

Intelligent Control and Learning Systems 3

Jing Wang
Jinglin Zhou
Xiaolu Chen

Data-Driven Fault Detection and Reasoning for Industrial Monitoring


OPEN ACCESS

 Springer

Intelligent Control and Learning Systems

Volume 3

Series Editor

Dong Shen , School of Mathematics, Renmin University of China, Beijing,
Beijing, China

The Springer book series Intelligent Control and Learning Systems addresses the emerging advances in intelligent control and learning systems from both mathematical theory and engineering application perspectives. It is a series of monographs and contributed volumes focusing on the in-depth exploration of learning theory in control such as iterative learning, machine learning, deep learning, and others sharing the learning concept, and their corresponding intelligent system frameworks in engineering applications. This series is featured by the comprehensive understanding and practical application of learning mechanisms. This book series involves applications in industrial engineering, control engineering, and material engineering, etc.

The Intelligent Control and Learning System book series promotes the exchange of emerging theory and technology of intelligent control and learning systems between academia and industry. It aims to provide a timely reflection of the advances in intelligent control and learning systems. This book series is distinguished by the combination of the system theory and emerging topics such as machine learning, artificial intelligence, and big data. As a collection, this book series provides valuable resources to a wide audience in academia, the engineering research community, industry and anyone else looking to expand their knowledge in intelligent control and learning systems.


More information about this series at <https://link.springer.com/bookseries/16445>

Jing Wang · Jinglin Zhou · Xiaolu Chen

Data-Driven Fault Detection and Reasoning for Industrial Monitoring

 Springer

Jing Wang 
Department of Automation, School
of Electrical and Control Engineering
North China University of Technology
Beijing, China

Jinglin Zhou 
College of Information Science
and Technology
Beijing University of Chemical Technology
Beijing, China

Xiaolu Chen 
College of Engineering
Peking University
Beijing, China



ISSN 2662-5458 ISSN 2662-5466 (electronic)
Intelligent Control and Learning Systems
ISBN 978-981-16-8043-4 ISBN 978-981-16-8044-1 (eBook)
<https://doi.org/10.1007/978-981-16-8044-1>

© The Editor(s) (if applicable) and The Author(s) 2022. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

After decades of development, automation and intelligence increase significantly in the process industry, and key technologies continue to make breakthroughs. In the era of “New Industrial Revolution”, it is of great significance to use modern information technology to promote intelligent manufacturing with the goal of safety, efficiency, and green. Obviously, safety has always been the lifeline of intelligent and optimized manufacturing in process industries.

With the increasing requirements for production safety and quality improvement, process monitoring and fault diagnosis have gained great attention in academic research and even in industrial applications. The widespread use of sensor networks and distributed control systems have facilitated access to a wealth of process data. How to effectively use the data generated during the production process and the process mechanism knowledge for process monitoring and fault diagnosis is a topic worth exploring for the large and complex process industrial systems. Fruitful academic results have been produced recently and widely used in the actual production process.

The authors of this book have devoted themselves to the theoretical and applied research work on data-driven industrial process monitoring and fault diagnosis methods for many years. They are deeply concerned with the flourishing development of data-driven fault diagnosis techniques. This book focuses on both multivariate statistical process monitoring (MSPM) and Bayesian inference diagnosis. It introduces the basic multivariate statistical modeling methods, as well as the authors’ latest achievements around the practical industrial needs, including multi-transition process monitoring, fault classification and identification, quality-related fault detection, and fault root tracing.

The main contributions given in this book are as follows:

(1) Soft-transition-based high-precision monitoring for multi-stage batch processes: Most batch processes obviously have several operation stages with different process characteristics. In addition, their data present obvious three-dimensional features with strong nonlinearity and time variability. So it is difficult to apply multivariate statistical methods directly to the monitoring of batch processes. This book proposes a soft-transition-based fault detection method. First, a two-step

stage division method based on Support Vector Data Description (SVDD) is given, then a dynamic soft-transition model of transition stages is constructed; finally, the monitoring in the original measurement space is given for each stage. Compared with the traditional method, the advantages of the proposed method are reflected in the following techniques: improvement of soft-transition process design, statistic decomposition, and fusion indicator monitoring. It just greatly increase the accuracy of batch process fault detection.

(2) Fault classification and identification for batch process with variable production cycle: Batch processes inevitably are subject to the changes in initial conditions and the external environment, which can cause changes in production cycles. However, current monitoring methods for batch processes generally require the equal production cycle and a complete production trajectory. Therefore, variable cycle and unknown values estimation in complete trajectory become the bottleneck for improving the diagnostic performance. This book gives a fault diagnosis method for batch processes based on kernel Fisher envelope analysis. It builds envelope surface models for normal conditions and all known fault condition, respectively. Then online fault diagnosis strategy is proposed based on these surface models. Further, the fusion of kernel Fisher envelope analysis and PCA is proposed for fault diagnosis of batch process. It effectively solves the fault classification and identification of unequal-length batch production process.

(3) Quality-related fault detection with fusion of global and local features: The key of manufacture is to guarantee the final product quality, yet it is difficulty or extreme cost to acquire quality information in real time. Therefore, it is great practical to monitor the process variables that have an impact on the final quality output in order to further enable quality-related fault detection and diagnosis. This book proposes an idea of quality-related projection with the fusion of global and local features to obtain the correlation between quality variables and process variables. It is well known that the partial least squares projection algorithm looks for global structural change information based on the process covariance maximization direction. The local preservation projection, or manifold learning approach can exactly maintain the local neighborhood structure and achieve nonlinear mapping by using linear approximation. The proposed fusion approach constructs potential geometric structures containing both global and local information, extracts meaningful low-dimensional structural information to represent the relationship between high-dimensional process variables and quality data. Thus, it effectively achieves the detection of quality-related faults for strongly nonlinear and strongly dynamic processes.

(4) Bayesian fault diagnosis and root tracing combined with process mechanism: Due to the complex interrelationships among system components, same fault source may have different manifestation signs in the different process variables. The traditional contribution graph in multivariate statistical monitoring is inefficiency in fault root tracing. This book proposes an uncertainty knowledge expression inference model, named probabilistic causal graph model, based on probability theory and graph theory. It intuitively and accurately reveals the qualitative and quantitative relationships between process variables. Then a framework for fault diagnosis and

root tracking based on the proposed model is given. Different modeling and inference techniques are given for the discrete and continuous system, respectively. So, the inference can perform real-time dynamic analysis of discrete alarm states or continuous process variables. The forward inference predicts the univariate and multivariate alarms or fault events, while the reverse implements the accurate fault root tracing and localization.

The book consists of 14 chapters divided into four parts:

Part I, Chaps. 1–4, is devoted to mathematical background. Chapter 1 gives the basic knowledge about process monitoring measure, common detection indicator, and its control limit. Chapters 2–3 focus on the basic multivariate statistical methods, including principal element analysis (PCA), partial least squares (PLS), canonical correlation analysis (CCA), canonical variable analysis (CVA), and Fisher discriminant analysis (FDA). To help readers learn the above theoretical methods, Chap. 4 gives a detailed introduction to the Tennessee Eastman (TE) continuous chemical simulation platform and the penicillin semi-batch reaction simulation platform. Readers can collect appropriate process data and conduct corresponding simulation experiments on these simulation platform.

Part II, Chaps. 5–8, are organized around the main contributions 1 and 2 of this book. Various improved fault detection and identification methods are given for batch process. Chapters 5–6 are given for contribution 1 aiming at the high-precision process monitoring of with many stages process, based on Support Vector Data Description (SVDD) soft-transition process, and fusion index design based on statistics decomposition. Chapters 7–8 are given for contribution 2 aiming at the fault identification for complex batch process with unequal cycle, based on the kernel Fisher envelope surface analysis and local linear embedded Fisher discriminant analysis, respectively.

Part III, Chaps. 9–12, are organized around the main contribution 3 of this book. To improve the statistical model between process variables and quality variables with nonlinear correlation, two different strategies are considered. First, under the idea of global and local feature fusion, the manifold structure are considered to extract the nonlinear correlations between them effectively. A unified framework of spatial optimization projection is constructed based on the effective fusion of two types of performance indices, global covariance maximization and local adjacency structure minimization. A variety of different performance combinations are given in Chaps. 9–11: QGLPLS, LPPLS and LLEPLS, respectively. Another strategy is to consider the nonlinearity as uncertainty, then robust L_1 -PLS is proposed in Chap. 12. It enhances the robustness of PLS method based on the latent structure regression with L_1 . The effectiveness and applicability of the above combination methods are discussed.

Part IV, Chaps. 13–14, are organized around the main contribution 4 of the book. The known industrial process flow structure is integrated with the industrial data analytic, and the qualitative causal relationships among process variables are established by multivariate causal analysis methods. The quantitative causal dependencies among process variables are characterized by conditional probability density estimation under this network structure. So, Bayesian causal probability graph model

of complex systems is realized for process variable failure prediction and reverse tracing. The specific implementation of the Bayesian inference, respectively, in discrete alarm variable analysis and continuous process variable analysis are given in this book.

Fault detection and diagnosis (FDD) is one of the core topics in modern complex industrial processes. It attracts the attention of scientists and engineers from various fields such as control, mechanics, mathematics, engineering, and automation. This book gives an in-depth study of various data-driven analysis methods and their applications in process monitoring, especially for data modeling, fault detection, fault classification, fault identification, and fault reasoning. Oriented toward the industrial big data analytic and industrial artificial intelligence, this book integrates multivariate statistical analysis, Bayesian inference, machine learning, and other intelligent analysis methods. This book attempts to establish a basic framework of complex industrial process monitoring suitable for various types of industrial data processing, and gives a variety of fault detection and diagnosis theories, methods, algorithms, and various applications. It provides data-driven fault diagnosis techniques of interest to advanced undergraduate and graduate students, researchers in the direction of automation and industrial safety. It also provides various applications of engineering modeling, data analysis, and processing methods for related practitioners and engineers.

Beijing, China
August 2021

Jing Wang
jwang@ncut.edu.cn

Jinglin Zhou
jinglinzhou@mail.buct.edu.cn

Xiaolu Chen
chenxiaolu@pku.edu.cn

Acknowledgements The authors thank National Natural Science Foundation of China (Grant No. 61573050, 61973023, 62073023, and 61473025) for funding the research over the past 8 years. The authors also thank Li Liu, Huatong Wei, Bin Zhong, Ruixuan Wang, and Shunli Zhang, the graduate students from the College of Information Science and Technology, Beijing University of Chemical Technology, for the hard work in system design and programming.

Contents

1	Background	1
1.1	Introduction	1
1.1.1	Process Monitoring Method	1
1.1.2	Statistical Process Monitoring	3
1.2	Fault Detection Index	5
1.2.1	T ² Statistic	6
1.2.2	Squared Prediction Error	9
1.2.3	Mahalanobis Distance	10
1.2.4	Combined Indices	11
1.2.5	Control Limits in Non-Gaussian Distribution	12
	References	13
2	Multivariate Statistics in Single Observation Space	17
2.1	Principal Component Analysis	18
2.1.1	Mathematical Principle of PCA	18
2.1.2	PCA Component Extraction Algorithm	20
2.1.3	PCA Base Fault Detection	22
2.2	Fisher Discriminant Analysis	23
2.2.1	Principle of FDA	24
2.2.2	Comparison of FDA and PCA	28
	References	29
3	Multivariate Statistics Between Two-Observation Spaces	31
3.1	Canonical Correlation Analysis	31
3.1.1	Mathematical Principle of CCA	32
3.1.2	Eigenvalue Decomposition of CCA Algorithm	34
3.1.3	SVD Solution of CCA Algorithm	34
3.1.4	CCA-Based Fault Detection	35

- 3.2 Partial Least Squares 36
 - 3.2.1 Fundamental of PLS 37
 - 3.2.2 PLS Algorithm 37
 - 3.2.3 Cross-Validation Test 41
- References 43
- 4 Simulation Platform for Fault Diagnosis 45**
 - 4.1 Tennessee Eastman Process 45
 - 4.2 Fed-Batch Penicillin Fermentation Process 49
 - 4.3 Fault Detection Based on PCA, CCA, and PLS 50
 - 4.4 Fault Classification Based on FDA 51
 - 4.5 Conclusions 58
- References 58
- 5 Soft-Transition Sub-PCA Monitoring of Batch Processes 59**
 - 5.1 What Is Phase-Based Sub-PCA 60
 - 5.2 SVDD-Based Soft-Transition Sub-PCA 62
 - 5.2.1 Rough Stage-Division Based on Extended Loading Matrix 62
 - 5.2.2 Detailed Stage-Division Based on SVDD 63
 - 5.2.3 PCA Modeling for Transition Stage 65
 - 5.2.4 Monitoring Procedure of Soft-Transition Sub-PCA 67
 - 5.3 Case Study 69
 - 5.3.1 Stage Identification and Modeling 69
 - 5.3.2 Monitoring of Normal Batch 71
 - 5.3.3 Monitoring of Fault Batch 71
 - 5.4 Conclusions 75
- References 76
- 6 Statistics Decomposition and Monitoring in Original Variable Space 79**
 - 6.1 Two Statistics Decomposition 80
 - 6.1.1 T^2 Statistic Decomposition 80
 - 6.1.2 SPE Statistic Decomposition 81
 - 6.1.3 Fault Diagnosis in Original Variable Space 82
 - 6.2 Combined Index-Based Fault Diagnosis 84
 - 6.2.1 Combined Index Design 84
 - 6.2.2 Control Limit of Combined Index 87
 - 6.3 Case Study 88
 - 6.3.1 Variable Monitoring via Two Statistics Decomposition 88
 - 6.3.2 Combined Index-Based Monitoring 93
 - 6.3.3 Comparative Analysis 97
 - 6.4 Conclusions 99
- References 100

- 7 Kernel Fisher Envelope Surface for Pattern Recognition** 101
 - 7.1 Process Monitoring Based on Kernel Fisher Envelope Analysis 101
 - 7.1.1 Kernel Fisher Envelope Surface 101
 - 7.1.2 Detection Indicator 104
 - 7.1.3 KFES-PCA-Based Synthetic Diagnosis in Batch Process 106
 - 7.2 Simulation Experiment Based on KFES-PCA 108
 - 7.2.1 Diagnostic Effect on Existing Fault Types 109
 - 7.2.2 Diagnostic Effect on Unknown Fault Types 114
 - 7.3 Conclusions 116
 - References 117
- 8 Fault Identification Based on Local Feature Correlation** 119
 - 8.1 Fault Identification Based on Kernel Discriminant Exponent Analysis 120
 - 8.1.1 Methodology of KEDA 120
 - 8.1.2 Simulation Experiment 122
 - 8.2 Fault Identification Based on LLE and EDA 127
 - 8.2.1 Local Linear Exponential Discriminant Analysis 128
 - 8.2.2 Neighborhood-Preserving Embedding Discriminant Analysis 130
 - 8.2.3 Fault Identification Based on LLEDA and NPEDA 133
 - 8.2.4 Simulation Experiment 134
 - 8.3 Cluster-LLEDA-Based Hybrid Fault Monitoring 135
 - 8.3.1 Hybrid Monitoring Strategy 135
 - 8.3.2 Simulation Study 140
 - 8.4 Conclusion 145
 - Reference 146
- 9 Global Plus Local Projection to Latent Structures** 147
 - 9.1 Fusion Motivation of Global Structure and Local Structure 148
 - 9.2 Mathematical Description of Dimensionality Reduction 150
 - 9.2.1 PLS Optimization Objective 150
 - 9.2.2 LPP and PCA Optimization Objectives 151
 - 9.3 Introduction to the GPLPLS 152
 - 9.4 Basic Principles of GPLPLS 154
 - 9.4.1 The GPLPLS Model 154
 - 9.4.2 Relationship Between GPLPLS Models 156
 - 9.4.3 Principal Components of the GPLPLS Model 157
 - 9.5 GPLPLS-Based Quality Monitoring 159
 - 9.5.1 Process and Quality Monitoring Based on GPLPLS 159
 - 9.5.2 Posterior Monitoring and Evaluation 161
 - 9.6 TE Process Simulation Analysis 162
 - 9.6.1 Model and Discussion 162
 - 9.6.2 Fault Diagnosis Analysis 163

- 9.6.3 Comparison of Different GPLPLS Models 167
- 9.7 Conclusions 170
- References 170
- 10 Locality-Preserving Partial Least Squares Regression 173**
 - 10.1 The Relationship Among PCA, PLS, and LPP 173
 - 10.2 LPPLS Models and LPPLS-Based Fault Detection 175
 - 10.2.1 The LPPLS Models 175
 - 10.2.2 LPPLS for Process and Quality Monitoring 179
 - 10.2.3 Locality-Preserving Capacity Analysis 181
 - 10.3 Case Study 182
 - 10.3.1 PLS, GPLS and LPPLS Models 182
 - 10.3.2 Quality Monitoring Analysis 184
 - 10.4 Conclusions 187
 - References 188
- 11 Locally Linear Embedding Orthogonal Projection to Latent Structure 189**
 - 11.1 Comparison of GPLPLS, LPPLS, and LLEPLS 190
 - 11.2 A Brief Review of the LLE Method 191
 - 11.3 LLEPLS Models and LLEPLS-Based Fault Detection 194
 - 11.3.1 LLEPLS Models 194
 - 11.3.2 LLEPLS for Process and Quality Monitoring 196
 - 11.4 LLEOPLS Models and LLEOPLS-Based Fault Detection 198
 - 11.5 Case Study 201
 - 11.5.1 Models and Discussion 201
 - 11.5.2 Fault Detection Analysis 203
 - 11.6 Conclusions 207
 - References 208
- 12 New Robust Projection to Latent Structure 211**
 - 12.1 Motivation of Robust L_1 -PLS 211
 - 12.2 Introduction to RSPCA Method 213
 - 12.3 Basic Principle of L_1 -PLS 214
 - 12.4 L_1 -PLS-Based Process Monitoring 217
 - 12.5 TE Simulation Analysis 219
 - 12.5.1 Robustness of Principal Components 220
 - 12.5.2 Robustness of Prediction and Monitoring Performance 223
 - 12.6 Conclusions 231
 - References 231
- 13 Bayesian Causal Network for Discrete Variables 233**
 - 13.1 Construction of Bayesian Causal Network 234
 - 13.1.1 Description of Bayesian Network 234
 - 13.1.2 Establishing Multivariate Causal Structure 235
 - 13.1.3 Network Parameter Learning 238

- 13.2 BCN-Based Fault Detection and Inference 239
- 13.3 Case Study 241
 - 13.3.1 Public Data Sets Experiment 241
 - 13.3.2 TE Process Experiment 243
- 13.4 Conclusions 248
- References 248
- 14 Probabilistic Graphical Model for Continuous Variables 251**
 - 14.1 Construction of Probabilistic Graphical Model 251
 - 14.1.1 Multivariate Casual Structure Learning 251
 - 14.1.2 Probability Density Estimation 252
 - 14.1.3 Evaluation Index of Estimation Quality 254
 - 14.2 Dynamic Threshold for the Fault Detection 256
 - 14.3 Forward Fault Diagnosis and Reverse Reasoning 258
 - 14.4 Case Study: Application to TEP 260
 - 14.5 Conclusions 265
 - References 265

Abbreviations

BCN	Bayesian causal network
BN	Bayesian network
CCA	Canonical correlation analysis
CDC	Complete decomposition contributions
CPLS	Concurrent projection to the latent structures
CPV	Cumulative percent variance
CVA	Canonical variables analysis
DAG	Directed acyclic graph
DCS	Distributed control system
EM	Expectation maximization
FA	False alarm
FAR	False alarm rate
FCR	Fault recognition rate
FDA	Fisher discriminant analysis
FDD	Fault detection and diagnosis
FDR	Fault detection rate
FLSA	Fused Lasso Signal Approximator
GLPLS	Global and local partial least squares
GPLPLS	Global plus local projection to latent structure
HMM	Hidden Markov model
ICA	Independent component analysis
JT	Junction tree
KCCCA	Kernel concurrent canonical correlation analysis
K-CV	K -fold cross-validation
KDE	Kernel density estimation
KEDA	Kernel exponential discriminant analysis
KFDA	Kernel Fisher discriminant analysis
KFEA	Kernel Fisher envelope analysis
KFES	Kernel Fisher envelope surface
KICA	Kernel independent component analysis
KNN	K nearest neighbors

KPCA	Kernel principal component analysis
KPLS	Kernel partial least squares
L_1 -PLS	Projection to latent structure based on the L_1 norm
LAD	Least absolute deviation
LiNGAM	Linear non-Gaussian acyclic model
LLE	Locally linear embedding
LLEDA	Local linear exponential discriminant analysis
LLEOPLS	Local linear embedded orthogonal projection to latent structure
LLEPLS	Local linear embedded projection of latent structure
LOO-CV	Leave-one-out cross-validation
LPP	Locality preserving projections
LPPLS	Locality preserving partial least squares
LS	Least squares
LWPR	Locally weighted projection regression
MCD	Minimum covariance determinant
MISE	Mean integral square error
MLE	Maximum likelihood estimation
MLP	Multi-layer perceptron
NIPALS	Nonlinear iterative partial least squares
NNPLS	Neural network partial least squares
NPEDA	Neighbourhood preserving embedding discriminant analysis
NPLSSLT	Nonlinear partial least squares based on slice transformation
O-NLPLS	Orthogonal nonlinear partial least squares
PC	Principal component
PCA	Principal component analysis
PCS	Principal component subspace
PDC	Partial decomposition contributions
PLS	Partial least squares
PNL	Post-nonlinear
PMA	Posterior monitoring assessment
PQAR	Posterior quality alarm rate
QPLS	Quadratic partial least squares
RBC	Reconstruction based contributions
RCP	Relative contribution plot
RMCD	Reweighted minimum covariance determinant
RNPLS	Recursive nonlinear partial least squares
RPCA	Robust principal component analysis
RRC	Relative rates of change
RS	Residual subspace
RSPCA	Robust sparse principal component analysis
SPCA	Spare principal component analysis
SPE	Squared prediction Error
SPLS	Spline function partial least squares
SPM	Statistical process monitoring
SVD	Singular value decomposition

SVDD	Support vector data description
TD	Time detected
TE	Tennessee Eastman
VIP	Variable importance in prediction

Chapter 1

Background



1.1 Introduction

Fault detection and diagnosis (FDD) technology is a scientific field emerged in the middle of the twentieth century with the rapid development of science and data technology. It manifests itself as the accurate sensing of abnormalities in the manufacturing process, or the health monitoring of equipment, sites, or machinery in a specific operating site. FDD includes abnormality monitoring, abnormal cause identification, and root cause location. Through qualitative and quantitative analysis of field process and historical data, operators and managers can detect alarms that affect product quality or cause major industrial accidents. It is help for cutting off failure paths and repairing abnormalities in a timely manner.

1.1.1 Process Monitoring Method

In general, FDD technique is divided into several parts: fault detection, fault isolation, fault identification, and fault diagnosis (Hwang et al. 2010; Zhou and Hu 2009). Fault detection is determining of the appearance of fault. Once a fault (or error) has been successfully detected, damage assessment needs to be performed, i.e., fault isolation (Yang et al. 2006). Fault isolation lies in determining the type, location, magnitude, and time of the fault (i.e., the observed out-of-threshold variables). It should be noted that fault isolation is not to isolation of specific components of a system with the purpose of stopping errors from propagating. In a sense, fault identification may have been a better choice. It also has the ability to determine its timely change. Isolation and identification are commonly used in the FDD process without strict distinction. Fault diagnosis determines the cause of the observed out-of-threshold variables in this book, so it is called as fault root tracing. During the process of fault tracing, efforts are made to locate the source of the fault and find the root cause.

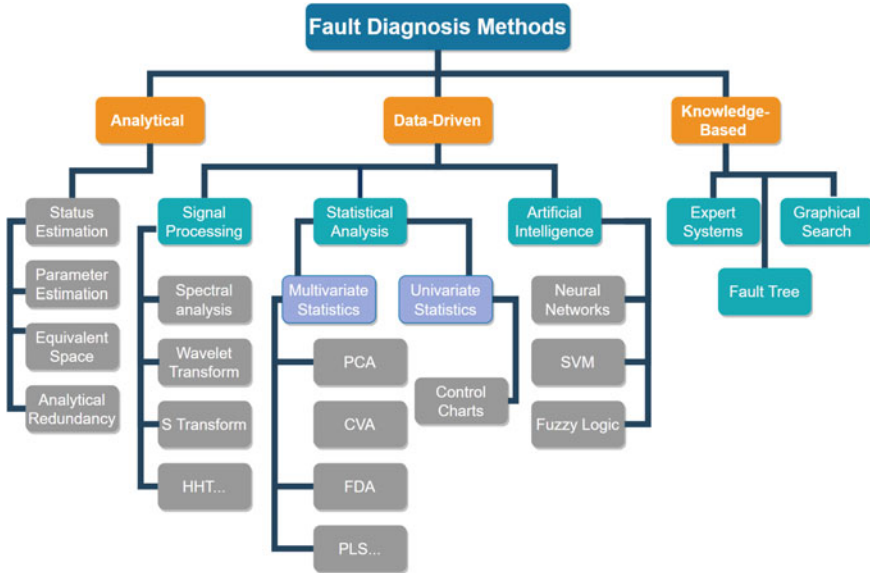


Fig. 1.1 Classification of fault diagnosis methods

FDD involves control theory, probability statistics, signal processing, machine learning, and many other research areas. Many effective methods have been developed, and they are usually classified into three categories, knowledge-based, analytical, and data-driven (Chiang et al. 2001). Figure 1.1 shows the classification of fault diagnosis methods.

(1) Analytical Method

The analytical model of the engineering system is obtained based on the mathematical and physical mechanism. Analytical model-based method represents to monitor the process real time according to the mathematical models often constructed from first principles and physical characteristics. Most analytical measures contain state estimation (Wang et al. 2020), parameter estimation (Yu 1997), parity space (Ding 2013), and analytical redundancy (Suzuki et al. 1999). The analytical method appears to be relatively simple and usually is applied to systems with a relatively small number of inputs, outputs, and states. It is impractical for modern complex system since it is not easy to establish an accurate mathematical model due to its complex characteristics such as nonlinearity, strong coupling, uncertainty, and ultra-high-dimensional input and output.

(2) Knowledge-Based Method

Knowledge-based fault diagnosis does not require an accurate mathematical model. Its basic idea is to use expert knowledge or qualitative relationship to develop the fault detection rules. The common approaches mainly include fault tree diagnosis (Hang et al. 2006), expert system diagnosis (Gath and Kulkarn 2014), directed graphs, fuzzy logic (Miranda and Felipe 2015), etc. The application of knowledge-based models

strongly relies on the complete process empirical knowledge. Once the information of the diagnosed object is known from expert experience and historical data, a variety of rules for appropriate reasoning is constructed. However, the accumulation of process experience and knowledge are time-consuming and even difficult. Therefore, this method is not universal and can only be applied to engineering systems which people are familiar with.

(3) Data-Driven Method

Data-driven method is based on the rise of modern information technology. In fact, it involves a variety of disciplines and techniques, including statistics, mathematical analysis, and signal processing. Generally speaking, the industrial data in the field are collected and stored by intelligent sensors. Data analysis can mine the hidden information contained in the data, establish the data model between input and output, help the operator to monitor the system status in real time, and achieve the purpose of fault diagnosis. Data-driven fault diagnosis methods are divided into three categories: signal processing-based, statistical analysis-based, and artificial intelligence-based (Zhou et al. 2011; Bersimis et al. 2007). The commonality of these methods is that high-dimensional variables are projected into the low-dimensional space with extracting the key features of the system. Data-driven method does not require an accurate model, so is more universal.

Both analytical techniques and data-driven methods have their own merits, but also have certain limitations. Therefore, the fusion-driven approach combining mechanistic knowledge and data could compensate the shortcomings of a single technique. This book explores the fault detection, fault isolation/identification, and fault root tracing problems mainly based on the multivariate statistical analysis as a mathematical foundation.

1.1.2 Statistical Process Monitoring

Fault detection and diagnosis based on multivariate statistical analysis has developed rapidly and a large number of results have emerged recently. This class of method, based on the historical data, uses multivariate projection to decompose the sample space into a low-dimensional principal element subspace and a residual subspace. Then the corresponding statistics are constructed to monitor the observation variables. Thus, this method also is called latent variable projection method.

(1) Fault Detection

The common multivariate statistical fault detection methods include principal component analysis (PCA), partial least squares (PLS), canonical correlation analysis (CCA), canonical variables analysis (CVA), and their extensions. Among them, PCA and PLS, as the most basic techniques, are usually used for monitoring processes with Gaussian distributions. These methods usually use Hotelling's T^2 and Squared Prediction Error (SPE) statistics to detect variation of process information.

It is worth noting that these techniques extract the process features by maximizing the variance or covariance of process variables. They only utilize the information of first-order statistics (mathematical expectation) and second-order statistics (variance and covariance) while ignoring the higher order statistics (higher order moments and higher order cumulants). Actually, there are few processes in practice that are subject to the Gaussian distribution. The traditional PCA and PLS are unable to extract effective features from non-Gaussian processes due to omitting the higher order statistics. It reduces the monitoring efficiency.

Numerous practical production conditions, such as strong nonlinearity, strong dynamics, and non-Gaussian distribution, make it difficult to directly apply the basic multivariate monitoring methods. To solve these practical problems, various extended multivariate statistical monitoring methods have flourished. For example, to deal with the process dynamics, dynamic PCA and dynamic PLS methods have been developed, which take into account the autocorrelation and cross-correlation among variables (Li and Gang 2006). To deal with the non-Gaussian distribution, independent component analysis (ICA) methods have also been developed (Yoo et al. 2004). To deal with the process nonlinearity, some extended kernel methods such as kernel PCA (KPCA), kernel PLS (KPLS), and kernel ICA (KICA) have emerged (Cheng et al. 2011; Zhang and Chi 2011; Zhang 2009).

(2) Fault Isolation or Identification

A common approach for separating faults is the contribution plot. It is an unsupervised approach that uses only the process data to find fault variables and does not require other prior knowledge. Successful separation based on the contribution plot includes the following properties: (1) each variable has the same mean value of contribution under the normal operation and (2) the faulty variables have very large contribution values under the fault conditions, compared with other normal variables. Alcalá and Qin summarized the commonly contribution plot techniques, such as complete decomposition contributions (CDC), partial decomposition contributions (PDC), and reconstruction-based contributions (RBC) (Alcalá and Qin 2009, 2011).

However, contribution plot usually suffers from the smearing effect, a situation in which non-faulty variables show larger contribution values, while the contribution values of the fault variables are smaller. Westerhuis et al. pointed out that one variable may affect other variables during the execution of PCA, thus creating a smearing effect (Westerhuis et al. 2000). Kerkhof et al. analyzed the smearing effect in three types of contribution indices, CDC, PDC, and RBC, respectively (Kerkhof et al. 2013). It was pointed that smearing effect is caused by the compression and expansion operations of variables from the perspective of mathematical decomposition. So it cannot be avoided during the transformation of data from measurement space to latent variable space. In order to eliminate the smearing effect, several new contribution indices are given based on dynamically calculating average value of the current and previous residuals (Wang et al. 2017).

If the historical data collected have been previously categorized into separate classes where each class pertains to a particular fault, fault isolation or identification can be transformed into pattern classification problem. The statistical methods, such as Fisher's discriminant analysis (FDA) (Chiang et al. 2000), have also been success-

fully applied in industrial practice to solve this problem. It assigns the data into two or more classes via three steps: feature extraction, discriminant analysis, and maximum selection. If the historical data have not been previously categorized, unsupervised cluster analysis may classify data into separate classes accordingly (Jain et al. 2000), such as the K-Means algorithm. More recently, neural network and machine learning techniques imported from statistical analysis theory have been receiving increasing attention, such as support vector data description (SVDD) covered in this book.

(3) Fault Diagnosis or Root Tracing

Fault root tracing based on Bayesian network (BN) is a typical diagnostic method that combines the mechanism knowledge and process data. BN, also known as probabilistic network or causal network, is a typical probabilistic graphical model. Since the end of last century, it has gradually become a research hotspot due to its superior theoretical properties in describing and reasoning about uncertain knowledge. BN was first proposed by Pearl, a professor at the University of California, in 1988, to solve the problem of uncertain information in artificial intelligence. BN represents the relationships between the causal variable in the form of directed acyclic graphs. In the fault diagnosis process of an industrial system, the observed variable is used as node containing all the information about the equipment, control quantities, and faults in the system. The causal connection between variables is quantitatively described as a directed edge with the conditional probability distribution function (Cai et al. 2017). Fault diagnosis procedure with BNs consists of BN structure modeling, BN parameter modeling, BN forward inference, and BN inverse tracing.

In addition to the probabilistic graphical model such as BN, the development of other causal graphical model has developed vigorously. These progresses aim at determining the causal relationship among the operating units of the system based on hypothesis testing (Zhang and Hyvärinen 2008; Shimizu et al. 2006). The generative model (linear or nonlinear) is built to explain the data generation process, i.e., causality. Then the direction of causality is tested under some certain assumptions. The most typical one is the linear non-Gaussian acyclic model (LiNGAM) and its improved version (Shimizu et al. 2006, 2011). It has the advantage of determining the causal structure of variables without pre-specifying their causal order. All these results are serving as a driving force for the development of probabilistic graphical model and playing a more important role in the field of fault diagnosis.

1.2 Fault Detection Index

The effectiveness of data-driven measures often depends on the characterization of process data changes. Generally, there are two types of changes in process data: common and special. Common changes are entirely caused by random noise, while specials refer to all data changes that are not caused by common causes, such as impulse disturbances. Common process control strategies may be able to remove most of the data changes with special reasons, but these strategies cannot remove the common cause changes inherent in the process data. As process data changes

are inevitable, statistical theory plays an important role in most process monitoring programs.

By defining faults as abnormal process conditions, it is easy to know that the application of statistical theory in the monitoring process actually relies on a reasonable assumption: unless the system fails, the data change characteristics are almost unchanged. This means that the characteristics of data fluctuations, such as mean and variance, are repeatable for the same operating conditions, although the actual value of the data may not be very predictable. The repeatability of statistical attributes allows automatic determination of thresholds for certain measures, effectively defining out-of-control conditions. This is an important step to automate the process monitoring program. Statistical process monitoring (SPM) relies on the use of normal process data to build process model. Here, we discuss the main points of SPM, i.e., fault detection index.

In multivariate process monitoring, the variability in the residual subspace (RS) is represented typically by squared sum of the residual, namely the Q statistic or the squared prediction error (SPE). The variability in the principle component subspace (PCS) is represented typically by Hotelling's T^2 statistic. Owing to the complementary nature of the two indices, combined indices are also proposed for fault detection and diagnosis. Another statistic that measures the variability in the RS is Hawkins' statistic (Hawkins 1974). The global Mahalanobis distance can also be used as a combined measure of variability in the PCS and RS. Individual tests of PCs can also be conducted (Hawkins 1974), but they are often not preferred in practice, since one has to monitor many statistics. In this section, we summarize several fault detection indices and provide a unified representation.

1.2.1 T^2 Statistic

Consider the sampled data with m observation variables $\mathbf{x} = [x_1, x_2, \dots, x_m]$ and n observations for each variable. The data are stacked into a matrix $\mathbf{X} \in \mathcal{R}^{n \times m}$, given by

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}, \quad (1.1)$$

firstly, the matrix \mathbf{X} is scaled to zero mean, and the sample covariance matrix is equal to

$$\mathbf{S} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}. \quad (1.2)$$

An eigenvalue decomposition of the matrix \mathbf{S} ,

$$\mathbf{S} = \bar{\mathbf{P}} \bar{\mathbf{\Lambda}} \bar{\mathbf{P}}^T = [\mathbf{P} \tilde{\mathbf{P}}] \text{diag}\{\mathbf{\Lambda}, \tilde{\mathbf{\Lambda}}\} [\mathbf{P} \tilde{\mathbf{P}}]^T. \quad (1.3)$$

The correlation structure of the covariance matrix \mathbf{S} is revealed, where \mathbf{P} is orthogonal. ($\mathbf{P}\mathbf{P}^T = \mathbf{I}$, in which, \mathbf{I} is the identity matrix) (Qin 2003) and

$$\begin{aligned} \mathbf{\Lambda} &= \frac{1}{n-1} \mathbf{T}^T \mathbf{T} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_k\} \\ \tilde{\mathbf{\Lambda}} &= \frac{1}{n-1} \tilde{\mathbf{T}}^T \tilde{\mathbf{T}} = \text{diag}\{\lambda_k + 1, \lambda_k + 2, \dots, \lambda_m\} \\ \lambda_1 &\geq \lambda_2 \geq \dots \geq \lambda_m, \quad \sum_{i=1}^k \lambda_i > \sum_{j=k+1}^m \lambda_j \\ \lambda_i &= \frac{1}{N-1} \mathbf{t}_i^T \mathbf{t}_i \approx \text{var}(\mathbf{t}_i) \end{aligned}$$

when n is very large. The score vector \mathbf{t}_i is the i -th column of $\bar{\mathbf{T}} = [\mathbf{T}, \tilde{\mathbf{T}}]$. The PCS is $S_p = \text{span}\{\mathbf{P}\}$ and the RS is $S_r = \text{span}\{\tilde{\mathbf{P}}\}$. Therefore, the matrix \mathbf{X} is decomposed into a score matrix $\bar{\mathbf{T}}$ and a loading matrix $\bar{\mathbf{P}} = [\mathbf{P}, \tilde{\mathbf{P}}]$, that is

$$\mathbf{X} = \bar{\mathbf{T}} \bar{\mathbf{P}}^T = \hat{\mathbf{X}} + \tilde{\mathbf{X}} = \mathbf{T} \mathbf{P}^T + \tilde{\mathbf{T}} \tilde{\mathbf{P}}^T = \mathbf{X} \mathbf{P} \mathbf{P}^T + \mathbf{X} (\mathbf{I} - \mathbf{P} \mathbf{P}^T), \quad (1.4)$$

The sample vector \mathbf{x} can be projected on the PCS and RS, respectively:

$$\mathbf{x} = \hat{\mathbf{x}} + \tilde{\mathbf{x}} \quad (1.5)$$

$$\hat{\mathbf{x}} = \mathbf{P} \mathbf{P}^T \mathbf{x} \quad (1.6)$$

$$\tilde{\mathbf{x}} = \tilde{\mathbf{P}} \tilde{\mathbf{P}}^T \mathbf{x} = (\mathbf{I} - \mathbf{P} \mathbf{P}^T) \mathbf{x}. \quad (1.7)$$

Assuming \mathbf{S} is invertible and with the definition

$$\mathbf{z} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{P}^T \mathbf{x}. \quad (1.8)$$

The Hotelling's T^2 statistic is given by Chiang et al. (2001)

$$T^2 = \mathbf{z}^T \mathbf{z} = \mathbf{x}^T \mathbf{P} \mathbf{\Lambda}^{-1} \mathbf{P}^T \mathbf{x}. \quad (1.9)$$

The observation vector \mathbf{x} is projected into a set of uncorrelated variables \mathbf{y} by $\mathbf{y} = \mathbf{P}^T \mathbf{x}$. The rotation matrix \mathbf{P} directly from the covariance matrix of \mathbf{x} guarantees that \mathbf{y} is correspond to \mathbf{x} . $\mathbf{\Lambda}$ scales the elements of \mathbf{y} to produce a set of variables with unit variance corresponding to the elements of \mathbf{z} . The conversion of the covariance matrix is demonstrated graphically in Fig. 1.2 for a two-dimensional observation space ($m = 2$) (Chiang et al. 2001).

The T^2 statistic is a scaled squared 2-norm of an observation vector \mathbf{x} from its mean. An appropriate scalar threshold is used to monitor the variability of the data in

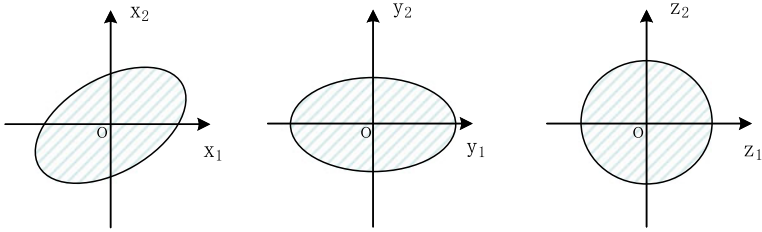
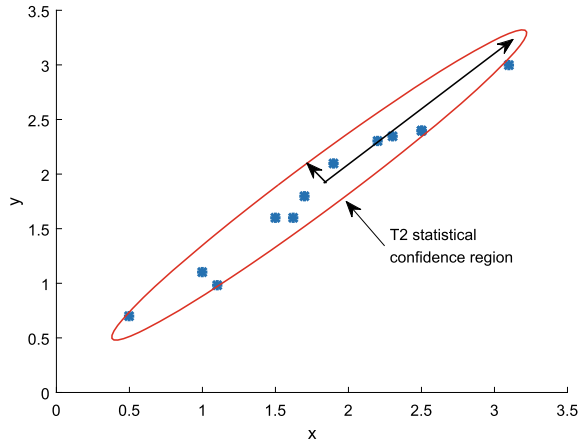


Fig. 1.2 A graphical illustration of the covariance conversion for the T^2 statistic

Fig. 1.3 An elliptical confidence region for the T^2 statistic



the entire m -dimensional observation space. It is determined based on an appropriate probability distribution with given significance level α . In general, it is assumed that

- the observations are randomly sampled and subject to a multivariate normal distribution.
- the mean vector and covariance matrix of observations sampled in the normal operations are equal to the actual ones, respectively.

Then the T^2 statistic follows a χ^2 distribution with m degrees of freedom (Chiang et al. 2001),

$$T_\alpha^2 = \chi_\alpha^2(m). \quad (1.10)$$

The set $T^2 \leq T_\alpha^2$ is an elliptical confidence region in the observation space, as illustrated in Fig. 1.3 for two process variables. This threshold (1.10) is applied to monitor the unusual changes. An observation vector projected within the confidence region indicates process data are in-control status, whereas outside projection indicates that a fault has occurred (Chiang et al. 2001).

When the actual covariance matrix for the normal status is not known but instead estimated from the sample covariance matrix (1.2), the threshold for fault detection

is given by

$$T_\alpha^2 = \frac{m(n-1)(n+1)}{n(n-m)} F_\alpha(m, n-m), \quad (1.11)$$

where $F_\alpha(m, n-m)$ is the upper $100\alpha\%$ critical point of the F-distribution with m and $n-m$ degrees of freedom (Chiang et al. 2001). For the same significance level α , the upper in-control limit in (1.11) is larger (more conservative) than that in (1.10). The two limits approach each other when the amount of observation increases ($n \rightarrow \infty$) (Tracy et al. 1992).

1.2.2 Squared Prediction Error

The SPE index measures the projection of the sample vector on the residual subspace:

$$\text{SPE} := \|\bar{\mathbf{x}}\|^2 = \|(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}\|^2. \quad (1.12)$$

The process is considered as normal if

$$\text{SPE} \leq \delta_\alpha^2, \quad (1.13)$$

where δ_α^2 denotes the upper control limit of SPE with a significant level of α . Jackson and Mudholkar gave an expression for δ_α^2 (Jackson and Mudholkar 1979)

$$\delta_\alpha^2 = \theta_1 \left(\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0}, \quad (1.14)$$

where

$$\theta_i = \sum_{j=k+1}^m \lambda_j^i, \quad i = 1, 2, 3, \quad (1.15)$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}, \quad (1.16)$$

where k is the number of retained principal components and z_α is the normal deviation corresponding to the upper percentile of $1 - \alpha$. Note that the above result is obtained under the following conditions.

- The sample vector \mathbf{x} follows a multivariate normal distribution.

- In deriving the control limits, an approximation is made to this distribution that is valid when θ_1 is very large.
- This result holds regardless of the number of principal components retained in the model.

When a fault occurs, the fault sample vector \mathbf{x} consists of the normal part superimposed on the faulty part. The fault causes the SPE to be larger than the threshold δ_α^2 , which results in the fault being detected.

Nomikos and MacGregor (1995) used the results in Box (1954) to derive an alternative upper control limit for SPE.

$$\delta_\alpha^2 = g\chi_{h;\alpha}^2 \quad (1.17)$$

where

$$g = \theta_2/\theta_1, \quad h = \theta_1^2/\theta_2. \quad (1.18)$$

The relationship between SPE threshold (1.14) and (1.17) is as follows: Nomikos and MacGregor (1995)

$$\delta_\alpha^2 \cong gh \left(1 - \frac{2}{9h} + z_\alpha \sqrt{\frac{2}{9h}} \right)^3$$

1.2.3 Mahalanobis Distance

Define the following Mahalanobis distance which forms the global Hotelling's T^2 test:

$$\mathbf{D} = \mathbf{X}^T \mathbf{S}^{-1} \mathbf{X} \sim \frac{m(n^2 - 1)}{n(n - m)} F_{m, n-m}, \quad (1.19)$$

where \mathbf{S} is the sample covariance of \mathbf{X} . When \mathbf{S} is singular with $\text{rank}(\mathbf{S}) = r < m$, Mardia discusses the use of the pseudo-inverse of \mathbf{S} , which in turn yields the Mahalanobis distance of the reduced-rank covariance matrix (Brereton 2015):

$$\mathbf{D}_r = \mathbf{X}^T \mathbf{S}^+ \mathbf{X} \sim \frac{r(n^2 - 1)}{n(n - r)} F_{r, n-r} \quad (1.20)$$

where \mathbf{S}^+ is the Moore-Penrose pseudo-inverse. It is straightforward to show that the global Mahalanobis distance is the sum of T^2 in PCS and $T_H^2 = \mathbf{x}^T \tilde{\mathbf{P}} \tilde{\mathbf{\Lambda}}^{-1} \tilde{\mathbf{P}}^T \mathbf{x}$ (Hawkins' statistic Hawkins 1974) in RS:

$$\mathbf{D} = T^2 + T_H^2. \quad (1.21)$$

When the number of observations n is quite large, the global Mahalanobis distance approximately obeys the χ^2 distribution with m degrees of freedom:

$$D \sim \chi_m^2. \quad (1.22)$$

Similarly, the reduced-rank Mahalanobis distance follows:

$$D_r \sim \chi_r^2. \quad (1.23)$$

Therefore, faults can be detected using the correspondingly defined control limits for D and D_r .

1.2.4 Combined Indices

In practice, better monitoring performance can be achieved in some cases by using a combined index instead of two indices to monitor the process. Yue and Qin proposed a combined index for fault detection that combines SPE and T^2 as follows: Yue and Qin (2001):

$$\varphi = \frac{\text{SPE}(X)}{\delta_\alpha^2} + \frac{T^2(X)}{\chi_{l,\alpha}^2} = X^T \Phi X, \quad (1.24)$$

where

$$\Phi = \frac{P\Lambda^{-1}P^T}{\chi_{l,\alpha}^2} + \frac{I - PP^T}{\delta_\alpha^2} = \frac{P\Lambda^{-1}P^T}{\chi_{l,\alpha}^2} + \frac{\tilde{P}\tilde{P}^T}{\delta_\alpha^2}. \quad (1.25)$$

Notice that Φ is symmetric and positive definite. To use this index for fault detection, the upper control limit of φ is derived from the results of Box (1954), which provides an approximate distribution with the same first two moments as the exact distribution. Using the approximate distribution given in Box (1954), the statistical data φ is approximated as follows:

$$\varphi = X^T \Phi X \sim g\chi_h^2, \quad (1.26)$$

where the coefficient

$$g = \frac{\text{tr}(\mathbf{S}\Phi)^2}{\text{tr}(\mathbf{S}\Phi)} \quad (1.27)$$

and the degree of freedom for χ_h^2 distribution is

$$h = \frac{[tr(\mathbf{S}\Phi)]^2}{tr(\mathbf{S}\Phi)^2}, \quad (1.28)$$

in which,

$$tr(\mathbf{S}\Phi) = \frac{l}{\chi_{l;\alpha}^2} + \frac{\sum_{i=l+1}^m \lambda_i}{\delta_\alpha^2} \quad (1.29)$$

$$tr(\mathbf{S}\Phi)^2 = \frac{l}{\chi_{l;\alpha}^4} + \frac{\sum_{i=l+1}^m \lambda_i^2}{\delta_\alpha^4} \quad (1.30)$$

After computing g and h , for a given significance level α , a control upper limit for φ can be obtained. A fault is detected by φ if

$$\varphi > g\chi_{h;\alpha}^2, \quad (1.31)$$

It is worth noting that Raich and Cinar suggest another combined statistic (Raich and Cinar 1996),

$$\varphi = c \frac{\text{SPE}(\mathbf{X})}{\delta_\alpha^2} + (1 - c) \frac{\text{T}^2(\mathbf{X})}{\chi_{l;\alpha}^2}, \quad (1.32)$$

where $c \in (0, 1)$ is a constant. They further give a rule that the statistic less than 1 is considered normal. However, this may lead to wrong results because even if the above statistic is less than 1, it is possible that $\text{SPE}(\mathbf{X}) > \delta_\alpha^2$ or $\text{T}^2(\mathbf{X}) > \chi_{l;\alpha}^2$ (Qin 2003).

1.2.5 Control Limits in Non-Gaussian Distribution

Nonlinear characteristics are the hotspot of current process monitoring research. Many nonlinear methods such as kernel principal component, neural network, and manifold learning are widely used in the component extraction of process monitoring. The principal component extracted by such methods may be independent of the Gaussian distribution. Thus, the control limits of the T^2 and Q statistical series are calculated by the probability density function, which can be estimated by the nonparametric kernel density estimation (KDE) method. The KDE applies to the T^2 and Q statistics because they are univariate although the processes represented by these statistics are multivariate. Therefore, the control limits for the monitoring

statistics (T^2 and SPE) are calculated from their respective PDF estimates, given by

$$\begin{aligned} \int_{-\infty}^{\text{Th}_{T^2,\alpha}} g(T^2)dT^2 &= \alpha \\ \int_{-\infty}^{\text{Th}_{\text{SPE},\alpha}} g(\text{SPE})d\text{SPE} &= \alpha, \end{aligned} \quad (1.33)$$

where

$$g(z) = \frac{1}{lh} \sum_{j=1}^l \mathbb{K}\left(\frac{z - z_j}{h}\right)$$

\mathbb{K} denotes a kernel function and h denotes the bandwidth or smoothing parameter.

Finally, the fault detection logic for the PCS and RS is as follows:

$$\begin{aligned} T^2 > \text{Th}_{T^2,\alpha} \text{ or } T_{\text{SPE}} > \text{Th}_{\text{SPE},\alpha}, & \quad \text{Faults} \\ T^2 \leq \text{Th}_{T^2,\alpha} \text{ and } T_{\text{SPE}} \leq \text{Th}_{\text{SPE},\alpha}, & \quad \text{Fault-free.} \end{aligned} \quad (1.34)$$

References

- Alcala CF, Qin SJ (2009) Reconstruction-based contribution for process monitoring. *Automatica* 45:1593–1600
- Alcala CF, Qin SJ (2011) Analysis and generalization of fault diagnosis methods for process monitoring. *J Process Control* 21:322–330
- Bersimis S, Psarakis S, Panaretos J (2007) Multivariate statistical process control charts: an overview. *Qual Reliab Eng Int* 23:517–543
- Box GEP (1954) Some theorems on quadratic forms applied in the study of analysis of variance problems. i. effect of inequality of variance in the one-way classification. *Ann Math Stat* 25:290–302
- Brereton RG (2015) The mahalanobis distance and its relationship to principal component scores. *J Chemom* 29:143–145
- Cai B, Huang L, Xie M (2017) Bayesian networks in fault diagnosis. *IEEE Trans Industr Inf* 13:2227–2240
- Cheng CY, Hsu MC, Chen CC (2011) Adaptive kernel principal component analysis (KPCA) for monitoring small disturbances of nonlinear processes. *Ind Eng Chem Res* 49:2254–2262
- Chiang LH, Russell EL, Braatz RD (2001) *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media
- Chiang LH, Russell EL, Braatz RD (2000) Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemom Intell Lab Syst* 50:243–252
- Ding SX (2013) *Model-based fault diagnosis techniques*. Springer, London

- Gath SJ, Kulkarni RV (2014) A review: expert system for diagnosis of myocardial infarction. *Comput Sci* 23:518–523
- Hang JR, Chang KH, Liao SH et al (2006) The reliability of general vague fault-tree analysis on weapon systems fault diagnosis. *Soft Comput* 10:531–542
- Hawkins DM (1974) The detection of errors in multivariate data using principal components. *J Am Stat Assoc* 69:340–344
- Hwang I, Kim S, Kim Y, Seah CE (2010) A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans Control Syst Technol* 18:636–653
- Jackson JE, Mudholkar G (1979) Control procedures for residuals associated with principal component analysis. *Technometrics* 21:341–349
- Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. *IEEE Trans Pattern Anal Mach Intell* 22:4–37
- Kerkhof PV, Vanlaer J, Gins G (2013) Analysis of smearing-out in contribution plot based fault isolation for statistical process control. *Chem Eng Sci* 104:285–293
- Li RG, Gang R (2006) Dynamic process fault isolation by partial DPCA. *Chem Biochem Eng Q* 20:69–78
- Miranda G, Felipe JC (2015) Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization. *Comput Biol Med* 64:334–346
- Nomikos P, MacGregor JF (1995) Multivariate SPC charts for monitoring batch processes. *Technometrics* 37:41–59
- Qin SJ (2003) Statistical process monitoring: basics and beyond. *J Chemom*, pp 480–502
- Raich A, Cinar A (1996) Statistical process monitoring and disturbance diagnosis in multivariate continuous processes. *AIChE J* 42:995–1009
- Shimizu S, Hoyer PO, Kerminen A (2006) A linear non-gaussian acyclic model for causal discovery. *J Mach Learn Res* 7:2003–2030
- Shimizu S, Inazumi T, Sogawa Y, Hyvärinen A, Kawahara Y, Washio T, Hoyer PO, Bollen K (2011) DirectLiNGAM: a direct method for learning a linear non-gaussian structural equation model. *J Mach Learn Res* 12(2):1225–1248
- Suzuki H, Kawahara T, Matsumoto S (1999) Fault diagnosis of space vehicle guidance and control systems using analytical redundancy. *Macromolecules* 31:86–95
- Tracy ND, Young JC, Mason RL (1992) Multivariate control charts for individual observations. *J Qual Control* 24:88–95
- Wang J, Ge WS, Zhou JL, Wu HY, Jin QB (2017) Fault isolation based on residual evaluation and contribution analysis and contribution analysis. *J Franklin Inst* 354:2591–2612
- Wang J, Shi Y, Zhou M, Wang Y, Puig V (2020) Active fault detection based on set-membership approach for uncertain discrete-time systems. *Int J Robust Nonlinear Control* 30:5322–5340
- Westerhuis J, Gurden S, Smilde A (2000) Generalized contribution plots in multivariate statistical process monitoring. *Chemom Intell Lab Syst* 51:95–114
- Yang CL, Masson GM, Leonetti RA (2006) On fault isolation and identification in t1t1-diagnosable systems. *IEEE Trans Comput C-35*:639–643
- Yoo CK, Lee JM, Vanrolleghem PA, Lee IB (2004) On-line monitoring of batch processes using multiway independent component analysis. *Chemom Intell Lab Syst*, p 15163
- Yu D (1997) Fault diagnosis for a hydraulic drive system using a parameter-estimation method. *Control Eng Pract* 5:1283–1291
- Yue H, Qin SJ (2001) Reconstruction based fault identification using a combined index. *Ind. Eng. Chem. Res.* 40:4403–4414
- Zhang Y (2009) Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chem Eng Sci* 64:801–811
- Zhang Y, Chi M (2011) Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS. *Chem Eng Sci* 66:64–72
- Zhang K, Hyvärinen A (2008) Distinguishing causes from effects using nonlinear acyclic causal models. In: *Proceedings of the 2008th international conference on causality: objectives and assessment*. *JMLR.org*, pp. 157–164

- Zhou DH, Hu YY (2009) Fault diagnosis techniques for dynamic systems. *Acta Automatica Sinica* 35:748–758
- Zhou DH, Li G, Li Y (2011) Data driven fault diagnosis technology for industrial process: based on PCA and PLS. Science Press, BeiJing

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 2

Multivariate Statistics in Single Observation Space



The observation data collected from continuous industrial processes usually have two main categories: process data and quality data, and the corresponding industrial data analysis is mainly for the two types of data based on the multivariate statistical techniques. Process data are collected by distributed control system (DCS) in real time with frequent sampling (its basic sampling period usually is 1s). For example, there are five typical variables in the process industries: temperature, pressure, flow rate, liquid level, and composition. Among them, temperature, pressure, flow rate, and liquid level are process variables. However, it is difficult to acquire the real-time quality measurement in general due to the limitation of quality sensors. Usually, the quality data are obtained by taking samples for laboratory test and their sampling frequency is much lower than that of process data. For example, product composition, viscosity, molecular weight distribution, and other quality-related parameters need to be obtained through various analytical instruments in the laboratory, such as composition analyzers, gel permeation chromatography (GPC), or mass spectrometry.

Process data and quality data belong to two different observation spaces, so the corresponding statistical analysis methods are correspondingly divided into two categories: single observation space and multiple observation spaces. This book introduces the basic multivariate statistical techniques from this perspective of observation space. This chapter focuses on the analysis methods in single observation space, including PCA and FDA methods. The core of these methods lies in the spatial projection oriented to different needs, such as sample dispersion or multi-class sample separation. This projection could extract the necessary and effective features while achieving the dimensional reduction. The next chapter focuses on the multivariate statistical analysis methods between two-observation space, specifically including PLS, CCA, and CVA. These methods aim at maximizing the correlation of variables in different observation spaces, and achieve the feature extraction and dimensional reduction.

2.1 Principal Component Analysis

As the modern industrial production system is becoming larger and more complex, the stored historical data not only has high dimensionality but also has strong coupling and correlation between the process variables. This also makes it impractical to monitor so many process variables at the same time. Therefore, we need to find a reasonable method to minimize the loss of information contained in the original variables while reducing the dimension of monitoring variables. If a small number of independent variables can be used to accurately reflect the operating status of the system, the operators can monitor these few variables to achieve the purpose of controlling the entire production process.

Principal component analysis (PCA) is one of the most widely used multivariate statistical algorithm (Pan et al. 2008). It is mainly used to monitor the process data with high dimensionality and strong linear correlation. It decomposes high-dimensional process variables into a few independent principal components and then establishing a model. The extracted features constitute the projection principal component subspace (PCS) of the PCA algorithm and this space contains most of the changes in the system. The remaining features constitute the residual subspace, which mainly contains the noise and interference during the monitoring process and a small amount of system change information (Wiesel and Hero 2009). Due to the integration of variables, PCA algorithm can be able to overcome the overlapping information caused by multiple correlations, and achieve dimensional reduction of high-dimensional data, simultaneously. It also highlights the main features and removes the noise and some unimportant features in the PCS.

2.1.1 Mathematical Principle of PCA

Suppose data matrix $\mathbf{X} \in \mathcal{R}^{n \times m}$, where m is the number of variables and n is the number of observations for each variable. Matrix \mathbf{X} can be decomposed into the sum of outer products of k vectors (Wang et al. 2016; Gao 2013):

$$\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \cdots + \mathbf{t}_k \mathbf{p}_k^T, \quad (2.1)$$

where \mathbf{t}_i is score vector, also called the principal component of the matrix \mathbf{X} , and \mathbf{p}_i is the feature vector corresponding to the principal component, also called load vector. Then (2.1) can also be written in the form of matrix:

$$\mathbf{X} = \mathbf{T} \mathbf{P}^T. \quad (2.2)$$

Among them, $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k]$ is called the score matrix and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k]$ is called the load matrix. The score vectors are orthogonal to each other,

$$\mathbf{t}_i^T \mathbf{t}_j = 0, i \neq j. \quad (2.3)$$

The following relationships exist between load vectors:

$$\begin{cases} \mathbf{p}_i^T \mathbf{p}_j = 0, i \neq j \\ \mathbf{p}_i^T \mathbf{p}_j = 1, i = j \end{cases} \quad (2.4)$$

It is shown that the load vectors are also orthogonal to each other and the length of each load vector is 1.

Multiplying the left and right sides of (2.2) by load vector \mathbf{p}_i and combining with (2.4), we can get

$$\mathbf{t}_i = \mathbf{X} \mathbf{p}_i. \quad (2.5)$$

Equation (2.5) shows that each score vector \mathbf{t}_i is the projection of the original data \mathbf{X} in the direction of the load vector \mathbf{p}_i corresponding to \mathbf{t}_i . The length of the score vector \mathbf{t}_i reflects the coverage degree of the original data \mathbf{X} in the direction of \mathbf{p}_i . The longer the length of \mathbf{t}_i , the greater the coverage degree or range of change of the data matrix \mathbf{X} in the direction of \mathbf{p}_i (Han 2012). The score vector \mathbf{t}_i is arranged as follows :

$$\|\mathbf{t}_1\| > \|\mathbf{t}_2\| > \|\mathbf{t}_3\| > \cdots > \|\mathbf{t}_k\|. \quad (2.6)$$

The load vector \mathbf{p}_1 represents the direction in which the data matrix \mathbf{X} changes most, and load vector \mathbf{p}_2 is orthogonal to \mathbf{p}_1 and represents the second largest direction of the data matrix \mathbf{X} changes. Similarly, the load vector \mathbf{p}_k represents the direction in which \mathbf{X} changes least. When most of the variance is contained in the first r load vectors and the variance contained in the latter $m - r$ load vectors is almost zero which could be omitted. Then the data matrix \mathbf{X} is decomposed into the following forms:

$$\begin{aligned} \mathbf{X} &= \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \cdots + \mathbf{t}_r \mathbf{p}_r^T + \mathbf{E} \\ &= \hat{\mathbf{X}} + \mathbf{E} = \mathbf{T} \mathbf{P}^T + \mathbf{E}, \end{aligned} \quad (2.7)$$

where $\hat{\mathbf{X}}$ is principle component matrix and \mathbf{E} is the residual matrix whose main information is caused by measurement noise. PCA divides the original data space into principal component subspace (PCS) and residual subspace (RS). These two subspaces are orthogonal and complementary to each other. The principal component subspace mainly reflects the changes caused by normal data, while the residual subspace mainly reflects the changes caused by noise and interference.

PCA is to calculate the optimal loading vectors \mathbf{p} by solving the optimization problem:

$$J = \max_{\mathbf{p} \neq \mathbf{0}} \frac{\mathbf{p}^T \mathbf{X}^T \mathbf{X} \mathbf{p}}{\mathbf{p}^T \mathbf{p}}. \quad (2.8)$$

The number r of principal components is generally obtained by cumulative percent variance (CPV). Use eigenvalue decomposition or singular value decomposition of

the covariance matrix of X and obtain all the eigenvalues λ_i . CPV is defined as follows:

$$CPV = \frac{\sum_{i=1}^r \lambda_i}{\sum_{i=1}^n \lambda_i}. \tag{2.9}$$

Generally, when the CPV value is greater than or equal to 85%, the corresponding number r is obtained.

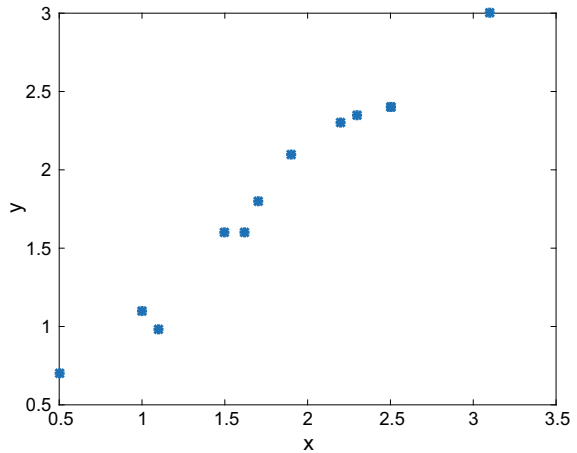
2.1.2 PCA Component Extraction Algorithm

There are two algorithms to implement PCA component extraction. Algorithm 1 is based on the singular value decomposition (SVD) of the covariance matrix and Algorithm 2 obtains each principal component based on Nonlinear Iterative Partial Least Squares algorithm (NIPALS), developed by H. Wold at first for PCA and later for PLS (Wold 1992). It gives more numerically accurate results compared with the SVD of the covariance matrix, but is slower to calculate.

The PCA dimensional reduction is illustrated by simple two-dimensional random data. Figure 2.1 shows the original random data sample in two-dimensional space. Figure 2.2 is a visualization with principal axis and confidence ellipse of the original data. The green ray gives the direction with the largest variance of the original data and the black ray shows the direction of second largest variance.

PCA projects the original data X from the two-dimensional space into one-dimensional subspace along the direction of maximum variance direction. The dimensional reduction is shown in Fig. 2.3.

Fig. 2.1 Two-dimensional raw random data



Algorithm 1 SVD-based component extraction algorithm

Input:Data matrix X .**Output:** r principal components.

[S1] Normalize the original data set $X = [\mathbf{x}^T(1), \mathbf{x}^T(2), \dots, \mathbf{x}^T(n)]^T \in R^{n \times m}$, in which $\mathbf{x} = [x_1, x_2, \dots, x_m] \in R^{1 \times m}$, with zero mean one variance.

[S2] Calculate the covariance matrix S of the Normalized data matrix X :

$$S = \frac{1}{n-1} X X^T. \quad (2.10)$$

[S3] Find the eigenvalues and eigenvectors of the covariance matrix S using eigenvalue decomposition:

$$\begin{aligned} |\lambda_i I - S| &= 0 \\ (\lambda_i I - S) \mathbf{p}_i &= 0. \end{aligned} \quad (2.11)$$

[S4] Sort the eigenvalues from large to small and determine the first r eigenvalues based on the CPV index. Construct the corresponding eigenvector matrix $P = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r]$ according to the eigenvectors $D = (\lambda_1, \dots, \lambda_r)$.

[S5] Calculate the score matrix T based on the following relationship:

$$X = T P. \quad (2.12)$$

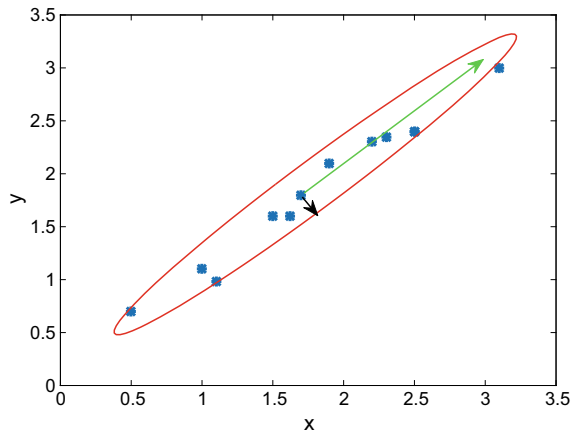
[S6] The normalized data matrix X is decomposed as follows:

$$X = \hat{X} + E = T P^T + \tilde{X}. \quad (2.13)$$

where \hat{X} is the principal component part of the data and \tilde{X} is the residual part.

return r principal components

Fig. 2.2 Visualization of the change principal axis and confidence ellipse of the original data



Algorithm 2 NIPALS-based component extraction algorithm

Input:Data matrix X .**Output:** r principal components.[S1] Normalize the original data X .[S2] Set $i = 1$ and choose a column x_j from X and mark it as $t_{1,i}$, that is, $t_{1,i} = x_j$.[S3] Calculate the load vector p_1

$$p_1 = \frac{X^T t_{1,i}}{t_{1,i}^T t_{1,i}}. \quad (2.14)$$

[S4] Normalize p_1 ,

$$p_1^T = \frac{p_1^T}{\|p_1\|}. \quad (2.15)$$

[S5] Calculate the score vector $t_{1,i+1}$,

$$t_{1,i+1} = \frac{X p_1}{p_1^T p_1}. \quad (2.16)$$

[S6] Compare the $t_{1,i}$ and $t_{1,i+1}$. If $\|t_{1,i+1} - t_{1,i}\| < \varepsilon$, and go to S7, where $\varepsilon > 0$ is a very small positive constant. If $\|t_{1,i+1} - t_{1,i}\| \geq \varepsilon$, set $i = i + 1$ and go back to S3.[S7] Calculate the residual $E_1 = X - t_1 p_1^T$, replace X with E_1 and return to S2 to calculate the next principal component t_2 until the CPV value meets the requirements.[S8] r principal components are obtained, namely:

$$X = t_1 p_1^T + t_2 p_2^T + \cdots + t_r p_r^T + \tilde{X} = T P^T + \tilde{X}, \quad (2.17)$$

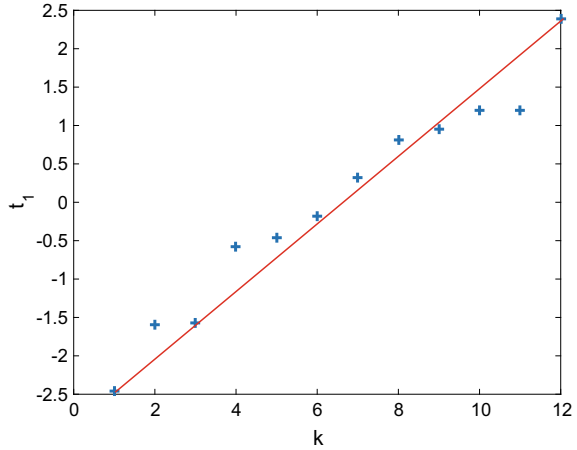
return r principal components

2.1.3 PCA Base Fault Detection

PCA can be applied to solve all kinds of data analysis problems, such as exploration and visualization of high-dimensional data sets, data compression, data preprocessing, dimensional reduction, removing data redundancy, and denoising. When it is applied to the field of FDD and the detection process is divided into offline modeling and online monitoring.

- (1) Offline modeling: use the training data to construct a principal component analysis model and calculate the monitored statistics, such as SPE and T^2 , and its control limits;
- (2) Online monitoring: when a new sample vector x is obtain, it can be decomposed into projections on PCS and RS (Zhang et al. 2017),

Fig. 2.3 Dimensional reduction results



$$\begin{aligned}
 \mathbf{x} &= \hat{\mathbf{x}} + \tilde{\mathbf{x}} \\
 \hat{\mathbf{x}} &= \mathbf{P}\mathbf{P}^T\mathbf{x} \\
 \tilde{\mathbf{x}} &= (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x},
 \end{aligned} \tag{2.18}$$

where $\hat{\mathbf{x}}$ is the projection of the sample \mathbf{x} in PCS and $\tilde{\mathbf{x}}$ is the projection of the sample in RS. Calculate the statistics, SPE (1.12) on RS and T^2 (1.9) on PCS of new sample \mathbf{x} , respectively. Compare the statistics of new sample with the control limits obtained from the training data. If the statistics of the new sample exceeds the control limit, it means that a fault has occurred, otherwise the system is in the normal operation.

$\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are not only orthogonal ($\hat{\mathbf{x}}^T\tilde{\mathbf{x}} = 0$) but also still statistically independent ($\mathbb{E}(\hat{\mathbf{x}}^T\tilde{\mathbf{x}}) = 0$). So, there are natural advantages to apply PCA algorithm to process monitoring. The flowchart of PCA based fault detection is shown in Fig. 2.4. In general, the fault detection process based on multivariate statistical analysis is similar as that of PCA, only the statistical model and statistics index are different.

2.2 Fisher Discriminant Analysis

Industrial processes are heavily instrumented and large amounts of data are collected online and stored in computer database. A lot of data are usually collected during out-of-control operations. When the data collected during an out-of-control operation has been previously diagnosed, the data can be classified into separate categories, where each category is related to a specific fault. When the data has not been diagnosed before, cluster analysis can help diagnose the operation of collecting data, and the data can be divided into a new category accordingly. If hyperplanes can separate the data

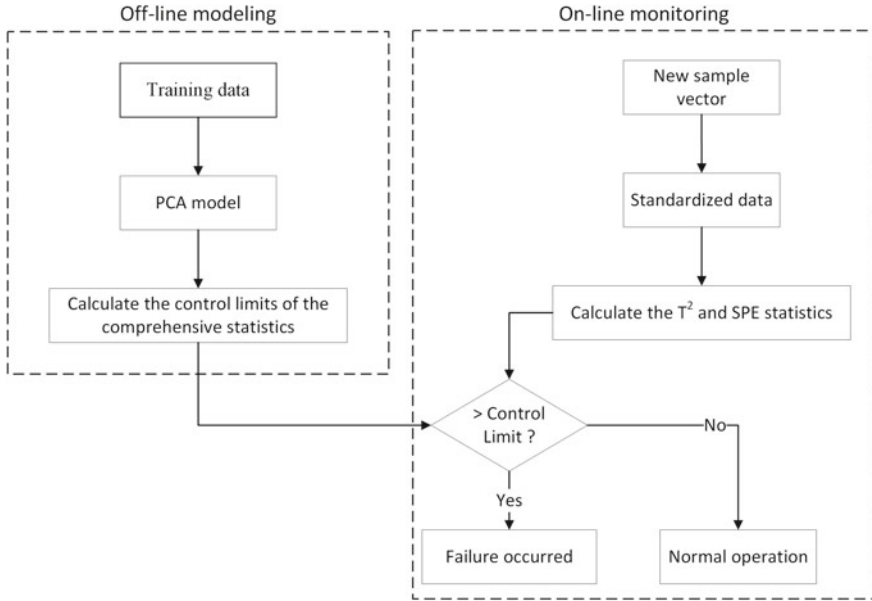


Fig. 2.4 PCA-based fault detection

in the class, as shown in Fig. 2.5, these separation planes can define the boundaries of each fault area. Once a fault is detected using the online data observation, the fault can be diagnosed by determining the fault area where the observation is located. Assuming that the detected fault is represented in the database, the fault can be correctly diagnosed in this way.

2.2.1 Principle of FDA

Fisher discriminant analysis (FDA), a dimensionality reduction technique that has been extensively studied in the pattern classification domain, takes into account the information between the classes. For fault diagnosis, data collected from the plant during in the specific fault operation are categorized into classes, where each class contains data representing a particular fault. FDA is a classical linear dimensionality reduction technique that is optimal in maximizing the separation between these classes. The main idea of FDA is to project data from a high-dimensional space into a lower dimensional space, and to simultaneously ensure that the projection maximizes the scatter between classes while minimizing the scatter within each class. It means that the high-dimensional data of the same class is projected to the low-dimensional space and clustered together, but the different classes are far apart.

Given training data for all classes $X \in R^{n \times m}$, where n and m are the number of observations and measurement variables, respectively. In order to understand FDA,

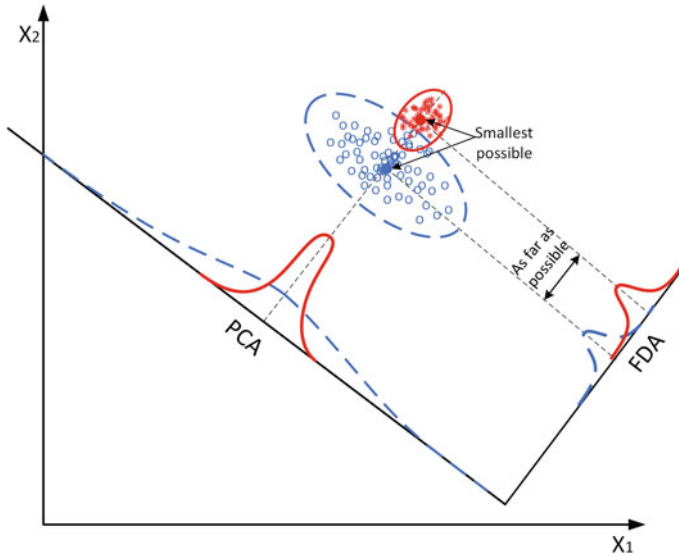


Fig. 2.5 Two-dimensional comparison of FDA and PCA

it is first necessary to define various matrices, including the total scatter matrix, intra-class (within-class) scatter matrix, and inter-class (between-class) scatter matrix. The total scatter matrix is

$$S_t = \sum_{i=1}^n (\mathbf{x}(i) - \bar{\mathbf{x}}) (\mathbf{x}(i) - \bar{\mathbf{x}})^T, \quad (2.19)$$

where $\mathbf{x}(i)$ represents the vector of measurement variables for the i -th observation and $\bar{\mathbf{x}}$ is the total mean vector.

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i). \quad (2.20)$$

The within-scatter matrix for class j is

$$S_j = \sum_{\mathbf{x}(i) \in \mathcal{X}_j} (\mathbf{x}(i) - \bar{\mathbf{x}}_j) (\mathbf{x}(i) - \bar{\mathbf{x}}_j)^T, \quad (2.10)$$

where \mathcal{X}_j is the set of vectors $\mathbf{x}(i)$ which belong to the class j and $\bar{\mathbf{x}}_j$ is the mean vector for class j :

$$\bar{\mathbf{x}}_j = \frac{1}{n_j} \sum_{\mathbf{x}(i) \in \mathcal{X}_j} \mathbf{x}(i), \quad (2.22)$$

where n_j is the number of observations in the j -th class. The **intra-class scatter matrix** is

$$\mathbf{S}_w = \sum_{j=1}^p \mathbf{S}_j, \quad (2.23)$$

where p is the number of classes. The **inter-class scatter matrix** is

$$\mathbf{S}_b = \sum_{j=1}^p n_j (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}) (\bar{\mathbf{x}}_j - \bar{\mathbf{x}})^T. \quad (2.24)$$

It is obvious that the following relationship always holds:

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w. \quad (2.25)$$

The maximum inter-class scatter means that the sample centers of different classes are as far apart as possible after projection ($\max \mathbf{v}^T \mathbf{S}_b \mathbf{v}$). The minimum intra-class scatter is equivalent to making the sample points of the same class after projection to be clustered together as much as possible ($\min \mathbf{v}^T \mathbf{S}_w \mathbf{v}$, $|\mathbf{S}_w| \neq 0$), where $\mathbf{v} \in \mathcal{R}^m$.

The optimal FDA project \mathbf{w} is obtained by

$$J = \max_{\mathbf{w} \neq 0} \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}. \quad (2.26)$$

Both the numerator and denominator have project vector \mathbf{w} . Considering that \mathbf{w} and $\alpha \mathbf{w}$, $\alpha \neq 0$ have the same effect, Let $\mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1$, then the optimal objective (2.26) becomes

$$\begin{aligned} J &= \max_{\mathbf{w}} \mathbf{w}^T \mathbf{S}_b \mathbf{w} \\ \text{s.t. } &\mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1. \end{aligned} \quad (2.27)$$

Firstly, let's consider the optimization of first FDA vector \mathbf{w}_1 . Solving (2.27) by Lagrange multiplier method.

$$L(\mathbf{w}_1, \lambda_1) = \mathbf{w}_1^T \mathbf{S}_b \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1^T \mathbf{S}_w \mathbf{w}_1 - 1)$$

Find the partial derivative of L with respect to \mathbf{w}_1 .

$$\frac{\partial L}{\partial \mathbf{w}_1} = 2\mathbf{S}_b \mathbf{w}_1 - 2\lambda_1 \mathbf{S}_w \mathbf{w}_1$$

The first FDA vector is equal to the eigenvectors \mathbf{w}_1 of the generalized eigenvalue problem.

$$\mathbf{S}_b \mathbf{w}_1 = \lambda_1 \mathbf{S}_w \mathbf{w}_1 \rightarrow \mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{w}_1 = \lambda_1 \mathbf{w}_1. \quad (2.28)$$

The first FDA vector boils down to finding the eigenvector \mathbf{w}_1 corresponding to the largest eigenvalue of the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$.

The second FDA vector is captured such that the inter-class scatter is maximized, while the intra-class scatter is minimized on all axes perpendicular to the first FDA vector and the same is true for the remaining FDA vectors. The k th FDA vectors is obtained by

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{w}_k = \lambda_k\mathbf{w}_k,$$

where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{p-1}$ and λ_k indicate the degree of overall separability among the classes by projecting the data onto \mathbf{w}_k .

When \mathbf{S}_w is invertible, the FDA vector can be computed from the generalized eigenvalue problem. This is almost always true as long as the number of observations n is significantly larger than the number of measurements m (the case in practice). If the \mathbf{S}_w matrix is not invertible, you can use PCA to project data into m_1 dimensions before executing FDA, in which m_1 is the number of non-zero eigenvalues of the covariance matrix \mathbf{S}_t .

The first FDA vector is the eigenvector associated with the largest eigenvalue, the second FDA vector is the eigenvector associated with the second largest eigenvalue, and so on. The large eigenvalue λ_k shows that when the data in classes are projected onto the associated eigenvector \mathbf{w}_k , there is a large overall separation of class means relative to the variance of the class, and thus, a large degree of separation among classes along the direction of \mathbf{w}_k . Since the rank of \mathbf{S}_b is less than p and at most $p - 1$ eigenvalues are not equal to zero. The FDA provides a useful ordering of eigenvectors only in these directions.

When FDA is used as a pattern classification, the dimensionality reduction technique is implemented for all classes of data at the same time. Denote $\mathbf{W}_a = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_a] \in \mathcal{R}^{m \times a}$. The discriminant function can be deduced as

$$g_j(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_j)^T \mathbf{W}_a \left(\frac{1}{n_j - 1} \mathbf{W}_a^T \mathbf{S}_j \mathbf{W}_a \right)^{-1} \mathbf{W}_a^T (\mathbf{x} - \bar{\mathbf{x}}_j) + \ln(p_i) - \frac{1}{2} \ln \left[\det \left(\frac{1}{n_j - 1} \mathbf{W}_a^T \mathbf{S}_j \mathbf{W}_a \right) \right]. \quad (2.29)$$

FDA can also be used to detect faults by defining an additional class of data on top of the fault class, i.e., data collected under normal operating conditions. The reliability of fault detection using (2.29) depends on the similarity between the data from normal operating conditions and the fault class data in the training set. Fault detection using FDA will yield small miss rates for known fault classes when a transformation \mathbf{W} exists such that data from normal operating conditions can be reasonably separated from other fault classes.

2.2.2 Comparison of FDA and PCA

As two classical techniques for dimensionality reduction of a single data set, PCA and FDA exhibit similar properties in many aspects. The optimization problems of PCA and FDA, respectively, formulated mathematically in (2.8) and (2.26), can also be captured as

$$J_{\text{PCA}} = \max_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^T \mathbf{S}_t \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \quad (2.30)$$

$$J_{\text{FDA}} = \max_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^T \mathbf{S}_t \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \quad (2.31)$$

In the special case, $\mathbf{S}_w = a\mathbf{I}$, $a \neq 0$, their vector optimization results are identical. This would occur if the data for each class could be described by a uniformly distributed ball (i.e., without a dominant direction), even if these balls had different sizes. The difference between these two techniques only occurs when the data used to describe either class appears elongated. These elongated shapes occur on highly correlated data sets, for example, the data collected in industrial processes. Thus, when FDA and PCA are applied to process data in the same way, the FDA vectors and the PCA loading vectors are significantly different. The different objectives of (2.30) and (2.31) show that the FDA has superior performance than PCA at distinguishing among fault classes.

Figure 2.5 illustrates a difference between PCA and FDA. The first FDA vector and the PCA loading vector are almost perpendicular. PCA is to map the entire data set to the coordinate axis that is most convenient to represent the data. The mapping does not use any classification information inside the data. Therefore, although the entire data set is more convenient to represent after PCA (reducing the dimensionality and minimizing the loss of information), it may become more difficult to classify. It is found that the projections of red and blue are overlapped in the PCA direction, but separated in the FDA direction. The two sets of data become easier to distinguish (it can be distinguished in low dimensions, reducing large amount of calculations) by FDA mapping.

To illustrate more clearly the difference between PCA and FDA, the following numerical example of binary classification is given.

$$\begin{aligned} \mathbf{x}_1 &= [5 + 0.05\boldsymbol{\mu}(0, 1); 3.2 + 0.9\boldsymbol{\mu}(0, 1)] \in \mathcal{R}^{2 \times 100} \\ \mathbf{x}_2 &= [5.1 + 0.05\boldsymbol{\mu}(0, 1); 3.2 + 0.9\boldsymbol{\mu}(0, 1)] \in \mathcal{R}^{2 \times 100} \\ \mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \in \mathcal{R}^{2 \times 200}, \end{aligned}$$

where $\boldsymbol{\mu}(0, 1) \in \mathcal{R}^{1 \times 100}$ is a uniformly distributed random vector on $[0, 1]$. \mathbf{X} is a two-mode data and its projection of FDA and PCA is shown in Fig. 2.6. The distribution of the data in the classes is somewhat elongated. The linear transformation of the data on the first FDA vector separates the two types of data better than the linear transformation of the data on the first PCA loading vector.

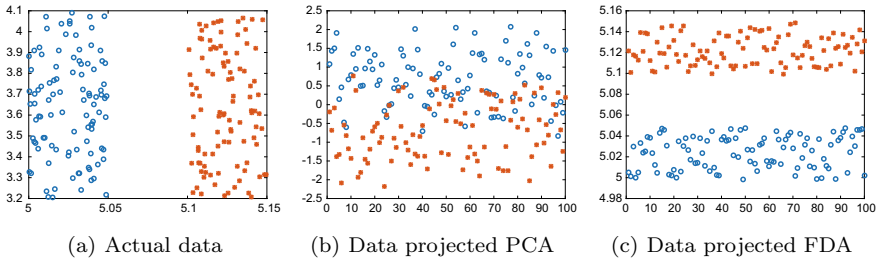


Fig. 2.6 Two-dimensional data projection comparison of FDA and PCA

Both PCA and FDA can be used to classify the original data after dimensionality reduction. PCA is an unsupervised method, i.e. it has no classification labels. After dimensionality reduction, unsupervised algorithms such as K-Means or self-organizing mapping networks are needed for classification. The FDA is a supervised method. It first reduces the dimensionality of the training data and then finds a linear discriminant function. The similarities and differences between FDA and PCA can be summarized as follows.

1. Similarities

- (1) Both functions are used to reduce dimensionality;
- (2) Both assume Gaussian distribution.

2. Differences

- (1) FDA is a supervised dimensionality reduction method, while PCA is unsupervised;
- (2) FDA dimensionality reduction can be reduced to the number of categories $k - 1$ at most, PCA does not have this restriction;
- (3) FDA is more dependent on the mean. If the sample information is more dependent on variance, the effect will not be as good as PCA;
- (4) FDA may overfit the data.

References

- Gao L (2013) Research of monitoring and fault diagnosis based on PCA for MIMO system. *Mech Eng Autom* 5:116–118
- Han T (2012) The study of fault diagnosis for industrial boiler based on principal component analysis. Dalian Maritime University
- Pan L, Li D, Ma J (2008) Principle and application of soft sensing technology. China Electric Power Press, Beijing
- Wang S, Li J, Gao X (2016) Exponential weighted principal component analysis method and its application in fault diagnosis. *Ind Instrum Autom Equip* 6:117–119

- Wiesel A, Hero A (2009) Decomposable principal component analysis. *IEEE Trans Signal Process* 57:4369–4377
- Wold S (1992) Nonlinear partial least squares modelling ii. spline inner relation. *Chemom Intell Lab Syst* 14(1–3):71–84
- Zhang K, Dong J, Peng K (2017) A novel dynamic non-gaussian approach for quality-related fault diagnosis with application to the hot strip mill process. *J Frankl* 354:702–721

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 3

Multivariate Statistics Between Two-Observation Spaces



As mentioned in the previous chapter, industrial data are usually divided into two categories, process data and quality data, belonging to different measurement spaces. The vast majority of smart manufacturing problems, such as soft measurement, control, monitoring, optimization, etc., inevitably require modeling the data relationships between the two kinds of measurement variables. This chapter's subject is to discover the correlation between the sets in different observation spaces.

The multivariate statistical analysis relying on correlation among variables generally include canonical correlation analysis (CCA) and partial least squares regression (PLS). They all perform linear dimensionality reduction with the goal of maximizing the correlation between variables in two measurement spaces. The difference are that CCA maximize **correlation**, while PLS maximize **covariance**.

3.1 Canonical Correlation Analysis

Canonical correlation analysis (CCA) was first proposed by Hotelling in 1936 (Hotelling 1936). It is a multivariate statistical analysis method that uses the correlation between two composite variables to reflect the overall correlation between two sets of variables. The CCA algorithm is widely used in the analysis of data correlation and it is also the basis of partial least squares. In addition, it is also used in feature fusion, data dimensionality reduction, and fault detection (Yang et al. 2015; Zhang and Dou 2015; Zhang et al. 2020; Hou 2013; Chen et al. 2016a, b).

3.1.1 Mathematical Principle of CCA

Assuming that there are l dependent variables $\mathbf{y} = (y_1, y_2, \dots, y_l)^T$ and m independent variables $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$. In order to capture the correlation between the dependent variables and the independent variables, n sample points are observed, which constitutes two data sets

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)]^T \in \mathbb{R}^{n \times m}$$

$$\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)]^T \in \mathbb{R}^{n \times l}$$

CCA draws on the idea of component extraction to find a canonical component u , which is a linear combination of variables x_i ; and a canonical component v , which is a linear combination of y_i . In the process of extraction, the correlation between u and v is required to be maximized. The correlation degree between u and v can roughly reflect the correlation between \mathbf{X} and \mathbf{Y} .

Without loss of generality, assuming that the original variables are all standardized, i.e., each column of the data set \mathbf{X} and \mathbf{Y} has mean 0 and variance 1, the covariance matrix of $\text{cov}(\mathbf{X}, \mathbf{Y})$ is equal to its correlation coefficient matrix, in which,

$$\text{cov}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \begin{bmatrix} \mathbf{X}^T \mathbf{X} & \mathbf{X}^T \mathbf{Y} \\ \mathbf{Y}^T \mathbf{X} & \mathbf{Y}^T \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix}$$

PCA is analyzed for Σ_{xx} or Σ_{yy} , while CCA is analyzed for Σ_{xy}

Now the problem is how to find the direction vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and then use them to construct the canonical components:

$$\begin{aligned} u &= \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_m x_m \\ v &= \beta_1 y_1 + \beta_2 y_2 + \dots + \beta_l y_l, \end{aligned} \tag{3.1}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_m]^T \in \mathbb{R}^{m \times 1}$, $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_l]^T \in \mathbb{R}^{l \times 1}$, such that the correlation between u and v is maximized. Obviously, the sample means of u and v are zero, and their sample variances are as follows:

$$\begin{aligned} \text{var}(u) &= \boldsymbol{\alpha}^T \Sigma_{xx} \boldsymbol{\alpha} \\ \text{var}(v) &= \boldsymbol{\beta}^T \Sigma_{yy} \boldsymbol{\beta} \end{aligned}$$

The covariance of u and v is

$$\text{cov}(u, v) = \alpha^T \Sigma_{xy} \beta.$$

One way to maximize the correlation of u and v is to make the corresponding correlation coefficient maximum, i.e.,

$$\max \rho(u, v) = \frac{\text{cov}(uv)}{\sqrt{\text{var}(u)\text{var}(v)}}. \tag{3.2}$$

In CCA, the following optimization objective is used:

$$\begin{aligned} J_{\text{CCA}} = \max \langle u, v \rangle &= \alpha^T \Sigma_{xy} \beta \\ \text{s.t. } \alpha^T \Sigma_{xx} \alpha &= 1; \beta^T \Sigma_{yy} \beta = 1. \end{aligned} \tag{3.3}$$

This optimization objective can be summarized as follows: to seek a unit vector α on the subspace of X and a unit vector β on the subspace of Y such that the correlation between u and v is maximized. Geometrically, $\rho(u, v)$ is again equal to the cosine of the angle between u and v . Thus, (3.3) is again equivalent to making the angle ω between u and v take the minimum value.

It can be seen from (3.3) that the goal of the CCA algorithm is finally transformed into a convex optimization process. The maximum value of this optimization goal is the correlation coefficient of X and Y , and the corresponding α and β are projection vectors, or linear coefficients. After the first pair of canonical correlation variables are obtained, the second to k th pair of canonical correlation variables that are not correlated with each other can be similarly calculated.

The following Fig. 3.1 shows the basic principle diagram of the CCA algorithm:

At present, there are two main methods which include eigenvalue decomposition and singular value decomposition for optimizing the above objective function to obtain α and β .

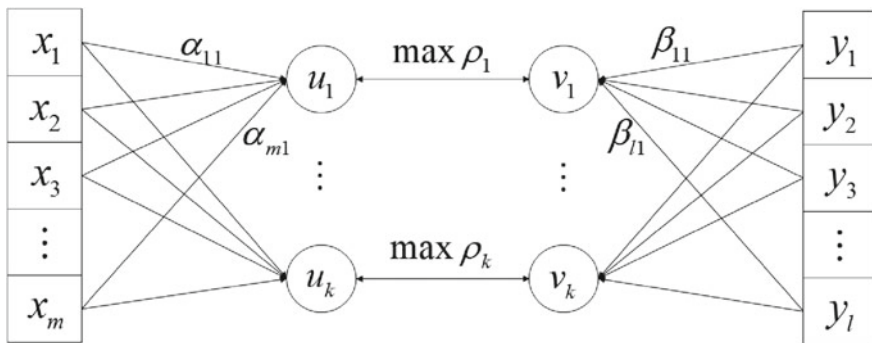


Fig. 3.1 Basic principle diagram of the CCA algorithm

3.1.2 Eigenvalue Decomposition of CCA Algorithm

Using the Lagrangian function, the objective function of (3.3) is transformed as follows:

$$\max J_{CCA}(\alpha, \beta) = \alpha^T \Sigma_{xy} \beta - \frac{\lambda_1}{2} (\alpha^T \Sigma_{xx} \alpha - 1) - \frac{\lambda_2}{2} (\beta^T \Sigma_{yy} \beta - 1). \quad (3.4)$$

Set $\frac{\partial J}{\partial \alpha} = 0$ and $\frac{\partial J}{\partial \beta} = 0$, then

$$\begin{aligned} \Sigma_{xy} \beta - \lambda_1 \Sigma_{xx} \alpha &= 0 \\ \Sigma_{xy}^T \alpha - \lambda_2 \Sigma_{yy} \beta &= 0. \end{aligned} \quad (3.5)$$

Let $\lambda = \lambda_1 = \lambda_2 = \alpha^T \Sigma_{xy} \beta$, and multiply (3.5) to the left by Σ_{xx}^{-1} and Σ_{yy}^{-1} , respectively, and get:

$$\begin{aligned} \Sigma_{xx}^{-1} \Sigma_{xy} \beta &= \lambda \alpha \\ \Sigma_{yy}^{-1} \Sigma_{yx} \alpha &= \lambda \beta. \end{aligned} \quad (3.6)$$

Substituting the second formula in (3.6) into the first formula, we can get

$$\Sigma_{xx}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \alpha = \lambda^2 \alpha \quad (3.7)$$

From (3.7), we can get the largest eigenvalue λ and the corresponding maximum eigenvector α only by eigenvalue decomposition of the matrix $\Sigma_{xx}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx}$. In the similar way, the vector β can be obtained. At this time, the projection vectors α and β of a set of canonical correlation variables can be obtained.

3.1.3 SVD Solution of CCA Algorithm

Let $\alpha = \Sigma_{xx}^{-1/2} \mathbf{a}$, $\beta = \Sigma_{yy}^{-1/2} \mathbf{b}$, and then we can get

$$\begin{aligned} \alpha^T \Sigma_{xx} \alpha = 1 &\rightarrow \mathbf{a}^T \Sigma_{xx}^{-1/2} \Sigma_{xx} \Sigma_{xx}^{-1/2} \mathbf{a} = 1 \rightarrow \mathbf{a}^T \mathbf{a} = 1 \\ \beta^T \Sigma_{yy} \beta = 1 &\rightarrow \mathbf{b}^T \Sigma_{yy}^{-1/2} \Sigma_{yy} \Sigma_{yy}^{-1/2} \mathbf{b} = 1 \rightarrow \mathbf{b}^T \mathbf{b} = 1 \\ \alpha^T \Sigma_{xy} \beta &= \mathbf{a}^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \mathbf{b}. \end{aligned} \quad (3.8)$$

In other words, the objective function of (3.3) can be transformed into as follows:

$$\begin{aligned} J_{\text{CCA}}(\mathbf{a}, \mathbf{b}) &= \arg \max_{\mathbf{a}, \mathbf{b}} \mathbf{a}^T \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2} \mathbf{b} \\ \text{s.t. } &\mathbf{a}^T \mathbf{a} = \mathbf{b}^T \mathbf{b} = 1. \end{aligned} \quad (3.9)$$

A singular value decomposition for matrix \mathbf{M} yields

$$\mathbf{M} = \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2} = \boldsymbol{\Gamma} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Lambda}_\kappa & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.10)$$

where κ is the number of principal elements or non-zero singular values, and $\kappa \leq \min(l, m)$, $\boldsymbol{\Lambda}_\kappa = \text{diag}(\lambda_1, \dots, \lambda_\kappa)$, $\lambda_1 \geq \dots \geq \lambda_\kappa > 0$.

Since all columns of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$ are standard orthogonal basis, $\mathbf{a}^T \boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}^T \mathbf{b}$ are vectors with only one scalar value of 1, and the remaining scalar value of 0. So, we can get

$$\mathbf{a}^T \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2} \mathbf{b} = \mathbf{a}^T \boldsymbol{\Gamma} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \mathbf{b} = \sigma_{ab}. \quad (3.11)$$

From (3.11), it can be seen that $\mathbf{a}^T \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1/2} \mathbf{b}$ maximizes actually the left and right singular vectors corresponding to the maximum singular values of \mathbf{M} . Thus, using the corresponding left and right singular vectors $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$, we can obtain the projection vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for a set of canonical correlation variables, namely,

$$\begin{aligned} \boldsymbol{\alpha} &= \boldsymbol{\Sigma}_{xx}^{-1/2} \mathbf{a} \\ \boldsymbol{\beta} &= \boldsymbol{\Sigma}_{yy}^{-1/2} \mathbf{b}. \end{aligned} \quad (3.12)$$

3.1.4 CCA-Based Fault Detection

When there is a clear input-output relationship between the two types of data measurable online, CCA can be used to design an effective fault detection system. The CCA-based fault detection method can be considered as an alternative to PCA-based fault detection method, and an extension of PLS-based fault detection method (Chen et al. 2016a).

Let

$$\begin{aligned} \mathbf{J}_s &= \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Gamma}(:, 1 : \kappa) \\ \mathbf{L}_s &= \boldsymbol{\Sigma}_{yy}^{-1/2} \boldsymbol{\Psi}(:, 1 : \kappa) \\ \mathbf{J}_{\text{res}} &= \boldsymbol{\Sigma}_{xx}^{-1/2} \boldsymbol{\Gamma}(:, \kappa + 1 : l) \\ \mathbf{L}_{\text{res}} &= \boldsymbol{\Sigma}_{yy}^{-1/2} \boldsymbol{\Psi}(:, \kappa + 1 : m). \end{aligned}$$

According to CCA method, $\mathbf{J}_s^T \mathbf{x}$ and $\mathbf{L}_s^T \mathbf{y}$ are closely related. However, in actual systems, measurement variables are inevitably affected by noise, and the correlation between $\mathbf{J}_s^T \mathbf{x}$ and $\mathbf{L}_s^T \mathbf{y}$ can be expressed as

$$\mathbf{L}_s^T \mathbf{y}(k) = \mathbf{A}_\kappa^T \mathbf{J}_s^T \mathbf{x}(k) + v_s(k), \quad (3.13)$$

where v_s is the noise term and weakly related to $\mathbf{J}_s^T \mathbf{x}$. Based on this, the residual vector is

$$\mathbf{r}_1(k) = \mathbf{L}_s^T \mathbf{y}(k) - \mathbf{A}_\kappa^T \mathbf{J}_s^T \mathbf{x}(k). \quad (3.14)$$

Assume that the input and output data obey the Gaussian distribution. It is known that linear transformation does not change the distribution of random variables, so the residual signal \mathbf{r}_1 also obeys the Gaussian distribution and its covariance matrix is

$$\boldsymbol{\Sigma}_{r_1} = \frac{1}{N-1} (\mathbf{L}_s^T \mathbf{Y} - \mathbf{A}_\kappa^T \mathbf{J}_s^T \mathbf{X}) (\mathbf{L}_s^T \mathbf{Y} - \mathbf{A}_\kappa^T \mathbf{J}_s^T \mathbf{X})^T = \frac{\mathbf{I}_\kappa - \mathbf{A}_\kappa^2}{N-1}. \quad (3.15)$$

Similarly, another residual vector can be obtained

$$\mathbf{r}_2(k) = \mathbf{J}_s^T \mathbf{x}(k) - \mathbf{A}_\kappa \mathbf{L}_s^T \mathbf{y}(k). \quad (3.16)$$

Its covariance matrix is

$$\boldsymbol{\Sigma}_{r_2} = \frac{1}{N-1} (\mathbf{J}_s^T \mathbf{U} - \mathbf{A}_\kappa \mathbf{L}_s^T \mathbf{Y}) (\mathbf{J}_s^T \mathbf{U} - \mathbf{A}_\kappa \mathbf{L}_s^T \mathbf{Y})^T = \frac{\mathbf{I}_\kappa - \mathbf{A}_\kappa^2}{N-1}. \quad (3.17)$$

It can be seen from formula (3.15)–(3.16) that the covariance of residual \mathbf{r}_1 and \mathbf{r}_2 are the same. For fault detection, the following two statistics can be constructed:

$$\mathbf{T}_1^2(k) = (N-1) \mathbf{r}_1^T(k) (\mathbf{I}_\kappa - \mathbf{A}_\kappa^2)^{-1} \mathbf{r}_1(k) \quad (3.18)$$

$$\mathbf{T}_2^2(k) = (N-1) \mathbf{r}_2^T(k) (\mathbf{I}_\kappa - \mathbf{A}_\kappa^2)^{-1} \mathbf{r}_2(k). \quad (3.19)$$

3.2 Partial Least Squares

Multiple linear regression analysis is relatively common and the least square method is generally used to estimate the regression coefficient in this type of regression method. But the least square technique often fails when there is multiple correlation between the independent variables or the number of samples is less than the number of variables. So the partial least square technique is developed to resolve this problem. S. Wold and C. Albano et al. proposed the partial least squares method for the first time and applied it to the field of chemistry (Wold et al. 1989). It aims at

the regression modeling between two sets of multi-variables with high correlation and integrates the basic functions of multiple linear regression analysis, principal component analysis, and canonical correlation analysis. PLS is also called the second-generation regression analysis method due to its simplification model in the data structure and correlation (Hair et al. 2016). It has developed rapidly and widely used in various fields recent years (Okwuashi et al. 2020; Ramin et al. 2018).

3.2.1 Fundamental of PLS

Suppose there are l dependent variables (y_1, y_2, \dots, y_l) and m independent variables (x_1, x_2, \dots, x_m) . In order to study the statistical relationship between the dependent variable and the independent variable, n sample points are observed, which constitutes a data set $(\mathbf{X} = [x_1, x_2, \dots, x_m] \in R^{n \times m}, \mathbf{Y} = [y_1, y_2, \dots, y_l] \in R^{n \times l})$ of the independent variables and the dependent variables.

To address the problems encountered in least squares multiple regression between \mathbf{X} and \mathbf{Y} , the concept of component extraction is introduced in PLS regression analysis. Recall that principal component analysis, for a single data matrix \mathbf{X} , finds the composite variable that best summarizes the information in the original data. The principal component \mathbf{T} in \mathbf{X} is extracted with the maximum variance information of the original data:

$$\max \text{var}(\mathbf{T}), \quad (3.20)$$

PLS extracts component vectors \mathbf{t}_i and \mathbf{u}_i from \mathbf{X} and \mathbf{Y} , which means \mathbf{t}_i is a linear combination of (x_1, x_2, \dots, x_m) , and \mathbf{u}_i is a linear combination of (y_1, y_2, \dots, y_l) . During the extracting of components, in order to meet the needs of regression analysis, the following two requirements should be satisfied:

- (1) \mathbf{t}_i and \mathbf{u}_i carry the variation information in their respective data set as much as possible, respectively;
- (2) The correlation between \mathbf{t}_i and \mathbf{u}_i is maximized.

The two requirements indicate that \mathbf{t}_i and \mathbf{u}_i should represent the data set \mathbf{X} and \mathbf{Y} as well as possible and the component \mathbf{t}_i of the independent variable has the best ability to explain the component \mathbf{u}_i of the dependent variable.

3.2.2 PLS Algorithm

The most popular algorithm used in PLS to compute the vectors in the calibration step is known as nonlinear iterative partial least squares (NIPALS). First, normalize the data to achieve the purpose of facilitating calculations. Normalize \mathbf{X} to get matrix \mathbf{E}_0 and normalize \mathbf{Y} to get matrix \mathbf{F}_0 :

$$\mathbf{E}_0 = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \vdots \\ x_{n1} & \cdots & x_{mn} \end{bmatrix}, \quad \mathbf{F}_0 = \begin{bmatrix} y_{11} & \cdots & y_{1l} \\ \vdots & \vdots & \vdots \\ y_{n1} & \cdots & y_{nl} \end{bmatrix} \quad (3.21)$$

In the first step, set \mathbf{t}_1 ($\mathbf{t}_1 = \mathbf{E}_0 \mathbf{w}_1$) to be the first component of \mathbf{E}_0 , and \mathbf{w}_1 is the first direction vector of \mathbf{E}_0 , which is a unit vector, $\|\mathbf{w}_1\| = 1$. Similarly, set \mathbf{u}_1 ($\mathbf{u}_1 = \mathbf{F}_0 \mathbf{c}_1$) to be the first component of \mathbf{F}_0 , and \mathbf{c}_1 is the first direction vector of \mathbf{F}_0 , which is a unit vector too, $\|\mathbf{c}_1\| = 1$.

According to the principle of principal component analysis, \mathbf{t}_1 and \mathbf{u}_1 should meet the following conditions in order to be able to represent the data variation information in X and Y well:

$$\begin{aligned} \max \text{var}(\mathbf{t}_1) \\ \max \text{var}(\mathbf{u}_1) \end{aligned} \quad (3.22)$$

On the other hand, \mathbf{t}_1 is further required to have the best explanatory ability for \mathbf{u}_1 due to the needs of regression modeling. According to the thinking of canonical correlation analysis, the correlation between \mathbf{t}_1 and \mathbf{u}_1 should reach the maximum value:

$$\max r(\mathbf{t}_1, \mathbf{u}_1). \quad (3.23)$$

The covariance of \mathbf{t}_1 and \mathbf{u}_1 is usually used to describe the correlation in partial least squares regression:

$$\max \text{Cov}(\mathbf{t}_1, \mathbf{u}_1) = \sqrt{\text{var}(\mathbf{t}_1) \text{var}(\mathbf{u}_1)} r(\mathbf{t}_1, \mathbf{u}_1) \quad (3.24)$$

Converting to the normal mathematical expression, \mathbf{t}_1 and \mathbf{u}_1 is solved by the following optimization problem:

$$\begin{aligned} \max_{\mathbf{w}_1, \mathbf{c}_1} \langle \mathbf{E}_0 \mathbf{w}_1, \mathbf{F}_0 \mathbf{c}_1 \rangle \\ \text{s.t.} \begin{cases} \mathbf{w}_1^T \mathbf{w}_1 = 1 \\ \mathbf{c}_1^T \mathbf{c}_1 = 1. \end{cases} \end{aligned} \quad (3.25)$$

Therefore, it needs to calculate the maximum value of $\mathbf{w}_1^T \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1$ under the constraints of $\|\mathbf{w}_1\|^2 = 1$ and $\|\mathbf{c}_1\|^2 = 1$.

In this case, the Lagrangian function is

$$s = \mathbf{w}_1^T \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 - \lambda_1 (\mathbf{w}_1^T \mathbf{w}_1 - 1) - \lambda_2 (\mathbf{c}_1^T \mathbf{c}_1 - 1). \quad (3.26)$$

Calculate the partial derivatives of s with respect to \mathbf{w}_1 , \mathbf{c}_1 , λ_1 , and λ_2 , and let them be zero

$$\frac{\partial s}{\partial \mathbf{w}_1} = \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 - 2\lambda_1 \mathbf{w}_1 = 0, \quad (3.27)$$

$$\frac{\partial s}{\partial \mathbf{c}_1} = \mathbf{E}_0^T \mathbf{F}_0 \mathbf{w}_1 - 2\lambda_2 \mathbf{c}_1 = 0, \quad (3.28)$$

$$\frac{\partial s}{\partial \lambda_1} = -(\mathbf{w}_1^T \mathbf{w}_1 - 1) = 0, \quad (3.29)$$

$$\frac{\partial s}{\partial \lambda_2} = -(\mathbf{c}_1^T \mathbf{c}_1 - 1) = 0. \quad (3.30)$$

It can be derived from the above formulas that

$$2\lambda_1 = 2\lambda_2 = \mathbf{w}_1^T \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 = \langle \mathbf{E}_0 \mathbf{w}_1, \mathbf{F}_0 \mathbf{c}_1 \rangle \quad (3.31)$$

Let $\theta_1 = 2\lambda_1 = 2\lambda_2 = \mathbf{w}_1^T \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1$, so θ_1 is the value of the objective function of the optimization problem (3.25). Then (3.27) and (3.28) are rewritten as

$$\mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 = \theta_1 \mathbf{w}_1, \quad (3.32)$$

$$\mathbf{F}_0^T \mathbf{E}_0 \mathbf{w}_1 = \theta_1 \mathbf{c}_1. \quad (3.33)$$

Substitute (3.33) into (3.32),

$$\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0 \mathbf{w}_1 = \theta_1^2 \mathbf{w}_1. \quad (3.34)$$

Substitute (3.32) into (3.33) simultaneously,

$$\mathbf{F}_0^T \mathbf{E}_0 \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 = \theta_1^2 \mathbf{c}_1. \quad (3.35)$$

Equation (3.34) shows that \mathbf{w}_1 is the eigenvector of matrix $\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0$ with the corresponding eigenvalue θ_1^2 . Here, θ_1 is the objective function. If we want to get its maximum value, \mathbf{w}_1 should be the unit eigenvector of the maximum eigenvalue of matrix $\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0$. Similarly, \mathbf{c}_1 should be the unit eigenvector of the largest eigenvalue of the matrix $\mathbf{F}_0^T \mathbf{E}_0 \mathbf{E}_0^T \mathbf{F}_0$.

Then the first components t_1 and \mathbf{u}_1 are calculated from the direction vectors \mathbf{w}_1 and \mathbf{c}_1 :

$$\begin{aligned} t_1 &= \mathbf{E}_0 \mathbf{w}_1 \\ \mathbf{u}_1 &= \mathbf{F}_0 \mathbf{c}_1. \end{aligned} \quad (3.36)$$

The regression equations of \mathbf{E}_0 and \mathbf{F}_0 is found by t_1 and \mathbf{u}_1 :

$$\begin{aligned} \mathbf{E}_0 &= t_1 \mathbf{p}_1^T + \mathbf{E}_1 \\ \mathbf{F}_0 &= \mathbf{u}_1 \mathbf{q}_1^T + \mathbf{F}_1^* \\ \mathbf{F}_0 &= t_1 \mathbf{r}_1^T + \mathbf{F}_1. \end{aligned} \quad (3.37)$$

The regression coefficient vectors in (3.37) are

$$\begin{aligned} p_1 &= \frac{\mathbf{E}_0^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2} \\ q_1 &= \frac{\mathbf{F}_0^T \mathbf{u}_1}{\|\mathbf{u}_1\|^2} \\ r_1 &= \frac{\mathbf{F}_0^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2}. \end{aligned} \quad (3.38)$$

\mathbf{E}_1 , \mathbf{F}_1^* and \mathbf{F}_1 are the residual matrices of the three regression equations.

Second step is to replace \mathbf{E}_0 and \mathbf{F}_0 with residual matrices \mathbf{E}_1 and \mathbf{F}_1 , respectively. Then find the second pair of direction vectors \mathbf{w}_2 , \mathbf{c}_2 , and the second pair of components \mathbf{t}_2 and \mathbf{u}_2 :

$$\begin{aligned} \mathbf{t}_2 &= \mathbf{E}_1 \bar{\mathbf{w}}_2 \\ \mathbf{u}_2 &= \mathbf{F}_1 \mathbf{c}_2 \\ \theta_2 &= \mathbf{w}_2^T \mathbf{E}_1^T \mathbf{F}_1 \mathbf{c}_2. \end{aligned} \quad (3.39)$$

Similarly, \mathbf{w}_2 is the unit eigenvector corresponding to the largest eigenvalue of matrix $\mathbf{E}_1^T \mathbf{F}_1 \mathbf{F}_1^T \mathbf{E}_1$, and \mathbf{c}_2 is the unit eigenvector of the largest eigenvalue of matrix $\mathbf{F}_1^T \mathbf{E}_1 \mathbf{E}_1^T \mathbf{F}_1$. Calculate the regression coefficient

$$\begin{aligned} p_2 &= \frac{\mathbf{E}_1^T \mathbf{t}_2}{\|\mathbf{t}_2\|^2} \\ r_2 &= \frac{\mathbf{F}_1^T \mathbf{t}_2}{\|\mathbf{t}_2\|^2}. \end{aligned} \quad (3.40)$$

The regression equation is updated:

$$\begin{aligned} \mathbf{E}_1 &= \mathbf{t}_2 \mathbf{p}_2^T + \mathbf{E}_2 \\ \mathbf{F}_1 &= \mathbf{t}_2 \mathbf{r}_2^T + \mathbf{F}_2. \end{aligned} \quad (3.41)$$

Repeat the calculation according to the above steps. If the rank of \mathbf{X} is R , the regression equation can be obtained:

$$\begin{aligned} \mathbf{E}_0 &= \mathbf{t}_1 \mathbf{p}_1^T + \cdots + \mathbf{t}_R \mathbf{p}_R^T \\ \mathbf{F}_0 &= \mathbf{t}_1 \mathbf{r}_1^T + \cdots + \mathbf{t}_R \mathbf{r}_R^T + \mathbf{F}_R. \end{aligned} \quad (3.42)$$

If the number of feature vectors used in the PLS modeling is large enough, the residuals could be zero. In general, it only needs to select a ($a \ll R$) components among them to form a regression model with better prediction. The number of principal components required for modeling is determined by cross-validation discussed

in Sect. 3.2.3. Once the appropriate component number is determined, the external relationship of the input variable matrix X as

$$X = TP^T + \bar{X} = \sum_{h=1}^a t_h p_h^T + \bar{X}. \quad (3.43)$$

The external relationship of the output variable matrix Y can be written as

$$Y = UQ^T + \bar{Y} = \sum_{h=1}^a u_h q_h^T + \bar{Y}. \quad (3.44)$$

The internal relationship is expressed as

$$\hat{u}_h = b_h t_h, \quad b_h = t_h^T u_h / t_h^T t_h. \quad (3.45)$$

3.2.3 Cross-Validation Test

In many cases, the PLS equation does not require the selection of all principal components for regression modeling, but rather, as in principal component analysis, the first d ($d \leq l$) principal components can be selected in a truncated manner, and a better predictive model can be obtained using only these d principal components. In fact, if the subsequent principal components no longer provide more meaningful information to explain the dependent variable, using too many principal components will only undermine the understanding of the statistical trend and lead to wrong prediction conclusions. The number of principal components required for modeling can be determined by cross-validation.

Cross-validation is used to prevent over-fitting caused by complex model. Sometimes referred to as the circular estimation, it is a statistically useful method for cutting data sample into smaller subset. This is done by first doing the analysis on a subset, while the other subset is used for subsequent confirmation and validation of this analysis. The subset used for analysis is called the training set. The other subset is called validation set and generally separated from the testing set. Two cross-validation methods often used in practice are K -fold cross-validation (K-CV) and leave-one-out cross-validation (LOO-CV).

K-CV divides the n original data into K groups (generally evenly divided), makes each subset of data into a validation set once separately. The rest of the $K - 1$ subsets are considered as the training set, so K-CV will result in K models. In general, K is selected between 5 and 10. LOO-CV is essentially N-CV. The process of determining the number of principal components will be described in detail using LOO-CV as an example.

All n samples are divided into two parts: the first part is the set of all samples excluding a certain sample i (containing a total of $n - 1$ samples) and a regression equation is fitted with this data set using d principal components; The second part is to substitute the i th sample that was just excluded into the fitted regression equation to obtain the predicted value $\hat{y}_{(i)j}(d)$, $j = 1, 2, \dots, l$ of y_j . Repeating the above test for each $i = 1, 2, \dots, n$, the sum of squared prediction errors for y_j can be defined as $\text{PRESS}_j(d)$.

$$\text{PRESS}_j(d) = \sum_{i=1}^n (y_{ij} - \hat{y}_{(i)j}(d))^2, j = 1, 2, \dots, l. \quad (3.46)$$

The sum of squared prediction errors of $Y = (y_1, \dots, y_l)^T$ can be obtained as

$$\text{PRESS}(d) = \sum_{j=1}^l \text{PRESS}_j(d). \quad (3.47)$$

Obviously, if the robustness of the regression equation is not good, the error is large and thus it is very sensitive to change in the samples, and the effect of this perturbation error will increase the $\text{PRESS}(d)$ value.

On the other hand, use all sample points to fit a regression equation containing d components. In this case, the fitted value of the i th sample point is $\hat{y}_{ij}(d)$. The fitted error sum of squares for y_j is defined as $\text{SS}_j(d)$ value

$$\text{SS}_j(d) = \sum_{i=1}^n (y_{ij} - \hat{y}_{ij}(d))^2. \quad (3.48)$$

The sum of squared errors of Y is

$$\text{SS}(d) = \sum_{j=1}^l \text{SS}_j(d). \quad (3.49)$$

Generally, $\text{PRESS}(d)$ is greater than $\text{SS}(d)$ because $\text{PRESS}(d)$ contains an unknown perturbation error and the fitting error decreases with the increase of components, i.e., $\text{SS}(d)$ is less than $\text{SS}(d - 1)$. Next, compare $\text{SS}(d - 1)$ and $\text{PRESS}(d)$. $\text{SS}(d - 1)$ is the fitting error of the regression equation that is fitted with all samples with d components; $\text{PRESS}(d)$ contains the perturbation error of the samples but with one more component. If the d component regression equation with perturbation error can be somewhat smaller than the fitting error of the $d - 1$ component regression equation, it is considered that adding one component t_d will result in a significant improvement in prediction accuracy. Therefore, it is always expected that the ratio of $\frac{\text{PRESS}(d)}{\text{SS}(d-1)}$ is as small as possible. The general setting

$$\frac{\text{PRESS}(d)}{\text{SS}(d-1)} \leq (1 - 0.05)^2 = 0.95^2. \quad (3.50)$$

IF $\text{PRESS}(d) \leq 0.95^2 \text{SS}(d-1)$, the addition of the component is considered beneficial. And conversely, if $\text{PRESS}(d) > 0.95^2 \text{SS}(d-1)$, the new addition of components is considered to have no significant improvement in reducing the prediction error of the regression equation.

In practice, the following cross-validation index is used. For each dependent variable y_j , define

$$Q_{dj}^2 = 1 - \frac{\text{PRESS}_j(d)}{\text{SS}_j(d-1)}. \quad (3.51)$$

For the full dependent variable \mathbf{Y} , the cross-validation index of component t_d is defined as

$$Q_d^2 = 1 - \frac{\text{PRESS}(d)}{\text{SS}(d-1)}. \quad (3.52)$$

The marginal contribution of component t_d to the predictive accuracy of the regression model has the following two scales (cross-validation index).

- (1) $Q_d^2 > 1 - 0.95^2 = 0.0975$, the marginal contribution of t_d component is significant; and
- (2) For $k = 1, 2, \dots, l$, there is at least one k such that $Q_{dj}^2 > 0.0975$ holds, at which point the addition of component t_d leads to a significant improvement in the prediction accuracy of at least one dependent variable y_k . Therefore it can also be argued that adding component t_d is clearly beneficial.

References

- Chen Z, Ding SX, Zhang K, Li Z, Hu Z (2016) Canonical correlation analysis-based fault detection methods with application to alumina evaporation process. *Control Eng Pract* 46:51–58
- Chen Z, Zhang K, Ding SX, Shardt YAW, Hu Z (2016) Improved canonical correlation analysis-based fault detection methods for industrial processes. *J Process Control* 41:26–34
- Hair JF, Hult GTM, Ringle C, Sarstedt M (2016) A primer on partial least squares structural equation modeling (PLS-SEM). Sage Publications, Thousand Oaks
- Hotelling H (1936) Relations between two sets of variates. *Biom Trust* 28(3/4):321–377
- Hou B (2013) Research on canonical correlation analysis algorithm based on sparse representation. Nanjing University of Science and Technology, Nanjing
- Li W, Yang J, Zhang J et al (2015) CCA-based algorithm for personalized privacy preservation in trajectory. *J Jilin Univ (Eng Technol Ed)* 45:630–638
- Li C, Zhang K, Liu W et al (2020) A person re-identification method based on feature fusion of canonical correlation analysis. *J Optoelectron Laser* 31:500–508

- Okwuashi O, Ndehedehe C, Attai H (2020) Tide modeling using partial least squares regression. *Ocean Dyn* 70(8):1089–1101
- Ramin N, Werner Z, Edwin L, Susanne S (2018) Domain-invariant partial-least-squares regression. *Anal Chem* 90(11):6693–6701
- Zhang K, Dou J (2015) An image feature-matching algorithm based on CCA. *J Yunnan Natl Univ (Nat Sci Ed)* 24(3):244–247
- Wold S, Kettaneh-Wold N, Skagerberg B (1989) Nonlinear PLS modeling. *Chemom Intell Lab Syst* 7:53–65

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 4

Simulation Platform for Fault Diagnosis



The previous chapters have described the mathematical principles and algorithms of multivariate statistical methods, as well as the monitoring processes when used for fault diagnosis. In order to validate the effectiveness of data-driven multivariate statistical analysis methods in the field of fault diagnosis, it is necessary to conduct the corresponding fault monitoring experiments. Therefore this chapter introduces two kinds of simulation platform, Tennessee Eastman (TE) process simulation system and fed-batch Penicillin Fermentation Process simulation system. They are widely used as test platforms for the process monitoring, fault classification, and identification of industrial process. The related experiments based on PCA, CCA, PLS, and FDA are completed on the TE simulation platforms.

4.1 Tennessee Eastman Process

The original TE industrial process control problem was developed by Downs and Vogel in 1993. It is used for the open and challenging control-related topics including multi-variable controller design, optimization, adaptive and predictive control, non-linear control, estimation and identification, process monitoring and diagnosis, and education. TE process model is established according to the actual chemical process. It has been widely used as a benchmark for control and monitoring research process. Figure 4.1 shows the flow diagram of TE process with five major units: reactor, condenser, compressor, vaporliquid separator, and stripper. Four kinds of gaseous material *A*, *C*, *D*, and *E* are input for reaction. In addition, a small amount of inert gas *B* is contained besides the above feeds. The final products are three liquid including *G*, *H*, and *F*, where *F* is the by-product.

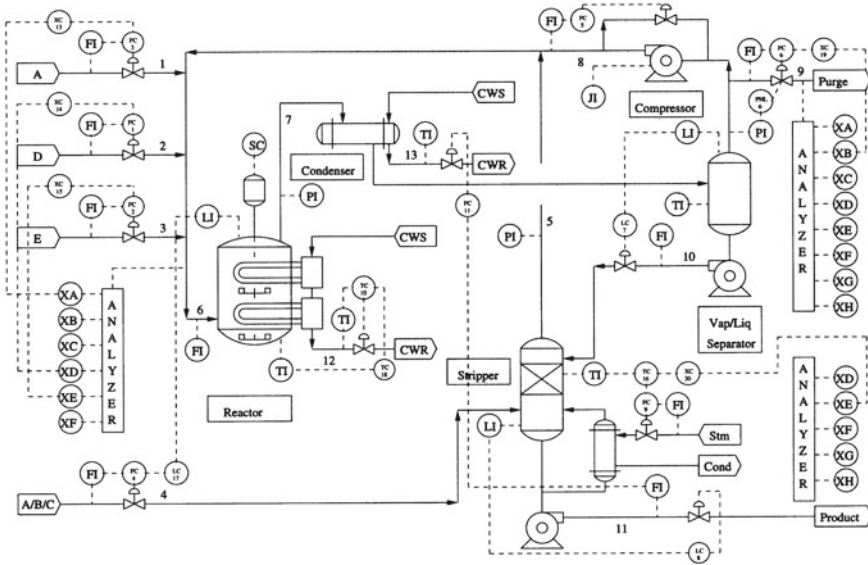
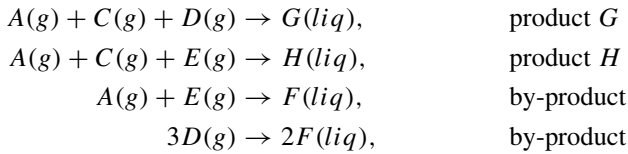


Fig. 4.1 Tennessee Eastman process



Briefly, TE process consists of two data modules: XMV module containing 12 manipulated variables (XMV(1)-XMV(12): $x_{23} - x_{34}$) and XMEAS module consisting of 22 process measured variables (XMEAS(1)-XMEAS(22): $x_1 - x_{22}$) and 19 component measured variables (XMEAS(23)-XMEAS(41): $x_{35} - x_{53}$), as listed in Tables 4.1 and 4.2.

In this book, the code provided is available on the website *online* at <http://depts.washington.edu/control/LARRY/TE/download.html>. Also, the code and data sets can be downloaded. The Simulink simulator allows an easy setting and generation of the operation modes, measurement noises, sampling time, and magnitudes of the faults. It is thus very helpful for the data-driven process monitoring study. 21 artificially disturbances (considered as faulty operations for fault diagnosis problem) in the TE process are shown in Table 4.3. In general, the entire TE data consists of training set and testing set, and each set includes 22 kinds of data under different simulation operations. Each kind of data has sampled measurements on 53 observed variables.

In the data set given in the web link above, d00.dat to d21.dat are training sets, and d00_te.dat to d21_te.dat are testing sets. d00.dat and d00_te.dat are samples under

Table 4.1 Monitoring variables in the TE process($x_1 - x_{34}$)

No.	Variable name	Units	No.	Variable name	Units
x_1	A feed (stream 1)	kscmh	x_{18}	Stripper temperature	°C
x_2	D feed (stream 2)	kg h^{-1}	x_{19}	Stripper steam flow	kg h^{-1}
x_3	E feed (stream 3)	kg h^{-1}	x_{20}	Compress work	KW
x_4	A and C feed (steam 4)	kscmh	x_{21}	Reactor cooling water outlet temperature	°C
x_5	Recycle flow (stream 8)	kscmh	x_{22}	Condenser cooling water outlet temperature	°C
x_6	Reactor feed rate (stream 6)	kscmh	x_{23}	D feed flow valve (stream 2)	%
x_7	Reactor pressure	kPa gauge	x_{24}	E feed flow valve (stream 3)	%
x_8	Reactor level	%	x_{25}	A feed flow valve (stream 1)	%
x_9	Reactor temperature	°C	x_{26}	A and C feed flow valve (stream 4)	%
x_{10}	Purge rate (stream 9)	kscmh	x_{27}	Compressor recycle valve	%
x_{11}	Product separator temperature	°C	x_{28}	Purge valve (stream 9)	%
x_{12}	Product separator level	%	x_{29}	Separator pot liquid flow valve (stream 10)	%
x_{13}	Product separator pressure	kPa gauge	x_{30}	Stripper liquid product flow valve (stream 11)	%
x_{14}	Product separator underflow (stream 10)	m^3h^{-1}	x_{31}	Stripper steam valve	%
x_{15}	Stripper level	%	x_{32}	Reactor cooling water flow valve	%
x_{16}	Stripper pressure	kPa gauge	x_{33}	Condenser cooling water flow valve	%
x_{17}	Stripper underflow (stream 11)	m^3h^{-1}	x_{34}	Agitator speed	

the normal operation conditions. The training samples of d00.dat are sampled under 25h running simulation. The total number of observations is 500. The d00_te.dat test samples are obtained under 48h running simulation, and the total number of observation data is 960. d01.dat–d21.dat (for training) and d01_te.dat–d21_te.dat (for testing) are sampled with different faults, in which the numerical label of the data set are corresponding to the fault type.

All the testing data set are obtained under 48h running simulation with the faults introduced at 8h. A total of 960 observations are collected, in which the first 160 observations are in the normal operation. It is worth to point out that the data sets

Table 4.2 Monitoring variables in the TE process($x_{35} - x_{53}$)

No.	Variable name	Stream	No.	Variable name	Stream
x_{35}	Composition <i>A</i>	6	x_{45}	Composition <i>E</i>	9
x_{36}	Composition <i>B</i>	6	x_{46}	Composition <i>F</i>	9
x_{37}	Composition <i>C</i>	6	x_{47}	Composition <i>G</i>	9
x_{38}	Composition <i>D</i>	6	x_{48}	Composition <i>H</i>	9
x_{39}	Composition <i>E</i>	6	x_{49}	Composition <i>D</i>	11
x_{40}	Composition <i>F</i>	6	x_{50}	Composition <i>E</i>	11
x_{41}	Composition <i>A</i>	9	x_{51}	Composition <i>F</i>	11
x_{42}	Composition <i>B</i>	9	x_{52}	Composition <i>G</i>	11
x_{43}	Composition <i>C</i>	9	x_{53}	Composition <i>H</i>	11
x_{44}	Composition <i>D</i>	9			

Table 4.3 Disturbances for the TE process

IDV	Process variable	Type
1	A/C feed ratio, B composition constant (stream 4)	Step
2	B composition, A/C feed 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
9	D feed temperature (stream 2)	Random
10	C feed temperature (stream 4)	Random
11	Reactor cooling water inlet temperature	Random
12	Condenser cooling water inlet temperature	Random
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown
21	Valve position (stream 4)	Constant

once generated by Leoand et al. (2001) is widely accepted for process monitoring and fault diagnosis research. The data sets are smoothed, filtered, and normalized. The monitored variables are variables $x_1 - x_{53}$.

4.2 Fed-Batch Penicillin Fermentation Process

Fed-batch fermentation processes are widely used in the pharmaceutical industry. The yield maximization is usually considered as the main goal in the batch fermentation processes. The different characteristics of batch operation from the continuous operation include strong nonlinearity, non-stationary conditions, batch-to-batch variability, and strong time-varying conditions. These features result that the yield is difficult to predict. Therefore, the fault detection, classification, and identification of batch/fed-batch processes shows more difficulties compared with the continuous TE process.

The model of fed-batch penicillin fermentation process is described by Birol et al. (2002)

$$\begin{aligned} X &= f(X, S, C_L, H, T) \\ S &= f(X, S, C_L, H, T) \\ C_L &= f(X, S, C_L, H, T) \\ P &= f(X, S, C_L, H, T, P) \\ CO_2 &= f(X, H, T) \\ H &= f(X, H, T), \end{aligned}$$

where X , S , C_L , P , CO_2 , H and T are biomass concentration, substrate concentration, dissolved oxygen concentration, penicillin concentration, carbon dioxide concentration, hydrogen ion concentration for pH ($[H^+]$), and temperature, respectively. The corresponding detailed mathematical model is given in Birol et al. (2002).

The research group with the Illinois Institute of Technology has developed a dynamic simulation of penicillin production based on an unstructured model, PenSim V2.0. This model has been used as a benchmark for statistical process monitoring studies of batch/fed-batch reaction process. The flow chart of the fermentation process is depicted in Fig. 4.2. The fermentation unit consists of a fermentation reactor and a coil-based heat exchange unit. The pH and temperature are automatically controlled by two PID controllers by adjusting the flow rates of acid/base and cold/hot water. The glucose substrate is fed continuously into the fermentation reactor in open-loop operation in the fed-batch operation mode.

Fourteen variables are considered in PenSim V2.0 model, shown in Table 4.4: 5 input variables (1–4, 14) and 9 process variables (5–13). Since variables 11–13 are not measured online in industry, only 11 variables are monitored here.

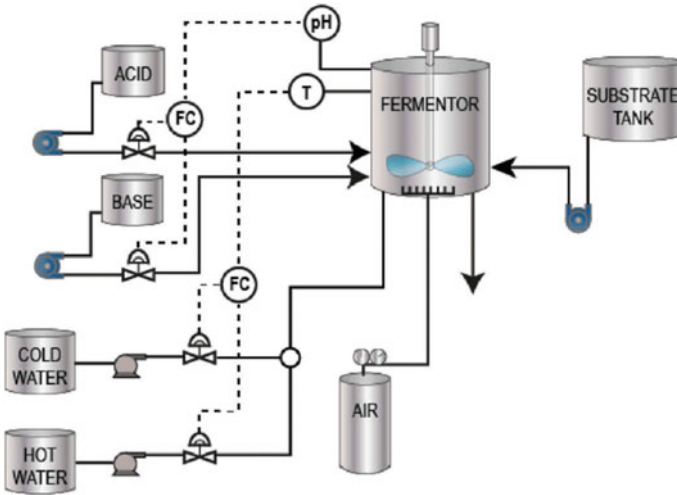


Fig. 4.2 Flow chart of the penicillin fermentation process

Table 4.4 Variables in penicillin fermentation process

No.	Variable
1	Aeration rate (L/h)
2	Agitator power input (W)
3	Substrate feed rate (L/h)
4	Substrate feed temperature (K)
5	Dissolved oxygen concentration (% saturation)
6	Culture volume (L)
7	Carbon dioxide concentration (mmol/L)
8	pH
9	Temperature in the bioreactor (K)
10	Generated heat (kcal/h)
11	Cooling water flow rate (L/h)
12	Penicillin concentration
13	Biomass concentration
14	Substrate concentration

4.3 Fault Detection Based on PCA, CCA, and PLS

This section tests the effectiveness of various multivariate statistical methods for the TE process. Faults in the standard TE data set are introduced at the 160 sampling. For comparison purposes, the normal operation data d00_te is chosen as to train the statistical model and faulty operation data d01_te-d21_te is used to test model and

detect fault. In the experiments for the PCA and PLS methods, the process variable matrix X consists of process variables (XMEAS (1–22)) and manipulated variables (XMV (1–11)). XMEAS (35) is used as the quality variable matrix Y for PLS. In the CCA experiment, the process variables (XMEAS (1–22)) are used as one data set, and the manipulated variables (XMV (1–11)) as another data set.

The fault detection rate (FDR) and false alarm rate (FAR) are defined as follows:

$$\begin{aligned} \text{FDR} &= \frac{\text{No.of samples}(J > J_{th}|f \neq 0)}{\text{total samples}(f \neq 0)} \times 100 \\ \text{FAR} &= \frac{\text{No.of samples}(J > J_{th}|f = 0)}{\text{total samples}(f = 0)} \times 100. \end{aligned} \quad (4.1)$$

Experiment and model parameters are determined as follows. The principal components of PCA are determined by the cumulative contribution of 90%. The number of principal components of PLS is selected as 6. T^2 and Q statistics are used to monitor process faults. It should be noted that in the monitoring of CCA, (3.18) and (3.19) are used as monitoring indices and the corresponding monitoring results are slightly different. For 21 fault types, the FDR for PCA, CCA, and PLS based on the control limit with 99% confidence level are shown in Table 4.5. It can be seen that the multivariate statistical methods listed in this section (including PCA, CCA, and PLS) can accurately detect the significant process faults.

Figures 4.3, 4.4, and 4.5 show the different monitoring results base on PCA, CCA, and PLS model for typical faults IDV(1), IDV(16), and IDV(20), respectively. Here, the black line is the statistic calculated from the real-time data and the red line is the normal statistic threshold from the offline model calculation.

It is easy to find that CCA has better detection for certain fault types from Table 4.5, such as faults IDV(10), IDV(16), IDV(19), and IDV(20). The monitoring results for faults IDV(16) and IDV(20) are shown in Figs. 4.4 and 4.5. Why does CCA show better detection capabilities than the other two methods in certain faults? Let's check the setting of process variable X for three methods. In contrast to PCA and PLS, CCA splits its X -space directly into two parts and extracts the latent variables by examining the correlation between these two parts, i.e., the latent variables extracted by CCA can better characterise the changes in the process.

4.4 Fault Classification Based on FDA

To further test the effectiveness of fault classification, samples from the 161th to the 700th of the 21 fault data sets and the normal data sets are used for training FDA model. The corresponding data from the 701th to the 960th samples are used to test FDA model and its classification ability. FDA in Sect. 2.2 is a classical method to validate the classification effect and identify the fault types. The following distance metric index is introduced to further quantify the difference between different faults:

Table 4.5 FDRs of PCA, CCA and PLS

IDV	PCA		CCA		PLS	
	T ²	SPE	T ₁ ²	T ₂ ²	T ²	SPE
1	99.13	99.88	99.38	99.63	99.75	99.38
2	98.38	95.13	95.63	96.13	98.63	97.75
3	1.00	3.00	0.25	0.50	3.75	1.88
4	50.88	99.88	100.00	97.38	40.63	96.88
5	23.75	23.88	100.00	100.00	25.50	25.88
6	99.00	100.00	100.00	100.00	99.25	100.00
7	100.00	100.00	100.00	83.00	99.13	100.00
8	97.00	86.25	87.00	92.25	96.88	96.75
9	1.50	2.00	0.13	0.13	2.13	2.25
10	27.88	36.13	78.75	79.38	57.00	31.25
11	52.50	61.63	77.00	56.88	41.88	65.75
12	98.38	90.25	97.00	99.00	99.00	96.75
13	93.75	95.13	94.38	94.25	95.50	94.25
14	99.88	98.88	100.00	99.88	99.88	100.00
15	1.25	2.00	0.63	0.75	4.50	1.13
16	12.13	36.25	85.00	86.63	29.75	19.25
17	79.50	95.88	91.38	95.25	80.13	89.75
18	89.13	90.50	89.50	89.50	89.50	89.50
19	11.63	16.50	84.38	84.25	1.63	13.38
20	31.13	52.75	70.38	75.50	41.75	45.38
21	41.25	48.75	26.63	36.88	56.38	43.00

$$D_2 = \left\| \text{FDA}_i - \text{FDA}_j \right\| ,$$

where FDA_i denotes the FDA feature vector of the i th fault.

The simulation results are shown in Fig. 4.6. The 22 kinds of data (including the normal operation and 21 faulty operation) can be roughly divided into two major categories: the first category is the faults that are significantly different from other faults, which contains faults IDV(2) (line with \diamond), IDV(6) (line with $*$), and IDV(18) (line with \circ); the other category is the set of faults whose characteristics are relatively close to each other.

The faults IDV(1), IDV(2), IDV(6), and IDV(20) are further analyzed. The FDA results for fault classification are shown in Fig. 4.7. The D_2 indices for these faults vary considerably, as the classification results clearly illustrated. Conversely, certain faults have very small differences in D_2 indices. For example, faults IDV(4), IDV(11), and IDV(14) have the similar FDA D_2 indices, shown in Fig. 4.8. These faults are difficult to classify accurately based on FDA model, as shown in Fig. 4.9.

Fig. 4.3 PCA, CCA, and PLS monitoring results for IDV(1)

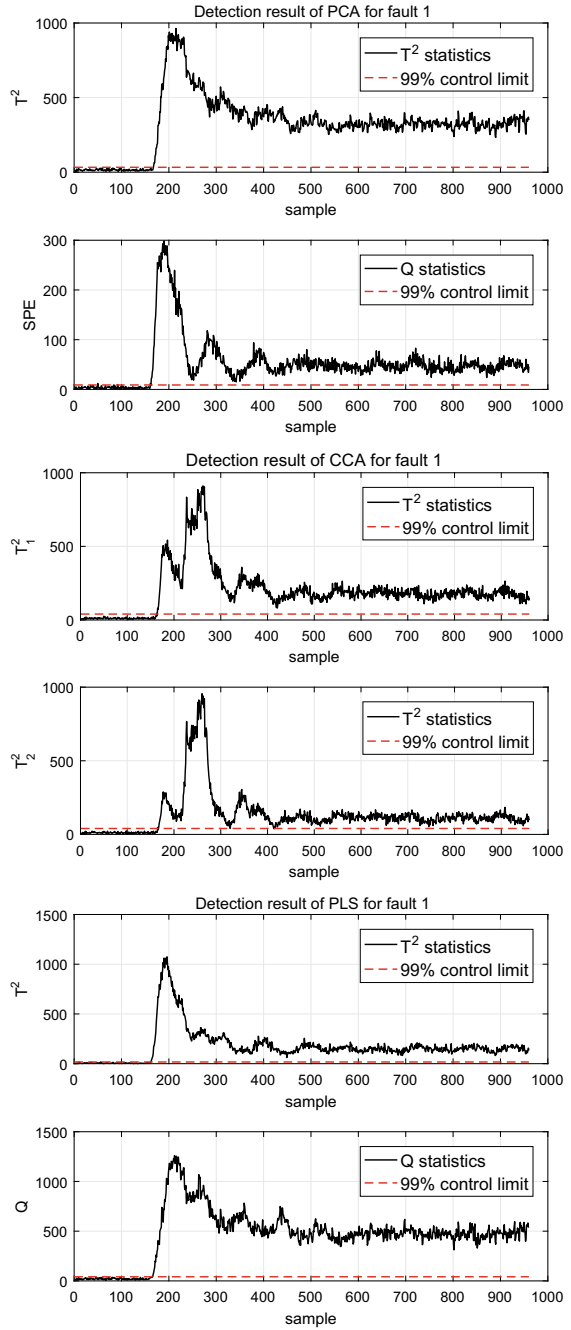


Fig. 4.4 PCA, CCA, and PLS monitoring results for IDV(16)

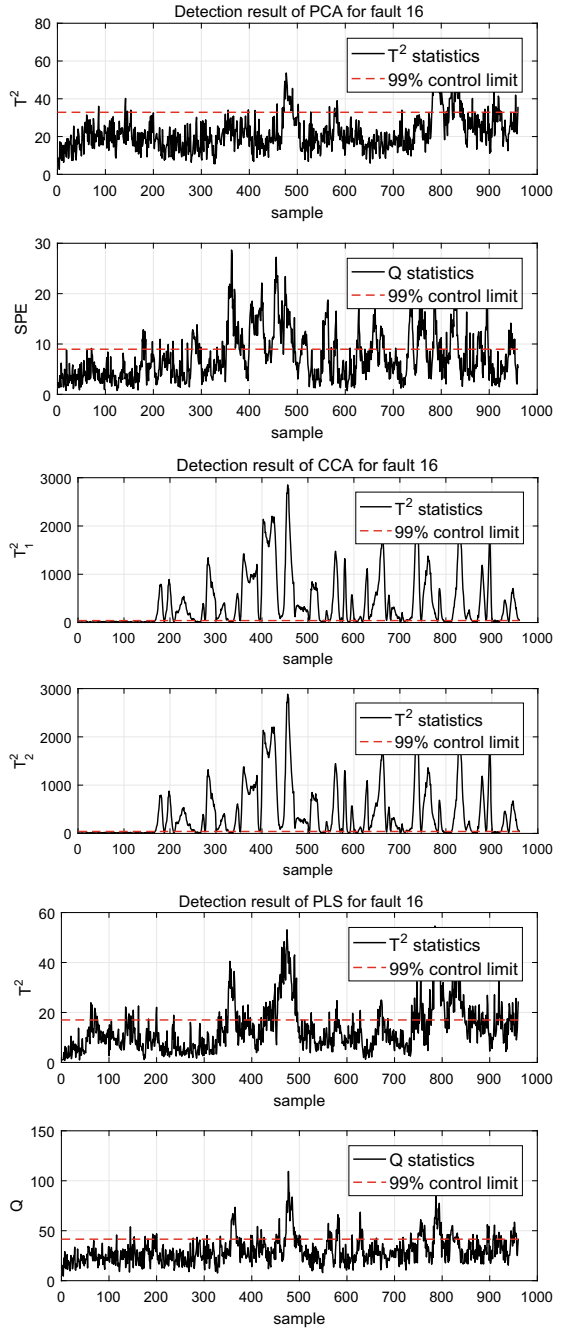
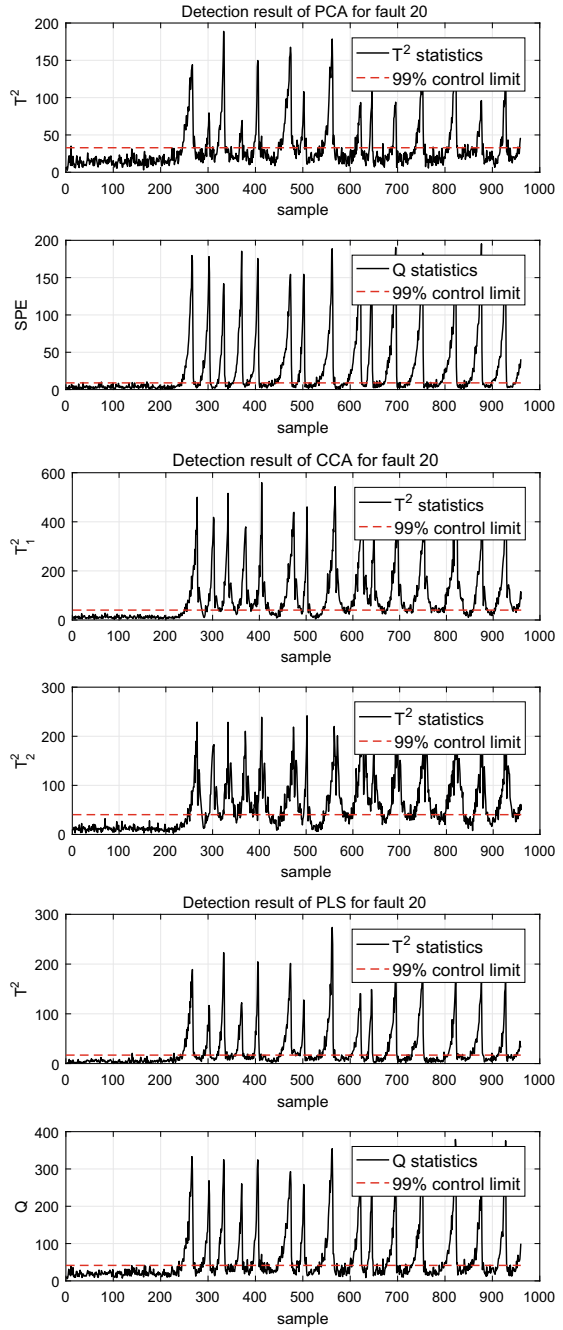


Fig. 4.5 PCA, CCA, and PLS monitoring results for IDV(20)



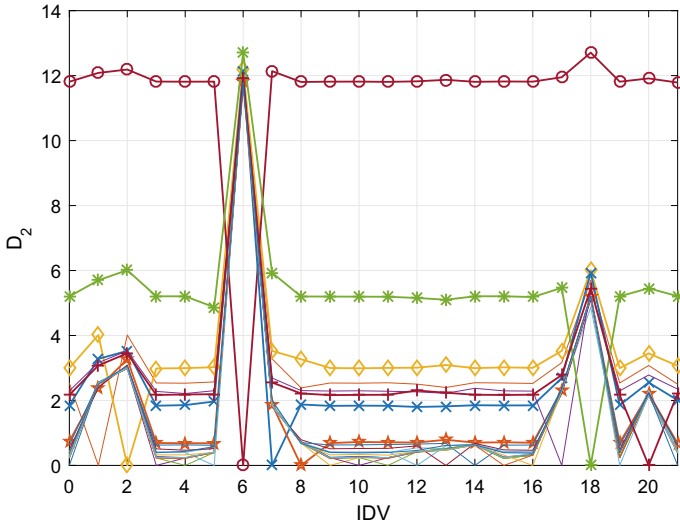


Fig. 4.6 D_2 index for different faults

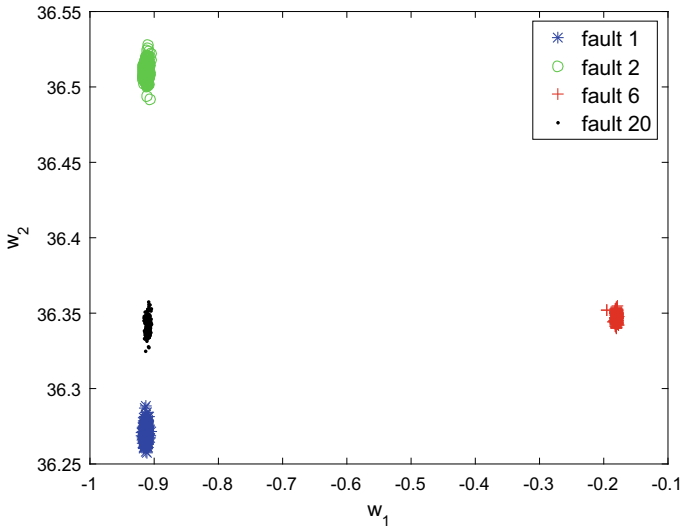


Fig. 4.7 FDA identification result for the fault 1, 2, 6, and 20

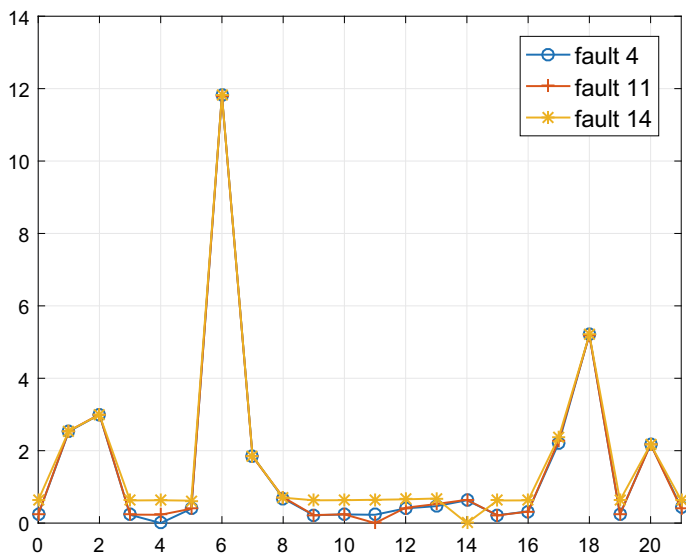


Fig. 4.8 D_2 indices for fault 4, 11, and 14

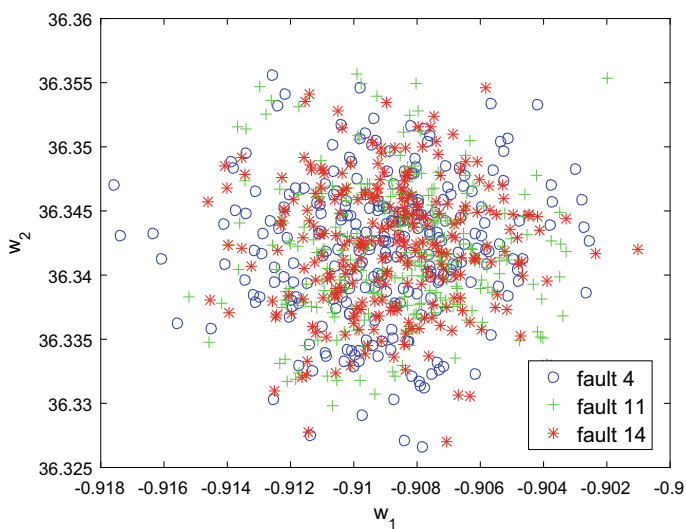


Fig. 4.9 FDA identification result for the fault 4, 11, and 14

4.5 Conclusions

Two kinds of simulation platforms are introduced for verifying the statistical monitoring methods and several experiments based on the traditional methods, PCA, PLS, CCA, and FDA, are finished. These basic experiments illustrate the characteristics of several methods and their fault detection effects. Actually, there are lots of improved methods to overcome the shortcomings and deficiencies of the original multivariate statistical analysis methods. Each method has its own conditions and scope of application. No one method completely outperforms the others in terms of performance. Furthermore, data-based fault detection methods need to be combined with the actual monitoring objects, and existing methods need to be improved according to its knowledge and characteristics. So this book focus on the fault detection (discrimination) strategies for batch processes and strong nonlinear systems.

References

- Birol G, Undey C, Cinar A (2002) A modular simulation package for fed-batch fermentation: penicillin production. *Comput Chem Eng*, pp 1553–1565
- Leoand M, Russell E, Braatz R (2001) *Tennessee eastman process*. Springer, London

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 5

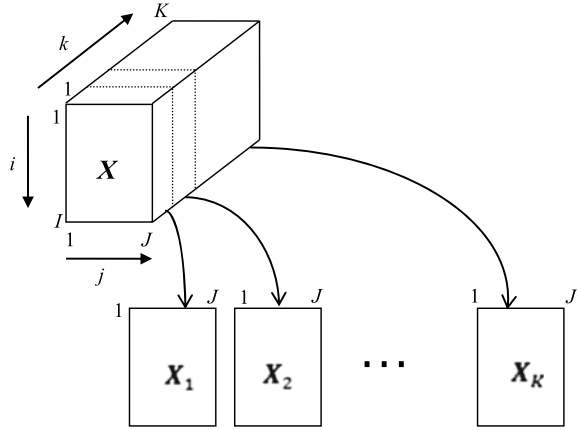
Soft-Transition Sub-PCA Monitoring of Batch Processes



Batch or semi-batch processes have been utilized to produce high-value-added products in the biological, food, semi-conductor industries. Batch process, such as fermentation, polymerization, and pharmacy, is highly sensitive to the abnormal changes in operating condition. Monitoring of such processes is extremely important in order to get higher productivity. However, it is more difficult to develop an exact monitoring model of batch processes than that of continuous processes, due to the common natures of batch process: non-steady, time-varying, finite duration, and nonlinear behaviors. The lack of exact monitoring model in most batch processes leads that an operator cannot identify the faults when they occurred. Therefore, effective techniques for monitoring batch process exactly are necessary in order to remind the operator to take some corrective actions before the situation becomes more dangerous.

Generally, many batch processes are carried out in a sequence of steps, which are called multi-stage or multi-phase batch processes. Different phases have different inherent natures, so it is desirable to develop stage-based models that each model represents a specific stage and focuses on a local behavior of the batch process. This chapter focuses on the monitoring method based on multi-phase models. An improved online sub-PCA method for multi-phase batch process is proposed. A two-step stage dividing algorithm based on support vector data description (SVDD) technique is given to divide the multi-phase batch process into several operation stages reflecting their inherent process correlation nature. Mechanism knowledge is considered firstly by introducing the sampling time into the loading matrices of PCA model, which can avoid segmentation mistake caused by the fault data. Then SVDD method is used to strictly refine the initial division and obtain the soft-transition sub-stage between the stable and transition periods. The idea of soft-transition is helpful for further improving the division accuracy. Then a representative model is built for each sub-stage, and an online fault monitoring algorithm is given based on the division techniques above. This method can detect fault earlier and avoid false alarm

Fig. 5.1 Batch-wise unfolding



because of more precise stage division, comparing with the conventional sub-PCA method.

5.1 What Is Phase-Based Sub-PCA

The general monitoring for batch process is phase/stage-based sub-PCA method, which divides the process into several phases (Yao and Gao 2009). The phase-based sub-PCA consists of three steps: **data matrix unfolding**, **phase division**, and **sub-PCA modeling**. Now the details of them are introduced.

1. Data Matrix Unfolding

Different from the continuous process, the historical data of batch process are composed of a three-dimensional array $X(I \times J \times K)$, where I is the number of batches, J is the number of variables, and K is the number of sampling times. The original data X should be conveniently rearranged into two-dimensional matrices prior to developing statistical models. Two traditional methods are widely applied: the batch-wise unfolding and the variable-wise unfolding, with the most used method is batch-wise unfolding. The three-dimensional matrix X should be cut into K time-slice matrix after the batch-wise unfolding is completed.

The three-dimensional process data $X(I \times J \times K)$ is batch-wise unfolded into two-dimensional forms $X_k(I \times J)$, ($k = 1, 2, \dots, K$). Then a time-slice matrix is placed beneath one another, but not beside as shown in Fig. 5.1 (Westerhuis et al. 1999; Wold et al. 1998). Sometimes batches have different lengths, i.e. the sampling number K are different. The process data need to be aligned before unfolding. There are many data alignment methods raised by former researchers, such as directly filling zeros to missing sampling time (Arteaga and Ferrer 2002), dynamic time warping (Kassida et al. 1998). These unfolding approaches do not require any estimation of unknown future data for online monitoring.

2. Phase Division

The traditional multivariate statistical analysis methods are valid in the continuous process, since all variables are supposed to stay around certain stable state and the correlation between these variables remains relatively stable. Non-steady-state operating conditions, such as time-varying and multi-phase behavior, are the typical characteristics in a batch process. The process correlation structure might change due to process dynamics and time-varying factors. The statistical model may be ill-suited if it takes the entire batch data as a single object, and the process correlation among different stages are not captured effectively. So multi-phase statistic analysis aims at employing the separate model for the forthcoming period, instead of using a single model for the entire process. The phase division plays a key role in batch process monitoring.

Many literature divided the process into multi-phase based on mechanism knowledge. For example, the division is based on different processing units or distinguishable operational phases within each unit (Dong and McAvoy 1996; Reinikainen and Hoskuldsson 2007). It is suggested that process data can be naturally divided into groups prior to modelling and analysis. This stage division directly reflects the operational state of the process. However, the known prior knowledge usually are not sufficient to divide processes into phases reasonably. Besides, Muthuswamy and Srinivasan identified several division points according to the process variable features described in the form of multivariate rules (Muthuswamy and Srinivasan 2003). Undey and Cinar used an indicator variable that contained significant landmarks to detect the completion of each phase (Undey and Cinar 2002). Doan and Srinivasan divided the phases based on the singular points in some known key variables (Doan and Srinivasan 2008). Kosanovich, Dahl, and Piovoso pointed out that the changes in the process variance information explained by principal components could indicate the division points between the process stages (Kosanovich and Dahl 1996). There are many results in this area but not give a clear strategy to distinct the steady phase and transition phase (Camacho and Pico 2006; Camacho et al. 2008; Yao and Gao 2009).

3. Sub-PCA Modeling

The statistical models are constructed for all the phases after the phase division and are not limited to PCA methods. Here, sub-PCA is representatively one of these sub-statistical monitoring methods. The final sub-PCA model of each phase is calculated by taking the average of the time-slice PCA models in the corresponding phase. The number of principal components of each phase are determined based on the relative cumulative variance.

The T^2 , SPE statistics and their corresponding control limits are calculated according to the sub-PCA model. Check the Euclidean distance of the new data from the center of each stage of clustering and determine at which stage the new data is located. Then, the corresponding sub-PCA model is used to monitor the new data. Fault warning is pointed according to the control limits of T^2 or SPE .

5.2 SVDD-Based Soft-Transition Sub-PCA

Industrial batch process operates in a variety of status, including grade changes, startup, shutdown, and maintenance operations. Transitional region between neighboring stages is very common in multistage process, which shows the gradual changeover from one operation pattern to another. Usually the transitional phases first show basic characteristic that are more similar to the previous stable phase and then more similar to the next stable phase at the end of the transition. The different transition phases undergo different trajectories from one stable mode to another, with change in characteristics that are more pronounced in sampling time and more complex than those within a phase. Therefore, valid process monitoring during transitions is very important. Up to now, few investigations about transition modeling and monitoring have been reported (Zhao et al. 2007). Here, a new transition identification and monitoring method base on the SVDD division method is proposed.

5.2.1 Rough Stage-Division Based on Extended Loading Matrix

The original three-dimensional array $X(I \times J \times K)$ is first batch-wise unfolded into two-dimensional form X_k . By subtracting the grand mean of each variable over all time and all batches, unfolding matrix X_k is centered and scaled.

$$X_k = \frac{[X_k - \text{mean}(X_k)]}{\sigma(X_k)}, \quad (5.1)$$

where $\text{mean}(X_k)$ and $\sigma(X_k)$ represent the mean value and the standard variance of matrix X_k , respectively. The main nonlinear and dynamic components of every variable are still left in the scaled matrix.

Suppose the unfolding matrix at each time-slice is X_k . Project it into the principle component subspace by loading matrix P_k to obtain the scores matrix T_k :

$$X_k = T_k P_k^T + E_k, \quad (5.2)$$

where E_k is the residual. The first few components in PCA which represent major variation of original data set X_k are chosen. The original data set X_k is divided into the score matrix $\hat{X}_k = T_k P_k^T$ and the residual matrix E_k . Here, \hat{X}_k is PCA model prediction. Some useful techniques, such as the cross-validation, have been used to determine the most appropriate retained numbers of principal components. Then the loading matrix P_k and singular value matrix S_k of each time-slice matrix X_k can be obtained.

As the loading matrix \mathbf{P}_k reflects the correlations of process variables, it usually is used to identify the process stage. Sometimes disturbances brought by measurement noise or other reasons will lead wrong division, because the loading matrix just obtained from process data is hard to distinguish between wrong data and transition phase data. Generally, different phases in the batch process could be firstly distinguished according to the mechanism knowledge.

The sampling time is added to the loading matrix on order to divide the process exactly. The sampling time is a continuously increasing data set, so it must also be centered and scaled before added to the loading matrix. Generally, the sampling time is centered and scaled not along the batch dimension like process data \mathbf{X} , but along the time dimension in one batch. Then the scaling time t_k is changed into a vector \mathbf{t}_k by multiplying unit column vector. So the new time-slice matrix is written as $\hat{\mathbf{P}}_k = [\mathbf{P}_k, \mathbf{t}_k]$, in which \mathbf{t}_k is a $1 \times J$ column vector with repeated value of current sampling time. The sampling time will not change too much with the ongoing of batch process, but have an obvious effect on the phase separation. Define the Euclidean distance of extended loading matrix $\hat{\mathbf{P}}_k$ as

$$\begin{aligned} \|\hat{\mathbf{P}}_i - \hat{\mathbf{P}}_j\|^2 &= [\mathbf{P}_i - \mathbf{P}_j, \mathbf{t}_i - \mathbf{t}_j] [\mathbf{P}_i - \mathbf{P}_j, \mathbf{t}_i - \mathbf{t}_j]^T \\ &= \|\mathbf{P}_i - \mathbf{P}_j\|^2 + \|\mathbf{t}_i - \mathbf{t}_j\|^2. \end{aligned} \quad (5.3)$$

Then the batch process can be divided into S_1 stages using K -means clustering method to cluster the extended loading matrices $\hat{\mathbf{P}}_k$.

Clearly, the Euclidean distance of the extended loading matrix $\hat{\mathbf{P}}_i$ includes both data differences and sampling time differences. The data at different stages differ significantly in sampling time. Therefore, when noise interference makes the data at different stages present the same or similar characteristics, the large differences in sampling times will keep the final Euclidean distance at a large value. This is because the erroneous division data is very different in sampling time from the data from the other stages, while the data from the transition stage has very little variation in sampling time. We can easily distinguish erroneous divisions in the transition phase from those caused by noise.

5.2.2 Detailed Stage-Division Based on SVDD

The extended time-slice loading matrices $\hat{\mathbf{P}}_k$ represent the local covariance information and underlying process behavior as mentioned before, so they are used in determining the operation stages by proper analyzing and clustering procedures. The process is divided into different stages and each separated process stage contains a series of successive samples. Moreover, the transition stage is unsuitable to be forcibly incorporated into one steady stage because of its variation complexity of process characteristics. The transiting alteration of process characteristics imposes disadvantageous effects on the accuracy of stage-based sub-PCA monitoring mod-

els. Furthermore, it deteriorates fault detecting performance if just a steady transition sub-PCA model is employed to monitor the transition stage. Consequently, a new method based on SVDD is proposed to separate the transition regions after the rough stage-division which is determined by the K -means clustering.

SVDD is a relatively new data description method, which is originally proposed by Tax and Duin for the one-class classification problem (Tax and Duin 1999, 2004). SVDD has been employed for damage detection, image classification, one-class pattern recognition, etc. Recently, it has also been applied in the monitoring of continuous processes. However, SVDD has not been used for batch process phase separating and recognition up to now.

The loading matrix of each stage is used to train the SVDD model of transition process. SVDD model first maps the data from original space to feature space by a nonlinear transformation function, which is called as kernel function. Then a hypersphere with minimum volume can be found in the feature space. To construct such a minimum volume hypersphere, the following optimization problem is obtained:

$$\begin{aligned} \min \varepsilon(R, A, \xi) &= R^2 + C \sum_i \xi_i \\ \text{s.t. } \left\| \hat{\mathbf{P}}_i - \mathbf{A} \right\|^2 &\leq R^2 + \xi_i, \xi_i \geq 0, \forall i, \end{aligned} \quad (5.4)$$

where R and A are the radius and center of hypersphere, respectively, C gives the trade-off between the volume of the hypersphere and the number of error divides. ξ_i is a slack variable which allows a probability that some of the training samples can be wrongly classified. Dual form of the optimization problem (5.4) can be rewritten as

$$\begin{aligned} \min \sum_i \alpha_i K(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_i) - \sum_{i,j} \alpha_i \alpha_j K(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_j) \\ \text{s.t. } 0 \leq \alpha_i \leq C_i, \end{aligned} \quad (5.5)$$

where $K(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_j)$ is the kernel function, and α_i is the Lagrange multiplier. Here, Gaussian kernel function is selected as kernel function. General quadratic programming method is used to solve the optimization question (5.5). The hypersphere radius R can be calculated according to the optimal solution α_i :

$$R^2 = 1 - 2 \sum_{i=1}^n \alpha_i K(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_i) + \sum_{i=1, j=1}^n \alpha_i \alpha_j K(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_j) \quad (5.6)$$

Here, the loading matrices $\hat{\mathbf{P}}_k$ are corresponding to nonzero parameter α_k . It means that they have effect on the SVDD model. Then the transition phase can be distinguished from the steady phase by inputting all the time-slice matrices $\hat{\mathbf{P}}_k$ into SVDD model. When a new data $\hat{\mathbf{P}}_{new}$ is available, the hyperspace distance from the new data to the hypersphere center should be calculated firstly

$$D^2 = \left\| \hat{\mathbf{P}}_{new} - \boldsymbol{\alpha} \right\|^2 = 1 - 2 \sum_{i=1}^n \alpha_i K \left(\hat{\mathbf{P}}_{new}, \hat{\mathbf{P}}_i \right) + \sum_{i=1, j=1}^n \alpha_i, \alpha_j K \left(\hat{\mathbf{P}}_i, \hat{\mathbf{P}}_j \right). \quad (5.7)$$

If the hyperspace distance is less than the hypersphere radius, i.e., $D^2 \leq R^2$, the process data $\hat{\mathbf{P}}_{new}$ belongs to steady stages; else (that is $D^2 > R^2$), the data will be assigned to transition stages. The whole batch is divided into S_2 stages at the detailed division, which includes S_1 steady stages and $S_2 \rightarrow S_1$ transition stages.

The mean loading matrix $\bar{\mathbf{P}}_s$ can be adopted to get sub-PCA model of s th stage because the time-slice loading matrices in one stage are similar. $\bar{\mathbf{P}}_s$ is the mean matrix of the loading matrices \mathbf{P}_k in s th stage. The principal components number a_s can be obtained by calculating the relative cumulative variance of each principal component until it reaches 85%. Then the mean loading matrix is modified according to the obtained principal components. The sub-PCA model can be described as

$$\begin{cases} \mathbf{T}_k = \mathbf{X}_k \bar{\mathbf{P}}_s \\ \bar{\mathbf{X}}_k = \mathbf{T}_k \bar{\mathbf{P}}_s^T \\ \bar{\mathbf{E}}_k = \mathbf{X}_k - \bar{\mathbf{X}}_k. \end{cases} \quad (5.8)$$

The T^2 and SPE statistic control limits are calculated:

$$\begin{aligned} T_{\alpha, s, i}^2 &\sim \frac{a_{s, i}(I-1)}{(I-a_{s, i})} F_{a_{s, i}, I-a_{s, i}, \alpha} \\ \text{SPE}_{k, \alpha} &= g_k \Xi_{h_k, \alpha}^2, \quad g_k = \frac{v_k}{2m_k}, \quad h_k = \frac{2m_k^2}{v_k}, \end{aligned} \quad (5.9)$$

where m_k and v_k are the mean and variance of all batches data at time k , respectively, $a_{s, i}$ is the number of retained principal components in batch i ($i = 1, 2, \dots, I$), and stage s . I is the number of batches, α is the significant level.

5.2.3 PCA Modeling for Transition Stage

Now a soft-transition multi-phase PCA modeling method based on SVDD is presented according to the mentioned above. It uses the SVDD hypersphere radius to determine the range of transition region between two different stages. Meanwhile, it introduces a concept of membership grades to evaluate quantitatively the similarity between current sampling time data and transition (or steady) stage models. The sub-PCA models for steady phases and transition phases are established respectively which greatly improve the accuracy of models. Moreover, they reflect the characteristic changing during the different neighboring stages. Time-varying monitoring models in transition regions are established relying on the concept of membership grades, which are the weighted sum of nearby steady phase and transition phase sub-

models. Membership grade values are used to describe the partition problem with ambiguous boundary, which can objectively reflect the process correlations changing from one stage to another.

Here, the hyperspace distance $D_{k,s}$ is defined from the sampling data at time k to the center of the s th SVDD sub-model. It is used as dissimilarity index to evaluate quantitatively the changing trend of process characteristics. Correlation coefficients $\lambda_{l,k}$ are given as the weight of soft-transition sub-model, which are defined, respectively, as

$$\left\{ \begin{array}{l} \lambda_{s-1,k} = \frac{D_{k,s} + D_{k,s+1}}{2(D_{k,s-1} + D_{k,s} + D_{k,s+1})} \\ \lambda_{s,k} = \frac{D_{k,s-1} + D_{k,s+1}}{2(D_{k,s-1} + D_{k,s} + D_{k,s+1})} \\ \lambda_{s+1,k} = \frac{D_{k,s-1} + D_{k,s}}{2(D_{k,s-1} + D_{k,s} + D_{k,s+1})} \end{array} \right. \quad (5.10)$$

where $l = s - 1, s$, and $s + 1$ is the stage number, which represent the last steady stage, current transition stage, and next steady stage, respectively. The correlation coefficient is inverse proportional to hyperspace distance. The greater the distance, the smaller the effect of the hyperspatial distance. The monitoring model for the transition phase of each time interval can be obtained from the weighted sum of the sub-PCA models, i.e.,

$$\mathbf{P}'_k = \sum_{l=s-1}^{s+1} \lambda_{l,k} \bar{\mathbf{P}}_l. \quad (5.11)$$

The soft-transition PCA model in (5.11) properly reflects the time-varying transiting development. The score matrix \mathbf{T}'_k and the covariance matrix \mathbf{S}'_k can be obtained at each time instance. The SPE statistic control limit is still calculated by (5.9). Different batches have some differences in transition stages. The average T^2 limits for all batches are used to monitor the process in order to improve the robustness of the proposed method. The T^2 statistical control limits can be calculated from historical batch data and correlation coefficients.

$$\mathbf{T}^{2'}_{\alpha} = \sum_{l=s-1}^{s+1} \sum_{i=1}^I \lambda_{l,i,k} \frac{\mathbf{T}^2_{\alpha_{s,i}}}{I}, \quad (5.12)$$

where i ($i = 1, 2, \dots, I$) is the batch number, $\mathbf{T}^2_{\alpha_{s,i}}$ is the sub-stage T^2 statistic control limit of each batch which is calculated by (5.9) for sub-stage s .

Now the soft-transition model of each time interval in transition stages is obtained. The batch process can be monitored efficiently by combining with the steady stage model given in Sect. 5.2.2.

5.2.4 Monitoring Procedure of Soft-Transition Sub-PCA

The whole batch process has been divided into several steady stages and transition stage after the two steps stage-dividing, shown in Sects. 5.2.1 and 5.2.2. The new soft-transition sub-PCA method is applied to get detailed sub-model shown Sect. 5.2.3. The details of modeling steps are given as follows:

- (1) Get normal process data of I batches, unfold them into two-dimensional time-slice matrix, then center and scale each time-slice data as (5.1).
- (2) Perform PCA on the normalized matrix of each time-slice and get the loading matrices \mathbf{P}_k , which represent the process correlation at each time interval. Add sampling time t into the loading matrix to get the extended matrices $\hat{\mathbf{P}}_k$.
- (3) Divide the process into S_1 stages roughly using k -means clustering on extended loading matrices $\hat{\mathbf{P}}_k$. Train the SVDD classifier for the original S_1 steady process stages.
- (4) Input again the extended loading matrices $\hat{\mathbf{P}}_k$ into the original SVDD model to divide explicitly the process into S_2 stages: the steady stage and the transition stage. Then retrain the SVDD classifier for these new S_2 stages. The mean loading matrix $\bar{\mathbf{P}}_s$ of each new steady stage should be calculated and the sub-PCA model is built in (5.8). The correlation coefficients $\lambda_{l,k}$ are calculated to get the soft-transition stage model \mathbf{S}'_k in (5.11) for transition stage t .
- (5) Calculate the control limits of SPE and T^2 to monitor new process data.

The whole flowchart of improved sub-PCA modeling based on SVDD soft-transition is shown in Fig. 5.2. The modeling process is offline, which is depending on the historical data of I batches.

The following steps should be adopted during online process monitoring.

- (1) Get a new sampling time-slice data \mathbf{x}_{new} , center and scale it based on the mean and standard deviation of prior normal I batches data.
- (2) Calculate the covariance matrix $\mathbf{x}_{new}^T \mathbf{x}_{new}$, the loading matrix \mathbf{P}_{new} can be obtained based on singular value decomposition. Then add sampling time t_{new} into it to obtain the extended matrix $\hat{\mathbf{P}}_{new}$. Input the new matrix $\hat{\mathbf{P}}_{new}$ into the SVDD model to identify which stages the new data belongs to.
- (3) If current time-slice data belongs to a transition stage, the weighted sum loading matrix \mathbf{P}'_{new} is employed to calculate the score vector \mathbf{t}_{new} and error vector \mathbf{e}_{new} ,

$$\begin{aligned} \mathbf{t}_{new} &= \mathbf{x}_{new} \mathbf{P}'_{new} \\ \mathbf{e}_{new} &= \mathbf{x}_{new} - \bar{\mathbf{x}}_{new} = \mathbf{x}_{new} \left(\mathbf{I} - \mathbf{P}'_{new} \mathbf{P}'_{new}{}^T \right) \end{aligned} \quad (5.13)$$

Or if it belongs to a steady one, the mean loading matrix $\bar{\mathbf{P}}_s$ would be used to calculate the score vector \mathbf{t}_{new} and error vector \mathbf{e}_{new} ,

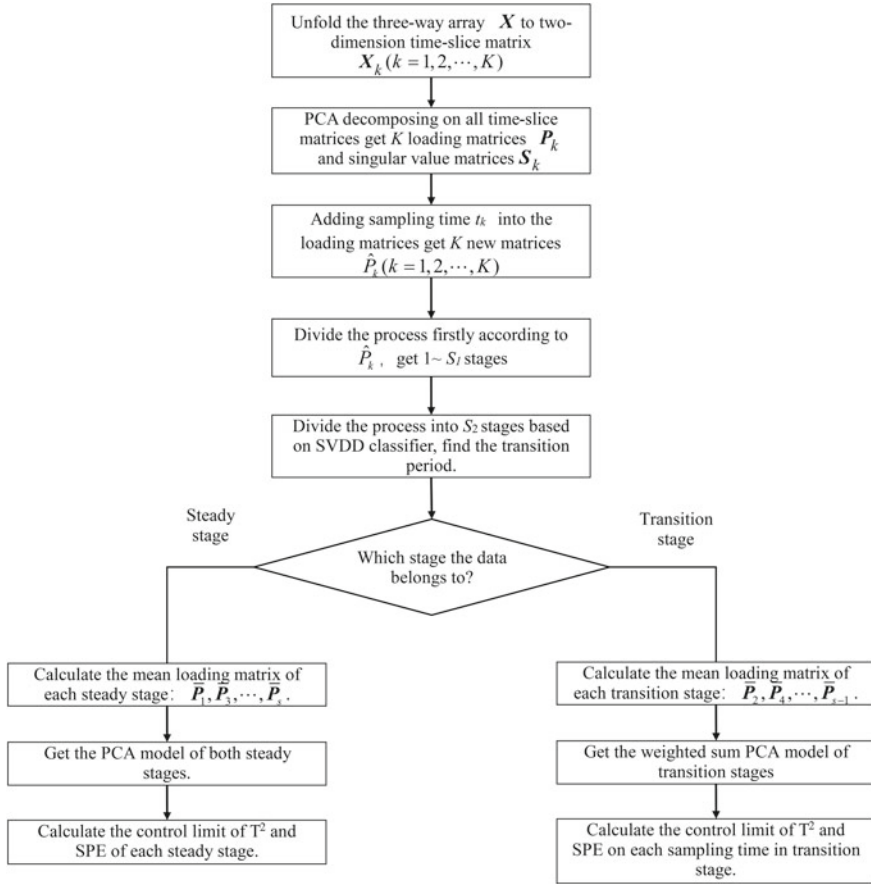


Fig. 5.2 Illustration of soft-transition sub-PCA modeling

$$\begin{aligned}
 t_{new} &= x_{new} \bar{P}_s \\
 e_{new} &= x_{new} - \bar{x}_{new} = x_{new} \left(I - \bar{P}_s \bar{P}_s^T \right).
 \end{aligned} \tag{5.14}$$

(4) Calculate the SPE and T^2 statistics of current data as follows:

$$\begin{aligned}
 T_{new}^2 &= t_{new} \bar{S}_s t_{new}^T \\
 SPE_{new} &= e_{new} e_{new}^T.
 \end{aligned} \tag{5.15}$$

(5) Judge whether the SPE and T^2 statistics of current data exceed the control limits. If one of them exceeds the control limit, alarm abnormal; if none of them does, the current data is normal.

5.3 Case Study

5.3.1 Stage Identification and Modeling

The Fed-Batch Penicillin Fermentation Process is used as a simulation case in this section. A detailed description of the Fed-Batch Penicillin Fermentation Process is available in Chap. 4. A reference data set of 10 batches is simulated under nominal conditions with small perturbations. The completion time is 400 h. All variables are sampled every 1 h so that one batch will offer 400 sampling data.

The rough division result based on K-mean method is shown in Fig. 5.3. Originally, the batch process is classified into 3 steady stage, i.e. $S_1 = 3$. Then SVDD classifier with Gaussian kernel function is used here for detailed division. The hypersphere radius of original 3 stages is calculated, and the distances from each sampling data to the hypersphere center are shown in Fig. 5.4.

As can be seen from the Fig. 5.4, the sampling data between two stages, such as the data during the time interval 28–42 and 109–200, are obviously out of the hypersphere. That means the data at this two time regions have significant difference from that of other steady stage. Therefore, these two stages are considered as transition stage. The process was further divided into 5 stages according to the detailed SVDD division, shown in Fig. 5.5

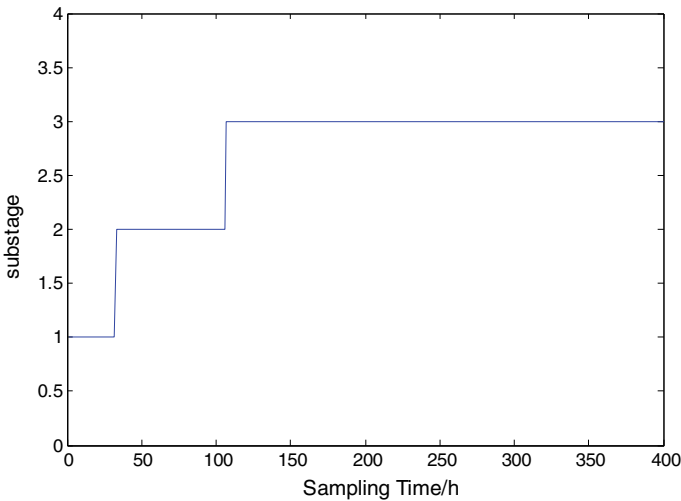


Fig. 5.3 Rough division result based on K-mean clustering

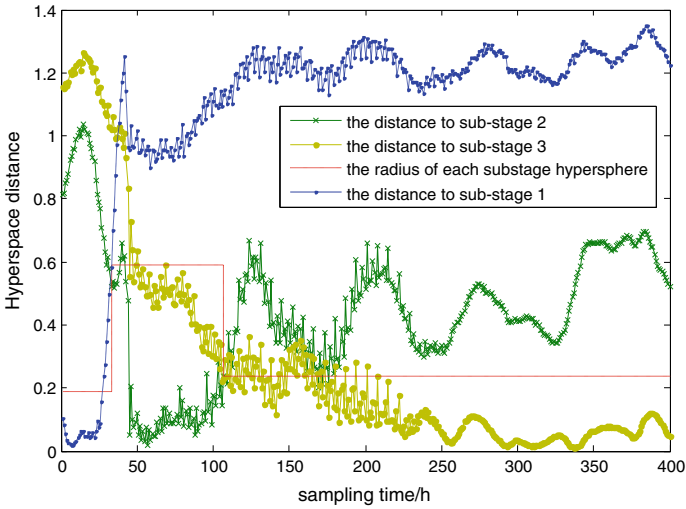


Fig. 5.4 SVDD stage classification result

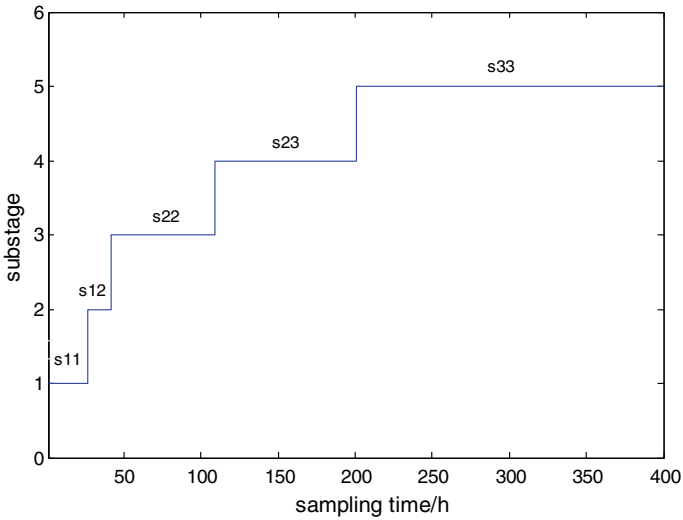


Fig. 5.5 Detailed process division result based on SVDD

It is obviously that the stages during the time interval 1–27, 43–109 and 202–400 are steady stages. The hyperspace distance of stage 28–42, 109–200 exceeded the radius of hypersphere obviously, so the two stages are separated as transition stage. Then the new SVDD classifier model is rebuilt. The whole batch process data set is divided into five stages using the phase identification method proposed in this chapter, that is $S_2 = 5$.

5.3.2 Monitoring of Normal Batch

Monitoring results of the improved sub-PCA methods for the normal batch are presented in Fig. 5.6. The blue line is the statistic corresponding to online data and the red line is control limit with 99% confidence, which is calculated based on the normal historical data. It can be seen that as a result of great change of hyperspace distance at about 30h in Fig. 5.4, the T^2 control limit drops sharply. The T^2 statistic of this batch still stays below the confidence limits. Both of the monitoring systems (T^2 and SPE) do not yield any false alarms. It means that this batch behaves normally during the running.

5.3.3 Monitoring of Fault Batch

Monitoring results of the proposed method are compared with that of traditional sub-PCA method in order to illustrate the effectiveness. Here two kinds of faults are used to test the monitoring system. Fault 1 is the agitator power variable with a decreasing 10% step at the time interval 20–100. They are shown in Figs. 5.7 and 5.8 that SPE statistic increases sharply beyond the control limit in both methods, while T^2 statistic which in fact reflects the changing of sub-PCA model did not beyond the control limit in traditional sub-PCA method. That means the proposed soft-transition method made a more exact model than traditional sub-PCA method.

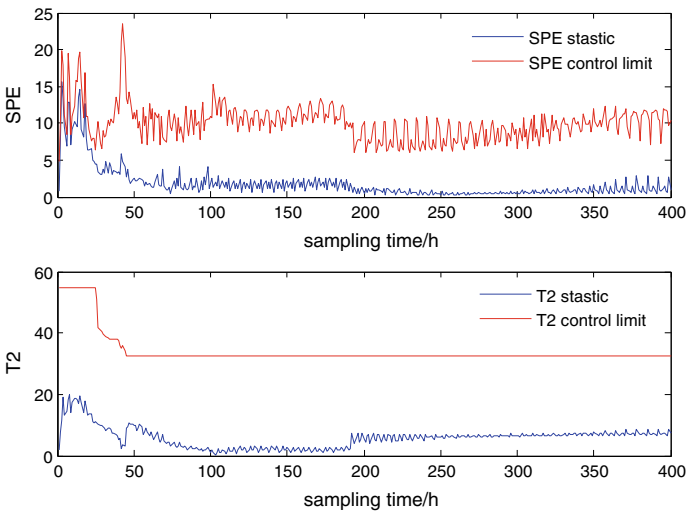


Fig. 5.6 Monitoring plots for a normal batch

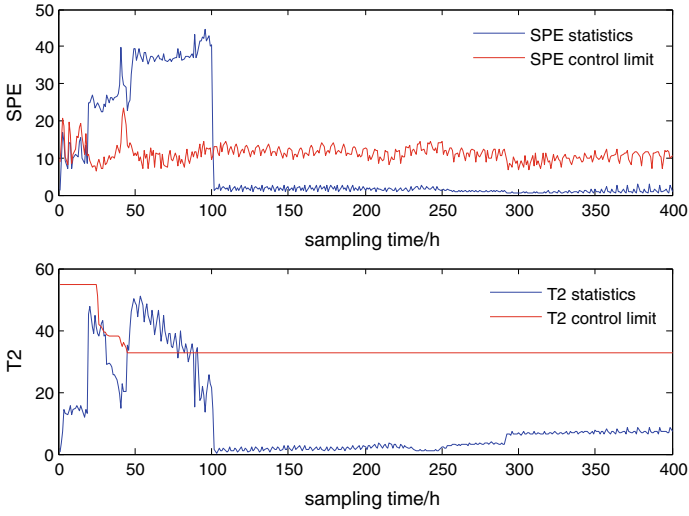


Fig. 5.7 The proposed soft-transition monitoring for fault 1

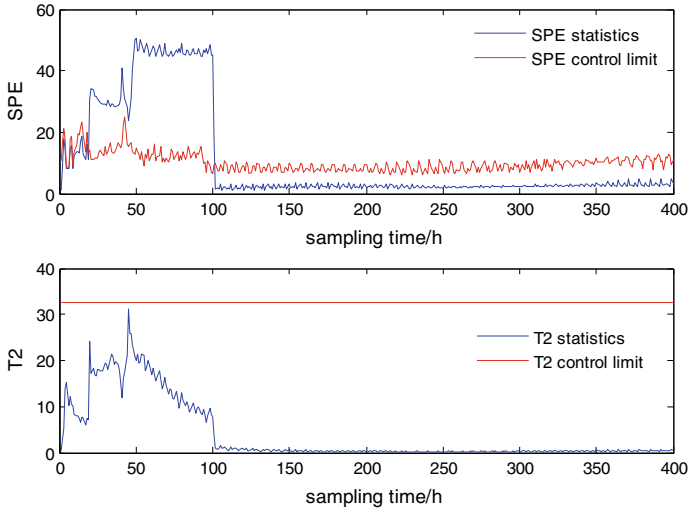


Fig. 5.8 The traditional Sub-PCA monitoring for fault 1

Fig. 5.9 Projection in principal components space of the proposed method

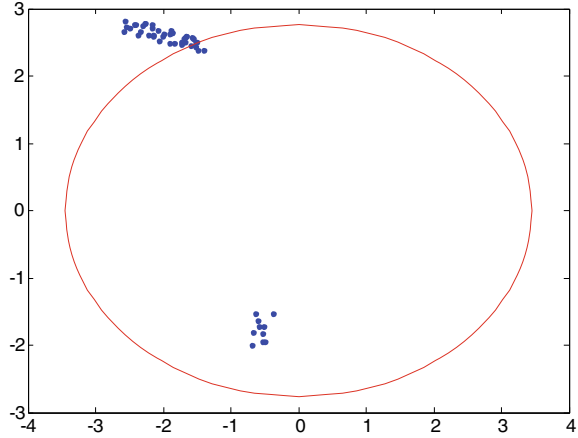
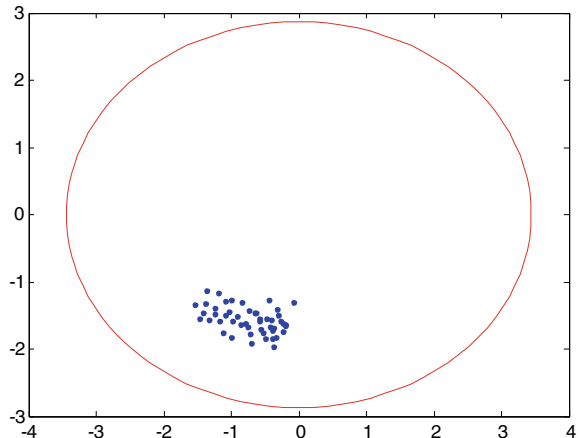


Fig. 5.10 Projection in principal components space of the traditional sub-PCA



The differences between these two methods can be seen directly at the projection map, i.e. Figs. 5.9 and 5.10. The blue dot is the projection of data in the time interval 50–100 to the first two principal components space, and the red line is control limit. Figure 5.10 shows that none of the data out of control limit using the traditional sub-PCA method. The reason is that the traditional sub-PCA does not divide transition stage. The proposed soft-transition sub-PCA can effectively diagnose the abnormal or fault data, shown in Fig. 5.9.

Fault 2 is a ramp decreasing with 0.1 slopes which is added to the substrate feed rate at the time interval 20–100. Online monitoring result of the traditional sub-PCA and proposed method are shown in Figs. 5.11 and Fig. 5.12. It can be seen that this fault is detected by both two methods. The SPE statistic of the proposed method is out of the limit about at 50h and the T^2 values alarms at 45h. Then both of them increase slightly and continuously until the end of fault.

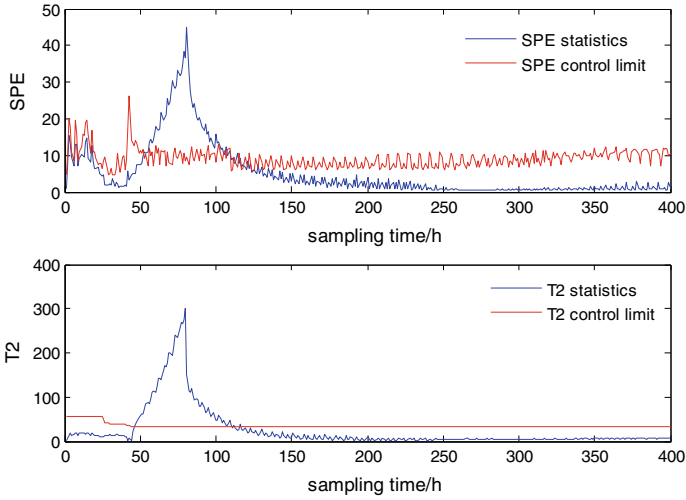


Fig. 5.11 Proposed Soft-transition monitoring results for fault 2

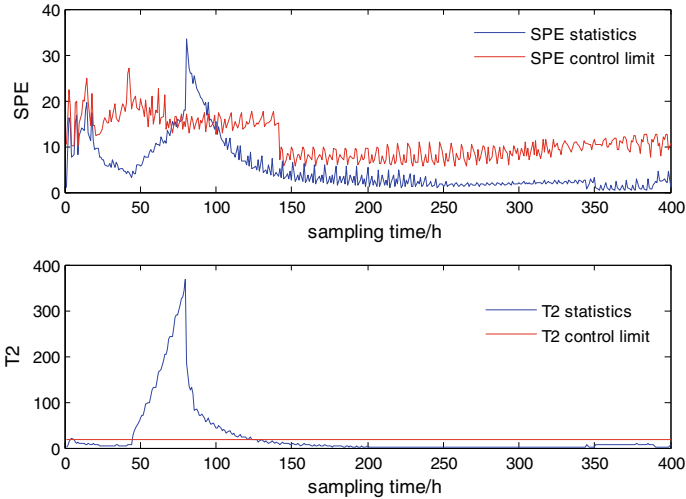


Fig. 5.12 The traditional Sub-PCA monitoring for fault 2

It is clearly shown in Fig. 5.12 that the SPE statistic of traditional sub-PCA did not alarm until about 75 h, which lags far behind that of the proposed method. Meanwhile, the T^2 statistic has a fault alarm at the beginning of the process. It is a false alarm caused by the changing of process initial state. In comparison, the proposed method has fewer false alarms, and the fault alarm time of the proposed method is obviously ahead of the traditional sub-PCA.

Table 5.1 Monitoring results of FA for other faults

Fault ID	Var. No.	Fault type	M/S (%)	Fault time (h)	Soft-transition sub-PCA			Trad. sub PCA (Camacho and Pico 2006)		
					Time (SPE)	Time (T ²)	FA	Time (SPE)	Time (T ²)	FA
1	2	Step	-15	20	20	28	0	20	none	9
2	2	Step	-15	100	100	100	0	100	101	1
3	3	Step	-10	190	190	199	0	190	213	11
4	3	Step	-10	30	48	45	0	81	45	5
5	1	Step	-10	20	20	20	0	20	48	1
6	1	Step	-10	150	150	151	0	150	151	2
7	3	Ramp	-5	20	28	40	0	28	41	1
8	2	Ramp	-20	20	31	45	0	44	34	6
9	1	Ramp	-10	20	24	30	0	21	28	10
10	3	Ramp	-0.2	170	171	171	0	170	173	3
11	2	Ramp	-20	170	181	195	0	177	236	1
12	1	Ramp	-10	180	184	188	0	185	185	2

The monitoring results for other 12 different faults are presented in Table 5.1. The fault variable No. (1, 2, 3) represents the aeration rate, agitator power and substrate feed rate, respectively, as shown in Chap.4. Here FA is the number of false alarm during the operation life.

It can be seen that the false alarms of the conventional sub-PCA method is obviously higher than that of the proposed method. In comparisons, the proposed method shows good robustness. The false alarms here are caused by the little change of the process initial state. The initial states are usually different in real situation, which will lead to the changes in monitoring model. Many false alarms are caused by these little changes. The conventional sub-PCA method shows poor monitor performance in some transition stage and even can't detect these faults because of the inaccurate stage division.

5.4 Conclusions

In a multi-stage batch process, the correlation between process variables changes as the stages are shifted. It makes MPCA and traditional sub-PCA methods inadequate for process monitoring and fault diagnosis. This chapter proposes a new phase identification method to explicitly identify stable and transitional phases. Each phase usually has its own dynamic characteristics and deserves to be treated separately. In particular, the transition phase between two stable phases has its own dynamic transition characteristics and it is difficult to identify.

Two techniques are adopted in this chapter to overcome the above problems. Firstly, inaccurate phase delineation caused by fault data is avoided in the rough division by introducing sampling times in the loading matrix. Then, based on the distance of the process data to the center of the SVDD hypersphere, transition phases can be identified from nearby stable phases. Separate sub-PCA models are given for these stable and transitional phases. In particular, the soft transition sub-PCA model is a weighted sum of the previous stable stage, the current transition stage and the next stable stage. It can reflect the dynamic characteristic changes of the transition phase.

Finally, the proposed method is applied to the penicillin fermentation process. The simulation results show the effectiveness of the proposed method. Furthermore, the method can be applied to the problem of monitoring any batch or semi-batch process for which detailed process information is not available. It is helpful when identifying the dynamic transitions of unknown batch or semi-batch processes.

References

- Arteaga F, Ferrer A (2002) Dealing with missing data in MSPC: several methods, different interpretations, some examples. *J Chemom* 16:408–418
- Camacho J, Pico J (2006) Multi-phase principal component analysis for batch processes modeling. *Chemom Intell Lab Syst* 81:127–136
- Camacho J, Pico J, Ferrer A (2008) Multi-phase analysis framework for handling batch process data. *J Chemom* 22:632–643
- Doan XT, Srinivasan R (2008) Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control. *Comput Chem Eng*, pp 230–243
- Dong D, McAvoy TJ (1996) Batch tracking via nonlinear principal component analysis. *AIChE J* 42:2199–2208
- Kassida A, MacGregor JF, Taylor PA (1998) Synchronization of batch trajectories using dynamic time warping. *Am Inst Chem Eng J* 44:864–875
- Kosanovich KA, Dahl KS, Piovoso MJ (1996) Improved process understanding using multiway principal component analysis. *Ind Eng Chem Res* 35:138–146
- Muthuswamy K, Srinivasan R (2009) Phase-based supervisory control for fermentation process development. *J Process Control* 13:367–382
- Reinikainen SP, Hoskuldsson A (2007) Multivariate statistical analysis of a multistep industrial processes. *Anal Chimica Acta* 595:248–256
- Tax DMJ, Duin RPW (1999) Support vector domain description. *Pattern Recognit Lett* 20:1191–1199
- Tax DMJ, Duin RPW (2004) Support vector domain description. *Mach Learn* 54:45–66
- Undey C, Cinar A (2002) Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Syst Mag*, pp 40–52
- Westerhuis JA, Kourti T, Macgregor JF (1999) Comparing alternative approaches for multivariate statistical analysis of batch process data. *J Chemom* 13:397–413
- Wold S, Kettaneh N, Friden H, Holmberg A (1998) Modelling and diagnosis of batch processes and analogous kinetic experiments. *Chemom Intell Lab Syst* 44:331–340

- Yao Y, Gao FR (2009) A survey on multistage/multiphase statistical modeling methods for batch processes. *Annu Rev Control* 33:172–183
- Yao Y, Gao FR (2009) Phase and transition based batch process modeling and online monitoring. *J Process Control* 19:816–826
- Zhao CH, Wang FL, Lu NY, Jia MX (2007) Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. *J Process Control* 17:728–741

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 6

Statistics Decomposition and Monitoring in Original Variable Space



The traditional process monitoring method first projects the measured process data into the principle component subspace (PCS) and the residual subspace (RS), then calculates T^2 and SPE statistics to detect the abnormality. However, the abnormality by these two statistics are detected from the principle components of the process. Principle components actually have no specific physical meaning, and do not contribute directly to identify the fault variable and its root cause. Researchers have proposed many methods to identify the fault variable accurately based on the projection space. The most popular is contribution plot which measures the contribution of each process variable to the principal element (Wang et al. 2017; Luo et al. 2017; Liu and Chen 2014). Moreover, in order to determine the control limits of the two statistics, their probability distributions should be estimated or assumed as specific one. The fault identification by statistics is not intuitive enough to directly reflect the role and trend of each variable when the process changes.

In this chapter, direct monitoring in the original measurement space is investigated, in which the two statistics are decomposed as a unique sum of the variable contributions of the original process variables, respectively. The monitoring of the original process variables is direct and explicit in the physical meaning, but it is relatively complicated and time consuming due to the need to monitor each variable in both SPE and T^2 statistics. To address this issue, a new combined index is proposed and interpreted in geometric space, which is different from other combined indices (Qin 2003; Alcalá and Qin 2010). The proposed combined index is an intrinsic method. Compared with the traditional latent space methods, the combined index-based monitoring does not require the prior distribution assumption to calculate the control limits. Thus, the monitor complexity is reduced greatly.

6.1 Two Statistics Decomposition

According to the traditional PCA method, the process variables \mathbf{x} could be divided into two parts: principal component $\hat{\mathbf{x}}$ and the residual \mathbf{e} :

$$\mathbf{x} = \mathbf{t}\mathbf{P}^T + \mathbf{e} = \hat{\mathbf{x}} + \mathbf{e}, \quad (6.1)$$

where \mathbf{P} is the matrix associated with the loading vectors that define the latent variable space, \mathbf{t} is the score matrix that contains the coordinates of \mathbf{x} in that space, and \mathbf{e} is the matrix of residuals. T^2 and SPE statistics are used to measure the distance from the new data to the model data. Generally, T^2 and SPE statistics should be analyzed simultaneously so that the cumulative effects of all variables can be utilized. However, most of the literatures have only considered the decomposition of T^2 . Therefore, this chapter considered the SPE statistical decomposition to obtain the original process variables monitored in T^2 and in the SPE statistical space.

6.1.1 T^2 Statistic Decomposition

The statistic can be reformulated as follows:

$$T^2 := \mathbf{D} = \mathbf{t}\mathbf{A}^{-1}\mathbf{t}^T = \mathbf{x}\mathbf{P}\mathbf{A}^{-1}\mathbf{P}^T\mathbf{x}^T = \mathbf{x}\mathbf{A}\mathbf{x}^T = \sum_{i=1}^J \sum_{j=1}^J a_{i,j}x_i x_j \geq 0, \quad (6.2)$$

where $\mathbf{A} = \mathbf{P}\mathbf{A}^{-1}\mathbf{P}^T \geq 0$, \mathbf{A}^{-1} is the inverse of the covariance matrix estimated from a reference population, and $a_{i,j}$ is the element of matrix \mathbf{A} .

One of the T^2 statistic decompositions (Birol et al. 2002) is given as follows:

$$\begin{aligned} \mathbf{D} &= \sum_{k=1}^J \frac{a_{k,k}}{2} \left[(x_k - x_k^*)^2 + (x_k^2 - x_k^{*2}) \right] \\ &= \sum_{k=1}^J a_{k,k} \left[(x_k^2 - x_k^* x_k) \right] \\ &= \sum_{k=1}^J c_k^D. \end{aligned} \quad (6.3)$$

The x_k^* is given as follows:

$$x_k^* = -\frac{\sum_{\substack{j=1 \\ j \neq k}}^N a_{k,j} x_j}{a_{k,k}},$$

where the c_k^D is the decomposed T^2 statistic of each variable x_k . Next, the T^2 statistic of each variable x_k can be calculated as follows:

$$c_k^D = a_{k,k} [(x_k^2 - x_k^* x_k)]. \quad (6.4)$$

The detailed T^2 statistic decomposition process is not shown in here, details can be found in Alvarez et al. (2007, 2010).

6.1.2 SPE Statistic Decomposition

The SPE statistic, which reflects the change of the random quantity in the residual subspace, also has a quadratic form:

$$\begin{aligned} \text{SPE} &:= \mathbf{Q} = \mathbf{e}\mathbf{e}^T = \mathbf{x} (\mathbf{I} - \mathbf{P}\mathbf{P}^T) (\mathbf{I} - \mathbf{P}\mathbf{P}^T)^T \mathbf{x}^T \\ &= \mathbf{x}\mathbf{B}\mathbf{x}^T = \sum_{i=1}^J \sum_{j=1}^J b_{i,j} x_i x_j, \end{aligned} \quad (6.5)$$

where $\mathbf{B} = (\mathbf{I} - \mathbf{P}\mathbf{P}^T) (\mathbf{I} - \mathbf{P}\mathbf{P}^T)^T$, $b_{i,j}$ is the element of matrix \mathbf{B} , and $b_{i,j} = b_{j,i}$. Similar to the decomposition of T^2 statistic, SPE statistics can also be decomposed into a series of new statistic of each variable.

Firstly, the SPE statistic \mathbf{Q} can be reformulated in terms of a single variable x_k :

$$\mathbf{Q} = \mathbf{Q}_k = b_{k,k} x_k^2 + \left(2 \sum_{j=1, j \neq k}^J b_{k,j} x_j \right) x_k + \sum_{i=1, i \neq k}^J \sum_{j=1, j \neq k}^J b_{i,j} x_i x_j. \quad (6.6)$$

The minimum value of \mathbf{Q}_k can be calculated as

$$\frac{\partial \mathbf{Q}_k}{\partial x_k} = 2b_{k,k} x_k^* + 2 \sum_{j=1, j \neq k}^J b_{k,j} x_j = 0 \Rightarrow x_k^* = - \sum_{j=1, j \neq k}^J b_{k,j} x_j / b_{k,k} \quad (6.7)$$

$$\mathbf{Q}_k^{\min} = -b_{k,k} x_k^{*2} + \sum_{i=1, i \neq k}^J \sum_{j=1, j \neq k}^J b_{i,j} x_i x_j. \quad (6.8)$$

The difference between the SPE statistic of x_k and \mathbf{Q}_k^{\min} is

$$\mathbf{Q} - \mathbf{Q}_k^{\min} = b_{k,k} (x_k - x_k^*)^2. \quad (6.9)$$

The sum of the \mathbf{Q}_k^{\min} for $k = 1, 2, \dots, J$ is

$$\begin{aligned}
\sum_{k=1}^J Q_k^{\min} &= \sum_{k=1}^J \left(-b_{k,k} x_k^{*2} + \sum_{i=1, i \neq k}^J \sum_{j=1, j \neq k}^J b_{i,j} x_i x_j \right) \\
&= (J-2) Q + \sum_{k=1}^J b_{k,k} (x_k^2 - x_k^{*2}).
\end{aligned} \tag{6.10}$$

The SPE statistic obtained from (6.10) can be evaluated as the sum of the contributions of each variable x_k :

$$\begin{aligned}
Q &= \sum_{k=1}^J \frac{b_{k,k}}{2} \left[(x_k - x_k^*)^2 + (x_k^2 - x_k^{*2}) \right] \\
&= \sum_{k=1}^J b_{k,k} \left[(x_k^2 - x_k^* x_k) \right] \\
&= \sum_{k=1}^J q_k^{\text{SPE}}.
\end{aligned} \tag{6.11}$$

The original process variables of the SPE statistic are used to monitor the system status:

$$q_k^{\text{SPE}} = b_{k,k} \left[(x_k^2 - x_k^* x_k) \right]. \tag{6.12}$$

So the novel SPE statistic can be evaluated as a unique sum of the contributions of each variable q_k^{SPE} ($k = 1, 2, \dots, J$), which is used for original process variable monitoring.

6.1.3 Fault Diagnosis in Original Variable Space

Similar to other PCS monitoring strategies, the proposed original variable monitoring technique consists of two stages that are executed offline and online. Firstly, the control limits of the two statistics (T^2 and SPE) for each time interval are determined by reference population of normal batches in the offline stage. Next, two statistics are calculated at each sampling during the online stage. If one of statistics exceeds the established control limit, then a faulty mode is declared.

The historical data of the batch process are composed of a three-dimensional array $\mathbf{X}(I \times J \times K)$, where I , J , and K are the number of batches, process variables, and sampling times, respectively. The three-dimensional process data must be unfolded into two-dimensional forms $\mathbf{X}_k(I \times J)$, $k = 1, 2, \dots, K$ before performing the PCA operation. The unfolding matrix \mathbf{X}_k is normalized to zero mean and unit variance in each variable. The main nonlinear and dynamic components of the variable are still left in the scaled data matrix \mathbf{X}_k .

The normalized data matrix \mathbf{X}_k is projected into principal component subspace by loading matrix \mathbf{P}_k to obtain the scores matrix \mathbf{T}_k :

$$\mathbf{X}_k = \mathbf{T}_k \mathbf{P}_k^T + \mathbf{E}_k,$$

where \mathbf{E}_k is the residual matrix. The two statistics associated with the i th batch for the j th variable in k th time interval are defined as $c_{i,j,k}^D$ and $q_{i,j,k}^{\text{SPE}}$.

The control limit of a continuous process can be determined by using the kernel density estimation (KDE) method. Another method has been used for calculating the control limit for batch process, which is determined by the mean and variance of each statistic (Yoo et al. 2004; Alvarez et al. 2007). The mean and variance of $c_{i,j,k}^D$ are calculated as follows:

$$\begin{aligned} \bar{c}_{j,k}^D &= \sum_{i=1}^I c_{i,j,k}^D / I \\ \text{var}(c_{j,k}^D) &= \sum_{i=1}^I (c_{i,j,k}^D - \bar{c}_{j,k}^D)^2 / (I - 1). \end{aligned} \quad (6.13)$$

The control limit of statistic $c_{i,j,k}^D$ is estimated as

$$c_{j,k}^{\text{limit}} = \bar{c}_{j,k}^D + \lambda_1 (\text{var}(c_{j,k}^D))^{1/2}, \quad (6.14)$$

where λ_1 is a predefined parameter. Similarly, the control limit of statistic is

$$q_{j,k}^{\text{limit}} = \bar{q}_{j,k}^{\text{SPE}} + \lambda_2 (\text{var}(q_{j,k}^{\text{SPE}}))^{\frac{1}{2}}, \quad (6.15)$$

where λ_2 is a predefined parameter,

$$\begin{aligned} \bar{q}_{j,k}^{\text{SPE}} &= \sum_{i=1}^I q_{i,j,k}^{\text{SPE}} / I \\ \text{var}(q_{j,k}^{\text{SPE}}) &= \sum_{i=1}^I (q_{i,j,k}^{\text{SPE}} - \bar{q}_{j,k}^{\text{SPE}})^2 / (I - 1). \end{aligned} \quad (6.16)$$

As above, the control limit calculation is very simple. Although the calculation increases, the extra calculations can be performed offline, there is no restriction during the online monitoring stage. The proposed monitoring technique corresponding to the offline and online stages is summarized as follows:

A. Offline Stage

1. Obtain the normal process data of I batches \mathbf{X} , unfold them into two-dimensional time-slice matrix \mathbf{X}_k , and then normalize the data.

2. Perform the PCA procedure on the normalized matrix \mathbf{X}_k of each time slice and obtain the loading matrices \mathbf{P}_k .
3. Calculate the statistics $c_{i,j,k}^D$ and $q_{i,j,k}^{\text{SPE}}$ of each variable in all the interval times for all batches, then calculate the variable contributions at each time interval using (6.4) and (6.12).
4. The control limits of statistics $c_{i,j,k}^D$ and $q_{i,j,k}^{\text{SPE}}$ are estimated as (6.14) and (6.15).

B. Online Stage

1. Collect new sampling time-slice data x_{new} , and then normalize based on the mean and variance of prior normal I batches data (modeling data).
2. Use \mathbf{P}_k to calculate the new statistics $c_{i,j,k}^D$ and $q_{i,j,k}^{\text{SPE}}$ of new sampling, and judge whether these statistics exceed the control limit. If one of them exceeds the control limit, then fault identification is performed to find the faulty variable that exceeds the control limit much greater than others; if none of them exceeds the control limit, then the current data are normal.

6.2 Combined Index-Based Fault Diagnosis

The monitoring method in the original process variables can avoid some of the disadvantages of traditional statistic approach in the latent variable space, such as indirectly monitoring (Yoo et al. 2004). However, the original variable monitoring method is relatively complicated due to the monitoring of each variable in both SPE and T^2 statistics. It means that each variable should be monitored twice, which increases the calculation. Thus, a new combined index, composed of the SPE and T^2 statistics, is proposed to decrease monitoring complexity.

6.2.1 Combined Index Design

In this section, we use symbol $\mathbf{X}(I \times J)$ to substitute the unfolding process data matrix $\mathbf{X}_k(I \times J)$ for general analysis. Similarly, $\mathbf{P}_k, \mathbf{T}_k, \mathbf{E}_k$ are substituted by $\mathbf{P}, \mathbf{T}, \mathbf{E}$. The process data \mathbf{X} could be decomposed into PCS and RS when performing PCA:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E}, \quad (6.17)$$

where $\hat{\mathbf{X}}$ is the PCS and \mathbf{E} is the RS. If the principal number is m , then a PCS with m -dimension and a RS with $(J - m)$ -dimension can be obtained. When new data x are measured, they are projected into the principal subspace:

$$\mathbf{t} = \mathbf{x}\mathbf{P}. \quad (6.18)$$

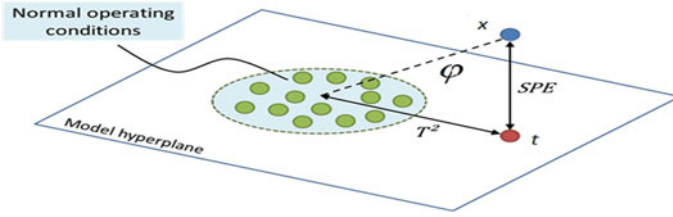


Fig. 6.1 Graphical representation of T^2 and SPE statistics

The principal component (PC) score vector $t(1 \times m)$ is the projection of new data x in the PCS. Subsequently, the PC score vector is projected back into the original process variables to estimate the process data $\hat{x} = tP^T$. The residual vector e is defined as

$$e = x - \hat{x} = x(I - PP^T). \tag{6.19}$$

Residual vector e reflects the difference between new data x and modeling data X in the RS. A graphical interpretation of T^2 and SPE statistics is shown in Fig. 6.1.

To describe the statistics clearly in the geometry, the principal component subspace is taken as a hyperplane. The SPE statistic checks the model validity by measuring the distance between the data in the original process variables and its projection onto the model plain. Generally, the T^2 statistic is described by the Mahalanobis distance of the project point t to the projection center of normal process data, which aims to check if the new observation is projected into the limits of normal operation. The residual space is perpendicular to the principal hyperplane. The SPE statistic shows the distance from the new data x to the principal component hyperplane.

A new distance index φ from the new data to the principal component projection center of the modeling data is given in the following. It can be used for monitoring instead of the SPE and T^2 indicators. Consider the singular value decomposition (SVD) of the covariance matrix $R_x = \mathbb{E}(X^T X)$ for given normal data X ,

$$R_x = U\Lambda U^T,$$

where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_m, \mathbf{0}_{J-m}\}$ is the eigenvalue of R_x . The original loading matrix $U_{J \times J}$ is a unitary matrix and $UU^T = I$. Each column of the unitary matrix is a set of standard orthogonal basis in its span space. The basis vectors of principal component space and residual space divided from matrix U are orthogonal to each other. Furthermore,

$$U = [P, P_e], \tag{6.20}$$

where $P \in R^{J \times m}$ is the loading matrix. $P_e \in R^{J \times (J-m)}$ can be treated as the loading matrix of residual space. Thus, P and P_e are presented by U as follows:

$$P = UF_1, P_e = UF_2, \tag{6.21}$$

where

$$\mathbf{F}_1 = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0}_{J-m} \end{bmatrix}_{J \times m}, \quad \mathbf{F}_2 = \begin{bmatrix} \mathbf{0}_m \\ \mathbf{I}_{J-m} \end{bmatrix}_{J \times m}, \quad (6.22)$$

where \mathbf{I}_m and \mathbf{I}_{J-m} are the m and $J - m$ dimension unit matrices, respectively, and $\mathbf{0}_m$ and $\mathbf{0}_{J-m}$ are the m and $J - m$ dimension zero matrices, respectively. Furthermore, the SPE and T^2 statistics are denoted by \mathbf{U} :

$$\begin{aligned} \mathbf{e} &= \mathbf{x}(\mathbf{I} - \mathbf{P}\mathbf{P}^T) = \mathbf{x}(\mathbf{U}\mathbf{U}^T - \mathbf{U}\mathbf{F}_1\mathbf{F}_1^T\mathbf{U}^T) \\ &= \mathbf{x}(\mathbf{U}\mathbf{U}^T - \mathbf{U}\mathbf{E}_1\mathbf{U}^T) = \mathbf{x}\mathbf{U}(\mathbf{I} - \mathbf{E}_1)\mathbf{U}^T = \mathbf{x}\mathbf{U}\mathbf{E}_2\mathbf{U}^T, \end{aligned} \quad (6.23)$$

where

$$\mathbf{E}_1 = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m, J-m} \\ \mathbf{0}_{J-m, m} & \mathbf{0}_{J-m} \end{bmatrix}, \quad \mathbf{E}_2 = \begin{bmatrix} \mathbf{0}_m & \mathbf{0}_{m, J-m} \\ \mathbf{0}_{J-m, m} & \mathbf{I}_{J-m} \end{bmatrix}. \quad (6.24)$$

Define $\mathbf{y} = \mathbf{x}\mathbf{U}$, then

$$\begin{aligned} \text{SPE} &:= \mathbf{Q} = \mathbf{e}\mathbf{e}^T = \mathbf{x}\mathbf{U}\mathbf{E}_2\mathbf{U}^T\mathbf{U}\mathbf{E}_2\mathbf{U}^T\mathbf{x}^T \\ &= \mathbf{x}\mathbf{U}\mathbf{E}_2\mathbf{U}^T\mathbf{x}^T = \mathbf{y}\mathbf{E}_2\mathbf{y}^T = \sum_{i=m+1}^J y_i^2. \end{aligned} \quad (6.25)$$

Similarly, we can describe the T^2 statistic as follows:

$$\begin{aligned} T^2 &:= \mathbf{D} = \mathbf{t}\mathbf{\Lambda}_m^{-1}\mathbf{t}^T = \mathbf{x}\mathbf{P}\mathbf{\Lambda}_m^{-1}\mathbf{P}^T\mathbf{x}^T \\ &= \mathbf{x}\mathbf{U}\mathbf{F}_1\mathbf{\Lambda}_m^{-1}\mathbf{F}_1^T\mathbf{U}^T\mathbf{x}^T = \mathbf{x}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{x}^T \\ &= \mathbf{y}\mathbf{\Lambda}^{-1}\mathbf{y}^T = \sum_{i=1}^m y_i^2\sigma_i^2, \end{aligned} \quad (6.26)$$

where

$$\mathbf{\Lambda}_m^{-1} = \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2\}, \quad \mathbf{\Lambda}^{-1} = [\mathbf{\Lambda}_m^{-1}, \mathbf{0}_{(J-m) \times (J-m)}].$$

The new combined index could be obtained directly by composing the two statistics as

$$\varphi = \mathbf{D} + \mathbf{Q} = \sum_{i=1}^m y_i^2\sigma_i^2 + \sum_{i=m+1}^J y_i^2. \quad (6.27)$$

It is proved via mathematical illustration that the two decomposed statistics can be geometrically added together directly. This result demonstrates that T^2 and SPE statistic can be combined primarily and that is an intrinsic property. Thus, the combined index is a more general and geometric representation compared with the other combined index. The monitoring strategy with the novel index is introduced in the next subsection.

6.2.2 Control Limit of Combined Index

In Sect. 6.1, the T^2 and SPE statistics are decomposed into two new statistics for each variable. To reduce the calculation of process monitoring, the two new statistics are combined into a new statistic φ to monitor the process.

$$\varphi_{i,j,k} = c_{i,j,k}^D + q_{i,j,k}^{\text{SPE}}, \quad (6.28)$$

where $\varphi_{i,j,k}$ is the combined statistic at sampling time k for the j th variable. The method mentioned in Sect. 6.1.3 can be used to calculate the control limit of the new statistic,

$$\varphi_{j,k}^{\text{limit}} = \bar{\varphi}_{j,k} + \kappa(\text{var}(\varphi_{j,k}))^{1/2}, \quad (6.29)$$

where κ is a predefined parameter, and

$$\begin{aligned} \bar{\varphi}_{j,k} &= \sum_{i=1}^I \varphi_{i,j,k} / I \\ \text{var}(\varphi_{j,k}) &= \sum_{i=1}^I (\varphi_{i,j,k} - \bar{\varphi}_{j,k})^2 / (I - 1). \end{aligned} \quad (6.30)$$

The online process monitoring can be performed according to comparing the new statistic and its control limit. There are several points to highlight for readers when the proposed control limit is used. Firstly, the mean and variance may be inaccurate for a small number of samples. As a result, a sufficient number of training samples should be collected during the offline stage. Secondly, the predefined parameter is important and it is designed by the engineers according to the actual process conditions. The tuning method regarding κ is similar to the Shewhart control chart. Equation (6.29) illustrates that the effect of variance depends on the predefined parameter κ and the fluctuation of control limits also relies on it on each sample. For example, the control limit is smooth when κ is selected to be a smaller value, and the control limit fluctuates when κ is selected to be a larger value.

If the combined statistic of the new sample has a significant difference from those of the reference data set, then a fault is detected. As a result, a fault isolation procedure is set up to find the fault roots. This fault response process is one of advantages in original process variable monitoring as each variable has a unique formulation and physical meaning. The proposed monitoring steps are similar as that in Sect. 6.1.2.

6.3 Case Study

A fed-batch penicillin fermentation process is considered in case study, and its detailed mathematical model is given in Birol et al. (2002). A detailed description of the fed-batch penicillin fermentation process is available in Chap. 4.

6.3.1 Variable Monitoring via Two Statistics Decomposition

Firstly, the original process variable monitoring algorithm mentioned in Sect. 6.1.2 is tested. The monitoring results of all variables would be interminable and tedious, so only several typical variables are shown here for demonstration or comparison. The monitoring result of variable 1 in a test normal batch is shown in Fig. 6.2. None of the two statistics ($c_{1,k}^D$ and $q_{1,k}^{SPE}$) exceeds its control limit, and the statistics ($c_{j,k}^D$ and $q_{j,k}^{SPE}$, $j = 2, \dots, 11$) of all the other variables do not exceed the control limits as well. The monitoring results of other variables are similar to that of variable 1, so we omitted them due to the restriction of the book length. These results show that proposed algorithm do not have a false alarm when it is used to monitor the normal batch.

Next, the fault batch data are used to test the proposed monitoring algorithm of the original process variables, and two types of faults are chosen here.

Fault 1: step type, e.g., a 20% step decrease is added in variable 3 at 200–250h.

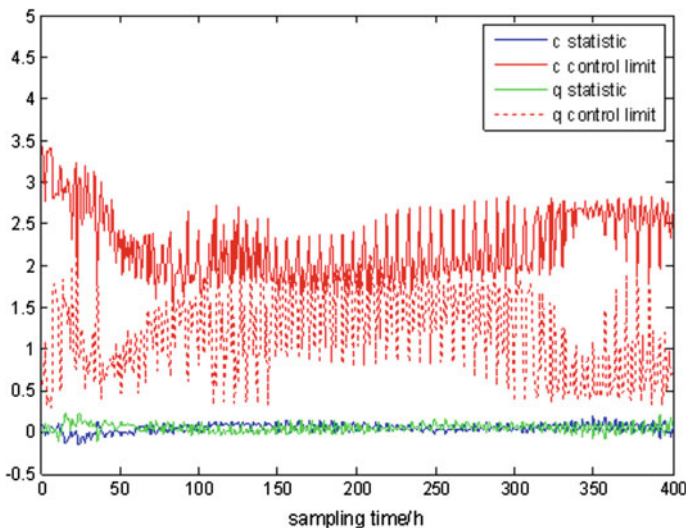


Fig. 6.2 Original variables monitoring for normal batch (variable 1)

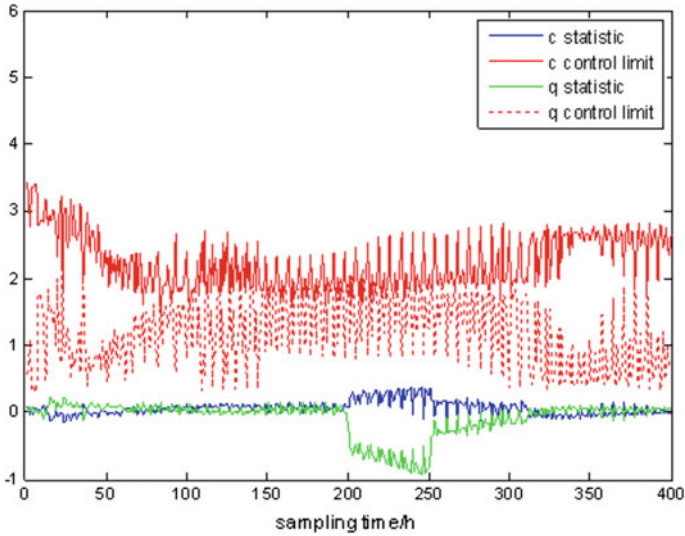


Fig. 6.3 Monitoring result for Fault 1 (variable 1)

The monitoring results are shown as follows. Figure 6.3 shows the monitoring result of variable 1 for fault 1, the statistics changes obviously during the fault occurrence. However, the statistics do not exceed the control limit, i.e., the process status exhibits changes, but variable 1 is not the fault source. The monitoring results of variables 2, 4, 8, 9, and 11 are almost the same as the result of variable 1, and these results are not presented here.

The monitoring results of variable 3 and variable 5 are shown in Figs. 6.4 and 6.5, respectively. Both of the variable statistics exceed the control limit at the sampling time 200h. Regarding the other variables of 6, 7, and 10, the statistics of these variables also exceed the control limit, and the simulation results of these variable are nearly the same as that of variable 5 (the results are not presented here).

The question is: which variable is the fault source, variable 3, 5, or others? From the amplitude of Figs. 6.4 and 6.5, it is easy to see that the two statistics for variable 3 exceed the control limits to a much greater extent than those for variable 5 and other variables. In particular, the Q statistic of variable 3 is 40 times greater than its control limit. From this perspective, variable 3 can be concluded to be the fault source, as it makes contribution to the statistics obviously. Note that there is no smearing effect in the proposed method. The smearing effect means that non-faulty variables exhibit larger contribution values, while the contribution of faulty variables is smaller. Because the statistics are decomposed into a unique sum of the variable contributions, each monitoring figure is plotted against the decomposed variable statistics. Furthermore, the proposed method may identify several faulty variables if they have larger contributions at close magnitudes.

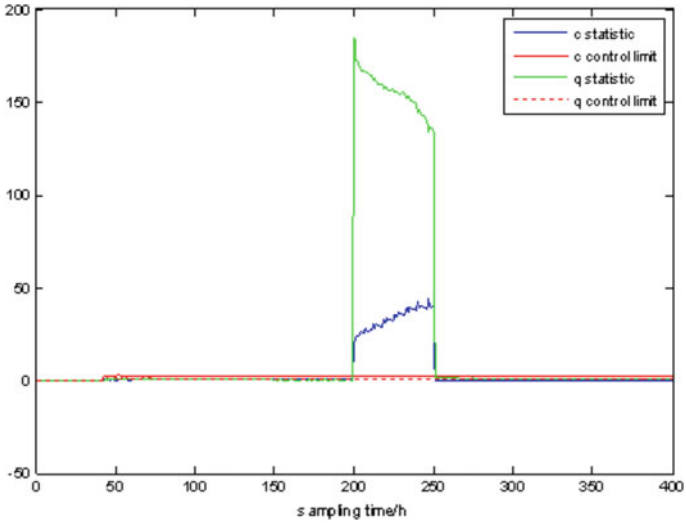


Fig. 6.4 Monitoring result for Fault 1 (variable 3)

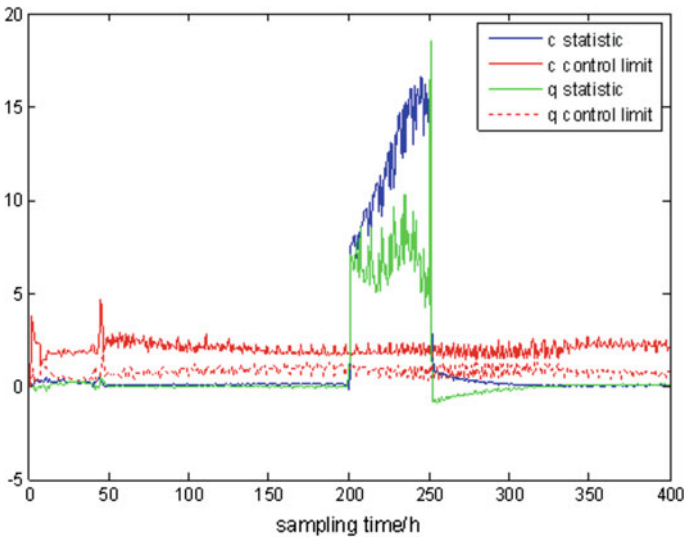


Fig. 6.5 Monitoring result for Fault 1 (variable 5)

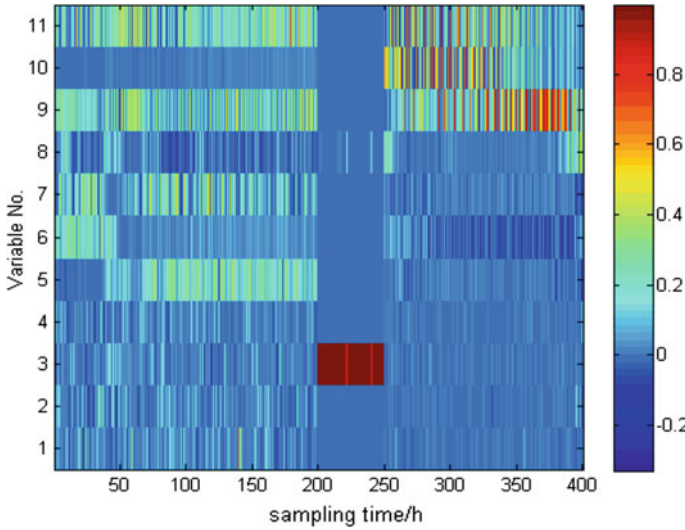


Fig. 6.6 Relative contribution rate of R_c for Fault 1

To confirm the monitoring conclusion, the relative statistical contribution rate of the j th variable at time k is defined as

$$R_c^{j,k} = c_{j,k}^D / \sum_{j=1}^J c_{j,k}^D$$

$$R_q^{j,k} = q_{j,k}^{SPE} / \sum_{j=1}^J q_{j,k}^{SPE}.$$

The relative statistic contribution rates of 11 variables are shown in Figs. 6.6 and 6.7. It is clear that variable 3 is the source of Fault 1. It is found that variables 9, 10, and 11 still have the higher contribution when the fault is eliminated because the fault in variable 3 causes the change of the other process variables. The effects on whole process still continue, even if the fault is eliminated, and the fault variable evolves from the original variable 3 to other process variables.

Fault 2: ramp type, i.e., fault involving a ramp increasing with a slope of 0.3 in variable 3 at 20–80 h.

The two monitor statistics of variable 3 are shown in Figs. 6.8 and 6.9. It can be seen that both of the two statistics exceed the control limits at approximately 50 h. The alarming time lags relative to the fault occurrence time (approximately 20 h) are found because this fault variable changes gradually. When the fault is eliminated after 80 h, the relationship among the variables changes back to normal. The T^2 statistic obviously declines under the control limit, while the SPE statistic still exceeds the control limit because the error caused by Fault 2 still exists.

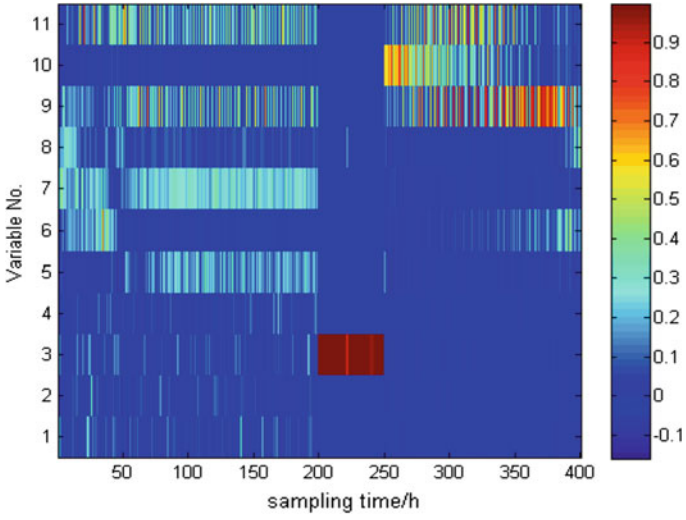


Fig. 6.7 Relative contribution rate of R_q for Fault 1

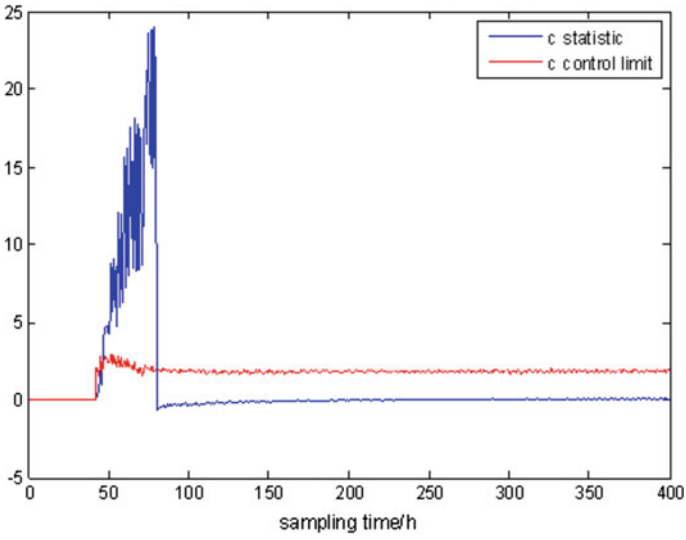


Fig. 6.8 Fault 2 monitoring by c statistic (variable 3)

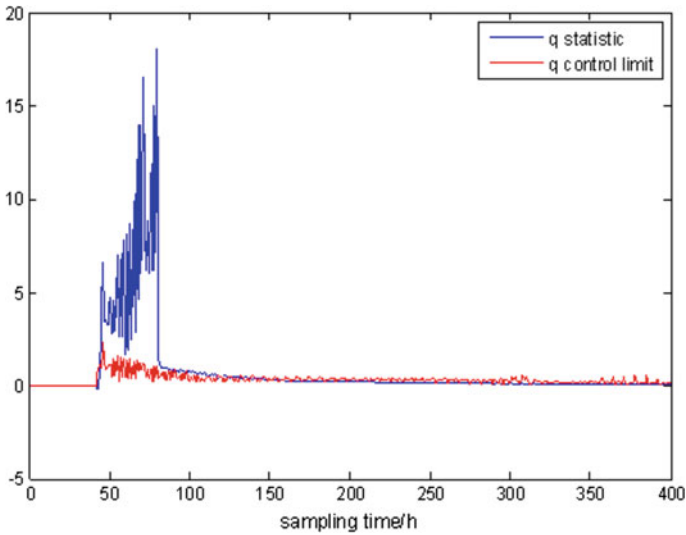


Fig. 6.9 Fault 2 monitoring by q statistic (variable 3)

6.3.2 Combined Index-Based Monitoring

The same test data in Sect. 6.3.1 are used to test monitoring effectiveness of the new combined index. Considering a normal batch, the monitoring result of φ statistic is shown in Fig. 6.10. Variable 1 is still monitored in this section, as was the case in Sect. 6.3.1 for comparison. It is shown that the new index φ of variable 1 is far below its control limit, as is the case for the new index values of the other variables. This method shows some good performances, and the number of false alarms is zero in normal batch monitoring. The new index is more stable than the two statistics, and it is easy to observe for operators.

Fault 1: step type, e.g., a 20% step decrease is added in variable 3 at 200–250h.

The new statistic φ of variable 1 does not exceed the control limit in Fig. 6.11, although it changes from 200h to 250h during the fault. The values of new statistic φ of variables 2, 4, 8, 9, and 11 also do not exceed the control limit. The corresponding monitoring statistics are omitted here. Thus, these variables have no direct relationship with the fault variable, i.e., they are not the fault source.

Furthermore, the monitoring results of variables 3 and 5 are shown in Figs. 6.12 and 6.13, respectively. The value statistics of variables 3 and 5 exceed their control limits obviously, as well as those of variables 6, 7, and 10. As discussed in Sect. 6.3.1, one can see that the statistic φ of variable 3 changes to a greater extent than other variables, so variable 3 is the potential fault source. This result shows that the proposed approach is an efficient technique for fault detection.

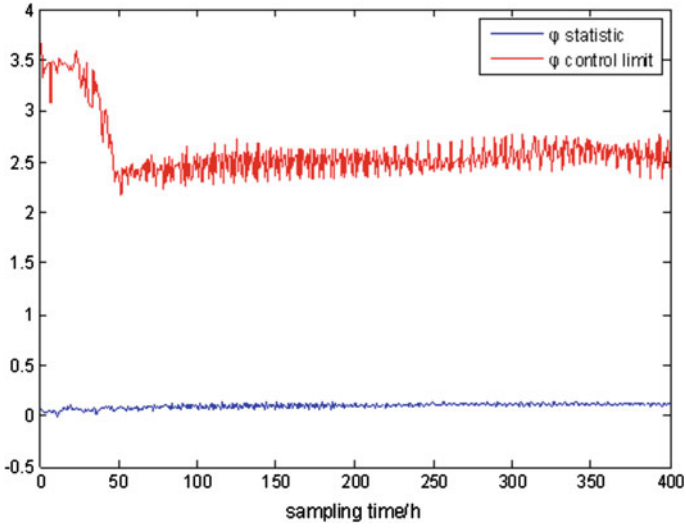


Fig. 6.10 Original variables monitoring based on combined index for normal batch (variable 1)

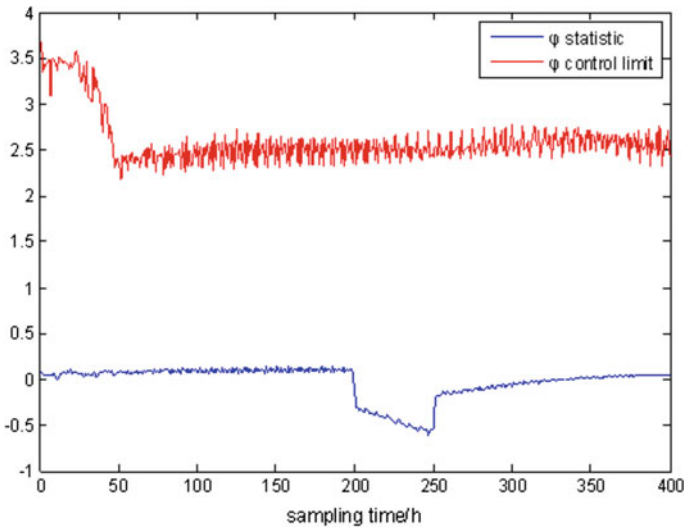


Fig. 6.11 Fault 1 monitoring based on combined index (variable 1)

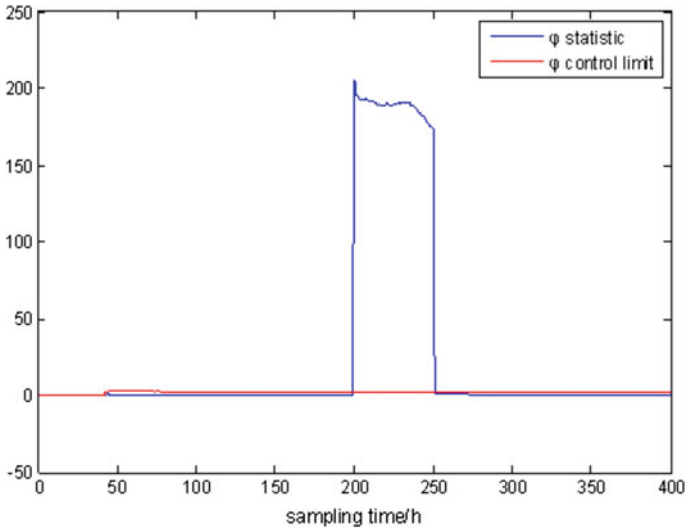


Fig. 6.12 Fault 1 monitoring based on combined index (variable 3)

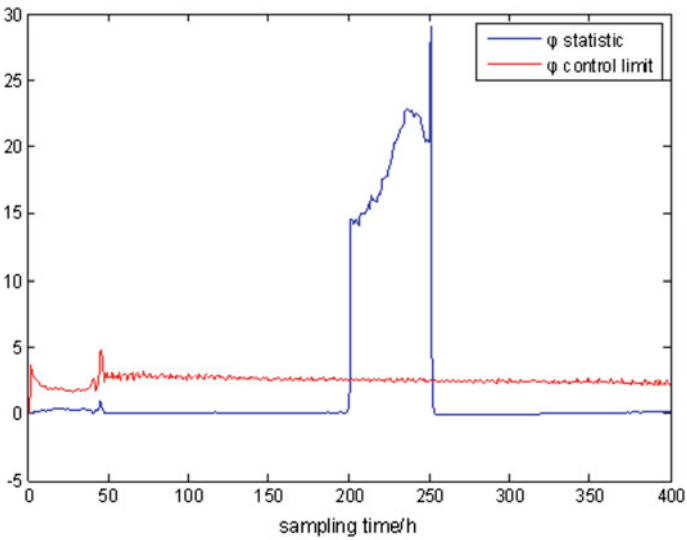


Fig. 6.13 Fault 1 monitoring based on combined index (variable 5)

The relative contribution of the new statistic is used to confirm the fault source, which is defined as

$$R_{\varphi}^k = \varphi_{j,k} / \sum_{j=1}^J \varphi_{j,k}$$

The relative contribution of variable 3 is nearly 100%, as shown in Fig. 6.14. So variable 3 is confirmed as the fault source. It is found that variables 9, 10, and 11 still have a higher contribution when the fault is eliminated because the fault in variable 3 causes the change of the other process variables and the effect on whole process still continues, even if the fault is eliminated.

Note that the relative contribution plot (RCP) is an auxiliary tool to locate the fault roots. It is only used for comparison with the proposed monitoring method to confirm diagnostic conclusions. Furthermore, the RCP is completely different from the traditional contribution diagram in this work. The RCP in this work is calculated using the original process variables, i.e., there is no smearing effect of the RCP. The contribution of each variable is independent of the other variables. Therefore, the proposed method is a novel and helpful approach in terms of original process variable monitoring. Furthermore, the color map of the fault contribution is intuitive. As a result, the map will promote the operator’s initiative to find the fault source, and engineers can find some useful information to avoid more serious accidents.

Fault 2: ramp type, i.e., fault involving a ramp increasing with a slope of 0.3 in variable 3 at 20–80 h.

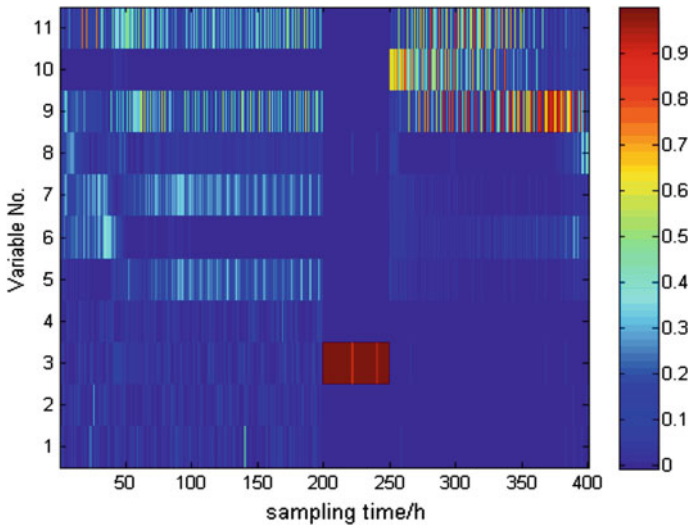


Fig. 6.14 Relative contribution rate of φ statistic for Fault 1

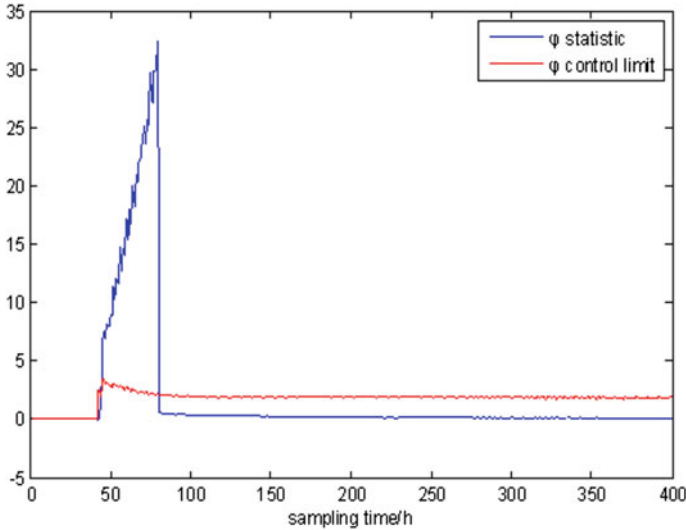


Fig. 6.15 Fault 2 monitoring of variable 3 by φ statistic

The monitoring result of variable 3 is shown in Fig. 6.15. It can be seen that the new statistic φ exceeds the control limit at approximately 50h, and then it falls below the control limit after 80h. The result shows that the combined index can detect different faults.

6.3.3 Comparative Analysis

The monitoring performances of different methods are compared. Several performance indices are given to evaluate the monitoring efficiency. False alarm (FA) is the number of false alarms during the operation life. Time detected (TD) is the time that the statistic exceeds the control limit under the fault operation, which can represent the sensitivity.

The monitoring results of the proposed method are compared with that of the traditional sub-PCA method (Lu et al. 2004) in latent space and the soft-transition sub-PCA (Wang et al. 2013) to illustrate the effectiveness. The FA and TD results for other 12 faults are presented in Tables 6.1 and 6.2, respectively. Fault variable numbers (1, 2, and 3) represent the aeration rate, agitator power, and substrate feed rate, as shown in Chap. 4. The fault type and occurring time for the variables are given in Table 6.1, and those input conditions are as same as those in Sects. 6.3.1 and 6.3.2.

Table 6.1 Monitoring results of FA for other faults

Fault ID	Var. No.	Fault type	M/S (%)	Fault time (h)	Original variables monitoring			Trad. sub-PCA (Lu et al. 2004)	Soft sub-PCA (Wang et al. 2013)
					c	q	φ	FA	FA
1	2	Step	-15	20	0	0	0	9	0
2	2	Step	-15	100	0	173	0	1	0
3	3	Step	-10	190	0	95	0	11	0
4	3	Step	-10	30	0	57	0	5	0
5	1	Step	-10	20	0	0	0	1	0
6	1	Step	-10	150	16	0	0	2	0
7	1	Ramp	-5	20	2	1	0	1	0
8	2	Ramp	-20	20	4	0	0	6	0
9	1	Ramp	-10	20	2	0	1	10	0
10	3	Ramp	-0.2	170	1	0	0	3	0
11	2	Ramp	-20	170	4	0	0	1	0
12	1	Ramp	-10	180	2	0	0	2	0

It can be seen from Table 6.1 that there are multiple false alarms applying the traditional sub-PCA method to detect faults, while the original process variable monitoring method shows less false alarms based on the combined index φ in this chapter. Among the three indices of the original spatial monitoring, the c and q statistics may have a large number of false alarms for different reasons, but the new combined index φ is more accurate because it can balance the two indices.

Table 6.2 indicates that the original process variable monitoring has accurate and timely detection results comparing with the other two detection methods. The detection delay is more than 10h for Fault 4, 7, 8 and 11 in the traditional sub-PCA and the soft-transition sub-PCA. Such a delay is inconceivable in a complex industrial process. While the difference between the detected time and the real fault time for the proposed approach is less than 10h, except for fault 4. This result is helpful and meaningful in practice. As a result, the proposed approach could provide more suitable process information to operators. Thus, the proposed monitoring method based on a combined index shows advantages of rapid detection and fewer false alarms compared with the traditional or soft-transition sub-PCA approaches, whose monitoring operation is in the latent space but not the original measurement space.

Table 6.2 Comparing the time of fault detected

Fault ID	Fault time (h)	Original process variables monitoring			Trad. sub-PCA (Lu et al. 2004)		Soft-trans. sub-PCA (Wang et al. 2013)	
		c	q	φ	SPE	T^2	SPE	T^2
1	20	20	20	20	20	None	20	28
2	100	100	100	100	100	101	100	100
3	190	191	190	190	190	213	190	199
4	30	45	45	45	81	45	48	45
5	20	20	20	20	20	48	20	20
6	150	151	150	150	150	151	150	151
7	20	27	26	25	28	41	28	40
8	20	30	26	26	44	34	31	45
9	20	24	22	23	21	28	24	30
10	170	171	170	170	170	173	171	171
11	170	179	175	175	177	236	181	195
12	180	184	182	182	185	185	184	188

6.4 Conclusions

A new multivariate statistical method for the monitoring and diagnosis of batch processes, which operates on the original process variables, was presented in this chapter. The proposed monitoring method is based on the decomposition of the T^2 and SPE statistics as a unique sum of each variable contribution. However, problems may arise if the number of variables is large when the original process variables technique is applied. To reduce the workload of the monitoring calculation, a new combined index was proposed. A mathematical illustration was given to prove that the two decomposed statistics can be added together directly. Compared to the traditional PCA method in latent space, the proposed method is sufficiently direct, and only one statistical index is utilized, thereby decreases the calculation burden.

The new original variable space monitoring method can detect a fault with a clear result based on each variable. The fault source can be determined directly from the statistical index rather than using the traditional contribution plot. Furthermore, the control limit of the new combined statistics is very simple, and it does not need to assume that it follows some probability distribution. The simulation results show that the new combined statistics can detect the fault efficiently. As the new statistic index is the combination of two decomposed statistics, it can avoid many problems introduced by the use of a single statistic, such as false alarms or missing alarms.

References

- Alcala CF, Qin SJ (2010) Reconstruction-based contribution for process monitoring with kernel principal component analysis. *Ind Eng Chem Res*, pp 7849–7857
- Alvarez CR, Brandolin A, Sanchez MC (2007) On the variable contributions to the D-statistic. *Chemom Intell Lab Syst* 88:189–196
- Alvarez CR, Brandolin A, Sanchez MC (2010) Batch process monitoring in the original measurement's space. *J Process Control* 20:716–725
- Birol G, Undey C, Cinar A (2002) A modular simulation package for fed-batch fermentation: penicillin production. *Comput Chem Eng*, pp 1553–1565
- Liu JL, Chen DS (2014) Fault isolation using modified contribution plots. *Comput Chem Eng*, pp 9–19
- Lu N, Gao F, Wang F (2004) Sub-PCA modeling and on-line monitoring strategy for batch processes. *AIChE J*, pp 255–259
- Luo LJ, Bao SY, Mao JF, Tang D (2017) Fault detection and diagnosis based on sparse PCA and two-level contribution plots. *Ind Eng Chem Res*, pp 225–240
- Qin SJ (2003) Statistical process monitoring: basics and beyond. *J Chemom*, pp 480–502
- Wang J, Ge WS, Zhou JL, Wu HY, Jin QB (2017) Fault isolation based on residual evaluation and contribution analysis and contribution analysis. *J Franklin Inst* 354:2591–2612
- Wang J, Wei HT, Cao LL, Jin QB (2013) Soft-transition sub-PCA fault monitoring of batch processes. *Ind Eng Chem Res*, pp 9879–9888
- Yoo CK, Lee JM, Vanrolleghem PA, Lee IB (2004) On-line monitoring of batch processes using multiway independent component analysis. *Chemom Intell Lab Syst*, pp 15163

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 7

Kernel Fisher Envelope Surface for Pattern Recognition



It is found that the batch process is more difficultly monitored compared with the continuous process, due to its complex features, such as nonlinearity, non-stable operation, unequal production cycles, and most variables only measured at the end of batch. Traditional methods for batch process, such as multiway FDA (Chen 2004) and multi-model FDA (He et al. 2005), cannot solve these issues well. They require complete batch data only available at the end of a batch. Therefore, the complete batch trajectory must be estimated real time, or alternatively only the measured values at the current moment are used for online diagnosis. Moreover, the above approaches do not consider the problem of inconsistent production cycles.

To address these issues, this chapter presents the modeling of kernel Fisher envelope surface (KFES) and applies it to the fault identification of batch process. This method builds separate envelope models for the normal and faulty data based on the eigenvalues projected to the two discriminant vectors of kernel FDA. The highlights of the proposed method include the kernel project aiming at the nonlinearity, data batch-wise unfolding, envelope modeling aiming at unequal cycles, and new detection indicator easily for online implementation.

7.1 Process Monitoring Based on Kernel Fisher Envelope Analysis

7.1.1 Kernel Fisher Envelope Surface

Consider the batch-wise data matrix with I batches, i.e.,

$$X(k) = [X^1(k), X^2(k), \dots, X^I(k)]^T,$$

where \mathbf{X}^i consists of $n_i (i = 1, \dots, I)$ row vectors and each row vector is a sample vector $\mathbf{X}_j^i(k), j = 1, \dots, n_i$ acquired at time k and batch i . Each batch has the same sampling period but different operation cycles, i.e., batch i has $n_i (i = 1, 2, \dots, I)$ sampling point. Suppose K is the largest sampling moment among all batches, i.e., $K = \max [n_1, n_2, \dots, n_I]$.

Let $\Phi(x)$ be a nonlinear mapping rule that maps the sample data from the original space \mathbf{X} into the high-dimensional space \mathbf{F} . Suppose that each batch is treated as a class, then the whole data set can be categorized as I classes. The optimal discriminant vector \mathbf{w} is obtained using the exponential criterion function in the feature space \mathbf{F} . Since computing $\Phi(x)$ is not always feasible, a kernel function can be introduced,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j). \quad (7.1)$$

This kernel function is introduced to allow the dot product in \mathbf{F} without directly computing Φ . According to the principle of reproducing kernel, any solution $\mathbf{w} \in \mathbf{F}$ of discriminant vector must lie in the span of all training samples of \mathbf{w} :

$$\mathbf{w} = \sum_{(i=1)}^n \alpha_i \Phi(\mathbf{x}_i) = \Phi \alpha, \quad (7.2)$$

where $\mathbf{x}_m, m = 1, \dots, n, n = n_1 + n_2 + \dots + n_I$ is the row vector of \mathbf{X} . $\Phi(x) = [\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_n)]$; $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$. The eigenvalues T_{ij} are obtained by projecting the sampled values $\Phi(\mathbf{x}_j^i)$ in space onto \mathbf{w} .

$$\begin{aligned} T_{ij} &= \mathbf{w}^T \Phi(\mathbf{x}_j^i) = \alpha^T \Phi^T \Phi(\mathbf{x}_j^i) \\ &= \alpha^T [\Phi(\mathbf{x}_1)^T \Phi(\mathbf{x}_j^i), \dots, \Phi(\mathbf{x}_n)^T \Phi(\mathbf{x}_j^i)] \\ &= \alpha^T \xi_j^i. \end{aligned} \quad (7.3)$$

The kernel sample vector ξ_j^i is defined as follows:

$$\xi_j^i = [K(\mathbf{x}_1, \mathbf{x}_j^i), K(\mathbf{x}_2, \mathbf{x}_j^i), \dots, K(\mathbf{x}_n, \mathbf{x}_j^i)]^T. \quad (7.4)$$

Consider the projection of within-class mean vector $\mathbf{m}_i^\Phi, i = 1, \dots, I$, the kernel within-class mean vector μ_i is obtained as

$$\mu_i = \left[\frac{1}{n_i} \sum_{j=1}^{n_i} K(\mathbf{x}_1, \mathbf{x}_j^i), \dots, \frac{1}{n_i} \sum_{j=1}^{n_i} K(\mathbf{x}_n, \mathbf{x}_j^i) \right]^T. \quad (7.5)$$

Then the kernel between-class scatter matrix \mathbf{K}_b is

$$\mathbf{K}_b = \sum_{i=1}^I \frac{n_i}{n} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^T. \quad (7.6)$$

Similarly, consider the projection of overall mean vector \mathbf{m}_0^Φ to the discriminant vector \mathbf{w} , the kernel overall mean vector $\boldsymbol{\mu}_0$ and between-class scatter matrix \mathbf{K}_w can be calculated as

$$\boldsymbol{\mu}_0 = \left[\frac{1}{n} \sum_{j=1}^n K(\mathbf{x}_1, \mathbf{x}_j), \dots, \frac{1}{n} \sum_{j=1}^n K(\mathbf{x}_n, \mathbf{x}_j) \right]^T \quad (7.7)$$

$$\mathbf{K}_w = \frac{1}{n} \sum_{i=1}^I \sum_{j=1}^{n_i} (\boldsymbol{\xi}_j^i - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_j^i - \boldsymbol{\mu}_i)^T. \quad (7.8)$$

The discriminant function with the objective of maximizing between class and minimizing within class is equivalent to

$$\begin{aligned} \max J(\boldsymbol{\alpha}) &= \frac{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_b \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_w \boldsymbol{\alpha})} \\ &= \frac{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T) \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T) \boldsymbol{\alpha})}, \end{aligned} \quad (7.9)$$

where $\mathbf{K}_b = \mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T$ and $\mathbf{K}_w = \mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T$ are eigenvalue decompositions of between-class and within-class scatter matrices, respectively. To construct the envelope surface model, it is usually assumed that two discriminant vectors are obtained, namely, the optimal discriminant vector and the suboptimal discriminant vector. The kernel sampling vector for sampling point k of batch i is $\boldsymbol{\xi}_k^i$, which is projected onto the two discriminant vectors to obtain the eigenvalues T_{ik}^1 and T_{ik}^2 .

The eigenvalue vectors of all batch at time k in the first two projection direction are $[T_{1k}^1, T_{2k}^1, \dots, T_{Ik}^1]$ and $[T_{1k}^2, T_{2k}^2, \dots, T_{Ik}^2]$. Their means of the two eigenvalue vectors are $mean_1(k)$ and $mean_2(k)$, respectively. Define that

$$\begin{aligned} \max_1(k) &= \max [|T_{1k}^1 - mean_1(k)|, \dots, |T_{Ik}^1 - mean_1(k)|] \\ \max_2(k) &= \max [|T_{1k}^2 - mean_2(k)|, \dots, |T_{Ik}^2 - mean_2(k)|], \end{aligned} \quad (7.10)$$

where $\max(k)$ is the larger between $\max_1(k)$ and $\max_2(k)$, for all $k = 1, 2, \dots, K$. Then the envelope surface in high-dimensional space is

$$(x_k - mean_1(k))^2 + (y_k - mean_2(k))^2 = \max(k)^2 (k = 1, 2, \dots, K), \quad (7.11)$$

where (x_k, y_k) is a projection of original data in the feature space, i.e., x_k is the eigenvalue in the optimal discriminant direction and y_k is the eigenvalue in the

suboptimal discriminant direction. Equation (7.11) gives the envelope surface with the maximum variation which allows the eigenvalues at different sampling times for this kind of data.

Unequal Cycle Discussion

Suppose the production period of each batch is different, i.e., n_i is varying with the batch i . The envelope surface model is similar as described above, but the difference lies in the composition of the eigenvalue vector. As a simple example, it is known that there are I batches of data in a training data set, and the sampling moment k for each batch varies from 1 to K , K is the largest sampling moment of all batches. Suppose only batch i does not reach the maximum sampling moment K , $k = 1, \dots, n_i, n_i \leq K$. The corresponding eigenvalue vectors are $[T_{1k}^1, T_{2k}^1, \dots, T_{I_k}^1]$ and $[T_{1k}^2, T_{2k}^2, \dots, T_{I_k}^2]$ if $k = 1, \dots, n_i$. When the time increases $k = n_i + 1, \dots, K$, the eigenvalue vectors are $[T_{1k}^1, T_{2k}^1, \dots, T_{(i-1)k}^1, T_{(i+1)k}^1, \dots, T_{I_k}^1]$ and $[T_{1k}^2, T_{2k}^2, \dots, T_{(i-1)k}^2, T_{(i+1)k}^2, \dots, T_{I_k}^2]$. Obviously, the parameters in envelope surface model (7.11), $\max(k)$, $\max_1(k)$, and $\max_2(k)$ are time varying with k .

7.1.2 Detection Indicator

Define the detection indicators as follows:

$$\begin{aligned} P_1(k) &= \frac{|T_k^1 - \text{mean}_1(k)|}{\max(k)} \\ P_2(k) &= \frac{|T_k^2 - \text{mean}_2(k)|}{\max(k)} \\ T(k) &= (T_k^1)^2 + (T_k^2)^2, \end{aligned} \tag{7.12}$$

where T_k^1 and T_k^2 are the eigenvalues obtained by mapping the real-time sampling vector \mathbf{x}_k onto the discriminant vector in the higher dimensional space. When the trajectory of eigenvalues at that moment is contained within the envelope surface, there must be $P_1(k) < 1$ and $P_2(k) < 1$ holds. If the difference between the new batch of data and the training data for this type of envelope surface model is large, the Gaussian kernel function used in the kernel Fisher criterion is almost zero, such that $T_k^1=0, T_k^2=0$, i.e., $T(k) = 0$. Thus, for a given measured data, using the above indicators, a judgement can be made. When $P_1(k) < 1$, $P_2(k) < 1$, and $T(k) = 0$ does not occur, the data sampled at that moment belong to this mode type. When $T(k) = 0$ occurs consistently, it indicates that the sampled data does not belong to this mode type.

It is assumed that it has been determined from the normal operating envelope surface model that the batch of data is faulty at some point. Fault identification is carried out using fault envelope surface models. Consider one of the fault envelope surface models, if $P_1(k) < 1$, $P_2(k) < 1$, and no $T(k) = 0$, then the batch fault is in current fault type. If $T(k) = 0$ appears consistently in each envelope model, then the fault that exists may be a new one. When that fault occurs multiple times, the pattern type needs to be updated and an additional envelope model need to be constructed for new fault.

The fault identification using the proposed kernel Fisher envelope surface analysis (KFES) is given as follows. Its fault monitoring flowchart is shown in Fig. 7.1.

Fault Monitoring Algorithm Based on KFES

Step 1: Collect the historical data with S fault categories. Construct S envelope surface models for each category based on the description in Sect. 7.1.1:

$$(x_k - \text{mean}_1^S(k))^2 + (y_k - \text{mean}_2^S(k))^2 = \max^S(k)^2, \quad (k = 1, 2, \dots, K). \quad (7.13)$$

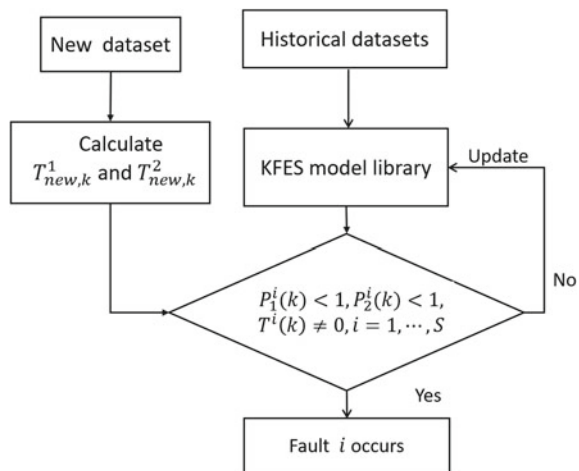
Then store all the model parameters $\text{mean}_1^S(k)$, $\text{mean}_2^S(k)$, and $\max^S(k)$, ($k = 1, 2, \dots, K$). Thus, the envelope model library $\text{Env-model}(S, k)$ is constructed.

Step 2: Sample the real-time data x_k . After normalization, the kernel sampling vector ξ_k is obtained.

Step 3: Under the known S fault envelope surface model at time k , project the kernel sampling vector ξ_k of x_k in the direction of the discriminant vectors. Calculate the corresponding project eigenvalues T_k^1 , T_k^2 and detection indicators. If $P_1^S(k) < 1$, $P_2^S(k) < 1$, and $T^S(k) \neq 0$, then the fault belongs to category S .

Step 4: If detection indicators in Step 3 are not satisfied for all known fault type, it is possible that a new fault has occurred. When that unknown fault lasts for a

Fig. 7.1 Fault monitoring flowchart based on fault envelope surface model



period of time, the model library needs to be updated. The envelope surface for this new fault is modeled according to the accumulated new batch data as Step 1, and augmented into the model library.

7.1.3 KFES-PCA-Based Synthetic Diagnosis in Batch Process

The basic idea of synthetic diagnosis integrates the advantage of KFES and PCA. It builds a multiway PCA model for normal operating in the historical database and calculates the monitoring statistics T^2 and SPE of PCA model and their control limits. The multiway PCA is used for fault detection. For the fault data in the historical database, the KFES is modeled for known fault categories. The KFES analysis is used for fault identification. The modeling and online monitoring process of synthetic diagnosis is shown in Fig. 7.2.

The normal operating data and S classes fault data were obtained from the historical data set. Firstly, the normal operating condition data $X(I \times J \times K)$ is expanded into two-dimensional matrix $X(I \times JK)$ in the time direction. After normalization, the data is unfolding again as $Y(IK \times J)$ in the batch direction. Perform multiway PCA on the matrix to obtain score matrix $T(IK \times R)$ and load matrix $P(J \times R)$, where R is the number of principal components. Then calculate the control limits of the statistics T^2 and SPE.

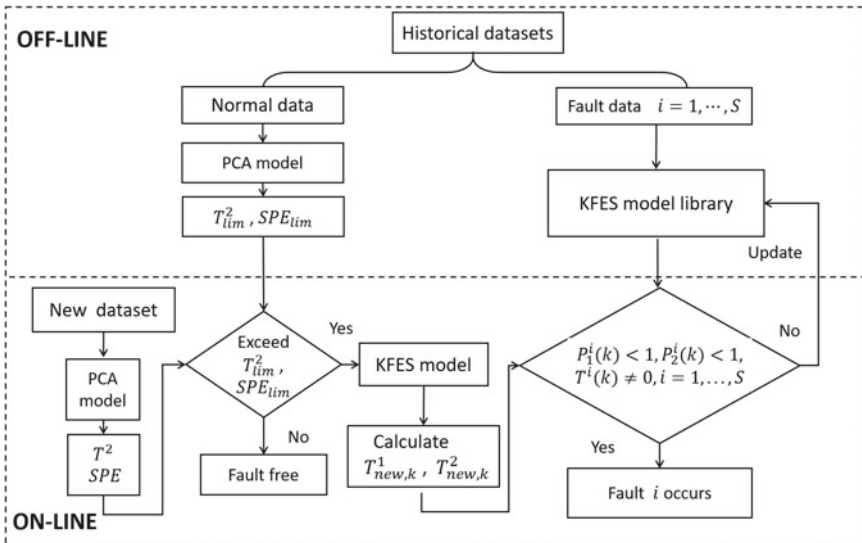


Fig. 7.2 Process monitoring flowchart based on KFES-PCA

Instead of using contribution maps, kernel Fisher envelope surface analysis is used for fault diagnosis. Assume that there are S classes in the fault data set. The envelope surface model is first constructed for each fault type. When the new data $x_{new,k}$ is obtained, it should be judged whether the current operation is normal by PCA model. If the T^2 and SPE exceed the control limits, and the fault is detected. Then we can identify the type of fault by KFES model library. If the eigenvalues do not satisfy the indicators in all the known fault models, this fault seems to be new. As long as enough data to KFES modeling are collected, update the new fault model in the model library.

Process Monitoring Algorithm Based on KFES-PCA

A. Offline Modeling

Step 1: Develop an improved multiway PCA model for normal operating conditions data, calculate the statistics T^2 and SPE, and determine the corresponding control T_{lim}^2 and SPE_{lim} based on the score matrix $T(KI \times R)$ and load matrix $P(J \times R)$ obtained from the normal model.

Step 2: Apply KFES analysis to the fault data and construct a fault envelope for each type of fault separately. Find the optimal discriminant weight matrix W_α , the mean $mean_1(k)$, $mean_2(k)$, and maximum $\max(k)$ of the eigenvalue vectors.

Step 3: Store T_{lim}^2 and SPE_{lim} , the discriminant weight matrix W_α for each fault type, the mean $mean_1(k)$, $mean_2(k)$, and the maximum $\max(k)$ of the eigenvalues.

B. Online Monitoring

Step 1: Normalize the new batch of data $x_{new,k}(J \times 1)$ at the k th sampling moment.

Step 2: Calculate the value of statistics T^2 and SPE and determine if they are over the limit, if not, back to the first step. Otherwise proceed to the next step.

Step 3: The known fault envelope surface model is used for fault identification at that moment. $x_{new,k}(J \times 1)$ is the sampling data obtained at the first k sampling moment, normalized and projected onto the discriminant weight matrix W_α of the kernel Fisher envelope model to obtain the eigenvalues T_k^1 and T_k^2 . The eigenvalues are substituted into the index, $P_1(k) < 1$, $P_2(k) < 1$, and no $T(k) = 0$, and the fault is in this fault type.

Step 4: If a fault has been detected based on step 2, but it does not belong to any known fault type obtained from step 3, this indicates that a new fault may have occurred. When that unknown fault has occurred several times, the mode type needs to be updated and the envelope surface model for that fault needs to be augmented with the accumulated batches of new faults in an offline situation.

7.2 Simulation Experiment Based on KFES-PCA

The fed-batch penicillin fermentation simulation platform is used to verify the effectiveness of the KFES-PCA method for fault diagnosis here. Eleven variables affecting the fermentation reaction were selected for modeling, and these variables were air flow, stirring power, substrate flow acceleration rate, temperature, etc. Three simulation failure types were selected as shown in Table 7.1. The total data sets (including 50 batches) were generated from the Pensim 2.0 simulation platform with 1 h sampling interval, consisting of 20 batches of normal operation, 10 batches of bottom flow acceleration rate drop failure, 10 batches of agitation power drop failure, and 10 batches of air flow drop failure. The normal operation data are obtained at different product cycles, one batch with 95 h, two batches with 96 h, two batches with 97 h, three batches with 98 h, five batches with 99 h, and seven batches with 100 h. Similarly, change the reaction duration of each batch, and change the time and amplitude of the failure occurrence. The failure batch data are collected.

Figure 7.3a–d gives the envelope surface of the kernel Fisher discriminant envelope model under the normal operation and three known fault operations offline trained, respectively. Here the x -axis and y -axis represent the direction of the optimal and suboptimal discriminant vector, and the z -axis represents time.

The traditional monitoring methods, such as MPCA and MFDA, require the modeling batches to be of equal length. However, the duration of the different batches tends to change in practice. Therefore, the data of different batches must be pre-processed with equal length when using these methods. The proposed KFES-PCA method unfolds the data in the batch direction during the preprocessing, which can simply cope with the unequal batches of data and therefore easily performed in practice.

The following experiments are designed to perform the online detection with the known fault and new unknown fault data, respectively. The two batches of test data are not included in the training data in order to obtain a valid validation. In addition, a comparative validation using the conventional contribution map method and the improved MFDA method is also carried out (Jiang et al. 2003).

Table 7.1 Types of faults in penicillin fermentation processes

Fault number	Fault type
1	Base flow rate down (step)
2	Agitator power down (step)
3	Air flow down (step)

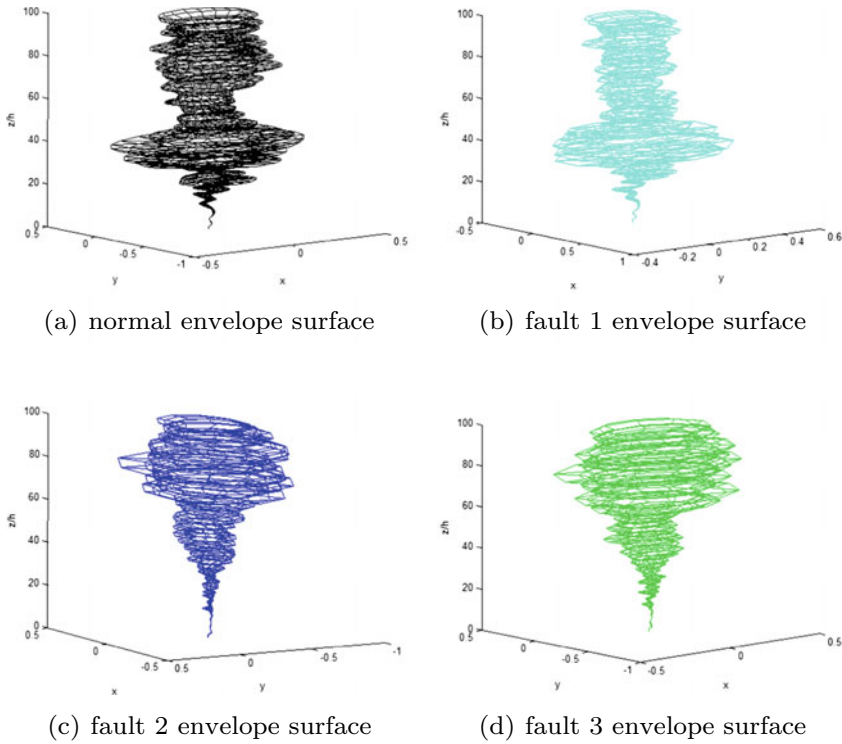


Fig. 7.3 Envelope surface for normal and three fault operations

7.2.1 Diagnostic Effect on Existing Fault Types

Experiment 1: Step Drop Fault at Stirring Power

A fault batch data is regenerated for testing with the stirring power drop fault. The fault occurs at 50h with a step disturbance of -12% in magnitude until the process ends. The sampled data is first monitored based on T^2 and SPE statistics, as shown in Fig. 7.4. It can be seen that T^2 and SPE continues to exceed the limit from 50h to process end. A failure can be detected when it occurs at 50h. Table 7.2 records the indicators when it is diagnosed using the envelope surface model of fault 2. It shows that there are $P_1(k) < 1$, $P_2(k) < 1$, and no $T(k) = 0$ with time through from 50h to 100h. So it is concluded that this fault of testing batch belongs to fault 2. Figure 7.5 shows the diagnosis results based on each envelope surface model. It can also be seen that the fault matches with the second type of fault, a mixing power drop fault.

The contribution plot is used to analyze the testing data at 50h, as shown in Fig. 7.6. It is found that the second variable contributes significantly to both the statistics T^2 and SPE. This also diagnoses that the fault belongs to fault 2. Therefore, the envelope surface model is equally successful in diagnosing the fault type when compared with the contribution plot method.

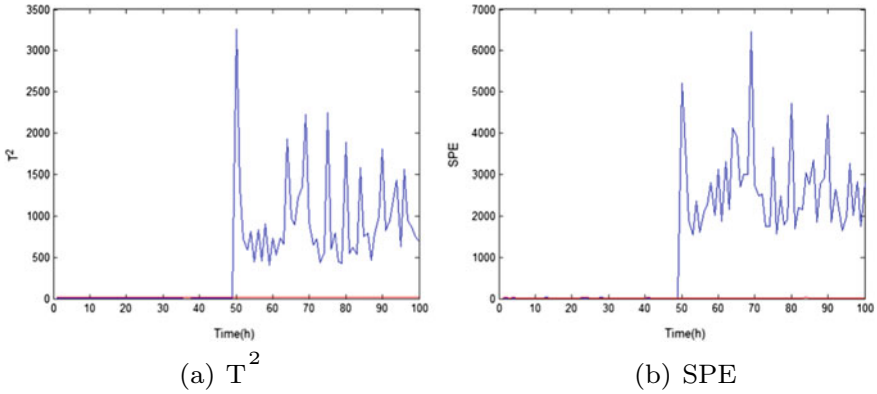


Fig. 7.4 Monitoring statistics of KFES-PCA method: experiment 1

Table 7.2 The indicators detected in fault 2 envelope surface: experiment 1

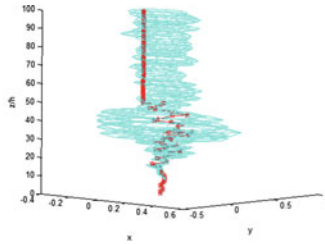
k	50	51	52	53	54	55	56	57	...	100
T_k^1	0.044	0.025	0.028	-0.011	0.032	0.062	0.110	0.083	...	-0.005
T_k^2	-0.159	-0.145	-0.233	-0.141	-0.173	-0.205	-0.271	-0.202	...	-0.241
$P_1(k)$	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1	...	< 1
$P_2(k)$	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1	...	< 1

The comparison experiment is finished based on the improved MFDA method, as shown in Fig. 7.7. The horizontal coordinate is time. The vertical coordinate is fault type, where 0 represents the normal operation, and 1, 2, 3, and 4 correspond to fault 1, fault 2, fault 3, and unknown fault, respectively. It can be seen that the improved MFDA has a relatively high rate of misdiagnosis and its diagnosis result is not ideal.

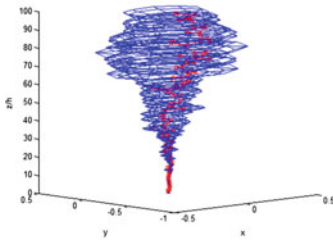
Experiment 2: Step Drop Fault at Air Flow

The testing fault is air flow drop failure and testing data is regenerated with the failure which occurred in 58 h, and its amplitude is -10% step disturbance until the process ends. The monitoring statistics T^2 and SPE are given in Fig. 7.8. The T^2 and SPE continue to exceed the control limits from 58 h to the end, so a fault is detected at 58 h in real time.

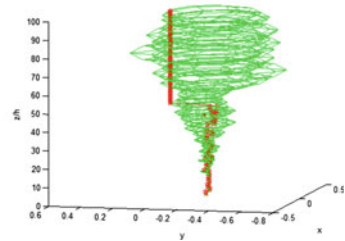
Figure 7.9 is the monitoring result using the proposed envelope surface model. Table 7.3 records the indicators when using the envelope surface model of fault 3. It can be seen that there are $P_1(k) < 1$, $P_2(k) < 1$, and no $T(k) = 0$ between 58 h and 100 h, so it is judged that the fault which occurred in this testing batch belongs to fault 3. Figure 7.9 shows all the diagnosis results with different envelope surface models. It can also be seen that this fault matches with the model of fault 3, i.e., the air flow drop fault.



(a) fault 1 envelope surface

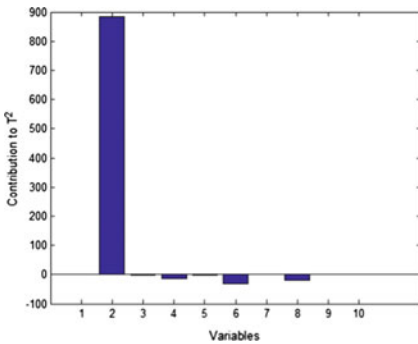


(b) fault 2 envelope surface

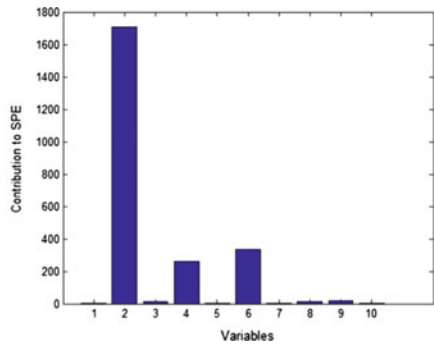


(c) fault 3 envelope surface

Fig. 7.5 Fault diagnosis based on envelope surfaces: experiment 1



(a) T^2 contribution



(b) SPE contribution

Fig. 7.6 Contribution plot to statistics T^2 and SPE at 50h

The contribution plot of the sampling data at 58h is shown in Fig. 7.10, where variables 1, 4, 6, and 8 contribute more to the statistic T^2 . The variable 3 contributed more to the statistic SPE. The diagnosis result is not significant. Therefore, the envelope surface method can successfully diagnose faults that are not diagnosed by the contribution plot.

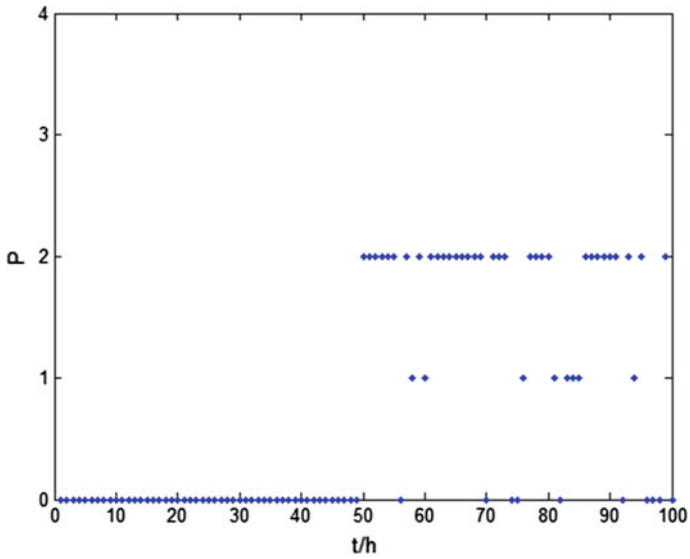


Fig. 7.7 Fault diagnosis based on improved MFDA: experiment 1

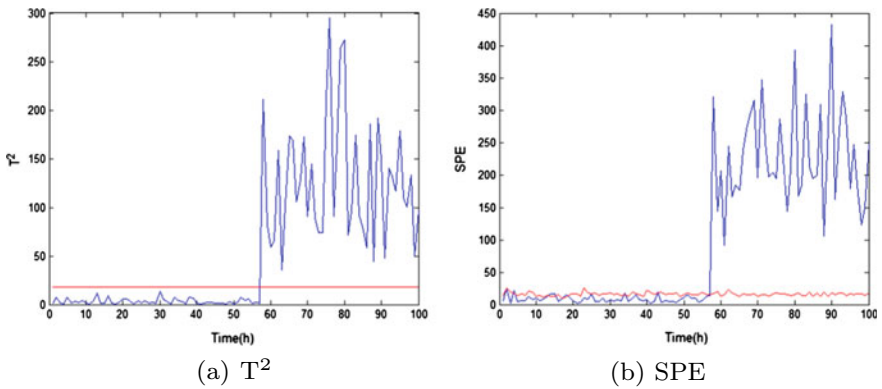


Fig. 7.8 Monitoring statistics of KFES-PCA method: experiment 2

The comparison results of the improved MFDA method are given in Fig. 7.11. It shows a relatively higher rate of misdiagnosis and its diagnosis result is not very satisfactory, compared with the proposed KFES-PCA.

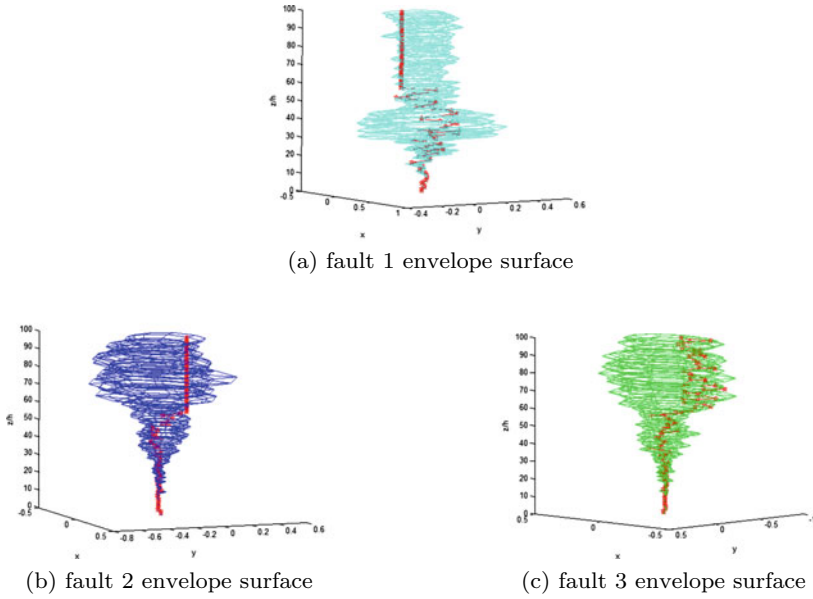


Fig. 7.9 Fault diagnosis based on envelope surfaces: experiment 2

Table 7.3 The indicators detected in fault 3 envelope surface: experiment 2

k	58	59	60	61	62	63	64	65	...	100
T_k^1	-0.110	-0.110	-0.171	-0.133	-0.220	-0.182	-0.100	-0.054	...	-0.066
T_k^2	-0.237	-0.162	-0.259	-0.141	-0.393	-0.378	-0.273	-0.332	...	-0.295
$P_1(k)$	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1	...	< 1
$P_2(k)$	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1	...	< 1

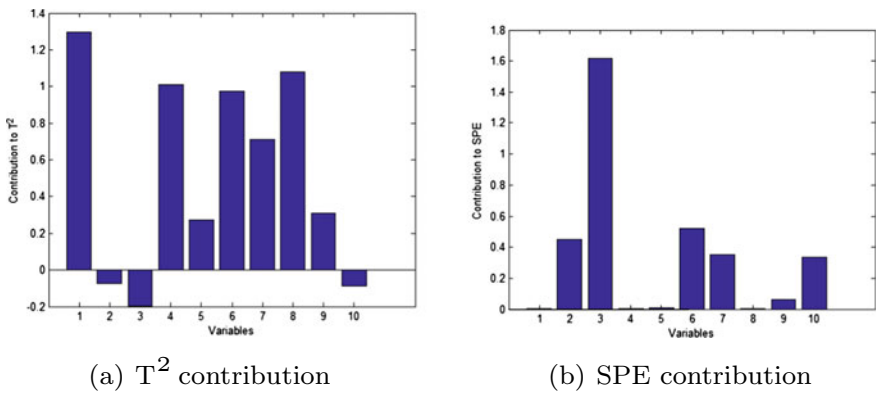


Fig. 7.10 Contribution plot to statistics T^2 and SPE at 58 h: experiment 2

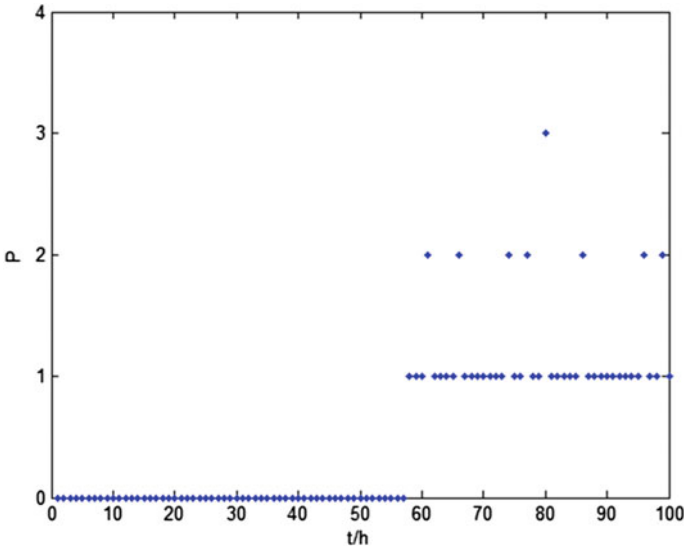


Fig. 7.11 Fault diagnosis based on improved MFDA: experiment 2

7.2.2 Diagnostic Effect on Unknown Fault Types

Experiment 3: Slope Drop Fault at Air Flow Rate

Here a new fault is used to test the diagnosis ability of the proposed KFES-PCA method. The slope faults different from the known three fault types are considered. The test fault is a ramp fault in which the air flow rate drops by -15% at 50h. Firstly, the T^2 and SPE statistics are used to detect this new fault. Figure 7.12 shows that the T^2 and SPE statistics both detect this fault in time at 50h.

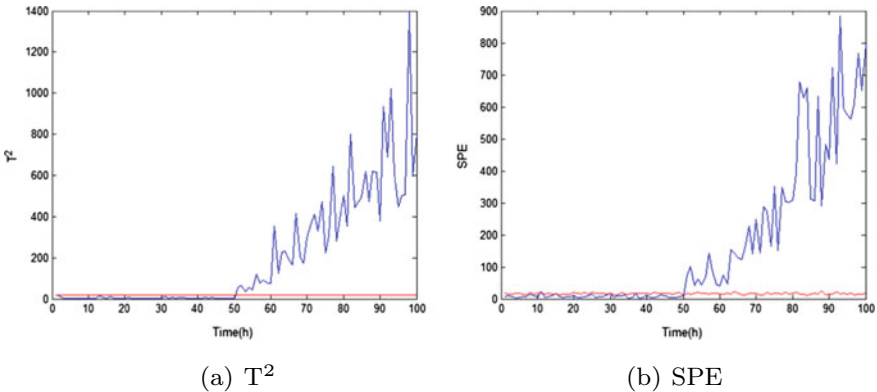
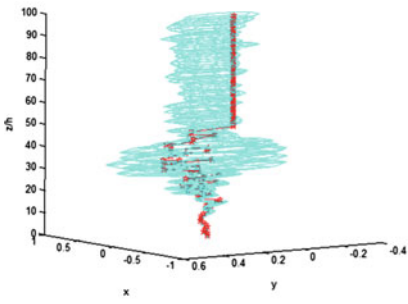


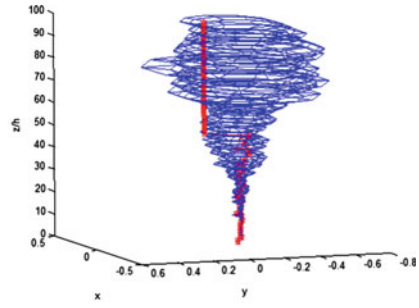
Fig. 7.12 Monitoring statistics of KFES-PCA: experiment 3

Table 7.4 The indicator detected in fault 3 envelope surface: experiment 3

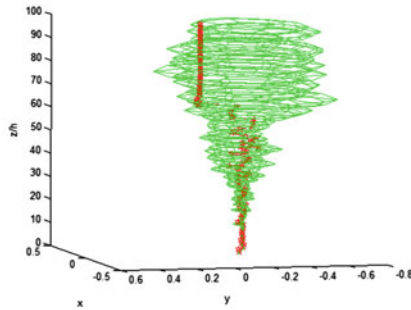
k	50	51	52	53	54	55	56	57	...	100
T_k^1	0	0	0	0	0	0	0	0	...	0
T_k^2	0	0	0	0	0	0	0	0	...	0
$T(k)$	0	0	0	0	0	0	0	0	...	0



(a) fault 1 envelope surface



(b) fault 2 envelope surface



(c) fault 3 envelope surface

Fig. 7.13 Fault diagnosis based on different envelope surfaces: experiment 3

The known envelope surface models are used to diagnose this fault. Table 7.4 records that all the indicators are zero when the envelope surface model of fault 3 is used for diagnosis. It means that no fault 3 has occurred. The same indicator results are obtained from the envelope surface models of other known faults. Figure 7.13 gives the diagnosis result under the different envelope surface models. So this fault does not belong to the known fault category and is diagnosed as a new fault. Therefore, the proposed method realizes the real-time diagnosis for unknown faults.

The diagnosis result of improved MFDA method is given in Fig. 7.14. It can be seen that the improved MFDA does not make a timely and correct diagnosis when the fault occurs. It gives a wrong diagnosis result, fault type 3. The correct result is

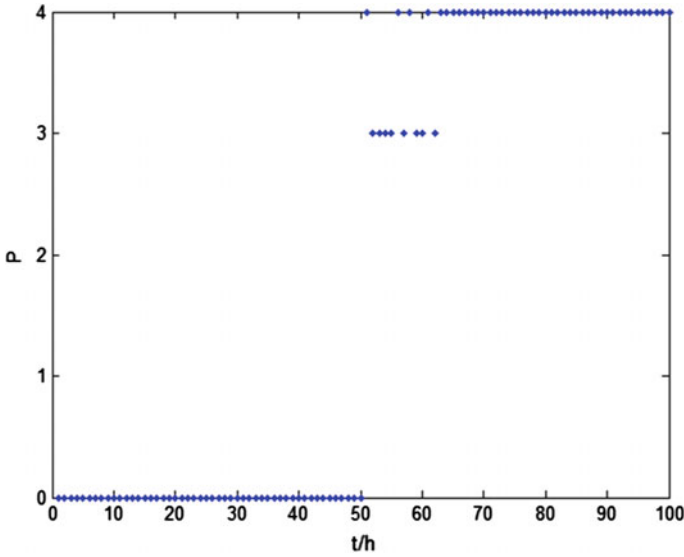


Fig. 7.14 Fault diagnosis based on improved MFDA: experiment 3

reported until 63 h. This fault is diagnosed as a new fault, and there is a 13 h delay. Therefore, the improved MFDA method failed to identify new faults.

7.3 Conclusions

This chapter describes a monitoring method based on KFES-PCA for batch processes. The production cycles of batch processes are often unequal, and monitoring methods for batch processes generally require batch data with consistent production cycles. Although data preprocessing can result in equal cycles, these methods can result in the loss of important information about faults. In addition, many existing monitoring methods often require a complete production trajectory for online monitoring, and filling or estimating unknown values inevitably leads to a decrease in diagnostic performance. To address the above two problems, the modeling process of the KFES method is described in detail and an online monitoring flowchart is presented. Furthermore, a batch fault diagnosis method integrating the KFES and the improved PCA method is proposed. The method is applied to a penicillin fermentation simulation platform and compared with the traditional contribution map method and the improved MFDA method. The results show that the proposed method has better monitoring performance, and it can diagnose faults early and effectively and has the ability to identify unknown faults.

References

- Chen YH (2004) Monitoring batch processes using multiway fisher discriminant analysis. *Inf Technol Sci Jilin Univ* 22:384–387
- He Q, Qin S, Wang J (2005) A new fault diagnosis method using fault directions in fisher discriminant analysis. *AIChE J* 51:555
- Jiang LY, Xie L, Wang SQ (2003) Fault diagnosis for batch processes by improved multi-model fisher discriminant analysis. *Chin J Chem Eng* 14:343–348

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 8

Fault Identification Based on Local Feature Correlation



Industrial data variables show obvious high dimension and strong nonlinear correlation. Traditional multivariate statistical monitoring methods, such as PCA, PLS, CCA, and FDA, are only suitable for solving the high-dimensional data processing with linear correlation. The kernel mapping method is the most common technique to deal with the nonlinearity, which projects the original data in the low-dimensional space to the high-dimensional space through appropriate kernel functions so as to achieve the goal of linear separability in the new space. However, the space projection from the low dimension to the high dimension is contradictory to the actual requirement of dimensionality reduction of the data. So kernel-based method inevitably increases the complexity of data processing. For this reason, we have proposed another kind of nonlinear processing approach based on the manifold learning, a class of unsupervised model that seeks to describe data sets as low-dimensional manifold embedded in high-dimensional spaces. It characterizes the original data as a low-dimensional manifold to achieve the goal of nonlinear correlation processing. This strategy is consistent with the goal of dimensionality reduction. Furthermore, manifold learning fits the nonlinear correlation by means of piecewise linearization in an intuitive sense. It has significantly less complexity compared to the kernel mapping method.

This chapter carries out the pattern classification techniques for multivariate variables with strong nonlinear correlation and applies them to the fault identification of batch process. Two kinds of pattern classification methods are given in this chapter: (1) kernel exponential discriminant analysis (KEDA): this method addresses the nonlinear correlation properties among multi-variables at two levels, kernel mapping and exponential discrimination, respectively. It can significantly improve the classification accuracy compared with the traditional FDA method. (2) The fusion method is based on manifold learning and discriminant analysis: two different fusion strategies, local linear exponential discriminant analysis (LLEDA) and neighborhood-preserving embedding discriminant analysis (NPEDA), are given, respectively. Here

locally linear embedding (LLE) is a popular algorithm of manifold learning. They both combine the advantage of global discriminant analysis with the local structure preserving. LLEDA is a parallel strategy to find a trade-off projection vector between the local geometric structure preserving and the global data classification. NPEDA is a cascaded strategy whose dimensionality reduction process is implemented in two serial steps. The two methods emphasize the intrinsic structure of the data while utilizing the global discriminant information, so they have better classification than the traditional EDA method. Finally, a kind of hybrid fault diagnosis scheme is given for the complex industrial process, which consists of PCA-based fault detection, hierarchical clustering-based pre-diagnosis, and LLEDA-based final identification.

8.1 Fault Identification Based on Kernel Discriminant Exponent Analysis

8.1.1 Methodology of KEDA

The kernel exponent discriminant analysis (KEDA) is also a discriminative classification method, which aims to find a series of discriminant vectors that can transform the data into the kernel space and achieve the greatest separation between different types of data in the projection direction.

Consider the batch process data set with I batches, i.e.,

$$\mathbf{X}(k) = [\mathbf{X}^1(k), \mathbf{X}^2(k), \dots, \mathbf{X}^I(k)]^T,$$

where \mathbf{X}^i consists of n_i , $i = 1, \dots, I$ row vectors, and each row vector is a sample vector $\mathbf{X}_j^i(k)$, $j = 1, \dots, n_i$ acquired at time k and batch i . According to the analysis from equations (7.1)–(7.9) in Sect. 7.1.1, the optimization function of kernel Fisher discrimination analysis (KFDA) is given as follows,

$$\begin{aligned} \max J(\boldsymbol{\alpha}) &= \frac{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_b \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T \mathbf{K}_w \boldsymbol{\alpha})} \\ &= \frac{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T) \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T) \boldsymbol{\alpha})}, \end{aligned} \quad (8.1)$$

where $\mathbf{K}_b = \mathbf{V}_b \boldsymbol{\Lambda}_b \mathbf{V}_b^T$ and $\mathbf{K}_w = \mathbf{V}_w \boldsymbol{\Lambda}_w \mathbf{V}_w^T$ are eigenvalue decompositions of between-class and within-class scatter matrices, respectively. $\boldsymbol{\Lambda}_b = \text{diag}(\lambda_{b1}, \lambda_{b2}, \dots, \lambda_{bn})$, and $\boldsymbol{\Lambda}_w = \text{diag}(\lambda_{w1}, \lambda_{w2}, \dots, \lambda_{wn})$ are the eigenvalues, $\mathbf{V}_b = (v_{b1}, v_{b2}, \dots, v_{bn})$, and $\mathbf{V}_w = (v_{w1}, v_{w2}, \dots, v_{wn})$ are the corresponding eigenvectors. The basic objective is to maximize the between-class distance and minimize the with-class distance simultaneously during the projection.

In order to improve the discrimination accuracy further, the discriminant function (8.1) is exponentiated:

$$\begin{aligned} \max J(\boldsymbol{\alpha}) &= \frac{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_b \exp(\boldsymbol{\Lambda}_b) \mathbf{V}_b^T) \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T (\mathbf{V}_w \exp(\boldsymbol{\Lambda}_w) \mathbf{V}_w^T) \boldsymbol{\alpha})} \\ &= \frac{\text{tr}(\boldsymbol{\alpha}^T \exp(\mathbf{K}_b) \boldsymbol{\alpha})}{\text{tr}(\boldsymbol{\alpha}^T \exp(\mathbf{K}_w) \boldsymbol{\alpha})}. \end{aligned} \quad (8.2)$$

The optimization problem (8.2) is transferred to the following generalized eigenvalue problem:

$$\begin{aligned} \exp(\mathbf{K}_b) \boldsymbol{\alpha} &= \boldsymbol{\Lambda} \exp(\mathbf{K}_w) \boldsymbol{\alpha} \\ \text{or} & \\ \exp(\mathbf{K}_w)^{-1} \exp(\mathbf{K}_b) \boldsymbol{\alpha} &= \boldsymbol{\Lambda} \boldsymbol{\alpha}, \end{aligned} \quad (8.3)$$

where $\boldsymbol{\Lambda}$ is the eigenvalue and $\boldsymbol{\alpha}$ is the corresponding eigenvector. The discriminant vectors are calculated from (8.3). Usually, the first two vectors, optimal, and suboptimal ones are selected for dimensionality reduction.

The within-class and between-class scatter matrices are exponentiated in KEDA. Consider the general property of exponential function, $e^x > x$ for any $x > 0$, so the scatter matrix of KEDA is greater than KFDA. It means KEDA has better discriminatory capability than KFDA. Moreover, if the amount of sample data is less than the number of variables, the rank of within-class scatter matrix is less than the dimension of variables. Now the within-class scatter matrix is singular, and its inversion does not exist. But both the within-class and between-class scatter matrices are exponentiated in KEDA. The exponentiated matrices must be full rank, so the singular problem caused by small samples is solved. Thus from this view, the KEDA method not only solves the small sample problem, but also efficiently classifies the sample data into different categories, which helps to improve the classification accuracy.

Let's consider the nonlinear mapping $\Phi(\mathbf{x}_k^i)$ of original sample \mathbf{x}_k^i and project it to the optimal and suboptimal discriminant directions, respectively. Then the eigenvalues $\mathbf{T}_i(k) = [T_{ik}^1, T_{ik}^2]^T$ and T_{ik}^2 are obtained, which represent the projection values in the optimal and suboptimal discriminant directions. Usually, the data in the same class shows the similar project eigenvalues in the direction of selected discrimination vectors. If the test data matches with the known fault class, it has maximum projection eigenvalue under this model, obviously nonzero. If the test data does not match with this class, the eigenvalue is small even close to zero. It is unrealistic to judge the data type simply based on the magnitude of eigenvalues. So difference degree D between two projection values $\mathbf{T}_i(k)$ and $\mathbf{T}_j(k)$ is defined as follows:

$$D_{i,j}(k) = 1 - \frac{(\mathbf{T}_i(k))^T \mathbf{T}_j(k)}{\|\mathbf{T}_i(k)\|_2 \|\mathbf{T}_j(k)\|_2}. \quad (8.4)$$

The smaller the difference D , the higher the model matched.

The KEDA-based fault classification and identification process for batch process is given as follows:

Step 1: Data preprocess. The three-dimensional data set $X(L \times J \times K)$ is batch-wise unfolded into two-dimensional data $X(LK \times J)$, normalized along the time in the batch cycle and variable-wise re-arranged.

Step 2: Kernel projection. The original data X is mapping to a high-dimensional feature space via a nonlinear kernel function, and the kernel sampling data $\xi_j^i = [K(x_1, x_j^i), K(x_2, x_j^i), \dots, K(x_n, x_j^i)]^T$ are obtained.

Step 3: KEDA modeling. The optimal kernel discriminant vectors are solved from the discriminant function equation (8.3). Project the sample data ξ_j^i to the selected kernel discriminant vectors and calculate the corresponding eigenvalues $T_i(k)$.

Step 4: Test calculation. The test sample $x_{j,new}(k)$ is collected and the corresponding eigenvalues $T_{i,new}(k)$ according to the known S classes model are calculated, respectively.

Step 5: Fault identification. The class of test data can be determined by calculating the difference degree between test sample and trained data (8.4).

8.1.2 Simulation Experiment

The proposed KEDA was used for fault identification in the penicillin fermentation process mentioned in Sect. 4.2. Here nine process variables were considered for monitoring and three faults are shown in Table 8.1. The data were generated by the penicillin simulator when the amplitude and time of fault are changed. A total of 40 batches were selected as the training data set: 10 batches for normal and known 3 faults. The KEDA method with Gaussian kernel function was used to find the optimal discriminant vectors for each type of model, and four different models were obtained.

Experiment 1: Data classification Figures 8.1, 8.2, 8.3, and 8.4 show the classification comparison of KFDA and KEDA for penicillin data: normal data and three types of fault data. When the test data are different from the known four types, the projections are also separated from each other. But the KFDA shows weaker classification performance: some faults are closer together and the boundaries are not easily distinguishable, such as fault 3 data (red \star) and test fault data (black \blacksquare) in Figs. 8.1 and 8.3. However, the KEDA works better for classifying these data, and

Table 8.1 Description of the fault type of penicillin process

No.	Faults	Types
1	Bottom logistics decline	Step
2	Decreased power of the mixer	Step
3	Decreased airflow	Step

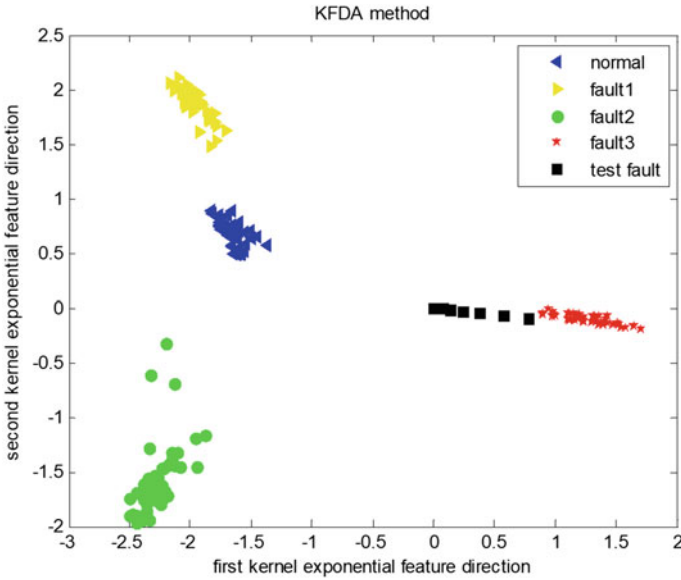


Fig. 8.1 Two-dimensional classification visualization: KFDA method

the red and black parts are classified clearly in Figs. 8.2 and 8.4. These plots show that the between-class and within-class distances have increased for different types of data in KEDA, but the between-class distance has increased by a larger magnitude than the within-class distance. So the different types of data can be better separated.

Experiment 2: Fault-type identification Let’s consider the testing data set, which also consists of the four types of data and an unknown fault data. Table 8.2 gives the eigenvalues of the four testing data calculated based on the KEDA model of fault 2. The eigenvalues are obtained by projecting the testing data to the selected optimal discriminant directions. If there is a large difference between the testing data and the training data, then the value of $\| \mathbf{u} - \mathbf{v} \|^2$ is large and the exponentiated Gaussian kernel function, $K(\mathbf{u}, \mathbf{v}) = \exp(-\| \mathbf{u} - \mathbf{v} \|^2 / (2\sigma)^2)$, is almost close to zero. However, sometimes the fault occurrence eigenvalues are not close to zero, as shown in Table 8.2. At this case, the eigenvalues of the test data need to be analyzed further.

It is impossible to show the values at any sampling instance, so we further analyze the statistical characterizes of eigenvalues projected to the optimal discrimination direction of known model. If the eigenvalue of testing data follows a normal distribution in a model, the testing data belongs to this kind of model. Conversely, if the eigenvalue does not follow a normal distribution, it means that the testing data does not match with this model. Figures 8.5, 8.6, and 8.7 give the statistical analysis of the testing data (normal, faults 1 and 3) in the known fault 3 model. The eigenvalue of fault 3 follows a normal distribution in the fault 3 model, while the normal data or fault 1 data do not follow a normal distribution.

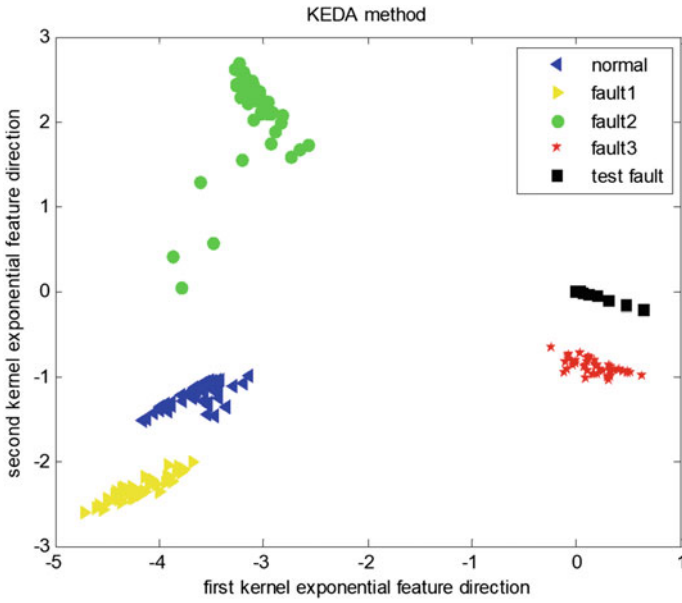


Fig. 8.2 Two-dimensional classification visualization: KEDA method

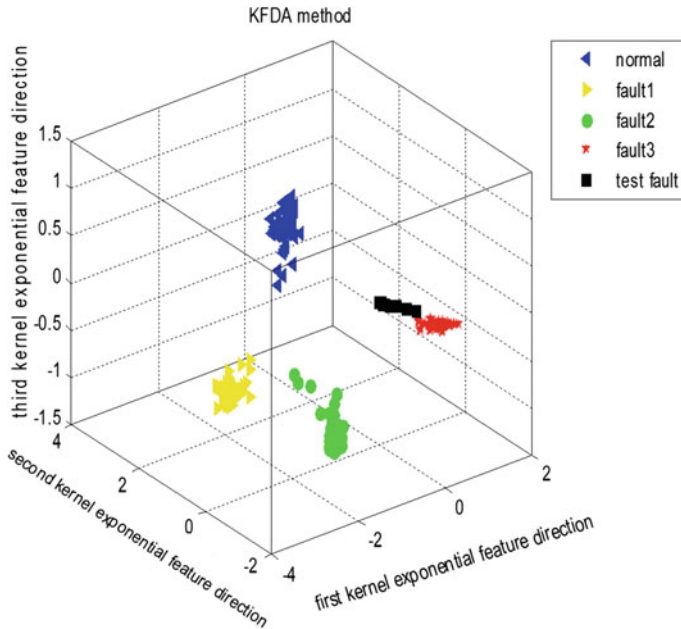


Fig. 8.3 Three-dimensional classification visualization: KFDA method

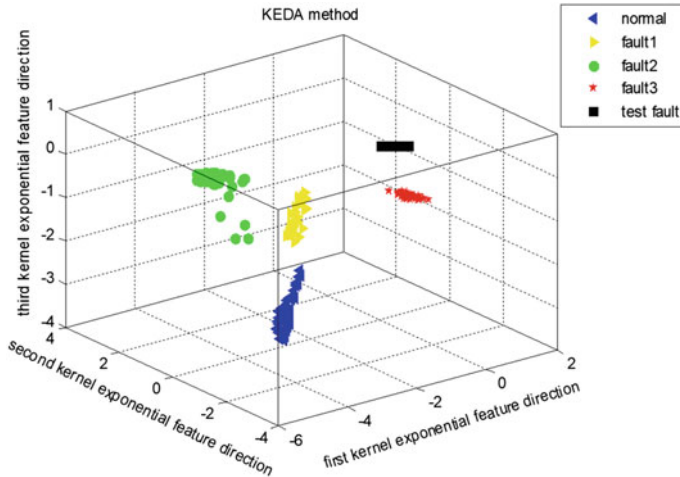


Fig. 8.4 Three-dimensional classification visualization: KEDA method

Table 8.2 The eigenvalues of test data in fault 2 model

Sampling instant	Eigenvalues of test data (T_k)				
	Normal	Fault 1	Fault 2	Fault 3	New fault
53	0.148	-0.148	-0.203	0	0
54	0.194	-0.194	0.0090	0	0
55	0.448	0	0.1660	0	0
56	0.187	0	0.1020	0	0
⋮	⋮	⋮	⋮	⋮	⋮
79	0.079	0	-0.024	0	0
80	0.103	0	-0.075	0	0
81	0.108	0	-0.084	0	0
82	0.041	0	-0.059	0	0

Moreover, the difference degree between test data and known model is used to determine the type of fault. The results are shown in Table 8.3. Since some of the test data have zero eigenvalues in the known model, and the denominators in the definition (8.4) are zero, the different degree cannot be calculated and expressed as “-”. The difference degree is small if the test data belongs to the known type model, and large if the test data does not belong to the model. It is found that the test data has the smallest different degree in the matching model.

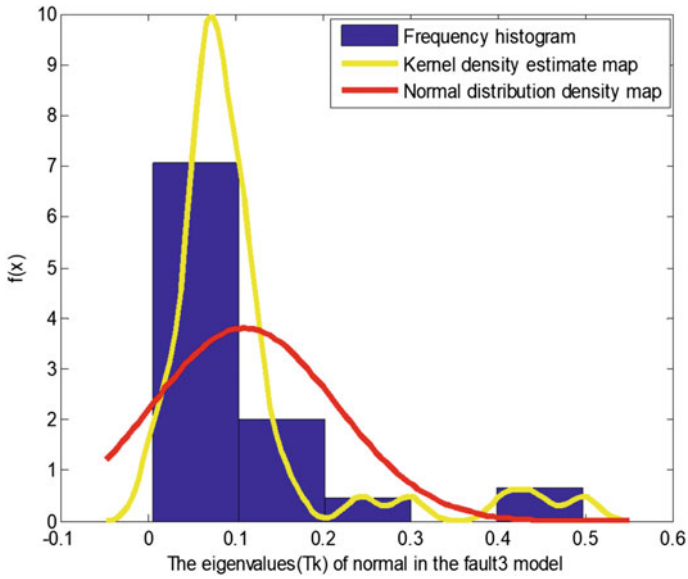


Fig. 8.5 The eigenvalues of test normal data in fault 3 model

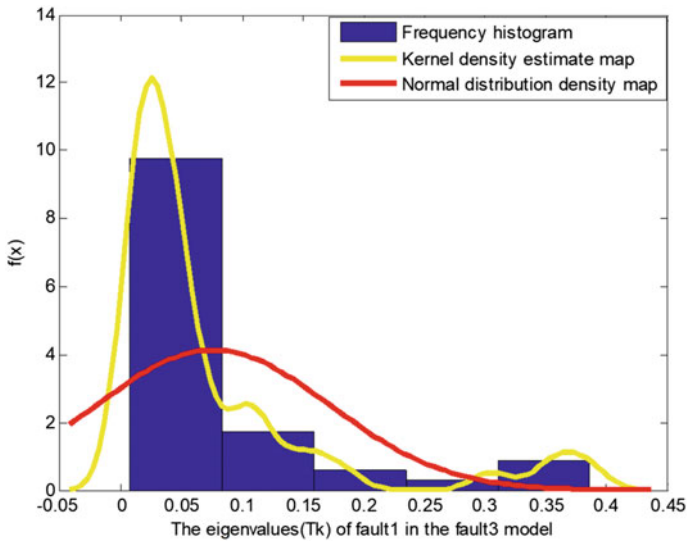


Fig. 8.6 The eigenvalues of test fault 1 data in fault 3 model

Fig. 8.7 The eigenvalues of test fault 3 data in fault 3 model

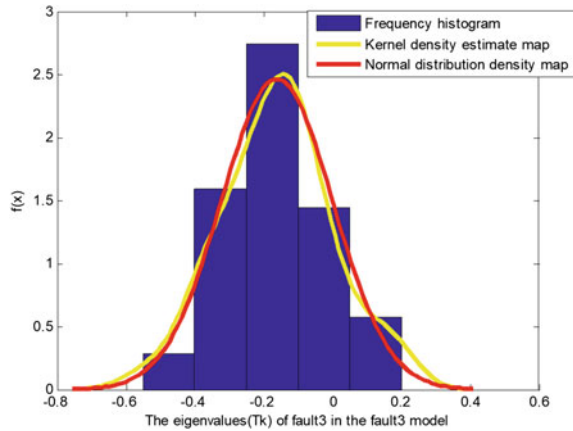


Table 8.3 The difference degree of test data in different models

Type of test data	Normal model	Fault 1 model	Fault 2 model	Fault 3 model
Normal	0.516679	0.669503	1.448272	1.630094
Fault 1	–	0.223966	–	1.578313
Fault 2	–	0.632128	0.550645	1.194915
Fault 3	–	–	–	0.553784
New fault	1.120218	–	–	1.137496

8.2 Fault Identification Based on LLE and EDA

The new dimensionality reduction approach based on the combination of EDA and LLE is proposed with two different combination performances, Local Linear Exponential Discriminant Analysis (LLEDA) and Neighborhood-Preserving Embedding Discriminant Analysis (NPEDA). This fusion idea combines the global discriminant analysis with local structure preservation during the dimensionality reduction process. LLEDA and NPEDA are solved by different optimization objectives, respectively, and the corresponding maximum values are derived to reduce the computational complexity. They both exhibit the good local preservation and global discrimination capabilities. The nonlinear analytics is transformed into an equivalent neighborhood holding problem based on the idea of piecewise linearization.

The main difference between the two methods is that LLEDA is a parallel strategy whereas NPEDA is a cascading strategy. LLEDA focuses on the global supervised discrimination balanced with local nonlinear dimensionality reduction. It finds a balanced projection vector between the local geometry and the data classification and results in an optimal subspace projection of the samples. When faults are difficult to distinguish, LLEDA method can improve the identification rate by adjusting the trade-off parameter between the global index and the local index. NPEDA is a

cascading strategy where the dimensionality reduction process is implemented in two successive steps: the first aims at maintaining the local geometric relationships and reconstructing each sample point using a linear weighted combination of nearest neighbors, the second at performing discriminant analysis on the reconstructed sample.

8.2.1 Local Linear Exponential Discriminant Analysis

The basic idea of LLEDA is to project the samples into the optimal discriminant space while maintaining the local geometric structure of the original data. The schematic diagram is shown in Fig. 8.8. LLEDA combines the advantages of LLE and EDA, which extracts the global classification information while compressing the dimensionality of the feature space without destroying local relationships. It finds a balance between global supervised discrimination and local preservation of nonlinearity through an adjusted trade-off parameter.

Consider the original data being mapped into a hidden space F via function A . An explicit linear mapping from X to Y , $Y = A^T X$ is constructed to circumvent the out-of-sample problem. The original LLE problem is written as follows:

$$\begin{aligned} \min \varepsilon(Y) &= \sum_{j=1}^n \left| y_j - \sum_{r=1}^k W_{jr} y_{jr} \right|^2 = \| Y(I - W) \|^2 \\ &= \text{tr}(Y(I - W)(I - W)^T Y^T) \\ &= \text{tr}(A^T X M X^T A). \end{aligned} \quad (8.5)$$

The LLEDA problem is proposed with the following objective function:

$$\max J(A) = \frac{\text{tr}(A^T \exp(S_b) A)}{\text{tr}(A^T \exp(S_w) A)} - \mu \cdot \text{tr}(A^T X M X^T A), \quad (8.6)$$

where μ is a trade-off parameter that balances the intrinsic geometry and global discriminant information. In general, (8.6) is equivalently transformed into an optimization problem with constraint,

$$\begin{aligned} \max J(A) &= \text{tr}(A^T \exp(S_b) A) - \mu \cdot \text{tr}(A^T X M X^T A) \\ \text{s.t. } &A^T \exp(S_w) A = I, \end{aligned} \quad (8.7)$$

where $A = [a_1, a_2, \dots, a_n]$. (8.7) is solved by introducing the Lagrangian multiplier:

$$L_1(a_i) = a_i^T (\exp(S_b) - \mu X M X^T) a_i + \theta(1 - a_i^T \exp(S_w) a_i), \quad (8.8)$$

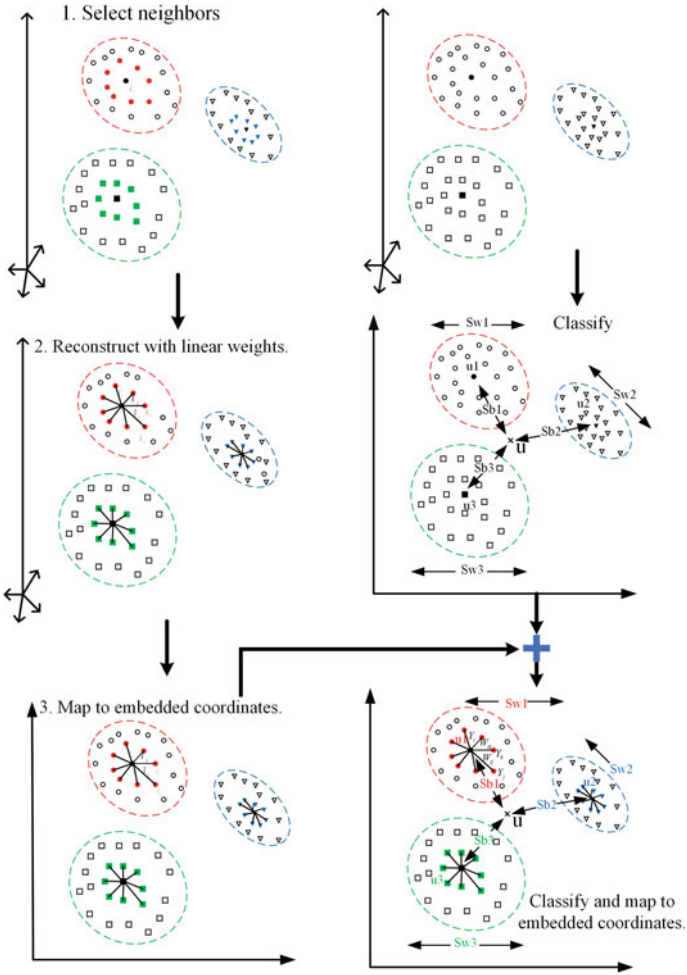


Fig. 8.8 The schematic diagram of LLEDA

where θ is Lagrangian multiplier. According to the zero gradient in $L_1(a_i)$ with respect to a_i , we have

$$\begin{aligned}
 &(\exp(S_b) - \mu X M X^T) a_i = \theta \exp(S_w) a_i \\
 &\text{or} \\
 &(\exp(S_w)^{-1} (\exp(S_b) - \mu X M X^T) a_i = \theta a_i,
 \end{aligned}
 \tag{8.9}$$

where θ is treated as a generalization eigenvalue. The discriminant matrix A is made up of the corresponding eigenvectors of the first d largest eigenvalues in (8.9).

8.2.2 Neighborhood-Preserving Embedding Discriminant Analysis

NPEDA is also to find a series of discriminative vectors and map the samples into a new space. The sample point is represented linearly by their neighbors to maintain the local geometry as much as possible during the projection process. The schematic diagram is shown in Fig. 8.9. NPEDA is a cascade strategy in which the dimensionality reduction process is divided into two successive steps, the first aiming at maintaining local geometric relationships and the second aiming at a discriminant analysis in which each sample point is reconstructed by a linearly weighted combination of its neighbors.

Rewrite the between-class scatter matrix S_b and the within-class scatter matrix S_w under the explicit linear mapping $Y = A^T X$:

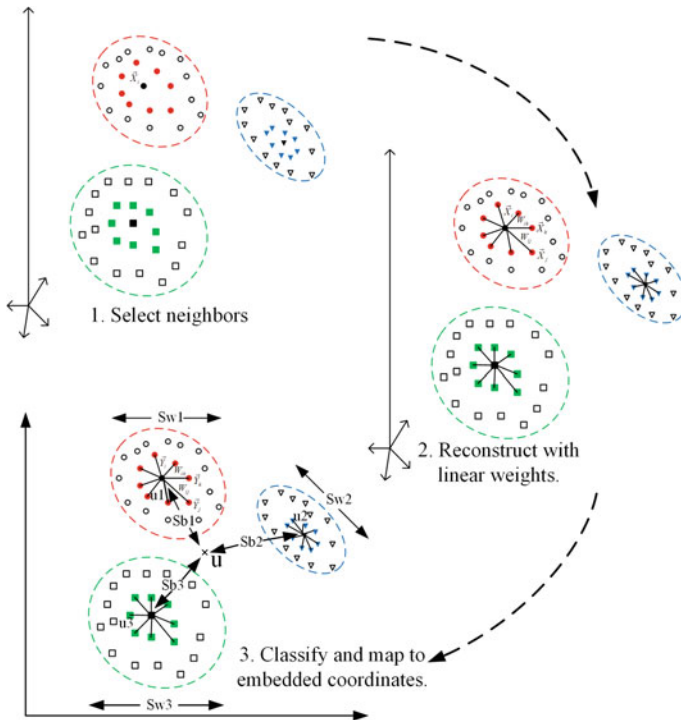


Fig. 8.9 The schematic diagram of NPEDA

$$\begin{aligned}
S_b &= \sum_{i=1}^c n_i (\bar{y}^i - \bar{y})^2 = \sum_{i=1}^c n_i (A^T \bar{x}^i - A^T \bar{x})^2 \\
&= A^T \left(\sum_{i=1}^c n_i (\bar{x}^i - \bar{x})(\bar{x}^i - \bar{x})^T \right) A \\
&= A^T \left(\sum_{i=1}^c \frac{1}{n_i} (\mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i) (\mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i)^T - 2n\bar{x}\bar{x}^T + n\bar{x}\bar{x}^T \right) A \\
&= A^T \left(\sum_{i=1}^c \sum_{j,k=1}^{n_i} \frac{1}{n_i} \mathbf{x}_j^i \mathbf{x}_k^{iT} - n_i \bar{x} \bar{x}^T \right) A \\
&= A^T (X B X^T - n\bar{x}\bar{x}^T) A \\
&= A^T X \left(B - \frac{1}{n} \mathbf{e} \mathbf{e}^T \right) X^T A,
\end{aligned} \tag{8.10}$$

where $\bar{x}^i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j^i$, $\bar{y} = \frac{\sum_{i=1}^c n_i \bar{y}^i}{\sum_{i=1}^c n_i}$, $\bar{x} = \frac{\sum_{i=1}^c n_i \bar{x}^i}{\sum_{i=1}^c n_i} = \frac{1}{n} \sum_{i=1}^c n_i \bar{x}^i$; $\mathbf{e} = [1, 1, \dots, 1]^T$ with dimension n , and

$$B_{ij} = \begin{cases} \frac{1}{n_k} & \mathbf{x}_i \text{ and } \mathbf{x}_j \in k\text{-th class.} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned}
S_w &= \sum_{i=1}^c \sum_{j=1}^{n_i} (y_j^i - \bar{y}^i)^2 = \sum_{i=1}^c \sum_{j=1}^{n_i} (A^T \mathbf{x}_j^i - A^T \bar{x}^i)^2 \\
&= A^T \left(\sum_{i=1}^c \left(\sum_{j=1}^{n_i} (\mathbf{x}_j^i - \bar{x}^i) (\mathbf{x}_j^i - \bar{x}^i)^T \right) \right) A \\
&= A^T \left(\sum_{i=1}^c \left(\sum_{j=1}^{n_i} \mathbf{x}_j^i \mathbf{x}_j^{iT} - n_i \bar{x}^i \bar{x}^{iT} \right) \right) A \\
&= A^T \left(\sum_{i=1}^c \left(X_i X_i^T - \frac{1}{n_i} X_i (\mathbf{e}_i \mathbf{e}_i^T) X_i^T \right) \right) A \\
&= A^T \sum_{i=1}^c (X_i L_i X_i^T) A,
\end{aligned} \tag{8.11}$$

where $L_i = I - \frac{1}{n_i} \mathbf{e}_i \mathbf{e}_i^T$, I is unit matrix, and $\mathbf{e}_i = [1, 1, \dots, 1]^T$ with dimension n_i .

The discriminant vectors A^* are solved by the following optimization problem:

$$A^* = \arg \max \frac{|A^T X (B - \frac{1}{n} \mathbf{e} \mathbf{e}^T) X^T A|}{|A^T \sum_{i=1}^c (X_i L_i X_i^T) A|}. \tag{8.12}$$

Considering that the original data is reconstructed by its neighbors less than ε :

$$\sum_{j=1}^n \|\mathbf{x}_j - \sum_{r=1}^k \mathbf{W}_{jr} \mathbf{x}_{jr}\|^2 < \varepsilon,$$

where ε is a small positive number. \mathbf{W} is reconstruction mapping matrix such that $\sum_{r=1}^k \mathbf{W}_{ir} = 1$. Then

$$\left\| \mathbf{x}_i - \sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir} \right\|^2 = \left\| \sum_{r=1}^k (\mathbf{W}_{ir} \mathbf{x}_i - \mathbf{W}_{ir} \mathbf{x}_{ir}) \right\|^2 = \|\mathbf{Q}_i \mathbf{W}_i\|^2,$$

where $\mathbf{Q}_i = [\mathbf{x}_i - \mathbf{x}_{i1}, \mathbf{x}_i - \mathbf{x}_{i2}, \dots, \mathbf{x}_i - \mathbf{x}_{ir}]$.

Matrix \mathbf{W} can be solved by Lagrange multiplier.

$$L_2 = \frac{1}{2} \|\mathbf{Q}_i \mathbf{W}_i\|^2 - \lambda_i \left[\sum_{r=1}^k \mathbf{W}_{ir} - 1 \right]$$

$$\frac{\partial L_2}{\partial \mathbf{W}_i} = \mathbf{Q}_i^T \mathbf{Q}_i \mathbf{W}_i - \lambda_i \mathbf{E} = \mathbf{C}_i \mathbf{W}_i - \lambda_i \mathbf{E} = 0,$$

where $\mathbf{W}_i = \lambda_i \mathbf{C}_i^{-1} \mathbf{E}$, $\mathbf{C}_i = \mathbf{Q}_i^T \mathbf{Q}_i$, $\mathbf{E} = [1, 1, \dots, 1]^T$ with dimension k .

Considering

$$\sum_{r=1}^k \mathbf{W}_{ir} = \mathbf{E}^T \mathbf{W}_i = 1 \implies \mathbf{E}^T \lambda_i \mathbf{C}_i^{-1} \mathbf{E} = 1 \implies \lambda_i = (\mathbf{E}^T \mathbf{C}_i^{-1} \mathbf{E})^{-1},$$

we have

$$\mathbf{W}_i = \lambda_i \mathbf{C}_i^{-1} \mathbf{E} = \frac{\mathbf{C}_i^{-1} \mathbf{E}}{\mathbf{E}^T \mathbf{C}_i^{-1} \mathbf{E}}.$$

The sample point is reconstructed by the optimal weights \mathbf{W} , i.e., $\mathbf{x}_j = \sum_{r=1}^k \mathbf{W}_{jr} \mathbf{x}_{jr}$. It is linearly represented by its neighbors by maintaining the local geometry in the dimensionality reduction process. Substitute it into (8.12) and NPEDA optimization is revised as follows:

$$A^* = \arg \max_A \frac{\left| \mathbf{A}^T \exp \left((\sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir}) (\mathbf{B} - \frac{1}{n} \mathbf{e} \mathbf{e}^T) (\sum_{r=1}^k \mathbf{W}_{ir} \mathbf{x}_{ir})^T \right) \mathbf{A} \right|}{\left| \mathbf{A}^T \exp \left(\sum_{i=1}^c (\sum_{r=1}^k \mathbf{W}_{jr} \mathbf{X}_{jr}^i) \mathbf{L}_i (\sum_{r=1}^k \mathbf{W}_{jr} \mathbf{X}_{jr}^i)^T \right) \mathbf{A} \right|} \quad (8.13)$$

$$= \arg \max_A \frac{\left| \mathbf{A}^T \exp(\mathbf{S}_{nb}) \mathbf{A} \right|}{\left| \mathbf{A}^T \exp(\mathbf{S}_{nw}) \mathbf{A} \right|}.$$

Equation (8.13) is equivalently to solve the maximum eigenvalue of the generalized eigenvalue decomposition problem:

$$\begin{aligned} \exp(\mathbf{S}_{nb})\mathbf{A} &= \sigma \exp(\mathbf{S}_{nw})\mathbf{A} \\ \text{or} \\ \exp(\mathbf{S}_{nw})^{-1} \exp(\mathbf{S}_{nb})\mathbf{A} &= \sigma \mathbf{A}, \end{aligned} \quad (8.14)$$

where σ is the generalized eigenvalue and the linear transformation matrix \mathbf{A} of NPEDA is the eigenvector corresponding to the first d largest eigenvalues of $(\exp(\mathbf{S}_{nw}))^{-1} \exp(\mathbf{S}_{nb})$.

8.2.3 Fault Identification Based on LLEDA and NPEDA

In this section, the LLEDA and NPEDA methods are implemented for fault identification with monitoring flowchart, as shown in Fig. 8.10. The fault recognition rate (FCR) is introduced to test the identification effectiveness. FCR of fault model i is defined as the percentage of test data identified in this corresponding model out of the total number of samples tested:

$$\text{FCR}(i) = \frac{n_{i,identify}}{n_{all}} \times 100\%, \quad (8.15)$$

where $n_{i,identify}$ denotes the sample size identified as fault i and n_{all} denotes the sample size of all samples of fault i . The identification process is given as follows,

1. Process data are collected under the normal and faulty conditions, and standardized.
2. The between-class scatter matrix \mathbf{S}_b and the within-class scatter matrix \mathbf{S}_w are calculated by the LLEDA (or NPEDA) method, respectively.
3. The discriminant vector \mathbf{A} is obtained by maximizing the between class dispersion matrix \mathbf{S}_b and minimizing the with class dispersion matrix \mathbf{S}_w .
4. The discriminant function $g(x)$ of the online data x is observed by the projection of discriminant vector \mathbf{A} in the normal model:

$$\begin{aligned} g(x) &= -\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}^i)^T \mathbf{A} \left(\frac{1}{n_i - 1} \mathbf{A}^T \exp(\mathbf{S}_w^i) \mathbf{A} \right)^{-1} \mathbf{A}^T (\mathbf{x} - \bar{\mathbf{x}}^i) \\ &\quad + \ln(c) - \frac{1}{2} \ln \left[\det \left(\frac{1}{n_i - 1} \mathbf{A}^T \exp(\mathbf{S}_w^i) \mathbf{A} \right) \right]. \end{aligned} \quad (8.16)$$

If the value of the discriminant function exceeds the normal limitation, a fault occurs.

5. The fault type of online data can be determined when its posterior probability value is maximum. The posterior probability of data \mathbf{x} in fault c_i class is calculated as

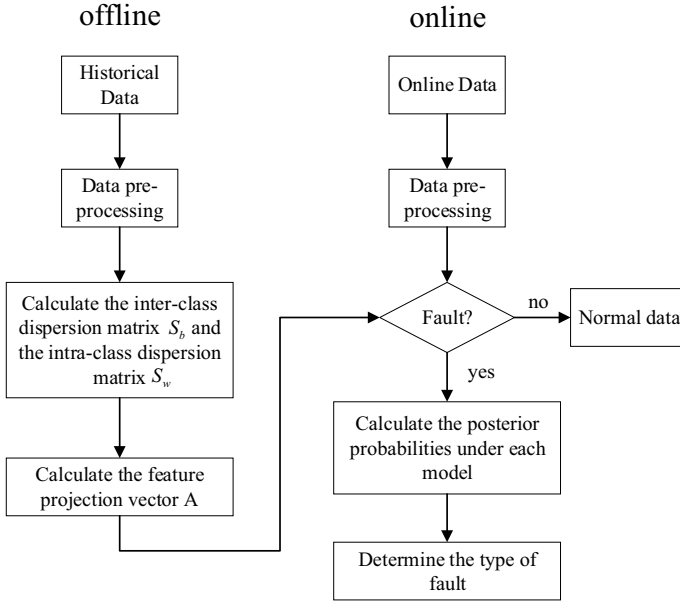


Fig. 8.10 Flowchart of fault identification with LLEDA and NPEDA methods

$$P(\mathbf{x} \in \mathbf{c}_i | \mathbf{x}) = \frac{P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) P(\mathbf{x} \in \mathbf{c}_i)}{\sum_{i=1}^c P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) P(\mathbf{x} \in \mathbf{c}_i)}, \quad (8.17)$$

where $P(\mathbf{x} \in \mathbf{c}_i)$ is the prior probability and $P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i)$ is the conditional probability density function of the sample \mathbf{x} :

$$P(\mathbf{x} | \mathbf{x} \in \mathbf{c}_i) = \frac{\exp[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}^i)^T \mathbf{A} \mathbf{P}_q \mathbf{A}^T (\mathbf{x} - \bar{\mathbf{x}}^i)]}{(2\pi)^{\frac{m}{2}} [\frac{1}{n_i - 1} \mathbf{A}^T (\sum_{\mathbf{x} \in \mathbf{c}_i} (\mathbf{x} - \bar{\mathbf{x}}^i)(\mathbf{x} - \bar{\mathbf{x}}^i)^T) \mathbf{A}]^{\frac{1}{2}}}, \quad (8.18)$$

where $\mathbf{P}_q = [\frac{1}{n_i - 1} \mathbf{A}^T (\sum_{\mathbf{x} \in \mathbf{c}_i} (\mathbf{x} - \bar{\mathbf{x}}^i)(\mathbf{x} - \bar{\mathbf{x}}^i)^T) \mathbf{A}]^{-1}$.

8.2.4 Simulation Experiment

Multi-classification methods, FDA, EDA, LLE+FDA, LLEDA, and NPEDA, were carried to evaluate the classification performance in TE simulation platform. TE operation lasted for 48h, with faults occurring in the 8thh and sampled every 3 min. 400 training data were selected for building the classification model and 400 testing data for evaluating the performance of the model. Three different types of faults were considered: faults 2, 8, and 13. Fault 2 refers to a step change in the B component feed

with the A/C feed ratio remaining constant. Fault 8 refers to a random change in the A, B and C feed component variables. Fault 13 refers to a slow drift in the reaction dynamics. Here faults 8 and 13 are difficult identified due to its random variation and slow drift. The training and testing data for the three types of faults were projected onto the first and second eigenvectors, respectively, by different methods and the classification results are shown in Fig. 8.11.

Table 8.4 shows the identification rate for faults 2, 8, and 13 under different classification methods. Here the number of discrimination directions, i.e., reduction order, is considered from 1 to 10. It is shown that the identification rates are improved with increasing the number of discrimination vectors. The recognition rate for fault 2 is high, almost close to 100%. The recognition rate for faults 8 and 13 gradually increases as the number of discrimination vectors increases. NPEDA and LLEDA show higher recognition rates on faults 2, 8, and 13, compared with other methods, such as FDA and LLE+EDA.

Figure 8.12 shows the posterior probability values for the different test data under the LLEDA and NPEDA methods. The larger posteriori probability values mean the higher possibility of the test data belong to this category. Furthermore, the diagnostic results are related to the classification capability. If the classification performance is good, higher identification rate is achieved.

8.3 Cluster-LLEDA-Based Hybrid Fault Monitoring

8.3.1 Hybrid Monitoring Strategy

Generally, the data collected from an actual industrial process are unlabeled and initially undiagnosed. It is worth noting that the LLEDA method performs well in fault identification, but it is a supervised algorithm that requires the known classification of the historical data set. To overcome this problem, the supervised LLEDA method is extended into an unsupervised learning method by introducing the cluster analysis method. The cluster method can obtain the fault data category information which is input to LLEDA modeling module as a prior. To make better use of the proposed cluster-LLEDA classification method, a hybrid fault monitoring strategy is given, as shown in Fig. 8.13.

Figure 8.13 indicates that the hybrid fault monitoring strategy is mainly divided into three parts, **historical data analysis**, **fault model library establishment**, and **online detection and fault identification**. First, the historical data of industrial processes is roughly detected by PCA to label the fault data. Then hierarchical clustering technique is used to classify the process data detected as fault into different types. The model library is established for all fault types by LLEDA, which further extracts the fault features and obtain fine identification. Finally, the online detection and fault identification are realized.

The procedure of historical data analysis part is summarized as follows:

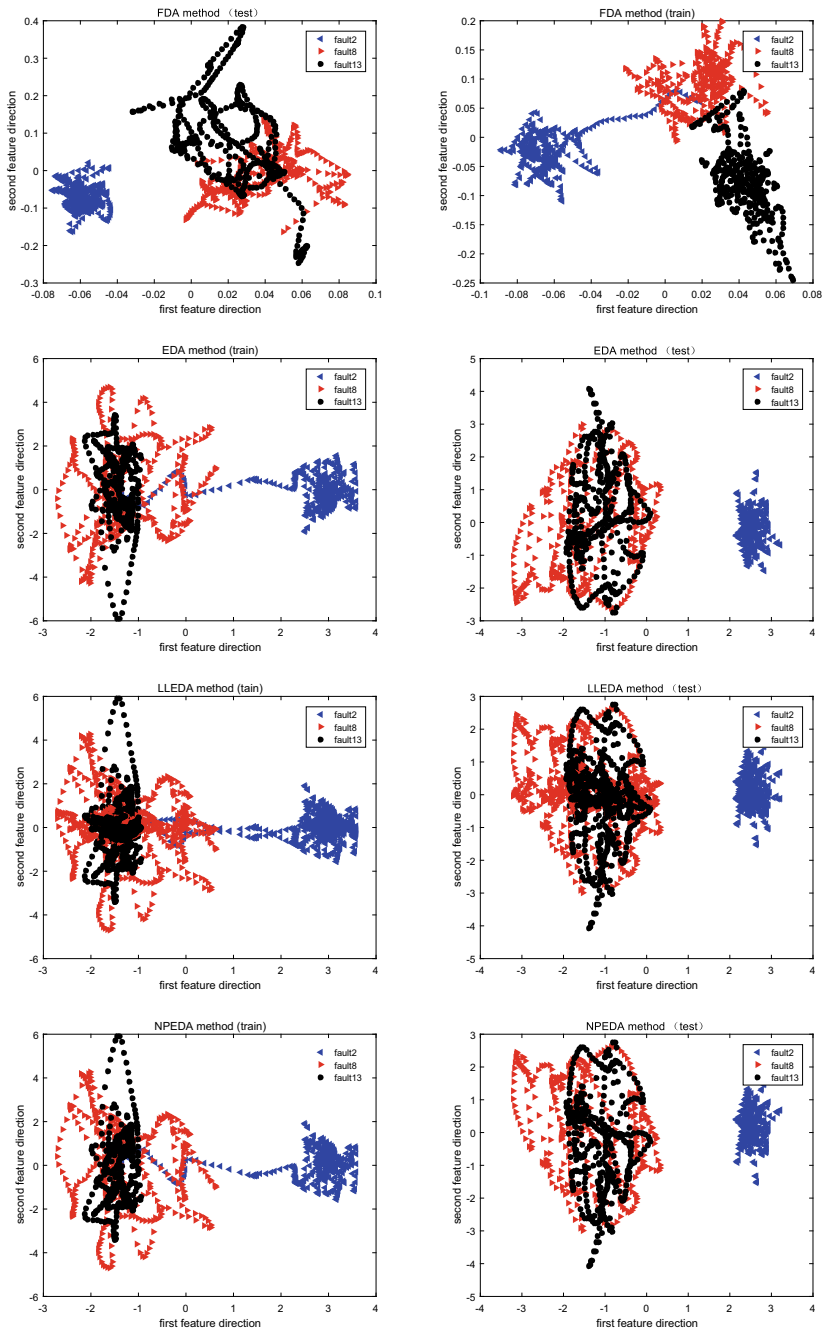


Fig. 8.11 Projection of different fault data on the first two feature vectors

Table 8.4 Comparison of identification rate for faults 2, 8, and 13

Reduction order	Fault No.	FDA	EDA	LLE+FDA	LLEDA	NPEDA
1	Fault 2	1	1	1	1	1
	Fault 8	0.4425	0.2125	0.4625	0.2125	0.2125
	Fault 13	0.415	0.6875	0.4175	0.6875	0.6875
2	Fault 2	1	1	1	1	1
	Fault 8	0.3525	0.475	0.48	0.4175	0.475
	Fault 13	0.36	0.6325	0.3475	0.6875	0.6325
3	Fault 2	1	1	1	1	1
	Fault 8	0.4375	0.67	0.3825	0.5975	0.67
	Fault 13	0.29	0.55	0.3375	0.6275	0.55
4	Fault 2	1	1	1	0.9925	1
	Fault 8	0.47	0.8325	0.425	0.705	0.8325
	Fault 13	0.2825	0.6575	0.295	0.565	0.6575
5	Fault 2	1	1	995	1	1
	Fault 8	0.625	0.8825	0.4875	0.815	0.8825
	Fault 13	0.53	0.6375	0.3025	0.5975	0.6325
6	Fault 2	1	1	1	1	1
	Fault 8	0.664	0.9325	0.62	0.895	0.9325
	Fault 13	0.5125	0.7225	0.25	0.6225	0.7225
7	Fault 2	1	1	9925	1	1
	Fault 8	0.695	0.8925	0.6	0.9125	0.8925
	Fault 13	0.49	0.7425	0.2425	0.725	0.7425
8	Fault 2	1	1	9825	1	1
	Fault 8	0.7275	0.88	0.7075	0.885	0.88
	Fault 13	0.4775	0.74	0.2275	0.7125	0.74
9	Fault 2	1	1	0.99	1	1
	Fault 8	0.745	0.88	0.6575	0.89	0.88
	Fault 13	0.49	1	0.995	1	1
10	Fault 2	0.99	1	0.995	1	1
	Fault 8	0.7625	0.8725	0.5825	0.8825	0.8725
	Fault 13	0.47	0.735	0.225	0.7125	0.735

1. Collect and standardize the normal process data from the DCS historical database.
2. Analyze the collected process data by PCA to extract the independent principle components, establish PCA model of the normal operation, and calculate the statistics of the data.
3. Calculate the statistics T^2 and SPE and their control limit.

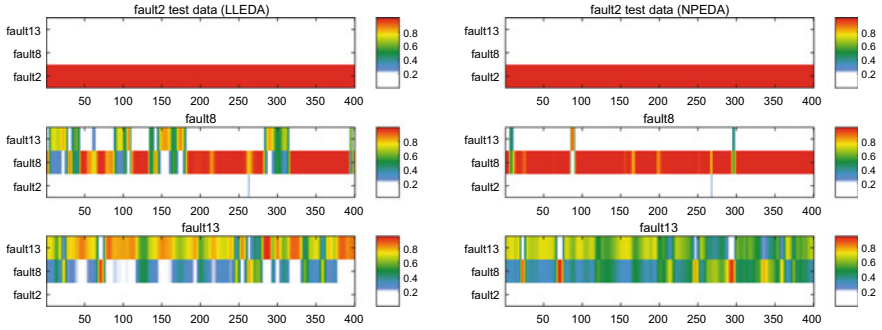


Fig. 8.12 Diagnosis results of faults 2, 8, and 13 by LLEDA and NPEDA methods

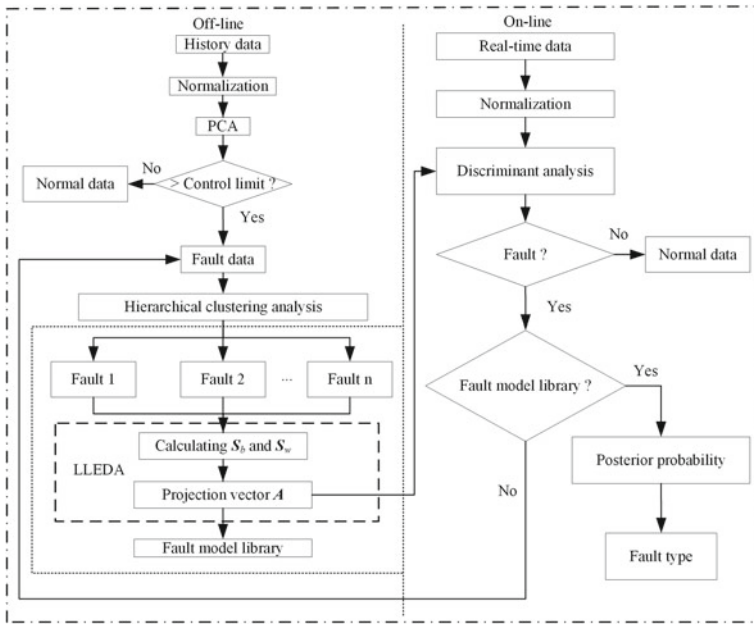


Fig. 8.13 Hybrid fault detection and diagnosis information process

The procedure of fault model library establishment is summarized as follows:

1. Perform hierarchical clustering analysis on the abnormal operation data and divide them into different fault categories.
2. Calculate the between-class and within-class scatter matrices S_b and S_w , find the corresponding projection vector A based on LLEDA method, and establish the fault model library for all fault classes.

The procedure of online detection and fault identification is summarized as follows:

1. Sample the real-time data and standardize it.
2. Perform the discriminant analysis based on LLEDA method, project the sample data to the projection direction, and extract the feature vector.
3. Project the sample data to the projection vector A based on the normal model and judge the current operation is normal or abnormal by observing whether the discriminant function exceeds the limit.
4. If a fault occurs, calculate the posterior probability in each fault model to identify the fault type. If the sample data is not in the existing fault category, this new fault will be modeled and introduced into the fault model library.

Clustering Analysis The hierarchical clustering algorithm is more widely used and has the advantages of simple calculation, fast and easy to obtain similar results, without knowing the number of clusters in advance (Saxena et al. 2017). The clustering starts with n samples each as a class, specifies the distance between samples and the clustering between classes. Then the two closest classes are merged into a new class, and the distance between the new class and the other classes are calculated. Repeat the merging process between the two closest classes, and the number of classes is reduced by one after each merging. The merging will stop until all samples are merged into one class or a certain condition is met.

The class is denoted by G in the cluster analysis. Suppose class G has m samples denoted by the column vector $x_i (i = 1, 2, \dots, m)$, d_{ij} is the distance between x_i and x_j , and D_{KL} is the distance between two different categories G_K and G_L . The squared distance D_{KL}^2 between G_K and G_L is defined as follows:

$$D_{KL}^2 = \frac{1}{n_K n_L} \sum_{x_i \in G_K, x_j \in G_L} d_{ij}^2. \quad (8.19)$$

The recursive formula for between-class squared clustering is

$$D_{ML}^2 = \frac{n_K}{n_M} D_{KJ}^2 + \frac{n_K}{n_M} D_{LJ}^2. \quad (8.20)$$

The inconsistency coefficient Y is used to determine the final number of clusters c . Here Y is a matrix of $(n - 1) \times 4$, where the first column is the mean of all link lengths (i.e., merging class distances) involved, the second column is the standard deviation of all the related link lengths, the third column is the number of related links, and the fourth column is the inconsistency coefficient.

For the links obtained by the k th merging class, the inconsistency coefficient is calculated as follows:

$$Y(k, 4) = \frac{(Z(k, 3) - Y(k, 1))}{(Y(k, 2))}, \quad (8.21)$$

where the input $Z_{(n-1) \times 3}$ is a matrix of systematic clustering trees. Under the condition that guarantees the number of classes as small as possible, the change of the inconsistency coefficient determines the final value of classes number.

8.3.2 Simulation Study

The experiment uses the Tennessee Eastman (TE) process to evaluate the effectiveness of the proposed hybrid method.

Experiment 1: Failure Initial Screening and Classification The TE data set was first detected by the PCA method, and the fault detection results are shown in Fig. 8.14, the final T^2 and SPE statistics obtained were 0.4951 and 0.6882, respectively. The specific detection is shown in Table 8.5. The results show that the recognition rate of faults 1, 2, 6, 7, 8, 12, 13, 14, 17, and 18 is high, and the recognition rate of other faults is low. This indicates that the significant faults can be detected, while the potential faults cannot be detected.

Therefore, PCA-based fault detection methods can only coarsely split the data set and detect significant faults. Potential faults can be identified with a high fault identification rate only in the case of known fault categories. In the coarse separation stage of historical data, the fault data can be identified not only by PCA method, but also by improved PCA or other fault detection methods to further improve the identification rate.

After the historical data analysis, the fault data set is collected and clustered into different fault classes by using the hierarchical clustering method. According to the inconsistency coefficient, the final number of fault classes is 10. As the fault type is in a large number, it is difficult to display the classified fault data together in a tree diagram. As example, we select the faults 1, 2, and 6 to demonstrate the clustering effect of the hierarchical cluster analysis algorithm. Fault 1 is a step change in the A/C feed ratio with component B remaining unchanged, while fault 2 is a step change

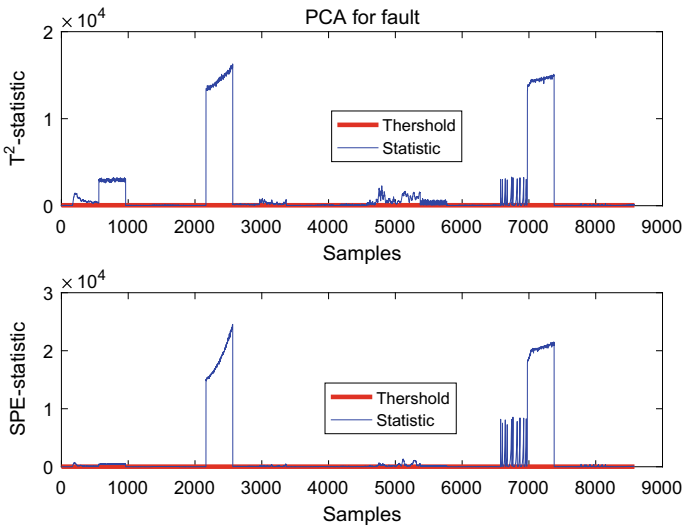


Fig. 8.14 Fault detection based on PCA

Table 8.5 Fault recognition rate based on PCA

Fault No.	T ²	SPE	Fault No.	T ²	
Fault 1	0.995	0.9988	Fault 12	0.9875	0.99
Fault 2	0.9825	0.9925	Fault 23	0.9513	0.9625
Fault 3	0.0225	0.2675	Fault 14	0.9988	1
Fault 4	0.41	1	Fault 15	0.0488	0.2625
Fault 5	0.2625	0.5025	Fault 16	0.2325	0.6937
Fault 6	0.99	1	Fault 17	0.8013	0.975
Fault 7	1	1	Fault 18	0.8912	0.9375
Fault 8	0.975	0.9825	Fault 19	0.0675	0.5913
Fault 9	0.0362	0.235	Fault 20	0.3738	0.735
Fault 10	0.4163	0.7638	Fault 21	0.3775	0.6687
Fault 11	0.5212	0.8163			

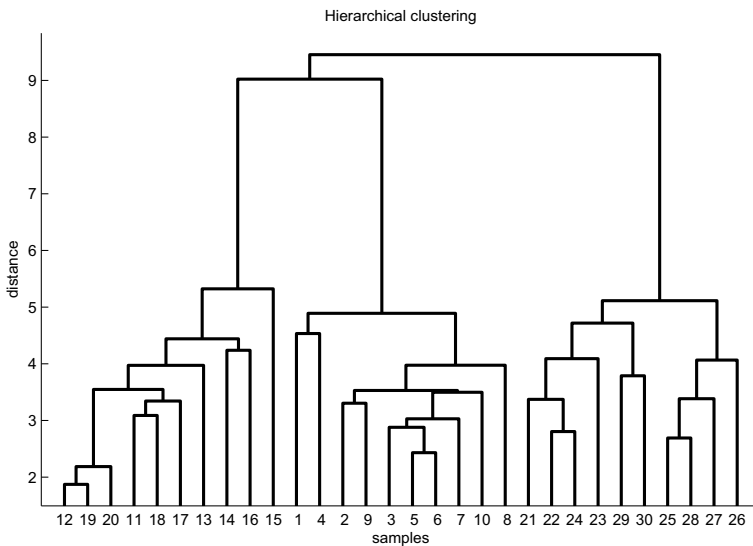


Fig. 8.15 Hierarchical cluster analysis

in component B with the A/C ratio remaining unchanged. Fault 6 is a step change in the feed loss of A. The hierarchical clustering tree diagram is given Fig. 8.15. The final number of categories is three according to the inconsistency coefficient, which is consistent with the actual classification.

Now the fault data have been divided into 10 classes by hierarchical cluster analysis. Obviously, the dimension is high and its visualization effect is poor. In order to improve the visualization effect and reflect the change trend and the interrelationship between each variable at the same time, the parallel coordinate visualization method

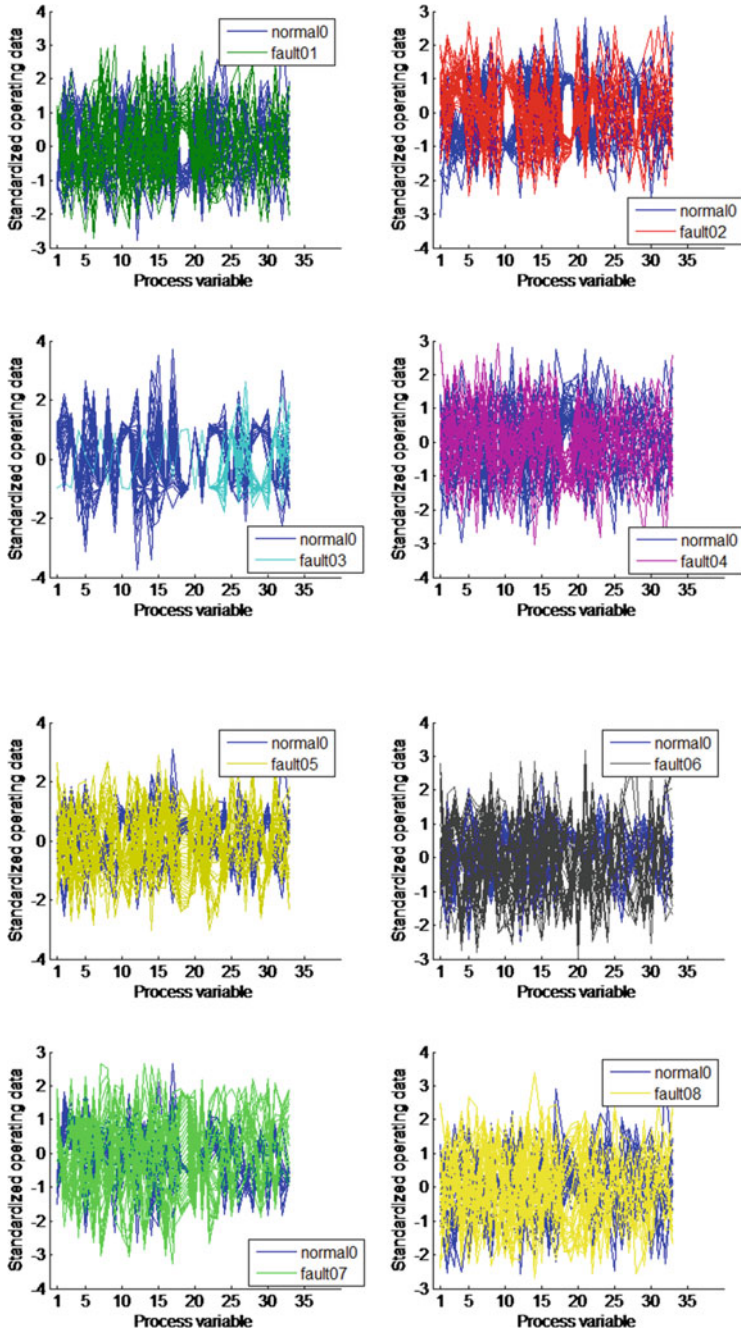


Fig. 8.16 Parallel coordinate visualization of fault data

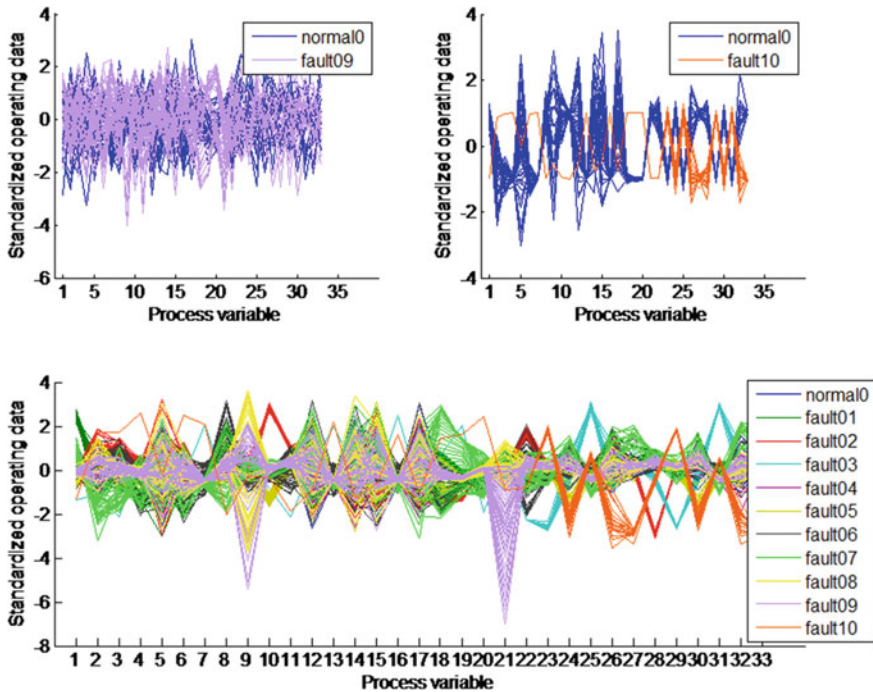


Fig. 8.16 (continued)

is selected. It is a visualization technique that allows the high-dimensional variables to be represented by a series of axes parallel to each other. The value of the variables is corresponding to the positions on the axes.

The visualization results for each type of fault data are shown in Fig. 8.16. The blue dash in each subplot indicates the normal data and the other color dashes indicate different fault data. Since each variable in the TE data has a corresponding actual physical meaning, the type of fault can be judged by comparing the other color dashes with the blue dash in each variable. These faults can be labeled for establishing the fault model library.

Experiment 2: LLEDA-based Fault Identification The fault identification method used here is LLEDA, which increases the distance between different classes and improves the classification ability even if fault samples are small. Here faults 4, 8, and 13 are selected as example to show the identification results. Fault 4 is a minor fault, which is manifested in the step change of the inlet temperature of the reactor cooling water, but the other 50 variables are still in a stable state, and the change is less than 2% compared with the normal data. Fault 13 refers to the slow drift of reactor kinetic constants when the fault occurs, which will cause a violent reaction of each variable, and the final product G is always in a fluctuating state. Fault 8 refers

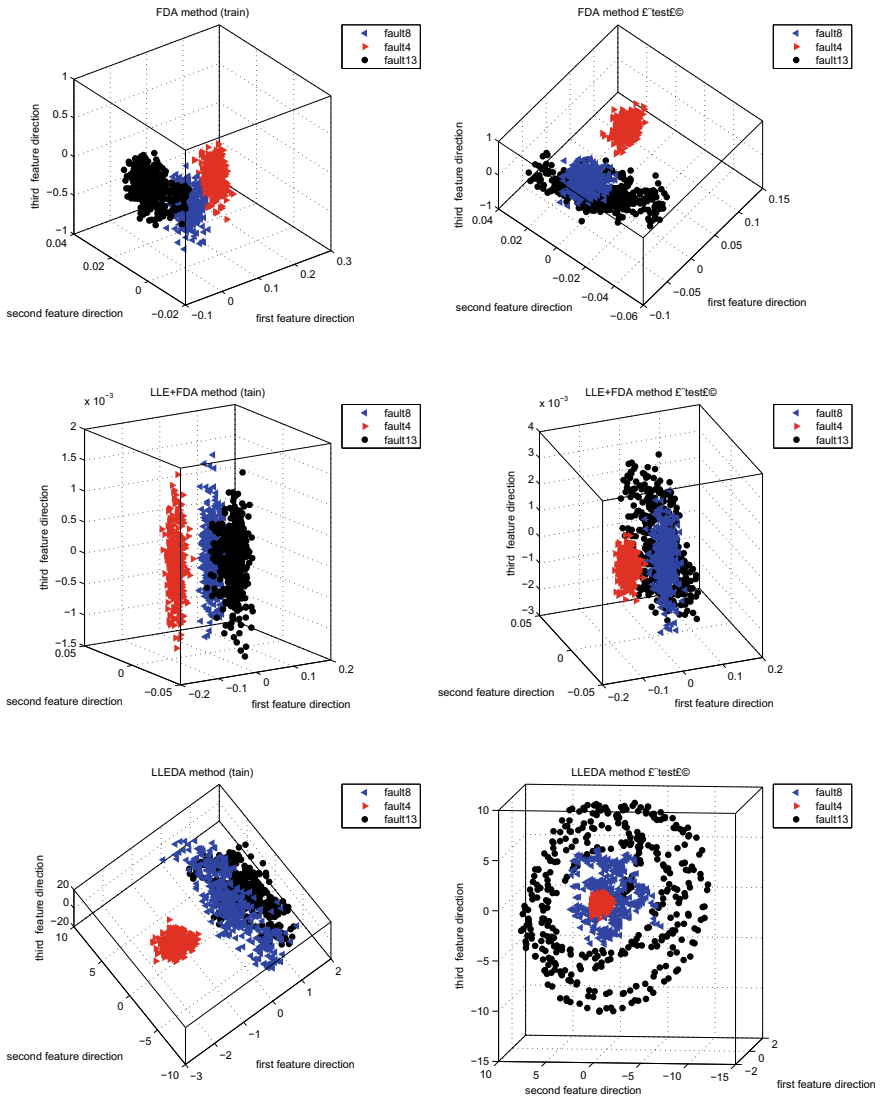


Fig. 8.17 Projection of different fault data on feature vectors

to the change of random variables of A, B, and C feed ingredients when the fault occurs.

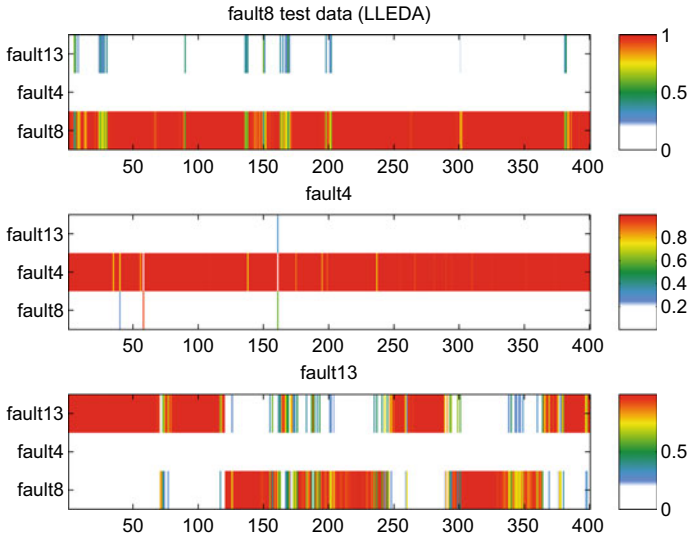


Fig. 8.18 Diagnosis results of fault 4, 8, and 13 by LLEDA methods

To better observe the classification in spatial structure, the training data and testing data of the three faults are projected onto the first three feature vectors by different methods. The classification results are shown in Fig. 8.17.

Figure 8.18 shows the posterior probability values of different test data by LLEDA method under different models. The posterior probability values are larger when the samples belong to category i . The colored bars indicate the diagnostic result, i.e., probability values, in which color bar from bottom to top is corresponding to the probability values 0–1 (white indicates that the probability of identification is 0 and red indicates that the probability value of identification is 1.) In this way, the fault identification results are visualized. The diagnosis result is related to the classification ability. The better classification performance leads to a higher fault recognition rate. Here fault 13 is in poor classification owing to the small number of feature vectors. The recognition rate of faults can be improved by increasing the number of feature vectors.

8.4 Conclusion

This chapter presents three discriminant analysis methods, KEDA, LLEDA and NPEDA, that can handle nonlinearities and avoid small sample data problems. Nor-

mal and faulty data models are developed, and these models are used to check whether abnormal behavior occurs, and variance-based performance metrics are used to identify the type of data tested. Especially, two new supervised dimensionality reduction methods, LLEDA and NPEDA, are proposed which combines the advantages of local linear embedding and exponential discriminant analysis methods, taking into account both global and local information. The nonlinear data is piecewise linearized by maintaining the internal structure during the extraction of the eigenvalues. They overcome the singularity problem of within-class scatter matrices, and therefore show good performance for the small sample problem.

Furthermore, the hybrid process monitoring and fault identification algorithm is proposed in this chapter, which effectively combines the PCA initial detection, the classification of hierarchical clustering, and the discriminative analysis of LLEDA. This hybrid method ensures the monitoring and diagnosis is performed directly on the collected data without a priori knowledge.

Reference

Saxena A, Prasad M, Gupta A, Bharill N, Patel O, Tiwari A, Er M, Ding W, Lin C (2017) A review of clustering techniques and developments. *Neurocomputing* 267:664–681

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 9

Global Plus Local Projection to Latent Structures



Owing to the raised demands on process operation and product quality, the modern industrial process becomes more complicated when accompanied by the large number of process and quality variables produced. Therefore, quality-related fault detection and diagnosis are extremely necessary for complex industrial processes. Data-driven statistical process monitoring plays an important role in this topic for digging out the useful information from these highly correlated process and quality variables, because the quality variables are measured at a much lower frequency and usually have a significant time delay (Ding 2014; Aumi et al. 2013; Peng et al. 2015; Zhang et al. 2016; Yin et al. 2014). Monitoring the process variables related to the quality variables is significant for finding potential harm that may lead to system shutdown with possible enormous economic loss.

PLS is a typical multivariate statistical analysis technique in two coordinate space, which is well suitable for the quality-related fault detection and process monitoring. However, actual industrial data are often with the features of strong nonlinear dynamic and coupled, etc. PLS method only considers the static linear mapping between multiple sources of data, so it is difficult to achieve accurate detection results by directly applying PLS. It becomes an important direction how to introduce the local structure-preserving capability to the global structure projection of PLS, in order to extract the complex features of industrial data. This idea of global structure and local structure fusion can usually be implemented by two strategies, plus and embedding. This chapter focuses on the idea of plus, global, and local partial least squares (GLPLS) which is introduced first. Global plus local projection to latent structure (GPLPLS) method is further proposed, and three different performance functions are given from the projection requirements of input measurement space and output measurement space, separately or simultaneously. The next two chapters focus on the idea of embedding, two different embedding methods, locality-preserving partial least squares (LPPLS) and local linear embedded projec-

tion of latent structure (LLEPLS), are proposed, which use LPP and LLE as local structure-preserving technique, respectively.

9.1 Fusion Motivation of Global Structure and Local Structure

Currently, partial least squares (PLS), which is one of those data-driven methods (Severson et al. 2016; Ge et al. 2012; Li et al. 2010; Zhao 2014; Zhang and Qin 2008), is widely used because of its advantages in extracting the latent variables by establishing the relationship between input and output space for quality-relevant process monitoring (Qin 2010). It maintains the maximum correlation between quality and process variables and has better quality-related fault detection capability. However, the nature of PLS is a linear projection, which is not applicable for nonlinear systems. It uses only global structural information with information such as mean and variance and performs poorly in systems with strong local nonlinear characteristics.

Nonlinear PLS methods can be divided into two categories: external nonlinear PLS models and internal nonlinear PLS models, as shown in Fig. 9.1.

External nonlinear PLS models are used as a class of nonlinear PLS models that introduce nonlinear transformations in the input and/or output variables. An example is kernel partial least squares (KPLS) (Rosipal and Trejo 2001; Godoy et al. 2014; Rosipal and Trejo 2001), which is used to describe the nonlinear relationship between the independent variables and for extending the linear relationship between the inputs and outputs. KPLS effectively solves the nonlinear problem between the principal components for input space and output space, but the selection of kernel function is more difficult in practical applications. Similarly, the kernel concurrent canonical correlation analysis (KCCCA) algorithm is proposed for quality-relevant nonlinear process monitoring that considers the nonlinearity in the quality variables (Zhu et al. 2017). Kernel-based methods map the original data into a (possibly

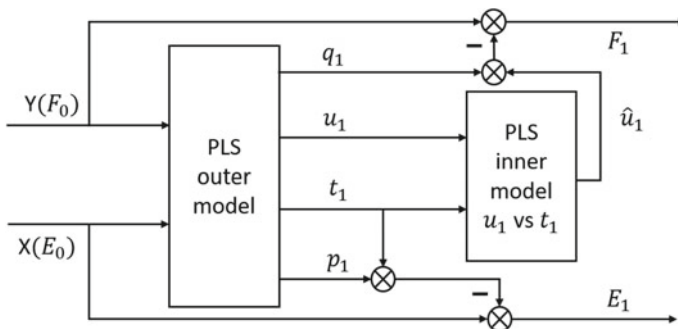


Fig. 9.1 Outer and inner model presentation for linear PLS decomposition

high-dimensional) Hilbert space (eigenspace), but the projection in the eigenspace is complex, the direction and length of the projection cannot be determined, and the choice of kernel function is not straightforward.

Inner nonlinear PLS model is where the internal linear model between latent variables is replaced by a nonlinear model, but its external model remains unchanged, such as quadratic partial least squares (QPLS) (Wold et al. 1989), spline function PLS (SPLS) (Wold 1992), and neural network PLS (NNPLS) (Qin and McAvoyn 1992, 1996) approaches. Recursive nonlinear PLS (RNPLS) models are built by extending the input and output matrices on top of PLS (Li et al. 2005); nonlinear PLS (NPLSSLT) based on the slice transformation (SLT) can be used for nonlinear correction, where SLT-based segmented linear mapping functions are used to construct nonlinear relationships between input and output score vectors (Shan et al. 2015); and nonlinear iterative partial least square algorithm (NIPALS) is improved by assuming that the score vector is a linear projection of the original variables in the internal nonlinear PLS, at the cost of increased computational complexity and optimization complexity.

PLS methods have nonlinearities in both the outer model and the inner model. An example is the orthogonal nonlinear PLS method (O-NLPLS) which considers orthogonal correlated nonlinearities between the input and output variables (Doymaz et al. 2003). This method retains the orthogonality properties of the PCA method due to the fact that it is based on a neural network architecture. Similarly, RBF network is used to identify the nonlinearity of the input variables and to establish the nonlinear relationship between the input and output variables (Zhao et al. 2006; Shimizu et al. 2006).

The different linear PLS representations are mathematically equivalent. However, using different nonlinear PLS methods results in different performance and characteristics. Existing nonlinear PLS methods have some shortcomings, such as the problem of choosing kernel functions or latent structures for unknown nonlinear systems; the problem of increasing computational complexity when using neural networks for nonlinear mapping; and the lack of a superior PLS decomposition algorithm. Therefore, how to simplify the nonlinear PLS modeling problem is an urgent need to be solved.

Considering that PLS and its extended algorithms only focus attention on the global structural information and cannot extract the local adjacent structural information of the data well, they are not suitable for the extraction of nonlinear features. Therefore, the local linearization method for dealing with nonlinear problems is taken into account. In recent years, locality-preserving projections (LPP) (He and Niyogi 2003; He et al. 2005), which belong to the manifold learning method have been proposed to solve the local adjacent structural feature problem and effectively make up for this deficiency. In addition, there are many other manifold learning methods, such as isometric feature mapping (Tenenbaum et al. 2000), local linear embedding (LLE) (Roweis and Saul 2000), Laplace feature map (Belkin and Niyogi 2003), etc.

Manifold learning methods preserve the local features by projecting the global structure to an approximate linear space, and by constructing a neighborhood graph to explore the inherent geometric features and manifold structure from the sample data sets. But these methods cannot consider the overall structure and lack a detailed

analysis and explanation of the correlation between process and quality variables. Therefore, combining the global projection methods, such as PLS, and the manifold learning method, such as LPP and LLE, has become a new topic of concern for a growing number of engineers.

Regarding the combination of global and local information, Zhong et al. proposed a quality-related global and local partial least squares (GLPLS) model (Zhong et al. 2016). The GLPLS method integrates the advantages of the LPP and PLS methods, and extracts meaningful low-dimensional representations from the high-dimensional process and quality data. The principal components in GLPLS preserve the local structural information in their respective data sheets as much as possible. However, the correlation between the process and quality variables is not enhanced, and the constraints of LPP are removed in the optimization objective function. Therefore, the monitoring results are seriously affected.

After further analysis of the geometric characteristics of LPP and PLS, a new integration method called the locality-preserving partial least squares (LPPLS) model that was proposed by Wang et al. pays more attention to the locality-preserving characteristics (Wang et al. 2017). LPPLS can exploit the underlying geometrical structure, which contains the local characteristics, in input and output space. Although the maximization of correlation degree between the process and quality variables was considered, the global characteristics were converted into a combination of multiple local linearized characteristics and were not expressed directly. In many processes, the linear relationship may be the most important, and the best way is to describe it directly rather than through a combination of multiple local linearized characteristics.

9.2 Mathematical Description of Dimensionality Reduction

9.2.1 PLS Optimization Objective

PLS algorithm is used to model the relationship between the normalized data sets $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)] \in R^{n \times m}$ ($\mathbf{x} = [x_1, x_2, \dots, x_m]^T$) and $\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)]^T \in R^{n \times l}$ ($\mathbf{y} = [y_1, y_2, \dots, y_l]$). \mathbf{X} is the process variable and \mathbf{Y} is the quality variable. m and l are the dimensionality of the input and output spaces, and n is the number of samples. \mathbf{X} and \mathbf{Y} are decomposed as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \bar{\mathbf{X}} \quad (9.1)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \bar{\mathbf{Y}}, \quad (9.2)$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_d] \in R^{n \times d}$, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d] \in R^{n \times d}$ are the score matrices of \mathbf{X} and \mathbf{Y} , respectively. $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_d] \in R^{m \times d}$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_d] \in R^{l \times d}$ are the load matrices of \mathbf{X} and \mathbf{Y} . $\bar{\mathbf{X}} \in R^{n \times m}$ and $\bar{\mathbf{Y}} \in R^{n \times l}$ are the residual matrices of \mathbf{X} and \mathbf{Y} . d is the number of latent variables. The weight vectors \mathbf{w} and \mathbf{c} are derived by the NIPALS algorithm such that the covariance of

score vectors \mathbf{t} and \mathbf{u} is maximized.

$$\begin{aligned}\max \text{cov}(\mathbf{t}, \mathbf{u}) &= \sqrt{\text{Var}(\mathbf{t})\text{Var}(\mathbf{u})}r(\mathbf{t}, \mathbf{u}) \\ &= \sqrt{\text{Var}(\mathbf{X}\mathbf{w})\text{Var}(\mathbf{Y}\mathbf{c})}r(\mathbf{X}\mathbf{w}, \mathbf{Y}\mathbf{c}).\end{aligned}\quad (9.3)$$

Equation (9.3) is actually equivalent to solving the following optimization problem:

$$\begin{aligned}\max_{\mathbf{w}, \mathbf{c}} &< \mathbf{X}\mathbf{w}, \mathbf{Y}\mathbf{c} > \\ \text{s.t.} & \|\mathbf{w}\| = 1, \|\mathbf{c}\| = 1\end{aligned}\quad (9.4)$$

or

$$\begin{aligned}J_{\text{PLS}} &= \max \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} \\ \text{s.t.} & \|\mathbf{w}\| = 1, \|\mathbf{c}\| = 1.\end{aligned}\quad (9.5)$$

9.2.2 LPP and PCA Optimization Objectives

LPP aims to project points in space \mathbf{X} into low-dimensional space $\Phi = [\phi^T(1), \phi^T(2), \dots, \phi^T(n)]^T \in \mathbf{R}^{n \times d}$ ($d < m$, $\phi = [\phi_1, \dots, \phi_d]$) via the projection matrix $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_d] \in \mathbf{R}^{m \times d}$, that is,

$$\phi(i) = \mathbf{x}(i)\mathbf{W}, \quad (i = 1, 2, \dots, n). \quad (9.6)$$

The optimal mapping of the input space can be obtained by solving the following minimization problem:

$$\begin{aligned}J_{\text{LPP}}(\mathbf{w}) &= \min \frac{1}{2} \sum_{i,j=1}^n \|\phi_i - \phi_j\|^2 s_{xij} \\ &= \min (\mathbf{w}^T \mathbf{X}^T \mathbf{D}_x \mathbf{X} \mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{S}_x \mathbf{X} \mathbf{w}) \\ \text{s.t.} & \mathbf{w}^T \mathbf{X}^T \mathbf{D}_x \mathbf{X} \mathbf{w} = 1,\end{aligned}\quad (9.7)$$

where $\mathbf{S}_x = [s_{xij}] \in \mathbf{R}^{n \times n}$ is the neighboring relationship matrix between x_i and x_j . $\mathbf{D}_x = [d_{xii}]$ is a diagonal matrix, $d_{xii} = \sum_j s_{xij}$, and

$$s_{xij} = \begin{cases} e^{-\frac{\|\mathbf{x}(i) - \mathbf{x}(j)\|^2}{2\delta_x^2}}, & \mathbf{x}(i) \text{ and } \mathbf{x}(j) \in \text{“neighbors”} \\ 0, & \text{otherwise} \end{cases} \quad (9.8)$$

δ_x is the neighbors parameter. Compute the “neighbors” of $\mathbf{x}(i)$ and $\mathbf{x}(j)$ by K-nearest neighbors method.

The LPP problem (9.7) in space X is updated as follows:

$$\begin{aligned} J_{\text{LPP}}(\mathbf{w}) &= \max \mathbf{w}^T \mathbf{X}^T \mathbf{S}_x \mathbf{X} \mathbf{w} \\ \text{s.t. } &\mathbf{w}^T \mathbf{X}^T \mathbf{D}_x \mathbf{X} \mathbf{w} = 1. \end{aligned} \quad (9.9)$$

The local structure information of X is contained in the matrices $\mathbf{X}^T \mathbf{S}_x \mathbf{X}$ and $\mathbf{X}^T \mathbf{D}_x \mathbf{X}$. The magnitude of the diagonal element values indicates the magnitude of the role of the corresponding variables in preserving the local structure. The non-diagonal elements correspond to the correlation between the observed variables. Similarly, the optimization problem for PCA can be expressed as follows:

$$\begin{aligned} J_{\text{PCA}}(\mathbf{w}) &= \max \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \\ \text{s.t. } &\mathbf{w}^T \mathbf{w} = 1. \end{aligned} \quad (9.10)$$

Based on the similarity of the optimization goals of LPP and PCA, combined with the component extraction idea of PCA included in PLS, we naturally consider fusing the LPP features into PLS to weaken the limitation of PLS, lack of local feature extraction capabilities. The simplest feature fusion method is to re-synthesize the two optimization goals, such as the GLPLS (Zhong et al. 2016), into a new optimization goal through some trade-off parameters.

9.3 Introduction to the GLPLS

GLPLS method is given in this chapter to obtain the relationship between the quality and measurement variables while maintaining the local characteristics as much as possible. The main idea is to integrate the LPP method to preserve the local structural characteristics and the PLS method to perform the relevant quality statistical analysis. As a result, GLPLS method is able not only to identify the latent characteristics direction for both the measurement and the quality data space but also to preserve (to the greatest extent possible) the local structural characteristics in the two hidden subspaces.

Consider both the manifold structure for process variables X and the product output variables Y by introducing parameters λ_1 and λ_2 to control the trade-off between the extraction of the global and local features. Therefore, the objective of GLPLS-based method is defined as

$$\begin{aligned} J_{\text{GLPLS}}(\mathbf{w}, \mathbf{c}) &= \arg \max \{ \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} + \lambda_1 \mathbf{w}^T \boldsymbol{\theta}_x \mathbf{w} + \lambda_2 \mathbf{c}^T \boldsymbol{\theta}_y \mathbf{c} \} \\ \text{s.t. } &\mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1, \end{aligned} \quad (9.11)$$

where $\boldsymbol{\theta}_x = \mathbf{X}^T \mathbf{S}_x \mathbf{X}$ and $\boldsymbol{\theta}_y = \mathbf{Y}^T \mathbf{S}_y \mathbf{Y}$ represent the local structure information of process variables and quality variables, respectively. \mathbf{S}_x , \mathbf{S}_y , \mathbf{D}_1 , and \mathbf{D}_2 are the local

feature parameter of the LPP algorithm. Parameters λ_1 and λ_2 are used to control the weight coefficients between global and local features.

It can be found from (9.11) that the objective function of GLPLS contains the objective function of the PLS algorithm $\mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c}$ and a part of the optimization problem of LPP algorithm $\mathbf{w}^T \mathbf{X}^T \mathbf{S}_x \mathbf{X} \mathbf{w}$ and $\mathbf{c}^T \mathbf{Y}^T \mathbf{S}_y \mathbf{Y} \mathbf{c}$.

The optimization function (9.11) seems to be a good combination of the PLS algorithm global characteristics and the LPP algorithm local persistence characteristics. Is that really the case? Let us analyze the solution of the optimization problem first. To solve the optimization objective function (9.11), the following Lagrange function is introduced:

$$\begin{aligned} \psi(\mathbf{w}, \mathbf{c}) = & \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} + \lambda_1 \mathbf{w}^T \boldsymbol{\theta}_x \mathbf{w} + \lambda_2 \mathbf{c}^T \boldsymbol{\theta}_y \mathbf{c} \\ & - \eta_1 (\mathbf{w}^T \mathbf{w} - 1) - \eta_2 (\mathbf{c}^T \mathbf{c} - 1). \end{aligned} \quad (9.12)$$

Then, according to the conditions for extremum, (9.11) is resolved as follows (Zhong et al. 2016):

$$J_{\text{GLPLS}}(\mathbf{w}, \mathbf{c}) = \eta_1 + \eta_2. \quad (9.13)$$

Let $\lambda_1 = \eta_1$, $\lambda_2 = \eta_2$, \mathbf{w} is best projection vector, which is the corresponding eigenvector of the largest eigenvalue $(\mathbf{I} - \boldsymbol{\theta}_x)^{-1} \mathbf{X}^T \mathbf{Y} (\mathbf{I} - \boldsymbol{\theta}_y)^{-1} \mathbf{Y}^T \mathbf{X}$, \mathbf{c} is best projection vector, which is the corresponding eigenvector of the largest eigenvalue $(\mathbf{I} - \boldsymbol{\theta}_y)^{-1} \mathbf{Y}^T \mathbf{X} (\mathbf{I} - \boldsymbol{\theta}_x)^{-1} \mathbf{X}^T \mathbf{Y}$, that is,

$$\begin{aligned} (\mathbf{I} - \boldsymbol{\theta}_x)^{-1} \mathbf{X}^T \mathbf{Y} (\mathbf{I} - \boldsymbol{\theta}_y)^{-1} \mathbf{Y}^T \mathbf{X} \mathbf{w} &= 4\eta_1 \eta_2 \mathbf{w} \\ (\mathbf{I} - \boldsymbol{\theta}_y)^{-1} \mathbf{Y}^T \mathbf{X} (\mathbf{I} - \boldsymbol{\theta}_x)^{-1} \mathbf{X}^T \mathbf{Y} \mathbf{c} &= 4\eta_1 \eta_2 \mathbf{c}. \end{aligned} \quad (9.14)$$

Equation (9.13) shows that the optimal solution of GLPLS is $\eta_1 + \eta_2$, but in the actual calculation process (9.14), the optimal solution obtained by GLPLS algorithm is $\eta_1 \eta_2$. Obviously, in most cases, the conditions for maximizing $\eta_1 + \eta_2$ and $\eta_1 \eta_2$ are different.

In order to explain the reason for this result, we once again return to the GLPLS optimization objective (9.11). Equation (9.11) is a global (PLS) and local (LPP) feature combination optimization problem. It is undeniable that this combination is reasonable to a certain extent. However, the latent variables of PLS are chosen to manifest their variation as much as possible, and the correlation between latent variables is as strong as possible. But the LPP method only needs to keep the local structure information as much as possible when constructing its latent variables. In other words, although the local features of the process variables ($\mathbf{x}(\boldsymbol{\theta}_x = \mathbf{X}^T \mathbf{S}_x \mathbf{X})$) and the quality variables ($\mathbf{y}(\boldsymbol{\theta}_y = \mathbf{Y}^T \mathbf{S}_y \mathbf{Y})$) are enhanced, the correlation between the local features is not enhanced. Therefore, this direct combination of global and local features may lead to erroneous results.

In the GLPLS method, the LPP is used to maintain local structural features. Locally linear embedding (LLE) is also a commonly used manifold learning algorithm. Like the LPP algorithm, the LLE algorithm also converts a global nonlinear problem into a combination of multiple local linear problems by maintaining local

structural information, but the LLE algorithm has fewer adjustable parameters than the LPP algorithm. Therefore, the LLE algorithm is another good solution to the problem of a strongly local nonlinear process system. The LLE algorithm has been briefly introduced in Chap. 11, and its optimization objective function is transformed into a general maximization form. Therefore, in the next section, we combine the PLS method and the LLE/LPP method in a new way, trying to maintain the global and local structural information of the process variables and quality variables at the same time, and enhance the correlation between them.

9.4 Basic Principles of GPLPLS

9.4.1 The GPLPLS Model

According to the Taylor series expansion, a nonlinear function can be written as follows:

$$F(Z) = A(Z - Z_0) + g(Z - Z_0), \quad (9.15)$$

where $A(Z - Z_0)$ and $g(Z - Z_0)$ represent the linear part and the nonlinear part, respectively. In many real systems, especially near the balance point (Z_0), the linear part is primary and the nonlinear part is secondary. The PLS method is difficult to model nonlinear systems well. Because the PLS method uses the linear dimensionality reduction method PCA to obtain the principal components, which only establishes the relationship between the linear part of the input variable space (\mathbf{X}) and the output variable space (\mathbf{Y}). In order to obtain a better model with local nonlinear features, the KPLS model (Rosipal and Trejo 2001) maps the original data to a high-dimensional feature space, while the LPPLS model (Wang et al. 2017) transforms nonlinear features into a combination of multiple local linearized features. Both of these methods can solve some nonlinear problems. However, the feature space of the KPLS model is not easy to determine, and the main linear part of the LPPLS model is more suitable to be directly described by global structural features.

In fact, the PLS optimization (9.5) includes two goals for the selected latent variable: one is that the latent variable contains variance varying as much as possible and the other is that the correlation between the latent variables of the input space and the output space is as strong as possible. Although the GLPLS model combines global and local feature information, the combination of the two is not coordinated. How does one combine the two features to maintain the same objective? According to the expression of a nonlinear function (9.15), the input and output spaces can both be divided into two parts: the linear and nonlinear parts. By introducing local structure information, the nonlinear part can be transformed into a combination of multiple local linear problems.

Inspired by the role of the PCA model ($\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}$) in the PLS model ($\mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c}$) and the limitation of the GLPLS algorithm, this section proposes a novel dimen-

sionality reduction method. It combines global (PCA) and local (LLE/LPP) features to extract latent variables of nonlinear systems. Therefore, the input space X or the output space Y is mapped to the new feature space X_F and Y_F , respectively. The new feature space contains a global linear subspace and multiple local linear subspaces. Use the new feature space X_F and Y_F to replace the original space X and Y , respectively. Consequently, a new objective function of the global plus local projection to latent structure (GPLPLS) method is shown in the following new optimization objective

$$J_{GPLPLS}(w, c) = \arg \max \{w^T X_F^T Y_F c\} \tag{9.16}$$

$$s.t. w^T w = 1, c^T c = 1,$$

where X_F and Y_F satisfy $X_F = X + \lambda_x \theta_x^{\frac{1}{2}}$ and $Y_F = Y + \lambda_y \theta_y^{\frac{1}{2}}$.

It is found that the new feature spaces X_F and Y_F are both divided into linear part (X, Y) and nonlinear part ($\lambda_x \theta_x^{\frac{1}{2}}, \lambda_y \theta_y^{\frac{1}{2}}$), similar as (9.15). Figure 9.2 shows the principle of the GPLPLS method. Here X^{global} and Y^{global} are the corresponding linear part in the input space and the output space, respectively. They will be projected to the dimensionality reduction space by the traditional global projection method, PLS. X^{local} and Y^{local} are the corresponding nonlinear parts, which will be dimensionality reduction projected by the local-preserving projection method (LPP).

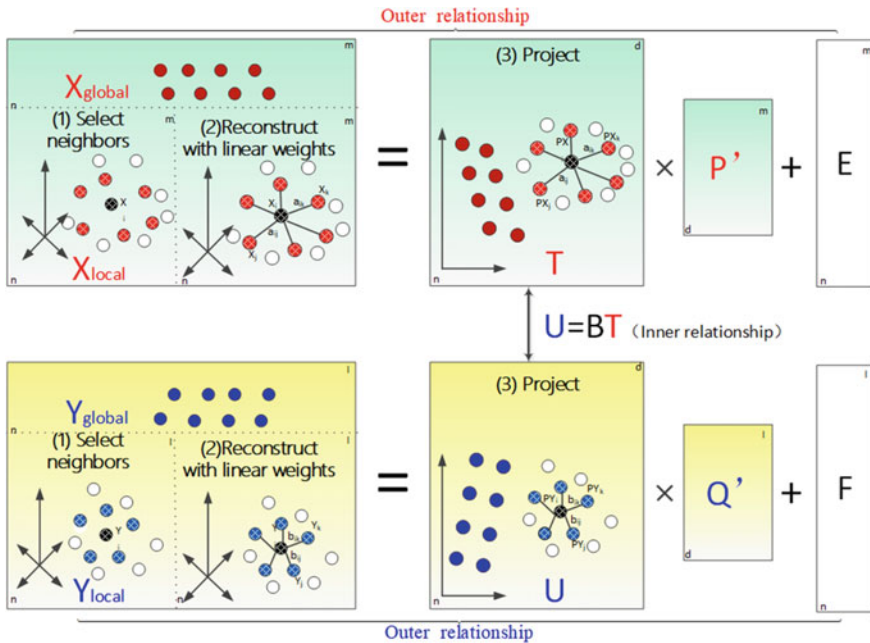


Fig. 9.2 The schematic diagram of the GPLPLS method

The core of extracting the principle components is PCA. So the linear model of \mathbf{X} and \mathbf{Y} is established by (9.16). It actually contains two relations: one relationship is that the input and output spaces are divided into “score” and “load” (external relationship), and the other relationship is the relationship between the latent variables of the input space and output space (internal relationship). These two relationships can also be seen from the schematic diagram (Fig. 9.2) of the GPLPLS model. Obviously, we can keep only the internal model, or the external model, or retain the local structure information of the internal model and the external model at the same time. Therefore, by setting four different values of λ_x and λ_y , four different optimization objective functions can be set as follows:

- (1) PLS optimization objective function: $\lambda_x = 0, \lambda_y = 0$.
- (2) GPLPLS _{x} optimization objective function: $\lambda_x > 0, \lambda_y = 0$.
- (3) GPLPLS _{y} optimization objective function: $\lambda_x = 0, \lambda_y > 0$.
- (4) GPLPLS _{$x+y$} optimization objective function: $\lambda_x > 0, \lambda_y > 0$.

9.4.2 Relationship Between GPLPLS Models

The optimization objective function of the GPLPLS method is given by (9.16). There are three GPLPLS models according to different values of λ_x and λ_y . What is the relationship between the three GPLPLS models? What is the difference between their modeling? These issues will be discussed in this section.

Suppose the original relationship is $\mathbf{Y} = f(\mathbf{X})$. Local linear embedding or local-preserving projection can be regarded as the equilibrium point of system linearization. From this perspective, the models with different combinations of λ_x and λ_y are as follows:

- (1) PLS model: $\hat{\mathbf{Y}} = \mathbf{A}_0\mathbf{X}$.
- (2) GPLPLS _{x} model: $\hat{\mathbf{Y}} = \mathbf{A}_1[\mathbf{X}, \mathbf{x}_{z_i}]$.
- (3) GPLPLS _{y} model: $\hat{\mathbf{Y}} = \mathbf{A}_2[\mathbf{X}, \mathbf{f}(\mathbf{x}_{l_j})]$.
- (4) GPLPLS _{$x+y$} model: $\hat{\mathbf{Y}} = \mathbf{A}_3[\mathbf{X}, \mathbf{x}_{z_i}, \mathbf{f}(\mathbf{x}_{l_j})]$.

Here \mathbf{x}_{z_i} ($i = 1, 2, \dots, k_x$) and $\mathbf{y}_{l_j} = \mathbf{f}(\mathbf{x}_{l_j})$ ($j = 1, 2, \dots, k_y$) are the local feature points of the input space and output space, respectively. $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$, and \mathbf{A}_3 are the model coefficient matrices. Obviously, PLS uses a simple linear approximation of the original system. This approximation effect is generally not good for a nonlinear relatively strong system. The GPLPLS uses the method of spatial local decomposition and approximates the original system with the sum of multiple simple linear models. GPLPLS _{x} or GPLPLS _{y} is a special case of GPLPLS _{$x+y$} . It seems that these three combinations have embraced all the possible GPLPLS models. Let us go back to the GPLPLS _{$x+y$} model’s optimization function again.

$$\begin{aligned}
J_{\text{GPLPLS}_{x+y}}(\mathbf{w}, \mathbf{c}) &= \arg \max_{\mathbf{w}, \mathbf{c}} \{ \mathbf{w}^T (\mathbf{X} + \lambda_x \boldsymbol{\theta}_x^{\frac{1}{2}})^T (\mathbf{Y} + \lambda_y \boldsymbol{\theta}_y^{\frac{1}{2}}) \mathbf{c} \} \\
&= \arg \max_{\mathbf{w}, \mathbf{c}} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} + \lambda_x \mathbf{w}^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \mathbf{Y} \mathbf{c} \right. \\
&\quad \left. + \lambda_y \mathbf{w}^T \mathbf{X}^T \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c} + \lambda_x \lambda_y \mathbf{w}^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c} \right\} \\
\text{s.t. } &\mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1.
\end{aligned} \tag{9.17}$$

Obviously, (9.17) contains two coupled components ($\boldsymbol{\theta}_x^{\frac{1}{2}T} \mathbf{Y}$ and $\mathbf{X}^T \boldsymbol{\theta}_y^{\frac{1}{2}}$), which represent the correlation between the linear primary part and the nonlinear part. In some cases, these coupled components may have a negative impact on modeling. On the other hand, in addition to the external relationship between the input and output space which can be extended to a combination of linear and nonlinear, the internal relationship between the input and output space (the final model) can also be described as a combination of linear and nonlinear. Therefore, it is natural that we can model the linear and nonlinear parts without considering the coupling component between the two parts. Correspondingly, there is no need to consider the coupling component between the linear and nonlinear parts in the optimization function of the model. Therefore, the optimization objective of the following GPLPLS_{xy} model can be obtained:

$$\begin{aligned}
J_{\text{GPLPLS}_{xy}}(\mathbf{w}, \mathbf{c}) &= \arg \max_{\mathbf{w}, \mathbf{c}} \{ \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} + \lambda_{xy} \mathbf{w}^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c} \} \\
\text{s.t. } &\mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1.
\end{aligned} \tag{9.18}$$

Among them, λ_{xy} parameters control the trade-off between global and local features.

9.4.3 Principal Components of the GPLPLS Model

In this section, we will introduce how to obtain the principal components of the GPLPLS model. In order to facilitate the comparison with the traditional linear PLS model, denoted by $\mathbf{E}_{0F} = \mathbf{X}_F$ and $\mathbf{F}_{0F} = \mathbf{Y}_F$. The optimization objective functions of four GPLPLS models are included in the following optimization objectives:

$$\begin{aligned}
J_{\text{GPLPLS}}(\mathbf{w}, \mathbf{c}) &= \arg \max_{\mathbf{w}, \mathbf{c}} \{ \mathbf{w}^T \mathbf{X}_F^T \mathbf{Y}_F \mathbf{c} + \lambda_{xy} \mathbf{w}^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c} \} \\
\text{s.t. } &\mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1,
\end{aligned} \tag{9.19}$$

where at least one of $[\lambda_x, \lambda_y]$ and λ_{xy} is nonzero. The steps of obtaining latent variables of the GPLPLS model (9.19) are as follows.

First, the Lagrangian multiplier factor is introduced to transform the objective function (9.19) into the following unconstrained form:

$$\begin{aligned} \Psi(\mathbf{w}_1, \mathbf{c}_1) = & \mathbf{w}_1^T \mathbf{E}_{0F}^T \mathbf{F}_{0F} \mathbf{c}_1 + \lambda_{xy} \mathbf{w}_1^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c}_1 \\ & - \lambda_1 (\mathbf{w}_1^T \mathbf{w}_1 - 1) - \lambda_2 (\mathbf{c}_1^T \mathbf{c}_1 - 1). \end{aligned} \quad (9.20)$$

Let $(\partial\Psi)/(\partial\mathbf{w}_1) = 0$ and $(\partial\Psi)/(\partial\mathbf{c}_1) = 0$, we can find the optimal solution of \mathbf{w}_1 and \mathbf{c}_1 . Then the objective function (9.19) is transformed as

$$\left[\mathbf{E}_{0F}^T \mathbf{F}_{0F} + \lambda_{xy} \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \right]^T \left[\mathbf{E}_{0F}^T \mathbf{F}_{0F} + \lambda_{xy} \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \right] \mathbf{w}_1 = \theta^2 \mathbf{w}_1 \quad (9.21)$$

$$\left[\mathbf{F}_{0F}^T \mathbf{E}_{0F} + \lambda_{xy} \boldsymbol{\theta}_y^{\frac{1}{2}T} \boldsymbol{\theta}_x^{\frac{1}{2}} \right]^T \left[\mathbf{F}_{0F}^T \mathbf{E}_{0F} + \lambda_{xy} \boldsymbol{\theta}_y^{\frac{1}{2}T} \boldsymbol{\theta}_x^{\frac{1}{2}} \right] \mathbf{c}_1 = \theta^2 \mathbf{c}_1, \quad (9.22)$$

where $\theta = \mathbf{w}^T \mathbf{X}_F^T \mathbf{Y}_F \mathbf{c} + \lambda_{xy} \mathbf{w}^T \boldsymbol{\theta}_x^{\frac{1}{2}T} \boldsymbol{\theta}_y^{\frac{1}{2}} \mathbf{c}$. The target vectors \mathbf{w}_1 and \mathbf{c}_1 are calculated from (9.21) and (9.22). After obtaining the target vector (that is, the direction vector of the latent variables), the latent variables \mathbf{t}_1 and \mathbf{u}_1 , the load vectors \mathbf{p}_1 and \mathbf{q}_1 , and the residual matrices \mathbf{E}_1 and \mathbf{F}_1 can be calculated as follows:

$$\mathbf{t}_1 = \mathbf{E}_{0F} \mathbf{w}_1, \quad \mathbf{u}_1 = \mathbf{F}_{0F} \mathbf{c}_1 \quad (9.23)$$

$$\mathbf{p}_1 = \frac{\mathbf{E}_{0F}^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2}, \quad \mathbf{q}_1 = \frac{\mathbf{F}_{0F}^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2} \quad (9.24)$$

$$\mathbf{E}_{1F} = \mathbf{E}_{0F} - \mathbf{t}_1 \mathbf{p}_1^T, \quad \mathbf{F}_{1F} = \mathbf{F}_{0F} - \mathbf{t}_1 \mathbf{q}_1^T. \quad (9.25)$$

Similar to the PLS method, the other latent variables of the GPLPLS model can be obtained by continuing to decompose the residual matrices \mathbf{E}_{iL} and \mathbf{F}_{iL} ($i = 1, 2, \dots, d - 1$). Usually, the first d latent variables are used to produce a better predictive regression model and d can be determined by the cross-validation test (Zhou et al. 2010).

The above is the establishment of the GPLPLS model and its principal component extraction process. Now let's compare GPLPLS model with the GLPLS model.

First of all, GPLPLS likes the GLPLS method at the main idea, i.e., to combine local and global structural features (covariance). Obviously, the GPLPLS method integrates global and local structural features better than the GLPLS method. Different from the GLPLS method, the GPLPLS method not only maintains the local structural features, but also extracts the relevant information in the input space and output space as much as possible. Therefore, the GPLPLS method can extract the largest global correlation as much as possible, while extracting the local structural correlation between process and quality variables.

Compared with the LPPLS method (Chap. 10) and LLEPLS method (Chap. 11), all the characteristics of the LPPLS method are described by local features. This indiscriminate description has advantages in strongly nonlinear systems, but it may not necessarily have advantages in linearly dominant but locally nonlinear systems. The GPLPLS method proposed in this chapter is a process aimed at linear advan-

tages, but it still maintains some nonlinear relationships. It integrates global features (covariance) and nonlinear correlation (multivariate) as much as possible.

9.5 GPLPLS-Based Quality Monitoring

9.5.1 Process and Quality Monitoring Based on GPLPLS

The GPLPLS-based monitoring method is very similar to the PLS method. The common monitoring indicators of PLS are T^2 and SPE. In Chap. 11, it has been explained in detail that SPE statistics is not suitable for monitoring residual space of PLS. Therefore, in this chapter, the process monitoring based on the GPLPLS method uses statistics to monitor the principal component subspace and the remaining subspace. The monitoring process is also divided into two parts: offline training and online monitoring. The detailed process is as follows.

The input space X and the output space Y of the GPLPLS model are mapped to a low-dimensional space defined by a small number of latent variables $[t_1, \dots, t_d]$. The decomposition of E_{0F} and F_{0F} is as follows:

$$\begin{aligned} E_{0F} &= \sum_{i=1}^d t_i p_i^T + \bar{E}_{0L} = T P^T + \bar{E}_{0F} \\ F_{0F} &= \sum_{i=1}^d t_i q_i^T + \bar{F}_{0L} = T Q^T + \bar{F}_{0F}, \end{aligned} \quad (9.26)$$

where $T = [t_1, t_2, \dots, t_d]$ is the score matrix. $P = [p_1, \dots, p_d]$ and $Q = [q_1, \dots, q_d]$ are the load matrices of the process variable E_{0F} and the quality variable F_{0F} , respectively. Use E_{0F} instead of t_i :

$$T = E_{0F} R = \left(I + \lambda_x S_x^{\frac{1}{2}} \right) E_{0F} R, \quad (9.27)$$

where $R = [r_1, \dots, r_d]$ is the decomposition matrix, and

$$r_i = \prod_{j=1}^{i-1} (I_n - w_j p_j^T) w_i.$$

It is noted that E_{0F} contains the results of locality-preserving learning. Operations (9.26) and (9.27) are executable during the model training. But the data is sampled real time during the process of online monitoring. The individual real-time data cannot be constructed for the transformational matrix S_x or S_y for the locality learning.

Considering the practical application of (9.26) and (9.27), they should be transformed as the decomposition of normalized matrices \mathbf{E}_0 and \mathbf{F}_0 ,

$$\mathbf{E}_0 = \mathbf{T}_0 \mathbf{P}^T + \bar{\mathbf{E}}_0 \quad (9.28)$$

$$\mathbf{F}_0 = \mathbf{T}_0 \bar{\mathbf{Q}}^T + \bar{\mathbf{F}}_0 = \mathbf{E}_0 \mathbf{R} \bar{\mathbf{Q}}^T + \bar{\mathbf{F}}_0, \quad (9.29)$$

where $\mathbf{T}_0 = \mathbf{E}_0 \mathbf{R}$, $\bar{\mathbf{E}}_0 = \mathbf{E}_0 - \mathbf{T}_0 \mathbf{P}^T$, and $\bar{\mathbf{Q}} = \mathbf{T}_0^+ \mathbf{F}_0$.

During the online monitoring for new samples \mathbf{x} and \mathbf{y} (standardized data), an oblique projection is introduced in the input space \mathbf{x} :

$$\mathbf{x} = \hat{\mathbf{x}} + \mathbf{x}_e \quad (9.30)$$

$$\hat{\mathbf{x}} = \mathbf{R} \mathbf{P}^T \mathbf{x} \quad (9.31)$$

$$\mathbf{x}_e = (\mathbf{I} - \mathbf{R} \mathbf{P}^T) \mathbf{x}. \quad (9.32)$$

The statistics T_{pc}^2 and T_e^2 of the principal component space and the remaining subspace are calculated as follows:

$$\mathbf{t} = \mathbf{R}^T \mathbf{x} \quad (9.33)$$

$$T_{pc}^2 := \mathbf{t}^T \mathbf{\Lambda}^{-1} \mathbf{t} = \mathbf{t}^T \left\{ \frac{1}{n-1} \mathbf{T}_0^T \mathbf{T}_0 \right\}^{-1} \mathbf{t} \quad (9.34)$$

$$T_e^2 := \mathbf{x}_e^T \mathbf{\Lambda}_e^{-1} \mathbf{x}_e = \mathbf{x}_e^T \left\{ \frac{1}{n-1} \mathbf{x}_e^T \mathbf{x}_e \right\}^{-1} \mathbf{x}_e, \quad (9.35)$$

where $\mathbf{\Lambda}$ and $\mathbf{\Lambda}_e$ are covariance matrices. T_{pc}^2 and T_e^2 are statistics with the threshold $\text{Th}_{pc,\alpha}$ and Th_e , respectively. Considering the statistics T_{pc}^2 and T_e^2 are not obtained through normalized data \mathbf{E}_0 , and the output variables may not obey the Gaussian distribution. Therefore, the corresponding thresholds cannot be calculated from F-distribution. So their probability density functions should be estimated first by non-parametric kernel density estimation (KDE) (Lee et al. 2010).

The fault diagnosis logic based on the GPLPLS model is as follows:

$$\begin{array}{ll} T_{pc}^2 > \text{Th}_{pc,\alpha} & \text{Quality-relevant faults} \\ T_{pc}^2 > \text{Th}_{pc,\alpha} \text{ or } T_e^2 > \text{Th}_e & \text{Process-relevant faults} \\ T_{pc}^2 \leq \text{Th}_{pc,\alpha} \text{ and } T_e^2 \leq \text{Th}_e & \text{Fault free} \end{array} \quad (9.36)$$

The process monitoring of GPLPLS algorithm with multiple input and multiple output data is as follows:

- (1) Standardize the original data \mathbf{X} and \mathbf{Y} . Calculate \mathbf{T}_0 , $\bar{\mathbf{Q}}$, and \mathbf{R} based on GPLPLS algorithm (9.28) and (9.29). Determine the number of principal components d by cross-validation.

- (2) Construct the input remaining subspace \overline{x}_e .
- (3) The thresholds are calculated according to the non-parametric KDE estimation, and the fault diagnosis is performed with the detection logic (9.36).

9.5.2 Posterior Monitoring and Evaluation

Many quality-related process monitoring methods have been verified on the well-known TE process simulation platform. The goal of most methods is to make the quality-related alarm rate as high as possible, but the reasonability of monitoring result seems to receive little attention. Therefore, similar to the performance evaluation index of the control loop, we introduce a posterior monitoring assessment (PMA) index to evaluate the reasonability of quality-related alarm rate. PMA is defined as follows:

$$\text{PMA} = \frac{\mathbb{E}(\mathbf{y}_N^2)}{\mathbb{E}(\mathbf{y}_F^2)}, \quad (9.37)$$

where $\mathbb{E}(\cdot)$ is the mathematical expectation, \mathbf{y}_N and \mathbf{y}_F are the output data of the training data set and the output data of the fault data set, respectively. It is noted that they are both normalized by the mean and standard deviation of \mathbf{y}_N . $\text{PMA} \rightarrow 1$ indicates that the quality of the fault data is close to normal operation; $\text{PMA} > 1$ indicates the data quality is better than the normal. Moreover, PMA far from 1 means that the quality is very different from the normal, and the corresponding quality-related index T^2 (PLS method) or T_{pc}^2 (GPLPLS method) should be higher, and the others should be lower.

However, the widespread controllers reduce the impact of certain failures, especially small fault. So a single PMA indicator cannot truly reflect the dynamic changes, two PMA indicators are adopted to describe dynamic and steady-state effects, respectively,

$$\text{PMA}_1 = \min \left\{ \frac{\mathbb{E}(\mathbf{Y}_N^2(k_0 : k_1, i))}{\mathbb{E}(\mathbf{Y}_F^2(k_0 : k_1, i))} \right\}, \quad i = 1, 2, \dots, l \quad (9.38)$$

$$\text{PMA}_2 = \min \left\{ \frac{\mathbb{E}(\mathbf{Y}_N^2(k_2 : n, i))}{\mathbb{E}(\mathbf{Y}_F^2(k_2 : n, i))} \right\}, \quad i = 1, 2, \dots, l, \quad (9.39)$$

where $k = 0, 1, 2$ is constant. It is noted that the worst strategy is selected in order to ensure the rationality of the evaluation. Moreover, the two PMA indicators are only used to test whether the previous fault detection results are reasonable. Their evaluations are objective but not indicate whether the fault is quality related, compared with the detection based on GPLPLS model. The quality testing is necessary for further diagnosis.

9.6 TE Process Simulation Analysis

Process monitoring and fault diagnosis based on the GPLPLS model are tested on the TE simulation platform. The monitoring performance of several models, such as PLS, a concurrent projection to the latent structures (CPLS) (Qin 2012), and GPLPLS, are compared. The input and output spaces are projected and decomposed into five subspaces in CPLS: input principle subspace, input residual subspace, output principle subspace, output residual subspace, and joint input-output subspace. Just focusing on the quality-related faults, the principle and residual subspaces of input are replaced by the input remaining subspace x_e in CPLS model, and the corresponding monitoring statistics are replaced by T_e^2 . The output principle and residual subspaces in the CPLS model are not considered in order to highlight process-based quality monitoring. Two different data sets are used from (Zhang et al. 2017) and (Wang et al. 2017).

9.6.1 Model and Discussion

The input matrix is composed of process variables [XMEAS(1:22)] and manipulated variables [XMV(1:11), except XMV(5) and XMV(9)]. The output matrix is composed of mass variable [XMEAS (35), XMEAS (36)]. The training data is normal data IDV(0) and the test data is 21 fault data IDV(1-21). The threshold is calculated based on the confidence level 99.75% (see equation (1.10) for detail).

The simulation parameters of the GPLPLS model, especially the GPLPLS_{xy} model) are $k_x = 22$, $k_y = 23$, $\lambda_x = \lambda_y = 0$, $\lambda_{xy} = 1$, $k_0 = 161$. Note that the local nonlinear structure features are extracted by the LLE method. Number of principal components of PLS, CPLS, and GPLPLS models are 6, 6, and 2, respectively, determined by the cross-validation method. $k_1 = n = 960$, $k_2 = 701$. The detection results including FDR, FAR, and indicator PMA are listed in Table 9.1.

With these two PMA indices in Table 9.1, 21 faults are divided into two types: quality-independent faults ($PMA_1 > 0.9$ or $PMA_1 + PMA_2 > 1.5$) including IDV(3,4,9,11,14,15,19) and quality-related faults. Furthermore, the quality-related faults are further subdivided into four types:

Type 1: fault has a slight impact on quality, [IDV(10,16,17, and 20)], $0.5 < PMA_i < 0.8$ $i = 1, 2$.

Type 2: fault is quality recoverable, [IDV(1,5, and 7)], $PMA_1 < 0.35$ and $PMA_2 > 0.65$.

Type 3: fault has a serious impact on quality, [IDV(2, 6, 8, 12, 13, and 18)], $PMA_i < 0.1$ $i = 1, 2$.

Type 4: fault causes the output variables to drift slowly, [IDV(21)].

Table 9.1 FDRs of PLS, CPLS, GPLPLS_{xy}, and PMA

IDV	PLS		CPLS		GPLPLS _{xy}		PMA	
	T ² _{pc}	T ² _e	T ² _{pc}	T ² _e	T ² _{pc}	T ² _e	PMA ₁	PMA ₂
1	99.63	99.88	84.13	99.75	35.00	99.75	0.2040	0.6930
2	98.50	97.25	94.75	98.25	74.00	97.88	0.0660	0.0580
3	1.00	0.88	0.13	1.13	0.25	1.25	0.7720	0.8670
4	19.13	100.00	7.25	100.00	0.50	100.00	0.8880	0.9277
5	22.00	100.00	17.38	100.00	13.25	100.00	0.3018	0.9461
6	99.25	100.00	98.25	100.00	96.88	100.00	0.0029	0.0026
7	100.00	100.00	97.88	100.00	26.00	100.00	0.1439	0.9721
8	96.00	97.00	76.13	97.88	72.63	97.88	0.0596	0.0951
9	0.50	1.00	0.38	1.63	0.38	0.75	0.8977	0.8465
10	26.38	82.63	16.38	84.63	17.50	84.75	0.5888	0.5064
11	26.63	75.75	8.13	77.13	1.50	77.25	0.7830	0.6956
12	97.50	99.88	83.75	99.75	71.88	99.88	0.0404	0.0232
13	94.88	95.00	88.00	95.13	75.50	95.25	0.0229	0.0208
14	91.50	100.00	20.88	100.00	0.38	100.00	1.0721	0.8580
15	1.25	1.38	1.25	2.88	3.13	3.25	0.9027	0.5710
16	20.13	37.63	9.13	44.00	8.63	44.00	0.7770	0.5355
17	77.38	96.63	36.50	97.00	8.75	96.63	0.6443	0.6862
18	89.38	90.13	89.00	89.88	87.00	90.13	0.0049	0.0037
19	0.50	41.13	0.00	39.00	0.00	36.13	0.9453	0.8859
20	30.50	90.75	20.13	88.25	12.50	90.25	0.6700	0.7366
21	41.88	47.63	37.25	45.75	21.25	50.75	0.2342	0.1063

This classification is not only a preliminary result depending on the choice of parameters k_0 , k_1 , and k_2 , but it also has a reference value. All methods show the consistent results for the serious quality-related faults, which are not discussed in the next fault detection analysis.

9.6.2 Fault Diagnosis Analysis

Form the above results, it is found that for some faults, their detection results are not consistent with different methods, including quality-recoverable faults, slight quality-related faults, and quality-independent faults. The detailed analysis for the three situations is given below. For all the monitoring graphs, the horizontal axis represents the sample, the vertical axis represents the statistics (the picture above

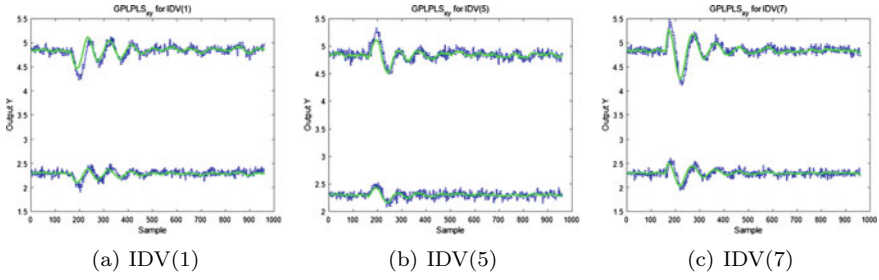


Fig. 9.3 Output prediction for IDV(1), IDV(5), and IDV(7) using the GPLPLS_{xy} method

represents T_{pc}^2 , the picture below represents T_e^2), and the red dotted line is the threshold with confidence level 99.75%. The blue line is the actual monitoring value. For all prediction graphs, the horizontal axis represents the sample, the vertical axis represents the output value, the blue dashed line is actual value, and the green line is for the prediction.

(1) Quality-recoverable fault

Quality-recoverable faults include IDV(1), IDV(5), and IDV(7). They are all step-change faults, but the feedback or cascade controller can reduce their effect on quality during the actual process. Therefore, the quality variables in the faults IDV(1), IDV(5), and IDV(7) should return to normal. The output prediction is shown in Fig. 9.3. As an example, the corresponding fault monitoring results for IDV(7) are shown in Fig. 9.4 which correspond to the PLS, CPLS, and GPLPLS_{xy} models, respectively. Here the statistics T_{pc}^2 and T_e^2 detected the input space for process-related faults. For the GPLPLS_{xy} model, the value of the T_{pc}^2 statistic returns to the normal value, while the T_e^2 statistic still maintains a high value. This means that these faults are quality-recoverable faults. PLS and CPLS reported that these faults are quality-related faults but give many false alarms, especially for IDV(7). The statistical value of T_{pc}^2 is also very close to the threshold, but still exceeds the threshold. They still indicated the fault alarm even when the operation have returned to normal under the controller. They fail to grasp the essence of the fault detection problem

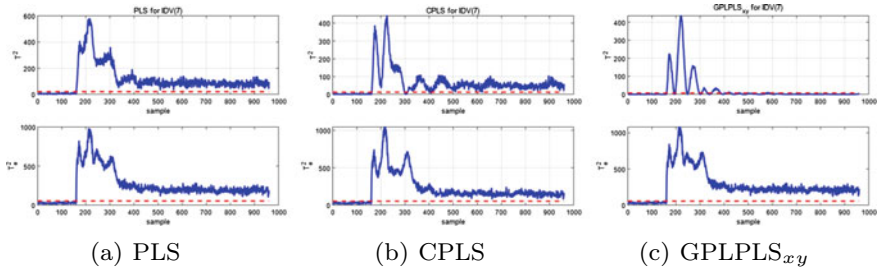


Fig. 9.4 PLS, CPLS, and GPLPLS_{xy} monitoring results for IDV(7)

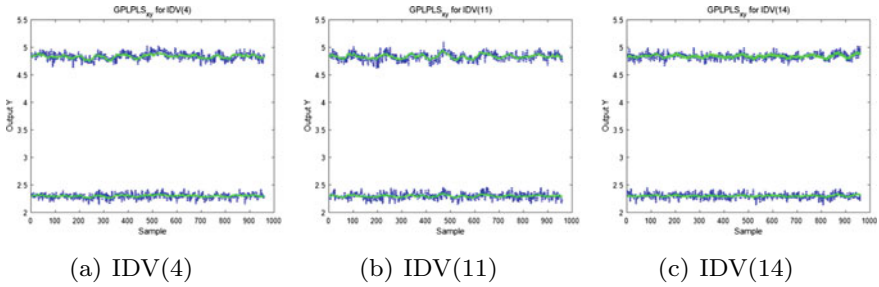


Fig. 9.5 Output prediction for IDV(4), IDV(11), and IDV(14) using the GPLPLS_{xy} method

with recoverable quality. In this case, the GPLPLS_{xy} method can accurately reflect the process and quality changes.

(2) Quality-independent fault

Quality-independent faults include IDV(4), IDV(11), and IDV(14), but they are related to process. All these faults are related to the reactor cooling water, and these interferences hardly affect the quality of output products. The corresponding output quality prediction of GPLPLS_{xy} methods is shown in Fig. 9.5. The monitoring results for IDV(14) by PLS, CPLS, and GPLPLS_{xy} methods are shown in Fig. 9.6. In the GPLPLS_{xy} model, T_{pc}^2 are almost under the threshold, which indicates that these faults are not related to quality. But for PLS and CPLS models, these faults are detected both in T_{pc}^2 and T_e^2 . In other words, PLS or CPLS model shows that these interferences are related to quality. Compared with PLS, CPLS method can filter out fault alarm to a certain extent in T_{pc}^2 , but still has higher alarm than GPLPLS_{xy}. For quality-independent fault, PLS and CPLS have a high detection rate, but fails to indicate the quality-independent faults.

(3) Slight quality-related faults

Faults, such as IDV(10), IDV(16), IDV(17), and IDV(20), have a slight impact on quality. Few people study this type of failure. Their quality-related alarm rates are similar to quality-recoverable faults. Although they are quality related, they have little impact on quality. Their T_{pc}^2 value of related monitoring statistics is relatively

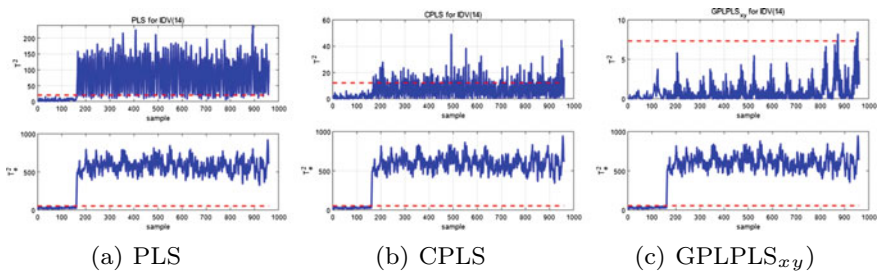


Fig. 9.6 PLS, CPLS, and GPLPLS_{xy} monitoring results for IDV(14)

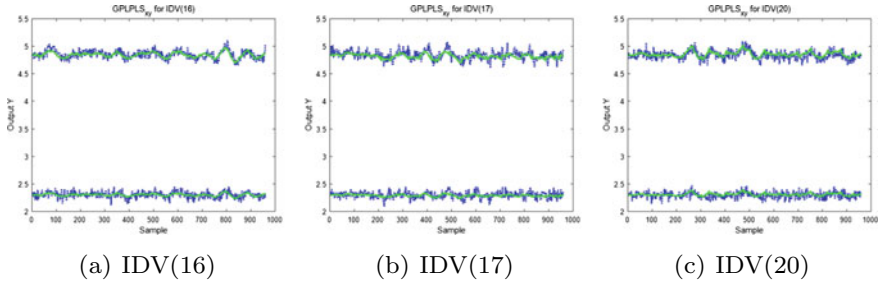


Fig. 9.7 Output predicted values for IDV(16), IDV(17), and IDV(20) using the GPLPLS_{xy} method

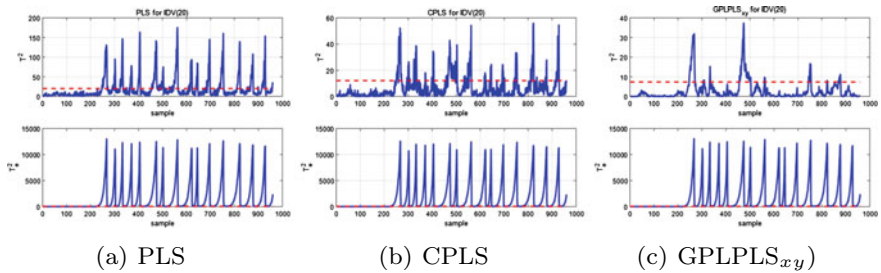


Fig. 9.8 PLS, CPLS, and GPLPLS_{xy} monitoring results for IDV(20)

small. To some extent, these faults can also be regarded as failures that have nothing to do with quality. Many methods, such as the PLS method, fail to detect them accurately. The output prediction values of GPLPLS_{xy} models are shown in Fig. 9.7. The monitoring results of the three models for fault IDV(20) are shown in Fig. 9.8. It can be seen that the monitoring results of the GPLPLS_{xy} model are the most accurate, and the PLS and CPLS models give false alarm results. In the GPLPLS_{xy} model, process changes better match quality changes.

From the three situations analyzed above, it can be seen that the GPLPLS method can filter harmful alarm situations. It can be used for minor quality-related failures, quality-unrelated failures, and quality-recoverable failures. There are two possible reasons for the good fault diagnosis performance of the GPLPLS method: first, the principal component of the GPLPLS method is based on the global features of nonlinear local structural features, and the method enhances its nonlinear mapping ability. Secondly, the GPLPLS method uses a non-Gaussian threshold, which makes it possible to process the signal that does not necessarily satisfy the Gaussian assumption.

9.6.3 Comparison of Different GPLPLS Models

For the same data set above, the FDRs of the other three GPLPLS_x, GPLPLS_y, and GPLPLS_{x+y} models (local nonlinear structural features are all extracted by the LLE method) are shown in Table 9.2, where $K = [k_x, k_y]$. It can be seen from the table that the results of these methods are very good, and consistent conclusions are drawn. Especially the FDR of GPLPLS_{x+y} model and the GPLPLS_{xy} model are very similar.

In order to discuss these models more clearly, fault IDV (7) is selected for further analysis. It can be seen from Table 9.2 that the monitoring results of IDV(7) by the GPLPLS_y model are obviously inconsistent with other methods. T² statistics give a higher alarm (79.25%). According to the previous analysis, this alarm is an annoying false alarm. The other three models have relatively low alarm rates for fault IDV (7), near 26%, which means that the monitoring effect is very good. The possible reason for false alarm is that the GPLPLS_y model only enhances the local nonlinear structure characteristics in the output space. It is linear to the input space and the output space

Table 9.2 FDRs of GPLPLS methods with LLE local feature

IDV	GPLPLS _x		GPLPLS _y		GPLPLS _{x+y}		GPLPLS _{xy}	
	$k_x=16$		$k_y=16$		$K = [22, 24]$		$K = [22, 23]$	
	T ² _{pc}	T ² _e	T ² _{pc}	T ² _e	T ² _{pc}	T ² _e	T ² _{pc}	T ² _e
1	35.50	99.75	38.75	99.75	35.13	99.75	35.00	99.75
2	70.75	98.38	95.13	98.13	74.00	98.38	74.00	98.38
3	0.00	1.38	1.00	1.25	0.25	1.13	0.25	1.38
4	0.00	100.00	1.25	100.00	0.50	100.00	0.50	100.00
5	10.75	100.00	19.25	100.00	13.25	100.00	13.25	100.00
6	96.13	100.00	98.75	100.00	96.88	100.00	96.88	100.00
7	23.50	100.00	79.25	100.00	26.25	100.00	26.00	100.00
8	68.63	97.88	81.88	97.88	72.63	97.88	72.63	97.88
9	0.00	1.50	0.75	1.38	0.38	1.13	0.38	1.25
10	13.88	84.75	21.13	84.75	17.50	84.38	17.50	84.50
11	0.88	77.50	2.88	77.00	1.50	76.63	1.50	76.75
12	68.25	99.88	87.00	99.75	71.88	99.88	71.88	99.88
13	72.63	95.25	88.00	95.13	75.50	95.13	75.50	95.25
14	0.00	100.00	3.25	100.00	0.50	100.00	0.38	100.00
15	0.88	2.50	1.38	3.50	3.13	1.63	3.13	1.63
16	7.13	45.38	12.88	43.50	8.63	42.63	8.63	43.75
17	1.88	96.88	11.38	97.00	8.88	96.75	8.75	96.88
18	86.38	90.00	88.88	90.00	87.00	90.00	87.00	89.88
19	0.00	38.25	0.00	38.50	0.00	37.75	0.00	37.38
20	8.63	90.63	22.50	89.75	12.50	90.50	12.50	90.38
21	14.00	52.75	31.63	44.25	21.25	49.63	21.25	50.25

is nonlinear. Process monitoring results may be better. However, the input space of the TE simulation process may also have strong nonlinearity, which leads to the poor monitoring results of GPLPLS_y model, and the other three models show higher consistency with this type of fault.

The above results of the GPLPLS models are obtained by combining with the LLE method to retain local nonlinear structural features. Below, the monitoring results of the GPLPLS model combined with another local retention algorithm LPP method are given, as shown in Table 9.3, where $\Sigma = [\sigma_x, \sigma_y]$. It can be seen that Table 9.3 gives consistent conclusions, so the analysis will not be performed here.

Many methods have the similar fusion idea of global projection and local preserving, such as GLPLS, LPPLS, and others. These methods all need to adjust parameters, and different parameters have different results. In order to be as consistent as possible with the existing results of other methods, we chose the same data set in Wang et al. (2017) for the following tests.

Table 9.3 FDRs of GPLPLS methods with LPP local feature

IDV	GPLPLS _x		GPLPLS _y		GPLPLS _{x+y}		GPLPLS _{xy}	
	$k_x=16$		$k_y=16$		$K = [22, 24]$		$K = [22, 23]$	
	$\sigma_x = 2$		$\sigma_x = 0.05$		$\Sigma = [2, 1]$		$\Sigma = [0.05, 1.3]$	
	T_{pc}^2	T_e^2	T_{pc}^2	T_e^2	T_{pc}^2	T_e^2	T_{pc}^2	T_e^2
1	49.00	99.75	61.13	99.75	47.63	99.75	41.75	99.75
2	56.38	98.38	94.88	98.13	46.75	98.38	60.00	98.38
3	0.13	1.25	0.75	1.75	0.50	1.25	0.25	1.25
4	0.38	100.00	1.38	100.00	0.50	100.00	0.38	100.00
5	12.88	100.00	19.63	100.00	13.75	100.00	12.75	100.00
6	96.75	100.00	99.13	100.00	97.00	100.00	96.88	100.00
7	26.00	100.00	56.75	100.00	27.13	100.00	26.63	100.00
8	71.50	97.88	85.25	97.88	72.63	97.88	72.00	97.88
9	0.25	1.00	1.00	1.50	0.38	1.25	0.38	1.25
10	18.25	84.00	21.50	84.75	18.38	84.25	18.00	84.25
11	2.00	76.75	3.00	77.13	2.38	77.00	1.88	77.00
12	71.25	99.88	88.38	99.75	71.75	99.88	71.63	99.88
13	75.13	95.25	84.75	95.13	75.75	95.25	75.63	95.25
14	0.25	100.00	8.75	100.00	0.38	100.00	0.50	100.00
15	2.75	1.75	1.50	3.13	3.38	1.88	2.75	1.88
16	9.13	42.88	9.75	45.00	9.50	43.00	9.13	43.00
17	5.63	97.00	14.25	96.88	7.50	97.00	5.50	97.00
18	86.75	89.88	89.38	90.00	86.88	89.88	86.75	89.88
19	0.00	36.75	0.13	39.00	0.00	37.13	0.00	37.13
20	10.25	90.38	25.13	89.38	11.75	90.38	11.25	90.38
21	20.38	49.50	29.50	43.63	20.88	49.00	20.88	49.38

In the following comparison experiment, input variable matrix X is composed of process variables [XMEAS (1 : 22)] and 11 manipulated variables [XMEAS (23 : 33)] except XMV (12). The quality variable matrix Y includes XMEAS (35) and XMEAS (38). The model parameters based on the combination of manifold learning algorithm and PLS are set as follows:

- (1) The GLPLS model: $\delta_x = 0.1, \delta_y = 0.8, k_x = 12, k_y = 12$.
- (2) The LPPLS model: $\delta_x = 1.5, \delta_y = 0.8, k_x = 20, k_y = 15$.
- (3) The GPLPLS model: $k_x = 11, k_y = 16$ (mainly refers to the GPLPLS_{xy} model).

Table 9.4 lists the FDR values of different quality-related monitoring methods, corresponding to PLS, CPLS, GLPLS, and GPLPLS models, and the corresponding detection threshold is calculated with confidence level of 99.75%. The last two columns are FDRs calculated based on the PMA value of this data set.

It can be seen from Tables 9.1 and 9.4 that although the data sets are different, the results of PMA are similar. Therefore, the quality-related monitoring results should be similar, and it is obvious that the GPLPLS model gives consistent conclusions. The higher FDR of other models than GPLPLS is due to not good to distinguish

Table 9.4 FDRs comparison for different quality-related methods

IDV	PLS	CPLS	GLPLS	GPLPLS	PMA1	PMA2
1	99.13	96.13	99.75	66.75	0.20	0.68
2	98.00	81.25	97.63	92.75	0.07	0.06
3	0.38	0.50	1.13	0.50	0.77	1.19
4	0.63	0.13	98.88	0.25	0.89	1.02
5	21.88	20.38	21.38	17.63	0.30	1.04
6	99.25	99.25	99.38	96.38	0.00	0.00
7	36.75	35.63	83.63	27.75	0.14	1.03
8	92.50	87.75	93.38	74.88	0.06	0.07
9	0.63	0.38	0.75	0.00	0.90	0.81
10	30.00	28.00	23.13	13.88	0.59	0.81
11	1.38	0.25	53.50	0.38	0.78	0.76
12	87.50	84.75	87.75	75.50	0.04	0.03
13	93.88	85.00	95.25	79.75	0.02	0.02
14	33.50	1.63	96.88	0.00	1.07	0.77
15	0.63	0.75	1.50	0.50	0.90	0.57
16	14.25	12.63	9.00	8.00	0.78	0.53
17	56.00	37.13	96.75	1.63	0.64	0.70
18	88.00	88.00	90.25	86.75	0.01	0.00
19	0.00	0.00	2.50	0.00	0.95	0.75
20	26.63	27.75	36.25	10.25	0.67	0.78
21	29.88	24.50	44.38	8.63	0.23	0.09

whether these faults are quality related. Although GLPLS has similar fusion idea of global feature and local structure, its weak monitoring performance is caused by the inappropriate parameters and model construction. Because it is difficult to select suitable parameters, the parameter determination method is still an open issue.

In summary, GPLPLS model shows good monitoring performance. It is suitable for the combination of global structure and local structure features, so the output prediction results and fault monitoring results of the model are better than other models.

9.7 Conclusions

This chapter proposes a new statistical monitoring model based on the global plus local projection to latent structure (GPLPLS) model. This model not only maintains the global and local structural characteristics of the data, but also pays more attention to the correlation between the extracted principal components. First, the GLPLS method is introduced, and it is pointed out that the model construction of this method is unreasonable, and then the GPLPLS method is proposed to maintain the global and local features with a new structure. Then a monitoring model based on the GPLPLS method is established, and the monitoring performance of the proposed method is verified on the TE process simulation platform. The results show that compared with PLS, CPLS, and GLPLS, GPLPLS method has better process monitoring performance for quality-related fault.

References

- Aumi S, Corbett B, Clarke-Pringle T (2013) Data-driven model predictive quality control of batch processes. *AIChE J* 59:2852–2861
- Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 15:1373
- Ding SX (2014) Data-driven design of monitoring and diagnosis systems for dynamic processes: a review of subspace technique based schemes and some recent results. *J Process Control* 24:431–449
- Doymaz F, Palazoglu A, Romagnoli JA (2003) Orthogonal nonlinear partial least-squares regression. *Ind Eng Chem Res* 42(23):5836–5849
- Ge Z, Song Z, Gao F (2012) Nonlinear quality prediction for multiphase batch processes. *AIChE J* 58:1778–1787
- Godoy JL, Zumoffen DA, Vega JR, Marchetti JL (2014) New contributions to non-linear process monitoring through kernel partial least squares. *Chemom Intell Lab Syst* 135:76–89
- He X, Niyogi P (2003) Locality preserving projections. *Adv Neural Inf Process Syst* 16:186–197
- He X, Yan S, Hu Y, Niyogi P, Zhang HJ (2005) Face recognition using laplacianfaces. *IEEE Trans on Pattern Analysis and Machine Intell* 27:328–340
- Lee JM, Yoo CK, Lee IB (2010) Statistical process monitoring with independent component analysis. *J Process Control* 14:467–485
- Li C, Ye H, Wang G, Zhang J (2005) A recursive nonlinear PLS algorithm for adaptive nonlinear process modeling. *Chem Eng Technol* 28:141–152

- Li G, Qin SJ, Zhou D (2010) Geometric properties of partial least squares for process monitoring. *Automatica* 46:204–210
- Peng K, Zhang K, Dong J (2015) Quality-relevant fault detection and diagnosis for hot strip mill process with multi-specification and multi-batch measurements. *J Franklin Inst* 352:987–1006
- Qin SJ (2010) Statistical process monitoring: basics and beyond. *J Chemom* 17:480–502
- Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis. *Annu Rev Control* 36:220–234
- Qin SJ, McAvoy TJ (1992) Nonlinear PLS modeling using neural networks. *Comput Chem Eng* 16:379–391
- Qin SJ, McAvoy TJ (1996) Nonlinear FIR modeling via a neural net PLS approach. *Comput Chem Eng* 20:147–159
- Rosipal R, Trejo LJ (2001) Kernel partial least squares regression in reproducing kernel hilbert space. *J Mach Learn Res* 2:97–123
- Rosipal R, Trejo LJ (2001) Kernel partial least squares regression in reproducing kernel hilbert space. *J Mach Learn Res* 2(Dec):97–123
- Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
- Severson K, Chaiwatanodom P, Braatz R (2016) Perspectives on process monitoring of industrial systems. *Annu Rev Control* 1:1
- Shan P, Peng S, Tang L, Yang C, Zhao Y, Xie Q, Li C (2015) A nonlinear partial least squares with slice transform based piecewise linear inner relation. *Chemom Intell Lab Syst* 143:97–110
- Shimizu S, Hoyer PO, Kerminen A (2006) A linear non-gaussian acyclic model for causal discovery. *J Mach Learn Res* 7:2003–2030
- Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323
- Wang J, Zhong B, Zhou J (2017) Quality-relevant fault monitoring based on locality preserving partial least squares statistical models. *Ind Eng Chem Res* 56:7009–7020
- Wold S (1992) Nonlinear partial least squares modelling ii. spline inner relation. *Chemom Intell Lab Syst* 14(1–3):71–84
- Wold S, Kettaneh-Wold N, Skagerberg B (1989) Nonlinear PLS modeling. *Chemom Intell Lab Syst* 7:53–65
- Yin S, Ding S, Xie X, Luo H (2014) A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans Industr Electron* 61:6418–6428
- Zhang Y, Qin SJ (2008) Improved nonlinear fault detection technique and statistical analysis. *AIChE J* 54:3207–3220
- Zhang K, Dong J, Peng K (2016) A novel dynamic non-gaussian approach for quality-related fault diagnosis with application to the hot strip mill process. *J Franklin Inst* 354:702–721
- Zhang K, Dong J, Peng K (2017) A novel dynamic non-gaussian approach for quality-related fault diagnosis with application to the hot strip mill process. *J Franklin* 354:702–721
- Zhao C (2014) Quality-relevant fault diagnosis with concurrent phase partition and analysis of relative changes for multiphase batch processes. *Intell Control Autom IEEE*, pp 1372–1377
- Zhao S, Zhang J, Xu Y, Xiong Z (2006) Nonlinear projection to latent structures method and its applications. *Ind Eng Chem Res* 45(11):3843–3852
- Zhong B, Wang J, Zhou J, Wu H, Jin Q (2016) Quality-related statistical process monitoring method based on global and local partial least-squares projection. *Ind Eng Chem Res* 55:1609–1622
- Zhou D, Li G, Qin SJ (2010) Total projection to latent structures for process monitoring. *AIChE J* 56:168–178
- Zhu J, Ge Z, Song Z, Zhou L, Chen G (2017) Large-scale plant-wide process modeling and hierarchical monitoring: a distributed bayesian network approach. *J Process Control*, pp 91–106

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 10

Locality-Preserving Partial Least Squares Regression



This chapter proposes another nonlinear PLS method, named as locality-preserving partial least squares (LPPLS), which embeds the nonlinear degenerative and structure-preserving properties of LPP into the PLS model. The core of LPPLS is to replace the role of PCA in PLS with LPP. When extracting the principal components of \mathbf{t}_i and \mathbf{u}_i , two conditions must satisfy: (1) \mathbf{t}_i and \mathbf{u}_i retain the most information about the local nonlinear structure of their respective data sets. (2) The correlation between \mathbf{t}_i and \mathbf{u}_i is the largest. Finally, a quality-related monitoring strategy is established based on LPPLS.

First, the geometric interpretation of PCA in PLS and LPP is introduced. LPPLS model and LPPLS-based quality-related process monitoring method are proposed. Here three different types of LPPLS models are also given in the same framework, facing three nonlinear cases: nonlinearly correlated in the input space \mathbf{X} or the output space \mathbf{Y} , as well as between them. A typical algorithm for extracting principal components is derived. Then, the feasibility and effectiveness of LPPLS method is verified by artificial 3-D data and Tennessee Eastman Process simulations.

10.1 The Relationship Among PCA, PLS, and LPP

For the normalized data sets of process variables $\mathbf{X} = [\mathbf{x}^T(1), \mathbf{x}^T(2), \dots, \mathbf{x}^T(n)]^T \in R^{n \times m}$ ($\mathbf{x} \in R^{1 \times m}$) and quality variable $\mathbf{Y} = [\mathbf{y}^T(1), \mathbf{y}^T(2), \dots, \mathbf{y}^T(n)]^T \in R^{n \times l}$ ($\mathbf{y} \in R^{1 \times l}$), where m and l are the dimension of the process and quality variables spaces, and n is the number of samples, the principal component extraction of PCA, LPP, and PLS is actually equivalent to the following constrained optimization problem.

$$J_{\text{PCA}}(\mathbf{w}) = \max \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \quad (10.1)$$

$$\text{s.t. } \mathbf{w}^T \mathbf{w} = 1$$

$$J_{\text{LPP}}(\mathbf{w}) = \max \mathbf{w}^T \mathbf{X}^T \mathbf{S}_x \mathbf{X} \mathbf{w} \quad (10.2)$$

$$\text{s.t. } \mathbf{w}^T \mathbf{X}^T \mathbf{D}_x \mathbf{X} \mathbf{w} = 1$$

$$J_{\text{PLS}}(\mathbf{w}, \mathbf{c}) = \max \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} \quad (10.3)$$

$$\text{s.t. } \mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1$$

The meaning of related variables such as \mathbf{w} , \mathbf{c} has been given in Chap. 9. Also, in Chap. 9, to weaken the limitations of PLS's lack of local feature extraction capabilities, the input space \mathbf{X} and the output space \mathbf{Y} , are mapped into a new feature space \mathbf{X}_F and \mathbf{Y}_F that includes a global linear subspace and a plurality of local linear subspaces. Consequently, the following new optimization objective function of the global plus local projection to latent structures (GPLPLS) method is immediately obtained using the feature space \mathbf{X}_F or \mathbf{Y}_F to replace the original space \mathbf{X} or \mathbf{Y} ,

$$J_{\text{GPLPLS}}(\mathbf{w}, \mathbf{c}) = \arg \max \{ \mathbf{w}^T \mathbf{X}_F^T \mathbf{Y}_F \mathbf{c} \} \quad (10.4)$$

$$\text{s.t. } \mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1,$$

where $\mathbf{X}_F = \mathbf{X} + \lambda_x \boldsymbol{\theta}_x^{\frac{1}{2}}$, $\mathbf{Y}_F = \mathbf{Y} + \lambda_y \boldsymbol{\theta}_y^{\frac{1}{2}}$.

Although adding local features to the global features makes the GPLPLS model show excellent performance in fault detection, the GPLPLS model does not fully implement local feature extraction or its local features are only extracted approximately. The main reason is that the constraint condition of the GPLPLS model is still the constraint condition of PCA or PLS. Of course, this combination way generally cannot guarantee the constraints of PCA and LPP at the same time.

Only the nonlinear part of the function is described by the local features, and the linear part is still characterized by the traditional covariance matrix in Chap. 9. In fact, the characteristics of the linear part can also be described by local characteristics. In this way, we can regard the linear part and the nonlinear part as a whole, thereby avoiding unnecessary parameter trade-offs. In the following context, we attempt to analyze the differences and similarities between PCA and LPP.

The local characteristics of \mathbf{X} of LPP are contained in the matrices $\mathbf{X}^T \mathbf{S}_x \mathbf{X}$ and $\mathbf{X}^T \mathbf{D}_x \mathbf{X}$. To study the similarity of LPP and PCA, the matrix \mathbf{S}_x and \mathbf{D}_x are decomposed into $\mathbf{S}_x^{\frac{1}{2}T} \mathbf{S}_x^{\frac{1}{2}}$ and $\mathbf{D}_x^{\frac{1}{2}T} \mathbf{D}_x^{\frac{1}{2}}$, respectively. Then LPP criteria (10.2) is further transformed as

$$J_{\text{LPP}}(\mathbf{w}) = \max \mathbf{w}^T \mathbf{X}_M^T \mathbf{X}_M \mathbf{w} \quad (10.5)$$

$$\text{s.t. } \mathbf{w}^T \mathbf{M}_x^T \mathbf{M}_x \mathbf{w} = 1,$$

where $\mathbf{M}_x = \mathbf{D}_x^{\frac{1}{2}} \mathbf{X}$, $\mathbf{X}_M = \mathbf{S}_x^{\frac{1}{2}} \mathbf{X}$.

Comparing (10.5) and (10.1), it can be found that the structure in the mathematical description of the optimization problem of LPP and PCA is similar. "PCA selects

a subspace consisting of the eigenvectors corresponding to the largest eigenvalues of the global covariance matrix, while LPP selects a subspace consisting of the eigenvectors corresponding to the smallest eigenvalues of the local covariance matrix (He et al. 2005)". Therefore, LPP can replace PCA in the PLS decomposition process, thus achieving the preservation of strong local nonlinearity.

PCA is used to extract a set of components that transforms the original data X to a set of t-scores T in the PLS criteria (10.3) of forming latent variables. PCA and PLS only extract global linear features and therefore do not reflect the local information of the sample and its nonlinear features. Actually PCA is not the only method of extracting principle components. LPP, converting the global nonlinearity into a combination of multiple local linearities, also can be used for extracting principle components. Therefore, LPP is suitable for systems with strong local nonlinear features.

10.2 LPPLS Models and LPPLS-Based Fault Detection

10.2.1 The LPPLS Models

Based on (10.3), the two criteria for selecting latent vectors u_i and t_i for PLS are as follows:

- (1) The linear variation on latent vectors is manifested as much as possible;
- (2) The correlation between is as strong as possible.

The optimization objective for extracting the first component pairs (t_1, u_1) is

$$\begin{aligned} J_{\text{PLS}}(\mathbf{w}_1, \mathbf{c}_1) &= \max \mathbf{w}_1^T \mathbf{X}^T \mathbf{Y} \mathbf{c}_1 \\ \text{s.t. } &\mathbf{w}_1^T \mathbf{w}_1 = 1, \mathbf{c}_1^T \mathbf{c}_1 = 1. \end{aligned} \quad (10.6)$$

The optimization objective (10.6) is used for fast extraction of principal components in PLS. Define $\mathbf{E}_0 = \mathbf{X}$, $\mathbf{F}_0 = \mathbf{Y}$, then the latent variables t_1 and c_1 are calculated by $t_1 = \mathbf{E}_0 \mathbf{w}_1$ and $u_1 = \mathbf{F}_0 \mathbf{c}_1$, where \mathbf{c}_1 and \mathbf{w}_1 are the eigenvectors corresponding to the maximum eigenvalues of the following matrices.

$$\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0 \mathbf{w}_1 = \theta_1^2 \mathbf{w}_1 \quad (10.7)$$

$$\mathbf{F}_0^T \mathbf{E}_0 \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 = \theta_1^2 \mathbf{c}_1. \quad (10.8)$$

Considering the similarity between LPP and PCA discussed in the previous section, LPP is used to extract the principle components (10.3) in PLS decomposition instead of PCA, i.e., the LPPLS model. Three LPPLS models (types I, II, and III) are developed to address the different nonlinear relationships.

The type I LPPLS model is given to deal with this case where the input space X has a nonlinear relationship and the correlation between the input X and the output Y is linear. The principal components of the input space X of the type I LPPLS are extracted by LPP and the principal components of the output space Y are extracted by PCA. The optimization objectives are as follows:

$$\begin{aligned} J_{\text{LPPLS}_I}(\mathbf{w}, \mathbf{c}) &= \max \mathbf{w}^T \mathbf{X}_M^T \mathbf{Y} \mathbf{c} \\ \text{s.t. } \mathbf{c}^T \mathbf{c} &= 1, \mathbf{w}^T \mathbf{M}_x^T \mathbf{M}_x \mathbf{w} = 1. \end{aligned} \quad (10.9)$$

The type II LPPLS model is given to deal with the nonlinearly correlation between the input space X and output space Y , but linearly correlation in the input space X . The principal components in input space X are extracted by PCA and the principal components of the output space Y are extracted by LPP. The optimization function is

$$\begin{aligned} J_{\text{LPPLS}_{II}}(\mathbf{w}, \mathbf{c}) &= \max \mathbf{w}^T \mathbf{X}^T \mathbf{Y}_M \mathbf{c} \\ \text{s.t. } \mathbf{w}^T \mathbf{w} &= 1, \mathbf{c}^T \mathbf{M}_y^T \mathbf{M}_y \mathbf{c} = 1 \end{aligned} \quad (10.10)$$

in which

$$\begin{aligned} \mathbf{Y}_M &= \mathbf{S}_y^{\frac{1}{2}} \mathbf{Y}, \mathbf{S}_y = \mathbf{S}_y^{\frac{1}{2}T} \mathbf{S}_y^{\frac{1}{2}} \\ \mathbf{M}_y &= \mathbf{D}_y^{\frac{1}{2}} \mathbf{Y}, \mathbf{D}_y = \mathbf{D}_y^{\frac{1}{2}T} \mathbf{D}_y^{\frac{1}{2}} \end{aligned}$$

where \mathbf{S}_y and \mathbf{D}_y are similar as the \mathbf{S}_x and \mathbf{D}_x and it has a different neighbors parameter δ_y in (9.8).

The type III LPPLS model is given for the nonlinear correlation between the input space X and the output space Y as well as among the input spaces X . In this case, the principal components of the input space X and output space Y are both extracted by the LPP. Its corresponding optimization objective function is

$$\begin{aligned} J_{\text{LPPLS}_{III}}(\mathbf{w}, \mathbf{c}) &= \max \mathbf{w}^T \mathbf{X}_M^T \mathbf{Y}_M \mathbf{c} \\ \text{s.t. } \mathbf{w}^T \mathbf{M}_x^T \mathbf{M}_x \mathbf{w} &= 1, \mathbf{c}^T \mathbf{M}_y^T \mathbf{M}_y \mathbf{c} = 1. \end{aligned} \quad (10.11)$$

The criteria for the selection of latent vectors \mathbf{u}_i and \mathbf{t}_i for type III LPPLS are as follows:

- (1) The nonlinear variation on the latent vector is manifested as much as possible;
- (2) The correlation between latent vectors is as strong as possible.

Discussion one of the aims of is to choose factors \mathbf{u}_i and \mathbf{t}_i that better represent the nonlinear variation of the factor changes. GLPLS's optimization objective is given in (10.12) (Zhong et al. 2016).

$$\begin{aligned} J_{\text{GLPLS}}(\mathbf{w}, \mathbf{c}) &= \max \{ \mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c} + \beta_1 \mathbf{w}^T \mathbf{X}_M^T \mathbf{X}_M \mathbf{w} + \beta_2 \mathbf{c}^T \mathbf{Y}_M^T \mathbf{Y}_M \mathbf{c} \} \\ \text{s.t. } \mathbf{w}^T \mathbf{w} &= 1, \mathbf{c}^T \mathbf{c} = 1, \end{aligned} \quad (10.12)$$

where the parameters β_1 and β_2 are the trade-off between global and local feature extraction. Here the embedding properties and data screening of LPP are removed because the constraints $\mathbf{w}^T \mathbf{X}^T \mathbf{D}_x \mathbf{X} \mathbf{w} = 1$ and $\mathbf{c}^T \mathbf{Y}^T \mathbf{D}_y \mathbf{Y} \mathbf{c} = 1$ of LPP are removed in (10.12). GLPLS model is a fusion of the PLS model with the partial LPP model. “The best vectors \mathbf{w} and \mathbf{c} from (10.12) ensure maximum correlation (PLS) and relative or local optimal data filtering and embedding capabilities for \mathbf{X} and \mathbf{Y} (Zhong et al. 2016)”. On the other hand, $\mathbf{w}^T \mathbf{X}^T \mathbf{S}_x \mathbf{X} \mathbf{w}$ and $\mathbf{c}^T \mathbf{Y}^T \mathbf{S}_y \mathbf{Y} \mathbf{c}$ are only used to introduce the local features in the input and output space, but not the correlation features between them. However, the LPP model is fully embedded in the LPPLS model. It is embedded in the outer layer, inner layer or both of the PLS model, i.e., three types of LPPLS models. At the same time, the correlation information in the input and output spaces is retained.

Type III LPPLS is used as an example to show the extracting of principal components. Supposed the first component pairs is $(\mathbf{t}_1, \mathbf{u}_1)$. Define $\mathbf{E}_{0L} = \mathbf{X}_M$ and $\mathbf{F}_{0L} = \mathbf{Y}_M$ in order to facilitate comparison with the traditional linear PLS.

First, the optimization (10.11) for the first component pair $(\mathbf{t}_1, \mathbf{u}_1)$ is converted into an unconstrained problem by the Lagrangian multiplier,

$$\Psi(\mathbf{w}_1, \mathbf{c}_1) = \mathbf{w}_1^T \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 - \lambda_1 (\mathbf{w}_1^T \mathbf{M}_x^T \mathbf{M}_x \mathbf{w}_1 - 1) - \lambda_2 (\mathbf{c}_1^T \mathbf{N}_y^T \mathbf{N}_y \mathbf{c}_1 - 1). \quad (10.13)$$

Let $\frac{\partial \Psi}{\partial \mathbf{w}_1} = 0$ and $\frac{\partial \Psi}{\partial \mathbf{c}_1} = 0$, then the optimal pair of \mathbf{w}_1 and \mathbf{c}_1 is obtained

$$\mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 = 2\lambda_1 \mathbf{M}_x^T \mathbf{M}_x \mathbf{w}_1 \quad (10.14)$$

$$\mathbf{F}_{0L}^T \mathbf{E}_{0L} \mathbf{w}_1 = 2\lambda_2 \mathbf{N}_y^T \mathbf{N}_y \mathbf{c}_1. \quad (10.15)$$

Equations (10.14) and (10.15) are respectively multiplied by \mathbf{w}_1^T and \mathbf{c}_1^T on the left, then,

$$\theta_1 := 2\lambda_1 = 2\lambda_2 = \mathbf{w}_1^T \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 = \mathbf{c}_1^T \mathbf{F}_{0L}^T \mathbf{E}_{0L} \mathbf{w}_1. \quad (10.16)$$

Comparing (10.11) and (10.16), it is found that θ_1 is the objective function value. Substitute (10.16) into (10.14) and (10.15), and the relationship between \mathbf{w}_1 and \mathbf{c}_1 is obtained,

$$\mathbf{w}_1 = \frac{1}{\theta_1} (\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 \quad (10.17)$$

$$\mathbf{c}_1 = \frac{1}{\theta_1} (\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L} \mathbf{w}_1. \quad (10.18)$$

Substitute (10.18) into (10.14) and substitute (10.17) into (10.15), the following equations about the first vector pair are obtained,

$$(\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} (\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L} \mathbf{w}_1 = \theta_1^2 \mathbf{w}_1 \quad (10.19)$$

$$(\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L} (\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 = \theta_1^2 \mathbf{c}_1. \quad (10.20)$$

The optimal weight vectors \mathbf{w}_1 and \mathbf{c}_1 is obtained by the maximum eigenvalue of (10.19) and (10.20). Now the potential variables \mathbf{u}_1 and \mathbf{t}_1 are calculated as follows:

$$\mathbf{t}_1 = \mathbf{E}_{0L} \mathbf{w}_1, \quad \mathbf{u}_1 = \mathbf{F}_{0L} \mathbf{c}_1.$$

Calculation of the load vector:

$$\mathbf{p}_1 = \frac{\mathbf{E}_{0L}^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2}, \quad \bar{\mathbf{q}}_1 = \frac{\mathbf{F}_{0L}^T \mathbf{t}_1}{\|\mathbf{t}_1\|^2}.$$

Residual matrixes \mathbf{E}_{1L} and \mathbf{F}_{1L} are

$$\mathbf{E}_{1L} = \mathbf{E}_{0L} - \mathbf{t}_1 \mathbf{p}_1^T, \quad \mathbf{F}_{1L} = \mathbf{F}_{0L} - \mathbf{u}_1 \bar{\mathbf{q}}_1^T.$$

The first optimal weight vector \mathbf{w}_1 of PLS (10.7) is the eigenvectors of matrix $\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0$, while in LPPLS (10.19), it is corresponding to the eigenvectors of matrix $(\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} (\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L}$. The optimization problem with maximum eigenvalue in (10.19) are very similar to the traditional linear PLS. Therefore, the traditional NIPALS technique is convenient to extract the remaining principle components.

The other latent variables are calculated based on the residual matrixes \mathbf{E}_{iL} and \mathbf{F}_{iL} , $i = 1, 2, \dots, d - 1$.

$$\mathbf{t}_{i+1} = \mathbf{E}_{iL} \mathbf{w}_{i+1}, \quad \mathbf{u}_{i+1} = \mathbf{F}_{iL} \mathbf{c}_{i+1},$$

where \mathbf{w}_{i+1} is the eigenvector corresponding to the maximum eigenvalue θ_{i+1}^2 of matrix $(\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{iL}^T \mathbf{F}_{iL} (\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{iL}^T \mathbf{E}_{iL}$.

Similarly, \mathbf{c}_{i+1} is the eigenvector corresponding to the maximum eigenvalue of $(\mathbf{N}_y^T \mathbf{N}_y)^{-1} \mathbf{F}_{iL}^T \mathbf{E}_{iL} (\mathbf{M}_x^T \mathbf{M}_x)^{-1} \mathbf{E}_{iL}^T \mathbf{F}_{iL}$. Then,

$$\mathbf{p}_{i+1} = \frac{\mathbf{E}_{iL}^T \mathbf{t}_{i+1}}{\|\mathbf{t}_{i+1}\|^2}, \quad \bar{\mathbf{q}}_{i+1} = \frac{\mathbf{F}_{iL}^T \mathbf{t}_{i+1}}{\|\mathbf{t}_{i+1}\|^2}.$$

Finally, d latent variables of LPPLS are determined using the cross-validation method.

10.2.2 LPPLS for Process and Quality Monitoring

X and Y is projected to a low-dimensional space by latent variables (t_1, \dots, t_d) . The neighboring mapping of original data E_{0L} and F_{0L} is decomposed as follows:

$$\begin{aligned} E_{0L} &= \sum_{i=1}^d t_i p_i^T + E = T P^T + \bar{E} \\ F_{0L} &= \sum_{i=1}^d t_i q_i^T + F = T \bar{Q}^T + \bar{F}, \end{aligned} \quad (10.21)$$

where $T = [t_1, t_2, \dots, t_d]$ are the latent score vectors. $P = [p_1, \dots, p_d]$ and $\bar{Q} = [\bar{q}_1, \dots, \bar{q}_d]$ are load matrices for E_{0L} and F_{0L} , respectively. T is represented by the neighboring mapping data E_{0L} ,

$$T = E_{0L} R = S_x^{\frac{1}{2}} E_0 R, \quad (10.22)$$

where $R = [r_1, \dots, r_d]$,

$$r_i = \prod_{j=1}^{i-1} (I_n - w_j p_j^T) w_i$$

Similarly as GPLPLS method, (10.21) and (10.22) are difficult to apply in practice since the locality transformation matrix S cannot be obtained during the online measurements. So they are changed to the direct decomposition of E_0 and F_0 ,

$$E_0 = S_x^{-\frac{1}{2}} (T P^T + \bar{E}) = T_0 P^T + E' \quad (10.23)$$

$$F_0 = S_y^{-\frac{1}{2}} (S_x^{\frac{1}{2}} T_0 \bar{Q}^T + \bar{F}), \quad (10.24)$$

where $T_0 = E_0 R$, $E' = S_x^{-\frac{1}{2}} \bar{E}$.

Process and quality monitoring for new scaled and mean-centered data samples x and y is performed by the oblique projection of the input data x .

$$\begin{aligned} x &= \hat{x} + x_e \\ \hat{x} &= R P^T x \\ x_e &= (I - P R^T) x. \end{aligned} \quad (10.25)$$

The residual space still contains much variation information (Qin and Zheng 2012), but it is not the main focus of LPPLS. To facilitate the comparison with traditional monitoring methods, this chapter will directly adopt traditional fault monitoring indices without any modification. The T^2 and Q statistics are defined,

$$\begin{aligned}
\mathbf{t} &= \mathbf{R}^T \mathbf{x} \\
\mathbf{T}^2 &= \mathbf{t}^T \mathbf{A}^{-1} \mathbf{t} = \mathbf{t}^T \left(\frac{1}{n-1} \mathbf{T}_0^T \mathbf{T}_0 \right)^{-1} \mathbf{t} \\
\mathbf{Q} &= \|\mathbf{x}_e\|^2 = \mathbf{x}^T (\mathbf{I} - \mathbf{P} \mathbf{R}^T) \mathbf{x},
\end{aligned} \tag{10.26}$$

where \mathbf{A} is the sample covariance matrix. The matrix $\tilde{\mathbf{X}}$ or \mathbf{E}_{0L} of type III LPPLS is not a scaled and mean-centered one. Moreover in nonlinear systems, the output variables may not obey the Gaussian distribution even if the input variables obey it. So the control limits of the statistics of \mathbf{T}^2 and \mathbf{Q} are not computed according to the F and χ^2 distributions. It should be calculated based on their probability density functions obtained by non-parametric kernel density estimation method (Lee et al. 2004).

Remark 10.1 The LPPLS decomposition (10.23) is similar to linear PLS, but its residual space \mathbf{E}' is related to the locally preserved projection matrix $\mathbf{S}_x^{\frac{1}{2}}$. It is difficult to obtain the locally retained projection matrix $\mathbf{S}_x^{\frac{1}{2}}$ for new data during online fault detection. But its covariance matrix \mathbf{A} of the samples and the statistics of \mathbf{T}^2 and \mathbf{Q} (10.26) are not directly related to the locally retained projection matrix $\mathbf{S}_x^{\frac{1}{2}}$ which is a useful feature for online monitoring

Although matrix $\mathbf{S}_L := \mathbf{S}_y^{-\frac{1}{2}} \mathbf{S}_x^{\frac{1}{2}} \in \mathbb{R}^{n \times n}$ is constant, the regression equation (10.24) cannot be used for output projections. As mentioned above, the first reason is that the locally preserved projection matrices $\mathbf{S}_x^{\frac{1}{2}}$ and $\mathbf{S}_y^{\frac{1}{2}}$ for the new data are difficult to obtain. Another is that direct application of least squares solution $\mathbf{S}_R = \mathbf{E}_0^+ \mathbf{S}_L \mathbf{E}_0$ may lead to poor prediction performance. The prediction performance directly determines whether a model needs to be updated in practice. The regression equation can be constructed based on \mathbf{F}_0 and \mathbf{T}_0 based on (10.23),

$$\mathbf{F}_0 = \mathbf{T}_0 \mathbf{Q}^T + \tilde{\mathbf{F}}. \tag{10.27}$$

Remark 10.2 In the special case of $\mathbf{S}_L = \mathbf{I}$, (10.24) and (10.27) are equal. In most cases, the regression coefficients ($\tilde{\mathbf{Q}}$ and \mathbf{Q}) are significantly different. But considering both $\tilde{\mathbf{Q}}$ and \mathbf{Q} are least squares solutions for any type of regression equation, so the regression errors $\tilde{\mathbf{F}}$ and \mathbf{F} are equivalent in theory. Therefore, the latter regression equation (10.27) can be used to predicts the corresponding output of the new input data.

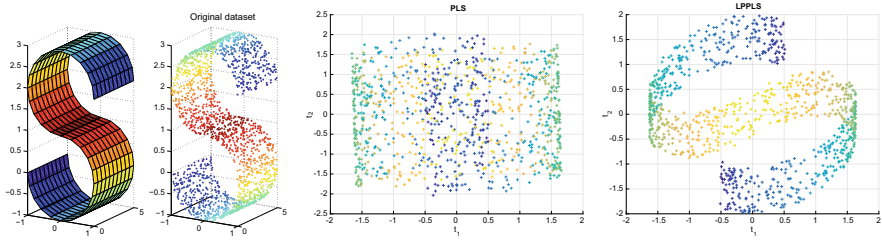


Fig. 10.1 Projection results of PLS, and LPPLS models for S-curve data set with $Y = 2x_1 - x_3$. Type I LPPLS model is used

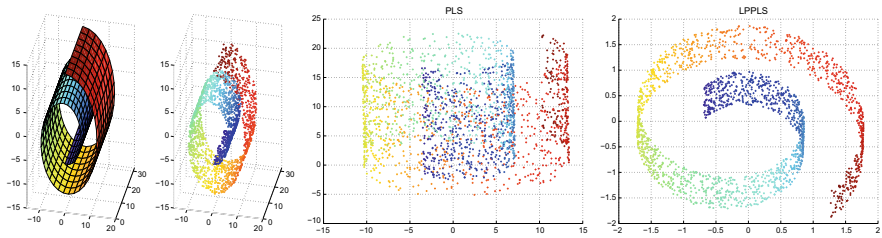


Fig. 10.2 Projection results of PLS, and LPPLS models for Swiss roll data set with $Y = x_1x_3$. Type III LPPLS model is used

10.2.3 Locality-Preserving Capacity Analysis

Here two three-dimensional artificial data sets are used to explain the locality-preserving capacity of LPPLS, S-curves and Swiss roll. They are common to validate the performance of manifold learning algorithm.

$$\begin{aligned}
 X_1 &= [x_1; x_2; x_3] \\
 &= [\cos(\alpha), -\cos(\alpha)]; 5v_1; [\sin(\alpha), 2 - \sin(\alpha)] \\
 X_2 &= [x_1; x_2; x_3] \\
 &= [t \cos(t); 2v_3; t \sin(t)],
 \end{aligned}$$

where $\alpha = (1.5v_2 - 1)/\pi$, $t = 3\pi/2(1 + 2v_4)$. v_1, v_2, v_3 and v_4 are uniformly distributed on $(0, 1)$. Two kinds of output function is defined as $y = 2x_1 - x_3$ (linear) and $y = x_1x_3$ (nonlinear).

1000 sample points are randomly generated in the 3-D space $[x_1, x_2, x_3]$, and the dimensionality reduction process for PLS and LPPLS model is performed. The projection results of the two models in two dimensions are shown in Figs. 10.1 and 10.2, respectively.

The projection results show that PLS does not preserve the local structural information for the S-curves and Swiss roll. In other words, the data is not correctly classified by color. However, LPPLS preserves the local structural features and has

good classification results. LPPLS model improves the local preserving capability of PLS model; moreover, LPPLS can better discriminate the boundary features. Thus, LPPLS method can be used to detect faults related to output variables in systems with strong nonlinearity.

10.3 Case Study

Validation of the proposed LPPLS-based fault detection method is performed on the Tennessee Eastman Process simulation platform (Lyman and Georgakis 1995). TEP is described in detail in the article found in (Lee et al. 2006). The related data sets are downloaded from “<http://web.mit.edu/braatzgroup/links.html>”. PCA (Dunia and Qin 1998; Good et al. 2010) and other global-local preserving projections methods (Luo 2014; Bao et al. 2016; Luo et al. 2016) did not merge any information in the output space, so only the LPPLS method and two quality-related monitoring methods (PLS method and GLPLS method) are compared.

10.3.1 PLS, GLPLS and LPPLS Models

The input variable matrix $X = [x_1, x_2, \dots, x_{33}]^T$ consists of 22 process variables (XMEAS(1:22):= $x_1 : x_{22}$) and 11 manipulated variables ($x_{23} : x_{33}$) except XMV(12). The quality variable matrix $Y = [y_1; y_2]$ is composed of the components G of stream 9 and the components E of stream 11, i.e., XMEAS (35) (y_1) and (38) (y_2). The training set is the normal data IDV(0) containing 960 samples. The test set is the fault data IDV(1:21). Each fault data have 960 samples (the first 160 samples are normal and the last 800 samples are faulty). The model parameters are $\delta_x = 1.5$, $\delta_y = 0.8$, $K_x = 20$ and $K_y = 15$, where K_x and K_y are the adjacent parameters in the input space and output space, respectively. Regression coefficients obtained by PLS, GLPLS, and LPPLS models are shown in Table 10.1. The relative errors of training are shown in Fig. 10.3. Here the relative error is calculated as error = $(y_i - y_{i,tr})/y_i$, $i = 1, 2$ and $y_{i,tr}$ is the corresponding output of the training model.

The training error in Fig. 10.3 shows that the training results of the PLS, GLPLS, and LPPLS models satisfy the modeling requirements. The output prediction experiments of these models are finished under all the fault conditions (i.e., the test data set), and similar prediction abilities are obtained for most cases. Give fault IDV(21) as an example, the output prediction of three models are shown in Fig. 10.4. y_1 and y_2 are at the top and bottom of these figures, respectively. Fault IDV(21) is caused by a slow drift in the output variables to drift slowly (Lee et al. 2006), but the prediction performances of three methods still are good even in this fault case. So the generalization capability of three models is verified.

Table 10.1 Regression coefficients of PLS, GLPLS, and LPPLS models

	PLS		GLPLS		LPPLS	
	y1	y2	y1	y2	y1	y2
	18.1489	-0.7162	13.6677	-3.0777	212.8754	-77.0014
x ₁	-0.0593	0.0855	0.0387	0.0392	-0.0496	0.0932
x ₂	0.0000	0.0000	0.0001	0.0000	-0.0001	0.0000
x ₃	-0.0001	0.0000	0.0000	0.0000	-0.0001	0.0000
x ₄	0.0261	-0.0149	0.1011	-0.0058	0.0271	-0.0182
x ₅	-0.0055	0.0015	0.0058	0.0046	0.0000	0.0030
x ₆	0.0003	-0.0009	0.0041	0.0000	-0.0056	-0.0007
x ₇	-0.0009	0.0000	-0.0009	0.0003	-0.0002	-0.0003
x ₈	-0.0013	0.0000	-0.0125	0.0003	-0.0061	-0.0003
x ₉	-0.0656	0.0229	-0.1396	-0.0028	-0.1016	0.0447
x ₁₀	-0.0946	0.0128	-0.0293	0.0440	-0.4048	0.0257
x ₁₁	0.0223	-0.0027	0.0240	-0.0007	0.0296	0.0000
x ₁₂	-0.0009	0.0002	-0.0008	-0.0008	-4.0733	1.5519
x ₁₃	-0.0009	0.0000	-0.0005	0.0002	0.0003	-0.0001
x ₁₄	0.0005	0.0002	0.0018	-0.0001	0.0000	0.0001
x ₁₅	0.0007	-0.0004	-0.0004	0.0001	-0.8701	0.5530
x ₁₆	-0.0011	0.0001	-0.0009	0.0004	-0.0031	0.0002
x ₁₇	0.0007	0.0000	0.0016	-0.0001	-0.2341	0.0077
x ₁₈	0.0101	-0.0051	-0.0220	0.0039	-0.0251	-0.0167
x ₁₉	0.0001	-0.0001	0.0005	0.0000	0.0001	-0.0001
x ₂₀	-0.0001	-0.0020	0.0076	-0.0025	-0.0005	-0.0012
x ₂₁	0.0145	0.0035	0.0949	0.0218	-0.0094	0.0074
x ₂₂	0.0044	-0.0036	0.0152	0.0026	-0.0033	-0.0054
x ₂₃	-0.0043	0.0008	0.0017	0.0069	-0.0047	0.0010
x ₂₄	-0.0040	-0.0019	0.0106	-0.0030	-0.0044	-0.0024
x ₂₅	-0.0006	0.0009	-0.0001	0.0005	0.0001	0.0005
x ₂₆	-0.0003	-0.0002	0.0000	0.0008	0.0006	-0.0003
x ₂₇	-0.0053	-0.0039	-0.0095	-0.0027	-0.0146	-0.0042
x ₂₈	-0.0007	0.0003	-0.0034	-0.0003	0.0011	0.0002
x ₂₉	-0.0003	0.0001	-0.0003	-0.0003	1.3836	-0.5273
x ₃₀	0.0003	-0.0002	-0.0002	0.0000	0.3753	-0.2391
x ₃₁	0.0004	-0.0005	-0.0004	-0.0001	0.0054	0.0007
x ₃₂	-0.0017	-0.0004	0.0046	-0.0017	0.0022	-0.0010
x ₃₃	-0.0007	0.0002	0.0001	-0.0001	-0.0990	0.0037

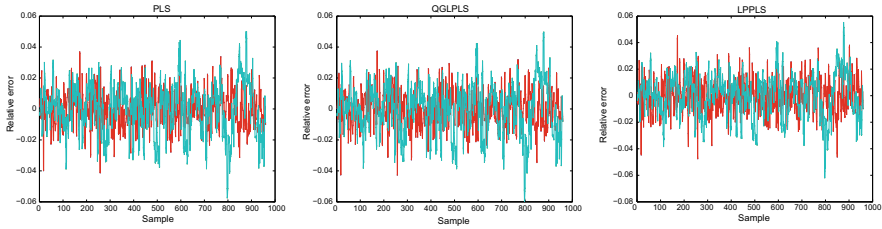


Fig. 10.3 Relative errors of PLS, GLPLS, and LPPLS models

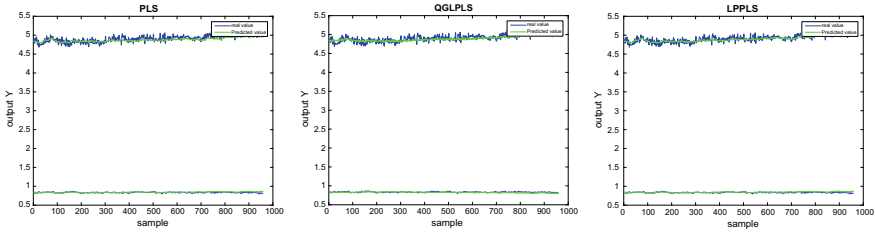


Fig. 10.4 Prediction results for IDV(21) of PLS, GLPLS, and LPPLS models

10.3.2 Quality Monitoring Analysis

The T^2 statistic represents the mapping between process variables and quality variables for PLS and its related methods. The alarm in T^2 statistic indicates a quality-related fault. In contrast, the Q statistic represents only the residuals in the input space, therefore, its alarm indicates that the fault is not quality related. Table 10.2 gives the monitoring FDR whose control limits are calculated with confidence level 99.75%, respectively.

The product quality consists of component G (XMEAS(35)) and component E (XMEAS(38)). Faults IDV(3,4,9,11,14,15,19) have almost no effect on product quality, but the remaining faults cause significant changes in the quality variables. The FDR results of the LPPLS method match the above actual TPE case, which detects quality-related faults with much higher accuracy than the PLS and GLPLS models (e.g., IDV(5) and IDV(12) in Table 10.2). In this section, the performance for fault detection is further examined based on three fault scenarios, including disturbance of reactor cooling water, disturbance of condenser cooling water, and a constant position of the steam 4 valves.

Experiment 1: Disturbance in Reactor Cooling Water (Quality-Independent Fault)

The faults related to the reactor cooling water are IDV (4), IDV (11), and IDV (14). As mentioned above, they have little effect on the product quality but are process related. The results of monitoring the variation of the reactor cooling water are shown in Fig. 10.5. Here IDV (14) is given for example in order to compare with other quality-related methods, such as GPLPLS given in Chap. 9.

Table 10.2 FDR of PLS, GLPLS, and LPPLS models

	PLS		GLPLS		LPPLS	
	T ²	Q	T ²	Q	T ²	Q
IDV(1)	99.13	99.38	99.75	99.38	98.63	99.38
IDV(2)	98.00	98.25	97.63	98.25	98.13	97.88
IDV(3)	0.38	0.13	1.13	0.63	0.50	0.13
IDV(4)	0.63	86.00	98.88	67.00	0.25	85.88
IDV(5)	21.88	16.00	21.38	22.25	99.63	100.00
IDV(6)	99.25	100.00	99.38	100.00	100.00	100.00
IDV(7)	36.75	100.00	83.63	100.00	37.63	100.00
IDV(8)	92.50	94.00	93.38	97.13	92.25	94.75
IDV(9)	0.63	0.50	0.75	0.88	0.63	0.50
IDV(10)	30.00	4.38	23.13	26.63	49.00	31.00
IDV(11)	1.38	57.88	53.50	52.50	2.88	59.00
IDV(12)	87.50	91.00	87.75	97.88	95.50	97.50
IDV(13)	93.88	93.00	95.25	94.25	94.13	93.88
IDV(14)	33.50	100.00	96.88	99.88	2.50	100.00
IDV(15)	0.63	0.38	1.50	0.88	0.75	0.25
IDV(16)	14.25	3.13	9.00	12.50	53.38	38.75
IDV(17)	56.00	85.38	96.75	85.25	52.75	86.50
IDV(18)	88.00	89.25	90.25	89.88	87.88	89.25
IDV(19)	0.00	4.13	2.50	1.63	3.25	7.88
IDV(20)	26.63	34.00	36.25	35.38	28.13	34.00
IDV(21)	29.88	39.75	44.38	33.75	42.38	38.63

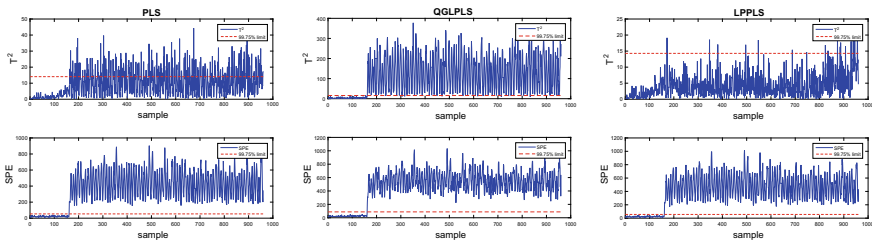


Fig. 10.5 PLS, GLPLS, and LPPLS monitoring for IDV(14)

The faults related to the reactor cooling water will cause the variation of reactor temperature, but the reactor temperature is controlled by a cascade controller. So any disturbances, including step fault IDV(4), random fault IDV(11), and valve sticking disturbances IDV(14), do not affect the product quality. Table 10.2 shows the fault detection rates for the PLS, GLPLS, and LPPLS methods. The Q statistics of all three methods detect these process-related faults in the input space with higher FDR. The FDR values for LPPLS for the T^2 statistic are much smaller than other methods, which indicates that these faults are quality-independent. Fault IDV(14) is a special case. When the traditional analysis methods, such as filtering or PLS, are applied to this fault, most information about the fault feature are lost. This leads to this fault is difficult to detect in the input space, thus preventing it from detecting the fault in the input space. Now Let's check the detection result for fault IDV(14). FDR in the T^2 statistic for PLS and GLPLS model are 33.5% and 96.88%, far higher than LPPLS. It means that PLS and GLPLS distinguish fault IDV(14) as quality related. The FDR of LPPLS in T^2 statistic is 2.5%, near to that of GPLPLS (Tables 9.2 and 9.3). So LPPLS can effectively filter the quality-irrelevant faults, similar as GPLPLS method.

Experiment 2: Disturbance in Condenser Cooling Water (Quality-Related Fault)

These faults include the quality-related faults IDV (5) and IDV (12). The fault IDV (5) is caused by a step change in the cooling water flow rate of the condenser. Since the series controller compensates for this step change, the separator temperature returns to setpoint. The PLS and GLPLS have similar predicted results, returning to the setpoint 10h after the fault. But LPPLS-based monitoring provides a persistent alarm in statistic (T^2) (Fig. 10.6). "The persistence of the fault detection statistic is demonstrated by the fact that it continues to alert the operator to process anomalies even though all process variables appear to have returned to their normal values, especially important in quality-related process fault detection (Lee et al. 2006)". In fact, the disturbance in condenser cooling water, such as its flow rate, always affects the output quality. It should be pointed that the cooling water flow rate of the condenser plays an important role both in the output quality and the safety of the chemical plant. This fault cannot be eliminated by the series controller and should be alarming. Although the controller can compensate the variations caused by this fault, the process-related monitoring in Q statistic, (Fig. 10.6), provides a consistent alarm. Experimental results show that the PLS and GLPLS models do not actually capture the source of the fault, while LPPLS does.

Experiment 3: Constant Position in Valve of Steam 4

Fault IDV (21) due to the slow output drift has been little studied. The sensitivity of fault detection is related to the magnitude of the mass drift. Therefore, fast detection of fault IDV(21) is beneficial for quality control. The process monitoring results are shown in Fig. 10.7. For GLPLS, LPPLS, and PLS, this fault is fully detected as quality-related after about 650, 720, and 780 samples, respectively. LPPLS and GLPLS detect the fault IDV(21) faster than PLS method.

The following conclusions are drawn from the above experiments.

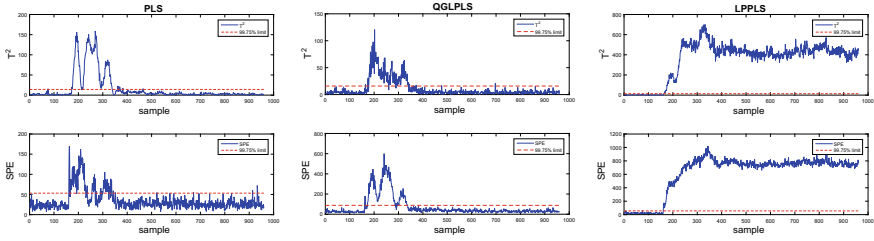


Fig. 10.6 PLS, GLPLS, and LPPLS monitoring for IDV(5)

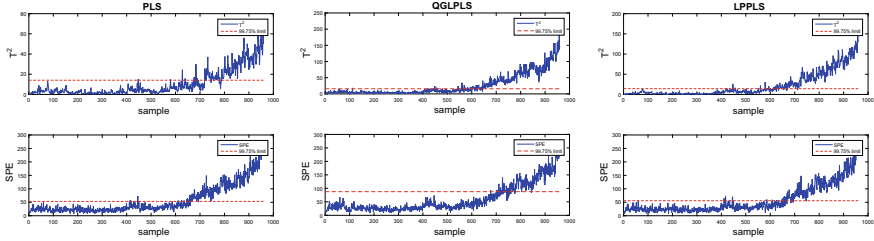


Fig. 10.7 PLS, GLPLS, and LPPLS monitoring for IDV(21)

- PLS is a linear model, so it cannot accurately identify some faults for the strong nonlinear systems.
- GLPLS and LPPLS shows better extracting for nonlinear correlation by introducing the locality-preserving ability of LPP strategy.
- GLPLS aims at preserving the local features in the input space and output space, but lacks the correlation between them. GLPLS is actually a linear PLS plus partial locality preserving, in which the role of LPP is not fully reflected. This may lead to the false detection or missed detection in fault detection.
- LPPLS makes full use of the LPP algorithm to achieve local nonlinear structure preservation. It decomposes the global nonlinear problem into a combination of multiple local linear problems by introducing local structure information. Therefore, LPPLS establishes an more effective model for the nonlinear correlation between the input space and the output space compared with GLPLS.

10.4 Conclusions

In this chapter, the LPPLS statistical model is proposed and the LPPLS-based quality-related fault detection and prediction is given. LPPLS not only retains the local information of the original data, but also maintains the correlation between X and Y to the maximum extent, thus achieving accurate prediction of quality variables. The LPPLS encapsulates the excellent detection performance for locally nonlinear

systems, due to the local feature extraction ability controlled by two parameters, δ_x and δ_y . Experiment results on the artificial three-dimensional data sets, S-curve and Swiss roll, show that LPPLS maintains local structural features well. The experiment results on TEP simulator show that LPPLS extracts the local nonlinear features more effectively and has better fault detection performance than PLS and GLPLS models.

References

- Bao S, Luo L, Mao J, Tang D (2016) Improved fault detection and diagnosis using sparse global-local preserving projections. *J Process Control* 47:121–135
- Dunia R, Qin SJ (1998) Joint diagnosis of process and sensor faults using principal component analysis. *Control Eng Pract* 6(4):457–469
- Good RP, Kost D, Cherry GA (2010) Introducing a unified PCA algorithm for model size reduction. *IEEE Trans Semicond Manuf* 23(2):201–209
- He X, Yan S, Hu Y, Niyogi P, Zhang HJ (2005) Face recognition using laplacianfaces. *IEEE Trans on Pattern Analysis and Machine Intell* 27:328–340
- Lee J, Yoo C, Lee I (2004) Statistical process monitoring with independent component analysis. *J Process Control* 14:467–485
- Lee C, Choi S, Lee I-B (2006) Variable reconstruction and sensor fault identification using canonical variate analysis. *J Process Control* 16(7):747–761
- Luo L (2014) Process monitoring with global-local preserving projections. *Ind Eng Chem Res* 53(18):7696–7705
- Luo L, Bao S, Mao J, Tang D (2016) Nonlocal and local structure preserving projection and its application to fault detection. *Chemom Intell Lab Syst* 157:177–188
- Lyman PR, Georgakis C (1995) Plant-wide control of the tennessee eastman problem. *Comput Chem Eng* 19:321–331
- Qin S, Zheng Y (2012) Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. *AIChE J* 59:496–504
- Zhong B, Wang J, Zhou J, Wu H, Jin Q (2016) Quality-related statistical process monitoring method based on global and local partial least-squares projection. *Ind Eng Chem Res* 55:1609–1622

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 11

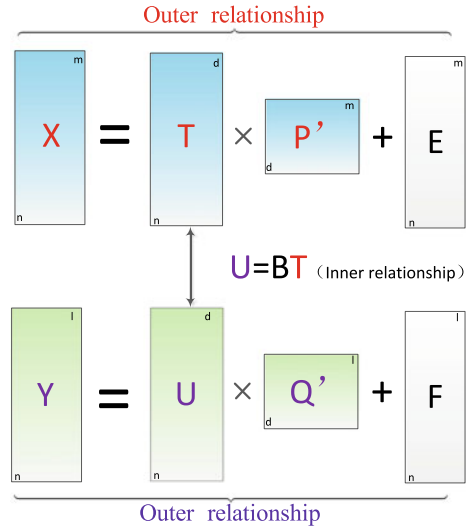
Locally Linear Embedding Orthogonal Projection to Latent Structure



Quality variables are measured much less frequently and usually with a significant time delay by comparison with the measurement of process variables. Monitoring process variables and their associated quality variables is essential undertaking as it can lead to potential hazards that may cause system shutdowns and thus possibly huge economic losses. Maximum correlation was extracted between quality variables and process variables by partial least squares analysis (PLS) (Kruger et al. 2001; Song et al. 2004; Li et al. 2010; Hu et al. 2013; Zhang et al. 2015). In order to deal with the nonlinear correlation of industrial data, this chapter proposes another two nonlinear PLS methods, named as Local Linear Embedded Projection of Latent Structure (LLEPLS). LLEPLS is an oblique projection on the input data space. By further decomposing the LLEPLS model, Local Linear Embedded Orthogonal Projection of Latent Structure (LLEOPLS) is proposed which the orthogonal projection on the input space is obtained. LLEPLS or LLEOPLS also extracts the maximum relevant information and preserves the local nonlinear structure between input and output simultaneously.

LLEPLS or LLEOPLS project the input and output space into three subspaces from the view of statistical analysis: (1) joint input-output subspace, aiming at finding the nonlinear relationship between the input and output. It also can be used for quality prediction. (2) output-residual subspace, aiming at monitoring the quality-related fault which cannot be predicted from the process data. (3) orthogonal input-residual subspace, aiming at identifying whether the predictable fault is quality related. The corresponding monitoring strategies are established based on the LLEPLS and LLEOPLS model, respectively.

Fig. 11.1 Outer- and inner-model presentation for PLS decomposition



11.1 Comparison of GPLPLS, LPPLS, and LLEPLS

PLS has a better performance compared to PCA in quality-relevant faults. As shown in Fig. 11.1, the output space (Y) and input space (X) are decomposed for the PLS model. Here the external relationship is the “foundation” and the internal relationship is the “result”. For nonlinear PLS, the desired “results” cannot be obtained by internal structure adjustment (Zhang and Qin 2008), if the external relationships are linear. Therefore, it is possible to build better internal relationships by starting with the analysis of external relationships. The nonlinear function usually is approximated by a series of locally weighted linear model. For example, (Wang et al. 2014; Yin et al. 2016, 2017) use the locally weighted projection regression (LWPR) or few univariate regressions to learn the nonlinearity of external relationships. This PLS regression can be considered as multi-KPLS regression with Gaussian kernel to some extent.

The location-preserving partial least squares (LPPLS) model (given in Chap. 10) is another external nonlinear PLS model and its structure is relatively simple compared to the KPLS model (Wang et al. 2017). However, the LPPLS model has at least two limitations. The first one is that the local geometric structure (uniform weights) cannot be preserved better, or the σ parameter (Gaussian weights) (Kokiopoulou and Saad 2007) is difficult to be selected properly. The second is an oblique decomposition of the measurement process variables. The LPPLS model extracts the principal components and retains local structure by locality-preserving projection (LPP). LLE, another nonlinear dimensionality reduction technique, transforms the global nonlinear problem into a combination of several local linear problems by introducing local geometric information. Compared with LLE method, the local preserving strategy of

LPP is more complex, and its parameters (Gaussian weights) are more and not easy to tuned.

The global plus local projection to latent structure (GPLPLS) (given in Chap. 9) integrates the advantages of PLS and LLE methods. The distinctive feature of the GPLPLS model is that the local nonlinear features are enhanced by LLE in the PLS decomposition (Zhou et al. 2018). GPLPLS uses the strategy of plus but not embedding, in which the new feature space is divided into linear part (global projection) and nonlinear part (local preserving). It confirms that the LLE plus PLS algorithm is able to perform the decomposition of the input and output space, and effectively preserve the local geometric structure. However, this combination needs further research, such as how to combine more effectively, how to make the orthogonal decomposition be completed, and also how to quantitatively evaluate the monitoring effect.

Based on the above analysis, Local Linear Embedded Projection of Latent Structure (LLEPLS) is proposed. It extracts the maximum correlation information between input and output, at the same time reveals and preserves the intrinsic nonlinear structure of the original data. The principal components of the input space (or measured variables space) extracted by LLEPLS still contain the variations orthogonal to Y . These variations are output irrelevant and do not contribute to the output prediction. Moreover, LLEPLS is an oblique projection on the input space. Orthogonalization is an alternative solution for these issues. Then the local linear embedded orthogonal projection to latent structure (LLEOPLS) model is proposed in order to explain further the LLEPLS prediction model and detect quality-related faults. LLEOPLS eliminates the T^2 statistic including variations orthogonal to the output. LLEOPLS differs significantly from other existing nonlinear PLS models in orthogonal projections with local geometric structure preservation and less easily fixed parameters.

11.2 A Brief Review of the LLE Method

Given the normalized data set $X = [\mathbf{x}^T(1), \mathbf{x}^T(2), \dots, \mathbf{x}^T(n)]^T \in R^{n \times m}$, ($\mathbf{x} = [x_1, x_2, \dots, x_m] \in R^{1 \times m}$) of the model, where n is the sampling time and m is the number of input variables. LLE algorithm introduces the local structural information and transforms the global nonlinear problem into a combination of multiple local linear problems. It is outstanding at the locally nonlinear processes.

The size of neighborhood k_x is crucial for the local geometric structure. According to the distance measures such as Euclidean distance, the K nearest neighbors (KNN) of the sample can be selected (Kouropteva et al. 2002),

$$k_{x,opt} = \arg \min_{k_x} (1 - \rho_{D_x D_{\phi_x}}^2), \quad (11.1)$$

where D_x and D_{ϕ_x} denotes the distance matrices (between point pairs) in X and Φ_x (Φ_x given in (11.4)), and ρ denotes the standard linear correlation coefficient between D_x and D_{ϕ_x} .

Next, the k_x nearest neighbors of the sample $\mathbf{x}(i)$ can be obtained. Then $\mathbf{x}(i)$ can be linearly expressed based on its the k_x nearest neighbors $\mathbf{x}(j)$ by the following optimization object,

$$J(\mathbf{A}_x) = \min \sum_{i=1}^n \left\| \mathbf{x}(i) - \sum_{j=1}^{k_x} a_{ij,x} \mathbf{x}(j) \right\|^2 \quad (11.2)$$

$$\text{s.t.} \quad \sum_{j=1}^{k_x} a_{ij,x} = 1,$$

where $[a_{ij,x}] := \mathbf{A}_x \in R^{n \times k_x}$, ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, k_x$) denotes the weight coefficients. Usually, points belonging to the space X are projected onto a new low-dimensional reduced space $\Phi_x = [\phi_x^T(1), \phi_x^T(2), \dots, \phi_x^T(n)]^T \in R^{n \times d}$, ($d < m$, $\phi_x \in R^{1 \times d}$) determined by the following optimization:

$$J_{\text{LLE}}(\mathbf{W}) = \min \sum_{i=1}^n \left\| \phi_x(i) - \sum_{j=1}^{k_x} a_{ij,x} \phi_x(j) \right\|^2 \quad (11.3)$$

$$\text{s.t.} \quad \Phi_x^T \Phi_x = \mathbf{I}.$$

In order to further analysis, a linear mapping matrix $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_d] \in R^{m \times d}$ is introduced with the guarantee of local embedding,

$$\phi_x(i) = \mathbf{x}(i)\mathbf{W}, \quad (i = 1, 2, \dots, n). \quad (11.4)$$

where \mathbf{w}_j , $j = 1, \dots, d$ denotes the projection vector. Then the optimization (11.3) is rewritten as

$$J_{\text{LLE}}(\mathbf{W}) = \min \text{tr}(\mathbf{W}^T \mathbf{X}^T \mathbf{M}_x^T \mathbf{M}_x \mathbf{X} \mathbf{W}) \quad (11.5)$$

$$\text{s.t.} \quad \mathbf{W}^T \mathbf{X}^T \mathbf{X} \mathbf{W} = \mathbf{I},$$

where $\mathbf{M}_x = (\mathbf{I} - \mathbf{A}_x) \in R^{n \times n}$. SVD operation is performed on \mathbf{M}_x in order to simplify the dimensionality reduction problem,

$$\mathbf{M}_x = [\mathbf{U}_x \ \bar{\mathbf{U}}_x]^T \begin{bmatrix} \mathbf{S}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_x \\ \bar{\mathbf{V}}_x \end{bmatrix}$$

Then, the minimum value problem (11.5) is changed as follows:

$$J_{\text{LLE}}(\mathbf{W}) = \max \text{tr}(\mathbf{W}^T \mathbf{X}_M^T \mathbf{X}_M \mathbf{W}) \quad (11.6)$$

$$\text{s.t.} \quad \mathbf{W}^T \mathbf{X}^T \mathbf{X} \mathbf{W} = \mathbf{I},$$

where $X_M := \begin{bmatrix} S_x^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \bar{V}_x \\ \bar{V}_x \end{bmatrix} X = S_{V_x} X$. Generally, LLE should choose the reduced dimension d in (11.3) in advance, but PCA can determine the corresponding dimension based on the specific criteria such as the cumulative contribution. The optimization problem (11.6) is further rewritten,

$$\begin{aligned} J_{\text{LLE}}(\mathbf{w}) &= \max \mathbf{w}^T X_M^T X_M \mathbf{w} \\ \text{s.t. } & \mathbf{w}^T X^T X \mathbf{w} = 1, \end{aligned} \quad (11.7)$$

where $\mathbf{w} \in R^{m \times 1}$. The criteria of determining the number of principal components in PCA can be directly applied to LLE. Based on the SVD algorithm, the matrix X_M is decomposed into a “load matrix” $P_d = [p_1, p_2, \dots, p_d]$ and a “score matrix” $T_d = [t_1, t_2, \dots, t_d]$

$$X_M^T X_M = [P_{d0} \ P_{r0}] \begin{bmatrix} A_d & \\ & A_r \end{bmatrix} \begin{bmatrix} P_{d0} \\ P_{r0} \end{bmatrix}$$

and defined $P_d = P_{d0}/\|X P_{d0}\|$, $P_r = P_{r0}/\|X P_{r0}\|$, and

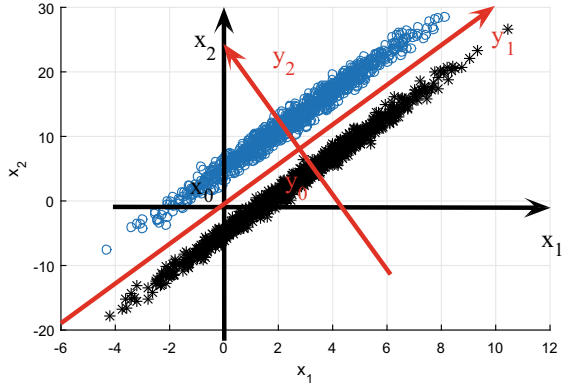
$$\begin{aligned} X_M &= T_d P_d^T + T_r P_r^T \\ &= P_d P_d^T X_M + (I - P_d P_d^T) X_M, \end{aligned} \quad (11.8)$$

where $T_d = X_M P_d$, $T_r = X_M P_r$.

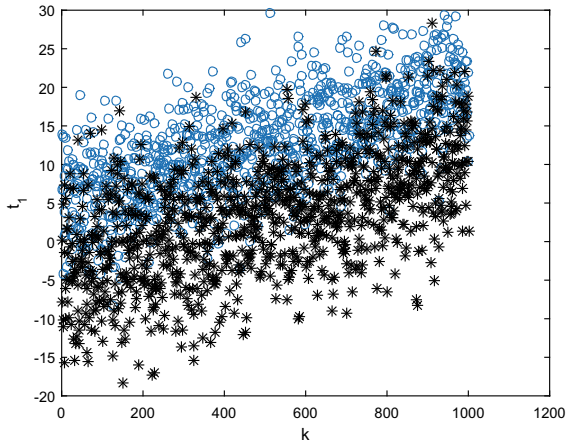
It is observed from (11.7) and (11.8) that the projection direction of LLE can be obtained by maximizing the variance. Thus, the LLE constructs a new PLS regression with the local geometric structure-preserving ability according to the component extraction criteria.

Variance (factor variation) is used to extract the latent variables in PLS algorithm. It transforms the original data X and Y into a set of t-scores T and u-scores U . The latent factors T and U are chosen by maximizing the factor variation. It aims at using fewer dimensions but retaining more features of the original data. PLS is a linear dimensionality reduction technique, but does not explore the intrinsic structure of original data. It is not conducive to data classification, but may make data mixed together. The phenomena that may occur with PLS are given in Fig. 11.2, similar as the PCA. Figure 11.2a shows a two-mode data space X and Fig. 11.2b give its first principal component t_1 in PCA. The contribution of the first principal component of t_1 is 99%. As shown in Fig. 11.2b, the blue ‘o’ and black ‘*’ points in the one-dimensional coordinate system are mixed together. The second principal component is discarded due to its small contribution although it maintains the local geometric structure.

Fig. 11.2 PCA decomposition and its project of a two-mode data space



(a) Two-mode data space of \mathbf{X}



(b) First principal component t_1 in PCA

11.3 LLEPLS Models and LLEPLS-Based Fault Detection

11.3.1 LLEPLS Models

In order to extract the first component pair (t_1, u_1) , the traditional PLS optimization is expressed as

$$\begin{aligned}
 J_{\text{PLS}}(\mathbf{w}_1, \mathbf{c}_1) &= \max \mathbf{w}_1^T \mathbf{X}^T \mathbf{Y} \mathbf{c}_1 \\
 \text{s.t. } &\mathbf{w}_1^T \mathbf{w}_1 = 1, \mathbf{c}_1^T \mathbf{c}_1 = 1.
 \end{aligned}
 \tag{11.9}$$

Define $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{F}_0 = \mathbf{Y}$. The PLS latent variables t_1 and c_1 of are obtained from $t_1 = \mathbf{E}_0 \mathbf{w}_1$ and $u_1 = \mathbf{F}_0 \mathbf{c}_1$. Here \mathbf{c}_1 and \mathbf{w}_1 are the eigenvectors corresponding to

the maximum eigenvalues of matrices,

$$\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0 \mathbf{w}_1 = \theta_1^2 \mathbf{w}_1 \quad (11.10)$$

$$\mathbf{F}_0^T \mathbf{E}_0 \mathbf{E}_0^T \mathbf{F}_0 \mathbf{c}_1 = \theta_1^2 \mathbf{c}_1. \quad (11.11)$$

Locality linearly embedded partial least squares (LLEPLS) is proposed to optimize the function as follows:

$$\begin{aligned} J_{\text{LLEPLS}}(\mathbf{w}_1, \mathbf{c}_1) &= \max \mathbf{w}_1^T \mathbf{X}_M^T \mathbf{Y}_M \mathbf{c}_1 \\ \text{s.t. } \mathbf{w}_1^T \mathbf{X}^T \mathbf{X} \mathbf{w}_1 &= 1, \mathbf{c}_1^T \mathbf{Y}^T \mathbf{Y} \mathbf{c}_1 = 1 \end{aligned} \quad (11.12)$$

in which,

$$\begin{aligned} \mathbf{Y}_M &= \begin{bmatrix} \mathbf{S}_y^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_y \\ \bar{\mathbf{V}}_y \end{bmatrix} \mathbf{Y} = \mathbf{S}_{V_y} \mathbf{Y} \\ \mathbf{M}_y &= \mathbf{I} - \mathbf{A}_y = [\mathbf{U}_y \ \bar{\mathbf{U}}_y]^T \begin{bmatrix} \mathbf{S}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_y \\ \bar{\mathbf{V}}_y \end{bmatrix} \end{aligned}$$

where \mathbf{A}_y is accompanied by its neighbors with different parameters k_y , similar as \mathbf{A}_x . \mathbf{S}_y , \mathbf{V}_y and \mathbf{U}_y are also similar to \mathbf{S}_x , \mathbf{V}_x and \mathbf{U}_x .

The criteria of LLEPLS component decomposition and latent factors extraction are given as follows:

- (1) The latent factors \mathbf{u}_i and \mathbf{t}_i are chosen to maximize the nonlinear variation of the factors (by local linear embedding).
- (2) The correlation between potential factors \mathbf{u}_i and \mathbf{t}_i should be as strong as possible.

Then, the latent variable calculation process of LLEPLS model is given as follows. Denote $\mathbf{E}_{0L} = \mathbf{X}_M$ and $\mathbf{F}_{0L} = \mathbf{Y}_M$, similar as the traditional linear PLS. The constrained optimization problem (11.12) is transformed by introducing a Lagrange multiplier vector,

$$\begin{aligned} \Psi(\mathbf{w}_1, \mathbf{c}_1) &= \mathbf{w}_1^T \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 - \lambda_1 (\mathbf{w}_1^T \mathbf{X}^T \mathbf{X} \mathbf{w}_1 - 1) \\ &\quad - \lambda_2 (\mathbf{c}_1^T \mathbf{Y}^T \mathbf{Y} \mathbf{c}_1 - 1). \end{aligned} \quad (11.13)$$

The optimal \mathbf{w}_1 and \mathbf{c}_1 is solved by $\frac{\partial \Psi}{\partial \mathbf{w}_1} = 0$ and $\frac{\partial \Psi}{\partial \mathbf{c}_1} = 0$. Next, the optimization problem (11.13) is solved by the maximum eigenvalue problem,

$$(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L} \mathbf{w}_1 = \theta_1^2 \mathbf{w}_1 \quad (11.14)$$

$$(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{F}_{0L}^T \mathbf{E}_{0L} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{E}_{0L}^T \mathbf{F}_{0L} \mathbf{c}_1 = \theta_1^2 \mathbf{c}_1. \quad (11.15)$$

The first optimal weight vector \mathbf{w}_1 in the conventional linear PLS (11.10) is corresponding to the matrix $\mathbf{E}_0^T \mathbf{F}_0 \mathbf{F}_0^T \mathbf{E}_0$. For the LLEPLS (11.14), the optimal \mathbf{w}_1 is

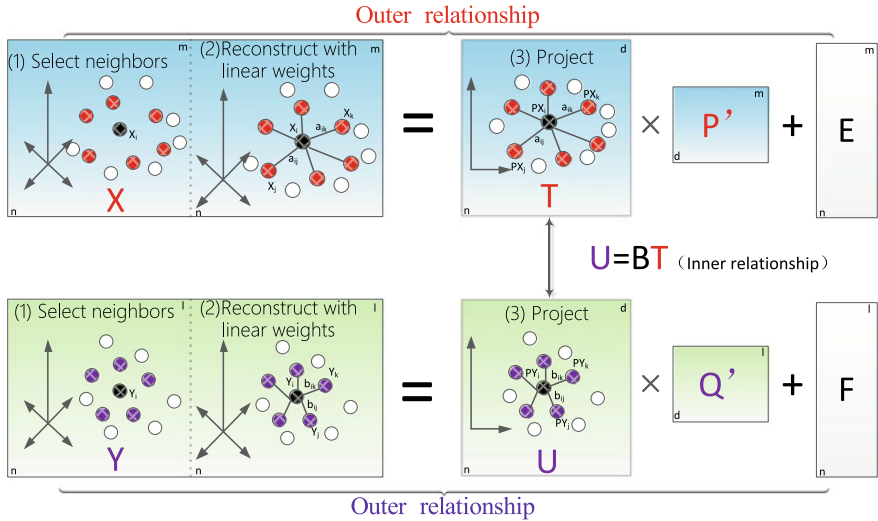


Fig. 11.3 Outer- and inner-model presentation for LLEPLS decomposition

derived from the corresponding matrix $(X^T X)^{-1} E_{0L}^T F_{0L} (Y^T Y)^{-1} F_{0L}^T E_{0L}$. These matrices are particularly similar. The extraction and modeling of the residual components can be done by traditional PLS methods.

It is worth pointing out that the columns of the input space X and/or the output space Y may not be full rank. The inverse of $X^T X$ and/or $Y^T Y$ does not exist. Similar as the S_x in (11.6), the corresponding matrix inverse can be obtained for X and/or Y . It does not affect the following analysis, so both cases will be treated indiscriminately in the rest of this chapter.

The first d components are obtained to predict the regression model, where d is determined by cross-validation tests. Similar to the outer- and inner-model presentation for PLS decomposition, the corresponding LLEPLS decomposition is shown in Fig. 11.3. It is found that that the new feature space X_F and Y_F are both constructed by the nonlinear part, i.e., the local structure information. Compared with the decomposition of GPLPLS shown in Fig. 9.2, the global linear part is eliminated.

11.3.2 LLEPLS for Process and Quality Monitoring

The linear localization embedding in the low-dimensional space of X and Y is formed by few latent variables (t_1, \dots, t_d) in the LLEPLS model. The neighborhood mappings of E_{0L} and F_{0L} are decomposed as follows:

$$\begin{aligned}
\mathbf{E}_{0L} &= \sum_{i=1}^d \mathbf{t}_i \mathbf{p}_i^T + \bar{\mathbf{E}}_{0L} = \mathbf{T} \mathbf{P}^T + \bar{\mathbf{E}}_{0L} \\
\mathbf{F}_{0L} &= \sum_{i=1}^d \mathbf{t}_i \mathbf{q}_i^T + \bar{\mathbf{F}}_{0L} = \mathbf{T} \mathbf{Q}^T + \bar{\mathbf{F}}_{0L},
\end{aligned} \tag{11.16}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_d]$ denotes the score vectors, $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_d]$ and $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_d]$ denote the loading matrices of \mathbf{E}_{0L} and \mathbf{F}_{0L} , respectively. Score \mathbf{T} is represented in terms of the neighboring mapping data \mathbf{E}_{0L} ,

$$\mathbf{T} = \mathbf{E}_{0L} \mathbf{R} = \mathbf{S}_{V_x} \mathbf{E}_0 \mathbf{R}, \tag{11.17}$$

where $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_d]$, and

$$\mathbf{r}_i = \prod_{j=1}^{i-1} (\mathbf{I}_n - \mathbf{w}_j \mathbf{p}_j^T) \mathbf{w}_i.$$

Equations (11.16) and (11.17) are difficult to directly apply in practice due to the calculation of locality-preserving matrix S , so the decomposition for the scaled and mean-centered \mathbf{E}_0 and \mathbf{F}_0 are given,

$$\mathbf{E}_0 = \mathbf{T}_0 \mathbf{P}^T + \bar{\mathbf{E}}_0 \tag{11.18}$$

$$\begin{aligned}
\mathbf{F}_0 &= \mathbf{T}_0 \bar{\mathbf{Q}}^T + \bar{\mathbf{F}}_0 \\
&= \mathbf{E}_0 \mathbf{R} \bar{\mathbf{Q}}^T + \bar{\mathbf{F}}_0,
\end{aligned} \tag{11.19}$$

where $\mathbf{T}_0 = \mathbf{E}_0 \mathbf{R}$, $\bar{\mathbf{Q}} = \mathbf{T}_0^+ \mathbf{F}_0$.

Now let's consider the monitoring of new samples \mathbf{x} and subsequently on \mathbf{y} . First the samples are scaled and mean-centered, an oblique projection is derived on the input data space \mathbf{x} .

$$\begin{aligned}
\mathbf{x} &= \hat{\mathbf{x}} + \mathbf{x}_e \\
\hat{\mathbf{x}} &= \mathbf{P} \mathbf{R}^T \mathbf{x} \\
\mathbf{x}_e &= (\mathbf{I} - \mathbf{P} \mathbf{R}^T) \mathbf{x}.
\end{aligned} \tag{11.20}$$

The statistics T^2 and Q are calculated as follows:

$$\begin{aligned}
\mathbf{t} &= \mathbf{R}^T \mathbf{x} \\
T^2 &= \mathbf{t}^T \mathbf{\Lambda}^{-1} \mathbf{t} = \mathbf{t}^T \left(\frac{1}{n-1} \mathbf{T}_0^T \mathbf{T}_0 \right)^{-1} \mathbf{t} \\
Q &= \|\mathbf{x}_e\|^2 = \mathbf{x}^T (\mathbf{I} - \mathbf{P} \mathbf{R}^T) \mathbf{x},
\end{aligned} \tag{11.21}$$

where $\mathbf{\Lambda}$ is the sample covariance matrix.

The space of measured variables, i.e., input space, is divided into two subspaces: score subspace and residual subspace. LLEPLS detects the quality-related faults by the T^2 statistic in the score subspace and detects the quality-irrelevant faults by Q -statistics in the residual subspace. The PLS scores which constitute the T^2 statistic still includes the variation orthogonal to Y . Therefore, LLEPLS still has deficiencies in the quality-related fault detection.

11.4 LLEOPLS Models and LLEOPLS-Based Fault Detection

As demonstrated in (Li et al. 2010), (Ding et al. 2013), the standard PLS performs a diagonal decomposition of the measured process variables. The LLEPLS model (11.16) also is a oblique decomposition operation (11.20) on the measured process variables, which is similar to the standard PLS model. Thus, the major part of the measured process variables may include variations orthogonal to the output variables. In other words, the principle component still include the output irrelevant variation, and the residual part may include a large of output-related variation. In addition, the number of principal components is often dependent on the operator's decision and is likely to cause the problems of component redundancy. In order to solve these problem, it is necessary to further decompose the LLEPLS model in equation (11.18) and get an orthogonal decomposition for the measured process variables. In this model, the regression coefficient $R\bar{Q}^T$ in equation (11.19) are used to describe the relationship between E_0 and F_0 . Performing the SVD operation on $R\bar{Q}^T$ to obtain orthogonal decomposition,

$$R\bar{Q}^T = U_{pc} S_{pc} V_{pc}^T, \quad (11.22)$$

where S_{pc} contains all non-zero singular values in descending order. V_{pc} and U_{pc} are the corresponding right and left singular vectors. Then,

$$\begin{aligned} F_0 &= E_0 U_{pc} S_{pc} V_{pc}^T + \bar{F}_0 \\ &= T_{pc} Q_{pc}^T + \bar{F}_0, \end{aligned} \quad (11.23)$$

where $T_{pc} = E_0 U_{pc}$, $Q_{pc} = V_{pc} S_{pc}$. The output-residual subspace \bar{F}_0 indicates an unpredictable output but may include some variation.

Furthermore, E_0 decomposes into two orthogonal subspaces by T_{pc} .

$$\begin{aligned} E_0 &= \hat{E}_0 + X_e \\ &= T_{pc} U_{pc}^T + E_0 (I - U_{pc} U_{pc}^T), \end{aligned} \quad (11.24)$$

where $\hat{E}_0 := T_{pc} U_{pc}^T$ and $X_e = E_0 (I - U_{pc} U_{pc}^T)$. X_e denotes the orthogonal input-residual subspace. The new data samples x and subsequently y are

orthogonal projected on the input data space \mathbf{x} for process and quality monitoring,

$$\begin{aligned}
 \mathbf{x} &= \hat{\mathbf{x}} + \mathbf{x}_e \\
 \hat{\mathbf{x}} &= \mathbf{U}_{pc} \mathbf{U}_{pc}^T \mathbf{x} \\
 \mathbf{x}_e &= (\mathbf{I} - \mathbf{U}_{pc} \mathbf{U}_{pc}^T) \mathbf{x} \\
 \mathbf{t}_{pc} &= \mathbf{U}_{pc} \mathbf{x} \\
 \mathbf{y}_e &= \mathbf{y} - \mathbf{Q}_{pc} \mathbf{t}_{pc}.
 \end{aligned} \tag{11.25}$$

The LLEOPLS model is given in (11.23) and (11.24) with many parameters to be determined in prior. The selection of the optimal parameters has been described for LLE (Kouropyteva et al. 2002). The optimal parameters $[k_x, k_y]$ of LLEOPLS model is determined by simultaneously considering the characteristics of the LLE itself and the relationship between the input and output spaces. The following optimization is given for determining the parameters $[k_x, k_y]$:

$$\begin{aligned}
 [k_x, k_y]_{\text{opt}} &= \arg \min_{k_x, k_y} \left(1 - \rho_{\mathbf{D}_x \mathbf{D}_{\hat{\mathbf{x}}}}^2 + 1 - \rho_{\mathbf{D}_y \mathbf{D}_{\hat{\mathbf{y}}}}^2 \right. \\
 &\quad \left. + 1 - \rho_{\mathbf{D}_y \mathbf{D}_y}^2 \Big|_{\text{train}} + 1 - \rho_{\mathbf{D}_y \mathbf{D}_y}^2 \Big|_{\text{pre}} \right),
 \end{aligned} \tag{11.26}$$

where $\hat{\mathbf{y}} = \mathbf{Q}_{pc} \mathbf{t}_{pc} \cdot \cdot|_{\text{train}}$ and $\cdot \cdot|_{\text{pre}}$ are the training data set and the testing data sets, respectively. The first two terms in (11.26), $1 - \rho_{\mathbf{D}_x \mathbf{D}_{\hat{\mathbf{x}}}}^2$ and $1 - \rho_{\mathbf{D}_y \mathbf{D}_{\hat{\mathbf{y}}}}^2$, aim at evaluating the geometric similarity between the embedding space and the high-dimensional space. The last two terms, $1 - \rho_{\mathbf{D}_y \mathbf{D}_y}^2$ and $1 - \rho_{\mathbf{D}_y \mathbf{D}_y}^2$, indicate the effect of the model which indirectly reflects the role of the first two terms. Cross-validation is used to ensure the training results of the model. The last term is the most important part in (11.26),

$$[k_x, k_y]_{\text{opt}} = \arg \min_{k_x, k_y} \left(1 - \rho_{\mathbf{D}_y \mathbf{D}_y}^2 \Big|_{\text{pre}} \right). \tag{11.27}$$

A generalized LLEOPLS model with the optimal parameters k_x and k_y can be used to monitor the operation of the system. The T^2 statistics can monitor the output-related score (\mathbf{T}_{pc}), output-residual part and input-residual part,

$$\begin{aligned}
 T_{pc}^2 &= \mathbf{t}_{pc}^T \mathbf{A}_{pc}^{-1} \mathbf{t}_{pc} = \mathbf{t}_{pc}^T \left\{ \frac{1}{n-1} \mathbf{T}_{pc}^T \mathbf{T}_{pc} \right\}^{-1} \mathbf{t}_{pc} \\
 T_e^2 &= \mathbf{x}_e^T \mathbf{A}_{x,e}^{-1} \mathbf{x}_e = \mathbf{x}_e^T \left\{ \frac{1}{n-1} \mathbf{X}_e^T \mathbf{X}_e \right\}^{-1} \mathbf{x}_e \\
 T_{y,e}^2 &= \mathbf{y}_e^T \mathbf{A}_{y,e}^{-1} \mathbf{y}_e = \mathbf{y}_e^T \left\{ \frac{1}{n-1} \mathbf{Y}_e^T \mathbf{Y}_e \right\}^{-1} \mathbf{y}_e,
 \end{aligned} \tag{11.28}$$

where \mathbf{A}_{pc} , $\mathbf{A}_{x,e}$ and $\mathbf{A}_{y,e}$ denotes the sample covariance matrices. $\mathbf{Y}_e := \bar{\mathbf{F}}_0 = \mathbf{F}_0 - \mathbf{T}_{pc} \mathbf{Q}_{pc}^T$.

The \mathbf{T}_{pc} of the LLEOPLS method is not obtained from a scaled and mean-centered matrix \mathbf{E}_{0L} . The control limits of the T^2 statistical series usually are calculated based on the probability density function estimated by the non-parametric KDE method. The T_{pc}^2 and T_e^2 statistics both are univariate although the processes represented by these statistics are multivariate. Then the control limits for the monitoring statistics (T_{pc}^2 , T_e^2 and $T_{y,e}^2$) are obtained from the corresponding PDF estimation,

$$\int_{-\infty}^{\text{Th}_{pc,\alpha}} g(T_{pc}^2) dT_{pc}^2 = \alpha$$

$$\int_{-\infty}^{\text{Th}_{x_e,\alpha}} g(T_e^2) dT_e^2 = \alpha$$

$$\int_{-\infty}^{\text{Th}_{y_e,\alpha}} g(T_{y,e}^2) dT_{y,e}^2 = \alpha,$$

where

$$g(z) = \frac{1}{lh} \sum_{j=1}^l \mathcal{K} \left(\frac{z - z_j}{h} \right),$$

where $\mathcal{K}(\cdot)$ and h are kernel function and its bandwidth or smoothing parameter, respectively.

Finally, the fault detection logic for the output-residue subspace is given,

$$\begin{aligned} T_{y,e}^2 > \text{Th}_{y_e,\alpha} & \quad \text{Unpredictable output faults} \\ T_{y,e}^2 \leq \text{Th}_{y_e,\alpha} & \quad \text{Fault-free in unpredictable output.} \end{aligned} \quad (11.29)$$

$T_{y,e}^2$ includes the output information, so it is suitable for monitoring the output-residual subspace. But this posteriori quality monitoring is not the focus. Instead, process-based quality monitoring is of greater interest. Fault detection logic for the input space is (Zhou et al. 2018):

$$\begin{aligned} T_{pc}^2 > \text{Th}_{pc,\alpha} & \quad \text{Quality-relevant faults} \\ T_{pc}^2 > \text{Th}_{pc,\alpha} \text{ or } T_e^2 > \text{Th}_{x_e,\alpha} & \quad \text{Process-relevant faults} \\ T_{pc}^2 \leq \text{Th}_{pc,\alpha} \text{ \& } T_e^2 \leq \text{Th}_{x_e,\alpha} & \quad \text{Fault-free.} \end{aligned} \quad (11.30)$$

The monitoring process of LLEOPLS algorithm for the complex industrial system is given as follows:

1. The original data X and Y is scaled to zero mean and unit variance.
2. The LLE and PLS optimization objectives ((11.4) and (11.9)) are combined. Then perform the LLEPLS operation for X and Y to yield T_0 , \bar{Q} and R as well as the output-residual subspace Y_e , based on (11.18) and (11.19).
3. The number of LLEPLS factors d is determined by cross-validation.
4. Perform SVD on $R\bar{Q}^T$. Further access to U_{pc} , T_{pc} and Q_{pc} .
5. Build the input-residual subspace X_e .
6. Calculate the control limits (11.28) and finish the fault monitoring according to the fault detection logic (11.30).

11.5 Case Study

The fault detection strategy based on the proposed LLEPLS and LLEOPLS model is performed on the Tennessee Eastman Process (TEP) simulation platform (Lyman and Georgakis 1995). To better demonstrate the effectiveness and rationality of the proposed monitoring strategy, the PLS monitoring strategy and the concurrent projection to latent structure (CPLS) model (Qin and Zheng 2012) are compared. With the CPLS algorithm, the input and output spaces are projected into five subspaces: the input-principle subspace, the input-residual subspace, the output-principle subspace, the output-residual subspace, and the joint input-output subspace. When only the monitoring capability of quality-related faults is considered, the input-residual subspace replaces the input-residual and -principle subspace in the CPLS model. The T_e^2 replaces the corresponding monitoring strategy. In order to emphasize the process-based quality monitoring, the output-residual subspace in LLEOPLS model will not be considered. Similarly, the output-principle and -residual subspaces in CPLS model are not considered.

11.5.1 Models and Discussion

All process measurement variables (XMEAS (1:22)) and manipulation variables (XMV (1:11)) form the input variables matrix X . The quality variable matrix Y consists of XMEAS (35) and (38). The training data set is normal data IDV(0) and the testing data consists of the 21 fault data IDV(1-21). The optimal parameters of LLEPLS and LLEOPLS are $k_x = 24$ and $k_y = 20$. The number of principal components of the PLS, CPLS, LLEPLS, and LLEOPLS models are 6, 6, 5, and 5, respectively.

From the analysis of previous Chaps. 9 and 10, it is known that faults IDV(3,4), IDV(9,11), IDV(14,15), and IDV(19) had almost no effect on product quality but other faults produced significant variations in quality variables when select component G (XMEAS(35)) and component E (XMEAS(38)) as product quality variables. The FDR and FAR of PLS, LLEPLS, CPLS, and LLEOPLS at the control limit

Table 11.1 FDR of PLS, LLEPLS, CPLS, and LLEOPLS

IDV	PLS		CPLS		LLEPLS		LLEOPLS		PQAR
	T ²	Q	T ²	T _e ²	T ²	Q	T _{pc} ²	T _e ²	
1	99.13	99.38	96.13	99.88	84.88	99.25	29.63	99.75	28.25
2	98.00	98.25	81.25	98.25	92.88	97.00	78.50	98.25	77.00
3	0.38	0.13	0.50	1.25	0.50	0.25	0.50	1.75	0.88
4	0.63	86.00	0.13	100.00	0.50	1.25	0.25	100.00	0.25
5	21.88	16.00	20.38	100.00	22.38	13.75	14.13	100.00	21.88
6	99.25	100.00	99.25	100.00	99.75	99.75	97.00	100.00	95.25
7	36.75	100.00	35.63	100.00	36.63	47.50	25.25	100.00	33.88
8	92.50	94.00	87.75	97.88	88.00	87.38	77.38	97.88	75.38
9	0.63	0.50	0.38	1.25	1.00	0.38	0.38	1.88	0.88
10	30.00	4.38	28.00	86.25	28.38	16.63	14.50	88.25	17.13
11	1.38	57.88	0.25	77.50	1.13	10.13	1.38	77.75	1.38
12	87.50	91.00	84.75	99.88	87.13	85.50	75.13	99.88	81.00
13	93.88	93.00	85.00	95.25	92.75	89.13	85.00	95.25	85.13
14	33.50	100.00	1.63	100.00	0.13	96.38	0.13	100.00	0.00
15	0.63	0.38	0.75	2.50	0.38	2.88	1.75	3.88	0.25
16	14.25	3.13	12.63	89.38	17.88	10.38	7.63	91.25	8.63
17	56.00	85.38	37.13	96.88	4.50	66.38	2.00	96.75	5.63
18	88.00	89.25	88.00	90.13	87.75	88.38	87.50	90.25	86.38
19	0.00	4.13	0.00	91.13	0.13	0.13	0.38	91.38	0.25
20	26.63	34.00	27.75	90.38	22.75	19.88	11.25	90.88	4.38
21	29.88	39.63	24.50	43.88	33.50	19.63	16.25	53.75	16.75

with confidence level 99.75% are shown in Tables 11.1 and 11.2, respectively. Based on the two tables, the monitoring results for LLEOPLS are a little different from the other monitoring results which are almost the same as FAR, such as IDV(14) and IDV(17). They are considered as quality-related faults in the method of PLS. However, LLEOPLS method indicates that they are quality-irrelevant faults.

Which monitoring results are more credible? The following is given to assess whether the final result of the fault detection is reasonable by quantifying the posterior quality alarm rate (PQAR).

$$PQAR = \frac{\text{No. of samples } (\{|(Y_F)|\} > 3 \mid f \neq 0)}{\text{total samples } (f \neq 0)} \times 100, \tag{11.31}$$

where Y_F are the scaled and mean-centered data, which is the output data of the fault cases. The PQAR is also given in Table 11.1. The 21 faults are divided into two categories by PQAR. Type I is quality-independent ($PQAR_i < 6, i = 1, 2, \dots, 21$), including IDV(3,4,9,11,14,15,17,19,20). Type II is quality-relevant faults, and further

Table 11.2 FAR of PLS, LLEPLS, CPLS, and LLEOPLS

IDV	PLS		CPLS		LLEPLS		LLEOPLS	
	T ²	Q	T ²	T _e ²	T ²	Q	T _{pc} ²	T _e ²
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63
2	0.00	0.00	0.00	0.00	0.63	0.63	0.63	0.00
3	0.00	0.63	0.00	0.00	0.63	0.00	0.63	3.13
4	0.00	0.00	0.00	0.63	0.00	0.63	0.63	0.63
5	0.00	0.00	0.00	0.63	0.00	0.63	0.63	0.63
6	0.00	0.63	0.00	0.00	0.00	0.63	0.00	0.00
7	0.00	0.63	0.00	0.00	0.00	0.00	0.63	0.00
8	0.00	0.00	0.63	0.00	0.00	0.00	0.63	0.00
9	1.25	0.00	0.63	0.63	0.63	0.00	0.00	0.63
10	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.00
11	0.00	0.00	0.00	0.63	0.00	0.63	0.63	0.63
12	0.00	0.00	0.63	0.63	0.00	0.00	1.25	0.63
13	0.00	0.00	0.63	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.63	0.00	0.63	0.00
15	0.00	0.63	0.00	0.00	0.00	0.63	0.63	0.63
16	1.25	0.00	2.50	2.50	3.13	0.00	1.25	1.25
17	0.00	0.00	0.00	0.00	0.00	0.63	0.00	0.63
18	0.00	0.63	0.00	0.63	0.63	0.63	0.00	1.25
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	1.25	0.63	0.00	2.50	0.63	0.63	0.63	1.88

classified into three categories: IDV(16) has a slight effect on quality; IDV(1, 2, 5, 6, 7, 8, 10, 12, 13, 18) has a serious effect on quality; and IDV(21) causes a slow drift of the output variable. Apparently, the LLEOPLS method achieves a consistent conclusion (T_{pc}²). That is, the LLEOPLS model can eliminate the quality-independent interference alarms better. However, there are still some differences in alarm rates between PQAR and T_{pc}², such as IDV (5), IDV (7), and IDV (20). What causes this difference? Next, the differences between the LLEOPLS method and the other methods are further analyzed based on the PQAR and T_{pc}² alarm rates.

11.5.2 Fault Detection Analysis

The differences in fault detection results are discussed for the PLS (CPLS) model and the LLEPLS (LLEOPLS) model, respectively. Several cases exist for output variables or process variables with no faults or minor faults (IDV(3,9,15)). Both approaches

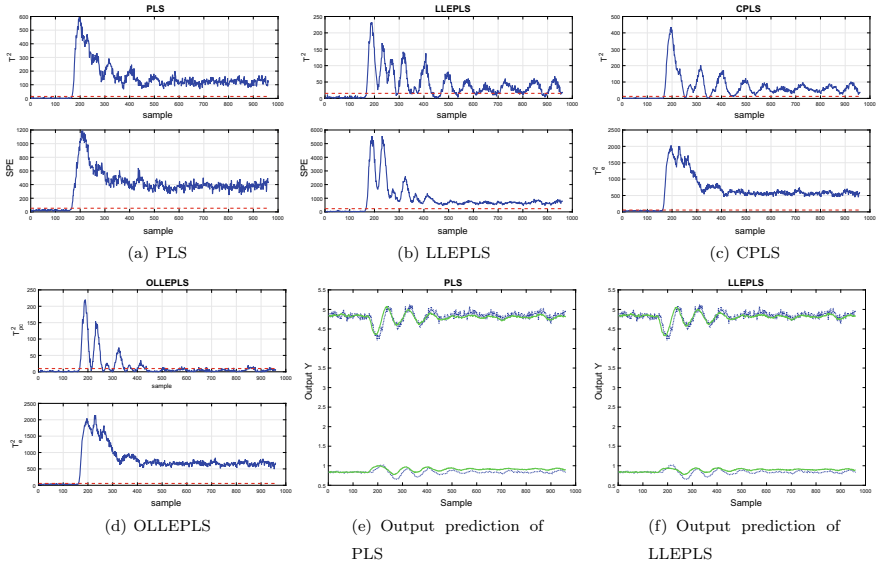


Fig. 11.4 PLS, LLEPLS, CPLS, and LLEOPLS monitoring result for IDV(1) and the output predicted values

provide consistent conclusions. For other faults, there are some differences in their diagnostic results. For two failure cases, including quality-recoverable failures and quality-irrelevant failures, the analysis is as follows. Subplots (a-d) of Figs. 11.4, 11.5 and 11.6 are monitoring result based on the statistics T^2_{pc} and T^2_e , respectively. The blue line shows the monitored value and the red dashed line shows the control limit of 99.75%. In the corresponding subplots (e) and (f) give the output prediction, where the blue dashed line is the measurement value and the green line is predicted value.

Experiment 1: Quality-Recoverable Faults

Consider the fault IDV(1), IDV(5), IDV(7). All these fault conditions are step faults, but the in-process feedback controller or cascade controller can compensate the changes in the output variables; therefore, the product quality variables under the fault condition IDV (1), IDV (5), and IDV (7) tend to return to normal. The monitoring results of IDV (1) are shown in Fig. 11.4 by the PLS, LLEPLS, CPLS, LLEOPLS methods.

It is easy to find that the T^2_e statistics in CPLS and LLEOPLS method can detect the process-related faults. The T^2_{pc} statistic of the LLEOPLS model returns back to the control limit which indicates that those faults are quality recoverable. Existing work in the literatures reports the high detection rates of these faults. For example, PLS, CPLS, and LLEPLS methods give many false alarms based on T^2 for IDV(1). In this case, the LLEOPLS method can accurately reflect the changes in both process variables and quality variables.

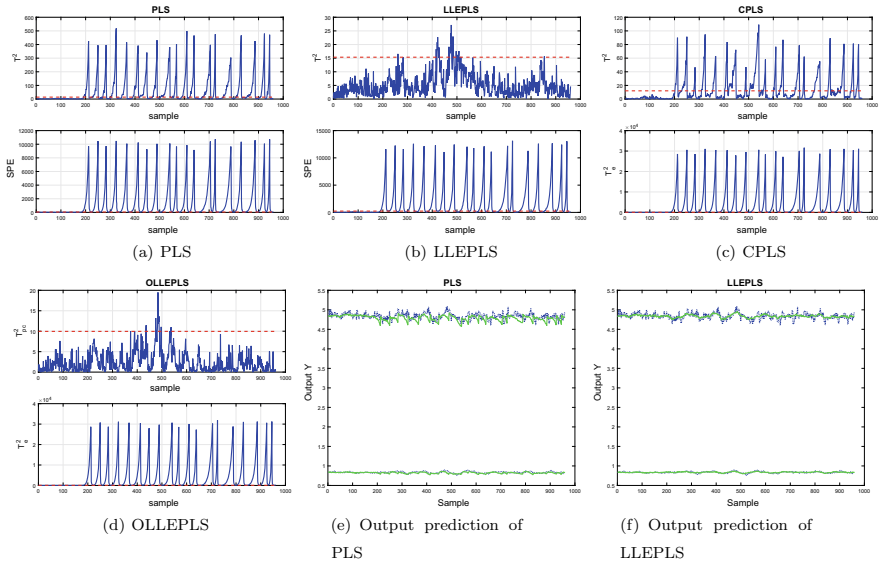


Fig. 11.5 PLS and LLEPLS monitoring result for IDV(17) and the output predicted values

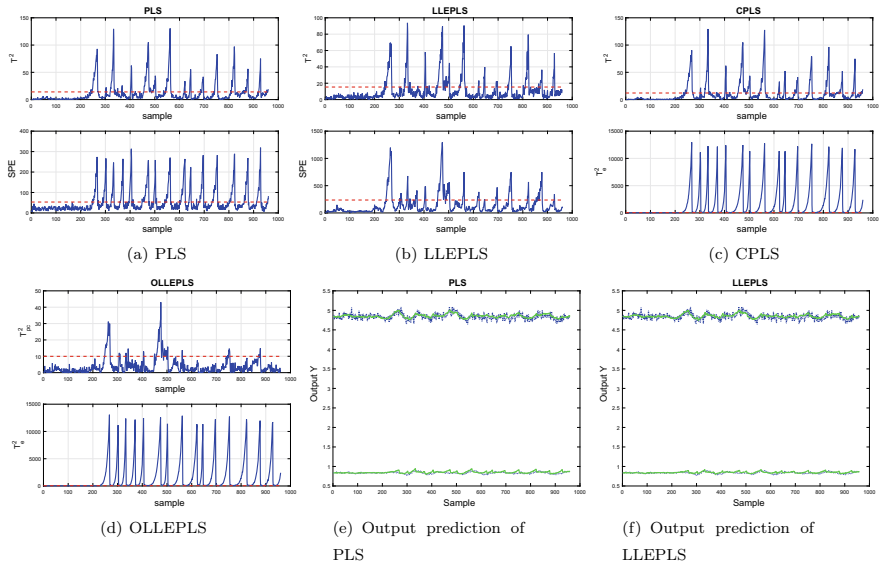


Fig. 11.6 PLS and LLEPLS monitoring result for IDV(20) and the output predicted values

For IDV(1), a huge difference between $FDR(T^2)$ and PQAR can be observed. On the one hand, $FDR(T^2 \text{ or } T_{pc}^2)$ is based on the principal components of the process variables (without time delay), while PQDR is obtained based on the actual output values (with time delay). They are not equivalent. Moreover, considering that the data used for modeling are under normal operating, but not under fault conditions. The nonlinearity feature may not be fully excited (i.e., these nonlinearities appear to be linear in the normal and steady operation). When fault occurs, nonlinearity is fully excited and may lead to false alarms and missed alarms due to the inability of the original model to predict the output. In fact, T^2 is considered to monitor the quality-related fault, which implies the assumption that the output of the system can still be well predicted by the model in case of a failure. Although the variation of the predicted value of the PLS model (XMEAS(38)) follows the variation of the actual output value, the predicted value is too large which results in a much larger FDR (T^2 in the PLS, CPLS, and LLEPLS models) than the PQAR. Nevertheless, the monitoring results of CPLS and LLEPLS are closer to reality by the orthogonalization strategy and the local linear embedding strategy.

Experiment 2: Quality-Irrelevant Faults

Fault IDV(4,11,14,17,19,20) are quality-irrelevant, in which IDV(4), IDV(11), IDV(14), and IDV(17) are considered as quality-independent but process related. The monitoring results and output predictions for IDV(17) are shown in Fig. 11.5. As shown in Fig. 11.5e, f, the PLS model cannot predict the output values well while the LLEPLS model can predict the output values very accurately. So many false alarms generated by T^2 of the PLS method. There are two possible reasons: PLS model does not map the nonlinear functions well, and its principal components contain the variations orthogonal to the output variables. Although CPLS improves the orthogonal part of PLS, its nonlinearity extracting ability is still poor. In contrast, the LLEPLS model captures the nonlinear structure well and filters out these false alarms by LLE.

IDV(20) is another touchstone for fault detection. The monitoring results and their output predictions are shown in Fig. 11.6. The detection of all methods is not good based on PQAR, but LLEOPLS method is the best. It is found from the predicted results that LLEPLS model can predict the output variation well. With the removal of the orthogonal component, there remains a question why T_{pc}^2 still fails to yield consistent results. One of the underlying reasons is that the nonlinear dynamics excited by IDV(20) cannot be well described by the parameters $[k_x, k_y] = [24, 20]$, which in turn leads to a wrong classification. Another reason could be the different control limits between PQDR and T_{pc}^2 . The statistical results of PQDR are obtained by assuming that the output variables obey a Gaussian distribution, and subsequently, their control limits are determined by a threefold standard deviation criterion. However, the 99.75% control limit of T_{pc}^2 was obtained by non-parametric estimation. This differs from the results of the Gaussian assumption. The control limit of T_{pc}^2 with confidence level 99.75% for the non-parametric KDE is 9.9583, but under the Gaussian assumption is 12.0708). In fact, the monitoring results of T_{pc}^2 of LLEOPLS

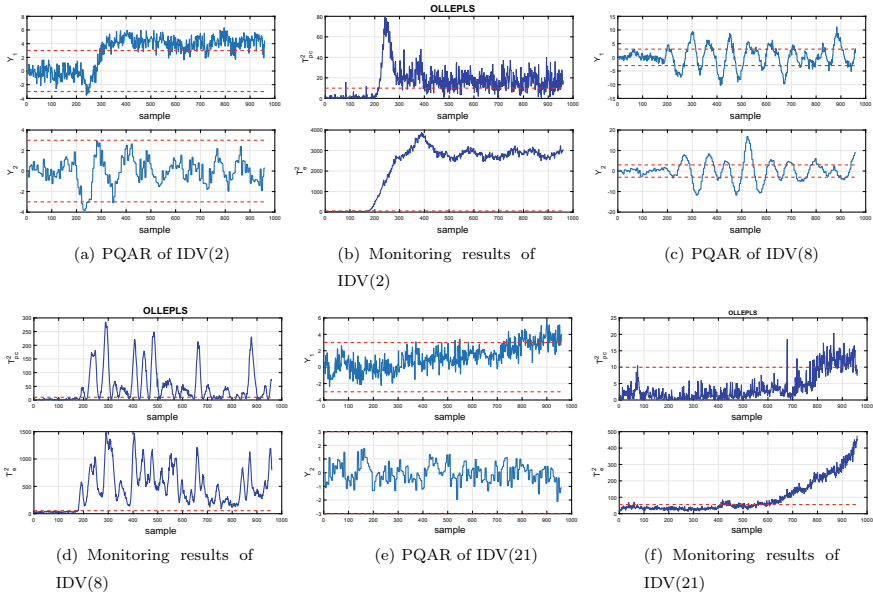


Fig. 11.7 PQAR and the corresponding LLEOPLS monitoring results

show that most of the alarms are transient alarms and few are continuous, where the transient alarms may be caused by noise.

Experiment 3: Other Quality-Related Faults

For other quality-related faults, the FDA results are essentially the same for these methods given in Table 11.1. However, the FDA results are significantly different for IDV(2), IDV(8), IDV(21), etc. The superiority of the proposed method is further verified by comparing the PQAR of IDV(2), IDV(8), IDV(21). The monitoring results are shown in Fig. 11.7. Although fault IDV(2) and IDV(8) are quality-related, the quality certainly meets the production requirements even in these fault condition. So the quality-related alarm is not higher. The monitoring results of the proposed LLEOPLS method are consistent with PQAR.

11.6 Conclusions

Nonlinear regression modeling and analysis is a particularly tricky task. LLEPLS model transforms the nonlinear regression problem into a combination of multiple local linear regression problems using the local linear embedding feature. It not only

allows the local properties of the original data to be preserved, but also allows the correlation between the input space and the output space to be maximized, further accurately predicting the quality variables. While the T_{pc}^2 statistic of LLEPLS model contains the orthogonal variation of the output. In order to eliminate it, the input space of LLEPLS is further orthogonally decomposed, and the corresponding statistical criteria are established, i.e., LLEOPLS is obtained. The characteristics of the LLEOPLS model with nonlinear mapping and orthogonal decomposition are further clarified by comparing with the PLS, CPLS, and LLEPLS models in TEP benchmark simulation. Simulation results show that the LLEOPLS model is more effective for nonlinear systems and yields better (more consistent) fault detection performance, compared with the PLS, CPLS, and LLEPLS models. Although LLEOPLS has good quality-related monitoring performance for nonlinear processes, it has some limitations, such as that the low-dimensional manifold in which the sampled data are located is linear and that the noise subjects to Gaussian distribution. These are the directions of our further research.

References

- Ding S, Yin S, Peng K, Shen B (2013) A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *IEEE Trans Industr Electron* 9:2239
- Hu Y, Ma H, Shi H (2013) Enhanced batch process monitoring using just-in-time-learning based kernel partial least squares. *Chemom Intell Lab Syst* 123:15–27
- Kokiopoulou E, Saad Y (2007) Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. *IEEE Trans Pattern Anal Mach Intell* 29(12):2143–2156
- Kouropyteva O, Okun O, Pietikäinen M (2002) Selection of the optimal parameter value for the locally linear embedding algorithm. *FSKD* 2:359–363
- Kruger U, Chen Q, Sandoz DJ, McFarlane RC (2001) Extended PLS approach for enhanced condition monitoring of industrial processes. *AIChE J* 47(9):2076–2091
- Li G, Qin SJ, Zhou D (2010) Geometric properties of partial least squares for process monitoring. *Automatica* 46:204–210
- Lyman PR, Georgakis C (1995) Plant-wide control of the tennessee eastman problem. *Comput Chem Eng* 19:321–331
- Qin S, Zheng Y (2012) Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. *AIChE J* 59:496–504
- Song K, Wang H, Li P (2004) PLS-based optimal quality control model for TE process. In: 2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583), vol 2, pp 1354–1359
- Wang G, Yin S, Kaynak O (2014) An LWPR-based data-driven fault detection approach for nonlinear process monitoring. *IEEE Trans Industr Inf* 10(4):2016–2023
- Wang J, Zhong B, Zhou J (2017) Quality-relevant fault monitoring based on locality preserving partial least squares statistical models. *Ind Eng Chem Res* 56:7009–7020
- Yin S, Xie XC, Lam J, Cheung KC, Gao HJ (2016) An improved incremental learning approach for KPI prognosis of dynamic fuel cell system. *IEEE Trans Cybern* 46(12):3135–3144
- Yin S, Xie XC, Sun W (2017) A nonlinear process monitoring approach with locally weighted learning of available data. *IEEE Trans Industr Electron* 64(2):1507–1516
- Zhang Y, Qin SJ (2008) Improved nonlinear fault detection technique and statistical analysis. *AIChE J* 54:3207–3220

- Zhang Y, Sun R, Fan Y (2015) Fault diagnosis of nonlinear process based on KCPLS reconstruction. *Chemom Intell Lab Syst* 140:49–60
- Zhou JL, Zhang SL, Zhang H, Wang J (2018) A quality-related statistical process monitoring method based on global plus local projection to latent structures. *Ind Eng Chem Res* 57:5323–5337

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 12

New Robust Projection to Latent Structure



In many actual nonlinear systems, especially near the equilibrium point, linearity is the primary feature and nonlinearity is the secondary feature. For the system that deviates from the equilibrium point, the secondary nonlinearity or local structure feature can also be regarded as the small uncertainty part, just as the nonlinearity can be used to represent the uncertainty of a system (Wang et al. 2019). So this chapter also focuses on how to deal with the nonlinearity in PLS series method, but starts from an different view, i.e., robust PLS. Here the system nonlinearity is considered as uncertainty and a new robust L_1 -PLS is proposed.

The traditional PLS and its nonlinear improvement methods are usually to maximize the covariance between the input and output data, i.e., the square of L_2 norm. L_2 norm has the feature of clear physical meaning and convenient calculation, and its solution are unique unbiased and dense. While it is powerless for systems with rich local features such as nonlinear systems or uncertain systems. The proposed robust L_1 -PLS aims at the robustness of the feature extraction and the regression coefficients. This method maintains the signal relative size during the feature extraction. Moreover, it guarantees the features are robust to outliers in the global statistical view and sensitive to the local structure information.

12.1 Motivation of Robust L_1 -PLS

Many robust PLS methods have been developed to increase the robustness of traditional PLS method recently. Branden (2004) and Hubert (2008) replaced the empirical variance-covariance matrix in PLS by a robust covariance estimator, and used the minimum covariance determinant (MCD) estimator and the reweighted MCD estimator (RMCD) for low-dimensional data sets. Turkmen (2013) proposed the influence function analysis for the robust PLS estimator. Currently, the existing robust PLS methods use robust covariance estimation techniques with the identification of multivariate outliers to maintain robustness (Fortuna et al. 2007; Filzmoser 2016). These

methods actually perform with a potential assumption that the signal is subject to Gaussian distribution, which is not satisfied for many industrial processes. Usually the industrial data are full of lots of outliers and follow either heavy-tailed distribution (Doman'ski 2019) or multipeak distribution (Wang 2000). In other words, the statistical properties of this kind of data cannot be described by the robust covariance matrix estimation. Furthermore, outliers may contain very important information, so the outliers cannot be simply deleted or replaced (Liu et al. 2018). The data also have some nondominant local structure features besides the outliers. Robust covariance estimation methods also do not handle the small uncertainty correctly.

Recently, a robust PCA (RPCA) (Kwak 2008) and a robust sparse PCA (RSPCA) (Meng et al. 2012) were proposed, which the two methods maximized the L_1 norm rather than the square of L_2 norm of the input data. Experiments showed that they are efficient and robust for the data with inherent uncertainty and outliers. However, the two improved RPCA methods do not obtain any useful information from the output quality variables, so it is difficult to directly apply them to quality-relevant process monitoring and fault diagnosis (Zhou et al. 2018). The monitoring system will automatically alarm if a fault is detected whether it affects the product quality or not. Many alarms do not make sense for the final production quality.

It is known that the least absolute deviation (LAD) regression is often better than the least squares (LS) regression for non-Gaussian signals, especially those with a heavy-tailed distribution. While LAD regression is immune to outliers. Moreover, the solution of LAD regression is not unique, and it is necessary to introduce the optimal technique to obtain an optimal solution. So the LAD regression of high-dimensional system is a time-consuming task. To improve the efficiency of the LAD algorithm, the idea of partial least squares (PLS) regression is used to extend the conventional LAD regression to partial LAD regression. The PLS-based monitoring method decomposes the process space through the correlation between the quality and the process variables, which can reflect the quality-relevant product changes in the process variables (Wang et al. 2017; Zhou et al. 2018).

In order to enhance the robustness of the PLS method in a new way, this chapter proposes a novel dual robustness projection to latent structure regression method based on the L_1 norm, L_1 -PLS. The optimization objective during the principle components extraction in the PLS method is a square of L_2 norm, i.e., the least squares regression problem. L_1 -PLS use the L_1 norm maximization to replace the square of the L_2 norm maximization in the traditional PLS methods. The L_1 norm penalty terms are added to the direction vectors in the latent structure construction. Moreover, the partial LAD regression is used to obtain the regression coefficients. Therefore, the L_1 -PLS regression method achieves dual robust capabilities including robust principle components and regression coefficients. On the other hand, the L_1 norm optimization target also has the certain capability of local structural feature retention, compared with the L_2 norm optimization goal.

L_1 -PLS is distinguished from other existing robust PLS methods in several respects:

- (1) The noises, outliers, and local structure features generally enter the system through the direction vectors, and the L_1 norm can maintain the relative size of the original signal; its direction vectors are robust to outliers and contain more local structure features even if there is no preprocessing of outliers. This facilitates the L_1 norm to obtain the global and local features of the system at the same time without destroying the integrity of the samples;
- (2) The L_1 -PLS method with the L_1 norm penalty term to the direction vectors can obtain the sparse principle components, and filter out the disturbance variables or those sparse PCs that are robust to disturbance variables;
- (3) The regression coefficients are obtained by the partial LAD regression. The corresponding regression model is also robust to outliers or uncertainties, and the model has better predictive performance.

12.2 Introduction to RSPCA Method

Consider the input data $X = [x(1), \dots, x(n)] \in R^{m \times n}$, where $x = [x_i, \dots, x_m]$; m and n are the dimensionality of the input data and the size of the input matrix. The traditional PCA method aims to find the d ($d < m$) dimensional linear subspace with the largest input data variance. The objective function is as follows:

$$W^* = \arg \max \|W^T X\|_2^2, \text{ s.t. } W^T W = I_d, \quad (12.1)$$

where $W = [w_1^T, \dots, w_d^T]^T \in R^{m \times d}$ is weight matrix. $\|\cdot\|_2$ represents the L_2 norm of a matrix or vector.

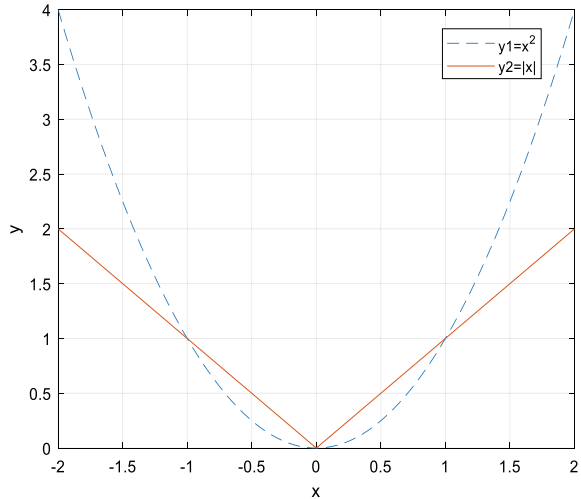
However, the principal components based on the PCA are usually a linear combination of the original variables usually with the non-zero weights. The non-zero weight results in that many irrelevant variables are included in the final model and cause unnecessary interference. Therefore, the sparse PCA (SPCA) method was proposed to achieve the sparse expression of the principal components as much as possible (Liu 2014). Its objective function is

$$W^* = \arg \max \|W^T X\|_2^2, \text{ s.t. } W^T W = I_d, \|W\|_1 < s, \quad (12.2)$$

where $\|\cdot\|_1$ is the L_1 norm of a matrix or vector. It is introduced as constraint or penalty term to enhance the sparsity of the principal components. s is the number of non-zero weights. The L_1 norm penalty term ($\|W\|_1 < s$) realizes the sparse expression of the direction vector.

Figure 12.1 shows the amplifying effect curve of L_1 norm and L_2 norm on noise. The blue dotted line is the square of the L_2 norm (for one-dimensional data, it is equivalent to the L_2 norm), and the red line is the L_1 norm. Obviously, the L_2 norm has an inhibitory effect on the data in $|x| \leq 1$ and has an enlarged effect on the data in $|x| > 1$. The L_1 norm maintains the relative size of the original data and has a

Fig. 12.1 The expanding effects of the L_1 norm and L_2 norm curve



relatively small expansion effect on all data. In order to further improve the robustness of SPCA, the RSPCA method is proposed to reduce the sensitivity of the principal components to outliers. The L_2 norm in the objective function is substituted by L_1 norm (Zou et al. 2006). The optimization function of RSPCA is given as follows:

$$w^* = \arg \max \|X^T w\|_1, \text{ s.t. } w^T w = 1, \|w\|_1 < s. \quad (12.3)$$

Here the optimization problem is a form of L_1 norm maximization with an L_1 norm penalty term simultaneously. In order to obtain the principal components of the RSPCA method, the optimal direction vector w^* is calculated by Algorithm 3.

The convergence of Algorithm 3 and the rationality of the obtained sparse direction vectors have been theoretically verified (Zou et al. 2006). However, Algorithm 3 indicates that the sparseness of the data needs to be given in prior during the calculation of the sparse direction vector. Generally speaking, the sparsity of input data is unknown and it contains uncertainty. More importantly, the RSPCA method cannot be directly applied to quality-related process monitoring. Therefore, this chapter introduces the L_1 norm into the PLS method.

12.3 Basic Principle of L_1 -PLS

The double robust projection to latent structure (L_1 -PLS) method is given based on the L_1 norm, aiming at improving the robustness of the traditional PLS method. The PLS method extracts principal components from the input space and output space, and the principal components should satisfy the following conditions: carry the maximum variation information (representation) of their respective variable spaces as much as

Algorithm 3 RSPCA algorithm for one sparse PC**Input:**Data matrix X , sparsity s .**Output:**The s sparse PC w^* .

- 1: Initialization $w(0) \in R^{1 \times m}$, set $w(0) = \frac{w(0)}{\|w(0)\|_2}$, and $k = 0$.
- 2: Let $v = (v_1, \dots, v_m)^T = \sum_{i=1}^n p_i(t) X_i$, where $p_i(k) = \begin{cases} 1, & w^T(k) X_i \geq 0 \\ -1, & w^T(k) X_i < 0 \end{cases}$ and X_i is the i th column of the matrix X . Let γ be the $(s + 1)$ largest element in $|v|$.
- 3: Let $\beta = (\beta_1, \dots, \beta_m)^T$, where $\beta_i = \text{sgn}(v_i)(|v_i| - \gamma)_+$, $i = 1, \dots, m$, and $(z)_+ = \begin{cases} z, & x > 0 \\ 0, & x \leq 0 \end{cases}$, $\text{sgn}(z) = \begin{cases} 1; & z > 0 \\ 0; & z = 0 \\ -1; & z < 0 \end{cases}$. Make $w(k + 1) = \frac{\beta}{\|\beta\|_2}$, and $k = k + 1$.
- 4: If $w(k) \neq w(k + 1)$, return to Step 2; otherwise continue to Step 5.
- 5: If there is i such that $w^T(k) X_i = 0$ and $\text{sgn}\left(\sum_{j=1}^m |w(k)_j X_{j,i}|\right) \neq 0$, then let $\frac{w^T(k) + \Delta w}{\|w^T(k) + \Delta w\|_2}$ and return to Step 2; otherwise continue to Step 6; Δw is a small non-zero random vector.
- 6: Set $w^* = w(k)$ and stop iteration.
- 7: **return** w^* ;

possible, and the degree of correlation between different variable spaces is as large as possible (correlation). Take the extraction of the first principal component as an example. The PLS method is expressed as follows:

$$\begin{aligned} E_0^T F_0 F_0^T E_0 w_1 &= \theta^2 w_1 \\ F_0^T E_0 E_0^T F_0 c_1 &= \theta^2 c_1, \end{aligned} \quad (12.4)$$

where w_1 and c_1 are the direction vector of the principle components t_1 and u_1 . The optimization problem (12.4) is transformed into finding the unit direction vectors w_1 and c_1 corresponding to the maximum eigenvalue θ^2 of matrices $E_0^T F_0 F_0^T E_0$ and $F_0^T E_0 E_0^T F_0$, respectively. It can be seen that the solution of (12.4) satisfies the requirements about the representation and correlation in PLS method.

Then, multiply both sides of the equation (12.4) by w_1^T and c_1^T , respectively, and obtain

$$\begin{aligned} w_1^T E_0^T F_0 F_0^T E_0 w_1 &= \theta^2, & \text{s.t. } w_1^T w_1 &= 1 \\ c_1^T F_0^T E_0 E_0^T F_0 c_1 &= \theta^2, & \text{s.t. } c_1^T c_1 &= 1. \end{aligned} \quad (12.5)$$

To simplify further, we can get

$$\begin{aligned} w_1^* &= \arg \max \|w_1^T E_0^T F_0\|_2^2, & \text{s.t. } w_1^T w_1 &= 1 \\ c_1^* &= \arg \max \|c_1^T F_0^T E_0\|_2^2, & \text{s.t. } c_1^T c_1 &= 1. \end{aligned} \quad (12.6)$$

The optimal problem of the traditional PLS (12.4) is expressed as L₂ norm optimization in (12.6). w_1^* and c_1^* are the optimal direction vectors.

It is known that the noise is flowed into the regression model through the direction vector (\mathbf{w}_1 and \mathbf{c}_1) in most cases, which affects the estimation of the regression parameters in the PLS method. Similar as the idea of equation (12.3), we replace the maximization of the L_2 norm in the objective function (12.6) with the maximization of L_1 norm. Moreover, the L_1 norm penalty term is added to the direction vector. Therefore, the objective function of the L_1 -PLS method based on the L_1 norm is given as follows:

$$\begin{aligned} \mathbf{w}_1^* &= \arg \max \left\| \mathbf{w}_1^T \mathbf{E}_0^T \mathbf{F}_0 \right\|_1, \quad \text{s.t.} \quad \mathbf{w}_1^T \mathbf{w}_1 = 1, \quad \|\mathbf{w}_1\|_1 < s_1 \\ \mathbf{c}_1^* &= \arg \max \left\| \mathbf{c}_1^T \mathbf{F}_0^T \mathbf{E}_0 \right\|_1, \quad \text{s.t.} \quad \mathbf{c}_1^T \mathbf{c}_1 = 1, \quad \|\mathbf{c}_1\|_1 < s_2, \end{aligned} \quad (12.7)$$

where s_1 and s_2 are the sparsity of input spatial data and output spatial data, respectively.

According to the above analysis, although the direction vectors (\mathbf{w}_1 and \mathbf{c}_1) in (12.4) contains the correlation between the input data \mathbf{E}_0 and the output data \mathbf{F}_0 , fortunately, they can be solved separately in (12.7). Therefore, Algorithm 3 also is suitable for the solution of (12.7) by replacing the corresponding input data matrix \mathbf{X} with $\mathbf{E}_0^T \mathbf{F}_0$ and $\mathbf{F}_0^T \mathbf{E}_0$, respectively. It is noted that the solution of \mathbf{w}_1 and \mathbf{c}_1 are independent but not jointed by Algorithm 3.

Once the optimal direction vectors \mathbf{w}_1 and \mathbf{c}_1 are obtained, the score vectors in the latent space, i.e., the first principle component pair, \mathbf{t}_1 and \mathbf{u}_1 can be calculated

$$\mathbf{t}_1 = \mathbf{E}_0 \mathbf{w}_1, \mathbf{u}_1 = \mathbf{F}_0 \mathbf{c}_1. \quad (12.8)$$

Next, the regression coefficients (loading vectors) of \mathbf{F}_0 and \mathbf{E}_0 to \mathbf{t}_1 will be established. In the traditional PLS model, the regression coefficients \mathbf{p}_1 and \mathbf{q}_1 are estimated by least squares, namely,

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{E}_0^T \mathbf{t}_1 / \|\mathbf{t}_1\|^2 \\ \mathbf{q}_1 &= \mathbf{F}_0^T \mathbf{t}_1 / \|\mathbf{t}_1\|^2. \end{aligned} \quad (12.9)$$

Similarly, least squares estimation is also susceptible to outliers, and the least absolute deviation (LAD) method is introduced to deal with this problem. Therefore, in order to further improve the robustness, LAD regression is used to solve the regression coefficients in the L_1 -PLS algorithm, namely,

$$\begin{aligned} \mathbf{p}_1^* &= \arg \min \left\| \mathbf{E}_0 - \mathbf{t}_1 \mathbf{p}_1^T \right\|_1 \\ \mathbf{q}_1^* &= \arg \min \left\| \mathbf{F}_0 - \mathbf{t}_1 \mathbf{q}_1^T \right\|_1, \end{aligned} \quad (12.10)$$

where \mathbf{p}_1^* and \mathbf{q}_1^* are the optimal loading vectors of (12.10).

Obviously, the essence of (12.10) is also the form of L_1 norm. When there are few outliers, it is not necessary to use the norm to solve the regression coefficient. Due to the direction vector has been solved by maximizing the L_1 norm, the influence of

the outlier has been reduced, and as can be seen from Fig. 12.1. When the outlier is small, the L₂ norm and the L₁ norm have the same effect.

Calculate the residual matrix E_1 and F_1 :

$$E_1 = E_0 - t_1 p_1^T, F_1 = F_0 - t_1 q_1^T \quad (12.11)$$

Similar as the extraction of the first principal components pair, the other principal components are calculated iteratively by decomposing the residuals E_i and F_i ($i = 1, \dots, d - 1$). The extraction of principal components is stopped until the model determined by the extracted principal components satisfies the desired requirements.

The dual robustness of the L₁-PLS algorithm is reflected in the following two aspects:

1. Different from the PLS algorithm, Algorithm 3 is used to calculate the direction vector each time. By maximizing the L₁ norm in the objective function, and adding the L₁ norm penalty term to the direction vector, the robustness of the L₁-PLS algorithm is improved. This achieves the robustness of principal component extraction.
2. In the case of many outliers, the regression coefficients can be calculated using least absolute estimation, which can overcome the shortcomings of least squares estimation that is easily affected by outliers, and further enhance the robustness of the L₁-PLS algorithm.

12.4 L₁-PLS-Based Process Monitoring

It is found that only the calculation process of the direction vector w_1 and c_1 (12.7) or the regression coefficient p_1 and q_1 (12.10) is improved in the L₁-PLS method, and other steps are not affected. Therefore, the monitoring process based on the L₁-PLS method is the same as the PLS method. In the process monitoring based on the L₁-PLS method, the T² and T_e² statistics are still used to monitor the principal component subspace and the remaining subspace. Then, the L₁-PLS-based monitoring is described in detail in Algorithm 4 (offline process training) and Algorithm 5 (online process monitoring). The corresponding flowchart is shown in Fig. 12.2.

In Algorithms 4 and 5, Λ and Λ_e represent the sample covariance matrix. The non-parametric kernel density estimation (KDE) method (1.33) is used to estimate the corresponding control limits of T² and T_e².

There is still a key problem in the implementation of Algorithm 4: the sparsity degree s_1 and s_2 need to be given in prior. There are two common strategies to determine s_1 and s_2 . (1) The first one is the variable importance in prediction (VIP) method (Farrés et al. 2015). It judges whether the variable is an irrelevant variable based on the VIP score of the j th predicted value of the response variable. Usually, the “greater than ϵ ” criterion is used as the selection criterion. More precisely, the threshold ϵ

Algorithm 4 L_1 -PLS method for Offline process training

Input:

Normal data sets $\mathbf{X} = [x_1, \dots, x_m] \in R^{n \times m}$, $\mathbf{Y} = [y_1, \dots, y_l] \in R^{n \times l}$, sparsity s_1 and s_2 .

Output:

The control limits T_{lim}^2 and $T_{e,\text{lim}}^2$.

(1) Normalized \mathbf{X} and \mathbf{Y} as \mathbf{E}_0 and \mathbf{F}_0 ,

(2) For $i = 1, \dots, d$ (d is obtained by cross-validation):

(2.1) Apply Algorithm 3 to the projected matrices $\mathbf{E}_{i-1}^T \mathbf{F}_{i-1}$ and $\mathbf{F}_{i-1}^T \mathbf{E}_{i-1}$ to get the direction vectors \mathbf{w}_i and \mathbf{c}_i , respectively.

(2.2) Calculate the score vectors: $\mathbf{t}_i = \mathbf{E}_{i-1} \mathbf{w}_i$, $\mathbf{u}_i = \mathbf{F}_{i-1} \mathbf{c}_i$.

(2.3) Calculate the load vectors:

$$\mathbf{p}_1 = \mathbf{E}_0^T \mathbf{t}_1 / \|\mathbf{t}_1\|^2 \quad \text{or} \quad \mathbf{p}_1^* = \arg \min \|\mathbf{E}_0 - \mathbf{t}_1 \mathbf{p}_1^T\|_1$$

$$\mathbf{q}_1 = \mathbf{F}_0^T \mathbf{u}_1 / \|\mathbf{u}_1\|^2 \quad \text{or} \quad \mathbf{q}_1^* = \arg \min \|\mathbf{F}_0 - \mathbf{u}_1 \mathbf{q}_1^T\|_1$$

(2.4) Calculate the Residual matrix: $\mathbf{E}_i = \mathbf{E}_{i-1} - \mathbf{t}_i \mathbf{p}_i^T$, $\mathbf{F}_i = \mathbf{F}_{i-1} - \mathbf{u}_i \mathbf{q}_i^T$.

(3) Describe \mathbf{t}_i with the original matrix \mathbf{E}_0 : $\mathbf{T} = \mathbf{E}_0 \mathbf{R}$,

$$\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_d], \text{ in which } \mathbf{r}_i = \prod_{j=1}^{i-1} (\mathbf{I}_n - \mathbf{w}_j \mathbf{p}_j^T) \mathbf{w}_i.$$

$$\hat{\mathbf{E}} = \mathbf{T} \mathbf{P}^T = \mathbf{E}_0 \mathbf{R} \mathbf{P}^T$$

$$\bar{\mathbf{E}} = \mathbf{E}_0 - \hat{\mathbf{E}} = \mathbf{E}_0 (\mathbf{I}_n - \mathbf{R} \mathbf{P}^T)$$

(4) For a normalized data sample \mathbf{x} , calculate its estimate, residual and the corresponding PC value.

$$\hat{\mathbf{x}} = \mathbf{R} \mathbf{P}^T \mathbf{x}$$

$$\mathbf{t} = \mathbf{R} \mathbf{x}$$

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} = (\mathbf{I} - \mathbf{R} \mathbf{P}^T) \mathbf{x}$$

(5) Calculate the statistics T^2 and T_e^2 :

$$T^2 := \mathbf{t} \mathbf{A}^{-1} \mathbf{t}^T = \mathbf{t} \left(\frac{1}{n-1} \mathbf{T}^T \mathbf{T} \right)^{-1} \mathbf{t}^T$$

$$T_e^2 := \mathbf{e} \mathbf{A}_e^{-1} \mathbf{e}^T = \mathbf{e} \left(\frac{1}{n-1} \bar{\mathbf{E}}^T \bar{\mathbf{E}} \right)^{-1} \mathbf{e}^T$$

return T_{lim}^2 and $T_{e,\text{lim}}^2$;

should be adjusted based on the distribution of the overall data in different situations. (2) The second strategy is the selectivity ratio method (Branden and Hubert 2004). The variable selection ratio is calculated according to the ratio of the interpretation of the \mathbf{X} variable on the \mathbf{Y} target projection component to the residual variance. Then F test is performed to define the boundary between important variables and irrelevant variables. Since the VIP method is simple and easy to implement, the VIP method is selected to determine the sparsity s_1 and s_2 here.

Algorithm 5 L₁-PLS method for Online process monitoring**Input:**New normalized data sets \mathbf{x}_{new} and \mathbf{y}_{new} .**Output:**

Online process monitoring results.

(1) Calculate the new score vectors: $\mathbf{t}_{new} = \mathbf{x}_{new} \mathbf{R}$.

(2) Calculate the new prediction matrix and new residual:

$$\begin{aligned}\tilde{\mathbf{x}}_{new} &= \mathbf{t}_{new} \mathbf{P}^T = \mathbf{x}_{new} \mathbf{R} \mathbf{P}^T \\ \mathbf{e}_{new} &= \mathbf{x}_{new} - \tilde{\mathbf{x}}_{new} = \mathbf{x}_{new} (\mathbf{I}_n - \mathbf{R} \mathbf{P}^T).\end{aligned}$$

(3) Calculate the new statistics T_{new}^2 and $T_{e,new}^2$:

$$\begin{aligned}T_{new}^2 &= \mathbf{t}_{new} \mathbf{\Lambda}^{-1} \mathbf{t}_{new}^T = \mathbf{t}_{new} \left\{ \frac{1}{n-1} \mathbf{T}^T \mathbf{T} \right\}^{-1} \mathbf{t}_{new}^T \\ T_{e,new}^2 &:= \mathbf{e}_{new} \mathbf{A}_e^{-1} \mathbf{e}_{new}^T = \mathbf{e}_{new} \left\{ \frac{1}{n-1} \bar{\mathbf{E}}^T \bar{\mathbf{E}} \right\}^{-1} \mathbf{e}_{new}^T\end{aligned}$$

(4) Compare T_{new}^2 and $T_{e,new}^2$ with the corresponding control limits T_{lim}^2 and $T_{e,lim}^2$.**return** Online process monitoring results.

It is worth noting that the role of sparsity is to achieve variable selection. If the established system model contains many irrelevant variables, giving the sparsity is helpful to limit the number of irrelevant variables, so as to realize L₁-sparse-PLS. However, if the sparsity of the input data is uncertain, the sparsity degree s_1 and s_2 can be set equal to the variable number in the input and output space, respectively, to eliminate the uncertainty caused by the sparsification. In this view, the proposed L₁-PLS method is uniformly called as L₁-(S)PLS method based on the different sparsity.

12.5 TE Simulation Analysis

In this simulation, the input variable X is composed of 31 variables [XMEAS(1:22)] and [XMV(1:11) (except XMV(5) and XMV(9))]. The output variable Y consists of the quality components G (XMEAS(35)) and H (XMEAS(36)). Two simulation examples are used to verify the effectiveness of the L₁-PLS method for fault detection.

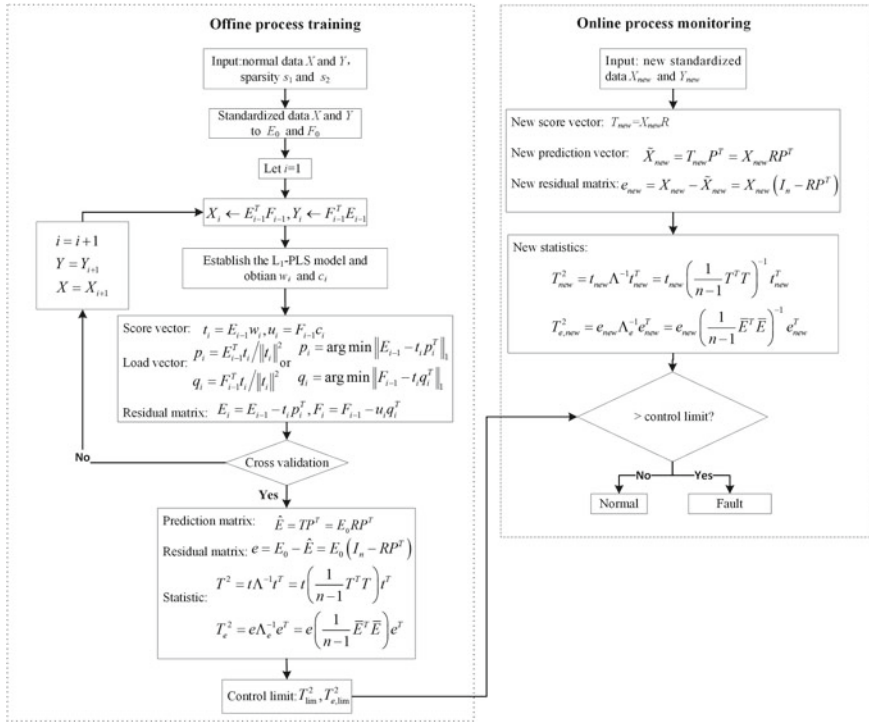


Fig. 12.2 The Flow chart of Algorithms 4 and 5

12.5.1 Robustness of Principal Components

The robustness of the L_1 -PLS method is mainly implemented on the direction vectors, which directly reflects the robustness of the PCs. The variation of the PC structure caused by outliers therefore is the focus of robustness analysis. Here results of PLS and RPLS methods are given for comparison. The input and output data ($X \in R^{960 \times 31}, Y \in R^{960 \times 2}$) are sampled from the TE process under the normal operation for training data. In order to test further the proposed L_1 -PLS, the outliers are added in the input space in the following form:

$$X(k) = X^*(k) + \Xi_j(k), \quad (12.12)$$

where $X^*(k)$ is the k th normal sample ($k = 1, 2, \dots, 960$) Ξ_j is the j -th randomly generated outlier that obey Gaussian distribution $\Xi_j \sim N(0, 2000)$. For ease of verification, three kinds of repeatable outliers that are generated using a specific random seed are added to the training set,

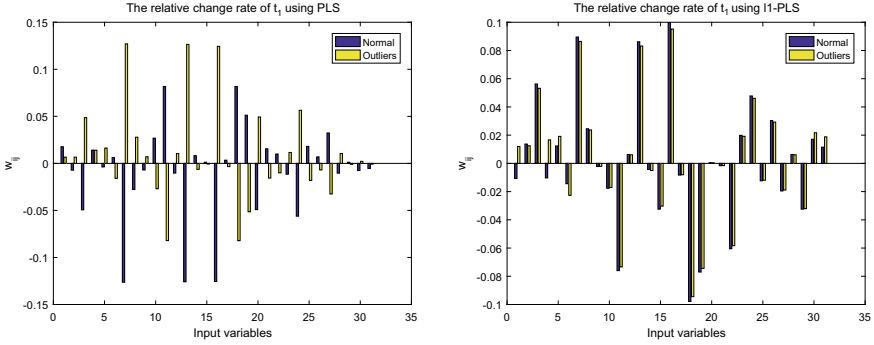


Fig. 12.3 The relative change rates of t_1 using PLS and L_1 -PLS

$$\begin{aligned}
 \mathcal{E}_1(12) &= [-71.294, 4.929, 35.199, -0.100]^T && \text{for } x_{14:17} \\
 \mathcal{E}_2(140) &= [4.164, -16.912, -66.307]^T && \text{for } x_{29:31} \\
 \mathcal{E}_3(200) &= [-1.960, 42.969, 77.737, -19.239, -72.776, 7.439]^T && \text{for } x_{1:6}.
 \end{aligned}$$

Outlier $\mathcal{E}_1(12)$ means that only the 14, 15, 16, and 17th variables at the 12th sample time $X(12)$ are abnormal, and the other variables at other sample times are still normal. The other two outliers have similar meanings.

The sparsity s_1 and s_2 in the L_1 -PLS method are set to 31 and 2. The sparsity is equal to the variable number of input and output space, respectively. In other words, the L_1 -PLS method can reflect the changes in all variables. The components numbers d are determined using cross-validation. They are 6, 6, and 2 for PLS, RPLS, and L_1 -PLS methods, respectively. The principle components are $t_i = \sum_{j=1}^n w_{ij}x_j, i = 1, \dots, d$, in which w_{ij} is the j th element of r_i . The coefficients w_{ij} are used to reflect whether the outliers affect the principle components. The relative rates of change (RRC) indices are defined as follows:

$$\begin{aligned}
 RRC_{1,i} &= \max\{|w_{ij,normal} - w_{ij,outliers}|\} \\
 RRC_{2,i} &= ||w_{i,normal} - w_{i,outliers}||_1,
 \end{aligned} \tag{12.13}$$

where $w_{i,normal} = [w_{ij}]_{normal}$ and $w_{i,outliers} = [w_{ij}]_{outliers}$ are the normalized coefficient vectors with normal samples and adding outliers samples for the i_{th} PC, respectively.

RRC_1 represents the maximum absolute deviation of the two coefficient sets, which indicates the worst changes of the normalized w_{ij} . RRC_2 represents the sum of the absolute deviations of the two coefficient sets, which indicates the overall change of the normalized w_{ij} .

The normalized coefficient w_{ij} values of the first two PCs (t_1 and t_2) of the PLS, RPLS and L_1 -PLS methods are shown in Figs. 12.3 and 12.4. The corresponding indices $RRC_i, i = 1, 2$ are given in Table 12.1 (a smaller value is better).

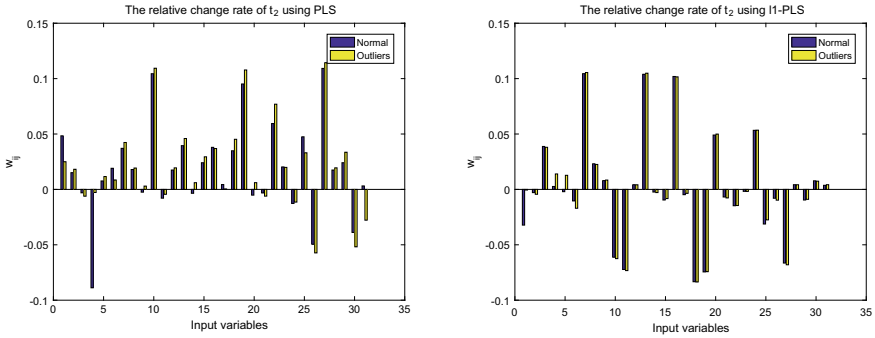


Fig. 12.4 The relative change rates of t_2 using PLS and L_1 -PLS

Table 12.1 RRC_i of t_1 and t_2 of the PLS, L_1 -PLS and L_1 -SPLS methods

	PLS		RPLS		L_1 -PLS	
	t_1	t_2	t_1	t_2	t_1	t_2
RRC_1	0.919	0.349	0.005	0.040	0.110	0.121
RRC_2	7.093	1.247	0.016	0.130	0.368	0.329

It can be seen from Figs. 12.3–12.4 and Table 12.1 that no matter which method is used, the outliers will always affect the structure of the PCs to some extent. In general, the outliers have a large adverse effect on the PCs extraction of the PLS method, and thus results in the largest change in its PC structures. With the robust covariance estimation method, the outliers have little effect on the PCs extraction of the RPLS method. L_1 -PLS method only uses the L_1 norm to be insensitive to outliers, without any outliers processing. Outliers that cause changes in the structure of its two PCs are nearly identical and within an acceptable range, whether in the RRC_1 or RRC_2 . The samples considered to be outliers may be a true reflection of the system state when the data set follows a heavy-tailed distribution (Doman’ski 2019). It is more important to retain all the samples to extract the PCs, although the outliers have a certain influence on the direction vectors.

By further analyzing the structure of t_1 and t_2 , it can be easily found that the extracted PCs by those methods are quite different. In order to better explain the structural differences of t_1 and t_2 in different methods, IDV(14) is taken as an example for in-depth analysis. The typical process variable monitoring results of IDV(14) are given in Fig. 12.5, in which, x_9 , x_{21} and x_{30} have similar monitoring results. Among the t_1 and t_2 , the sum of the absolute weights for x_9 , x_{21} and x_{30} of the PLS method (0.062) is more than twice that of the L_1 -PLS method (0.025).

These weight differences do not significantly affect the output prediction and the monitoring performance in the normal operation. But these differences are amplified in the fault modes. For example, consider the monitoring under the fault modes IDV(14) and IDV(17). The role of x_{21} and x_{30} (especially x_{30}) in the PLS method is

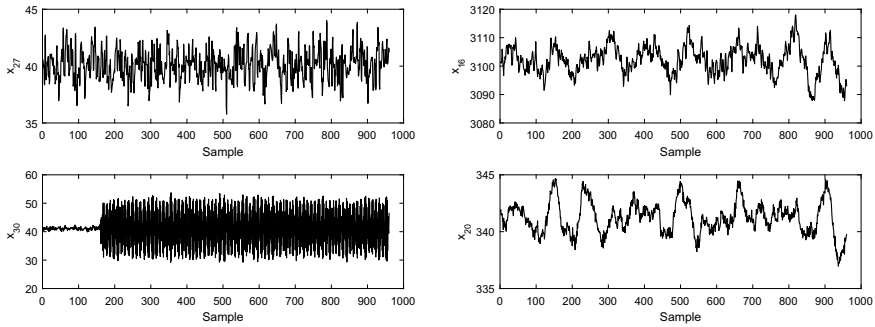


Fig. 12.5 Typical process variable monitoring results of IDV(14)

exaggerated, leading to incorrect predictions and quality-relevant monitoring results (see Figs. 12.6 and 12.7). Correspondingly, the L_1 norm can better maintain the relative size of those variables, therefore, the role of x_{21} and x_{30} in the extracted PCs is not exaggerated. In other words, the extracted PCs by the L_1 norm better capture the relationship between the input space and output space.

12.5.2 Robustness of Prediction and Monitoring Performance

The robustness of the principal components of the L_1 -PLS method is discussed in the previous section. But the number of principal components of the three methods is different, which only reflects one aspect of the robustness. Now, the robustness of prediction performance and monitoring is analyzed further, especially the prediction performance directly reflects the quality of the model. There are 21 types of faults in the TE process. The fault IDV(21) is a fault that the output drifts slowly, caused by the constant change of the steam valve position. So it does not reflect the robustness of the model. Therefore, the first 20 faults are analyzed in this simulation experiment. In this simulation, the sparsity in the L_1 -SPLS model is determined by the VIP method: input space $s_1 = 14$, output space $s_2 = 2$.

Experiment 1: Prediction Performance Analysis

In this experiment, the L_1 -PLS model shows good output prediction results for the 20 fault data sets. L_1 -PLS(outliers) and PLS(outliers) mean that the two models are trained by the normal operation data with adding outliers, described in previous Sect. 12.5.1. In order to illustrate the above conclusions more clearly, four faults IDV(7), IDV(14), IDV(17), and IDV(18) are selected to compare the prediction performance of the PLS model and the L_1 -PLS model. The output prediction results are good for all fault modes, but the four faults come from four different fault types, and the results of the L_1 -PLS model and the PLS model are quite different. Figures 12.6 and 12.7 give the output prediction results of the fault IDV(7), IDV(14), IDV(17),

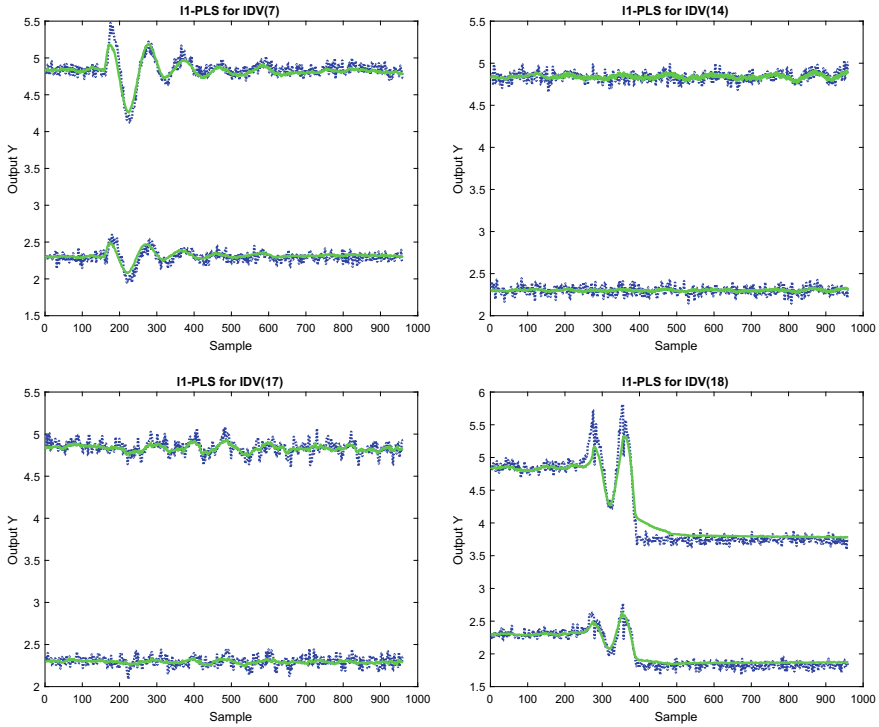


Fig. 12.6 Output predicted values for IDV(7), IDV(14), IDV(17), and IDV(18) using PLS(outliers)

and IDV(18). The horizontal axis represents data samples, and the vertical axis represents output values. The blue dashed line is the actual output value, and the green is the predicted output value.

In these prediction and monitoring diagrams, the first 160 samples are normal data, and the last 800 samples are data under different fault modes. The output prediction of fault IDV(7) shows a consistent conclusion under the step-change fault. The feedback controller or cascade controller reduces the impact of faults and abnormal values on product quality. For the other three types of fault IDV(14), IDV(17), and IDV(18), there are some differences in their output prediction results. When the system is under the normal operation, the PLS and L_1 -PLS models have the same good prediction results. However, after adding outliers, the PLS method cannot accurately predict the output (Fig. 12.6), while the L_1 -PLS method still quickly detects the output changes and makes correct predictions (Fig. 12.7). In particular, for faults IDV(17) and IDV(18), the PLS method gives a serious wrong predictions. Experiments show that the prediction performance of the L_1 -PLS method is better than PLS. Even if the data is contaminated by outliers, L_1 -PLS can still predict the output accurately. In other words, the L_1 -PLS model has stronger robust prediction performance.

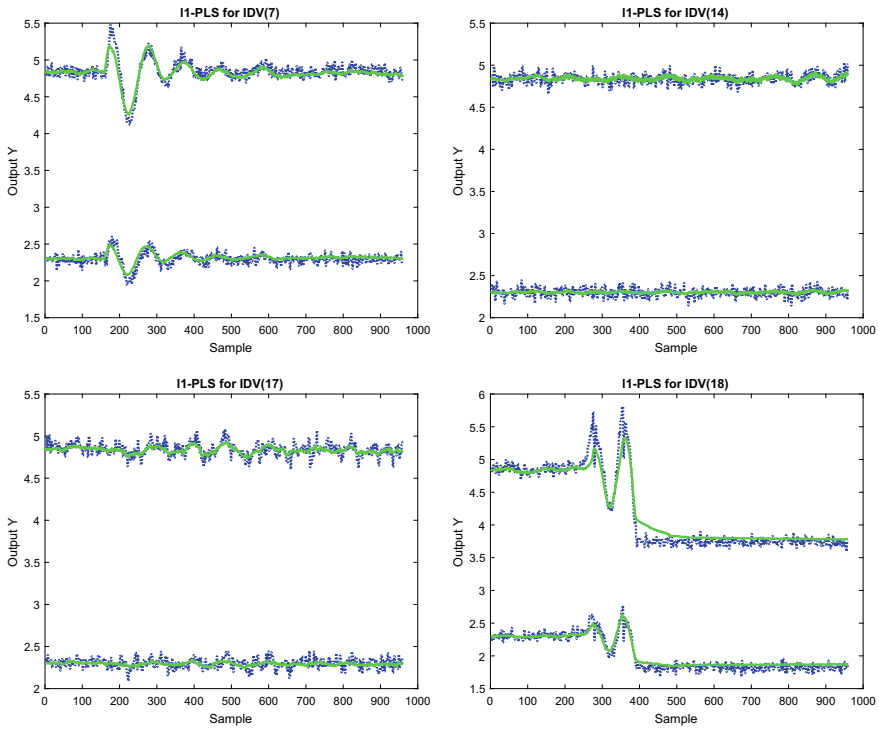


Fig. 12.7 Output predicted values for IDV(7), IDV(14), IDV(17), and IDV(18) using L_1 -PLS(outliers)

Experiment 2: Monitoring Performance Analysis

The robustness of monitoring performance is mainly verified by the accuracy of fault detection. The detection indices are FDR and FAR (4.1), the control limit is calculated with the confidence level 99.75% for both PLS and L_1 -PLS methods. The FAR results of the two models are basically same, this indicates that the proposed L_1 -PLS method does not increase the risk of false alarms, so it is not analyzed in this section. Table 12.2 lists the FDR results of the first 20 faults without adding outliers, corresponding to the models PLS, L_1 -PLS and L_1 -SPLS respectively. Table 12.3 shows the FDR results of 20 faults after adding outliers, corresponding to the models PLS (outliers), L_1 -PLS (outliers), and L_1 -SPLS (outliers).

For serious quality-related faults IDV(2), IDV(6), IDV(8), IDV(12), IDV(13), and IDV(18), the six models give consistent results. Therefore, these faults are not analyzed in this chapter. For other types of faults, their results are very different, including the quality-irrelevant faults, the quality-recoverable faults, and slight quality-related faults. The detailed analysis of the three situations is given below. In the monitoring figures of this section, the blue line represents the value of the statistic, where the

Table 12.2 FDRs of PLS, L₁-PLS, and L1-SPLS

IDV	PLS		L ₁ -PLS		L1-SPLS	
	T ²	T _e ²	T ²	T _e ²	T ²	T _e ²
1	99.63	99.75	60.00	99.75	31.38	99.75
2	98.50	98.25	98.25	98.38	98.25	98.38
3	1.00	1.38	0.75	1.75	0.50	1.75
4	19.13	100.00	0.88	100.00	0.00	100.00
5	22.00	100.00	18.38	100.00	17.13	100.00
6	99.25	100.00	98.38	100.00	98.13	100.00
7	100.00	100.00	68.75	100.00	31.38	100.00
8	96.00	97.88	89.00	97.88	88.50	97.88
9	0.50	1.13	0.25	1.38	0.38	1.38
10	26.38	84.25	19.13	85.38	15.63	85.38
11	26.63	76.50	1.13	77.88	0.88	77.88
12	97.50	99.88	84.00	99.88	84.00	99.88
13	94.88	95.13	82.13	95.25	82.25	95.25
14	91.50	100.00	0.38	100.00	0.00	100.00
15	1.25	2.63	1.00	3.75	0.63	3.75
16	20.13	42.75	9.00	46.13	7.00	46.13
17	77.38	96.75	10.00	97.00	1.63	97.00
18	89.38	90.13	88.75	90.13	88.75	90.13
19	0.50	34.50	0.13	37.88	0.00	37.88
20	30.50	90.50	20.75	90.38	19.00	90.38

upper curve is T², and the lower is T_e². The system alarms if the blue line exceeds the red control limit.

Case 1: Quality Irrelevant Fault

It can be found from Table 12.2 that very low alarm values are given for faults IDV(3), IDV(9), IDV(15), and IDV(19). However, the alarm values of the L₁-PLS and L₁-SPLS models are lower, which indicates that fewer false alarms will occur during the monitoring. It can also be seen from the corresponding Figs. 12.8, 12.9, 12.10, 12.11, and 12.12, the alarm points of the latter two models are much less. For faults IDV(4), IDV(11), and IDV(14), they are all related to the reactor cooling water and hardly affect the quality of output products. The PLS model gives a higher alarm value, which may lead to serious false alarms, while the L₁-PLS model effectively avoids these alarms and reduces the number of false alarms. In addition, the L₁-PLS model eliminates most of the false alarms in the monitoring Figs. 12.8, 12.9, 12.10, and the L1-SPLS model almost eliminates all false alarms.

When adding outliers, the PLS model provides the same wrong results for quality-irrelevant faults. The specific FDR values are shown in Table 12.3. However, the

Table 12.3 FDRs of PLS(outliers), L₁-PLS(outliers), and L₁-SPLS(outliers)

IDV	PLS(outliers)		L ₁ -PLS(outliers)		L ₁ -SPLS(outliers)	
	T ²	T _e ²	T ²	T _e ²	T ²	T _e ²
1	99.88	99.75	28.38	99.75	36.63	99.75
2	98.63	98.25	98.00	98.25	98.25	98.25
3	3.25	0.88	0.13	1.13	0.63	1.13
4	7.63	100.00	0.25	100.00	0.00	100.00
5	24.88	27.88	14.75	28.38	16.88	28.38
6	99.75	100.00	98.38	100.00	98.25	100.00
7	100.00	100.00	59.88	100.00	29.50	100.00
8	96.50	97.75	84.50	97.88	88.00	97.88
9	0.88	0.88	0.00	1.00	0.38	1.00
10	37.50	77.63	11.00	80.50	15.25	80.50
11	16.00	73.75	0.50	74.75	0.88	74.75
12	95.88	99.25	78.50	99.25	83.63	99.25
13	95.50	95.00	80.25	95.25	82.00	95.25
14	89.75	100.00	0.00	100.00	0.00	100.00
15	4.38	0.50	0.13	0.88	0.75	0.88
16	33.88	28.38	4.50	35.13	6.25	35.13
17	76.88	96.63	6.13	96.63	1.50	96.63
18	90.00	89.88	88.00	89.88	88.63	89.88
19	1.13	28.38	0.00	30.00	0.00	30.00
20	36.50	77.13	15.63	77.75	19.75	77.75

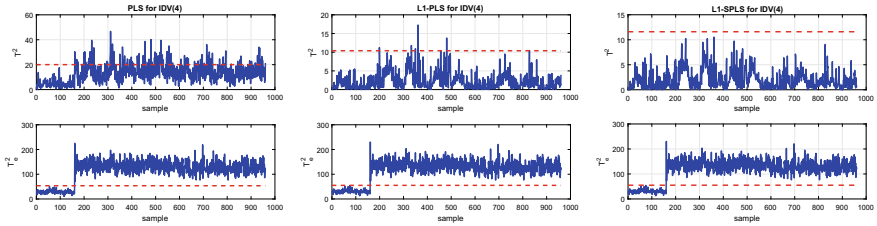


Fig. 12.8 PLS, L₁-PLS and L₁-SPLS monitoring results for IDV(4)

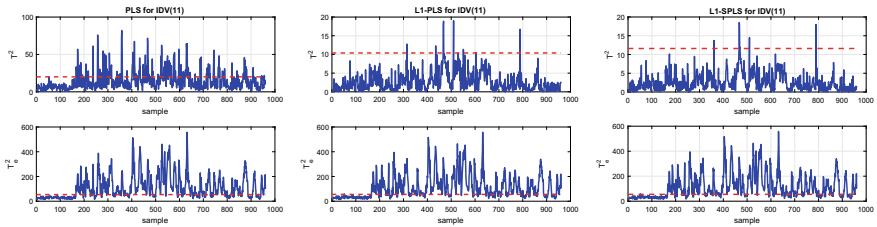


Fig. 12.9 PLS, L₁-PLS and L₁-SPLS monitoring results for IDV(11)

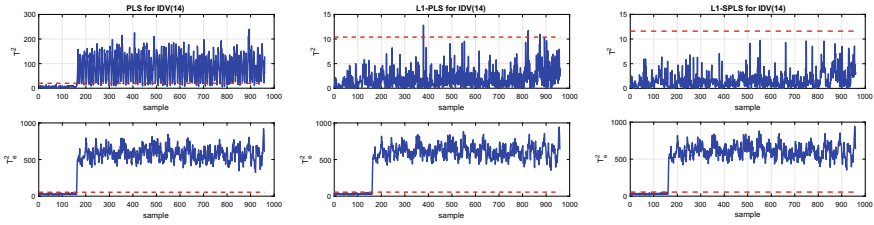


Fig. 12.10 PLS, L₁-PLS and L₁-SPLS monitoring results for IDV(14)

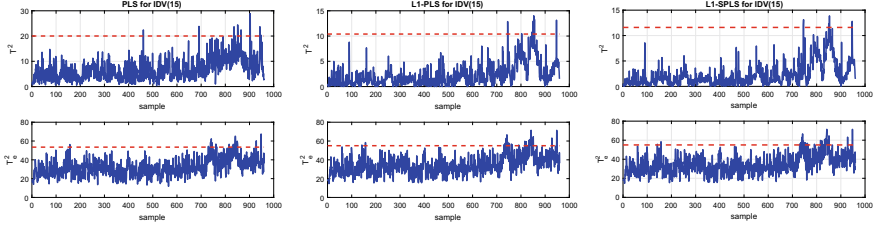


Fig. 12.11 PLS, L₁-PLS and L₁-SPLS monitoring results for IDV(15)

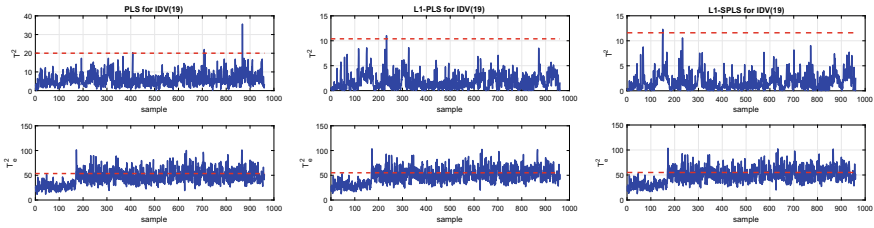


Fig. 12.12 PLS, L₁-PLS and L₁-SPLS monitoring results for IDV(19)

monitoring effect of the L₁-PLS model is still very good, for fault IDV(9), IDV(14), and IDV(19). The detection rate has been reduced to 0, which means that false alarms are completely eliminated in these cases. Therefore, the L₁-(S)PLS model will not interfere with the fault monitoring results after adding outliers. It should be noted that the monitoring performance of the L₁-PLS model after adding outliers (Table 12.3) is better than the normal conditions (Table 12.2). The possible reason is outliers, and the total noise in the input data becomes larger. The L₁-PLS method can filter out noise more effectively during the modeling. Therefore, the established model is more accurate and the monitoring performance is improved.

Case 2: Quality-Recoverable Fault

Faults IDV(1), IDV(5), and IDV(7) are quality-recoverable faults. The prediction value should tend to return to normal, but the statistic should be kept at a higher value. Figure 12.13 shows the monitoring results of the three models on the fault IDV(1). It can be seen that both the L₁-PLS and L₁-SPLS model methods give the

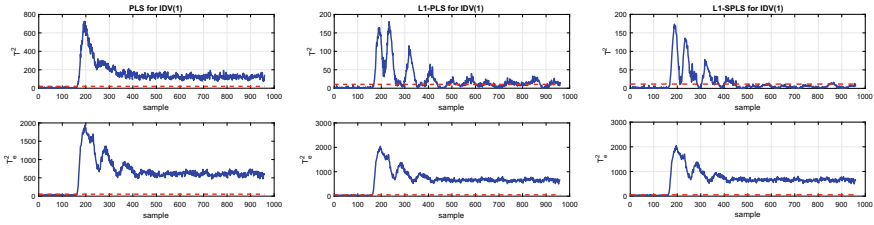


Fig. 12.13 PLS, L_1 -PLS and L_1 -SPLS monitoring results for IDV(1)

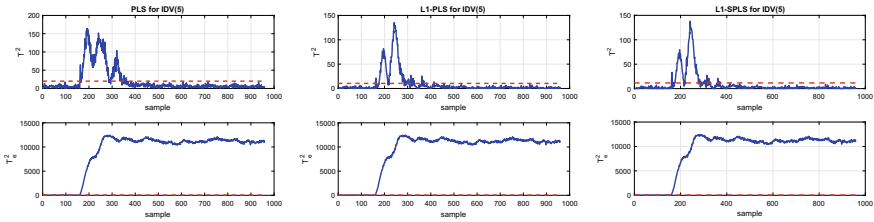


Fig. 12.14 PLS, L_1 -PLS and L_1 -SPLS monitoring results for IDV(5)

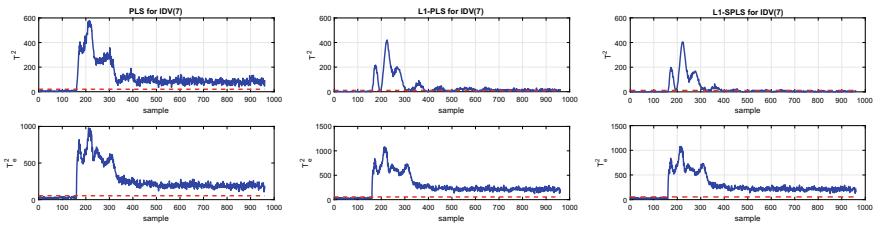


Fig. 12.15 PLS, L_1 -PLS and L_1 -SPLS monitoring results for IDV(7)

correct alarm results. In the PLS model, the value of the statistic exceeds the control limit, so a false alarm is generated in the process monitoring. For the fault IDV(5), it is also a process-related fault. It can be seen from Tables 12.2 and 12.3 that the fault detection rates of the L_1 -PLS and L_1 -SPLS models are lower than the PLS model, which means that the monitoring results are more accurate. Figures 12.14 and 12.16, respectively, show the monitoring diagrams of the three models for the fault IDV(5) in the normal case (without adding outliers) and with adding outliers. For fault IDV(7), the corresponding monitoring results are shown in Fig. 12.15. The PLS model gives completely wrong result, while the results of the other two models are more accurate.

The detection result for fault IDV(1) obtained by the L_1 -PLS (outliers) model seems to be better than the L_1 -PLS model, and the monitoring results are more reasonable. In addition, for the fault IDV (5), although the monitoring results of the L_1 -PLS and L_1 -SPLS (outliers) models may not be ideal, as shown in Fig. 12.14. The T_e^2 statistics of the L_1 -PLS and L_1 -SPLS models can detect the input space

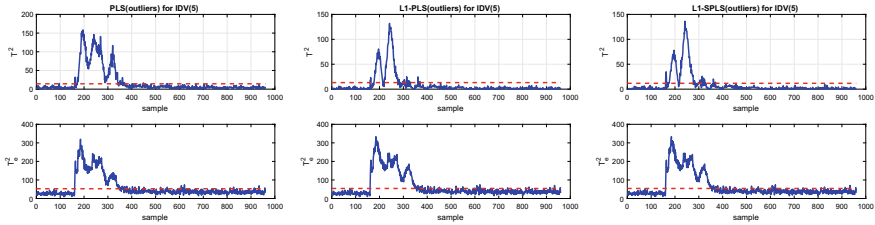


Fig. 12.16 PLS(outliers), L_1 -PLS(outliers) and L_1 -SPLS(outliers) monitoring results for IDV(5)

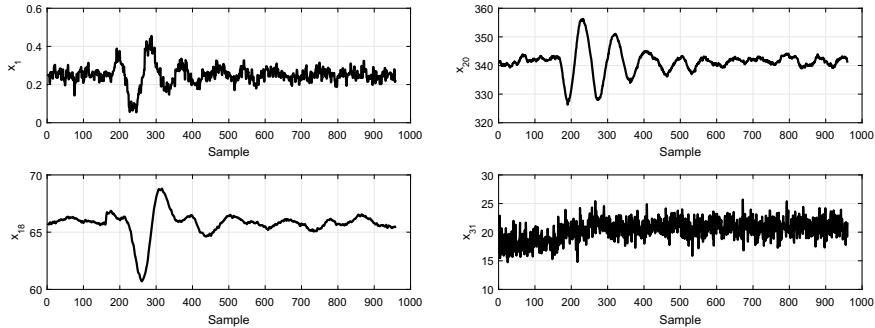


Fig. 12.17 Typical process variable monitoring results of IDV(5)

process-related faults. But the PLS (outliers), L_1 -PLS (outliers) and L_1 -SPLS (outliers) models gave wrong results (Fig. 12.16).

There are two possible reasons for this phenomenon. Firstly, the outliers were added directly without being regulated by the dynamic system, so its influence on the extraction of the principal components cannot be determined directly. Secondly, the typical process dynamics corresponding to fault IDV(5) is shown in Fig. 12.17. Only the variable 31 is a step change in all the monitored variables, and the rest gradually returns to the normal under the action of controller. In terms of the composition of the principal components, the contribution of variable 31 to the principal components is small. Therefore, its role is more in the residual space in the normal case (without adding outliers). After the outlier is added, its contribution to the principal component increases, which means its role in the residual space is weakened. It in turn causes the monitoring indicators in the residual space to return back to normal. On the other hand, the percentage of its contribution to the principal component is still small, so the monitoring indicators on the principal metric space also do not significantly reflect its characteristics.

Case 2: Slight Quality Related Fault

Fault IDV (16) and IDV (17) have a slight impact on quality, which means that they have almost no impact on output quality. Figure 12.18 shows the monitoring results of the three models after adding outliers. The fault monitoring results of PLS (outliers)

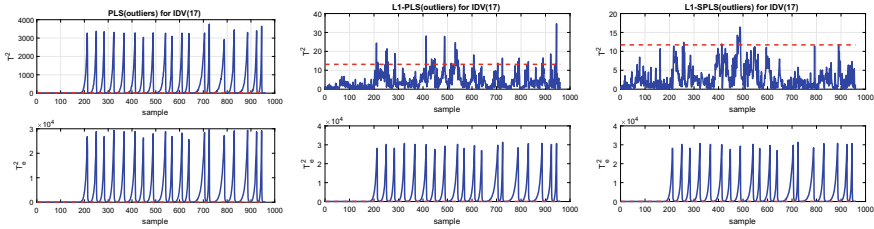


Fig. 12.18 PLS(outliers), L_1 -PLS(outliers), and L_1 -SPLS(outliers) monitoring results for IDV(17)

model is very bad, there have been many false positives. The L_1 -PLS (outliers) model and L_1 -SPLS (outliers) model effectively reduce these false alarms. It can also be seen from the corresponding FDR that the monitoring results of the L_1 -PLS (outliers) model and the L_1 -SPLS (outliers) model are more reasonable.

It can be seen from the above comparison results that even if outliers are added to the input data, the monitoring results of the L_1 -(S)PLS model have also been greatly improved. In other words, the L_1 -(S)PLS model improves the robustness performance and fault detection performance.

12.6 Conclusions

This chapter proposes a quality-related statistical monitoring method of double robust projection to latent structure (L_1 -PLS), which enhances the robustness of the PLS algorithm from two aspects. On the one hand, the L_1 -PLS method replaces the L_2 norm in the objective function with the L_1 norm, and adds the L_1 norm penalty term to the direction vector; On the other hand, the regression coefficient of the L_1 -PLS algorithm can also be obtained by the L_1 norm. Therefore, the L_1 -PLS algorithm has double robustness. Then a monitoring model based on the L_1 -PLS method is established, the robust performance and monitoring performance are verified on the TE process simulation platform. The results show that the L_1 -PLS method has better robustness and better performance in process monitoring and fault diagnosis.

References

- Branden KV, Hubert M (2004) Robustness properties of a robust partial least squares regression method. *Anal Chim Acta* 515:229–241
- Doman'ski PD (2019) Control quality assessment using fractal persistence measures. *ISA Transactions*
- Farrés M, Platikanov S, Tsakovski S, Tauler R (2015) Comparison of the variable importance in projection (VIP) and of the selectivity ratio (SR) methods for variable selection and interpretation. *J Chemom* 29(10):528–536

- Filzmoser P (2016) Identification of multivariate outliers: a performance study. *Austrian J Stat* 34:127–138
- Fortuna L, Graziani S, Rizzo A, Xibilia MG (2007) *Soft sensors for monitoring and control of industrial processes*. Springer, London
- Hubert M, Rousseeuw PJ, Van AS (2008) High-breakdown robust multivariate methods. *Stat Sci* 23:92–119
- Kwak N (2008) Principal component analysis based on l_1 -norm maximization. *IEEE Trans Pattern Anal Mach Intell* 30(9):1672–1680
- Liu JL (2014) Developing a soft sensor based on sparse partial least squares with variable selection. *J Process Control* 24:1046–1056
- Liu K, Chen YQ, Domański PD, Zhang X (2018) A novel method for control performance assessment with fractional order signal processing and its application to semiconductor manufacturing. *Algorithms* 11(7)
- Meng D, Zhao Q, Xu Z (2012) Improve robustness of sparse PCA by L_1 -norm maximization. *Pattern Recogn* 45:487–497
- Turkmen AS, Billor N (2013) Influence function analysis for the robust partial least squares (RoPLS) estimator. *Commun Stat-Theory Methods* 42:2818–2836
- Wang H (2000) *Bounded dynamic stochastic systems: modeling and control*. Springer, London
- Wang J, Zhong B, Zhou J (2017) Quality-relevant fault monitoring based on locality preserving partial least squares statistical models. *Ind Eng Chem Res* 56:7009–7020
- Wang Y, Karimi HR, Shen H, Fang Z, Liu M (2019) Fuzzy-model-based sliding mode control of nonlinear descriptor systems. *IEEE Trans Cybern* 49(9):3409–3419
- Zhou JL, Zhang SL, Zhang H, Wang J (2018) A quality-related statistical process monitoring method based on global plus local projection to latent structures. *Ind Eng Chem Res* 57:5323–5337
- Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. *J Comput Graph Stat* 15:265–286

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 13

Bayesian Causal Network for Discrete Variables



Ensuring the safety of industrial systems requires not only detecting the faults, but also locating them so that they can be eliminated. The previous chapters have discussed the fault detection and identification methods. Fault traceability is also an important issue in industrial system. This chapter and Chap. 14 aim at the fault inference and root tracking based on the probabilistic graphical model. This model explores the internal linkages of system variables quantitatively and qualitatively, so it avoids the bottleneck of multivariate statistical model without clear mechanism. The exacted features or principle components of multivariate statistical model are linear or nonlinear combinations of system variables and have not any physical meaning. So the multivariate statistical model is good at fault detection and identification, but not at fault root tracking.

Bayesian network (BN) can estimate and predict the potentially harmful factors of the general system, but its structure learning has some deficiencies when it is applied to the complex system, such as complex training mechanism and variable causalities. In order to simplify the network structure, lots of assumptions should be presupposed and it inevitably causes the loss of generality. Usually, a generative model (linear or nonlinear) is built to explain the data generating process, i.e., the causalities. A variety of causal discovery methods have been proposed recently to find the causalities (Hyvärinen et al. 2010; Hong et al. 2017). The most classical method is the linear non-gaussian acyclic model (LiNGAM) (Shimizu et al. 2010), in which the full structure of BN is identifiable without pre-specifying a causal order of the variables. The improved LiNGAM method is proposed to estimate the causal order of variables without any prior structure knowledge and provide better statistical performance (Shimizu et al. 2011). The nonlinear causality of a pair of variables is discovered in Johnson and Bhattacharyya (2015), where the proposed method shows a limitation when dealing with the multivariate variables.

The above approaches exploit the complexity of the marginal and conditional probability distributions in one way or the other. Despite the large number of methods for bivariate causal discovery have been proposed over the last few years, their practical performance has not been studied systematically. These methods have yet

to be applied to the actual industrial systems which usually do not meet the linear and bivariate assumptions. To address the above issues, this chapter proposes a more generalized multivariate post-nonlinear acyclic causal model for the complex industrial process. The proposed multivariate post-nonlinear acyclic causal model, named as Bayesian Causal Network (BCN), can easily find the multi-variables causality. It shows more compact structure and consistency with mechanism, compared with the traditional BN structure. In addition, it avoids the complex learning mechanism of traditional BN, so is easier to implement without compromising accuracy.

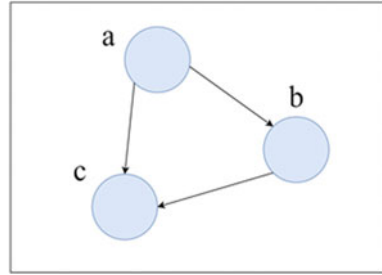
13.1 Construction of Bayesian Causal Network

It is known that there are many ways to describe the system characteristic according to the observational data and expert knowledge, such as graph model (Hipel et al. 2011), neural network model (Li et al. 2016), fuzzy model (Jiang et al. 2015). The graph model is composed of points and lines to describe the system structure and the causal relationships among variables. It provides an effective method for studying various systems, especially the complex systems. Bayesian network, a typical graph model, is the main method to deal with the knowledge representations and uncertainties based on the probability theory. It builds the causality and probability within the process components and the system variables from the prior knowledge and process data. BN consists of the structure learning and the parameter learning, in which the structure learning aims at determining the causalities within system variables and the parameter learning aims at revealing the quantitative relationship of these causalities. Bayesian network has been applied to fault diagnosis, financial analysis, automatic target recognition, military, and many other areas (Zhu et al. 2017).

13.1.1 Description of Bayesian Network

Bayesian network, also known as Belief Network or directed acyclic graphical model, is a probabilistic graphical model. It first proposed by Judea Pearl in 1985 (Pearl 1986). It is an uncertainty processing model that simulates the causal relationship in the human reasoning process, and its network topology is a directed acyclic graph (DAG). The nodes in the directed acyclic graph represent the random variables, including the observable variables, hidden variables, unknown parameters, etc. Variables or propositions that are believed to have a causal relationship (or non-conditional independence) are connected by arrows (in other words, the arrow connecting two nodes represents whether the two random variables have a causal relationship or are not conditionally independent). If two nodes are connected by a single arrow, it means that one of the nodes is “cause” and the other is “effect”, a conditional probability value is used to describe the causality degree quantitatively.

Fig. 13.1 Bayesian network example



For example, assume that node A directly affects node B , then $A \rightarrow B$. The arrow from A to B is used to establish a directed arc (A, B) from node A to node B , and the weight (its connection strength) is determined by the conditional probability $P(B|A)$. In short, a BN is formed by drawing the random variables in a directed graph according to whether they are conditionally independent. It usually uses circle to represent the random variables (nodes) and arrow to represent the conditional dependencies. Figure 13.1 gives a simple Bayesian network (Ishak et al. 2011).

13.1.2 Establishing Multivariate Causal Structure

Model-based causal discovery assumes a generative model to explain the data generating process. When the existing knowledge about the data model is unavailable, the assumed model should be sufficiently general so that it can be adapted to approximate the real data generation process. Furthermore, the model should be identified such that it could distinguish the causes from the effects. A nonlinear and multivariable system always possesses the following three characteristics (Chen et al. 2018):

1. The multivariate causalities are usually nonlinear.
2. The final target variable is affected by its cause variables and some noise who is independent from the causes.
3. Sensors or measurements may introduce nonlinear distortions into the observed value of the variables.

To discover the causality of multivariable in complex industrial systems, a more generalized multivariate post-nonlinear acyclic causal model with inner additive noise is proposed. The model is in the form of graph theory and Bayesian network structure. Assume that there is a DAG to represent the relationship among multiple observed variables. Mathematically, the generating process of X_i is

$$X_i = f_{i,2}(f_{i,1}(\mathbf{PA}_i) + e_i), \quad (13.1)$$

where the observed variables $X_i, i = \{1, 2, \dots, n\}$ are arranged in a causal order, such that no later variable causes any earlier variable. \mathbf{PA}_i is the direct cause of

X_i . $f_{i,1}$ denotes the nonlinear effect of this cause, and $f_{i,2}$ denotes the invertible post-nonlinear distortion in variable X_i . e_i is the independent disturbance which is a continuous-valued random variable with non-gaussian distributions of non-zero variances. Model (13.1) satisfies the aforementioned three characteristics: function $f_{i,1}$ accounts for the nonlinear effect of the causes PA_i ; e_i is the noise effect during the transmission from PA_i to X_i ; invertible function $f_{i,2}$ reflects the nonlinear distortion caused by the sensor or measurement.

Randomly select a pair of variables X_i and X_j , $i, j = \{1, 2, \dots, n\}$. Assume that the pair (X_i, X_j) has the causal relation $X_i \rightarrow X_j$. It's data generating process can be described in a generated model,

$$X_j = f_{j,2}(f_{j,1}(X_i) + e_j), \quad (13.2)$$

where e_j is independent from X_i . Define $s_i \triangleq f_{j,1}(X_i)$, $s_j \triangleq e_j$, and s_i is independent from s_j .

Rewrite the generating process $X_i \rightarrow X_j$ as follows:

$$\begin{aligned} X_i &= f_{j,1}^{-1}(s_i), \\ X_j &= f_{j,2}(s_i + s_j). \end{aligned} \quad (13.3)$$

X_i and X_j in (13.3) are post-nonlinear (PNL) mixtures of independent sources s_i and s_j . The PNL mixing model can be seen as a special case of the general nonlinear independent component analysis (ICA) model. Here we use nonlinear ICA method to solve this problem (13.3).

Generally there are two possibility to describe the causal relation between any two random variables X_i and X_j , ($X_i \rightarrow X_j$ or $X_j \rightarrow X_i$). We should identify the correct relation by judging which one satisfies the assumed model (13.2). If the causal relation is $X_i \rightarrow X_j$ (i.e., X_i and X_j satisfy the model (13.2)), we can invert the data generating process (13.2) to recover the disturbance e_j , which is expected to be independent from X_i . Two steps are used to examine the possible causal relationships between variables.

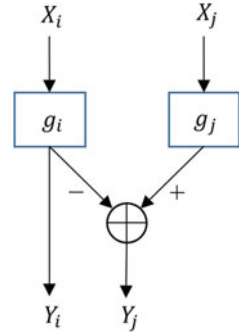
In the first step, recover the disturbance e_j corresponding to the assumed causal relation $X_i \rightarrow X_j$ based on the constrained nonlinear ICA. If this causal relation holds, there exist nonlinear functions $f_{j,2}^{-1}$ and $f_{j,1}$ such that

$$e_j = f_{j,2}^{-1}(X_j) - f_{j,1}(X_i), \quad (13.4)$$

where e_j is independent from X_i . Thus perform nonlinear ICA using the structure in Fig. 13.2 and the outputs of system are

$$\begin{aligned} Y_i &= X_i, \\ Y_j &= e_j = g_j(X_j) - g_i(X_i). \end{aligned} \quad (13.5)$$

Fig. 13.2 Constrained nonlinear ICA system used to verify if the causal relation $X_i \rightarrow X_j$ holds



The nonlinearities g_i and g_j is modeled by Multi-layer perceptrons (MLP's), and the parameters in g_i and g_j are learned by making Y_i and Y_j as independent as possible, i.e., minimizing the mutual information between Y_i and Y_j ,

$$I(Y_i, Y_j) = H(Y_i) + H(Y_j) - H(Y), \tag{13.6}$$

where $H(Y)$ is the joint entropy of $Y = (Y_i, Y_j)^T$,

$$\begin{aligned} H(Y) &= -\mathbb{E}[\log p_Y(Y)] \\ &= -\mathbb{E}[\log p_Y(X) - \log |J|] \\ &= H(X) + \mathbb{E}[\log |J|]. \end{aligned} \tag{13.7}$$

The joint density of $Y = (Y_i, Y_j)^T$ is $p_Y(Y) = p_X(X)/|J|$. J is the Jacobian matrix of the transformation from (X_i, X_j) to (Y_i, Y_j) , i.e.,

$$\begin{aligned} J &= \frac{\partial(Y_i, Y_j)}{\partial(X_i, X_j)}, \\ |J| &= \begin{vmatrix} 1 & 0 \\ g'_i & g'_j \end{vmatrix} = |g'_j|. \end{aligned} \tag{13.8}$$

Substitute (13.7) and (13.8) into (13.6), we have

$$\begin{aligned} I(Y_i, Y_j) &= H(Y_i) + H(Y_j) - \mathbb{E}[\log |J|] - H(X) \\ &= -\mathbb{E}[\log p_{Y_i}(Y_i)] - \mathbb{E}[\log p_{Y_j}(Y_j)] - \mathbb{E}[\log |g'_j|] - H(X), \end{aligned} \tag{13.9}$$

$$\tag{13.10}$$

where $H(X)$ does not depend on the parameters in g_i and g_j and can be considered as constant. The minimization problem (13.10) is solved by gradient-descent methods, and the details of the optimization are skipped.

In the second step, verify if the estimated disturbance Y_j is independent from the assume cause Y_i based on the statistical independence test. The kernel-based

statistical test is adopted with the significance level = 0.01 (Giga 2014). Denote the test statistic as $test_{i \rightarrow j}$. If $test_{i \rightarrow j} > test_{j \rightarrow i}$, it indicates that Y_i and Y_j are not independent, that is $X_i \rightarrow X_j$ does not hold. Repeat the above procedure with X_i and X_j exchanged to verify if $X_j \rightarrow X_i$ holds. If $test_{i \rightarrow j} < test_{j \rightarrow i}$, it concludes that X_i causes X_j . g_i and g_j provide an estimate of $f_{j,1}$ and $f_{j,2}^{-1}$, respectively.

For a complex system, there are n process variables. Following a test sequence, $X_1 \rightarrow X_2, X_1 \rightarrow X_3, \dots, X_{n-1} \rightarrow X_n$, the N group statistics should be tested,

$$N = n + (n - 1) + (n - 2) + \dots + 1 = \frac{n(n - 1)}{2}. \quad (13.11)$$

The total computation is in direct proportion to $2 \times N$. As the number of variables increases, the amount of computation will increase as well. The measured statistics in the positive order (or in the reverse order) are stored as

$$\begin{aligned} \mathbf{A} &= [test_{X_1 \rightarrow X_2}, test_{X_1 \rightarrow X_3}, \dots, test_{X_{n-1} \rightarrow X_n}], \\ \mathbf{B} &= [test_{X_2 \rightarrow X_1}, test_{X_3 \rightarrow X_1}, \dots, test_{X_n \rightarrow X_{n-1}}]. \end{aligned} \quad (13.12)$$

Comparing the corresponding elements of the vectors \mathbf{A} and \mathbf{B} , the causal direction of this pair of variables is determined according to the smaller statistic. Once the causality of all variables is found based on the above cyclic search, integrate them into a DAG.

13.1.3 Network Parameter Learning

The multivariate causality model gives a framework similar to the Bayesian network to find the internal structure of the complex systems. Its graphical structure expresses the causal interactions and direct/indirect relations as probabilistic networks. Its parameter represents the intensity of the complex inter-relationships among the cause-effect variables.

Consider a finite set $U = \{X_1, \dots, X_n\}$ of discrete random variables where each variable X_i may take on several discrete status from a finite set. A Bayesian network is an annotated directed acyclic graph that encodes a joint probability distribution over a set of random variables U . Formally, the Bayesian network for U is constructed as a pair $\mathbf{B} = \langle \mathbf{G}, \Theta \rangle$. \mathbf{G} is a directed acyclic graph whose vertices correspond to the random variables X_1, \dots, X_n . Θ is the parameters set that quantifies the network with $\theta_{ijk} = p(x_i^k | pa_i^j)$ and $\sum_k \theta_{ijk} = 1$, where x_i^k is the discrete status of X_i and pa_i^j is one of components in the complete parent set PA_i of X_i in \mathbf{G} . Every variable X_i is conditionally independent of its non-descendants given its parents (Markov condition). The joint probability distribution over set U is

$$P_B(X_1, \dots, X_n) = \prod_{i=1}^n P_B(X_i | \mathbf{P}A_i) = \prod_{i=1}^n \theta_{X_i | \prod \mathbf{P}A_i}. \quad (13.13)$$

The parameters of the causality Bayesian network are mainly learned from the statistics analysis of sample data. The maximum likelihood estimation method (MLE) is one of the most classical and effective algorithms in parameter learning.

Give a data set $D = \{D_1, \dots, D_N\}$ of all Bayesian network nodes. The goal of parameter learning is to find the most probable values for Θ . These values best explain the data set D , which can be quantified by the log likelihood function $\log p(D|\theta)$, denoted $L_D(\theta)$. Assume that all samples are drawn independently from the underlying distribution. According to the conditional independence assumptions, we have

$$L_D(\theta) = \log \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{n_{ijk}}, \quad (13.14)$$

where q_i is the number of combinations of the parent nodes pa_i^j , r_i is the number of the node X_i status. n_{ijk} indicates how many elements of D contain both x_i^k and pa_i^j . If the data set D is complete, MLE method can be described as a constrained optimization problem,

$$\begin{aligned} & \max L_D(\theta), \\ & \text{st. } g_{ij}(\theta) = \sum_{k=1}^{r_i} \theta_{ijk} - 1 = 0, \forall i = 1, \dots, n, \forall j = 1, \dots, q_i. \end{aligned} \quad (13.15)$$

Its global optimum solution is

$$\theta_{ijk} = \frac{n_{ijk}}{n_{ij}}, \quad (13.16)$$

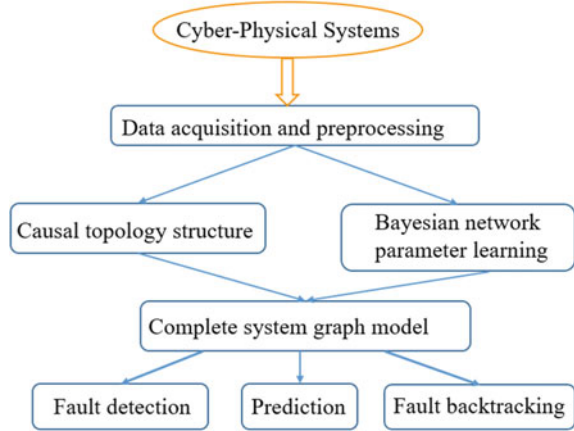
where $n_{ij} = \sum_{k=1, \dots, r_i} n_{ijk}$.

13.2 BCN-Based Fault Detection and Inference

The complete monitoring model is established via combining the multivariate causal structure and the Bayesian parameters learning. The qualitative and quantitative relationships among the process variables are revealed to the greatest extent. Then this model is forward used to accurately predict the operation status and detect faults of the critical process variables (i.e., forward inference). Similarly, it also can be inversely used to find the source of the faults (i.e., backward inference). The overall block diagram of the proposed method is shown in Fig. 13.3.

Causality network prediction or inference is to calculate the probability of the hypothesis variables at certain status according to the network topology and con-

Fig. 13.3 Overall design block diagram



ditional probability distribution of the evidence variable. An inference or query $P(Q = q | E = e_0)$ is to calculate the posterior probability of a query variable Q being at its specific value q in the condition of given evidence e_0 for node E .

There are many existing network inference algorithms, such as variable elimination algorithm and junction tree algorithm (JT). These algorithms utilize the hypothesis variables and specific independence relations induced by the evidence in BN to simplify the updating task. JT implements the inference procedure in four steps (Borsotto et al. 2006),

1. Cluster the nodes into several cliques;
2. Connect the cliques to form a junction tree;
3. Propagate information in the network;
4. Answer a query.

The inference starts from a root clique. The core step of message propagation consists of a message collection phase and a distribution phase. The cliques of the junction tree are connected by separators such that the so-called junction tree property holds. When a message is passed from one clique X to another clique Y , it is mediated by the separate set S between the two cliques. Every conditional probability distribution of the original BN is associated with a clique such that the domain of the distribution is a subset of the clique domain (we use the notation $dom(\phi)$ to refer to the domain of a potential ϕ). The set of distributions ϕ_X associated with a clique X are in standard junction tree architectures combined to form the initial clique X .

$$\phi_X = \prod_{\phi \in \phi_X} \phi. \quad (13.17)$$

For a clique, a potential or a message is a mapping from the value assignments of the nodes to the set $[0, 1.0]$. A message pass from X to Y occurs with two procedures: projection and absorption based on the Hugin architecture (architecture is proposed

by Jensen et al. 1990). The projection procedure saves the current potential and assigns a new one to S :

$$\phi_S^{old} \leftarrow \phi_S, \text{ and } \phi_S \leftarrow \sum_{X \setminus S} \phi_X. \quad (13.18)$$

The absorption procedure assigns a new potential to Y using both the old and the new tables of S ,

$$\phi_Y \leftarrow \phi_Y \frac{\phi_S}{\phi_S^{old}}, \quad (13.19)$$

where ϕ_S is the current separator potential, ϕ_S^{old} is the old separator potential, ϕ_X is the clique potential for X , ϕ_Y is the clique potential for Y .

The query answering step has two procedures. First, the marginalization procedure calculates the joint probability of Q and $E = e_0$: $P(Q, E = e_0) = \sum_{X \setminus Q} \phi_X$. Second, the normalization procedure calculates the inference result,

$$P(Q = q | E = e_0) = \frac{P(Q = q, E = e_0)}{\sum_Q P(Q, E = e_0)}. \quad (13.20)$$

The fault of operational variables is an intervention that has various effects on the production process. The main task in fault detection is to predict the system output and detect whether a fault occurs. The object of causal inference is to find the real root cause under the faulty intervention.

13.3 Case Study

In order to evaluate the performance of the proposed method, the experiment results are reported from three aspects: the causal direction identification of multi-variables, network parameter learning, and probability inference.

13.3.1 Public Data Sets Experiment

Four published data sets proposed by Mooij and Janzing (Leoand et al. 2001) are used to test the effectiveness of the nonlinear multivariate causal model. The cause-effect pairs are available at <http://webdav.tuebingen.mpg.de/cause-effect/>, which is considered as the benchmark for testing causal detection algorithms. The four data sets are (1) the ground altitude and temperature sampled at 349 stations, US; (2) census income data set which contains weighted census data extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau.

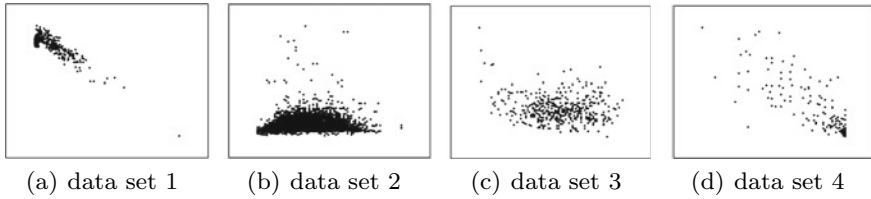


Fig. 13.4 Scatter plots of four data sets, **a–d** corresponding to data sets (1)–(4), respectively

Table 13.1 Independence test statistics under different assumption of causal directions

Causal assumption	$x \rightarrow y$	$y \rightarrow x$
#1	1.7×10^{-3}	6.5×10^{-3}
#2	1.2×10^{-4}	6.7×10^{-4}
#3	3.5×10^{-3}	8.1×10^{-3}
#4	2.2×10^{-3}	5.7×10^{-3}

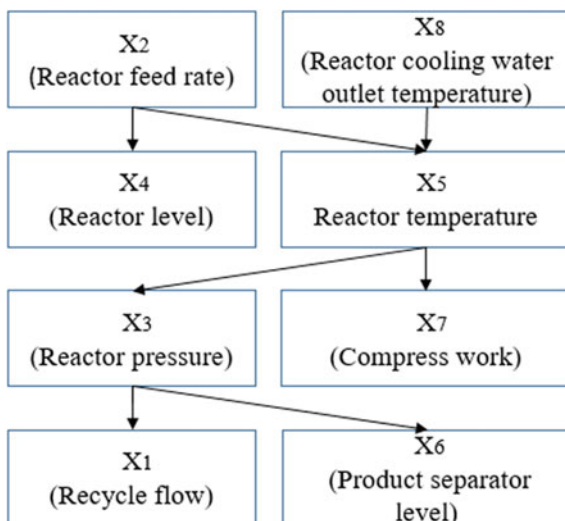
Table 13.2 Causal results of the public data sets

Data sets	#1	#2	#3	#4
Data information	x : altitude y : temperature	x : age y : wage per hour	x : age y : heart rate	x : population y : infant mortality rate
Real direction	$x \rightarrow y$	$x \rightarrow y$	$x \rightarrow y$	$x \rightarrow y$
Test results	$x \rightarrow y$	$x \rightarrow y$	$x \rightarrow y$	$x \rightarrow y$
True or false	True	True	True	True

The variables include age and wage per hour; (3) the attribute information (age and heart rate) from Cardiac Arrhythmia database; (4) the population with sustainable access to improved drinking water sources (%) total, and the infant mortality rate (per 1000 live births) both sexes, 2006. Each data set consists of two random variables which their cause-effect relationship is known. The four data sets have different attributes, which is sufficient to show the general and comprehensive nature of the data.

Figure 13.4 gives the scatter plots of the selected data sets (1)–(4). Table 13.1 shows the statistics of independence test on x and y for data sets (1)–(4) under different assumption of causal directions. The statistics are calculated separately based on these different assumptions. Comparing the test statistics under two different assumption in Table 13.1, the causal direction of each set all are determined as $x \rightarrow y$. Table 13.2 summarizes the causal results obtained by the multivariate causality model. It is found that the test results are consistent to the real causal relationship. We can conclude that the proposed method can correctly identify the causal direction regardless the diversity of data.

Fig. 13.5 The network Bnet0 from the mechanism analysis



13.3.2 TE Process Experiment

In order to illustrate the applicability of the proposed method in the actual complex industrial process, the network topology of TE process is established and used to predict the alarm variables. TE platform simulates an actual chemical process, a detailed description of the TE process is given in Chap. 4.

Experiment 1: Build Causal Structure

In this experiment, eight important process variables are selected to calculate their causality in order to facilitate the result visualization. From the mechanism analysis of TE process, it is known that when the reactor feed X_2 increases, the material is first entered into the reactor, so the reactor level X_4 must increase. So the reactor feed X_2 directly affects the reactor level X_4 . The temperature of cooling water X_8 and the reactor feed X_2 are the main factors of affecting the reactor temperature X_5 . The reactor pressure X_3 changes synchronized with the reactor temperature X_5 according to the general physical principle. In addition, once the chemical reaction in the reactor is more intense, the compressor module power X_7 will be synchronized to strengthen due to the sequential loop. At the same time, the reactor pressure X_3 also has an obvious influence on the recovered flow X_1 and the material level X_6 in the separator. Now the initial structure of the causality network is built based on the mechanism analysis (including the expert prior knowledge and the intuitive correlation analysis of process variable), named as Bnet0 shown in Fig. 13.5.

The pre-defined fault is random variations in A, B, C compositions in stream 4. The corresponding data of eight variables are collected from the simulation platform. The reaction length is 700 h to ensure that the data is sufficient to reflect the system process. 500 sampling data are obtained after the equal time decimating. The causal direction of the paired variables is shown in Table 13.3. Three different causality

Table 13.3 Causal direction of TE variables

Variables information	Statistic (positive/reverse)	Causal direction
X_2 : Reactor feed rate X_5 : Reactor temperature	$5.7 \times 10^{-6} / 8.2 \times 10^{-6}$	$X_2 \rightarrow X_5$
X_5 : Reactor temperature X_8 : Reactor cooling water outlet temperature	$7.1 \times 10^{-6} / 2.9 \times 10^{-6}$	$X_8 \rightarrow X_5$
X_2 : Reactor feed rate X_4 : Reactor level	$3.4 \times 10^{-4} / 8.5 \times 10^{-4}$	$X_2 \rightarrow X_4$
X_5 : Reactor temperature X_7 : Compress work	$7.3 \times 10^{-4} / 9.2 \times 10^{-4}$	$X_5 \rightarrow X_7$
X_3 : Reactor pressure X_5 : Reactor temperature	$7.6 \times 10^{-5} / 4.5 \times 10^{-5}$	$X_5 \rightarrow X_3$
X_3 : Reactor pressure X_6 : Product separator level	$2.9 \times 10^{-6} / 3.9 \times 10^{-6}$	$X_3 \rightarrow X_6$
X_1 : Recycle flow X_3 : Reactor pressure	$6.6 \times 10^{-6} / 2.7 \times 10^{-6}$	$X_3 \rightarrow X_1$

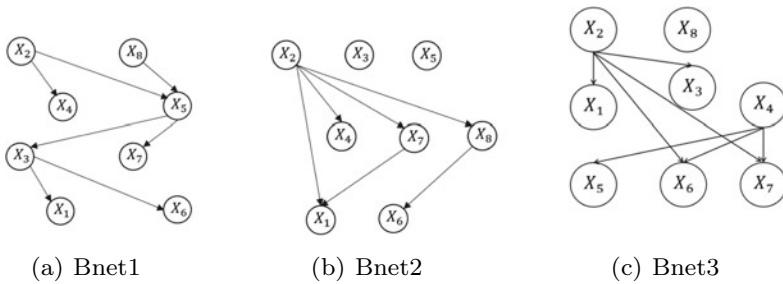


Fig. 13.6 The network compare: **a** Bnet1, **b** Bnet2, **c** Bnet3

models are compared, including (1) Bnet1, the proposed multivariate post-nonlinear acyclic causal model, shown in Fig. 13.6a; (2) Bnet2, an alternative network obtained from the traditional BN structure learning method-K2 algorithm which needs to set the node order, shown in Fig. 13.6b; (3) Bnet3, the network structure learned with the expectation maximization (EM) algorithm, shown in Fig. 13.6c.

Comparing the process analysis structure Bnet0 and Bnet1 determined by the proposed Bayesian Causal Network, it is found that Bnet1 is exactly consistent to Bnet0. The structure determined using the proposed method exactly matches the mechanism and expert knowledge, which indicates that the causal structure is credible and accurate. However, Bnet2 and Bnet3 learned from the traditional BN methods are not consistent with the mechanism. They show a big gap from the actual physical relationship. It demonstrates that the general BN learning method fails when it is applied to the complex nonlinear systems, while the proposed multivariate causality model proves its superiority.

Table 13.4 Threshold setting for alarm status in different variables

Alarm status	X_1 (km^3/h)	X_2 (km^3/h)	X_3 (kPa)	X_4 (%)	X_5 ($^{\circ}C$)	X_6 (%)	X_7 (KW)	X_8 ($^{\circ}C$)
1	<31	<46	< 2789	< 62.5	< 122.7	< 45	< 268	< 102.25
2	31–32	46–47	2789–2796	62.5–63.8	122.7–122.87	45–47.2	268–272.3	102.25–102.41
3	32–33	47–48.3	2796–2802	63.8–66	122.87–122.93	47.2–52.2	272.3–274	102.41–102.55
4	33–34	48.3–49.5	2804–2809	66–66.8	122.93–123.2	52.2–53	274–280	102.55–102.7
5	> 34	> 49.5	> 2809	> 66.8	> 123.2	> 53	> 280	> 102.7

Experiment 2: Parameter Learning Once the TE network structure is determined, the alarm prediction model can be obtained by parameter learning of this causality structure network. In general, the process alarm event can be divided into five-alarm levels, namely, high-high alarm (HH), high alarm(H), normal(N), low alarm(L) and low-low alarm(LL), corresponding to the number 1,2,3,4,5. The first step is to discretize the continuous variables into five-alarm levels by setting different thresholds, shown in Table 13.4.

Here the MLE algorithm is adopted to learn the network parameters and get a complete probability table. Suppose that the initial probability of the alarm level in the normal condition is theoretically divided equally. Then the conditional probability values for all variables are calculated based on the BN parameter learning. Considering two root nodes X_2 and X_8 , their corresponding probabilities for five status are 0.0843, 0.2211, 0.4704, 0.2026 and 0.0217, respectively. The probability of other descendant variables as shown in Fig. 13.7. Hot plot is used to show the probability since the precise value has nothing meaning for the alarm prediction and inference. The color represents the probability range between 0 and 1.

It should be concerned with the probability value of close to 1. These are the key points in determining the inference results. When the probability is less than 0.5, the result situation will not likely appear in the actual inference. Figure 13.7a shows the probability of X_5 under the combined action of X_2 and X_8 . The abscissa is the status condition of X_8 and X_2 , and the ordinate is the probability value for five-alarm status of X_5 displayed in corresponding color. $P(X_5 = 1|X_8 = 1, 2 \text{ and } X_2 = 1) \approx 1$ in the lower left corner of Fig. 13.7a. It means that X_5 occurs the low-low alarm with the probability close to 1 when X_2 and X_8 are in the low-low alarm status. $P(X_5 = 5|X_8 = 4, 5 \text{ and } X_2 = 5) \approx 1$ in the upper right corner of Fig. 13.7a. It means that X_5 occurs the high-high alarm with the probability close to 1 when X_2 and X_8 are in the high-high alarm status. These inference results are consistent with the actual mechanism.

Figure 13.7b–e reflects the probability relationship between bivariate variables. Figure 13.7b shows the probability of X_4 under the action of X_3 . $P(X_4 = 5|X_3 =$

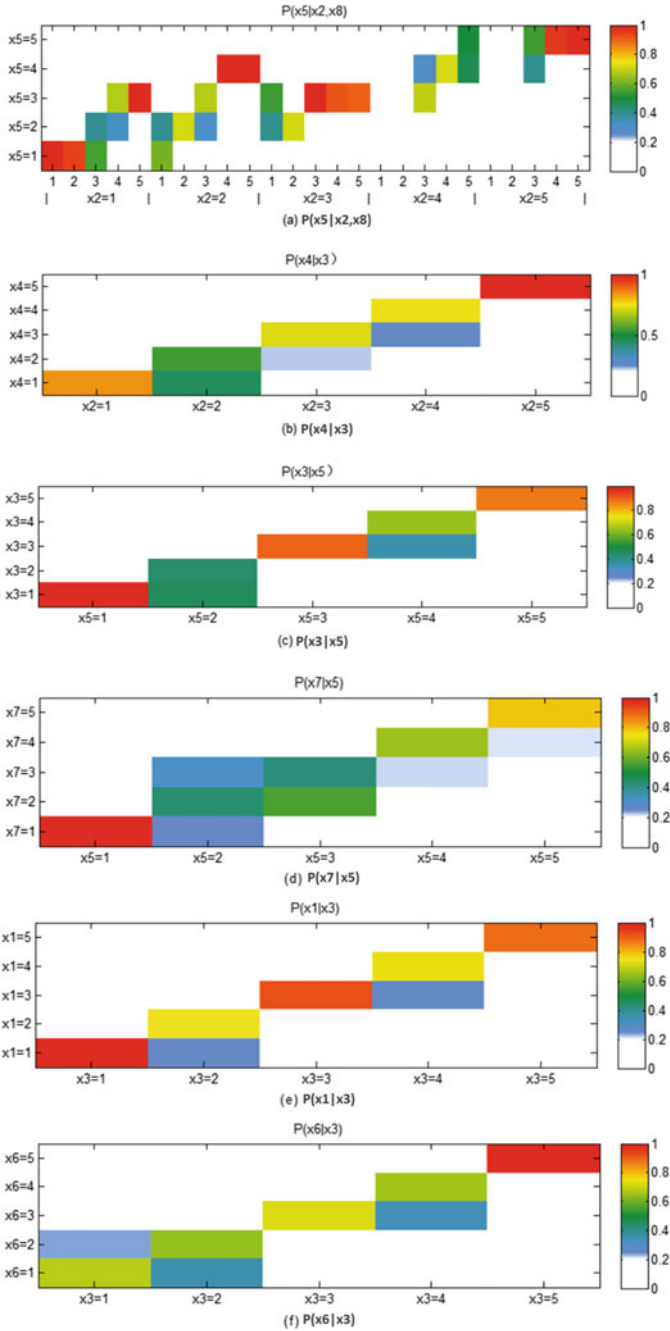


Fig. 13.7 Conditional probability of the descendant variables: **a** $P(X_5|X_8, X_2)$, **b** $P(X_4|X_3)$, **c** $P(X_3|X_5)$, **d** $P(X_7|X_5)$, **e** $P(X_1|X_3)$, **f** $P(X_6|X_3)$

Table 13.5 Alarm level prediction of compress work X_7

No.	X_2	X_8	X_5	X_7	\hat{X}_7	Max Prob.
1	1	2	1	2	1	0.4571
2	2	1	2	1	1	0.6501
3	1	2	2	2	2	0.7627
4	2	1	2	2	2	0.6729
5	1	2	2	1	1	0.6896
6	3	3	2	3	1	0.8760
7	3	3	2	3	3	0.6344
8	3	3	3	2	2	0.8563
9	3	3	2	3	2	0.3454
10	2	3	3	3	3	0.5073
11	3	3	3	2	3	0.4432
12	3	2	3	3	3	0.5696
13	4	3	4	4	3	0.3128
14	3	4	4	4	4	0.6284
15	4	5	5	5	5	0.7557
16	4	3	4	4	5	0.3783
17	5	5	4	4	4	0.7947
18	4	5	4	4	4	0.8325
19	5	4	5	4	5	0.6454
20	5	4	4	5	5	0.8113

5) ≈ 1 in the upper right corner. It means that the probability of X_4 occurs the high-high alarm with the probability close to 1 when X_3 in the high-high alarm status. However, $P(X_4 = 1|X_3 = 5) = 0$ in the lower right corner. It means that X_4 occurs the high-high alarm with the probability close to 0 when X_3 in the low-low alarm status. $P(X_4 = 1 \text{ and } X_4 = 2|X_3 = 2) \approx 0.5$ in the green area. It means the probability of X_4 occurs the low alarm or low-low alarm almost same when X_3 in the low alarm status. Similarly, the inference results obtained from Fig. 13.7c–e are consistent with the mechanism.

Experiment 2: Alarm Prediction Alarm prediction is a top-down inference according to the evidences inference conclusion. The probabilistic analysis calculates the likelihood of each status for the result variable may occur. The discrete status corresponding to the maximum probability is the alarm prediction result.

Using the established multivariate causality network model, compress work X_7 is predicted when its parent variables X_2 , X_8 and X_5 are known. The prediction results for model Bnet1 are shown in Table 13.5, where \hat{X}_7 is the prediction value of X_7 .

The total prediction accuracy for the 20 simulation experiments is 75%. When the maximum probability of the predicted value is greater than 0.5, the prediction result is confident. Furthermore, the predictions with a high probability is consistent

with the true status. When the maximum probability of the predicted value is less than 0.5, the prediction result is not believable and accurate. The mis-predictions confuse between the adjacent status, such as the normal status 2 and Low alarm 3 (or high alarm 2). The simulation results show that the multivariate causality network can find the intrinsic relationships among various process variables, and give precise fault or alarm prediction.

13.4 Conclusions

This chapter proposes a multivariate causality model to analyze the causal direction of multivariable and final determine the network topology. The proposed method can describe the system structure more accurate than the traditional BN structure learning method especially when the industrial process is high complex. Combined with the network parameters learning and evidence inference technique, an accurate monitoring and alarm prediction can be performed. The validity of the proposed method is verified via the public data set and TE process. An compact network structure and confident alarm prediction are obtained for the TE process based on the causal analysis and probability inference. Both the methodology and the simulation results show that the proposed multivariate causality model has great value for the process industry modeling and monitoring.

There are some issues worth further discussion. The computing efficiency of the proposed multivariate post-nonlinear acyclic causal modeling method should be considered when solving the large-scale causal analysis problems in the real world. Developing the efficient algorithm to find the causal relationship of multiple variables based on the general functional causal models is still an important topic. To improve the computational efficiency, a feasible solution is to limit the complexity of the causal structure, such as decreasing the number of direct causes of each variable. Moreover, a smart optimization procedure instead of the exhaustive search should be considered further.

References

- Borsotto M, Zhang W, Kapanci E, Pfeffer A, Crick C (2006) A junction tree propagation algorithm for bayesian networks with second-order uncertainties. In: 18th IEEE international conference on tools with artificial intelligence, 2006. ICTAI '06
- Chen X, Wang J, Zhou J (2018) Fault detection and backtrace based on graphical probability model. In: 2018 prognostics and system health management conference (PHM-Chongqing)
- Giga M (2014) Statistical tests, test of independence. *Nihon Ika Daigaku Igakkai Zasshi* 10(2):115–119
- Hipel KW, Kilgour DM, Fang L (2011) The graph model for conflict resolution. *Wiley encyclopedia of operations research and management science*

- Hong Y, Hao Z, Mai G, Chen B, Rui X (2017) Inferring causal direction from multi - dimensional causal networks for assessing harmful factors in security analysis. *IEEE Access* 5:20009–20019
- Hyvärinen A, Zhang K, Shimizu S, Hoyer PO (2010) Estimation of a structural vector autoregression model using non-gaussianity. *J Mach Learn Res* 11(2010):1709–1731
- Ishak MB, Leray P, Amor NB (2011) A two-way approach for probabilistic graphical models structure learning and ontology enrichment. In: International conference on knowledge engineering ontology development
- Jensen FV, Lauritzen SL, Olesen KG (1990) Bayesian updating in causal probabilistic networks by local computations. *Comput Stat Quarterly* 4:269–282
- Jiang Y, Deng Z, Chung FL, Wang S (2015) Multi-task TSK fuzzy system modeling using inter-task correlation information. *Inf Sci* 298:512–533
- Johnson RA, Bhattacharyya GK (2016) *Statistics and causality: methods for applied empirical research*. Wiley, Hoboken
- Leoand M, Russell E, Braatz R (2001) *Tennessee eastman process*. Springer, London
- Li X, Zhao L, Wei L, Yang MH, Fei W, Zhuang Y, Ling H, Wang J (2016) Deepsaliency: multi-task deep neural network model for salient object detection. *IEEE Trans Image Process Publ IEEE Signal Process Soc* 25(8):3919
- Pearl J (1986) Fusion, propagation, and structuring in belief networks. *Artif Intell* 29(3):241–288
- Shimizu S, Hoyer PO, Kerminen A (2006) A linear non-gaussian acyclic model for causal discovery. *J Mach Learn Res* 7(4):2003–2030
- Shimizu S, Inazumi T, Sogawa Y, Hyvarinen A, Kawahara Y, Washio T, Hoyer PO, Bollen K (2011) DirectLiNGAM: a direct method for learning a linear non-gaussian structural equation model. *J Mach Learn Res* 12(2):1225–1248
- Zhu J, Ge Z, Song Z, Zhou L, Chen G (2017) Large-scale plant-wide process modeling and hierarchical monitoring: a distributed bayesian network approach. *J Process Control*, pp 91–106

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Chapter 14

Probabilistic Graphical Model for Continuous Variables



Most of the sampled data in complex industrial processes are sequential in time. Therefore, the traditional BN learning mechanisms have limitations on the value of probability and cannot be applied to the time series. The model established in Chap. 13 is a graphical model similar to a Bayesian network, but its parameter learning method can only handle the discrete variables. This chapter aims at the probabilistic graphical model directly for the continuous process variables, which avoids the assumption of discrete or Gaussian distributions.

This chapter expands the previous work in Chap. 13 from the random discrete variables to the random continuous variables. In addition to enhancing the effect of causal structure and parameter learning on the continuous variables, kernel density estimation is used to construct the node association strength of the causal graph network in the form of probability density. The conditional probability density is obtained from the mathematical operation between the low-dimensional probability density and the high-dimensional joint probability density. This non-parametric estimation method directly estimates the probability density of continuous variables and avoids the limitations of traditional Gaussian assumptions. Moreover, this chapter strictly derives the evaluation indicators for the KDE estimation quality. The proposed causal learning mechanism does not have any restrictions, such as linear, nonlinear, or distribution functions. It establishes an accurate causal probability graphical model to detect faults and find the root cause of the fault.

14.1 Construction of Probabilistic Graphical Model

14.1.1 Multivariate Casual Structure Learning

The first step of building a graphical model is to construct a causal topological relationship. The causal hypothesis model is a post-nonlinear model. It can determine the

causal relationship between multiple variables through hypothesis testing. Detailed information can be found in Chap. 13 (Chen et al. 2018).

Consider a model which represents the causal relationship between variables. Here a generative model is used to explain the data generation process. When the existing mechanism of the data model cannot be determined, the hypothetical model should be sufficiently versatile so that it can be adapted to approximate the actual data generation process. In addition, the model should be identified so that cause and effect can be distinguished.

In order to discover the causality of multiple variables in a complex system, a more generalized multivariable nonlinear acyclic causal model with internal additive noise is given same as Chap. 13. The model adopts the form of graph theory and Bayesian network structure. Assume that a directed acyclic graph (DAG) represents the relationship between multiple observed variables. Select a pair of variables X_i and $X_j, i, j = \{1, 2, \dots, n\}$ from the system, respectively. If X_i is X_j 's parent node and its data generating process is described in a post-nonlinear(PNL) mixing model. The generation process of X_i is $X_j = f_{j,2}(f_{j,1}(X_i) + e_j)$, where $f_{i,1}$ denotes the nonlinear effect of the causes, and $f_{i,2}$ denotes the invertible post-nonlinear distortion in variable X_i . e_j is the independent disturbance. Here it is applicable to a combination of hypothesis testing and nonlinear independent component analysis (ICA) to solve this problem (Shimizu et al. 2011). To describe in simplified language, it can be divided into two steps:

1. The nonlinear ICA method with constraints is used to calculate the interference e_j corresponding to the assumed causality $X_i \rightarrow X_j$;
2. The statistical independence test is used to determine the independent relationship between the estimated interference e_j and the assumed cause X_i .

For any pair of variables in the system, two causal assumptions can be made. The causality is assumed positive and negative, and the direction of the causality is determined by comparing the statistical information obtained by calculation. After $n(n-1)$ hypotheses and tests, the causality of all system variables is determined finally. Therefore, this multivariate nonlinear acyclic causal modeling method will not have the limitation of Bayesian network structure learning. It can effectively establish the causal structure of the process.

14.1.2 Probability Density Estimation

Section 14.1.1 completed the construction of the causal structure of the model. The complete graph model also should include the quantitative relationships between nodes which is described as probabilistic connection of nodes here. The probability density of the node variable is determined by the non-parametric probability density estimation method. Because the child node is affected by its parent node, the probabilistic connection relationship manifests itself in the conditional probability

density. Kernel Density Estimation (KDE) is a prominent method to estimate the non-parametric probability density. The explicit form of the density function is the main advantage of KDE method (Chen et al. 2018).

Let $X_1, X_2, X_3, \dots, X_n$ be a set of samples of the random variable X . Its density function $f(x), x \in R, X$ is unknown. The distribution density function $f(x)$ can be derived from its corresponding cumulative distribution function $F(x)$,

$$f(x) = \frac{dF(x)}{dx} \approx \frac{F(x+h) - F(x-h)}{2h}, \quad (14.1)$$

where $h > 0$ is the window width. The empirical distribution function $F_n(x) = \frac{1}{n} \sum_i I(X_i \leq x)$ is used to estimate $F(x)$. Substitute it into (14.1),

$$\begin{aligned} \hat{f}(x) &= \frac{dF(x)}{dx} \approx \frac{F(x+h) - F(x-h)}{2h} \\ &= \frac{1}{2nh} \sum_i I(x-h < X_i \leq x+h) \\ &= \frac{1}{nh} \sum_i K_0 \left(\frac{X_i - x}{h} \right). \end{aligned} \quad (14.2)$$

(14.2) gives the KDE for $f(x)$ with a window width h and a kernel function $K_0 = \frac{1}{2}I(|u| \leq 1)$.

The more general kernel density estimate is

$$\hat{f}(x) = \frac{1}{nh} \sum_i^n K \left(\frac{X_i - x}{h} \right), \quad (14.3)$$

where $\hat{f}(x)$ gives the estimate of the probability density function. n, h, K are the number of samples, window width and kernel function.

Conditional probability density calculation requires additional mathematical operations. Similarly, consider two random sample sets $X_1, X_2, X_3, \dots, X_n$ and $Y_1, Y_2, Y_3, \dots, Y_n$, where X is cause variable and Y is effect variable. The joint probability density of x and y is defined as

$$\hat{f}(x, y) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 h_2} K \left(\frac{x - X_i}{h_1}, \frac{y - Y_i}{h_2} \right), \quad (14.4)$$

where h_1 and h_2 are the window width corresponding to the cause variable x and the effect variable y , respectively.

According to the definition of conditional probability, the conditional density $f(y|x)$ is obtained as follows:

Table 14.1 Common kernel functions

Number	Kernel function	Expression
1	Uniform	$\frac{1}{2}I(u \leq 1)$
2	Triangle	$(1 - u)I(u \leq 1)$
3	Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}\mu^2)$
4	Epanechnikov	$\frac{3}{4}(1 - \mu^2)I(u \leq 1)$

$$f(y|x) = \frac{f(x, y)}{f(x)}. \quad (14.5)$$

The kernel function here affects the precision of kernel density estimation. How to select an appropriate kernel function is an important issue. Usually, the following properties should be considered: symmetry, non-negative, and normality (Zeng et al. 2017). The mathematical description of common kernel functions is given in Table 14.1 (Jiang and Nicholas 2014).

It can be seen from the KDE expression that the kernel function K , sample size n and its window width h are the main contributing factors of $f(x)$. Once the number of samples n is fixed, K and h directly affect the accuracy of the system model parameters. Furthermore, the effectiveness of fault detection and root cause diagnosis will fluctuate directly. Therefore, in order to estimate the probability density more accurately and improve the estimation quality of KDE, a KDE evaluation criterion is given in the next section. There are already data showing that the choice of kernel function has a negligible effect on the result of kernel density estimation (Silverman 1998), so the optimization of K is not considered here.

14.1.3 Evaluation Index of Estimation Quality

According to the definition of kernel density, consider the following two cases: (1) the value of the window width h is very large. The average compression transformation $\frac{x-X_i}{h}$ can remove the local details of the probability density function, which results in the smoothness of probability density estimation curve. A relatively low resolution is shown at this case, and the estimation deviation is enlarged; (2) the value of the window width is very small. On the contrary, the influence of the randomness of probability density will increase, and the important characteristics of density will be masked. It causes the larger fluctuation of density estimation and the stability is easy to be deteriorated. The estimation variance is too large at this case (Jiang and Nicholas 2014).

The requirements about the accurate estimation include much closer to the true values and remaining stable for different observations. These two attributes are described by the estimated deviation and variance which are given as

$$\begin{aligned} \text{Bias}\{\hat{f}(x)\} &= \mathbb{E}[\hat{f}(x)] - f(x) \\ \text{Var}\{\hat{f}(x)\} &= \mathbb{E}[\hat{f}(x)^2] - [\mathbb{E}\hat{f}(x)]^2. \end{aligned} \quad (14.6)$$

The probability density function of the child nodes in the causal model is affected by the parent nodes. Its probability density usually is multidimensional. Consider a two-dimensional kernel density function $f(x, y)$ as an example. Its deviation and variance are

$$\begin{aligned} \text{Bias}\{\hat{f}(x, y)\} &= \mathbb{E}[\hat{f}(x, y)] - f(x, y) \\ \text{Var}\{\hat{f}(x, y)\} &= \mathbb{E}[\hat{f}(x, y)^2] - [\mathbb{E}\hat{f}(x, y)]^2. \end{aligned} \quad (14.7)$$

Here the mean square integral error (MISE) is introduced as the evaluation index of KDE. The MISE index has a unique advantage to evaluate the difference between the estimated function and the true function. At the same time, it also guarantees the fitness and smoothness of kernel estimation.

One-dimensional MISE is defined as

$$\text{MISE}[\hat{f}(x)] = \mathbb{E} \int [\hat{f}(x) - f(x)]^2 dx. \quad (14.8)$$

Two-dimensional MISE is defined as

$$\text{MISE}[\hat{f}(x, y)] = \mathbb{E} \iint [\hat{f}(x, y) - f(x, y)]^2 dx dy. \quad (14.9)$$

The above MISE indices are simplified as, and the details can be found from the supporting information in Chen et al. (2018),

$$\begin{aligned} \text{MISE}[\hat{f}(x)] &= \int \text{Var}(\hat{f}(x)) + \int \text{Bias}^2(\hat{f}(x)) dx \\ &= \frac{1}{nh} \int K^2(t) dt + \frac{1}{4} h^4 \left[\int t^2 K(t) dt \right]^2 \int [f''(x)]^2 dx \end{aligned} \quad (14.10)$$

$$\begin{aligned} \text{MISE}[\hat{f}(x, y)] &= \frac{1}{nh_1 h_2} \int K^2(t) dt + \frac{1}{4} h_1^4 h_2^4 \\ &\quad \times \left[\int t^2 K^2(t) dt \right]^2 \iint (\nabla f(x, y))^2 dx dy. \end{aligned} \quad (14.11)$$

It is found from (14.10) and (14.11) that the values of the functions $\int K^2(t) dt$ and $\int t^2 K(t) dt$ are related to the kernel function K . They are not difficult to calculate if the mathematical expression of kernel function is substituted into the above equations. Generally speaking, window width h has a greater impact on MISE value, so optimizing h is critical. Here (14.10) and (14.11) are also used as optimization objectives to find the best window width h .

For one-dimensional probability density, let $d \left(\text{MISE} \left[\hat{f}(x) \right] \right) / dh = 0$. Then

$$h_{opt} = \sqrt[5]{\frac{\int K^2(t)dt}{n[\int t^2 K(t)dt]^2 \int f''(x)^2 dx}}. \quad (14.12)$$

For two-dimensional probability density, let

$$\begin{aligned} \frac{\partial \text{MISE}[\hat{f}(x, y)]}{\partial h_1} &= h_1^3 h_2^4 \left(\int t^2 K(t)dt \right)^2 \iint (\nabla f(x, y))^2 dx dy \\ &\quad - \frac{1}{n h_1^2 h_2} \int K^2(t)dt \\ &= 0, \\ \frac{\partial \text{MISE}[\hat{f}(x, y)]}{\partial h_2} &= h_2^3 h_1^4 \left(\int t^2 K(t)dt \right)^2 \iint (\nabla f(x, y))^2 dx dy \\ &\quad - \frac{1}{n h_2^2 h_1} \int K^2(t)dt \\ &= 0. \end{aligned} \quad (14.13)$$

Then

$$\begin{aligned} h_1^{opt} &= \sqrt[5]{\frac{\int K^2(t)dt}{n h_2^5 (\int t^2 K(t)dt)^2 \iint (\nabla f(x, y))^2 dx dy}} \\ h_2^{opt} &= \sqrt[5]{\frac{\int K^2(t)dt}{n h_1^5 (\int t^2 K(t)dt)^2 \iint (\nabla f(x, y))^2 dx dy}}. \end{aligned} \quad (14.14)$$

If the kernel function is predetermined, $\frac{\int K^2(t)dt}{(\int t^2 K(t)dt)^2} = C(k)$ is a constant. Usually the true probability density functions $f(x)$ and $f(x, y)$ are unknown. The estimated probability density function (14.3) and (14.4) are substituted into (14.12) and (14.14), respectively. Then the optimal parameter h for one-dimensional estimation or h_1 and h_2 for two-dimensional estimation are obtained.

14.2 Dynamic Threshold for the Fault Detection

Generally speaking, the process variables show obvious difference in their measurements in the normal operation and faulty operation. Then the measurement difference must be reflected in the probability density distribution. System failure detection is to find their differences based on the appropriate thresholds. Here, it is not feasible to use the confidence interval of the normal state to directly distinguish the fault. The actual process data are usually accompanied by a lot of noise, the distribution is not

ideal even in the normal operation. Therefore, its confidence cannot be completely described as a constant horizontal line. The constant confidence line is further difficult to distinguish the normal operation and the fault operation. Therefore, the idea of dynamic threshold is introduced. Fused Lasso (FL) method is common to denoise in the field of signal processing. Here it is used to design the dynamic confidence limits. It can provide the required reasonable range for each node based on the normal data.

The Fused Lasso Signal Approximator (FLSA) aims at eliminating noise and smoothing data (Bensi et al. 2013). The real-valued observations $\mathbf{y} = \beta\mathbf{x}$ is obtained by finding the sequence β_1, \dots, β_N that minimizes the criterion,

$$J_{FL} = \frac{1}{2} \sum_{k=1}^N (\mathbf{y}_k - \beta_k \mathbf{x}_k)^2 + \lambda_1 \sum_{k=1}^N |\beta_k| + \lambda_2 \sum_{k=2}^N |\beta_k - \beta_{k-1}|, \quad (14.15)$$

where λ_1 and λ_2 are tuning parameters, $\mathbf{x}_1, \dots, \mathbf{x}_N$ is the feature variables. The objective of J_{FL} consists of three parts: $\frac{1}{2} \sum_{k=1}^N (\mathbf{y}_k - \beta_k \mathbf{x}_k)^2$ is the traditional index of the least squares algorithm. It strives for the regression accuracy of the model for all the existed measurements $[x_k, y_k]$. The last two parts $\lambda_1 \sum_{k=1}^N |\beta_k| + \lambda_2 \sum_{k=2}^N |\beta_k - \beta_{k-1}|$ encourages the sparsity of regression coefficients and their differences. The parameters λ_1 and λ_2 are adjusted to trade-off the regression accuracy and denoising power. (14.15) is totally a denoising problem if $\lambda_1 = 0$.

Here the hidden Markov model (HMM) and the maximum likelihood estimation method are used for optimization calculation. The HMM posits an emission probability $Pr(\mathbf{y}_k | \beta_k)$ that is a standard normal distribution, and a transition probability $Pr(\beta_{k+1} | \beta_k)$ that is double exponential with parameter λ_2 (where Pr denotes probability).

The Viterbi algorithm is a typical dynamic programming algorithm for this HMM problem, which the detailed description be found in (Rabiner et al. 1989). The objective function (14.15) is rewritten as maximization in a more general form,

$$J_{FL} = \sum_{k=1}^N e_k(\beta_k) - \lambda_2 \sum_{k=2}^N d(\beta_k, \beta_{k-1}), \quad (14.16)$$

where $e_k(\mathbf{b}) = \sum_{i=1}^R \mathbf{y}_{ik} v_i(\mathbf{b})$.

Denote the variable sequences $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$ as the shorthand $\mathbf{x}_{1:k}$. Rewrite the criterion (14.16) as follows:

$$\begin{aligned} J_{FL} &= \max_{\beta_{1:N}} \left[\sum_{k=1}^N e_k(\beta_k) - \lambda_2 \sum_{k=2}^N d(\beta_k, \beta_{k-1}) \right] \\ &= \max_{\beta_N} [e_N(\beta_N)] + \max_{\beta_{1:(N-1)}} \left[\sum_{k=1}^{N-1} e_k(\beta_k) - \lambda_2 \sum_{k=2}^N d(\beta_k, \beta_{k-1}) \right] \end{aligned} \quad (14.17)$$

and

$$\begin{aligned}
f_N(\boldsymbol{\beta}_N) &:= \max_{\boldsymbol{\beta}_{1:(N-1)}} \left[\sum_{k=1}^{N-1} e_k(\boldsymbol{\beta}_k) - \lambda_2 \sum_{k=2}^N d(\boldsymbol{\beta}_k, \boldsymbol{\beta}_{k-1}) \right] \\
&= \max_{\boldsymbol{\beta}_{N-1}} [e_{N-1}(\boldsymbol{\beta}_{N-1}) + \lambda_2 d(\boldsymbol{\beta}_N, \boldsymbol{\beta}_{N-1})] \\
&\quad + \max_{\boldsymbol{\beta}_{1:(N-2)}} \left[\sum_{k=1}^{N-2} e_k(\boldsymbol{\beta}_k) - \lambda_2 \sum_{k=2}^{N-1} d(\boldsymbol{\beta}_k, \boldsymbol{\beta}_{k-1}) \right].
\end{aligned} \tag{14.18}$$

The definitions of functions $f_{N-1}(\boldsymbol{\beta}_{N-1}), f_{N-2}(\boldsymbol{\beta}_{N-2}), \dots, f_2(\boldsymbol{\beta}_2)$ are similar to $f_N(\boldsymbol{\beta}_N)$. The maximization problem is solved further iteratively. It is summarized by introducing the intermediate functions with k ranging from 2 to N ,

$$\begin{aligned}
\delta_1(\mathbf{b}) &:= e_1(\mathbf{b}) \\
\psi_k(\mathbf{b}) &:= \arg \max_{\tilde{\mathbf{b}}} [\delta_{k-1}(\tilde{\mathbf{b}}) - \lambda_2 |b - \tilde{\mathbf{b}}|] \\
f_k(\mathbf{b}) &:= \delta_{k-1}(\psi_k(\mathbf{b})) - \lambda_2 |b - \psi_k(\mathbf{b})| \\
\delta_k &:= e_k(\mathbf{b}) + f_k(\mathbf{b}).
\end{aligned} \tag{14.19}$$

The functions $\psi_k(\cdot)$ take part in the backward pass of the algorithm. This backward pass computes $\hat{\boldsymbol{\beta}}_1, \dots, \hat{\boldsymbol{\beta}}_N$ through a recursion identical to that of the Viterbi algorithm for HMMs:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_N &= \arg \max_b \{\delta_N(\mathbf{b})\} \\
\hat{\boldsymbol{\beta}}_k &= \psi_{k+1}(\hat{\boldsymbol{\beta}}_{k+1}) \quad \text{for } k = N-1, N-2, \dots, 1.
\end{aligned} \tag{14.20}$$

So far, the above FL theory is implemented to obtain the dynamic threshold of the data model. During the process of fault detection, the KDE estimated probability values are the input variable of the FLSA algorithm for smoothing. The influence of data noise on the estimated probability density function is eliminated and a credible threshold is found to distinguish the normal operation and the faulty operation.

14.3 Forward Fault Diagnosis and Reverse Reasoning

Detailed theoretical supports have been supplemented enough in last section, including the construction of probability graph models, the selection of probability density estimation evaluation indicators and parameter optimization, and the setting of dynamic thresholds for fault detection. The established model structure is determined by the causal direction between operating units, which represents the qualitative relationship between nodes. The non-parametric KED estimation is used to obtain the parameters of the graph model, i.e., the causal probability relationship. Probability can quantitatively describe the dependence between process variables. The evalu-

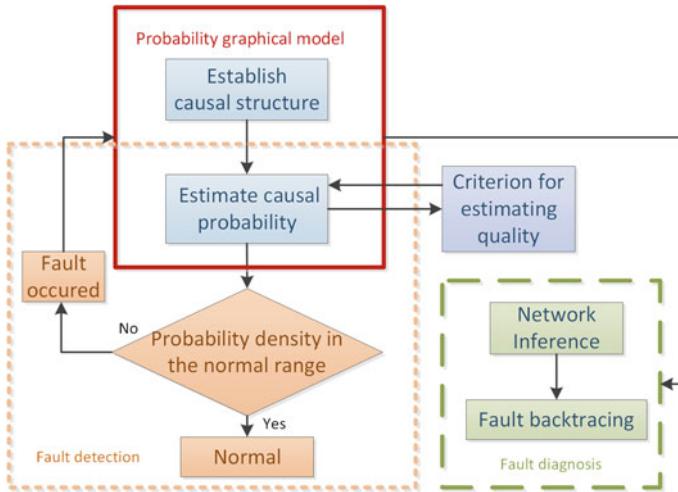


Fig. 14.1 The overall framework

ation index of the probability relationship estimation is derived and calculated to ensure the accuracy of the graphical model.

Now this section combines and implements the above theoretical methods into a certain fault detection and diagnosis framework, which can be used to diagnose abnormal events in the system and locate the root cause of the fault. The overall framework of the proposed method is represented in Fig. 14.1.

The main steps for fault detection and root tracing are summarized based on the detail flow chart in Fig. 14.2,

1. Construct a cause-effect network structure for the selected process variables from the industrial process;
2. List all the probability density functions that need to be estimated, including one-dimensional densities for root nodes, multidimensional joint densities, or the corresponding conditional probability densities for child nodes;
3. Estimate the (conditional) probability densities of each node based on KDE method;
4. Calculate the dynamic threshold for the health status of each node by input all the density values to FLSA;
5. Collect test data and detect whether faults occur compared with the dynamic threshold;
6. Reverse reasoning based on the graph model in the case of failure. Starting from the faulty node, check which parent nodes of the faulty node is faulty in turn. Remove all non-faulty parent nodes and clarify the fault propagation path until the fault root is found.

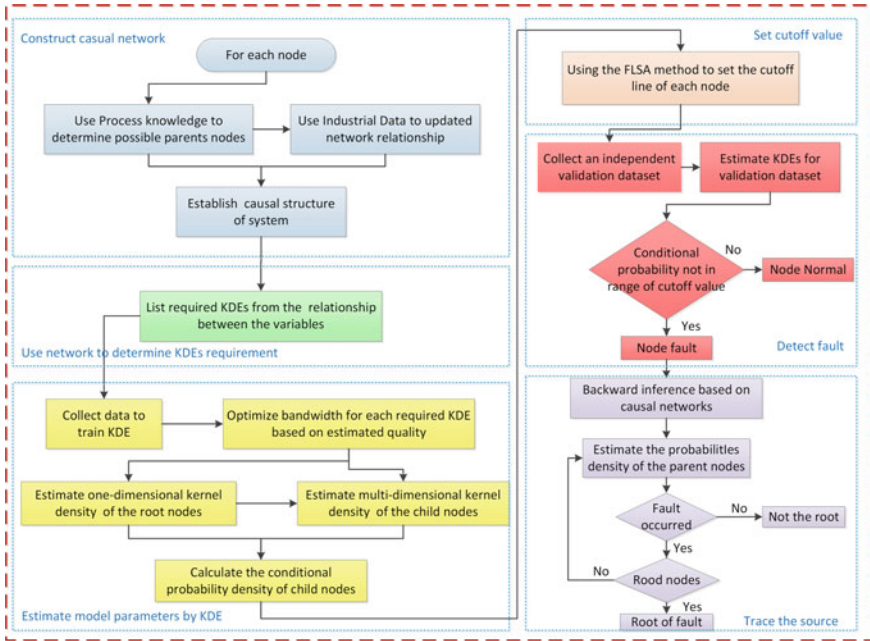
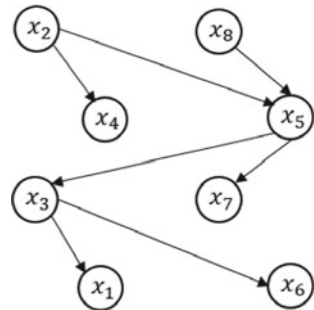


Fig. 14.2 Flowchart for detecting and tracing faults

Fig. 14.3 The casual structure of partial TE process



14.4 Case Study: Application to TEP

The proposed methods are verified on Tennessee Eastman (TE) process simulator. The TE process contains a total of 52 process variables and measurement variables. Eight variables in the reactor module are selected to test the causal structure, same as Chap. 13. The physical meanings of these variables are listed in Table 14.2. According to the causal analysis method, it is not difficult to obtain the causal relationship between eight variables (the detail analysis also can be found in Chap. 13). The corresponding topology is shown in Fig. 14.3.

Table 14.2 Process manipulated variables

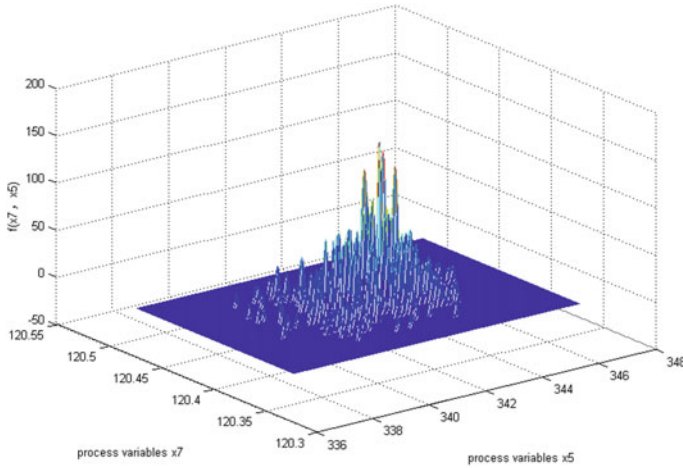
Variable (Symbol in the Fig. 3)	Physical meaning	Units
$x_1(v_5)$	Recycle flow	km^3/h
$x_2(v_6)$	Reactor feed rate	km^3/h
$x_3(v_7)$	Reactor pressure	kPa
$x_4(v_8)$	Reactor level	%
$x_5(v_9)$	Reactor temperature	$^{\circ}\text{C}$
$x_6(v_{12})$	Product separator level	%
$x_7(v_{20})$	Compress work	KW
$x_8(v_{21})$	Reactor cooling water outlet temperature	$^{\circ}\text{C}$

List all the probability density function and conditional probability density of nodes in the causal graph. In total, $f(x_2)$, $f(x_8)$, $f(x_4|x_2)$, $f(x_5|x_8)$, $f(x_7|x_5)$, $f(x_3|x_5)$, $f(x_1|x_3)$, $f(x_6|x_3)$ need to be estimated. Here the root nodes x_2 and x_8 have one-dimensional probability density function. Optimize the window width h to obtain an accurate probability estimate.

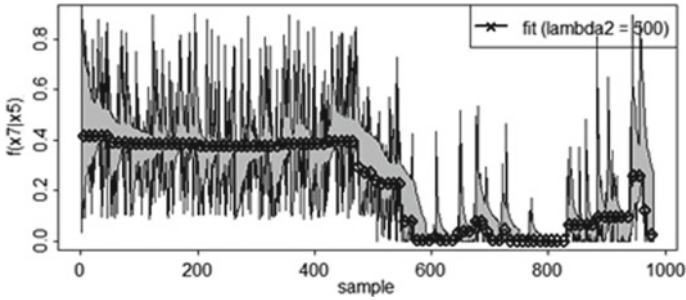
The training data set contains 960 samples in the normal operation. These data are used to obtain the KDE of the model. Combine the causal structure constructed in the previous step to get a complete graphical model. Fault IDV(4) is a minor fault which is used as a test sample to verify the effectiveness and sensitiveness of the proposed method to minor faults. The fault IDV(4), a step change of the reactor cooling water inlet temperature, is introduced in the middle of the reaction. Then 960 samples are obtained as the testing data set, in which the first 480 samples are normal and the following 480 are faulty data.

In order to be able to trace the root cause of the fault, the child nodes must be selected here to test the fault. Randomly select one of the child nodes x_7 of the graphical model as the experimental object. According to the causal structure, it is easy to see that x_7 is directly related to x_5 . Here x_5 is the parent node of x_7 , so first calculate the conditional probability density $f(x_7|x_5)$. Figure 14.4 gives the graphical representation of the probability relationship between these two variables. Figure 14.4a depicts the probability density of normal data and fault data as a function of sampling time. Based on the fusion lasso method, the obtained KDE estimation is used as a rough signal for denoising and restoration. The crossed line in Fig. 14.4b represents the KDE recovered after denoising, which is set as the dynamical threshold. It can be clearly seen that after about 480 samples, the conditional probability of x_7 has exceeded the normal limit. Based on the FLSA method, the obtained KDE estimation is used as a rough signal for denoising and restoration.

Fault tracing refers to finding the root cause of failure in x_7 . The existing graph model can clearly show the causal relationship between nodes, so the propagation path of the fault can be easily analyzed. Carry out the reverse reasoning based on the established causal structure parameter model. Start from the failure variable and



(a) 3D plot of joint probability density of x_7 and x_5



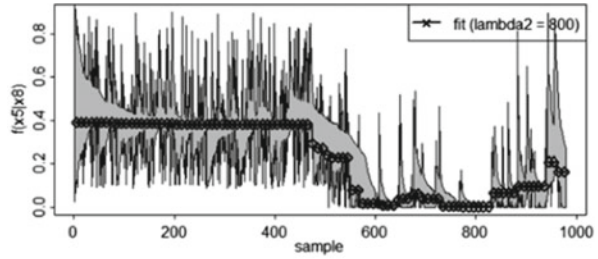
(b) Conditional probability of x_7 under x_5 corresponds to the sampling time

Fig. 14.4 Conditional probability of x_7 under x_5

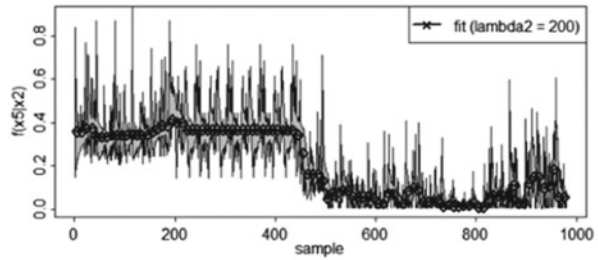
calculate the probability density function of its parent node in turn. The probability density curves obtained under normal and fault conditions are compared to determine whether the variables on each path are faulty. Continue this step until finding the root cause of the failure. In order to conversely infer the roots of fault x_7 , it is necessary to calculate $f(x_5|x_8)$, $f(x_5|x_2)$, $f(x_2)$, $f(x_8)$ separately. Simulation results are shown in Fig. 14.5.

From the detection result graph, the true propagation path of the fault can be analyzed. The test shows that the root of the fault is x_8 . Corresponding to the physical meaning of the variable, the root cause is the temperature of the cooling water, and fault IDV(4) is a step change in the temperature of the cooling water. The result is consistent with the actual process.

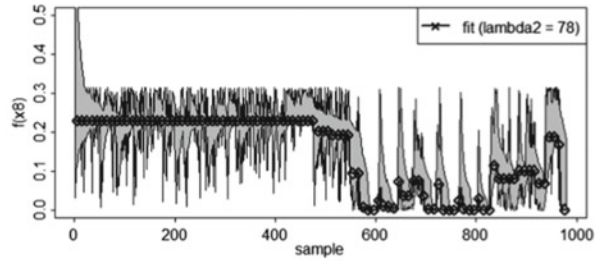
Fig. 14.5 Conditional probability densities of x_5 under x_8 , x_5 under x_2 ; probability densities of x_2, x_8



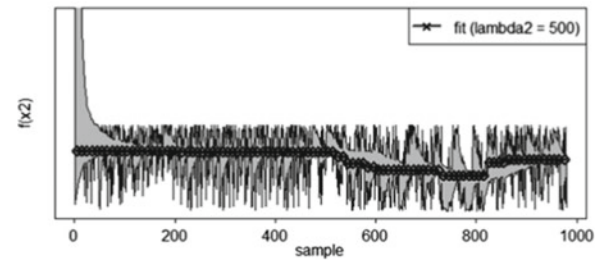
(a) Conditional probability density of x_5 under x_8



(b) Conditional probability density of x_5 under x_2



(c) Probability density of x_8



(d) Probability density of x_2

14.5 Conclusions

This chapter proposes a probability graph model directly for the continuous process variables aiming at the fault detection and root tracing. The model structure is determined by the causal relationship, and the probability relationship in the model is determined by the KDE method. For the child nodes in the causal structure, i.e., variables affected by other nodes, the conditional probability density functions are calculated based on the multidimensional joint probability density and the low-dimensional probability density. It reflects the strength relationship of the causal connection between the variables. An MISE index is rigorously derived to evaluate the estimation accuracy of KDE and optimize the KDE parameters. A dynamic threshold is constructed based on the FLSA algorithm to check the change of probability density, further to detect the fault. The experiment results in the TE process show that the proposed method not only accurately detects the occurrence of the failure, but also succeeds in finding its root cause.

References

- Bensi M, Kiureghian AD, Straub D (2013) Efficient bayesian network modeling of systems. *Reliab Eng Syst Saf* 112:200–213
- Chen X, Wang J, Zhou J (2018) Process monitoring based on multivariate causality analysis and probability inference. *IEEE Access* 6:6360–6369
- Chen X, Wang J, Zhou J (2018) Probability density estimation and bayesian causal analysis based fault detection and root identification. *Ind Eng Chem Res* 57(43):14656–14664
- Jiang W, Nicholas Z (2014) A probabilistic graphical model based stochastic input model construction. *J Comput Phys* 272(10):664–685
- Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. *Proc IEEE* 77:257–286
- Shimizu S, Inazumi T, Sogawa Y, Hyvarinen A, Kawahara Y, Washio T, Hoyer PO, Bollen K (2011) DirectLiNGAM: a direct method for learning a linear non-gaussian structural equation model. *J Mach Learn Res* 12(2):1225–1248
- Silverman BW (1998) *Density estimation for statistics and data analysis*. Routledge, Boca Raton
- Zeng J, Luo S, Cai J, Kruge U (2017) Nonparametric density estimation of hierarchical probabilistic graph models for assumption-free monitoring. *Ind Eng Chem Res* 56(5):1278–1287

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

