Jin Wang and Q. Peter He. Multivariate statistical process monitoring based on statistics pattern analysis. *Industrial and Engineering Chemistry Research*, 49(17):7858–7869, 2010.

[33] Jong-Min Lee, S. Joe Qin, and In-Beum Lee. Fault detection and diagnosis based on modified independent component analysis. *AIChE Journal*, 52(10):3501–3514, 2006.

[34] Paige Miller, Ronald E. Swanson, and Charles E. Heckler. Contribution plots: A missing link in multivariate quality control. *Applied Mathemathics and Computer Science*, 8:755–792, 1988.

[35] Theodora Kourti. Application of latent variable methods to process control and multivariate statistical process control in industry. *International Journal of Adaptive Control and Signal Processing*, 19(4):213–246, 2005.

[36] A. K. Conlin, E. B. Martin, and A. J. Morris. Confidence limits for contribution plots. *Journal of Chemometrics*, 14(5-6):725–736, 2000.

[37] D.G. Luenberger. Observing the state of a linear system. *IEEE Transactions of Military Electronics*, MIL-8:74–80, 1964.

[38] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transcations of the ASME, Journal of Basic Engineering*, 87:35–45, 1960.

[39] Eliana Zamprogna, Massimiliano Barolo, and Dale E. Seborg. Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis. *Journal of Process Control*, 15(1):39–52, 2005.

[40] N.R. Draper and H. Smith. *Applied Regression Analysis.* Wiley, New York, 1981.

[41] T. Kourti. 4.02 - Multivariate Statistical Process Control and Process Control, Using Latent Variables. In D. Brown Editors-in Chief:Stephen, Tauler Rom, and Walczak Beata, editors, *Comprehensive Chemometrics*, pages 21–54. Elsevier, Oxford, 2009.

[42] Theodora Kourti and John F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28(1):3–21, 1995.

[43] Rumana Sharmin, Uttandaraman Sundararaj, Sirish Shah, Larry Vande Griend, and Yi-Jun Sun. Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant. *Chemical Engineering Science*, 61(19):6372–6384, 2006.

[44] S. Lakshminarayanan, Sirish L. Shah, and K. Nandakumar. Modeling and control of multivariable processes: Dynamic PLS approach. *AIChE Journal*, 43(9):2307–2322, 1997.

[45] E. Zamprogna, M. Barolo, and D.E. Seborg. Composition estimations in a middle-vessel batch distillation column using artificial neural networks. *Chemical Engineering Research and Design*, 79(6):689–696, 2001.

[46] Eliana Zamprogna, Massimiliano Barolo, and D.E. Seborg. Estimating product composition profiles in batch distillation via partial least squares regression. *Control Engineering Practice*, 12(7):917–929, 2004.

[47] Thor Mejdell and Sigurd Skogestad. Estimation of distillation compositions from multiple temperature measurements using partial-least-squares regression. *Industrial and Engineering Chemistry Research*, 30(12):2543–2555, 1991.

[48] Manabu Kano, Koichi Miyazaki, Shinji Hasebe, and Iori Hashimoto. Inferential control system of distillation compositions using dynamic partial least squares regression. *Journal of Process Control*, 10(2-3):157–166, 2000.

[49] Luigi Fortuna, S. Graziani, Alessandro Rizzo, and M. G. Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes.* London: Springer, 2010.

[50] Thor Mejdell and Sigurd Skogestad. Composition estimator in a pilot-plant distillation column using multiple temperatures. *Industrial and Engineering Chemistry Research*, 30(12):2555–2564, 1991.

[51] J. V. Kresta, T. E. Marlin, and J. F. MacGregor. Development of inferential process models using PLS. *Computers and Chemical Engineering*, 18(7):597–611, 1994.

[52] S. J. Qin and T. J. McAvoy. Nonlinear FIR modeling via a neural net PLS approach. *Computers and Chemical Engineering*, 20(2):147–159, 1996.

[53] S.J. Qin and T.J. McAvoy. Nonlinear PLS modeling using neural networks. *Computers and Chemical Engineering*, 16(4):379–391, 1992.

[54] S.Joe Qin. Neural networks for intelligent sensors and control–practical issues and some solutions. In Omid Omidvar and David L. Elliott, editors, *Neural Systems for Control*, pages 213–234. Academic Press, San Diego, 1997.

[55] Michael H. Kaspar and W. Harmon Ray. Dynamic PLS modelling for process control. *Chemical Engineering Science*, 48(20):3447–3461, 1993.

[56] Ognjen Marjanovic, Barry Lennox, David Sandoz, Keith Smith, and Milton Crofts. Real-time monitoring of an industrial batch process. *Computers and Chemical Engineering*, 30(10-12):1476–1481, 2006.

[57] Tiina Komulainen, Mauri Sourander, and Sirkka-Liisa Jms-Jounela. An online application of dynamic PLS to a dearomatization process. *Computers and Chemical Engineering*, 28(12):2611–2619, 2004.

[58] Bao Lin, Bodil Recke, Jrgen K.H. Knudsen, and Sten Bay Jrgensen. A systematic approach for soft sensor development. *Computers and Chemical Engineering*, 31(5-6):419–425, 2007.

[59] Q. P. He and J. Wang. Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes. *AIChE Journal*, 57:107–121, 2011.

[60] H.J. Galicia, Q.P. He, and J. Wang. Statistics pattern analysis - fault detection and diagnosis. In *Proceedings of the ISA Automation Week*, Mobile, AL, USA, 2011.

[61] H.J. Galicia, Q.P. He, and J. Wang. Statistics Pattern Analysis based fault detection and diagnosis. *CPC VIII Conference*, 55, 2012.

[62] H.J. Galicia, Q.P. He, and J. Wang. A comprehensive evaluation of Statistics Pattern Analysis based process monitoring. In *International Symposium on Advanced Control of Chemical Processes ADCHEM*, 2012. Accepted.

[63] Hector J. Galicia, Q. Peter He, and Jin Wang. A reduced order soft sensor approach and its application to a continuous digester. *Journal of Process Control*, 21(4):489–500, 2011.

[64] H.J. Galicia, Q.P. He, J. Wang, Hodges R., G. Krishnagopalan, and H. Cullinan. Outlier detection and process monitoring with application to a recursive soft sensor update for digester control. *AIChE Annual Meeting*, 2010.

[65] H.J. Galicia, Q.P. He, J. Wang, Hodges R., G. Krishnagopalan, and H. Cullinan. A multi-model recursive subspace identification based dynamic soft sensor approach for digester control. *AIChE Annual Meeting*, 2009.

[66] H.J. Galicia, Q.P. He, J. Wang, Hodges R., G. Krishnagopalan, and H. Cullinan. A subspace identification based dynamic soft sensor for digester control. *AIChE Annual Meeting*, 2008.

[67] H. J. Galicia, Q. P. He, and J. Wang. Comparison of the performance of a reduced-order dynamic PLS soft sensor with different updating schemes for digester control. *Control Engineering Practice*, 2012. Accepted.

[68] H.J. Galicia, Q.P. He, and J. Wang. Adaptive kappa number prediction via reduced-order dynamic PLS soft sensor for the control of a continious Kamyr digester. In *TAPPI-Control Systems Conference*, 2012. Accepted.

[69] H.J. Galicia, Q.P. He, and J. Wang. Recursive update of a reduced-order dynamic PLS soft sensor and its application to digester control. In *Proceedings of the ISA Automation Week*, Mobile, AL, USA, 2011.

[70] H.J. Galicia, Q.P. He, and J. Wang. Adaptive reduced-order dynamic PLS soft sensor with application to digester control. In *Proceedings of the SHPE Conference*, Anaheim, CA, USA, 2011.

[71] H.J. Galicia, Q.P. He, and J. Wang. A Bayesian supervisory approach of outlier detection for recursive soft sensor update. *CPC VIII Conference*, 54, 2012.

[72] H.J. Galicia, Q.P. He, and J. Wang. Adaptive outlier detection and classification for online soft sensor update. In *International Symposium on Advanced Control of Chemical Processes ADCHEM*, 2012. Accepted.

[73] I. T. Jolliffe. *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[74] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(13):37–52, 1987.

[75] Jan J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 14(16):375–381, 1981.

[76] Barry M. Wise. *Adapting Multivariate Analysis for Monitoring and Modeling Dynamic Systems.* Ph.D. dissertation, 1991.

[77] H. Hotelling. The generalization of student's ratio. *Annals of Mathematical Statistics*, 2(3):360–378, 1931.

[78] Paul Nomikos and J. F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.

[79] Paul Nomikos. Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis. *ISA Transactions, World Batch Forum, Toronto, May 1996*, 35(3):259–266, 1996.

[80] S. Joe Qin, Sergio Valle, and Michael J. Piovoso. On unifying multiblock analysis with application to decentralized process monitoring. *Journal of Chemometrics*, 15(9):715–742, 2001.

[81] Evan Russell, Leo H. Chiang, and Richard D. Braatz. *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes.* Springer, London, 2000.

[82] Junghui Chen and Kun-Chih Liu. On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chemical Engineering Science*, 57(1):63–75, 2002.

[83] Weihua Li and S. Joe Qin. Consistent dynamic PCA based on errors-in-variables subspace identification. *Journal of Process Control*, 11(6):661–678, 2001.

[84] Rongfu Luo, Manish Misra, and David M. Himmelblau. Sensor fault detection via multiscale analysis and dynamic PCA. *Industrial and Engineering Chemistry Research*, 38(4):1489–1495, 1999.

[85] Ningyun Lu, Yuan Yao, Furong Gao, and Fuli Wang. Two-dimensional dynamic PCA for batch process monitoring. *AIChE Journal*, 51(12):3300–3304, 2005.

[86] G.E.P. Box, M. J. Gwilym, and G.C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall, Eglewood Cliffs, NJ, 1994.

[87] D. Dong and T. J. McAvoy. Nonlinear principal component analysis based on principal curves and neural networks. *Computers and Chemical Engineering*, 20(1):65–78, 1996.

[88] Ji-Hoon Cho, Jong-Min Lee, Sang Wook Choi, Dongkwon Lee, and In-Beum Lee. Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 60(1):279–288, 2005.

[89] B. Schlkopf, A. Smola, and K. Mller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1399, 1998.

[90] Sang Wook Choi, Changkyu Lee, Jong-Min Lee, Jin Hyun Park, and In-Beum Lee. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75(1):55–67, 2005.

[91] Zhiqiang Ge, Chunjie Yang, and Zhihuan Song. Improved kernel PCA-based monitoring approach for nonlinear processes. *Chemical Engineering Science*, 64(9):2245–2255, 2009.

[92] S. Mika, B. Schlkopf, A. J. Smola, K.-R. Mller, M. Scholz, and G. Rtsch. Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems 11*, pages 536–542, 1999.

[93] Takashi Takahashi and Takio Kurita. Robust de-noising by Kernel PCA artificial neural networks ICANN 2002. volume 2415 of *Lecture Notes in Computer Science*, pages 789–789. Springer Berlin / Heidelberg, 2002.

[94] S. Haykin. *Neural Networks.* Prentice Hall, Englewood Cliffs, NJ, 1999.

[95] D. Tracy, N. Multivariate control charts for individual observations. *Journal of Quality Technology*, 24(2):88–95, 1992.

[96] Ricardo Dunia, S. Joe Qin, Thomas F. Edgar, and Thomas J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42(10):2797–2812, 1996.

[97] Christian Jutten and Jeanny Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

[98] M. Girolami. *Self-Organising Neural Networks: Independent Component Analysis and Blind Source Separation.* Springer-Verlag, London, 1991.

[99] Pierre Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314, 1994.

[100] Aapo Hyvrinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[101] Aapo Hyvrinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10(1):1–5, 1999.

[102] A. Hyvrinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[103] Manabu Kano, Shouhei Tanaka, Shinji Hasebe, Iori Hashimoto, and Hiromu Ohno. Monitoring independent components for fault detection. *AIChE Journal*, 49(4):969–976, 2003.

[104] T. Lee. *Independent Component Analysis: Theory and Applications.* MA: Kluwer Academic, Boston, 1998.

[105] A. Hyvrinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley, New York, 2001.

[106] Jong-Min Lee, ChangKyoo Yoo, and In-Beum Lee. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5):467–485, 2004.

[107] Jong-Min Lee, ChangKyoo Yoo, and In-Beum Lee. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical Engineering Science*, 59(14):2995–3006, 2004.

[108] Jong-Min Lee, ChangKyoo Yoo, and In-Beum Lee. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical Engineering Science*, 59(14):2995–3006, 2004.

[109] Zhinong Li, Yongyong He, Fulei Chu, Jie Han, and Wei Hao. Fault recognition method for speed-up and speed-down process of rotating machinery based on independent component analysis and factorial hidden Markov model. *Journal of Sound and Vibration*, 291(1-2):60–71, 2006.

[110] George Stefatos and A. Ben Hamza. Dynamic independent component analysis approach for fault detection and diagnosis. *Expert Systems with Applications*, 37(12):8606–8617, 2010.

[111] Abdulhamit Subasi and M. Ismail Gursoy. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, 37(12):8659–8666, 2010.

[112] Yingwei Zhang and Yang Zhang. Fault detection of non-Gaussian processes based on modified independent component analysis. *Chemical Engineering Science*, 65(16):4630–4639, 2010.

[113] Zhizhou Li, Guohua Liu, Rui Zhang, and Zhencai Zhu. Fault detection, identification and reconstruction for gyroscope in satellite based on independent component analysis. *Acta Astronautica*, 68(7-8):1015–1023, 2011.

[114] Yiu-Ming Cheung and Lei Xu. An empirical method to select dominant independent components in ICA for time series analysis. In *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, volume 6, pages 3883–3887 vol.6, 1999.

[115] A.D. Back and A.S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems*, 8(4):473–484, 1997.

[116] J. F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *Radar and Signal Processing, IEE Proceedings F*, 140(6):362–370, 1993.

[117] Yiu-ming Cheung and Lei Xu. Independent component ordering in ICA time series analysis. *Neurocomputing*, 41(14):145–152, 2001.

[118] Paul Nomikos and John F. MacGregor. Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30(1):97–108, 1995.

[119] Johan A. Westerhuis, Stephen P. Gurden, and Age K. Smilde. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51(1):95–114, 2000.

[120] T Adamson, G. Moore, M. Passow, J. Wong, and Y. Xu. Strategies for succesfully implementing fab-wide FDC methodologies in semiconductor manufacturing, 2006.

[121] M. A. A. Shoukat Choudhury, Sirish L. Shah, and Nina F. Thornhill. Diagnosis of poor control-loop performance using higher-order statistics. *Automatica*, 40(10):1719–1728, 2004.

[122] Ashish Singhal and Dale E. Seborg. Pattern matching in multivariate time series databases using a moving-window approach. *Industrial and Engineering Chemistry Research*, 41(16):3822–3838, 2002.

[123] Barry M. Wise, Neal B. Gallagher, Stephanie Watts Butler, Daniel D. White, and Gabriel G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13(3-4):379–396, 1999.

[124] Evan L. Russell, Leo H. Chiang, and Richard D. Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 51(1):81–93, 2000.

[125] Harald Cramer. *Mathematical Methods of Statistics.* Princeton University Press, Princeton, NJ, 1946.

[126] F. R. Pene. Rate of convergence in the multidimensional central limit theorem for stationary processes. application to the knudsen gas and to the sinai billiard. *Annals of Applied Probability*, 15:2331–2392, 2005.

[127] Jrme Dedecker and Emmanuel Rio. On mean central limit theorems for stationary sequences. *Annales de l'Institut Henri Poincar, Probabilits et Statistiques*, 44:693–726, 2008.

[128] E. B. Martin and A. J. Morris. Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control*, 6(6):349–358, 1996.

[129] Manabu Kano, Koji Nagao, Shinji Hasebe, Iori Hashimoto, Hiromu Ohno, Ramon Strauss, and Bhavik R. Bakshi. Comparison of multivariate statistical process monitoring methods with applications to the Eastman challenge problem. *Computers and Chemical Engineering*, 26(2):161–174, 2002.

[130] J.J. Downs and E.F. Vogel. A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3):245–255, 1993.

[131] K.H. Johansson. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *Control Systems Technology, IEEE Transactions*, 8(3):456–465, 2000.

[132] A. Banerjee and Y. Arkun. Control configuration design applied to the Tennessee Eastman plant-wide control problem. *Computers and Chemical Engineering*, 19(4):453–480, 1995.

[133] Leo H. Chiang, Evan L. Russell, and Richard D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer, London, 2001.

[134] P. R. Lyman and C. Georgakis. Plant-wide control of the Tennessee Eastman problem. *Computers and Chemical Engineering*, 19(3):321–331, 1995.

[135] N.L. Ricker. Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, 6(4):205–221, 1996.

[136] D. J. H. Wilson and G. W. Irwin. PLS modelling and fault detection on the Tennessee Eastman benchmark. *International Journal of Systems Science*, 31(11):1449–1457, 2000.

[137] Kristian Helland, Hans E. Berntsen, Odd S. Borgen, and Harald Martens. Recursive algorithm for partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 14(1-3):129–137, 1992.

[138] S. Qin. Recursive PLS algorithms for adaptive data modeling. *Computers and Chemical Engineering*, 22(4-5):503–514, 1998.

[139] Petr Kadlec, Ratko Grbic, and Bogdan Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Computers and Chemical Engineering*, 35(1):1–24, 2011.

[140] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt. Data-driven soft sensors in the process industry. *Computers and Chemical Engineering*, 33(4):795–814, 2009.

[141] M. Kocurek, T. Grace, and E. Malcolm. *Pulp and Paper Manufacture - Alkaline Pulping*. TAPPI-USA and CPPA-Canada, Norcross, GA, third edition edition, 1989.

[142] N. DeKing. *Pulp and paper global fact and price book 2003-2004*. Paper loop, Inc., Boston, 2004.

[143] J. J. Castro and F. J. Doyle. A pulp mill benchmark problem for control: application of plantwide control design. *Journal of Process Control*, 14(3):329–347, 2004.

[144] M. Paulonis and G. Krishnagopalan. Adaptive inferential control of kraft batch digesters as based on pulping liquor analysis. *TAPPI Journal*, 74:169–175, 1991.

[145] C.C. Smith and T.J. Williams. Mathematical modeling, simulation and control of the operation of kamyr continuous digester for kraft process. Technical Report 64, Purdue University, PLAIC, West Lafayette, IN 47907, 1974.

[146] Philip A. Wisnewski, Francis J. Doyle III, and Ferhan Kayihan. Fundamental continuous-pulp-digester model for simulation and control. *AIChE Journal*, 43(12):3175–3192, 1997.

[147] F.J. Doyle. Nonlinear inferential control for process applications. *Journal of Process Control*, 8(56):339–353, 1998.

[148] III Doyle, F.J. and F. Kayihan. Reaction profile control of the continuous pulp digester. *Chemical Engineering Science*, 54:2679–2688, 1999.

[149] Philip A. Wisnewski and Francis J. Doyle. A reduced model approach to estimation and control of a Kamyr digester. *Computers and Chemical Engineering*, 20(Supplement 2):S1053–S1058, 1996.

[150] Philip A. Wisnewski and Francis J. Doyle. Control structure selection and model predictive control of the Weyerhaeuser digester problem. *Journal of Process Control*, 8(5-6):487–495, 1998.

[151] Jay H. Lee and A. K. Datta. Nonlinear inferential control of pulp digesters. *AIChE Journal*, 40(1):50–64, 1994.

[152] Raja Amirthalingam and Jay H. Lee. Subspace identification based inferential control applied to a continuous pulp digester. *Journal of Process Control*, 9(5):397–406, 1999.

[153] R. Amirthalingam, S. W. Sung, and J. H. Lee. Two-step procedure for data-based modeling for inferential control applications. *AIChE Journal*, 46(10):1974–1988, 2000.

[154] Bhupinder S. Dayal and John F. MacGregor. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *Journal of Process Control*, 7(3):169–179, 1997.

[155] Weihua Li, H. Henry Yue, Sergio Valle-Cervantes, and S. Joe Qin. Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5):471–486, 2000.

[156] Sungyong Park and Chonghun Han. A nonlinear soft sensor based on multivariate smoothing procedure for quality estimation in distillation columns. *Computers and Chemical Engineering*, 24(2-7):871–877, 2000.

[157] C. T. Chou and Michel Verhaegen. Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, 33(10):1857–1869, 1997.

[158] Jin Wang and S. Joe Qin. A new subspace identification approach based on principal component analysis. *Journal of Process Control*, 12(8):841–855, 2002.

[159] Alessandro Chiuso and Giorgio Picci. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.

[160] Lennart Ljung and Tomas McKelvey. Subspace identification from closed loop data. *Signal Processing*, 52(2):209–215, 1996.

[161] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35:1215–1241, 1999.

[162] M. Gilson and P.V. den Hof. IV method for closed-loop system identification. In *Proceedings of 13th IFAC symposium on System Identification*, pages 537–542, Rotterdam, Netherlands, 2003.

[163] J. Wang and S.J. Qin. Closed loop subspace identification using the parity space. *Automatica*, 42:315–320, 2006.

[164] Biao Huang, Steven X. Ding, and S. Joe Qin. Closed-loop subspace identification: an orthogonal projection approach. *Journal of Process Control*, 15(1):53–66, 2005.

[165] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.

[166] P.A. Wisnewski and III Doyle, F.J. Model-based predictive control studies for a continuous pulp digester. *Control Systems Technology, IEEE Transactions on*, 9(3):435–444, 2001.

[167] Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. John Wiley and Sons, Hoboken, NJ, 2nd edition, 2011.

[168] Babu Joseph and Coleman B. Brosilow. Inferential control of processes: Part I. steady state analysis and design. *AIChE Journal*, 24(3):485–492, 1978.

[169] T. Tolliver and L. McCune. Distillation control design based on steady state simulation. *ISA Transactions*, 17:3–10, 1978.

[170] B. Bequette and T. Edgar. AIChE Annual Meeting, 1984.

[171] C. Georgakis and D. Kindt. AIChE Annual Meeting, 1984.

[172] Gade P. Rangaiah and Peruvemba R. Krishnaswamy. Estimating second-order plus dead time model parameters. *Industrial and Engineering Chemistry Research*, 33(7):1867–1871, 1994.

[173] G. P. Rangaiah and P. R. Krishnaswamy. Estimating second-order dead time parameters from underdamped process transients. *Chemical Engineering Science*, 51(7):1149–1155, 1996.

[174] K. Fujiwara, M. Kano, S. Hasebe, and A. Takinami. Soft-sensor development using correlation-based just-in-time modeling. *AIChE Journal*, 55(7):1754–1765, 2009.

[175] W. Larimore. *Statistical Methods in Control and Signal Processing.* Marcel Dekker, New York, 1997.

[176] J. Wang and Q. P. He. A practical solution for continuous digester control subspace identification based inferential control revisited. *Proceedings of Advanced Control of Industrial Processes-International Conference*, pages 433–438, 2008.

[177] R.E. Hodges. *Applications of near infrared spectroscopy in the pulp and paper industry.* Ph.D. dissertation, 1999.

[178] R.E. Hodges and G.A. Krishnagopalan. Near-infrared spectroscopy for on-line analysis of white and green liquors. *TAPPI Journal*, 82(9):101, 1999.

[179] Thor Mejdell and Bengt-Ove Andersson. Using temperature profile for product quality estimation on a distillation column. *ISA Transactions*, 33(1):27–34, 1994.

[180] L. Fortuna, S. Graziani, and M. G. Xibilia. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, 13(4):499–508, 2005.

[181] Manabu Kano and Yoshiaki Nakagawa. Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers and Chemical Engineering*, 32(1-2):12–24, 2008.

[182] Weiwu Yan, Huihe Shao, and Xiaofan Wang. Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers and Chemical Engineering*, 28(8):1489–1498, 2004.

[183] Feng Rui, Shen Wei, and Shao Huihe. A soft sensor modeling approach using support vector machines. In *American Control Conference, 2003. Proceedings of the 2003*, volume 5, pages 3702–3707 vol.5, 2003.

[184] Fredrik Lindgren, Paul Geladi, and Svante Wold. The kernel algorithm for PLS. *Journal of Chemometrics*, 7(1):45–59, 1993.

[185] Operator. Personal discussion with operator at MeadWestvaco Corp., 2007.

[186] M. Hubert and K. Vanden Branden. Robust methods for partial least squares regression. *Journal of Chemometrics*, 17(10):537–549, 2003.

[187] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[188] R. K. Pearson. Outliers in process modeling and identification. *Control Systems Technology, IEEE Transactions on*, 10(1):55–63, 2002.

[189] L. Davies and U. Gather. The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792, 1993.

[190] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[191] Harald Martens and Tormod Naes. *Multivariate Calibration*. John Wiley and Sons Ltd., 1989.

[192] I. N. Wakelinc and H. J. H. Macfie. A robust PLS procedure. *Journal of Chemometrics*, 6(4):189–198, 1992.

[193] David J. Cummins and C. Webster Andrews. Iteratively reweighted partial least squares: A performance analysis by Monte Carlo simulation. *Journal of Chemometrics*, 9(6):489–507, 1995.

[194] Juan A. Gil and Rosario Romera. On robust partial least squares (PLS) methods. *Journal of Chemometrics*, 12(6):365–378, 1998.

[195] Randy J. Pell. Multiple outlier detection for multivariate calibration using robust statistical techniques. *Chemometrics and Intelligent Laboratory Systems*, 52(1):87–104, 2000.

[196] R.D. Cook and S. Weisberd. *Residuals and Influence in Regression.* Chapman and Hall, London, 1982.

[197] B. Walczak and D. L. Massart. Robust principal components regression as a detection tool for outliers. *Chemometrics and Intelligent Laboratory Systems*, 27(1):41–54, 1995.

[198] G.E.P. Box. Some theorems on quadratic forms applied in the study of analysis of variance problems. *The Annals of Mathematical Statistics*, 25(2):290–302, 1954.

[199] Jin Wang and Q. Peter He. A Bayesian approach for disturbance detection and classification and its application to state estimation in run-to-run control. *IEEE Transactions on semiconductor manufacturing*, 20(2):126–136, 2007.

[200] A. Hu. *An optimal Bayesian process control for flexible manufacturing processes.* Ph.D. dissertation, 1992.

[201] H.J. Galicia, Q.P. He, and J. Wang. Statistics Pattern Analysis based virtual metrology for plasma etch processes. In *American Control Conference*, 2012. Accepted.

Appendices

Appendix A

Derivation of the linear relationship between $\mathbf{e}_k^c$ and $\mathbf{e}_k^s$

The state-space representation of the system which includes both the primary and secondary outputs is the following,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \begin{bmatrix} \mathbf{K}^s & \mathbf{K}^c \end{bmatrix} \begin{bmatrix} \mathbf{e}_k^s \\ \mathbf{e}_k^c \end{bmatrix} \tag{A.1}$$

$$\begin{bmatrix} \mathbf{y}_k^s \\ \mathbf{y}_k^c \end{bmatrix} = \begin{bmatrix} \mathbf{C}^s \\ \mathbf{C}^c \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{e}_k^s \\ \mathbf{e}_k^c \end{bmatrix} \tag{A.2}$$

Since we assume that the secondary output variables and the input variables can completely describe the system dynamics, the system can also be described by the following state-space representation,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}\mathbf{e}_k^s \tag{A.3}$$

$$\mathbf{y}_k^s = \mathbf{C}^s\mathbf{x}_k + \mathbf{e}_k^s \tag{A.4}$$

By comparing Eqn. (A.1) and Eqn.(A.3), we can see that

$$\begin{bmatrix} \mathbf{K}^s & \mathbf{K}^c \end{bmatrix} \begin{bmatrix} \mathbf{e}_k^s \\ \mathbf{e}_k^c \end{bmatrix} = \mathbf{K}\mathbf{e}_k^s \tag{A.5}$$

By rearranging the above equation we have

$$\begin{aligned}\mathbf{e}_k^c &= \left(\mathbf{K}^c\right)^\dagger \left(\mathbf{K} - \mathbf{K}^s\right) \mathbf{e}_k^s \\ &= F\mathbf{e}_k^s \qquad (A.6)\end{aligned}$$

where $\left(\mathbf{K}^c\right)^\dagger$ denotes the pseudo-inverse of $\mathbf{K}^c$. Eqn. (A.6) shows that $\mathbf{e}_k^c$ and $\mathbf{e}_k^s$ are linearly correlated under the assumption that $(\mathbf{A}, \mathbf{C}^s)$ is observable.

Appendix B

Detailed evaluation of $E\left\{\mathbf{e}_f^c(k)\left[[\mathbf{z}_p(k)]^T[\mathbf{z}_{f-1}(k)]^T\right]\right\}$

$$
\begin{aligned}
& E\left\{\mathbf{e}_f^c(k)\begin{bmatrix}\mathbf{z}_p(k)\\ \mathbf{z}_{f-1}(k)\end{bmatrix}^T\right\}\\
= & \lim_{n\to\infty}\frac{1}{n}\left\{\begin{bmatrix}
\mathbf{e}_k^c & \mathbf{e}_{k+1}^c & \cdots & \mathbf{e}_{k+n-1}^c\\
\mathbf{e}_{k+1}^c & \mathbf{e}_{k+2}^c & \cdots & \mathbf{e}_{k+n}^c\\
\vdots & \vdots & \cdots & \vdots\\
\mathbf{e}_{k+f-1}^c & \mathbf{e}_{k+f}^c & \cdots & \mathbf{e}_{k+f+n-2}^c
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_{k-p} & \mathbf{u}_{k-p+1} & \cdots & \mathbf{u}_{k-p+n-1}\\
\mathbf{y}_{k-p}^s & \mathbf{y}_{k-p+1}^s & \cdots & \mathbf{y}_{k-p+n-1}^s\\
\vdots & \vdots & \cdots & \vdots\\
\mathbf{u}_{k} & \mathbf{u}_{k+1} & \cdots & \mathbf{u}_{k+n-1}\\
\mathbf{y}_{k}^s & \mathbf{y}_{k+1}^s & \cdots & \mathbf{y}_{k+n-1}^s\\
\vdots & \vdots & \cdots & \vdots\\
\mathbf{u}_{k+f-2} & \mathbf{u}_{k+f-1} & \cdots & \mathbf{u}_{k+f+n-3}\\
\mathbf{y}_{k+f-2}^s & \mathbf{y}_{k+f-1}^s & \cdots & \mathbf{y}_{k+f+n-3}^s
\end{bmatrix}^T\right\}
\end{aligned}
\tag{B.1}
$$

$$
\begin{aligned}
&= E\begin{bmatrix}
\mathbf{e}_k^c\mathbf{u}_{k-p} & \mathbf{e}_{k+1}^c\mathbf{u}_{k-p} & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{u}_{k-p} \\
\mathbf{e}_k^c\mathbf{y}_{k-p}^s & \mathbf{e}_{k+1}^c\mathbf{y}_{k-p}^s & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{y}_{k-p}^s \\
\vdots & \vdots & \cdots & \vdots \\
\mathbf{e}_k^c\mathbf{u}_{k} & \mathbf{e}_{k+1}^c\mathbf{u}_{k} & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{u}_{k} \\
\mathbf{e}_k^c\mathbf{y}_{k}^s & \mathbf{e}_{k+1}^c\mathbf{y}_{k}^s & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{y}_{k}^s \\
\mathbf{e}_k^c\mathbf{u}_{k+1} & \mathbf{e}_{k+1}^c\mathbf{u}_{k+1} & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{u}_{k+1} \\
\mathbf{e}_k^c\mathbf{y}_{k+1}^s & \mathbf{e}_{k+1}^c\mathbf{y}_{k+1}^s & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{y}_{k+1}^s \\
\vdots & \vdots & \cdots & \vdots \\
\mathbf{e}_k^c\mathbf{u}_{k+f-2} & \mathbf{e}_{k+1}^c\mathbf{u}_{k+f-2} & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{u}_{k+f-2} \\
\mathbf{e}_k^c\mathbf{y}_{k+f-2}^s & \mathbf{e}_{k+1}^c\mathbf{y}_{k+f-2}^s & \cdots & \mathbf{e}_{k+f-1}^c\mathbf{y}_{k+f-2}^s
\end{bmatrix}^T \\
&= \begin{bmatrix}
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \cdots & \vdots \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
E\left\{\mathbf{e}_k^c\mathbf{u}_{k+1}\right\} & 0 & \cdots & 0 \\
E\left\{\mathbf{e}_k^c\mathbf{y}_{k+1}^s\right\} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
E\left\{\mathbf{e}_k^c\mathbf{u}_{k+f-2}\right\} & E\left\{\mathbf{e}_{k+1}^c\mathbf{u}_{k+f-2}\right\} & \cdots & 0 \\
E\left\{\mathbf{e}_k^c\mathbf{y}_{k+f-2}^s\right\} & E\left\{\mathbf{e}_{k+1}^c\mathbf{y}_{k+f-2}^s\right\} & \cdots & 0
\end{bmatrix}^T \neq \mathbf{0} \qquad \text{(B.2)}
\end{aligned}
$$

It can be seen from the above derivation that the lower triangular part of the matrix is non-zero due to feedback control, which is the reason why the linear relationship Eqn. (6.9) does not provide unbiased estimate of $\mathbf{y}_f^c(k)$ under closed-loop control when $f > 1$.

Appendix C

Detailed derivation of Equation (5.13) and (5.14) for mean and variance recursive update.

For variable mean, by plugging the definition of variable mean to the right hand side (RHS) of Equation (5.13):

$$\begin{aligned} RHS &= \frac{n-1}{n}\left(\frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right) + \frac{1}{n}x_n \\ RHS &= \frac{\sum_{i=1}^{n} x_i}{n} = \overline{x}_n = LHS \qquad \text{(C.1)} \end{aligned}$$

where LHS denotes the left hand side of Equation (5.13).

For variable variance in Equation (5.14):

$$
\begin{aligned}
RHS &= \frac{n-2}{n-1}\left[\frac{1}{n-2}\sum_{i=1}^{n-1}\left(x_i-\overline{x}_{n-1}\right)^2\right] \\
&\quad +\frac{\left(x_n-\overline{x}_n\right)^2}{n-1}+\left(\overline{x}_n-\overline{x}_{n-1}\right)^2 \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n-1}\left(x_i-\overline{x}_n+\overline{x}_n-\overline{x}_{n-1}\right)^2+\left(x_n-\overline{x}_n\right)^2\right] \\
&\quad +(\overline{x}_n-\overline{x}_{n-1})^2 \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n}(x_i-\overline{x}_n)^2+(n-1)(\overline{x}_n-\overline{x}_{n-1})^2\right. \\
&\quad \left.+2(\overline{x}_n-\overline{x}_{n-1})\sum_{i=1}^{n-1}(x_i-\overline{x}_n)\right]+(\overline{x}_n-\overline{x}_{n-1})^2 \\
&= \sigma_n^2+(\overline{x}_n-\overline{x}_{n-1})^2+\frac{2}{n-1}(\overline{x}_n-\overline{x}_{n-1}) \\
&\quad \left[(n-1)\left(\overline{x}_{n-1}-\overline{x}_n\right)\right]+(\overline{x}_n-\overline{x}_{n-1})^2 \\
&= \sigma_n^2+2(\overline{x}_n-\overline{x}_{n-1})^2-2(\overline{x}_n-\overline{x}_{n-1})^2 \\
&= \sigma_n^2 \\
&= LHS \qquad \text{(C.2)}
\end{aligned}
$$

Appendix D

Derivation of Equation (5.16).

$$\begin{aligned}
\overline{x}_{n-1} &= \frac{n}{n-1}\overline{x}_n - \frac{1}{n-1}x_n \\
\left(\overline{x}_n - \overline{x}_{n-1}\right)^2 &= \left(\overline{x}_n - \frac{n}{n-1}\overline{x}_n + \frac{1}{n-1}x_n\right)^2 \\
\left(\overline{x}_n - \overline{x}_{n-1}\right)^2 &= \left[\frac{1}{n-1}\left(x_n - \overline{x}_n\right)\right]^2
\end{aligned}$$

Therefore,

$$\frac{\left(\overline{x}_n - \overline{x}_{n-1}\right)^2}{\frac{1}{n-1}\left(x_n - \overline{x}_n\right)^2} = \frac{1}{n-1} \tag{D.1}$$

Appendix E

Detailed derivation of Equation (5.17) and (5.18) for block mean and variance recursive update.

For variable mean in Equation (5.17):

$$\begin{aligned} RHS &= \frac{(n-n_1)}{n}\frac{\sum_{i=1}^{n-n_1} x_i}{n-n_1} + \frac{n_1}{n}\frac{\sum_{i=n-n_1+1}^{n} x_i}{n_1} \\ &= \frac{1}{n}\left[\sum_{i=1}^{n-n_1} x_i + \sum_{i=n-n_1+1}^{n} x_i\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{x}_n = LHS \qquad \text{(E.1)} \end{aligned}$$

For variable variance in Equation (5.18):

$$\begin{aligned}
RHS &= \frac{n-n_1-1}{n-1}\sigma^2_{n-n_1} + \frac{n_1-1}{n-1}\sigma^2_{n,B} \\
&\quad + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n-n_1}\left(x_i-\overline{x}_{n-n_1}\right)^2 + \sum_{i=n-n_1+1}^{n}\left(x_i-\overline{x}_{n,B}\right)^2\right] \\
&\quad + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n-n_1}\left(x_i-\overline{x}_n+\overline{x}_n-\overline{x}_{n-n_1}\right)^2 + \sum_{i=n-n_1+1}^{n}\left(x_i-\overline{x}_n+\overline{x}_n-\overline{x}_{n,B}\right)^2\right] \\
&\quad + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \frac{1}{n-1}\left[\sum_{i=1}^{n}\left(x_i-\overline{x}_n\right)^2 + \left(n-n_1\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right)^2\right. \\
&\quad +2\sum_{i=1}^{n-n_1}\left(x_i-\overline{x}_n\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right) + n_1\left(\overline{x}_n-\overline{x}_{n,B}\right)^2 \\
&\quad \left.+2\sum_{i=n-n_1+1}^{n}\left(x_i-\overline{x}_n\right)\left(\overline{x}_n-\overline{x}_{n,B}\right)\right] + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \sigma^2_n + \frac{1}{n-1}\left[\left(n-n_1\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right)^2 - 2\left(n-n_1\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right)^2\right. \\
&\quad \left.+n_1\left(\overline{x}_n-\overline{x}_{n,B}\right)^2 - 2n_1\left(\overline{x}_n-\overline{x}_{n,B}\right)^2\right] + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \sigma^2_n - \frac{1}{n-1}\left[\left(n-n_1\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right)^2 + n_1\left(\overline{x}_n-\overline{x}_{n,B}\right)^2\right] \\
&\quad + \frac{n_1\left(n-n_1\right)}{n\left(n-1\right)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \qquad \text{(E.2)}
\end{aligned}$$

The first and second terms inside the bracket in Equation (E.2) can be rewritten using Equation (5.17) as follows:

$$\left(n-n_1\right)\left(\overline{x}_n-\overline{x}_{n-n_1}\right)^2 = \left(n-n_1\right)\left[\frac{n_1}{n}\overline{x}_{n,B}-\frac{n_1}{n}\overline{x}_{n-n_1}\right]^2 \qquad \text{(E.3)}$$

$$n_1\left(\overline{x}_n-\overline{x}_{n,B}\right)^2 = n_1\left[\frac{n-n_1}{n}\overline{x}_{n-n_1}-\frac{n-n_1}{n}\overline{x}_{n,B}\right]^2 \qquad \text{(E.4)}$$

Therefore, Equation (E.2) can be rearranged as follows:

$$\begin{aligned}
RHS &= \sigma_n^2 - \frac{1}{n-1}\left[\frac{(n-n_1)\,n_1^2}{n^2}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 + \frac{n_1\,(n-n_1)^2}{n^2}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2\right] \\
&\quad + \frac{n_1\,(n-n_1)}{n\,(n-1)}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
&= \sigma_n^2 - \frac{1}{n-1}\frac{n_1\,(n-n_1)}{n^2}\left[n_1+(n-n_1)\right]\left[\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2\right] \\
&\quad + \frac{n_1\,(n-n_1)^2}{n^2}\left(\overline{x}_{n-n_1}-\overline{x}_{n,B}\right)^2 \\
RHS &= \sigma_n^2 = LHS \qquad \text{(E.5)}
\end{aligned}$$