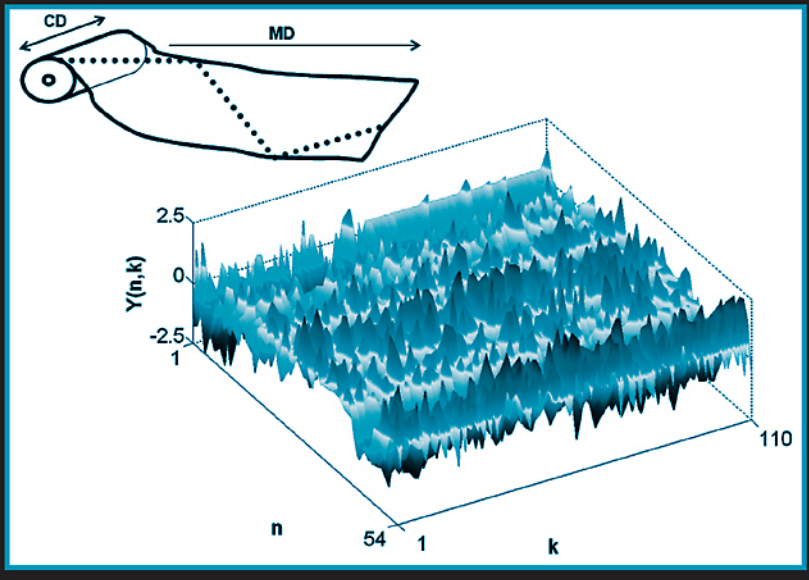
Chemical Process Performance Evaluation



Ali Cinar
Ahmet Palazoglu
Ferhan Kayihan

CRC Press
Taylor & Francis Group

Chemical Process Performance Evaluation

CHEMICAL INDUSTRIES

A Series of Reference Books and Textbooks

Founding Editor

HEINZ HEINEMANN
Berkeley, California

Series Editor

JAMES G. SPEIGHT
Laramie, Wyoming

1. *Fluid Catalytic Cracking with Zeolite Catalysts,* Paul B. Venuto and E. Thomas Habib, Jr.
2. *Ethylene: Keystone to the Petrochemical Industry,* Ludwig Kniel, Olaf Winter, and Karl Stork
3. *The Chemistry and Technology of Petroleum,* James G. Speight
4. *The Desulfurization of Heavy Oils and Residua,* James G. Speight
5. *Catalysis of Organic Reactions,* edited by William R. Moser
6. *Acetylene-Based Chemicals from Coal and Other Natural Resources,* Robert J. Tedeschi
7. *Chemically Resistant Masonry,* Walter Lee Sheppard, Jr.
8. *Compressors and Expanders: Selection and Application for the Process Industry,* Heinz P. Bloch, Joseph A. Cameron, Frank M. Danowski, Jr., Ralph James, Jr., Judson S. Swearingen, and Marilyn E. Weightman
9. *Metering Pumps: Selection and Application,* James P. Poynton
10. *Hydrocarbons from Methanol,* Clarence D. Chang
11. *Form Flotation: Theory and Applications,* Ann N. Clarke and David J. Wilson
12. *The Chemistry and Technology of Coal,* James G. Speight
13. *Pneumatic and Hydraulic Conveying of Solids,* O. A. Williams
14. *Catalyst Manufacture: Laboratory and Commercial Preparations,* Alvin B. Stiles

15. *Characterization of Heterogeneous Catalysts,* edited by Francis Delannay
16. *BASIC Programs for Chemical Engineering Design,* James H. Weber
17. *Catalyst Poisoning,* L. Louis Hegedus and Robert W. McCabe
18. *Catalysis of Organic Reactions,* edited by John R. Kosak
19. *Adsorption Technology: A Step-by-Step Approach to Process Evaluation and Application,* edited by Frank L. Slejko
20. *Deactivation and Poisoning of Catalysts,* edited by Jacques Oudar and Henry Wise
21. *Catalysis and Surface Science: Developments in Chemicals from Methanol, Hydrotreating of Hydrocarbons, Catalyst Preparation, Monomers and Polymers, Photocatalysis and Photovoltaics,* edited by Heinz Heinemann and Gabor A. Somorjai
22. *Catalysis of Organic Reactions,* edited by Robert L. Augustine
23. *Modern Control Techniques for the Processing Industries,* T. H. Tsai, J. W. Lane, and C. S. Lin
24. *Temperature-Programmed Reduction for Solid Materials Characterization,* Alan Jones and Brian McNichol
25. *Catalytic Cracking: Catalysts, Chemistry, and Kinetics,* Bohdan W. Wojciechowski and Avelino Corma
26. *Chemical Reaction and Reactor Engineering,* edited by J. J. Carberry and A. Varma
27. *Filtration: Principles and Practices: Second Edition,* edited by Michael J. Matteson and Clyde Orr
28. *Corrosion Mechanisms,* edited by Florian Mansfeld
29. *Catalysis and Surface Properties of Liquid Metals and Alloys,* Yoshisada Ogino
30. *Catalyst Deactivation,* edited by Eugene E. Petersen and Alexis T. Bell
31. *Hydrogen Effects in Catalysis: Fundamentals and Practical Applications,* edited by Zoltán Paál and P. G. Menon
32. *Flow Management for Engineers and Scientists,* Nicholas P. Cheremisinoff and Paul N. Cheremisinoff
33. *Catalysis of Organic Reactions,* edited by Paul N. Rylander, Harold Greenfield, and Robert L. Augustine
34. *Powder and Bulk Solids Handling Processes: Instrumentation and Control,* Koichi Iinoya, Hiroaki Masuda, and Kinnosuke Watanabe
35. *Reverse Osmosis Technology: Applications for High-Purity-Water Production,* edited by Bipin S. Parekh
36. *Shape Selective Catalysis in Industrial Applications,* N. Y. Chen, William E. Garwood, and Frank G. Dwyer
37. *Alpha Olefins Applications Handbook,* edited by George R. Lappin and Joseph L. Sauer
38. *Process Modeling and Control in Chemical Industries,* edited by Kaddour Najim

39. *Clathrate Hydrates of Natural Gases,* E. Dendy Sloan, Jr.
40. *Catalysis of Organic Reactions,* edited by Dale W. Blackburn
41. *Fuel Science and Technology Handbook,* edited by James G. Speight
42. *Octane-Enhancing Zeolitic FCC Catalysts,* Julius Scherzer
43. Oxygen in Catalysis, Adam Bielanski and Jerzy Haber
44. *The Chemistry and Technology of Petroleum: Second Edition, Revised and Expanded,* James G. Speight
45. *Industrial Drying Equipment: Selection and Application,* C. M. van't Land
46. *Novel Production Methods for Ethylene, Light Hydrocarbons, and Aromatics*, edited by Lyle F. Albright, Billy L. Crynes, and Siegfried Nowak
47. *Catalysis of Organic Reactions*, edited by William E. Pascoe
48. *Synthetic Lubricants and High-Performance Functional Fluids*, edited by Ronald L. Shubkin
49. *Acetic Acid and Its Derivatives*, edited by Victor H. Agreda and Joseph R. Zoeller
50. *Properties and Applications of Perovskite-Type Oxides*, edited by L. G. Tejuca and J. L. G. Fierro
51. *Computer-Aided Design of Catalysts*, edited by E. Robert Becker and Carmo J. Pereira
52. *Models for Thermodynamic and Phase Equilibria Calculations*, edited by Stanley I. Sandler
53. *Catalysis of Organic Reactions*, edited by John R. Kosak and Thomas A. Johnson
54. *Composition and Analysis of Heavy Petroleum Fractions*, Klaus H. Altgelt and Mieczyslaw M. Boduszynski
55. *NMR Techniques in Catalysis,* edited by Alexis T. Bell and Alexander Pines
56. *Upgrading Petroleum Residues and Heavy Oils*, Murray R. Gray
57. *Methanol Production and Use*, edited by Wu-Hsun Cheng and Harold H. Kung
58. *Catalytic Hydroprocessing of Petroleum and Distillates*, edited by Michael C. Oballah and Stuart S. Shih
59. *The Chemistry and Technology of Coal: Second Edition, Revised and Expanded,* James G. Speight
60. *Lubricant Base Oil and Wax Processing*, Avilino Sequeira, Jr.
61. *Catalytic Naphtha Reforming: Science and Technology*, edited by George J. Antos, Abdullah M. Aitani, and José M. Parera
62. *Catalysis of Organic Reactions*, edited by Mike G. Scaros and Michael L. Prunier
63. *Catalyst Manufacture,* Alvin B. Stiles and Theodore A. Koch
64. *Handbook of Grignard Reagents,* edited by Gary S. Silverman and Philip E. Rakita

65. *Shape Selective Catalysis in Industrial Applications: Second Edition, Revised and Expanded*, N. Y. Chen, William E. Garwood, and Francis G. Dwyer
66. *Hydrocracking Science and Technology*, Julius Scherzer and A. J. Gruia
67. *Hydrotreating Technology for Pollution Control: Catalysts, Catalysis, and Processes*, edited by Mario L. Occelli and Russell Chianelli
68. *Catalysis of Organic Reactions*, edited by Russell E. Malz, Jr.
69. *Synthesis of Porous Materials: Zeolites, Clays, and Nanostructures,* edited by Mario L. Occelli and Henri Kessler
70. *Methane and Its Derivatives*, Sunggyu Lee
71. *Structured Catalysts and Reactors*, edited by Andrzej Cybulski and Jacob A. Moulijn
72. *Industrial Gases in Petrochemical Processing*, Harold Gunardson
73. *Clathrate Hydrates of Natural Gases: Second Edition, Revised and Expanded*, E. Dendy Sloan, Jr.
74. *Fluid Cracking Catalysts*, edited by Mario L. Occelli and Paul O'Connor
75. *Catalysis of Organic Reactions*, edited by Frank E. Herkes
76. *The Chemistry and Technology of Petroleum: Third Edition, Revised and Expanded*, James G. Speight
77. *Synthetic Lubricants and High-Performance Functional Fluids: Second Edition, Revised and Expanded,* Leslie R. Rudnick and Ronald L. Shubkin
78. *The Desulfurization of Heavy Oils and Residua, Second Edition, Revised and Expanded,* James G. Speight
79. *Reaction Kinetics and Reactor Design: Second Edition, Revised and Expanded,* John B. Butt
80. *Regulatory Chemicals Handbook*, Jennifer M. Spero, Bella Devito, and Louis Theodore
81. *Applied Parameter Estimation for Chemical Engineers*, Peter Englezos and Nicolas Kalogerakis
82. *Catalysis of Organic Reactions,* edited by Michael E. Ford
83. *The Chemical Process Industries Infrastructure: Function and Economics*, James R. Couper, O. Thomas Beasley, and W. Roy Penney
84. *Transport Phenomena Fundamentals*, Joel L. Plawsky
85. *Petroleum Refining Processes*, James G. Speight and Baki Özüm
86. *Health, Safety, and Accident Management in the Chemical Process Industries*, Ann Marie Flynn and Louis Theodore
87. *Plantwide Dynamic Simulators in Chemical Processing and Control*, William L. Luyben
88. *Chemical Reactor Design*, Peter Harriott
89. *Catalysis of Organic Reactions*, edited by Dennis G. Morrell

90. *Lubricant Additives: Chemistry and Applications*, edited by Leslie R. Rudnick
91. *Handbook of Fluidization and Fluid-Particle Systems*, edited by Wen-Ching Yang
92. *Conservation Equations and Modeling of Chemical and Biochemical Processes*, Said S. E. H. Elnashaie and Parag Garhyan
93. *Batch Fermentation: Modeling, Monitoring, and Control*, Ali Çinar, Gülnur Birol, Satish J. Parulekar, and Cenk Ündey
94. *Industrial Solvents Handbook, Second Edition*, Nicholas P. Cheremisinoff
95. *Petroleum and Gas Field Processing*, H. K. Abdel-Aal, Mohamed Aggour, and M. Fahim
96. *Chemical Process Engineering: Design and Economics*, Harry Silla
97. *Process Engineering Economics*, James R. Couper
98. *Re-Engineering the Chemical Processing Plant: Process Intensification*, edited by Andrzej Stankiewicz and Jacob A. Moulijn
99. *Thermodynamic Cycles: Computer-Aided Design and Optimization*, Chih Wu
100. *Catalytic Naphtha Reforming: Second Edition, Revised and Expanded*, edited by George T. Antos and Abdullah M. Aitani
101. *Handbook of MTBE and Other Gasoline Oxygenates*, edited by S. Halim Hamid and Mohammad Ashraf Ali
102. *Industrial Chemical Cresols and Downstream Derivatives*, Asim Kumar Mukhopadhyay
103. *Polymer Processing Instabilities: Control and Understanding*, edited by Savvas Hatzikiriakos and Kalman B . Migler
104. *Catalysis of Organic Reactions*, John Sowa
105. *Gasification Technologies: A Primer for Engineers and Scientists*, edited by John Rezaiyan and Nicholas P. Cheremisinoff
106. *Batch Processes*, edited by Ekaterini Korovessi and Andreas A. Linninger
107. *Introduction to Process Control*, Jose A. Romagnoli and Ahmet Palazoglu
108. *Metal Oxides: Chemistry and Applications*, edited by J. L. G. Fierro
109. *Molecular Modeling in Heavy Hydrocarbon Conversions*, Michael T. Klein, Ralph J. Bertolacini, Linda J. Broadbelt, Ankush Kumar and Gang Hou
110. *Structured Catalysts and Reactors, Second Edition*, edited by Andrzej Cybulski and Jacob A. Moulijn
111. *Synthetics, Mineral Oils, and Bio-Based Lubricants: Chemistry and Technology*, edited by Leslie R. Rudnick
112. *Alcoholic Fuels*, edited by Shelley Minteer

113. *Bubbles, Drops, and Particles in Non-Newtonian Fluids, Second Edition*, R. P. Chhabra
114. *The Chemistry and Technology of Petroleum, Fourth Edition*, James G. Speight
115. *Catalysis of Organic Reactions*, edited by Stephen R. Schmidt
116. *Process Chemistry of Lubricant Base Stocks*, Thomas R. Lynch
117. *Hydroprocessing of Heavy Oils and Residua*, edited by James G. Speight and Jorge Ancheyta
118. *Chemical Process Performance Evaluation*, Ali Cinar, Ahmet Palazoglu, and Ferhan Kayihan

Chemical Process Performance Evaluation

Ali Cinar
Illinois Institute of Technology
Chicago, Illinois, U.S.A.

Ahmet Palazoglu
University of California
Davis, California, U.S.A.

Ferhan Kayihan
Integrated Engineering Technologies
Tacoma, Washington, U.S.A.

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an informa business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 0-8493-3806-9 (Hardcover)
International Standard Book Number-13: 978-0-8493-3806-9 (Hardcover)

Library of Congress Cataloging-in-Publication Data

Cinar, Ali.
Chemical process performance evaluation / Ali Cinar, Ahmet Palazoglu, Ferhan Kayihan.
p. cm. -- (Chemical industries ; 117)
Includes bibliographical references and index.
ISBN 0-8493-3806-9 (alk. paper)
1. Chemical process control--Statistical methods. 2. Chemical industry--Quality control--Statistical methods. I. Palazoglu, Ahmet. II. Kayihan, Ferhan, 1948- III. Title. IV. Series.

TP155.75.C55 2007
660'.281--dc22 2006051787

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To Mine, Bedirhan and to the memory of my parents
(A. Cinar)

To Mine, Aycan, Omer and my parents
(A. Palazoglu)

To Gulsevin, Arkan, Tarhan and to the memory of my parents
(F. Kayihan)

for Their Love, Support and Inspiration.

Preface

As the demand for profitability and competitiveness increases in the global marketplace, industrial manufacturing operations face a growing pressure to maintain safety, flexibility and environmental compliance. This is a result of pushing the operational boundaries to maximize productivity that may sometimes compromise the safe and rational operational practices. To minimize costly plant shut-downs and to diminish the probability of accidents and catastrophic events, an industrial plant is kept under close surveillance by computerized process supervision and control systems that collect data from process units and analyze the data to assess process status. Over the years, analysis and diagnosis methods have evolved from simple control charts to more sophisticated statistical techniques and signal processing capabilities. The goal of this book is to introduce the reader to the fundamentals and applications of a variety of process performance evaluation approaches, including process monitoring, controller performance monitoring and fault diagnosis. The material covered represents a culmination of decades of theoretical and practical research carried out by the authors and is based on the early notes that supported several short courses that the authors gave over the years. It is intended as advanced study material for graduate students and can be used as a textbook for undergraduate or graduate courses on process monitoring. By emphasizing the balance between the practice and the theory of statistical monitoring and fault diagnosis, it would also be an excellent reference for industrial practitioners, as well as a resource for training courses.

The reader is expected to have a rudimentary knowledge of statistics and have an awareness of general monitoring and control concepts such as fault detection, diagnosis and feedback control. The book will be constructed upon these basic building blocks, introducing new concepts and techniques when necessary. The early chapters of the book present the reader with the use of multivariate statistics and various tools that one can use for process monitoring and diagnosis. This includes a chapter on empirical process modeling and another chapter on the modeling of process signals. In later chapters, several fault diagnosis methods and the means to discriminate between sensor faults and process upsets are discussed in detail. Then, the statistical modeling techniques are extended to the assessment of control performance. The book concludes with an extensive discussion on the use of data analysis techniques for the special case of web and sheet processes. Several case studies are included to demonstrate the implementation of the discussed methods and hopefully to motivate the readers to explore these ideas further in solving their own specific problems. The focus of this

book is on continuous processes. However, there are a number of process applications, especially in pharmaceuticals and specialty chemicals, where the batch mode of operation is used. The monitoring of such processes has been discussed in detail in another book by Cinar *et al.* [41].

For further information on the authors, the readers are referred to the individual Web pages: Ali Cinar, *www.chee.iit.edu/∼ cinar/*, Ahmet Palazoglu, *www.chms.ucdavis.edu/research/web/pse/ahmet/*, and Ferhan Kayihan, *ietek.net/*. Furthermore, for supplementary materials and corrections, the readers can access the publisher's Web site *www.crcpress.com* [1].

We are indebted to all our students and colleagues who, over the years, set the challenges and provided the enthusiasm that helped us tackle such an exciting and rewarding set of problems. Specifically, we would like to thank our students S. Beaver, J. DeCicco, F. Doymaz, S. Kendra, F. Kosebalaban-Tokatli, A. Negiz, A. Norvilas, A. Raich, W. Sun, E. Tatara, C. Undey and J. Wong, who have conducted the research related to the techniques discussed in the book. We thank our colleagues, Y. Arkun, F. J. Doyle III, K. A. McDonald, T. Ogunnaike, J. A. Romagnoli and D. Smith for many years of fruitful discussions, colored with lots of fun and good humor. We also would like to acknowledge CRC Press / Taylor & Francis for supporting this book project. This has been a wonderful experience for us and we hope that our readers share our excitement about the future of the field of process monitoring and evaluation.

Ali Cinar
Ahmet Palazoglu
Ferhan Kayihan

[1] Under the menu Electronic Products located on the left side of the screen, click on Downloads & Updates. A list of books in alphabetical order with Web downloads will appear. Locate this book by a search, or scroll down to it. After clicking on the book title, a brief summary of the book will appear. Go to the bottom of this screen and click on the hyperlinked 'Download' that is in a zip file.

Contents

Nomenclature

1 Introduction **1**
1.1 Motivation and Historical Perspective 2
1.2 Outline 4

2 Univariate Statistical Monitoring Techniques **7**
2.1 Statistics Concepts 8
2.2 Univariate SPM Techniques 11
2.2.1 Shewhart Control Charts 11
2.2.2 Cumulative Sum (CUSUM) Charts 18
2.2.3 Moving Average Monitoring Charts for Individual Measurements 19
2.2.4 Exponentially Weighted Moving Average Chart . . . 22
2.3 Monitoring Tools for Autocorreleated Data 22
2.3.1 Monitoring with Charts of Residuals 26
2.3.2 Monitoring with Detecting Changes in Model Parameters 27
2.4 Limitations of Univariate Monitoring Techniques 32
2.5 Summary 35

3 Multivariate Statistical Monitoring Techniques **37**
3.1 Principal Components Analysis 37
3.2 Canonical Variates Analysis 43
3.3 Independent Component Analysis 43
3.4 Contribution Plots 46
3.5 Linear Methods for Diagnosis 48
3.5.1 Clustering 48
3.5.2 Discriminant Analysis 50
3.5.3 Fisher's Discriminant Analysis 53

3.6 Nonlinear Methods for Diagnosis . . . 58
3.6.1 Neural Networks . . . 58
3.6.2 Kernel-Based Techniques . . . 64
3.6.3 Support Vector Machines . . . 66
3.7 Summary . . . 69

4 Empirical Model Development **73**
4.1 Regression Models . . . 75
4.2 PCA Models . . . 78
4.3 PLS Regression Models . . . 79
4.4 Input-Output Models of Dynamic Processes . . . 83
4.5 State-Space Models . . . 89
4.6 Summary . . . 97

5 Monitoring of Multivariate Processes **99**
5.1 SPM Methods Based on PCA . . . 100
5.2 SPM Methods Based on PLS . . . 105
5.3 SPM Using Dynamic Process Models . . . 108
5.4 Other MSPM Techniques . . . 112
5.5 Summary . . . 114

6 Characterization of Process Signals **115**
6.1 Wavelets . . . 115
6.1.1 Fourier Transform . . . 116
6.1.2 Continuous Wavelet Transform . . . 119
6.1.3 Discrete Wavelet Transform . . . 123
6.2 Filtering and Outlier Detection . . . 127
6.2.1 Simple Filters . . . 128
6.2.2 Wavelet Filters . . . 131
6.2.3 Robust Filter . . . 133
6.3 Signal Representation by Fuzzy Triangular Episodes . . . 135
6.4 Development of Markovian Models . . . 138
6.4.1 Markov Chains . . . 139
6.4.2 Hidden Markov Models . . . 141
6.5 Wavelet-Domain Hidden Markov Models . . . 145
6.6 Summary . . . 147

7 Process Fault Diagnosis **149**
7.1 Fault Diagnosis Using Triangular Episodes and HMMs . . . 149
7.1.1 CSTR Simulation . . . 152
7.1.2 Vacuum Column . . . 155
7.2 Fault Diagnosis Using Wavelet-Domain HMMs . . . 157

7.2.1 *pH* Neutralization Simulation . . . 161
7.2.2 CSTR Simulation . . . 164
7.3 Fault Diagnosis Using HMMs . . . 166
7.3.1 Case Study of HTST Pasteurization Process . . . 167
7.4 Fault Diagnosis Using Contribution Plots . . . 174
7.5 Fault Diagnosis with Statistical Methods . . . 179
7.6 Fault Diagnosis Using SVM . . . 191
7.7 Fault Diagnosis with Robust Techniques . . . 192
7.7.1 Robust Monitoring Strategy . . . 192
7.7.2 Pilot-Scale Distillation Column . . . 198
7.8 Summary . . . 202

8 Sensor Failure Detection and Diagnosis 203
8.1 Sensor FDD Using PLS and CVSS Models . . . 204
8.2 Real-Time Sensor FDD Using PCA-Based Techniques . . . 215
8.2.1 Methodology . . . 218
8.2.2 Case Study . . . 224
8.3 Summary . . . 230

9 Controller Performance Monitoring 231
9.1 Single-Loop Controller Performance Monitoring . . . 233
9.2 Multivariable Controller Performance Monitoring . . . 237
9.3 CPM for MPC . . . 238
9.4 Summary . . . 248

10 Web and Sheet Processes 251
10.1 Traditional Data Analysis . . . 252
10.1.1 MD/CD Decomposition . . . 252
10.1.2 Time Dependent Structure of Profile Data . . . 256
10.2 Orthogonal Decomposition of Profile Data . . . 257
10.2.1 Gram Polynomials . . . 259
10.2.2 Principal Components Analysis . . . 262
10.2.3 Flatness of Scanner Data . . . 264
10.3 Controller Performance . . . 268
10.3.1 MD Control Performance . . . 269
10.3.2 Model-Based CD Control Performance . . . 271
10.4 Summary . . . 274

Bibliography 277

Index 305

Nomenclature

Symbols

a	Number of principal components retained for a PC model
$a_{i<j}$	Transition probability between states i and j
$\mathbf{A}<\ \mathbf{B}$	State and input coefficient matrices in continuous state–space systems
b_i	Inner relation regression coefficient in PLS
b_j	Probability distribution for observation j
$d_i{}^{Q}(\mathbf{x})$	Quadratic discrimination score for the ith population
c_A	Concentration of species A
$CONT_{i<j}^{T^2}$	Total contribution of variable x_j to T^2
$cont_{i<j}^{T^2}$	Contribution of variable x_j to the normalized score $t_i \lhd S_i^2$
$\mathbf{C}<\ \mathbf{D}$	State and input coefficient matrices in output equation of state–space systems
$d(\mathbf{x}<\mathbf{y})$	Distance between $\mathbf{x}$ and $\mathbf{y}$
$d_i(\mathbf{x})$	Linear discriminant score for the ith population
$\mathbf{E}$	Residuals matrix ($n \times m$)
$e(k)$	Prediction error (residual) at time k
$E_{[a<b]}$	Episode of a signal between points a and b
$\mathbf{F}$	Residuals matrix of quality variables in PLS
F	Feature space

$F_L(d)$, $F_H(d)$	Soft–thresholding and hard–thresholding wavelet filters
$F_W(d)$	Wiener wavelet filter
$\mathbf{F}$, $\mathbf{G}$	State and input coefficient matrices in discrete–time state–space systems
J	Cost function, CPM performance measure
$K(\mathbf{u}, \mathbf{v})$	Kernel function
$\mathbf{M}$	Sphering matrix in ICA
M	Control horizon in MPC
m	Number of process variables in a data set
n	Number of samples in a data set
O	An observable output sequence in a HMM
$\mathbf{P}$	Loadings matrix ($m \times a$)
$\mathbf{p}$	Loadings vector ($m \times 1$)
$\mathbf{p}_i$	PC loading i, ordered eigenvector i of $\mathbf{X^T X}$
P	Prediction horizon in MPC
$\mathbf{Q}$	Weight matrix of quality variables in PLS
q	Flow rate
q	Number of quality variables in a data set
q	Shift operator in time series models
q^{-1}	Backward shift operator in time series models
$\mathbf{Q}$, $\mathbf{R}$	Positive definite weight matrices in MPC
$\mathbf{R}$	Residuals block matrix in multipass sensor FDD
R_i	Range of variable i
r_i	Residual based on the PC model for fault i
r_l	Autocorrelation at lag l
$r_{s,index}$	Sensor index of residuals

$r_{x\text{‹}y}$	Crosscorrelation between x and y
$RCI_{j\text{‹}\alpha}$	Residual contribution index for jth variable with confidence level α
$\mathbf{S}$	Covariance matrix
S	A Markov state
s_i	Score distance based on the PC model for fault i
s_i^2	Variance of variable i
$SCI_{j\text{‹}\alpha}$	Scores contribution index for jth variable with confidence level α
$\mathbf{S}_B$	Between–class scatter matrix
$\mathbf{S}_W$	Within–class scatter matrix
$\mathbf{S}_Y$	Total scatter matrix
$\mathbf{T}$	Scores matrix $(n \times a)$
$\mathbf{t}$	Scores vector $(n \times 1)$
T	Length of observation sequence in a HMM
T	Temperature
T^2	Hotelling's T^2 statistic
$TRAN$	Matrix defined in Eq. 7.2
$\mathbf{U}$	Scores matrix of quality variables in PLS
v	An observation symbol in a HMM
$\mathbf{v}_P$	Plant noise
$\mathbf{v}_y$	Output sensor noise
$\mathbf{w}$	FDA vectors to maximize scatter between classes
$w(t-\tau)$	A STFT window function centered at τ
$\mathbf{W}_1$‹ $\mathbf{W}_2$	Disturbance coefficients matrix to state variables and outputs, respectively
$\mathbf{W}$	Weight matrix of process variables in PLS

$\mathbf{W}$	Projection matrix
$\bar{x}$	Sample mean of variable x
$\mathbf{X}$	Process variables data matrix ($x \times m$)
$\mathbf{Y}$	Quality variables data matrix ($x \times q$)
$z(k)$	A discrete signal evaluated at time instant k
$z(t)$	A continuous signal evaluated at time t

Greek Characters

β	Low–pass filter constant
$\boldsymbol{\beta}$	Vector of regression coefficients
Δ	Magnitude of step change
ϵ	Random variation (uncorrelated zero-mean Gaussian), measurement error
$\gamma_{\#}$	CPM performance measures (#: $hist$, des)
κ	Ridge parameter
λ	A HMM
λ	Forgetting factor
λ_i	ith eigenvalue
ω	Frequency
π	Initial HMM state distribution
π_i	Classes of events such as distinct operation modes $i = 1^{\epsilon} \cdots {}^{\epsilon} g$
$\mathbf{\Sigma}$	Covariance matrix
σ	Standard deviation
$\boldsymbol{\theta}$	Model parameters vector
θ_E	Euclidian angle between points a and b with vertex at the origin

θ_M	Mahalanobis angle between a and b with vertex at origin
τ	Target for the mean, first–order system time constant
$\mathbf{\Phi}$	MPC cost function
ϕ	Autoregressive parameter, residual Mahalanobis angle
$\phi(k)$	MPC cost function at time k
$\phi : X \rightarrow F$	Nonlinear map from input space X to feature space F
$\psi(t)$	A wavelet function
$\psi_{s,u}$	A wavelet function with dilation parameter s and translation parameter u

Subscripts

$0, \mathbf{0}$	Initial conditions
c	Coolant
f	Feed
min	Minimum value of a variable
m, max	Maximum values of a variable
r	Reference state/value
s	Steady–state

Superscripts

T	Transpose of a matrix

Abbreviations

AIC	Akaike information criteria
ANN	Artificial neural network
AR	Autoregressive

ARIMA	Autoregressive integrated moving average
ARL	Average run length
ARMA	Autoregressive moving average
ARMAX	Autoregressive moving average with exogenous inputs
ARX	Autoregressive model with exogenous inputs
ASM	Abnormal situation management
ASR	Automatic speech recognition
BESI	Backward elimination sensor identification
BJ	Box-Jenkins
BSSIR	Backward substitution for sensor identification and reconstruction
CC	Correlation coefficient
CWT	Continuous wavelet transform
CLP	Closed-loop potential
CPCA	Consensus principal components analysis
CPM	Controller performance monitoring
CQI	Continuous quality improvement
CSTR	Continuous stirred tank reactor
CUMPRESS	Cumulative prediction sum of squares
CUSUM	Cumulative sum
CV	Canonical variate
CVA	Canonical variates analysis
CVSS	Canonical variate state space (models)
CL	Centerline of SPM chart
DWT	Discrete wavelet transform
DCS	Distributed control system
DMC	Dynamic matrix control

ECM	Expected cost of misclassification
EM	Expectation maximization
EWMA	Exponentially weighted moving average
FDA	Fisher's discriminant analysis
FDD	Fault detection and diagnosis
FFT	Fast Fourier transform
FPE	Final prediction error
FT	Fourier transform
GUI	Graphical user interface
HMM	Hidden Markov model
HMT	Hidden Markov tree
HPCA	Hierarchical principal components analysis
HPLS	Hierarchical partial least squares
HTST	High–temperature short–time pasteurization
ICA	Independent component analysis
KBS	Knowledge–based system
KDE	Kernel density estimation
LCL	Lower control limit
LWL	Lower warning limit
LFCM	Liquid–fed ceramic melter
LQG	Linear quadratic Gaussian (control problem)
LV	Latent variable
MSE	Mean square error
MA	Moving average
MBPCA	Multiblock principal components analysis
MBPLS	Multiblock partial least squares

MIMO	Multi–input multi–output
MM	Moving median filter
MPC	Model predictive control
MSPM	Multivariate statistical process monitoring
MV	Multivariate
MVC	Minimum variance control
NAR	Nonlinear autoregressive
NARMAX	Nonlinear ARMAX
NLPCA	Nonlinear principal components analysis
NLTS	Nonlinear time series
NO	Normal operation
NOR	Normal operating region
O–NLPCA	Orthogonal nonlinear principal components analysis
OE	Output error
PC	Principal component
PCA	Principal components analysis
PCD	Parameter change detection (method)
PCR	Principal components regression
PLS	Partial least squares (Projection to latent structures)
PLS	Partial least squares
PRESS	Prediction sum of squares
RSVS	Redundant sensor voting system
RTKBS	Real–time knowledge–based systems
RVWLS	Recursive variable weighted least squares
RWLS	Recursive weighted least squares
SPE	Squared prediction error

SFCM	Slurry–fed ceramic melter
SISO	Single–input single-output
SNR	Signal–to–noise ratio
SPC	Statistical process control
SPM	Statistical process monitoring
SQC	Statistical quality control
STFT	Short–time Fourier transform
SV	Singular values or support vectors
SVD	Singular value decomposition
SVM	Support vector machine
UCL	Upper control limit
UWL	Upper warning limit
WT	Wavelet transform

1

Introduction

Today, a number of process and controller performance monitoring techniques can provide an inexpensive, algorithmic means to assure and maintain process quality and safety without resorting to costly investments in hardware. These techniques also help maximize hardware utilization and efficiency. This book represents a compilation and overview of such techniques to help the reader gain a healthy understanding of the fundamentals and the current developments and get a glimpse of what the future may hold. This book is intended to be a resource and a reference source for those who are interested in evaluating the potential of these techniques for specific applications, and learn their strengths and limitations.

The goal of *statistical process monitoring* (SPM) is to detect the occurrence and the nature of operational changes that cause a process to deviate from its desired target. The methodology for detecting changes is based on statistical techniques that deal with the collection, classification, analysis and interpretation of data. This, then, needs to be followed by *process diagnosis* that aims at locating the root cause of the process change and enables the process operators to take necessary actions to correct the situation, thereby returning the process back to its desired operation.

The detection and diagnosis tasks can be carried out on the process measurements to obtain critical insights into the performance of not only the process itself but also the automatic control system that is deployed to assure normal operation. Today, the integration of such tasks into the process control software associated with Distributed Control Systems (DCS) is in progress. The technologies continue to advance, especially in the incorporation of multivariate statistics as well as recent developments in signal processing methods such as wavelets and hidden Markov models.

This chapter will first present the motivations behind the application of various statistical techniques to process measurements along with a historical view of the key technological developments in this area. This will be followed by an overview of each chapter to help guide the reader.

1.1 Motivation and Historical Perspective

Traditional statistical process control (SPC) has focused on monitoring quality variables based on reports from the quality control laboratory and if the quality variables are outside the range of their specifications, making adjustments to recover their desired levels (hence controlling the process). Often, on-line analyzers/sensors may not be available or may be costly for certain quality attributes (e.g., saltiness of potato chips, trace impurity content of an aqueous stream, number average molecular weight of a polymer) and could require analytical tests that yield results in hours or days. Today, for swift and robust detection of abnormal process operation, the process variables, that are much more frequently and directly measured, are used to infer process status. In other words, system temperatures, pressures and stream flow rates can be used as indicators of certain product properties in an indirect but often reliable manner. An added advantage of the use of process variables is their direct link to process faults, reducing the time for fault diagnosis.

With the ever-increasing recognition of the consequences of plant accidents on the plant personnel and the surrounding communities [216], the use of process variables in determining the process status has become an integral element of abnormal situation management (ASM) practices. Naturally, statistical techniques have been in the forefront of tools that have been employed by plant operators to avoid plant failures and catastrophic events. A consortium, called ASM, led by Honeywell and several chemical and petrochemical companies (*www.asmconsortium.com*) was established in 1992 and continues to offer technology solutions on alarm management and decision support systems.

From a historical perspective, with the introduction of univariate control charts by Walter A. Shewhart [267] of Bell Labs, the statistical quality control (SQC) has become an essential element of quality assurance efforts in the manufacturing industry. It was W.E. Deming who championed Shewhart's use of statistical measures for quality monitoring and established a series of quality management principles that resulted in substantial business improvements both in Japan and the U.S. [52].

The leading edge research conducted at Kodak during the 1970s and 1980s resulted in J.E. Jackson's landmark papers and book [120, 121, 122] that reformulated the SQC concepts within the context of multivariate statistics. The key element of these techniques was the Principal Components Analysis (PCA) that was introduced much earlier by K. Pearson in 1901 [225, 226] and H. Hotelling in 1933 [113]. In fact, the history of PCA can be traced back to the 1870s when E. Beltrami and C. Jordan first formulated the singular value decomposition. PCA reveals the key direc-

tions in the data set that exhibit the largest variance, by exploiting the cross correlations among the set of variables considered. The manifestation of multivariate statistics in regression modeling has been the development of partial least squares (PLS) by H. Wold [331] and later by S. Wold and H. Martens [85]. These concepts have been introduced to the chemical engineering community by J.F. MacGregor who led the deployment of key technological advances in continuous and batch monitoring to a variety of industrial applications [146, 153]. These efforts were complemented by the development of performance indexes that quantify the effectiveness of control systems by Harris [103].

One of the most influential books on the subject of PCA was by I.T. Jolliffe [128] who published recently a new edition [129] of his book. The book by Smilde *et al.* [276] is the most recent contribution to the literature on multivariate statistics, with special emphasis on chemical systems. Two books coauthored by R. Braatz [38, 260] review a number of fault detection and diagnosis techniques for chemical processes. Cinar [41] coauthored a book on monitoring of batch fermentation and fault diagnosis in batch process operations.

The use of mathematical and statistical modeling methods to relate chemical data sets to the state of the chemical system is referred to as *chemometrics*. A key figure in the development of chemometrics and its application to industrial problems has been B.R. Kowalski [18, 147, 319] who led the Center for Process Analytical Chemistry (CPAC) that was established in 1984. To aid qualitative and quantitative analysis of chemical data, Eigenvector Technologies Inc., a developer of independent commercial software, has provided a number of software solutions, primarily as a Matlab® Toolbox [328].

The industrial importance of monitoring technologies in the sheet and web forming processes has been emphasized chiefly by DuPont in their polymer manufacturing activities and by Weyerhaeuser in papermaking. Among many academic contributions towards the fundamental development of both control and monitoring methodologies for sheet processes, the works of Rawlings and Chien [244], Rigopoulos *et al.* [250, 251], Jiao *et al.* [124], Featherstone and Braatz [73] and Skelton *et al.* [275] are particularly significant.

There is a substantial body of work, with a new emphasis, now originating from China and Singapore, as well as from academic institutions in Taiwan, Korea and Hong Kong that aim to respond to the ever-increasing demands on quality assurance in the expanding local manufacturing industries (see, for example, [28, 84]).

Many industrial corporations espoused continuous quality control (C-QI) using six-sigma principles [4] which establish management strategies to

maintain product quality levels. The material presented in this book provide the framework and the tools to implement six-sigma on multivariate processes.

1.2 Outline

The book follows a rational presentation structure, starting with the fundamentals of univariate statistical techniques and a discussion on the implementation issues in Chapter 2. After stating the limitations of univariate techniques, Chapter 3 focuses on a number of multivariate statistical techniques that permit the evaluation of process performance and provide diagnostic insight. To exploit the information content of process measurements even further, Chapter 4 introduces several modeling strategies that are based on the utilization of input-output process data. Chapter 5 provides statistical process monitoring techniques for continuous processes and three case studies that demonstrate the techniques.

Complementary to the statistical techniques presented before, Chapter 6 reviews a number of process signal modeling methods that originally emerged from the signal processing community, and shows how they can be utilized in the context of process monitoring and diagnosis. Chapter 7 presents several case studies that show how the techniques can be implemented. The special case of sensor failures and their detection and diagnosis is considered worthy of a separate chapter (Chapter 8).

When a failure occurs during operation, the cause can be attributed not only to the process equipment, or the sensor network but also to the controller. Controller performance monitoring (CPM), considered as a subset of plantwide process monitoring and diagnosis activities, deserves a separate discussion. Thus, Chapter 9 provides an overview of controller performance monitoring tools and offers a case study to illustrate the key concepts.

The final chapter (Chapter 10) focuses on web and sheet forming processes. It demonstrates how the statistical techniques can be applied to evaluate process and control performance for quality assurance and to acquire fundamental insight towards the operation of such processes.

The Nomenclature section defines the variables and special characters as well as the acronyms used in the book. The reader is cautioned that, given the breadth of the subjects covered, to sustain a consistent nomenclature in the book and still be able to maintain fidelity to the traditional (historical) use of nomenclature for various techniques is a difficult if not an impossible task. Yet, the use of various indices and variable definitions should be clear within the context of each technique, and every attempt is made to eliminate potential conflicts. In addition, given the uniqueness of web and

sheet processes, the nomenclature in Chapter 10 should be regarded as mostly independent of the rest of the book.

The reader should consult the Publisher's Web site *www.crcpress.com* for supplementary materials and updates.

2

Univariate Statistical Monitoring Techniques

Traditional approaches in process performance evaluation rely on characteristics and time trends of critical process variables such as controlled variables and manipulated variables. Ranges of variation of these variables, their frequency of reaching hard constraints, or any abnormal trends in their behavior have been used by many experienced plant personnel to track process performance. Variances of these variables and their histograms have also been used. More formal techniques for process performance evaluation rely on the extension of statistical process control (SPC) to continuous processes.

The first applications of SPC were in discrete parts manufacturing. When the measured dimensions of a machined part were significantly different from their desirable values (exceeding the tolerances), the manufacturing operation was stopped, adjustments were made and the manufacturing unit was restarted. Work stoppage for adjustment had a cost in terms of lost production time and parts manufactured during startup that do not meet the specifications. Consequently, manufacturing was interrupted to 'control' the process when the cost of off-specification production exceeded the cost of adjustment. The statistical techniques and graphical tools to assess this trade-off were called statistical process *control*. Adjustments in continuous processes such as distillation, reforming or catalytic cracking in refineries do not necessitate work stoppage, but the material and/or energy flow to the process is adjusted incrementally. Hence, there are no contributions to the cost of adjustment from work stoppage. Adjustments are made frequently by using automatic control techniques such as feedback and/or feedforward control [253]. To discriminate such control from SPC, the term engineering process control has been used in the SPC community. In fact, the task of performance evaluation has become 'monitoring' the operation of the process (which may be regulated using automatic control techniques) to

determine if the process is performing as desired. Consequently, the terms *statistical process monitoring* (SPM) and *automatic control* are used in this book.

Process monitoring is implemented as a periodically repeated **hypothesis testing** that checks if

- the *mean value* of a process variable has not shifted away from its *target value*, and
- the *spread* of a process variable has not changed significantly.

Simple graphical procedures (*monitoring charts*) are used to emulate hypothesis testing.

Some statistics concepts such as mean, range, and variance, test of hypothesis, and Type I and Type II errors are introduced in Section 2.1. Various univariate SPM techniques are presented in Section 2.2. The critical assumptions in these techniques include independence and identical distribution (*iid*) of data. The independence assumption is violated if data are autocorrelated. Section 2.3 illustrates the pitfalls of using such SPM techniques with strongly autocorrelated data and outlines SPM techniques for autocorrelated data. Section 2.4 presents the shortcomings of using univariate SPM techniques for multivariate data.

2.1 Statistics Concepts

One or more observations may be made at each sampling instant. The collection of all observations from a *population* at a specific sampling time is called *a sample*. Significant variation in process behavior is detected by monitoring changes in the *location* (central tendency) by inspecting the sample mean, median, or mode, and in the sample *spread* (scatter) by inspecting the sample range or standard deviation. Process variables may have different types of probability distributions. However, if a variable is influenced by many inputs having different probability distributions, then the probability distribution of the process variable approaches Normal (Gaussian) distribution asymptotically. The central limit theorem justifies the Normality assumption: Consider the independent random variables $x_1, x_2, \cdots, x_m$ with mean μ_i and variance σ_i , $i = 1, \cdots, m$. If $y = x_1 + x_2 + \cdots + x_m$ then the distribution of

$$\frac{1}{\sqrt{\sum_{i=1}^{m} \sigma_i^2}} \quad \left(y - \sum_{i=1}^{m} \mu_i\right) \tag{2.1}$$

approaches $N(0, 1)$ as m approaches infinity. Here, $N(0, 1)$ denotes the Normal probability distribution with mean 0 and variance 1.

Table 2.1. Population and sample statistics.

Statistic	Population (size m_p)	Sample (size m_s)
Mean	$\mu = \frac{1}{m_p}\sum_{i=1}^{m_p} x_i$	$\bar{x} = \frac{1}{m_s}\sum_{i=1}^{m_s} x_i$
Variance	$\sigma^2 = \frac{1}{m_p}\sum_{i=1}^{m_p}(x_i-\mu)^2$	$S^2 = \frac{1}{m_s-1}\sum_{i=1}^{m_s}(x_i-\bar{x})^2$
Range	$R_i = max(x_i) - min(x_i) \;\; i = 1, \cdots, m_p \; or \; m_s$	

The characteristics of a population that follows the Normal distribution are summarized by its mean and variance. Variance can also be inferred from the range of variables for small sample sizes. The convention on summation and representation of mean values is

$$\bar{x}_{i\cdot} = \frac{1}{m}\sum_{j=1}^{m} x_{ij} \;, \qquad \bar{x}_{\cdot\cdot} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \tag{2.2}$$

where n is the number of samples (groups) and m is the number of observations in a sample (sample size). The subscripts '.' indicate the index used in averaging. When there is no ambiguity, average values are denoted in the book using only $\bar{x}$ and $\bar{\bar{x}}$. The population and sample statistics for variables that have a Normal distribution are given in Table 2.1.

In chemical processes, often a single measurement of a process or a product variable is made at a sampling instant. The lack of multiple observations limits the use of classical Shewhart charts (Section 2.2.1). The single observation at each sampling time and the existence of random measurement errors have made SPM techniques based on cumulative sums, moving averages and moving ranges attractive for performance evaluation.

Often decisions have to be made about populations based on the information from a sample. A *statistical hypothesis* is an assumption or a guess about the population. It is expressed as a statement about the parameters of the probability distributions of the populations. Procedures that enable decision making whether to accept or reject a hypothesis are called *tests of hypotheses*. For example, if the equality of the mean of a variable (μ) to a value a is to be tested, the hypotheses are:

Null hypothesis: $H_0 : \quad \mu = a$
Alternate hypothesis: $H_1 : \quad \mu \neq a$

Two kinds of errors may be committed when testing a hypothesis: rejecting a hypothesis when it is true, and accepting a hypothesis when it is

false. The first is called Type I or α error. It is considered as the producer's risk since the manufacturer thinks that a product with acceptable properties is not acceptable to ship to customers and discards it. The second error is called Type II or β error. This is the consumer's risk because a defective product has not been detected and is sent to the customer. This can be summarized as,

Type I (α) error (Producer's risk): $$P\{reject\ H_0 \mid H_0\ is\ true\}$$

Type II (β) error (Consumer's risk): $$P\{fail\ to\ reject\ H_0 \mid H_0\ is\ false\}$$

In the development of the SPM chart, first α is selected to compute the confidence limit for testing the hypothesis. Then, a test procedure is designed to obtain a small value for β, if possible. α is a function of sample size and is reduced as sample size increases. Figure 2.1 displays graphically the α and β errors for a variable that has Normal distribution. In the upper plot, the area under the curve to the left of the line denoting the value $x = a$ is the α error. In the lower plot, the mean of x has shifted from x_1 to x_2. The area to the right of the line $x = a$ denotes the β error.

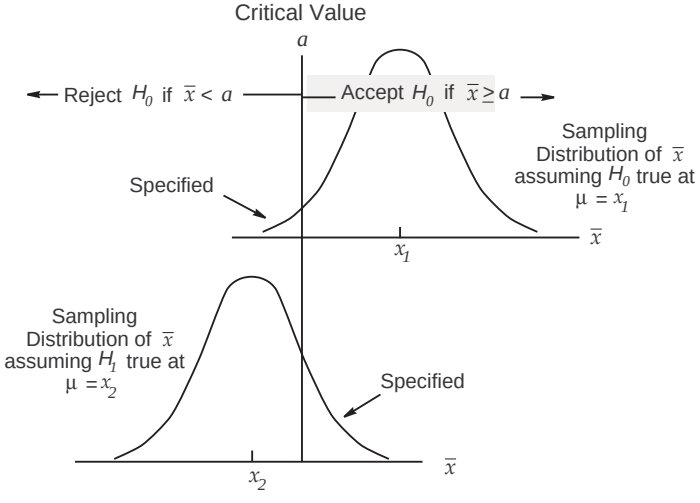


Figure 2.1. Type I (α) and Type II (β) errors.

The value for α error can be computed for simple SPC charts such as Shewhart charts using theoretical derivations. For more complex SPC

techniques this is not possible and other approaches such as computation of average run lengths (Section 2.2.1) are used to estimate α and β errors.

2.2 Univariate SPM Techniques

The SPM techniques used for monitoring a single variable include Shewhart, cumulative sum (CUSUM), moving average (MA), and exponentially weighted moving average (EWMA) charts. Shewhart charts consider only the current observation in assessing the process performance (Figure 2.2). CUSUM and MA charts give an equal weight to all observations that they use in performance assessment. While CUSUM charts consider all measurements since the beginning of the campaign, MA charts use a sliding window that discards old measurements. EWMA charts use a 'functional sliding window' by gradually forgetting past values and emphasizing the information in more recent observations.

Since in most chemical processes each measurement is made only once at each sampling time (no repeated measurements), all univariate monitoring charts will be developed for single observations except for Shewhart charts.

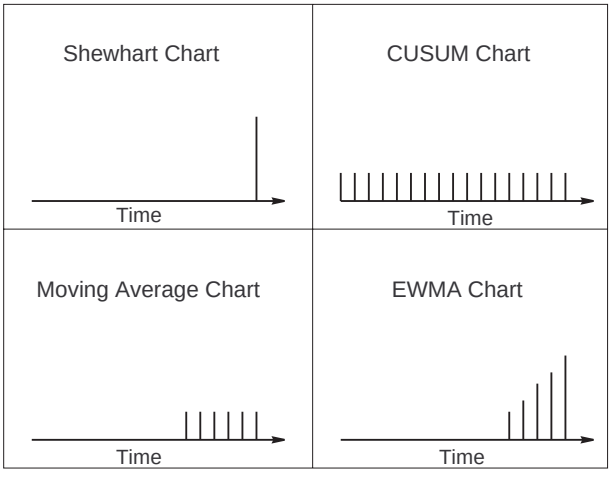


Figure 2.2. Schematic representation of univariate SPC charts.

2.2.1 Shewhart Control Charts

Shewhart charts indicate that a *special (assignable) cause* of variation is present when the sample data point plotted is outside the control limits. A

graphical *test of hypothesis* is performed by plotting the sample mean, and the range or standard deviation and comparing them against their control limits. A Shewhart chart is designed by specifying the *centerline* (CL), the *upper control limit* (UCL) and the *lower control limit* (LCL).

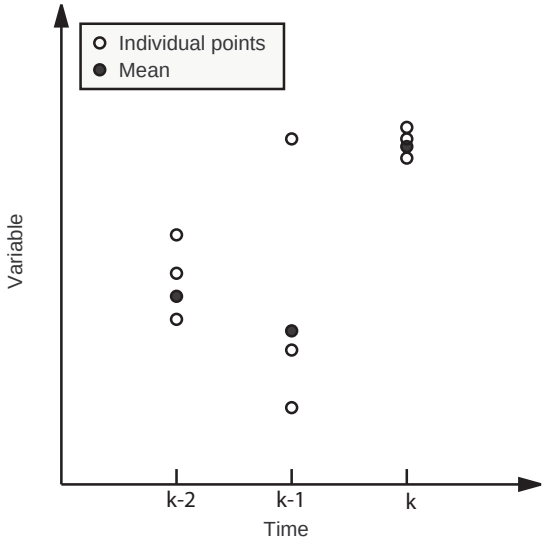


Figure 2.3. A dot diagram of individual observations of a variable.

Two Shewhart charts (sample mean and standard deviation or the range) are plotted simultaneously. Sample means are inspected to assess *between samples* variation (process variability over time) by plotting the Shewhart mean chart ($\bar{x}$ chart, $\bar{x}$ represents the average (mean) of x). However, one has to make sure that there is no significant change in *within sample* variation which may give an erroneous impression of changes in *between samples* variation. The mean values at times $k-2$ and $k-1$ in Figure 2.3 look similar but within sample variation at time $k-1$ is significantly different than that of the sample at time $k-2$. Hence, it is misleading to state that between sample variation is negligible and the process level is constant. Within sample variations of samples at times $k-2$ and k are similar, consequently, the difference in variation between samples is meaningful.

The *Range chart* (R chart), or the *standard deviation chart*, is used (S chart) to monitor *within sample* process variation or spread (process variability at a given time). The process spread must be in-control for proper interpretation of the $\bar{x}$ chart. The $\bar{x}$ chart must be used together with a spread chart.

The assumptions of Shewhart charts are:

- The distribution of the data is approximately Normal.
- The sample group sizes are equal.
- All sample groups are weighted equally.
- The observations are independent.

If only one observation is available, individual values can be used to develop the x chart (rather than the $\bar{x}$ chart) and the range chart is developed by using the 'moving range' concept discussed in Subsection 2.2.3.

Describing Variation The *location* or central tendency of a variable is described by its mean, median or mode. The *spread* or scatter of a variable is described by its range or standard deviation. For small sample sizes ($n < 6$, n = number of observations in a sampling time), the range chart or the standard deviation chart can be used. For larger sample sizes, the efficiency of computing the variance from the range is reduced drastically. Hence, the standard deviation charts should be used when $n > 10$.

Selection of Control Limits Three parameters affect the control limit selection:

i. the estimate of average level of the variable,

ii. the variable spread expressed as range or standard deviation, and

iii. a constant based on the probability of Type I error, α.

The '3σ' (σ denoting the standard deviation of the variable) control limits are the most popular control limits. The constant 3 yields a Type I error probability of 0.00135 on each side ($\alpha = 0.0027$). The control limits expressed as a function of population standard deviation σ are:

$$UCL = \text{Target} + 3\sigma, \qquad LCL = \text{Target} - 3\sigma \tag{2.3}$$

The $\bar{x}$ chart considers *only the current data value* in assessing the status of the process. *Run rules* have been developed to include historical information such as trends in data. The run rules sensitize the chart, but they also increase the false alarm probability. The warning limits are useful in developing additional run rules in order to increase the sensitivity of Shewhart charts. The warning limits are established at '2-sigma' level, which corresponds to $\alpha/2$=0.02275. Hence,

$$UWL = \text{Target} + 2\sigma \qquad LWL = \text{Target} - 2\sigma \tag{2.4}$$

If r run rules are used simultaneously and rule i has a Type I error probability of α_i, the overall Type I error probability α_{total} is

$$\alpha_{total} = 1 - \prod_{i=1}^{r}(1 - \alpha_i) \tag{2.5}$$

If 3 rules are used simultaneously and $\alpha_i = 0.05$, then $\alpha = 0.143$. For $\alpha_i = 0.01$, one would have $\alpha = 0.0297$.

Run rules, also known as Western Electric Rules [323], enable decision making based on trends in data. A process is declared out-of-control if any run rules are met. Some of the run rules are:

- One point outside the control limits.
- Two of three consecutive points outside the 2σ warning limits but still inside the control limits.
- Four of five consecutive points outside the 1σ limits.
- Eight consecutive points on one side of the centerline.
- Eight consecutive points forming a *run* up or a *run* down.
- A nonrandom or unusual pattern in the data.

Patterns in data could be any systematic behavior such as shifts in process level, cyclic (periodic) behavior, stratification (points clustering around the centerline), trends or drifts.

The Mean and Range Charts

Development of the $\bar{x}$ and R charts starts with the R chart. Since the control limits of the $\bar{x}$ chart depends on process variability, its limits are not meaningful before R is in-control.

The Range Chart

Range is the difference between the maximum and minimum observations in a sample. If there are n samples of size m, then

$$R_i = x_{\max,i} - x_{\min,i} \quad ; \quad \bar{R} = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{2.6}$$

The random variable R/σ is called the *relative range*. The parameters of its distribution depend on sample size m, with the mean being d_2 (Table 2.2). For example, $d_2 = 1.683$ for $m = 3$. An estimate of σ (the estimates are denoted by a hat $\hat{\cdot}$) can be computed from the range data by using

$$\hat{\sigma} = \bar{R}/d_2 \tag{2.7}$$

The standard deviation of R is estimated by using the standard deviation of R/σ, d_3:

$$\hat{\sigma}_R = d_3\sigma = d_3\frac{\bar{R}}{d_2} \tag{2.8}$$

The control limits of the R chart are

$$UCL, LCL = \bar{R} \pm 3d_3\frac{\bar{R}}{d_2} \tag{2.9}$$

Defining

$$D_3 = 1 - 3\frac{d_3}{d_2} \qquad \text{and} \qquad D_4 = 1 + 3\frac{d_3}{d_2} \tag{2.10}$$

the control limits of the R chart become

$$UCL = \bar{R}D_4 \qquad \text{and} \qquad LCL = \bar{R}D_3\ . \tag{2.11}$$

D_4 and D_3 for various values of m are given in Table 2.2.

The $\bar{x}$ chart

The estimator for the mean process level (centerline) is $\bar{\bar{x}}$. Since the estimate of *the standard deviation of the mean process level* σ is $\bar{R}/d_2$,

$$\frac{\sigma}{\sqrt{m}} = \frac{\bar{R}}{d_2\sqrt{m}} \tag{2.12}$$

The control limits for an $\bar{x}$ chart based on R are

$$UCL, LCL = \bar{\bar{x}} \pm A_2\bar{R}, \quad A_2 = \frac{3}{d_2\sqrt{m}} \tag{2.13}$$

and the values for A_2 are listed in Table 2.2.

The Mean and Standard Deviation Charts

The S chart is preferable for monitoring variation when the sample size is large or varying from sample to sample. Although S^2 is an unbiased estimate of σ^2, the sample standard deviation S is not an unbiased estimator of σ. For a variable with a Normal distribution, S estimates $c_4\sigma$, where c_4 is a parameter that depends on the sample size m. The standard deviation of S is $\sigma\sqrt{1-c_4^2}$. When σ is to be estimated from past data of n samples,

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{2.14}$$

and $\bar{S}/c_4$ is an unbiased estimator of σ. The exact values for c_4 are given in Table 2.2. An approximate relation based on sample size m is

$$c_4 \simeq \frac{4(m-1)}{4m-3} \tag{2.15}$$

Table 2.2. Control chart constants for various values of group size m.

	$\bar{X}$ and R Charts				$\bar{X}$ and S Charts			
	Chart for Averages ($\bar{X}$)	Chart for Range (R)			Chart for Averages ($\bar{X}$)	Chart for Standard Deviation (S)		
Group Size	Control Limits	Standard Deviation	Control Limits		Control Limits	Standard Deviation	Control Limits	
m	A_2	d_2	D_3	D_4	A_3	c_4	B_3	B_4
2	1.880	1.128	-	3.267	2.659	0.7979	-	3.267
3	1.023	1.693	-	2.574	1.954	0.8862	-	2.568
4	0.729	2.059	-	2.282	1.628	0.9213	-	2.266
5	0.577	2.326	-	2.114	1.427	0.9400	-	2.089
6	0.483	2.534	-	2.004	1.287	0.9515	0.030	1.970
7	0.419	2.704	0.076	1.924	1.182	0.9594	0.118	1.882
8	0.373	2.847	0.136	1.864	1.099	0.9650	0.185	1.815
9	0.337	2.970	0.184	1.816	1.032	0.9693	0.239	1.761
10	0.308	3.078	0.223	1.777	0.975	0.9727	0.284	1.716
11	0.285	3.173	0.256	1.744	0.927	0.9754	0.321	1.679
12	0.266	3.258	0.283	1.717	0.886	0.9776	0.354	1.646
13	0.249	3.336	0.307	1.693	0.850	0.9794	0.382	1.618
14	0.235	3.407	0.328	1.672	0.817	0.9810	0.406	1.594
15	0.223	3.472	0.347	1.653	0.789	0.9823	0.428	1.572
16	0.212	3.532	0.363	1.637	0.763	0.9835	0.448	1.552
17	0.203	3.588	0.378	1.622	0.739	0.9845	0.466	1.534
18	0.194	3.640	0.391	1.608	0.718	0.9854	0.482	1.518
19	0.187	3.689	0.403	1.597	0.698	0.9862	0.497	1.503
20	0.180	3.735	0.415	1.585	0.680	0.9869	0.510	1.490
21	0.173	3.778	0.425	1.575	0.663	0.9876	0.523	1.477
22	0.167	3.819	0.434	1.566	0.647	0.9882	0.534	1.466
23	0.162	3.858	0.443	1.557	0.633	0.9887	0.545	1.455
24	0.157	3.895	0.451	1.548	0.619	0.9892	0.555	1.445
25	0.153	3.931	0.459	1.541	0.606	0.9896	0.565	1.435

$$UCL_{\bar{X}}, LCL_{\bar{X}} = \bar{\bar{X}} \pm A_2\bar{R} \qquad UCL_{\bar{X}}, LCL_{\bar{X}} = \bar{\bar{X}} \pm A_3\bar{S}$$
$$UCL_R = D_4\bar{R} \qquad UCL_S = B_4\bar{S}$$
$$LCL_R = D_3\bar{R} \qquad LCL_S = B_3\bar{S}$$
$$\hat{\sigma} = \bar{R}/d_2 \qquad \hat{\sigma} = \bar{S}/c_4$$
$$D_3 = 1 - 3d_3/d_2 \qquad D_4 = 1 + 3d_3/d_2$$

The S Chart

The control limits of the S chart are

$$UCL, LCL = \bar{S} \pm 3\frac{\bar{S}}{c_4}\sqrt{1 - c_4^2}\,. \tag{2.16}$$

Defining the constants

$$B_3 = 1 - \frac{3}{c_4}\sqrt{1 - c_4^2} \quad \text{and} \quad B_4 = 1 + \frac{3}{c_4}\sqrt{1 - c_4^2} \tag{2.17}$$

the limits of the S chart are expressed as

$$UCL = B_4\bar{S} \qquad \text{and} \qquad LCL \ = B_3\bar{S} \tag{2.18}$$

The values for B_3 and B_4 are listed in Table 2.2.

The $\bar{x}$ Chart

When $\hat{\sigma} = \bar{S}/c_4$, the control limits for the $\bar{x}$ chart are

$$UCL, LCL = \bar{\bar{x}} \pm \frac{3}{c_4\sqrt{m}}\bar{S} \tag{2.19}$$

Defining the constant $A_3 = \frac{3}{c_4\sqrt{m}}$, the limits of the $\bar{x}$ chart become

$$UCL = \bar{\bar{x}} + A_3\bar{S} \qquad \text{and} \qquad LCL = \bar{\bar{x}} - A_3\bar{S} \tag{2.20}$$

with the values of A_3 given in Table 2.2.

Average Run Length

The *average run length* (ARL) is the average number of samples (or sample averages) plotted in order to get an indication that the process is out-of-control. ARL can be used to compare the efficacy of various SPC charts and methods. ARL(0) is the *in-control* ARL, i.e. the ARL to generate an out-of-control signal even though in reality the process remains in-control. The ARL to detect a shift in the mean of magnitude $\Delta\sigma$ is represented by ARL(Δ) where Δ is a constant and σ is the standard deviation of the variable. A good chart must have a high ARL(0) (for example, ARL(0) = 400 indicates that there is one false alarm on the average out of 400 successive samples plotted) and a low ARL(Δ) (bad news is displayed as soon as possible).

For a Shewhart chart, the ARL is calculated from

$$\text{ARL} \ = E[R] = \frac{1}{p} \tag{2.21}$$

where p is the probability that a sample exceeds the control limits, R is the run length and $E[\cdot]$ denotes the expected value. For an $\bar{x}$ chart with 3σ limits, the probability that a point will be outside the control limits even though the process is in control is $p = 0.0027$. Consequently, the ARL(0) is ARL $= 1/p = 1/0.0027 = 370$. For other types of charts such as CUSUM, it is difficult or impossible to derive ARL(0) values based on theoretical arguments. Instead, the magnitude of the level change to be detected is selected and Monte Carlo simulations are carried out to compute the run lengths, their averages and variances.

2.2.2 Cumulative Sum (CUSUM) Charts

The cumulative sum (CUSUM) chart incorporates all the information in a data sequence to highlight changes in the process average level. The values to be plotted on the chart are computed by subtracting the overall mean μ_0 from the data and then accumulating the differences. The quantity

$$S_i = \sum_{j=1}^{i} (x_j - \mu_0) \tag{2.22}$$

is plotted against the sample number i. CUSUM charts are more effective than Shewhart charts in detecting *small process shifts*, since they combine information from several samples. When several observations are available at each sampling time (sample size $m > 1$, the observation x_j is replaced by the sample average at time j, $\bar{x}_j$). The CUSUM values can be computed *recursively*

$$S_i = (x_i - \mu_0) + S_{i-1} \tag{2.23}$$

If the process is in-control at the target value μ_0, the CUSUM S_i should meander randomly in the vicinity of 0. If the process mean is shifted, an *upward or downward trend* will develop in the plot. Visual inspection of changes of slope indicates the sample number (and consequently the time) of the process shift. Even when the mean is on target, the CUSUM S_i may wander far from the zero line and give the appearance of a signal of change in the mean. Control limits in the form of a V-mask were employed when CUSUM charts were first proposed in order to decide that a statistically significant change in slope has occurred and the trend of the CUSUM plot is different than that of a random walk. CUSUM plots generated by a computer became more popular in recent years and the V-mask has been replaced by upper and lower confidence limits of one-sided CUSUM charts.

One-sided CUSUM charts are developed by plotting

$$S_i = \sum_{j=1}^{i} [\bar{x}_j - (\mu_0 + K)] \tag{2.24}$$

where K is the *reference value* to detect an increase in the mean level. If S_i becomes negative for $\mu_1 > \mu_0$, it is reset to zero. When S_i exceeds the decision interval H, a statistically significant increase in the mean level is declared. Values for K and H can be computed from the relations:

$$K = \frac{\Delta}{2}, \qquad H = \frac{d\Delta}{2} \tag{2.25}$$

Given the α and β probabilities, the size of the shift in the mean to be detected (Δ), and the standard deviation of the average value of the variable x ($\sigma_{\bar{x}}$), the parameters in Eq. 2.25 are:

$$\delta = \frac{\Delta}{\sigma_{\bar{x}}} \quad \text{and} \quad d = \left(\frac{2}{\delta^2}\right) \ln\left(\frac{1-\beta}{\alpha}\right) \tag{2.26}$$

A *two-sided* CUSUM chart can be generated by running two one-sided CUSUM charts simultaneously with the upper and lower reference values. The recursive formulae for *high* and *low side* shifts that include resetting to zero are

$$\begin{aligned} S_H(i) &= max\left[0, \bar{x}_i - (\mu_0 + K) + S_H(i-1)\right] \\ S_L(i) &= max\left[0, (\mu_0 - K) - \bar{x}_i + S_L(i-1)\right] \end{aligned} \tag{2.27}$$

respectively. The starting values are usually set to zero, $S_H(0) = S_L(0) = 0$. When $S_H(i)$ or $S_L(i)$ exceeds the *decision interval* H, the process is out-of-control. ARL-based methods are usually utilized to find the chart parameter values H and K. The rule of thumb for ARL(Δ) for detecting a shift of magnitude Δ in the mean when $\Delta \neq 0$ and $\Delta > K$ is

$$\text{ARL}(\Delta) = 1 + \frac{H}{\Delta - K} \tag{2.28}$$

2.2.3 Moving Average Monitoring Charts for Individual Measurements

Moving average (MA) *charts* are developed by selecting a data window length (l) that includes the consecutive samples used for computing the moving average. A new sample value is reported, the data window is moved by one sampling time increment, deleting the oldest data and including the most recent one. In MA charts, averages of the consecutive data groups of size l are plotted. The control limit computations are based on averages and standard deviation values computed from moving ranges. Since each MA point has $(l-1)$ common data points, the successive MAs are *highly autocorrelated* (autocorrelation is presented in Section 2.3). This autocorrelation is ignored in the usual construction of these charts. The MA control charts should not be used with strongly autocorrelated data. The MA charts detect small drifts efficiently (better than $\bar{x}$ chart) and they can be used when the original data do not have Normal distribution. The disadvantages of the MA charts are slow response to sudden shifts in level and the generation of autocorrelation in computed values.

Three approaches can be used for estimating S for individual measurements:

1. If a *rational blocking* of data exists, compute an estimate of S based on it. It is advisable to compare this estimate with the estimates obtained by using the other methods to check for discrepancies.

2. *The overall S estimate.* Use all the data together to calculate an overall standard deviation. This estimate of S will be inflated by the *between-sample* variation. Thus, it is an upper bound for $\hat{S}$. If there are changes in process level, compute S for each segment separately, then combine them by using

$$S_w = \sqrt{\frac{\sum_{i=1}^{h}(m_i - 1)S_i^2}{\sum_{i=1}^{h}(m_i - 1)}} \tag{2.29}$$

   where h is the number of segments with different process levels and m_i is the number of observations in each sample.

3. *Estimation of S by moving ranges of l successive data points.* Use differences of successive observations as if they were ranges of n observations. A plot of S for group size l versus l will indicate if there is between-sample variation. If the plot is flat, the between-sample variation is insignificant. This approach should not be used if there is a trend in data. If there are missing observations, all groups containing them should be excluded from computations.

The procedure for estimating S by moving ranges is:

1. Calculate moving ranges of size l, $l = 2, 3, \cdots$, using 25 to 100 observations.

$$MR(k) = \mid max(x_i) - min(x_i) \mid \ , \ i = (k - l + 1), k \tag{2.30}$$

2. Calculate the mean of the ranges for each l.

3. Divide the result of Step 2 by d_2 (Table 2.2) (for each l).

4. Tabulate and plot results for all l.

Process Level Monitoring by Moving Average (MA) Charts

In a moving average chart, the averages of consecutive groups of size l are computed and plotted. The control limit computations are based on these averages. Several original data points at the start and end of the chart are excluded, since there are not enough data to compute the moving average at these times. The procedure for developing the MA chart consists of the following steps:

1. Compute the moving average $MA(k)$ of span l at time k as

$$MA(k) = \frac{x(k) + x(k-1) + \cdots + x(k-l+1)}{l} \tag{2.31}$$

2. Compute the variance of $MA(k)$

$$V(MA(k)) = \frac{1}{l^2} \sum_{i=k-l+1}^{k} V(x_i) = \frac{\sigma^2}{l} \tag{2.32}$$

Hence, $\sigma = \bar{S}/c_4\sqrt{l}$ or $\sigma = \overline{MR}/d_2$, using $\overline{MR}$ for $\bar{R}$. The values for the parameters c_4 and d_2 are listed in Table 2.2.

3. Compute the control limits with the centerline at $\bar{x}$:

$$UCL, LCL = \bar{x} \pm \frac{3\bar{S}}{c_4\sqrt{l}} \quad \text{or} \quad = \bar{x} \pm \frac{3\overline{MR}}{d_2} \tag{2.33}$$

In general, the span l and the magnitude of the shift to be detected are inversely related.

Spread Monitoring by Moving Range Charts

In a moving range chart, the range of two consecutive sample groups of size l are computed and plotted. For $l \geq 2$,

$$MR(k) = \mid max(x_i) - min(x_i) \mid , \quad i = (k-l+1), k \tag{2.34}$$

The computation procedure is:

1. Select the range size l. Often $l = 2$.

2. Obtain estimates of $\overline{MR}$ and $\sigma = \overline{MR}/d_2$ by using the moving ranges $MR(k)$ of length l. For a total of n samples:

$$\overline{MR} = \frac{1}{n-l+1} \sum_{k=1}^{n-l+1} MR(k) \tag{2.35}$$

3. Compute the control limits with the centerline at $\overline{MR}$:

$$LCL = D_3\overline{MR}, \qquad UCL = D_4\overline{MR} \tag{2.36}$$

The values for the parameters D_3 and D_4 are listed in Table 2.2 and $\sigma_R = d_3\bar{R}/d_2$, and d_2 and d_3 depend on l.

2.2.4 Exponentially Weighted Moving Average Chart

The exponentially weighted moving average (EWMA) $z(k)$ is defined as

$$z(k) = wx(k) + (1 - w)z(k - 1) \tag{2.37}$$

where $0 < w \leq 1$ is a constant weight, $x(k)$ is the sample at time k, and the starting value at $k = 1$ is $z(0) = \bar{x}$. EWMA attaches a higher weight to more recent data and has a fading memory where old data are discarded from the average. Since the EWMA is a weighted average of several consecutive observations, it is insensitive to nonnormality in the distribution of the data. It is a very useful chart for plotting individual observations ($m = 1$). If $x(k)$ are independent random variables with variance σ^2, the variance of $z(k)$ is

$$\sigma_z^2(k) = \sigma^2 \Big(\frac{w}{2 - w} \Big) \big[1 - (1 - w)^{2k} \big] \tag{2.38}$$

The last term in brackets in Eq. 2.38 quickly approaches 1 as k increases and the variance reaches a limiting value. Often the asymptotic expression for the variance is used for computing the control limits. The weight constant w determines the memory of EWMA, the rate of decay of past sample information. For $w = 1$, the chart becomes a Shewhart chart. As $w \to 0$, EWMA approaches CUSUM. A good value for most cases is in the range $0.2 \leq w \leq 0.3$. A more appropriate value of w for a specific application can be computed by considering the ARL for detecting a specific magnitude of level shift or by searching w which minimizes the prediction error for a historical data set by an iterative least squares procedure. 50 or more observations should be utilized in such procedures. EWMA is also known as *geometric moving average*, *exponential smoothing*, or *first-order filter* (Section 6.2.1).

Upper and the lower control limits for an EWMA chart are calculated as

$$\begin{aligned} UCL(k) &= \mu_0 + 3\sigma_{z(k)} \\ CL &= \mu_0 \\ LCL(k) &= \mu_0 - 3\sigma_{z(k)} \end{aligned} \tag{2.39}$$

2.3 Monitoring Tools for Autocorreleated Data

Whenever there are *inertial elements* (capacity) in a process such as storage tanks, reactors or separation columns, the observations from such processes exhibit serial correlation over time. Successive observations are related to

each other. Characteristics of process disturbances in continuous processes include:

- Changes in level – typical forms of disturbance trajectories include step changes and exponential (overdamped) variations usually observed, for example, in feed composition, temperature or impurity levels,
- Drifts, ramps, or meandering trajectories that describe catalyst deactivation, fouling of heat transfer surfaces,
- Random variations such as erratic pump or control valve behavior.

The process mean $\mu(k)$ at time k varies over time with respect to the target or nominal value for the mean τ:

$$\mu(k) - \tau = \phi(\mu(k-1) - \tau) + \epsilon(k) + \delta_0(k)\Gamma \tag{2.40}$$

where $\epsilon(k)$ is an *iid* random variation in the mean, 'driving force' for random disturbances, ϕ is an autoregressive parameter $-1 \leq \phi \leq 1$, and Γ is the magnitude of an abrupt (step) or sustained incremental (ramp) level change in the variable. The serial correlation is mathematically described by the autoregressive term ϕ.

The strength of correlation dies out as the number of sampling intervals between observations increases. In other words, as the sampling interval increases, the correlation between successive samples decreases. In some industrial monitoring systems, a large sampling interval is selected in order to reduce correlation. The penalty for this mode of operation is *loss of information* about the dynamic behavior of the process. Such policies for circumventing the effects of autocorrelation in data should be avoided.

Statistics for Correlated Data

The correlation between observations made at different times (autocorrelation) is described mathematically by computing the *autocorrelation function*, the degree of correlation between observations made k time units apart ($k = 1, 2, \cdots$). The correlation coefficient is a measure of the *linear* association between two variables. It does not describe a cause-and-effect relation. The autocorrelation depends on *sampling interval*. Most statistical and mathematical software packages include routines for computing correlation and autocorrelation.

The sample *correlation function* between two variables x and y is denoted by $r_{x,y}$ and it is equal to:

$$r_{x,y} = \frac{\sum_{k=1}^{n}(x(k) - \bar{x})(y(k) - \bar{y})}{\left[\sum_{k=1}^{n}(x(k) - \bar{x})^2 \ \sum_{k=1}^{n}(y(k) - \bar{y})^2\right]^{1/2}} \tag{2.41}$$

where $\bar{x}$ and $\bar{y}$ are the sample means for x and y, respectively.

If the variable y is variable x shifted by l sampling times, the correlation between time shifted values of the same variable are described by

$$r_{x(k),x(k-l)} = r_l = \frac{\sum_{k=1}^{n-l}(x(k)-\bar{x})(x(k-l)-\bar{x})}{\sum_{k=1}^{n}(x(k)-\bar{x})^2} \tag{2.42}$$

Since the time series of only one variable is involved and the time lag l between the two time series is the parameter that changes, the autocorrelation coefficient is represented by r_l. The upper limit of the summation in the denominator varies with l. In order to have an equal number of data in both series, $n-l$ values are used in the summation.

The plot of autocorrelation r_l versus lag l is called autocorrelation function or *correlogram*. Usually the autocorrelation for $l = n/5$ lags are computed. *Confidence intervals* on individual sample autocorrelations can be computed for hypothesis testing: The approximate 95 % confidence interval for an individual r_t based on the assumption that all sample autocorrelations are equal to zero is $\pm 2/\sqrt{n}$.

A simple procedure is used for determining the number of lags l with non-zero autocorrelation:

- Compute the first $l = n/5$ autocorrelations.
- Compute the confidence interval $\pm 2/\sqrt{n}$
- Check if any autocorrelation coefficient is outside the confidence limit. Visual inspection of the plot of the autocorrelation function and numerical comparison of the autocorrelation coefficients with the confidence limits are the popular methods for the assessment of autocorrelation in data.

In general, the magnitude of r_l decreases as l increases.

Effects of Autocorrelation on SPC Methods

A process described by

$$\begin{aligned} x(k) &= \mu(k) + \epsilon_1(k) \\ \mu(k) - \tau &= \phi(\mu(k-1) - \tau) + \epsilon_2(k) + \delta(k_0)\Delta \end{aligned} \tag{2.43}$$

where $\epsilon_1(k)$ and $\epsilon_2(k)$ are *iid* random variables, ϕ is the autoregressive parameter with $-1 \leq \phi \leq 1$, $\mu(k)$ is the process mean at time k, and τ is the target or nominal value for the mean. Here, $\epsilon_1(k)$ denotes the inherent variability in the process due to causes such as measurement errors and $\epsilon_2(k)$ the random variation in the mean, 'driving force' for the disturbances.

Table 2.3. ARL for detecting a fault of magnitude δ by CUSUM and EWMA charts for two levels of ϕ.

		CUSUM		EWMA	
δ	σ_{ϵ_1}	$\phi = 0.25$	$\phi = 0.75$	$\phi = 0.25$	$\phi = 0.75$
0	0.9	383	188	355	186
0	0.1	130	35	136	36
0.5	0.9	37	37	37	37
0.5	0.1	32	27	31	27
1	0.9	11	16	10.7	14
1	0.1	11	16	4.3	15.6
2	0.9	4.4	7	4.2	6.7
2	0.1	4.6	8	4.2	7.5

Simulation studies were conducted to determine the effect of low and high values of ϕ and low and high values of σ_ϵ on the ARL of CUSUM and EWMA charts [104]. The chart parameters were CUSUM: $K = 0.5$ and $H = 5$; and EWMA: $w = 0.18$ and $UCL = 2.9\sigma_z$. The ARLs for a step change in the mean introduced at $k_0 = 1$ with a magnitude of $\Delta = (1-\phi)\Delta^*$ (hence, the ultimate mean shift magnitude is Δ^*) were tabulated.

A subset of the ARL results from this study listed in Table 2.3 indicate that the in-control ARL are very sensitive to the presence of autocorrelation, but the detection capabilities of CUSUM and EWMA for true shifts are not significantly affected. In the absence of autocorrelation, the ARL(0) for CUSUM is 465 and that for EWMA is 452. The ARL(0) for low levels of autocorrelation ($\phi = 0.25$) are 383 and 355, respectively, and they drop drastically to 188 and 186 for high levels of autocorrelation ($\phi = 0.75$), increasing the false alarm rates by a factor of 2.5.

The effects of autocorrelation on monitoring charts have also been reported by other researchers for Shewhart [186] and CUSUM [343, 6] charts. Modification of the control limits of monitoring charts by assuming that the process can be represented by an autoregressive time series model (see Section 4.4 for terminology) of order 1 or 2, and use of recursive Kalman filter techniques for eliminating autocorrelation from process data have also been proposed

[66].

Two alternative methods for monitoring processes with autocorrelated data are discussed in the following sections. One method relies on the existence of a process model that can predict the observations and computes

the residuals between the predicted and computed values at each sampling time. As described in Section 2.3.1, it assumes that the residuals will have a Normal distribution with zero mean and consequently regular SPM charts could be used on the residuals to monitor process behavior. The second method uses a process model as well, but here the model is updated at each sampling time using the latest observations. As outlined in Section 2.3.2, it is assumed that model parameters will not change significantly while there are no drastic changes in the process. Hence, SPM is implemented by monitoring the changes in the parameters of this recursive model.

2.3.1 Monitoring with Charts of Residuals

Autocorrelation in data affects the accuracy of the charts developed based on the *iid* assumption. One way to reduce the impact of autocorrelation is to estimate the value of the observation from a model and compute the error between the measured and estimated values. The errors, also called *residuals*, are assumed to have a Normal distribution with zero mean. Consequently regular SPM charts such as Shewhart or CUSUM charts could be used on the residuals to monitor process behavior. This method relies on the existence of a process model that can predict the observations at each sampling time. Various techniques for empirical model development are presented in Chapter 4. The most popular modeling technique for SPM has been time series models [1, 202] outlined in Section 4.4, because they have been used extensively in the statistics community, but in reality any dynamic model could be used to estimate the observations. If a good process model is available, the prediction errors (residual) $e(k) = y(k) - \hat{y}(k)$ can be used to monitor the process status. If the model provides accurate predictions, the residuals have a Normal distribution and are independently distributed with mean zero and constant variance (equal to the prediction error variance).

Conventional Shewhart, CUSUM, and EWMA SPM charts can be developed for the residuals [1, 202, 173, 259]. Data points that are out-of-control or unusual patterns on such charts indicate that *the model does not represent the process any more.* Often this implies that the original variable $x(k)$ is out-of-control. However, the model may continue to represent the process when $x(k)$ is out-of-control. In this case, the residuals chart does not signal this behavior.

To reduce the burden of model development, use of EWMA equations have been proposed as a forecasting model [202]. The accuracy of predictions will depend on the representation capability of the EWMA model for a specific process [70, 176, 261]. If the observations from a process are positively correlated and the process mean does not drift too quickly, then

the EWMA predictor would provide a good one-step-ahead forecast. If the EWMA model is a good predictor, then the sequence of prediction errors $e(k)$ should be uncorrelated.

Considering the fact that $e(k)$ indicates only the degree of disparity between observations collected and their model predictions, the residuals charts may not be reliable for signaling significant variations in process mean. Plots of residuals are good in detecting upsets such as events that affect observations directly (for example sampling and measurement errors). They may perform poorly in detecting shifts in the mean, especially when the correlation is high and positive. One alternative is to develop a Shewhart chart for the EWMA prediction errors and use it along with a Shewhart chart of the original data. This way, the chart of the original observations gives a clearer picture of process dynamics (the process is out-of-control if the confidence interval excludes the target), while the residuals chart displays process information after accounting for autocorrelation in data (the residuals may remain small if the model continues to describe the process behavior accurately).

2.3.2 Monitoring with Detection of Changes in Model Parameters

An alternative SPM framework for autocorrelated data is developed by monitoring variations in time series model parameters that are updated at each new measurement instant. Parameter change detection with recursive weighted least squares was used to detect changes in the parameters and the order of a time series model that describes stock prices in financial markets [263]. Here, the recursive least squares is extended with adaptive forgetting.

Consider an autocorrelated process described by an autoregressive model $AR(p)$,

$$y(k) = \phi_1 y(k-1) + \cdots + \phi_p y(k-p) + \phi_{p+1} + \epsilon(k) \tag{2.44}$$

where $\epsilon(k)$ is an uncorrelated zero-mean Gaussian process with variance σ_ϵ^2 and ϕ_{p+1} is the constant term (bias) parameter. The parameter change detection (PCD) method monitors the magnitude of changes in model parameters $\phi(k)$ and signals an out-of-control status when the changes are greater than a specified threshold value. The estimate $\hat{\phi}_{p+1}(k)$ for a general $AR(p)$ model contains the process variable level $\bar{y}_k$ implicitly as

$$\hat{\phi}_{p+1}(k) = \bar{y}_k \left(1 - \sum_{i=1}^{p} \hat{\phi}_i(k) \right) \tag{2.45}$$

The first step in the PCD monitoring scheme is to establish the null hypothesis H_0. An AR model is developed from *calibration* data. The model information includes the model parameter vector $\hat{\phi}_o(n)$, the inverse covariance matrix $\mathbf{P}_o(n)$, and the noise (disturbance) variance $\hat{\sigma}_\epsilon^2$. Based on this information, the mean and variance of the model parameters are computed. The test against the alternate hypothesis involves updating of model parameters recursively at each measurement instant through recursive variable weighted least squares (RVWLS) with adaptive forgetting filter (Eqs. 2.46 – 2.49) as new measurement information becomes available. RVWLS with adaptive forgetting algorithm is summarized next.

For the $AR(p)$ model Eq. 2.44, the $(p+1) \times 1$ column vector $\mathbf{x}(k)$ is defined as $\mathbf{x}(k) = [y(k-1) \;\; y(k-2) \;\; \cdots \;\; y(k-p) \;\; 1]^T$ where $[.]^T$ denotes the transpose. RVWLS with adaptive forgetting is given by Eqs. 2.46 – 2.49:

$$\hat{\phi}(k) = \hat{\phi}(k-1) + \frac{\mathbf{P}(k-1)\mathbf{x}(k)}{(\lambda(k-1) + \mathbf{x}(k)^T\mathbf{P}(k-1)\mathbf{x}(k))}\,\hat{\epsilon}(k) \tag{2.46}$$

$$\mathbf{P}(k) = \frac{1}{\lambda(k-1)}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{x}(k)\mathbf{x}(k)^T\mathbf{P}(k-1)}{(\lambda(k-1) + \mathbf{x}(k)^T\mathbf{P}(k-1)\mathbf{x}(k))}\right] \tag{2.47}$$

$$\lambda(k) = \left[1 - \mathbf{x}(k)^T\mathbf{K}(k)\right] \tag{2.48}$$

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{x}(k)}{(\lambda + \mathbf{x}(k)^T\mathbf{P}(k-1)\mathbf{x}(k))} \tag{2.49}$$

The unit delay of the forgetting factor λ in Eqs. 2.46 – 2.49 is necessary to avoid a solution of a quadratic equation at each time step for $\lambda(k)$. This improves the steady-state performance of the filter and allows tracking when model parameters are changing. A λ value close to 1 averages out the effect of $\epsilon(k)$ while a λ close to 0 tracks more quickly parameter variation in time. The steady-state performance of the RVWLS when the parameters are not time-varying deteriorates due to the estimation noise, if the value of λ is kept away from unity. A good compromise for λ is when $0.95 \leq \lambda \leq 1.0$, which is not suitable to track fast changes in the parameters. Therefore, a scheme is needed to make λ small when the parameters are varying and make it close to 1 at other times.

Detection, Estimation and Discrimination

Assume that n observations are available to form the *calibration data* set. The parameter estimates $\hat{\phi}_o(n)$ and the variance estimate $\hat{\sigma}_\epsilon^2$ of the noise process $\epsilon(k)$ are computed. Under the null hypothesis H_0, the distribution of the parameter estimates after time n becomes $\hat{\phi}(k) \sim N(\hat{\phi}_o(n), \mathbf{P}_o(n)\hat{\sigma}_\epsilon^2)$,

$k \geq n$. The sequential change detection algorithm is based on

$$P\left(\mid \hat{\phi}_i^d(k) \mid, \mid \hat{\phi}_i^d(k+1) \mid, \cdots, \mid \hat{\phi}_i^d(k+n_c) \mid > \gamma\sqrt{p_{o_{ii}}(n)}\hat{\sigma}_\epsilon\right) \leq 0.5^{n_c} \tag{2.50}$$

where $\hat{\phi}_i^d(k) = \hat{\phi}_i(k) - \hat{\phi}_{o_i}(n)$, $i = 1, \cdots p+1,\ k > n$ and $p_{o_{ii}}(n)$ represents the ith diagonal of the inverse covariance matrix $\mathbf{P}_o(n)$. The design parameters n_c and γ depend on the AR parameters: The parameter γ is a positive valued threshold that is adjusted to reduce false alarms. The parameter n_c represents the length of a run necessary for declaring the process to be out-of-control. The stopping time for the sequential detection is the time when n_c successive parameter estimates are outside the limits in either positive or negative direction. The most common value for the run length n_c is 7. Once a change is detected, estimation is performed by reducing the value of the forgetting factor $\lambda(k)$ to a small value λ_o at that time step and then setting $\lambda = 1$ until the filter is converged. Updated parameter estimates are utilized to distinguish between a level and a structure change in the underlying AR model. Proper values must be selected for n, N_a, λ_o, n_c, and γ to design the SPM charts. The *RL distribution* under change and no change conditions are used for assessing the performance of the SPM schemes and selecting the values of the PCD method parameters.

The filter is initialized using the null hypothesis. Change detection is done by using the stopping rule suggested by Eq. 2.50. Two indicators are utilized to summarize the conclusions reached by the detection phase. One indicator signals if a change is detected in model parameters and if so which parameter has changed. The second indicator signals the direction of change (positive or negative). Determining the values of the two indicators concludes the *detection* phase of the PCD method.

If the alternate hypothesis is accepted at the detection phase, *estimation* of change by PCD method is initiated by reducing the forgetting factor to a small value at the detection instant. This will cause the filter to converge quickly to the new values of model parameters. Shewhart charts for each model parameter are used for observing the new identified values of the model parameters. At this point the out-of-control decision made at the detection phase can be reassessed. If the identified values of the parameters are inside the range defined by the null hypothesis, then the detection decision can be reversed and the alarm is declared false.

The *discrimination* phase of the method runs in parallel with the estimation phase. It tries to find out whether the change experienced is in the autoregressive parameters or in the constant term (level) of the autocorrelated process variable. The parameter estimates from the estimation phase are used to estimate the level parameter $\bar{y}_k$ (Eq. 2.45). If the alternate hypothesis is accepted, the change experienced involves variation

in the process mean. If the null hypothesis is accepted, then the change experienced does not involve the level of the process variable. If the null hypothesis is accepted, and a subset of the AR model parameters except the constant term parameter show signs of change, it is deduced that the AR process exhibits only a structure change. If the alternate hypothesis is accepted and a subset of the identified AR parameters (including the constant term parameter) are out-of-control, then a combined structure and level change is experienced.

Example The PCD method is used for monitoring a laboratory-scale spray dryer operation where fine aluminum oxide powder is produced by drying dilute solutions of water and aluminum oxide. On-line particle size and velocity, inlet hot air and exhaust air temperatures were measured. The SPM scheme based on on-line temperature measurements checks if the process is operating under the selected settings, and producing the desired particle size distribution [213]. $AR(3)$ models are used for both temperatures. The exhaust air temperature is modeled by

$$\begin{aligned} T(k) = 0.5885T(k-1) &+ 0.2385T(k-2) + .. \\ .. &+ 0.1595T(k-3) + 1.5384 + e(k) \end{aligned} \tag{2.51}$$

with the standard deviation of $e(k)$ equal to 0.4414 for the in-control data used in developing the model (Hypothesis H_0) and 0.4915 for the data with the slurry pump speed disturbance (Figure 2.4).

Figure 2.4 shows new process data where the slurry pump speed was deliberately increased to 150% of its original value at the end of 90 *sec*, while keeping all remaining process variables at their desired settings. Due to the increased load for evaporation, the exit temperature of the air drops below its desired level. Figure 2.4 also illustrates how well the $AR(3)$ models generated under H_o perform in predicting the responses, despite the slurry pump speed disturbance. Good prediction is expected, since the AR model has a root at 0.99 for the exit temperature, acting as integrator. The residual Shewhart charts for level and spread obtained from H_0 (the AR(3) model) perform poorly. Residual CUSUM charts signal out-of-control status for level and spread (Figure 2.5). The level residual CUSUM (Figure 2.5a) first signals a positive deviation (false alarm).

The performance of the PCD method is displayed with Shewhart charts of parameters for the same disturbance (Figure 2.6) with solid lines describing the 95% control limits and the dashed lines describing the symmetric PCD scheme detection thresholds. The first AR parameter of the exit temperature model ($\hat{\phi}_1$) is diagnosed as changing in the positive direction by the PCD method at 111.5 *sec* (Figure 2.6, top left). The level residual CUSUM (Figure 2.5a) first detects an out-of-control status in the positive

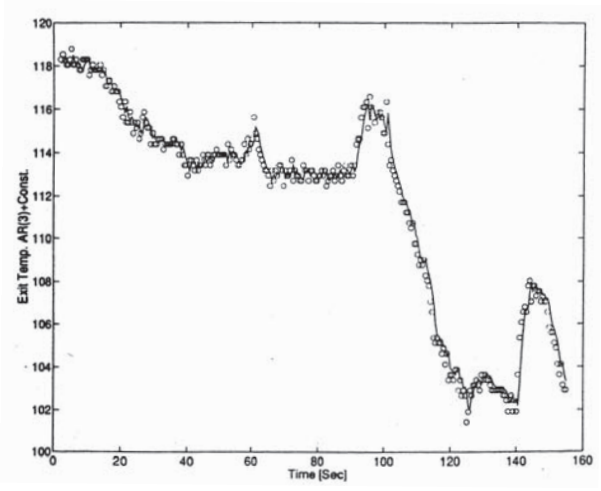


Figure 2.4. Process data (circles) and model predictions (solid line) for the exit air temperature from the spray dryer.

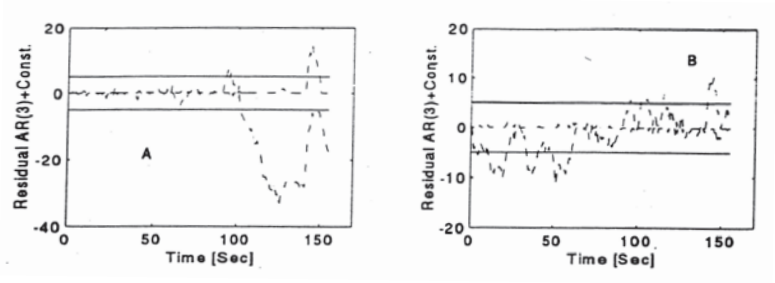


Figure 2.5. CUSUM monitoring charts of exit temperature residuals: (a) Level (mean), (b) Spread.

direction at 96 *sec* and then detects a negative shift at 102 *sec*. However, the behavior of the constant parameter in Figure 2.7 clearly indicates a bias shift in the negative direction.

To diagnose the kind of disturbance(s) experienced by the exit and inlet temperatures, the charts based on the *implicit levels* are depicted in Figure 2.7. The implicit level points calculated are shown by circles. While the level parameter remains essentially the same for the inlet temperature (not shown), the implicit level of the exit temperature changes drastically after 102 *sec*. As a result, only a structure change is detected for the inlet

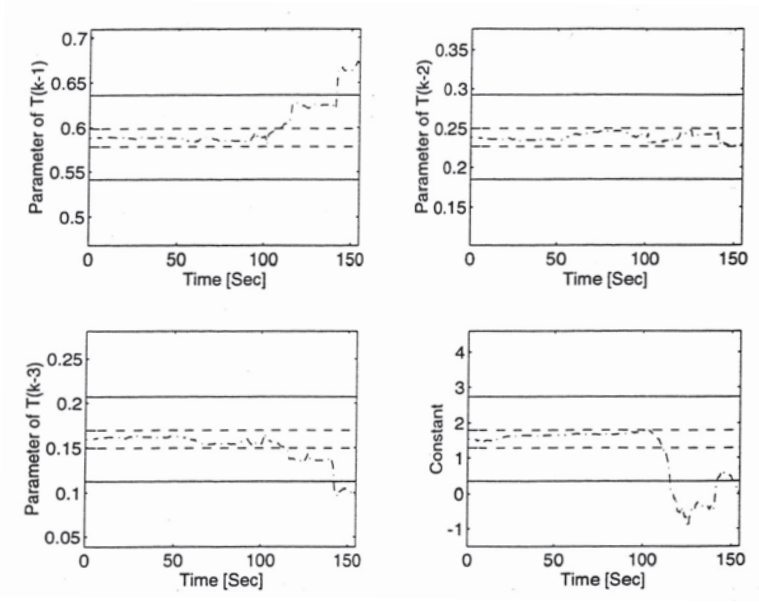


Figure 2.6. Shewhart charts for model parameters ϕ_1, ϕ_2, ϕ_3, and ϕ_4 (constant), respectively.

temperature, while changes in both level and structure are detected for the exit temperature.

2.4 Limitations of Univariate Monitoring Techniques

In the era of single-loop control systems in chemical processing plants, there was little infrastructure for monitoring multivariable processes by using multivariate statistical techniques. A limited number of process and quality variables were measured in most plants, and use of univariate SPM tools for monitoring critical process and quality variables seemed appropriate. The installation of computerized data acquisition and storage systems, the availability of inexpensive sensors for typical process variables such as temperature, flow rate, and pressure, and the development of advanced chemical analysis systems that can provide reliable information on quality variables at high frequencies increased the number of variables measured at

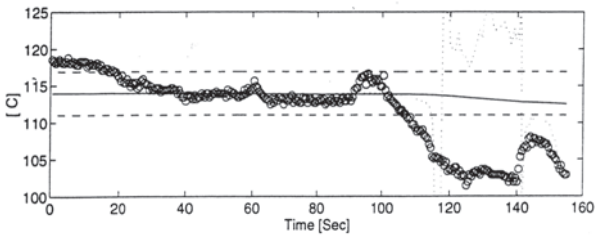


Figure 2.7. Diagnostic chart of dryer air exit temperature based on the implicit level parameter.

high frequencies and enabled a data-rich plant environment. This change accentuated the limitations of univariate SPM techniques for monitoring multivariable processes. The critical limitation is the exclusion of the correlation among various variables from the quantitative information provided by univariate SPM tools.

The outcome of this limitation is illustrated by monitoring a two-variable process (Figure 2.8). Shewhart charts of variables x_1 and x_2 are plotted along with the $x_1 - x_2$ biplot. The biplot shows on the x_1 versus x_2 plane the observed values of x_1 and x_2 for each sampling time. The sampling time stamps are not printed for simplifying the picture. Note the single data point marked by a circled cross. According to their Shewhart charts, both variables are in-control at all times. However, the biplot provides a different assessment. If one were to use the confidence limits of the Shewhart charts which form a rectangle that makes the borders of the biplot, the assessment is identical. But if it is assumed that the two variable process has a multivariate Normal distribution, then the confidence limits are represented by the ellipse that is mostly inside the rectangle of the biplot. However, most of the area inside the rectangle is outside the ellipse, and the ends of the ellipse extend beyond the corners of the rectangle. Based on the multivariate confidence limits, data points outside the ellipse are out-of-control. Hence, the data point marked by a circled cross indicates an out-of-control situation. In contrast, the portions of the ellipse outside the rectangle (upper left and lower right regions in the biplot) are in-control. While defective products (represented by the data point marked by a circled cross) would be shipped out as conforming to the specifications if univariate charts were used, good products with x_1, x_2 characteristics that are inside the ellipse but outside the rectangle would be discarded as defective.

The elliptical confidence region is generated by slicing the probability distribution 'bell' in Figure 2.9 by a plane parallel to the x_1, x_2 base plane

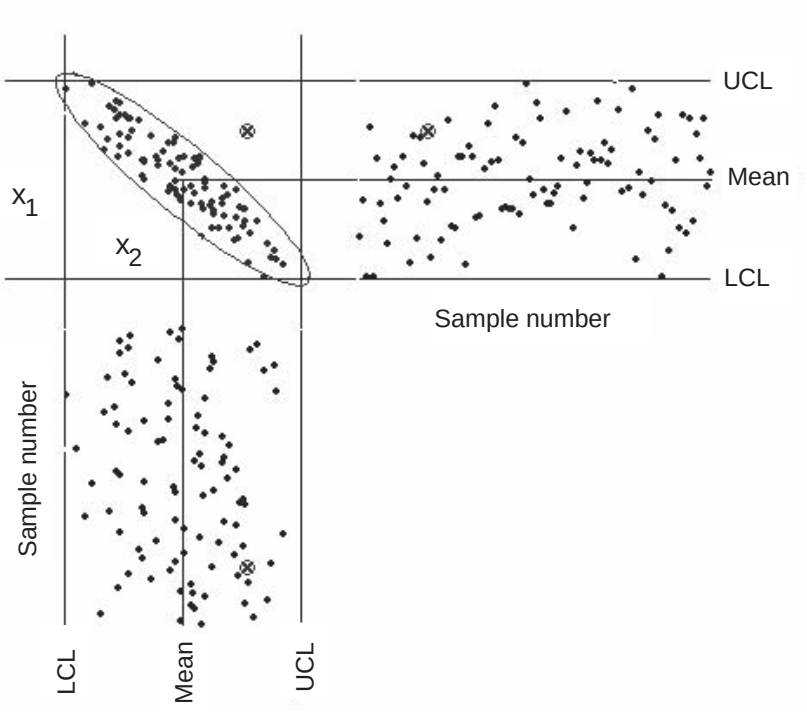


Figure 2.8. Monitoring of a two-variable process by two univariate Shewhart charts and a biplot of x_1 vs x_2.

of the figure. The probability distributions of x_1 or x_2 are the familiar 'bell-shaped curves' obtained by projecting the three-dimensional bell to the $f(x_1, x_2) - x_1$ or $f(x_1, x_2) - x_2$ vertical planes, respectively. Their confidence limits yield the familiar Shewhart chart limits. But, the slicing of the bell at a specific confidence level, given by the value of $f(x_1, x_2)$, yields an ellipse. The lengths of the major and minor axes of the ellipse are functions of the variances of x_1 and x_2, while their slopes are determined by the covariance of x_1 and x_2.

The shortcomings of using univariate charts for monitoring multivariable processes include too many false alarms, too many missed alarms and the difficulty of visualizing and interpreting 'the big picture' about the process status. Plant personnel are expected to form an opinion about the process status by integrating and interpretation from a large number of charts that ignore the correlation between the variables.

The appeal of multivariate process monitoring techniques is based on

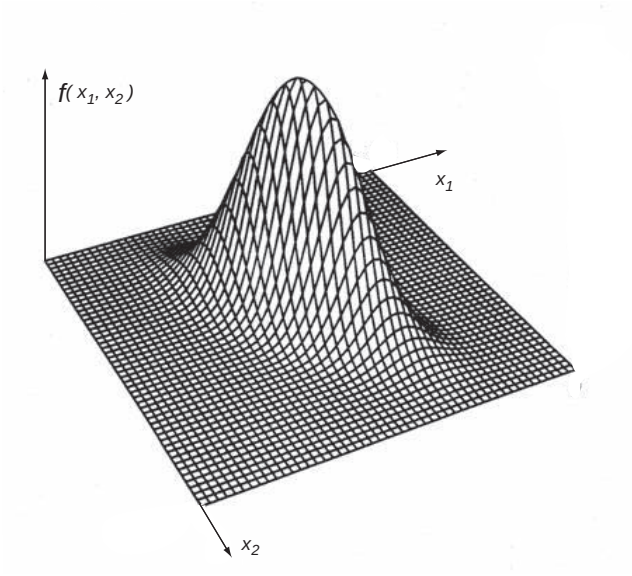


Figure 2.9. The plot of the probability distribution function of a two-variable (x_1, x_2) process.

their ability to capture the correlation information neglected by univariate monitoring techniques. Simple charts (no more complicated than Shewhart charts) can summarize the status of the process. While the mathematical and statistical techniques used are more complex, most multivariate process monitoring software shield these computations from the user and provide easy-to-interpret graphs for monitoring a process.

2.5 Summary

Various univariate statistical process monitoring techniques are discussed in this chapter. The philosophy and implementation of Shewhart charts are presented first. Then, cumulative sum (CUSUM) charts are introduced for monitoring processes with individual measurements and for detecting small changes in the mean. Moving average (MA) charts are presented and extended to exponentially weighted moving average (EWMA) charts that

attach more importance to recent data. Most chemical processes generate autocorrelated data. The impact of strong autocorrelation on univariate SPM techniques is reviewed and two SPM techniques for autocorrelated data are introduced. Finally, the limitations of univariate SPM techniques for monitoring multivariable processes are discussed. The statistical foundations for multivariate SPM techniques are introduced in Chapter 3, various empirical multivariable model development techniques are presented in Chapter 4, and the multivariable SPM methods for continuous processes are discussed in Chapter 5.

3

Multivariate Statistical Monitoring Techniques

Many process performance evaluation techniques are based on multivariate statistical methods. Various statistical methods that provide the foundations for model development, process monitoring and diagnosis are presented in this chapter. Section 3.1 introduces principal components analysis and partial least squares. Canonical variates analysis and independent components analysis are discussed in Sections 3.2 and 3.3. Contribution plots that indicate process variables that have made large contributions to significant changes in monitoring statistics are presented in Section 3.4. Statistical methods used for diagnosis of source causes of process abnormalities detected are introduced in Section 3.5. Nonlinear methods for monitoring and diagnosis are introduced in Section 3.6.

3.1 Principal Components Analysis

Principal Components Analysis (PCA) is a multivariable statistical technique that can extract the strong correlations of a data set through a set of empirical orthogonal functions. Its historic origins may be traced back to the works of Beltrami in Italy (1873) and Jordan in France (1874) who independently formulated the singular value decomposition (SVD) of a square matrix. However, the first practical application of PCA may be attributed to Pearson's work in biology [226] following which it became a standard multivariate statistical technique [3, 121, 126, 128].

PCA techniques can be used either as a detrending (filtering) tool for efficient data analysis and visualization or as a model-building structure to describe the expected variation under normal operation (NO). For a particular process, NO data set covers targeted operating conditions during satisfactory performance. PCA model is based on this representative

data set. The model can be used to detect outliers in data, provide data reconciliation and monitor deviations from NO that indicate excessive variation from normal target or unusual patterns of variation. Operation under various known upsets can also be modeled if sufficient historical data are available to develop automated diagnosis of source causes of abnormal process behavior [242].

Principal components (PC) are a new set of coordinate axes that are orthogonal to each other. The first PC indicates the direction of largest variation in data, the second PC indicates the largest variation unexplained by the first PC in a direction orthogonal to the first PC (Figure 3.1). The number of PCs is usually less than the number of measured variables.

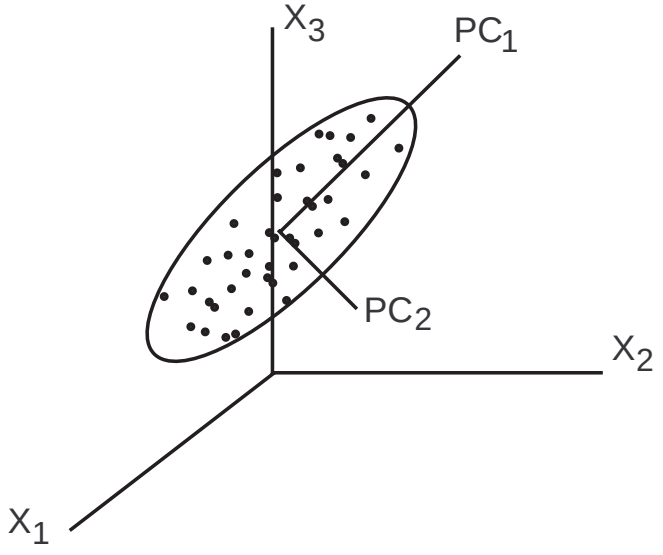


Figure 3.1. PCs of three-dimensional data set projected on a single plane. From [242], reproduced with permission. Copyright © 1996 AIChE.

PCA involves the orthogonal decomposition of the set of process measurements along the directions that explain the maximum variation in the data. For a continuous process, the elements of the $n \times m$ data matrix $\mathbf{X_D}$ are $x_{D,ij}$ where $i = 1, \cdots, n$ indicates the number of samples and $j = 1, \cdots, m$ indicates the number of variables. To remove magnitude and variance biases in data, $\mathbf{X_D}$ is mean-centered and variance-scaled to get $\mathbf{X}$. Each row of $\mathbf{X}$ represents the time series of a process measurement with mean 0 and variance 1 reflecting equal importance of each variable. If *a priori* knowledge about the relative importance about the variables is avail-

able, select variables can be given a slightly higher scaling weight than that corresponding to unit variance scaling [25, 94]. The directions extracted by the orthogonal decomposition of $\mathbf{X}$ are the eigenvectors $\mathbf{p}_i$ of $\mathbf{X}^T\mathbf{X}$ or the PC *loadings*

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \cdots + \mathbf{t}_a\mathbf{p}_a^T + \mathbf{E} \tag{3.1}$$

where $\mathbf{E}$ is $n \times m$ matrix of residuals. The dimension a is chosen such that most of the significant process information is taken out of $\mathbf{E}$, and $\mathbf{E}$ represents random error. If the directions are extracted sequentially, the first eigenvector is lined in the direction of maximum data variance and the second one, while being orthogonal to the first, is aligned in the direction of maximum variance of the residual, and so forth. The residual is obtained at each step by subtracting the variance already explained by the PC loadings already selected, and used as the 'data matrix' for the computation of the next PC loading.

The eigenvalues of the covariance matrix of $\mathbf{X}$ define the corresponding amount of variance explained by each eigenvector. The projection of the measurements (observations) onto the eigenvectors define new points in the measurement space. These points constitute the *score matrix*, $\mathbf{T}$ whose columns are $\mathbf{t_i}$ given in Eq. 3.1. The relationship between $\mathbf{T}, \mathbf{P}$, and $\mathbf{X}$ can also be expressed as

$$\mathbf{T} = \mathbf{XP}\ , \qquad \mathbf{X} = \mathbf{TP}^T + \mathbf{E} \tag{3.2}$$

where $\mathbf{P}$ is an $m \times a$ matrix whose jth column is the jth eigenvector of $\mathbf{X}^T\mathbf{X}$, and $\mathbf{T}$ is an $n \times a$ score matrix.

The PCs can be computed by spectral decomposition [126], computation of eigenvalues and eigenvectors, or singular value decomposition. The covariance matrix $\mathbf{S}$ ($\mathbf{S}$=$\mathbf{X}^T\mathbf{X}/(m-1)$) of data matrix $\mathbf{X}$ can be decomposed by *spectral decomposition* as

$$\mathbf{S} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \tag{3.3}$$

where $\mathbf{P}$ is a unitary matrix[1] whose columns are the normalized eigenvectors of $\mathbf{S}$ and $\mathbf{\Lambda}$ is a diagonal matrix that contains the ordered eigenvalues λ_i of $\mathbf{S}$. The scores $\mathbf{T}$ are computed by using the relation $\mathbf{T} = \mathbf{XP}$.

Singular value decomposition of the data matrix $\mathbf{X}$ is given as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3.4}$$

where the columns of $\mathbf{U}$ are the normalized eigenvectors of $\mathbf{X}\mathbf{X}^T$, the columns of $\mathbf{V}$ are the normalized eigenvectors of $\mathbf{X}^T\mathbf{X}$, and $\mathbf{\Sigma}$ is a 'diagonal' matrix having as its elements the singular values, or the positive

[1] A unitary matrix $\mathbf{A}$ is a complex matrix in which the inverse is equal to the conjugate of the transpose: $\mathbf{A}^{-1} = \mathbf{A}^*$. Orthogonal matrices are unitary. If $\mathbf{A}$ is a *real* unitary matrix then $\mathbf{A}^{-1} = \mathbf{A}^T$.

square roots of the magnitude ordered eigenvalues of $\mathbf{X}^T\mathbf{X}$. For an $n \times m$ matrix $\mathbf{X}$, $\mathbf{U}$ is $n \times n$, $\mathbf{V}$ is $m \times m$ and $\mathbf{\Sigma}$ is $n \times m$. Let the rank of $\mathbf{X}$ be denoted as r, $r \leq \min(m, n)$. The first r rows of $\mathbf{\Sigma}$ make a $r \times r$ diagonal matrix, the remaining $n - r$ rows are filled with zeros. Term by term comparison of the second equation in Eq. 3.2 and Eq. 3.4 yields

$$\mathbf{P} = \mathbf{V} \qquad \text{and} \qquad \mathbf{T} = \mathbf{U}\mathbf{\Sigma} \; . \tag{3.5}$$

For a data set that is described well by two PCs, the data can be displayed in a plane. The data are scattered as an ellipse whose axes are in the direction of PC loadings in Figure 3.1. For higher number of variables data will be scattered as an ellipsoid.

The selection of appropriate number of PCs or the maximum significant dimension a is critical for developing a parsimonious PCA model [120, 126, 258]. A quick method for computing an approximate value for a is to add PCs to the model until the percent of the cumulative variation explained by including additional PCs becomes small. The percent cumulative variation is given as

$$\%\ Cumulative\ Variance = \frac{\sum_{i=1}^{a} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \tag{3.6}$$

A more precise method that requires more computational time is cross-validation [155, 332]. It is implemented by excluding part of the data, performing PCA on the remaining data, and computing the prediction error sum of squares (PRESS) using the data retained (excluded from model development). The process is repeated until every observation is left out once. The order a is selected as that minimizes the overall PRESS. Two additional criteria for choosing the optimal number of PCs have also been proposed by Wold [332] and Krzanowski [155], related to cross-validation. Wold [332] proposed checking the ratio,

$$R = \frac{PRESS_a}{RSS_{a-1}} \tag{3.7}$$

where RSS_a is the residual sum of squares based on the PCA model after adding the ath principal component. When R exceeds unity upon addition of another PC, it suggests that the ath component did not improve the prediction power of the model and it is better to use $a - 1$ components. Another approach is based on the SCREE plots that indicate the dimension at which the smooth decrease in the magnitude of the covariance matrix eigenvalues appear to level off to the right of the plot [253].

PCA is simply an algebraic method of transforming the coordinate system of a data set for more efficient description of variability. The convenience of this representation is in the equivalence of data to measurable

and meaningful physical quantities like temperatures, pressures and compositions. In statistical analysis and modeling, the quantification of data variance is of great importance. PCA provides a direct method of orthogonal decomposition onto a new set of basis vectors that are aligned with the directions of maximum data variance.

The empirical formulations proposed for the automated selection of a usually give good results in finding a that captures the dominant correlations or variance in the data set with minimum number of PCs. But this is essentially a practical matter dependent on the particular problem and the appropriate balance between parsimony and information detail. One approach is demonstrated in the following example.

Example Let $m = 20$ and $n = 1000$ and generate $\mathbf{X}_{D1}$ by Gaussian random assignment of simulated data. Let $\mathbf{X}_1$ be the corresponding mean-centered and variance-scaled data set, which is essentially free of any structured correlation among the variables. PCA analysis of $\mathbf{X}_1$ identifies the orthogonal eigenvectors $\mathbf{U}_1$ and the associated eigenvalues $\{\lambda_1, ..., \lambda_{20}\}$ while separating the marginally different variance contributions along each PC. For this case $a = 0$ and the complete data representation is basically the random contributions, $\mathbf{X}_1 = \mathbf{X}_{1R} = \mathbf{E}$. Now generate a new set of data $\mathbf{X}_{D2}$ by a combination of $\mathbf{X}_{D1}$ and time-variant multiple (five) correlated functions within $m = 20$. $\mathbf{X}_2$ is the corresponding mean-centered and variance-scaled version of $\mathbf{X}_{D2}$. Note that mean-centering along the rows of $\mathbf{X}_{D2}$ removes any possibility of retaining a static correlation structure in $\mathbf{X}_2$. Thus, $\mathbf{X}_2$ has only random components and time dependent correlated variabilities contributing towards the overall variance of the data set. Figure 3.2 shows the comparison of two cases in terms of both eigenvalues and variance characteristics associated with sequential PCs. Eigenvalues are presented in a scaled form as λ_i/λ_1 and the variance contributions are plotted as fractional cumulative values as in Eq. 3.6. Random nature of $\mathbf{X}_1$ is evident in the similarity of eigenvalue magnitudes where each subsequent value is only marginally smaller than the previous one. As a result, contributions to overall variance with additional modes essentially form a linear trend confirming similarity in variabilities explained through each PC. On the other hand, the parts of the plots showing the characteristics of $\mathbf{X}_2$ reflect the distinct difference between the first three eigenvalues compared to the rest. The scaled eigenvalue plot shows that the asymptotic trend (slope) of the initial higher values when compared to the smaller values differentiate the first three eigenvalues from the rest suggesting that $a \approx 3$. With $a = 3$, almost 70% of the total variance can be captured. Note that starting with $a + 1$, the relative contributions of additional PCs can not be clearly differentiated from the contributions of higher orders. For some practical cases, the distinction between dominant and random modes may

not be as clear as this example demonstrates. However, combined with specific process knowledge, the two plots presented here are always useful in selecting the appropriate a.

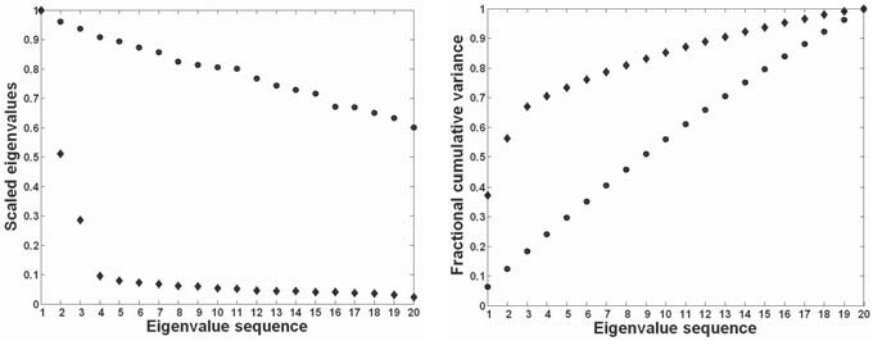


Figure 3.2. Scaled eigenvalues (left) and cumulative contributions of sequential PCs towards total variance for two simulated data sets. First data set has only normally distributed random numbers (circles) while the second one has time dependent correlated variables in addition to random noise (diamonds).

Partial Least Squares

Partial Least Squares (PLS) projections to latent structures, develops a biased model between two blocks of variables $\mathbf{X}$ and $\mathbf{Y}$. PLS selects latent variables so that variation in $\mathbf{X}$ which is most predictive of the $\mathbf{Y}$ is extracted. The PLS approach was developed in the 1970s by H. Wold for analyzing social sciences data by estimating model parameters using the Nonlinear Iterative Partial Squares (NIPALS). The method was further developed in the 1980s by S. Wold and H. Martens for more complex data structures in science and technology applications. PLS works on the sample covariance matrix $(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{X})$ [338]. The original PLS methodology provides a linear multivariate model. The modeling algorithm is described in Section 4.3. Nonlinear extensions can be developed by using variable transformations in the $\mathbf{X}$ and/or $\mathbf{Y}$ blocks if the nonlinearity is within these blocks or by using a nonlinear functional form in the so-called inner relation if the nonlinearity is between the $\mathbf{X}$ block and the $\mathbf{Y}$ block [61].

3.2 Canonical Variates Analysis

Canonical correlation analysis identifies and quantifies the associations between two sets of variables [126]. Canonical correlation analysis is conducted by using canonical variates. Consider n observations of two random vectors $\mathbf{x}$ and $\mathbf{y}$ of dimensions p and q forming data sets $\mathbf{X}_{p\times n}$ and $\mathbf{Y}_{q\times n}$ with $Cov(\mathbf{X}) = \mathbf{\Sigma}_{11}$, $Cov(\mathbf{X}) = \mathbf{\Sigma}_{22}$, and $Cov(\mathbf{X}, \mathbf{Y}) = \mathbf{\Sigma}_{12}$. Also $\mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21}^T$ and without loss of generality $p \leq q$.

For coefficient vectors $\mathbf{a}$ and $\mathbf{b}$ form the linear combinations $\mathbf{u} = \mathbf{a}^T\mathbf{X}$ and $\mathbf{v} = \mathbf{b}^T\mathbf{Y}$. Then, for the first pair $u_1,\ v_1$ the the maximum correlation

$$\max_{\mathbf{a},\mathbf{b}}\ Corr(u_1, v_1) = \rho_1 \tag{3.8}$$

is attained by the the linear combination (first canonical pair)

$$u_1 = \mathbf{e}_1^T\mathbf{\Sigma}_{11}^{-1/2}\mathbf{X} \qquad v_1 = \mathbf{f}_1^T\mathbf{\Sigma}_{22}^{-1/2}\mathbf{Y} \tag{3.9}$$

The kth pair of canonical variates $k = 2,\ 3,\ \cdots\ p$,

$$u_k = \mathbf{e}_k^T\mathbf{\Sigma}_{11}^{-1/2}\mathbf{X} \qquad v_k = \mathbf{f}_k^T\mathbf{\Sigma}_{22}^{-1/2}\mathbf{Y} \tag{3.10}$$

maximizes $Corr\ (u_k,\ v_k) = \rho_k$ among those linear combinations uncorrelated with the preceding $k-1$ canonical variables. Here $\rho_1^2,\ \rho_2^2,\ \cdots \rho_p^2$ are the eigenvalues of covariances $\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2}$ and $\mathbf{e}_1,\ \mathbf{e}_2,\ \cdots \mathbf{e}_p$ are the associated $p \times 1$ eigenvectors. ρ_i^2 are also the eigenvalues of covariances $\mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}$ with the corresponding $q \times 1$ eigenvectors $\mathbf{f}_1,\ \mathbf{f}_2,\ \cdots \mathbf{f}_p$. Detailed discussion of canonical variates and canonical correlations analysis are provided in most multivariate statistical analysis books [126].

Canonical variates will be used in the formulation of subspace state–space models in Section 4.5.

3.3 Independent Component Analysis

Independent Component Analysis (ICA) is a signal processing method for transforming multivariate data into statistically independent components expressed as linear combinations of observed variables [91, 119, 134]. Consider a process with m zero-mean variables $\mathbf{x} = (x_1\ \ x_2\ \cdots\ x_m)^T$. The zero-mean independent variables $\mathbf{s} = (s_1\ \ s_2\ \cdots\ s_l)^T$are defined by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.11}$$

where $\mathbf{A}$ is the mixing matrix of dimension $m \times l$ that will be determined. For n samples, Eq. 3.11 becomes

$$\mathbf{X} = \mathbf{AS} \tag{3.12}$$

where the dimensions of $\mathbf{X}$ and $\mathbf{S}$ are $m \times n$ and $l \times n$, respectively. The mathematical problem to solve is the estimation of $\mathbf{S}$ and $\mathbf{A}$ from $\mathbf{X}$. A separating matrix $\mathbf{W}_{l \times m}$ is calculated to achieve this so that the components of the reconstructed data matrix $\mathbf{Y} = \mathbf{WX}$ become as independent as possible from each other. The limitations of ICA are:

1. The signs, powers and orders of independent components (IC) can not be estimated.
2. Only non-Gaussian ICs can be estimated, only one of them can be Gaussian.

These limitations have little impact for their use in process monitoring because the estimations in limitation (1) is crucial only when an exact reconstruction of ICs is necessary and if the original signals are Gaussian, arbitrarily selecting one of the ICs as Gaussian yields ICs that are useful for monitoring [138].

To perform ICA, measured variables x_i are first transformed to uncorrelated, unit-variance variables z_j called *sphering* or prewhitening. This can be implemented by PCA. The relationship between $\mathbf{z}$ and $\mathbf{s}$ is expressed as

$$\mathbf{z} = \mathbf{Mx} = \mathbf{MAs} = \mathbf{Bs} \tag{3.13}$$

where $\mathbf{M}$ is the sphering matrix and $\mathbf{B} = \mathbf{MA}$. Since s_i are mutually independent and and z_j are mutually uncorrelated

$$E[\mathbf{zz}^T] = \mathbf{B}E[\mathbf{SS}^T]\mathbf{B}^T = \mathbf{BB}^T = \mathbf{I} \tag{3.14}$$

if the covariance of $\mathbf{s}$, $E[\mathbf{ss}^T]$, is an identity matrix. Hence, $\mathbf{B}$ is an orthogonal matrix according to Eq. 3.14. Since $\mathbf{M}$ is determined by PCA, estimation of $\mathbf{A}$ is reduced to the estimation of the orthogonal matrix $\mathbf{B}$.

Kurtosis or the fourth-order cumulant is used in computing $\mathbf{B}$. The fourth-order cumulant $\kappa_4(u)$ of a zero-mean variable u is

$$\kappa_4(u) = E[u^4] - 3E[y^2] \tag{3.15}$$

The columns of $\mathbf{B}$ are obtained by minimizing or maximizing $\kappa_4(\mathbf{b}^T\mathbf{z})$ under the constraint $\|\mathbf{b}\| = 1$ by using a gradient method [51, 138]. A learning algorithm based on the gradient method has the form

$$\mathbf{b}(l+1) = \mathbf{b}(l) \pm \mu \left\{ E\left[4 \left(\mathbf{b}(l)^T \mathbf{z} \right)^3 \mathbf{z} \right] - 12 \|\mathbf{b}(l)\|^2 \mathbf{b}(l) + 2\lambda \mathbf{b}(l) \right\} \tag{3.16}$$

where μ denotes a learning-rate parameter, λ a Lagrangian multiplier, and l an iteration index. A fixed-point algorithm can be used instead of a learning algorithm for finding the local extrema of the fourth-order cumulant [138]. The fixed-points $\mathbf{b}$ of Equation 3.16 satisfy

$$E\left[4\left(\mathbf{b}^T\mathbf{z}\right)^3\mathbf{z}\right] - 12\|\mathbf{b}\|^2\mathbf{b} + 2\lambda\mathbf{b} = 0 \tag{3.17}$$

and are obtained by iteration:

$$\mathbf{b}(l+1) = \lambda'\left\{E\left[\left(\mathbf{b}(l)^T\mathbf{z}\right)^3\mathbf{z}\right] - 3\|\mathbf{b}(l)\|^2\mathbf{b}(l)\right\} \tag{3.18}$$

The fixed-point algorithm for ICA is summarized by Kano *et al.* [138]:

1. Transform measured variables $\mathbf{x}$ to unit-variance uncorrelated variables $\mathbf{z}$ using Eq. 3.13. PCA can accomplish this transformation.

2. Start with a random initial vector $\mathbf{b}_i(0)$ of unit norm $\|\mathbf{b}\| = 1$. For $i \geq 2$, $\mathbf{b}_i(0)$ is projected using

$$\mathbf{b}_i(0) = \mathbf{b}_i(0) - \mathbf{B}_{i-1}\mathbf{B}_{i-1}^T\mathbf{b}_i(0) \tag{3.19}$$

   and then it is normalized so that $\|\mathbf{b}_i(0)\| = 1$.

   Start with $l = 0$.

   (a) $\mathbf{b}_i$ is updated using

$$\mathbf{b}_i(l+1) = E\left[\left(\mathbf{b}_i(l)^T\mathbf{z}\right)^3\mathbf{z}\right] - 3\mathbf{b}_i(l) \tag{3.20}$$

   The expected value is estimated by using a large number of samples.

   (b) $\mathbf{b}_i(l+1)$ is projected using

$$\mathbf{b}_i(l+1) = \mathbf{b}_i(l+1) - \mathbf{B}_{i-1}\mathbf{B}_{i-1}^T\mathbf{b}_i(l+1) \tag{3.21}$$

   and normalized so that $\|\mathbf{b}_i(l+1)\| = 1$.

   (c) If $|\mathbf{b}_i(l+1)^T\mathbf{b}_i(l)|$ is close enough to 1 go to the next step, otherwise let $l = l + 1$ and go back to Step (a).

3. Let $\mathbf{b}_i = \mathbf{b}_i(l+1)$, $i = i + 1$ and go back to Step 2. This iteration ends when $i = l$.

4. The independent components $\mathbf{Y}$ are obtained from

$$\mathbf{Y} = \mathbf{B}^T\mathbf{Z} = \mathbf{B}^T\mathbf{M}\mathbf{X} \tag{3.22}$$

   where $\mathbf{B} = \mathbf{B}_l$.

3.4 Contribution Plots

Multivariate process monitoring techniques use measurements of process variables to detect significant deviations in process operation from the desired or normal operation (NO) and trigger the need to determine special causes affecting the process. Multivariate monitoring charts such as T^2 and SPE charts (Section 5.1) indicate when the process goes out of control, but they do not provide information on the source causes of abnormal process operation. The engineers and plant operators need to determine the actual problem once an out-of-control situation is indicated. Miller *et al.* [197, 198] have introduced variable contributions and contribution plots concept to address this need. Contribution plots indicate the process variables that have contributed significantly to inflate T^2-statistic (or D), squared prediction error SPE-statistic (or Q) and scores. The fault diagnosis activity is completed by using process knowledge (of plant personnel or a knowledge-based system) to relate these process variables to various equipment failures and disturbances.

Contributions of process variables to the T^2-statistic.

Two different approaches for calculating variable contributions to T^2-statistic have been proposed. The first approach introduced by Miller *et al.* [198] and by MacGregor *et al.* [146, 177] calculates the contribution of each process variable to a separate score. T^2 can be written as

$$T^2 = \sum_{i=1}^{m} \frac{t_i^2}{\lambda_i} = \sum_{i=1}^{m} \frac{t_i^2}{s_i^2} \tag{3.23}$$

where, as before, t_i denotes the scores, λ_i the eigenvalues of $\mathbf{S}$, m the number of variables, and s_i^2 the variance of t_i (the ith ordered eigenvalue of $\mathbf{S}$). Each score can be written as

$$t_i = \mathbf{p}_i^T(\mathbf{x} - \bar{\mathbf{x}}) = \sum_{j=1}^{m} p_{i,j}(x_j - \bar{x}_j) \tag{3.24}$$

where $\mathbf{p}_i$ is the loading, the eigenvector of $\mathbf{S}$ corresponding to λ_i, and $p_{i,j}$, x_j, and $\bar{x}_j$ are associated with the jth variable. The contribution of each variable x_j to the score of PC i is given by Eq. 3.24

$$p_{i,j}(x_j - \bar{x}_j) \tag{3.25}$$

Considering that variables with high levels of contribution that are of the same sign as the score are responsible for driving T^2 to higher values, only those variables are included in the analysis [146]. For example, only variables with negative contributions are selected if the score is negative.

The overall contribution of each variable is computed by summing over all scores with high values. For each score with high values (using a threshold value of 2.5, for example) the variable contributions are calculated [146]. Then, the values over all the l high scores are summed for contributions that have the same sign as the score:

1. For all l high scores ($l \leq m$):

   i. Compute the contribution of variable x_j to the normalized score $(t_i/S_i)^2$

$$cont_{i,j}^{T^2} = \frac{t_i^2}{s_i^2} p_{i,j}(x_j - \bar{x}_j) = \frac{t_i^2}{\lambda_i} p_{i,j}(x_j - \bar{x}_j) \tag{3.26}$$

   ii. Set $cont_{i,j}^{T^2}$ to zero if it is negative (sign opposite to the score t_i)

2. Calculate the total contribution of variable x_j

$$CONT_j^{T^2} = \sum_{i=1}^{l} (cont_{i,j}^{D}) \tag{3.27}$$

The second approach was proposed by Nomikos [217] and implemented on batch process data. This approach calculates contributions of each process variable to the T^2-statistic rather than contributions of separate scores.

$$CONT_j^{T^2} = \sum_{i=1}^{m} \frac{t_i^2}{s_i^2} p_{i,j}(x_j - \bar{x}_j) \tag{3.28}$$

Contributions of process variables to the SPE-statistic.

Contribution to the SPE-statistic is calculated using the individual residuals. The contribution of variable j to the SPE at time k is

$$CONT_j^{SPE}(k) = (x_j(k) - \hat{x}_j(k))^2 \tag{3.29}$$

For a data set of length n:

$$CONT_j^{SPE} = (\mathbf{x}_j - \hat{\mathbf{x}}_j)(\mathbf{x}_j - \hat{\mathbf{x}}_j)^T = \sum_{i=1}^{n} (e_{i,j})^2 \tag{3.30}$$

where $\hat{\mathbf{x}}_j$ is the vector of predicted values of the (centered and scaled) measured variable j (with n observations) and e_j denotes the residuals.

It is always a good practice to check individual process variable plots for those variables diagnosed as responsible for flagging an out-of-control

situation. When the number of variables is large, analyzing contribution plots and corresponding variable plots to reason about the faulty condition may become tedious and challenging. This analysis can be automated and linked with real-time diagnosis [219, 304] by using knowledge-based systems.

3.5 Linear Methods for Diagnosis

Fault diagnosis determines the source cause(s) of abnormal process operation. The fault may be one of many that are already known because of previous experience or a new one. Fault diagnosis activity usually compares the performance of the process (trajectories of process variables) under the current fault to process behavior under various faults (fault signatures) to determine the current fault. A combination of statistical techniques and process knowledge should first be used to catalog process behaviors (fault signatures) from historical data. Pattern-matching methods for this activity have been proposed [270, 271, 273]. It is important to consider the effects of data compression methods used for storing historical data when such data are used for pattern matching and cataloging of faults [272]. The identification of fault signatures for faults that have not been determined by plant personnel may necessitate unsupervised learning. This can be achieved by clustering (Section 3.5.1). Once data clusters with various faults have been determined, discrimination and classification are used for fault diagnosis [63, 79]. Two linear statistical techniques, discriminant analysis (Section 3.5.2) and Fisher's discriminant analysis (Section 3.5.3), are introduced to illustrate the strengths and limitations of these techniques. Neural networks have also been used for fault classification and diagnosis [252, 311, 312]. NN-based classification is useful when a small number of faults in a closed set are to be diagnosed, but for more complex cases with multiple faults or new faults NN do not provide a reliable framework and they may converge to local optima during training. Support vector machines (SVM) provide another nonlinear technique for event classification and fault diagnosis (Section 3.6.3).

3.5.1 Clustering

Searching the data for groupings (classes) according to some characteristics is an important exploratory process. Cluster analysis performs grouping (classification) on the bases of similarity measures [126]. Items and cases are usually clustered by indicating proximity using some measure of distance or angle. Variables are usually grouped on the basis of measures of association such as correlation coefficients.

The distance $d(\mathbf{x}, \mathbf{y})$ between two items $\mathbf{x} = [x_1\ x_2\ \cdots\ x_m]^T$ and $\mathbf{y} = [y_1\ y_2\ \cdots\ y_m]^T$ can be expressed as the *Euclidian distance*,

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})} \tag{3.31}$$

or the *statistical distance* (or Mahalanobis distance),

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T\mathbf{S}^{-1}(\mathbf{x} - \mathbf{y})} \tag{3.32}$$

where $\mathbf{S}$ is the covariance matrix, or the *Minkowski metric*,

$$d(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^{m} |x_i - y_i|^p\right]^{1/p} \tag{3.33}$$

Other distance measures include the Canberra metric and the Czekanowski coefficient [126]. Clustering can be hierarchical such as grouping of species and subspecies in biology or nonhierarchical such as grouping of items. For fault diagnosis nonhierarchical clustering is used to group data to k clusters corresponding to k known faults.

k-means clustering is a popular nonhierarchical clustering method that assigns each item to the cluster having the nearest centroid (mean). It was proposed by MacQueen [178], and consists of

1. Partitioning the items into k initial clusters or specifying k initial mean values as seed points.

2. Proceeding through the list of items by assigning an item to the cluster whose mean is nearest (using a distance measure, usually the Euclidian distance)

3. Recalculation of the mean for the cluster receiving the new new item and the cluster losing the item.

4. Repeating Steps 2 and 3 until no more reassignments take place.

The traditional hierarchical and nonhierarchical (e.g., k-means) clustering algorithms [69] have a number of drawbacks that require caution in their implementation for time series data. The hierarchical clustering algorithms assume an implicit parent-child relationship between the members of a cluster which may not be relevant for time series data. However, they can provide good initial estimates of patterns that may exist in the data set. The k-means algorithm requires the estimate of the number of clusters (i.e., k) and its solution depends on the initial assignments as the optimization

can get stuck in local minima. Furthermore, time series data are inherently autocorrelated that violates the key assumption of independent data elements for traditional clustering algorithms. Beaver and Palazoglu [14] [16] proposed an agglomerative k-means algorithm that overcomes these drawbacks and can also present the results in terms of a dendrogram, thus facilitating the selection of final cluster solution depending on the desired level of resolution. The algorithm is referred to as k-PCA Models as it uses dynamic PCA as the prototype model for time series data. It is applied to data collected from the operation of a pilot-plant that exhibits cyclic dynamic response [15] and shows how the periods of faulty and normal operations can be distinguished from one another.

Displaying multivariate data in low-dimensional space can be useful for visual clustering of items. For example, plotting the scores of the first few pairs of principal components as biplots of the first versus the second or the third principal components can cluster normal process operation and operation under various faults. Examples of biplots and their interpretation for fault diagnosis are presented in Chapter 7.

Pattern-matching methods to catalog process behaviors (fault signatures) from historical data have been proposed [271, 270, 273]. For high-dimensional data, distance measures may not be enough to describe the locations of specific clusters with respect to one another. Angle measures provide additional information [243, 154].

3.5.2 Discriminant Analysis

Statistical discrimination and classification *separate* distinct sets of objects (or events), and *allocate* new objects (or events) into previously defined groups of objects, respectively [126]. *Discrimination* uses discrimination criteria called *discriminants* for converting salient features of objects from several known populations to quantitative information separating these populations as much as possible. *Classification* sorts new objects or events into previously labeled classes by using rules derived to optimally assign new objects to the labelled classes. A good classification procedure should yield few misclassifications. The probability of occurrence of an event may be greater if it belongs to a population that has a greater likelihood of occurrence. A good classification rule should take these 'prior probabilities of occurrence' into consideration and account for the costs associated with misclassification.

Consider a data set with g distinct events such as normal process operation and operation under $g-1$ different faults. The operation type (class) is determined on the basis of m measured variables $\mathbf{x} = [x_1\ x_2\ \cdots\ x_m]^T$ that are random variables. Denote the classes by $\pi_i,\ i = 1, \cdots, g$, their

prior probability by p_i $i = 1, \cdots, g$ and their probability density functions by $f_i(\mathbf{x})$. Assume that $f_i(\mathbf{x})$ are multivariate Normal density functions with population and sample means $\boldsymbol{\mu}_i$ and $\bar{\mathbf{x}}_i$, respectively and population and sample variances $\boldsymbol{\Sigma}_i$ and $\mathbf{S}_i$, respectively. The *cost of misclassification* is $c(k|i)$, the cost of allocating an object to π_k (for $k = 1, \cdots, g$) when it belongs to π_i (for $i = 1, \cdots, g$). If R_k is the set of $\mathbf{x}$'s classified as π_k, the probability of classifying an event as π_k when it actually belongs to π_i is

$$\begin{aligned} P(k|i) &= P(\text{classifying event as } \pi_k|\pi_i) \\ &= \int_{R_k} f_i(\mathbf{x})d\mathbf{x} \quad i, k = 1, \cdots, g \end{aligned} \tag{3.34}$$

with $P(i|i) = 1 - \sum_{k=1, k\neq i}^{g} P(k|i)$. The notation $P(a|b)$ indicates the conditional probability of observing a, premised on the presence of b. The conditional *expected cost of misclassification* (ECM) of an event in π_1 to any other class is

$$ECM(\pi_1) = \sum_{k=2}^{g} P(k|1)c(k|1) \tag{3.35}$$

This conditional expected cost of misclassifying an event belonging to π_1 occurs with prior probability p_1 (the probability of π_1). The conditional *overall* expected cost of misclassification is computed by multiplying each $ECM(\pi_1)$ with its prior probability and summing over all classes

$$\begin{aligned} ECM &= p_1 ECM(\pi_1) + \cdots + p_g ECM(\pi_g) \\ &= p_1 \sum_{k=2}^{g} P(k|1)c(k|1) + p_2 \sum_{k=1,\ k\neq 2}^{g} P(k|2)c(k|2) \\ &\quad + \cdots + p_g \sum_{k=1}^{g-1} P(k|g)c(k|g) \\ &= \sum_{i=1}^{g} p_i \left(\sum_{k=1,\ k\neq i}^{g} P(k|i)c(k|i) \right) \end{aligned} \tag{3.36}$$

The determination of the optimal classification procedure becomes selection of mutually exclusive and exhaustive classification regions $R_1, R_2, \cdots, R_g$ such that the ECM in Eq. 3.36 is minimized [126]. The classification regions that minimize Eq. 3.36 are defined by allocating $\mathbf{x}$ to that population $\pi_k,\ k = 1, \cdots, g$ for which

$$\sum_{i=1,\ i\neq k}^{g} p_i f_i(\mathbf{x})c(k|i) \tag{3.37}$$

is smallest [3, 126]. If all misclassification costs are equal, the event indicated by data $\mathbf{x}$ will be assigned to that population π_k for which the sum $\sum_{i=1,\ i\neq k}^{g} p_i f_i(\mathbf{x})$ is smallest. Hence, the omitted term $p_k f_k(\mathbf{x})$ is largest, and the minimum ECM rule for equal misclassification costs becomes [126]:

$$\text{Allocate } \mathbf{x} \text{ to } \pi_k \text{ if } p_k f_k(\mathbf{x}) > p_i f_i(\mathbf{x}) \text{ for all } i \neq k.$$

given prior probabilities, density functions, and misclassification costs (when they are not equal). This classification rule is identical to the rule that maximizes the 'posterior' probability $P(\pi_k|\mathbf{x})$ ($P(\mathbf{x})$ comes from π_k given that $\mathbf{x}$ was observed) where

$$P(\pi_k|\mathbf{x}) = \frac{p_k f_k(\mathbf{x})}{\sum_{i=1}^{g} p_i f_i(\mathbf{x})} = \frac{\text{(prior)} \ \times \ \text{(likelihood)}}{\sum[\text{(prior)} \ \times \ \text{(likelihood)}]} \tag{3.38}$$

with $k = 1, \cdots, g$. If the populations follow Normal distributions with mean vectors $\boldsymbol{\mu}_i$, covariance matrices $\boldsymbol{\Sigma}_i$, and *generalized variance* $|\boldsymbol{\Sigma}_i|$ (determinant of the covariance), $f_i(\mathbf{x})$ is defined as

$$f_i(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right] \tag{3.39}$$

for $i = 1, \cdots, g$ and all misclassification costs are equal, then $\mathbf{x}$ is allocated to π_k if

$$\begin{aligned} \ln p_k f_k(\mathbf{x}) &= \ln p_k - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \\ &= \max_i \ \ln \ p_i F_i(\mathbf{x}) \end{aligned} \tag{3.40}$$

The constant $p/2 \ \ln(2\pi)$ is the same for all populations and can be ignored in discriminant analysis. The *quadratic discrimination score* for the ith population ${d_i}^Q(\mathbf{x})$ is defined as [126]

$${d_i}^Q(\mathbf{x}) = \ln p_i - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \quad i = 1, \cdots, g \tag{3.41}$$

The generalized variance $|\boldsymbol{\Sigma}_i|$, the prior probability p_i and the Mahalanobis distance contribute to the quadratic score ${d_i}^Q(\mathbf{x})$. Using the discriminant scores, the minimum total probability of misclassification rule for Normal populations and unequal covariance matrices becomes [126]:

$$\text{Allocate } \mathbf{x} \text{ to } \pi_k \text{ if } {d_k}^Q(\mathbf{x}) \text{ is the } \textit{largest} \text{ of all } {d_i}^Q(\mathbf{x}),\ i = 1, \cdots, g.$$

In practice, population mean and covariances ($\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$) are unknown. Computations are based on historical data sets of classified observations, and sample mean ($\bar{\mathbf{x}}_i$) and covariance matrices ($\mathbf{S}_i$) are used in Eq. 3.41.

A simplification is possible if the population covariance matrices $\mathbf{\Sigma}_i$ are equal for all i. Then, $\mathbf{\Sigma}_i = \mathbf{\Sigma}$ and Eq. 3.41 reduces to

$$d_i{}^Q(\mathbf{x}) = \ln p_i - \frac{1}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}(\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}) + \boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i \quad (3.42)$$

Since the second and third terms are independent of i, they are the same for all $d_i{}^Q(\mathbf{x})$ and can be ignored in classification. Since the remaining terms consist of a constant for each i ($\ln p_i - 1/2\boldsymbol{\mu}_i^T\mathbf{\Sigma}\boldsymbol{\mu}_i$) and a linear combination of the components of $\mathbf{x}$, a *linear discriminant score* is defined as

$$d_i(\mathbf{x}) = \boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln p_i \quad (3.43)$$

An estimate of $d_i(\mathbf{x})$ can be computed based on the *pooled* estimate of $\mathbf{\Sigma}$ [126]:

$$\hat{d}_i(\mathbf{x}) = \bar{\mathbf{x}}_i^T\mathbf{S}_{pl}^{-1}\bar{\mathbf{x}} - \frac{1}{2}\bar{\mathbf{x}}_i^T\mathbf{S}_{pl}^{-1}\bar{\mathbf{x}}_i + \ln p_i \quad i = 1, \cdots, g \quad (3.44)$$

where

$$\mathbf{S}_{pl} = \frac{1}{n_1 + n_1 + \cdots + n_g - g}\left[(n_1 - 1)\mathbf{S}_1 + \cdots + (n_g - 1)\mathbf{S}_g\right] \quad (3.45)$$

and n_g denotes the data length (number of observations) in class g. The minimum total probability of misclassification rule for Normal populations with equal covariance matrices becomes [126]:

Allocate $\mathbf{x}$ to π_k if $\hat{d}_k(\mathbf{x})$ is the *largest* of all $\hat{d}_i(\mathbf{x})$, $i = 1, \cdots, g$.

3.5.3 Fisher's Discriminant Analysis

Fisher suggested to transform the multivariate observations $\mathbf{x}$ to another coordinate system that enhances the separation of the samples belonging to each class π_i [74]. Fisher's discriminant analysis (FDA) is optimal in terms of maximizing the separation among the set of classes. Suppose that there is a set of $n(= n_1 + n_2 + \cdots + n_g)$ m-dimensional (number of process variables) samples $\mathbf{x}_1, \cdots, \mathbf{x}_n$ belonging to classes π_i, $i = 1, \cdots, g$. The total scatter of data points ($\mathbf{S}_T$) consists of two types of scatter, *within-class scatter* $\mathbf{S}_W$ and *between-class scatter* $\mathbf{S}_B$. The objective of the transformation proposed by Fisher is to maximize $\mathbf{S}_B$ while minimizing $\mathbf{S}_W$. Fisher's approach does not require that the populations have Normal distributions, but it implicitly assumes that the population covariance matrices are equal, because a pooled estimate of the common covariance matrix ($\mathbf{S}_{pl}$) is used (Eq. 3.45).

FDA for data belonging to two classes

The transformation is based on a weighted sum of observations $\mathbf{x}$. In the case of two classes, the linear combination of the samples ($\mathbf{x}$) takes values $z_{11}, \cdots, z_{1p1}$ for the observations from the first population π_1 and the values $z_{21}, \cdots, z_{2p2}$ for the observations from the second population π_2. Denote the weight vector that transforms $\mathbf{x}$ to z by $\mathbf{w}$. FDA is illustrated for the case of two normal populations with a common covariance matrix in Figure 3.3. First consider separation using either x_1 or x_2 axis. The diagrams by the abscissa and ordinate indicate that several observations belonging to one class (π_1) are mixed with observations belonging to the other class (π_2). The linear discriminant function $z = \mathbf{w}^T\mathbf{x}$ defines the line in the upper portion of Figure 3.3 that observations are projected on in order to maximize the ratio of between-class scatter and within-class scatter [63, 126].

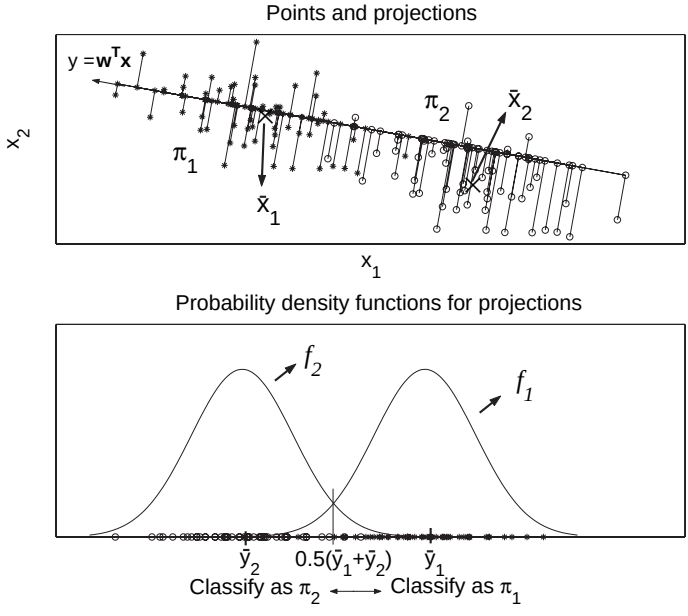


Figure 3.3. Fisher's discriminant technique for two populations ($g = 2$), $\pi_1(*)$ and $\pi_2(\circ)$, with equal covariances.

The separation of the two sets of z's can be assessed in terms of the difference between $\bar{z}_1$ and $\bar{z}_2$ expressed in standard deviation units:

$$\text{separation} = \frac{|\bar{z}_1 - \bar{z}_2|}{s_z} \tag{3.46}$$

where s_z^2 is the pooled estimate of the variance,

$$s_z^2 = \frac{1}{n_1 + n_2 - 2}\left[\sum_{j=1}^{n_1}(z_{1j} - \bar{z}_1)^2 + \sum_{j=1}^{n_2}(z_{2j} - \bar{z}_2)^2\right] . \tag{3.47}$$

The linear combination that maximizes the separation is [126]

$$\hat{z} = \mathbf{w}^T\mathbf{x} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\,\mathbf{S}_{pl}^{-1}\mathbf{x} \tag{3.48}$$

which maximizes the ratio

$$\frac{(\bar{z}_1 - \bar{z}_2)^2}{s_z^2} = \frac{(\mathbf{w}^T\bar{\mathbf{x}}_1 - \mathbf{w}^T\bar{\mathbf{x}}_2)^2}{\mathbf{w}^T\mathbf{S}_{pl}\mathbf{w}} = \frac{(\mathbf{w}^T\mathbf{d})^2}{\mathbf{w}^T\mathbf{S}_{pl}\mathbf{w}} \tag{3.49}$$

over all possible coefficient vectors $\mathbf{w}$ where $\mathbf{d} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$. The maximum of the ratio in Eq. 3.49 is $T^2 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$ [126]. For two populations with equal covariances, FDA corresponds to the particular case of the minimum ECM rule discussed in Section 3.5.2. The first terms in Eqs. 3.43 and 3.44 are the linear function obtained by FDA that maximizes the univariate between-class scatter relative to the within-class scatter (Eq. 3.48) [126].

The allocation rule of a new observation $\mathbf{x}_0$ to classes π_1 or π_2 based on FDA is [126]

Allocate $\mathbf{x}_0$ to π_1 if

$$(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}\mathbf{x}_0 \geq \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \tag{3.50}$$

Allocate $\mathbf{x}_0$ to π_2 otherwise.

Separation of Many Classes ($g > 2$)

The generalization of the *within-class scatter matrix* $\mathbf{S}_W$ for g classes is

$$\mathbf{S}_W = \sum_{i=1}^{g}(n_i - 1)\mathbf{S}_i \tag{3.51}$$

where n_i denotes the number of observations in class i and

$$\mathbf{S}_i = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T \qquad \bar{\mathbf{x}}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_{ij} \tag{3.52}$$

represents the covariance matrix and the mean vector for class i [63]. $\mathbf{S}_W/(n_1 + n_2 + \cdots + n_g - g) = \mathbf{S}_{pl}$ is an estimate of $\mathbf{\Sigma}$. The $\mathbf{w}$ that maximizes $\mathbf{w}^T\mathbf{S}_B\mathbf{w}/\mathbf{w}^T\mathbf{S}_{pl}\mathbf{w}$ also maximizes $\mathbf{w}^T\mathbf{S}_B\mathbf{w}/\mathbf{w}^T\mathbf{S}_W\mathbf{w}$.

Define the *between-class scatter matrix* $\mathbf{S}_B$ and the *total scatter matrix* $\mathbf{S}_T$ as [63, 118]:

$$\mathbf{S}_B = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \tag{3.53}$$

$$\mathbf{S}_T = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}})(\mathbf{x}_{ij} - \bar{\mathbf{x}})^T \tag{3.54}$$

where $\bar{\mathbf{x}}$ is the *total mean vector*

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{g} n_i\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{i=1}^{g}\sum_{j=1}^{n_i}\mathbf{x}_{ij} \tag{3.55}$$

and $n = \sum_{i=1}^{g} n_i$ denotes the total number of observations in all classes. Equation 3.54 can be rewritten by adding $-\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i$ to each term and rearranging the sums so that the total scatter is the sum of the within-class scatter and the between-class scatter as [63]:

$$\begin{aligned}
\mathbf{S}_T &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \\
&= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T + \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \quad (3.56) \\
&= \mathbf{S}_W + \mathbf{S}_B
\end{aligned}$$

The first FDA vector $\mathbf{w}_1$ that maximizes the scatter between classes ($\mathbf{S}_B$) while minimizing the scatter within classes ($\mathbf{S}_W$) is obtained as

$$\max_{\mathbf{W}\neq 0} \frac{\mathbf{W}^T\mathbf{S}_B\mathbf{W}}{\mathbf{W}^T\mathbf{S}_W\mathbf{W}} \tag{3.57}$$

under the assumption of $\mathbf{S}_W$ being invertible [63, 36]. The second FDA vector is calculated to maximize the scatter between classes while minimizing the scatter within classes among all axes perpendicular to the first FDA vector ($\mathbf{w}_1$). Additional FDA vectors are determined if necessary by using the same maximization objective and orthogonality constraint. These FDA vectors $\mathbf{w}_a$ form the columns of an optimal $\mathbf{W}$ that are the generalized eigenvectors corresponding to the largest eigenvalues in

$$\mathbf{S}_B\mathbf{w}_a = \lambda_a\mathbf{S}_W\mathbf{w}_a \tag{3.58}$$

where the magnitude ordered eigenvalues λ_a indicate the degree of overall separability among the classes by linearly transforming the data onto $\mathbf{w}_a$

[63, 36]. The eigenvalues in Eq. 3.58 can be computed as the roots of the characteristic polynomial $\mathtt{det}(\mathbf{S}_B - \lambda_a \mathbf{S}_W) = 0$ and then solving $(\mathbf{S}_B - \lambda_a \mathbf{S}_W)\mathbf{w}_a = 0$ directly for the eigenvectors $\mathbf{w}_a$ [63].

Classification with FDA

FDA is used to diagnose faults by modifying the *quadratic discrimination score* for the ith population defined in Eq. 3.41 in the FDA framework such that

$$d_i{}^Q(\mathbf{x}_0) = \ln p_i - \frac{1}{2}(\mathbf{x}_0 - \bar{\mathbf{x}}_i)^T \mathbf{W}_a \left(\mathbf{W}_a^T \mathbf{S}_i \mathbf{W}_a\right)^{-1} \mathbf{W}_a^T (\mathbf{x}_0 - \bar{\mathbf{x}}_i) - \frac{1}{2} \ln \left[det \left(\mathbf{W}_a^T \mathbf{S}_i \mathbf{W}_a\right)\right] \tag{3.59}$$

where $\mathbf{W}_a$ contains the first a FDA vectors [36]. The allocation rule is:

Allocate $\mathbf{x}_0$ to π_k if $d_k{}^Q(\mathbf{x}_0)$ is the *largest* of all $d_i{}^Q(\mathbf{x}_0),\ i = 1, \cdots, g.$

The classification rule in conjunction with Bayes' rule is used [126, 36] so that the *posterior* probability (Eq. 3.38) assuming $\sum_{i=1}^{g} P(\pi_k|\mathbf{x}) = 1$ that the class membership of the observation $\mathbf{x}_0$ is i. This assumption may lead to a situation where the observation will be classified wrongly to one of the fault cases which were used to develop the FDA discriminant when an unknown fault occurs. Chiang *et al.* [36] proposed several screening procedures to detect unknown faults. One of them involves FDA related T^2-statistic before applying Eq. 3.59 as

$$T_{i,a}^2 = (\mathbf{x} - \bar{\mathbf{x}}_i)^T \mathbf{W}_a \left(\mathbf{W}_a^T \mathbf{S}_i \mathbf{W}_a\right)^{-1} \mathbf{W}_a^T (\mathbf{x} - \bar{\mathbf{x}}_i) \tag{3.60}$$

so that it can be used to determine if the observation is associated with fault class i. The threshold for $T_{i,a}^2$ is defined as

$$T_{\alpha,a}^2 = \frac{a(n-1)(n+1)}{n(n-a)} F_\alpha(a, n-a) \tag{3.61}$$

where $F_\alpha(a, n-a)$ denotes the F distribution with a and $n-a$ degrees of freedom [126]. Chiang *et al.* [36] introduced another class of data that are collected under NO to allow the class information in the known fault data to improve the ability to detect faults. The first step then becomes the detection of an out-of-control situation. A threshold for NO class is developed based on Eq. 3.61 for detection; if $T_{i,a}^2 \geq T_{\alpha,a}^2$, there is an out-of-control situation. One proceeds with calculation at thresholds for each class i using Eq. 3.61. If $T_{i,a}^2 \geq T_{\alpha,a}^2$ for all $i = 1, \ldots, g$, then the observation $\mathbf{x}_0$ does not belong to any fault class i, and it is most likely associated with an unknown fault. If $T_{i,a}^2 \leq T_{\alpha,a}^2$ for some fault class i,

then $\mathbf{x}_0$ belongs to a known fault class. Once this is determined, Fisher's discriminant score in Eq. 3.59 can be used to assign it to a fault class π_i with the highest ${d_i}^Q(\mathbf{x}_0)$ of all ${d_i}^Q(\mathbf{x}_0),\ i = 1, \cdots, g$.

FDA and PCA can also be combined to avoid assigning an unknown fault to one of the known fault classes [118, 36, 260]. PCA is widely used for fault detection as discussed in Chapter 5. Chiang *et al.* [36] proposed two algorithms incorporating FDA and PCA. In the first algorithm (PCA/FDA), PCA is used to detect unknown faults and FDA to diagnose faults (by assigning them to fault classes). The NO class and classes with fault conditions are used to develop the PCA model. When a new observation $\mathbf{x}_0$ becomes available, T_a^2 value is calculated based on PCA as

$$T_a^2 = \mathbf{x}_0^T \mathbf{P}_a \lambda_a^{-1} \mathbf{P}_a^T \mathbf{x}_0 \tag{3.62}$$

where λ_a is $(a \times a)$ diagonal matrix containing eigenvalues and $\mathbf{P}$ are the loading vectors. A set of threshold values based on NO and the known fault classes using Eq. 3.61 is calculated. If $T_a^2 \leq T_{\alpha,a}^2$, it is concluded that this is a known class (either NO or faulty) and FDA assignment rule is used to diagnose the fault class (or NO class if it is in-control).

The second combined algorithm (FDA/PCA) deploys FDA initially to determine the most probable fault class i. Then it uses PCA T^2-statistic to find out if the observation $\mathbf{x}_0$ is truly associated with fault class i.

3.6 Nonlinear Methods for Diagnosis

This section introduces artificial neural networks, kernel-based techniques and support vector machines to establish the basis of monitoring techniques to be discussed in the subsequent chapters.

3.6.1 Neural Networks

Artificial neural networks (ANNs) can be used for modeling nonlinear systems, classification and fault diagnosis. ANNs have been inspired from the way the human brain works as an information-processing system in a highly complex, nonlinear and massively parallel fashion. Other names for ANNs include parallel distributed processors, connectionist models (or networks), self-organizing systems, neuro-computing systems and neuromorphic systems. ANNs have a large number of highly interconnected *nodes* also called as processing elements or artificial neurons. The first computational model of a biological neuron, the *binary threshold unit* was proposed by McCulloch and Pitts in 1943 [192]. Interest in ANNs was gradually revived in

1980s when Rumelhart *et al.* [257] popularized a much faster learning procedure called *back-propagation*, which could train a multi-layer perceptron to compute any desired function.

ANNs are nonlinear 'black-box' systems. This nonlinearity is distributed throughout the network. ANNs have the ability to adapt, or learn, in response to variations in their environment through training. They can be *retrained* to deal with minor changes in the operational and/or environmental conditions. When operating in a *non-stationary* environment, ANNs can be designed to adjust their synaptic weights in real-time. This is valuable in adaptive pattern classification and adaptive control. ANNs perform multivariable pattern recognition tasks very well. They can learn from examples (training) by constructing an *input-output mapping* for the system of interest. In the pattern classification case an ANN can be designed to provide information about similar and unusual patterns. Training and pattern recognition must be made by using a closed set of patterns. All possible patterns to be recognized should be present in the data set. A properly designed and implemented ANN is usually capable of robust computation. Its performance degrades gracefully under adverse operating conditions and when some of its connections are severed. ANNs have some serious limitations as well. Training ANNs may take long times when structurally complex ANNs or inappropriate optimization algorithms are used. ANNs may not produce reliable results if the size of input-output data is small. Their accuracy for modeling and classification improve when large amounts of historical data rich in variations are available. Training may cause the network to be accurate in some operating zones, but inaccurate in others. While trying to minimize the error during training, the optimization may get trapped in local minima. Like all data-based techniques, there is no guarantee of complete reliability or accuracy. In fault diagnosis applications, for example, ANNs may misdiagnose some faults 1% of the time while other faults in the same domain 25% of the time. It is hard to determine *a priori* (when back-propagation algorithm is used) what faults will be prone to higher levels of misdiagnosis. There are practical problems related to training data set selection [152, 165].

The basic structure of ANNs typically includes multi-layered, interconnected neurons (or computational units) that nonlinearly relate input-output data. A nonlinear model of a neuron, which forms the core of the ANNs is characterized by three basic attributes (Figure 3.4):

1. A set of *connections (synaptic weights)* describing the amount of influence a node has on nodes in the next layer; a positive weight causes one unit to excite another, while a negative weight causes one unit to inhibit another. The signal x_j at the input synapse j connected to

neuron k in Figure 3.4 is multiplied by weight w_{kj} (see Eq. 3.63).

2. A *summation operator* of input signals, weighted by the respective synapses of the neuron.

3. An *activation function* with limits on the amplitude of the output of a neuron. The amplitude range is usually given in a closed interval [0,1] or [-1,1]. Activation function $\varphi(\cdot)$ defines the output y_k of a neuron (see Eq. 3.65) in terms of the activation potential v_k (see Eq. 3.64). Typical activation functions include the unit step change and sigmoid functions.

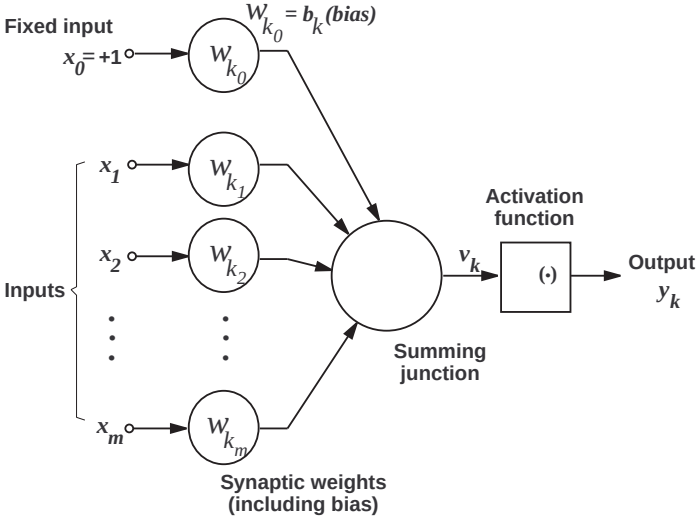


Figure 3.4. A nonlinear model of a single neuron as illustrated in [107].

A neuron k can be described by the following set of equations [107]:

$$u_k = \sum_{j=0}^{m} w_{kj} x_j \tag{3.63}$$

$$v_k = u_k + b_k \tag{3.64}$$

and

$$y_k = \varphi(v_k) \tag{3.65}$$

where $x_1, x_2, \ldots, x_j, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{kj}, .., w_{km}$ are the synaptic weights of neuron k, u_k is the linear combiner output of the

input signals, b_k is the bias, v_k is the activation potential (or induced local field), $\varphi(\cdot)$ is the activation function, and y_k is the output signal of the neuron. The bias is an external parameter providing an affine transformation to the output u_k of the linear combiner.

Several activation functions are used as appropriate to the task at hand:

1. *Threshold Function.* Also known as McCulloch-Pitts model [192]

$$\varphi(v) = \begin{cases} 1, & v \geqslant 0 \\ 0, & v < 0. \end{cases} \tag{3.66}$$

2. *Piecewise-linear Function.*

$$\varphi(v) = \begin{cases} 1, & v \geqslant +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leqslant -\frac{1}{2} \end{cases} \tag{3.67}$$

   where the amplification factor inside the linear region of operation is assumed to be the unity.

3. *Sigmoid Function.* This S-shaped function is by far the most common form of activation function used. A typical expression is

$$\varphi(v) = \frac{1}{1 + e^{-av}} \tag{3.68}$$

   where a is the slope parameter.

4. *Hyperbolic Tangent Function.* This is a form of sigmoid function but it produces values in the range $[-1, +1]$ instead of $[0, 1]$

$$\varphi(v) = \tanh(v) = \frac{e^{v} - e^{-v}}{e^{v} + e^{-v}} \ . \tag{3.69}$$

Processing units (neurons) are linked to each other to form a network associated with a learning algorithm. A neural network can be formed with any kind of topology (architecture). In general, three kinds of network topologies are used [107]:

- *Single-layer feedforward networks* include input layer of source nodes that projects onto an output layer of neurons (computation nodes), but not vice versa. They are also called *feedforward* networks. Since the computation takes place only on the output layer nodes, the input layer does not count as a layer (Figure 3.5(a)).

- *Multi-layer feedforward networks* contain an input layer connected to one or more layers of hidden neurons (hidden units) and an output layer (Figure 3.5(b)). The hidden units internally transform the data representation to extract higher-order statistics. The input signals are applied to the neurons in the first hidden layer, the output signals of that layer are used as inputs to the next layer, and so on for the rest of the network. The output signals of the neurons in the output layer reflect the overall response of the network to the activation pattern supplied by the source nodes in the input layer. This type of network is especially useful for pattern association (i.e., mapping input vectors to output vectors).

- *Recurrent networks* differ from feedforward networks in that they have at least one *feedback* loop. An example of this type of network is given in Figure 3.5(c) which is one of the earliest recurrent networks called Jordan network [131]. The activation values of the output units are fed back into the input layer through a set of extra units called the *state units*. Learning takes place in the connection between input and hidden units as well as hidden and output units. Recurrent networks are useful for pattern sequencing (i.e., following the sequences of the network activation over time). The presence of feedback loops has a profound impact on the learning capability of the network and on its performance [107]. Applications to chemical process modeling and identification have been reported [32, 310, 345].

Techniques for network architecture selection for feedforward networks have been proposed [151, 166, 234, 317, 318]. Once the network architecture is specified, an input-output data set is used to *train* the network. This involves the computation of appropriate values for the weights associated with each interconnection. The data are propagated forward through the network to generate an output to be compared with the actual output, and based on error magnitudes the weights are adjusted to minimize the error. The overall procedure of training can be seen as *learning* for the network from its environment through an interactive process of adjustments applied to its weights and bias levels. A number of learning rules such as *error-correction, memory-based, Hebbian, competitive, Boltzmann* learning have been proposed [107] to adjust network weights.

There are two *learning paradigms* that determine how a network relates to its environment. In *supervised learning* (learning with teacher), a *teacher* provides output targets for each input pattern, and corrects the network's errors explicitly. The teacher has knowledge of the environment (in the form of a historical set of input-output data) so that the neural network is provided with desired response when a training vector is available. The

desired response represents the optimum action to be performed to adjust neural network weights under the influence of the training vector and error signal. The *error signal* is the difference between the desired response (historical value) and the actual response (computed value) of the network. This corrective algorithm is repeated iteratively until a preset convergence criterion is reached. One of the most widely used supervised training algorithms is the *error back-propagation* or *generalized delta rule* [257, 321]. The alternative is learning without a teacher in which the network must find the regularities in the training data by itself. This paradigm has two subgroups: Reinforcement learning and unsupervised learning. In *Reinforcement learning/Neurodynamic programming*, where learning the relationship between inputs and outputs is performed through continued interaction with the environment to minimize a scalar index of performance [19]. In *unsupervised learning*, or *self-organized learning* there is no external teacher to oversee the learning process. Once the network is tuned to the statistical regularities of the input data, it forms internal presentations for encoding the input automatically [17, 107].

There are many educational and commercial software packages available for development and deployment of ANNs. Some of those packages such as Gensym's NeurOn-Line® Studio include data preprocessing modules to filter or scale data and eliminate outliers [89].

Autoassociative Neural Networks

Autoassociative neural networks provide a special five-layer network structure (Figure 3.6) that can implement nonlinear PCA by reducing variable dimensionality and producing a feature space map that retains the maximum possible amount of information from the original data set [150]. Autoassociative neural networks use conventional feedforward connections and sigmoidal or linear nodal transfer functions.

The network has three hidden layers, including a 'bottleneck' layer which is of a smaller dimension than either the input layer or the output layer. The network is trained to perform an identity mapping by approximating the input information at the output layer. Since there are fewer nodes in the bottleneck layer than the input or output layers, the bottleneck nodes implement data compression and encode the essential information in the inputs for its reconstruction in subsequent layers. In the NLPCA framework and terminology, autoassociative neural networks seek to provide a mapping of the form

$$\mathbf{T} = \mathbf{G}(\mathbf{X}) \tag{3.70}$$

where $\mathbf{G}$ is a nonlinear vector function composed of f individual nonlinear functions $\mathbf{G} = [G_1\ G_2\ \cdots\ G_f]$ analogous to the loading vectors $\mathbf{P}$. The inverse transformation that reconstructs the input information and restores

the original dimensionality of the data is implemented by a second nonlinear vector function $\mathbf{H} = [H_1\ H_2\ \cdots\ H_m]$ where m is the number of variables

$$\hat{X}_j = H_j(\mathbf{T}) \quad \text{and} \quad \mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}} \tag{3.71}$$

where the residual $\mathbf{E}$ indicates the loss of information that is minimized by the selection of functions $\mathbf{G}$ and $\mathbf{H}$. The number of bottleneck nodes is similar to the number of principal components retained for the selection of the subspace dimension that retains relevant information in data.

The limitations of autoassociative NNs to implement NLPCA is discussed by Malthouse [183]. Their use in process monitoring problems is reported in [55, 61].

3.6.2 Kernel-Based Techniques

If one does not wish to bias the boundaries of the NO region of a system, kernel density estimation (KDE) can be used to find the contours underneath the joint probability density of the PC pair, starting from the one that captures most of the information. Below, a brief review of KDE is presented first that will be used as part of the robust monitoring technique discussed in Section 7.7. Then, the use of kernel-based methods for formulating nonlinear Fisher's discriminant analysis (FDA) is discussed.

A kernel is a function K such that for all $\mathbf{u}, \mathbf{v} \in X$

$$K(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}) \cdot \phi(\mathbf{v}) \rangle \tag{3.72}$$

where ϕ is a mapping from the input space X to an (inner product) feature space F and $\langle \cdot, \cdot \rangle$ denotes the inner product. Some of the popular kernels are:

- Linear support vector machines: $K(\mathbf{u}, \mathbf{v}) = \mathbf{v}^T\mathbf{u}$
- Nonlinear support vector machines: $K(\mathbf{u}, \mathbf{v}) = (\tau + \mathbf{v}^T\mathbf{u})^d$
- Radial basis function kernel: $K(\mathbf{u}, \mathbf{v}) = exp\ (-\|\mathbf{u} - \mathbf{v}\|_2^2/\sigma^2)$
- Multi-layer perceptron: $tanh\ (\kappa_1\mathbf{v}^T\mathbf{u} + \kappa_2)$

Kernels are symmetric functions. Mercer's theorem provides a characterization of kernels: a symmetric function $K(\mathbf{u}, \mathbf{v})$ is a kernel function if and only if the matrix

$$\mathbf{K} = \big(K(\mathbf{u}_i, \mathbf{v}_i)\big)_{i,j=1}^{n} \tag{3.73}$$

is positive semi-definite (has nonnegative eigenvalues) [42]. Mercer's condition holds for all σ values in the radial basis function kernels and positive values of τ in polynomial kernels, but not for all positive choices of κ_1 and κ_2 in multi-layer perceptron kernels [287].

Kernel Density Estimation

The density function f of any random quantity x gives a natural description of the distribution of a data set x and allows probabilities (P) associated with x to be found as follows [268],

$$P\{a < x < b\} = \int_a^b f(x)dx \quad \forall\ a < b \tag{3.74}$$

A set of observed data points is assumed to be available as samples from an unknown probability density function. Density estimation is the construction of an estimate of the density function from the observed data. In *parametric* approaches, one assumes that the data belong to one of a known family of distributions and the required function parameters are estimated. This approach becomes inadequate when one wants to approximate a multi-model function, or for cases where the process variables exhibit nonlinear correlations [127]. Moreover, for most processes, the underlying distribution of the data is not known and most likely does not follow a particular class of density function. Therefore, one has to estimate the density function using a *nonparametric* (unstructured) approach.

The histogram is perhaps the most common yet the simplest density estimator. While its computational ease and graphical representation are a benefit for univariate signals, the visualization of higher dimensional data becomes problematic. To construct the histogram, one has to choose an origin and a bin width. The selection of these parameters determines the degree of smoothing inherent in the procedure. An alternative density estimate is the naive estimator which can be unsatisfactory as its bin width still needs to be established to produce a density estimate. Despite the simplicity of the histogram and naive estimates, their discontinuous representation of the density function causes difficulty if the derivatives of the estimate or smooth representation of the estimate are required [268]. Thus, a kernel or a wavelet density estimation method may be preferred [265, 262].

Kernel estimate with kernel K is defined by

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.75}$$

Here, h denotes the window width, and is also referred to as the smoothing parameter. The quality of a density estimate is primarily determined by the choice of the parameter h, and only secondarily by the choice of the kernel K [265, 21]. For applications, the kernel K is often selected as a symmetric probability density function, e.g., the Normal density.

To decide how much to smooth is a critical step in density estimation. A number of alternative measures exist to estimate h [265]. The appropriate

choice is, in fact, influenced by the purpose for which the density estimate is to be used. For robust monitoring (see Section 7.7), h is selected using least squares cross-validation [256, 21].

Kernel-Based FDA

One way to extend discriminant analysis to nonlinear cases is to use kernel-based FDA. Kernel methods embed data into a feature vector space and detect linear relationships in that space. The linear relations include regression, classification or principal components. If the feature space is selected 'properly', pattern recognition can be easy. Kernel-based algorithms are structured as two modules: A kernel function that implements embedding of data into the feature space and a learning algorithm that learns in the linear feature space. Kernel methods exploit information about the inner products between data items. The inner products in feature space can be very complex, but if the kernel is given, there is no need to specify what features of the data are being used. Usually the kernel function type such as polynomials, radial basis functions or splines is selected in advance according to the nature of the application and its parameters are computed for the specific problem information. Mercer's theorem is used to characterize if a symmetric function is a kernel. A Bayesian framework has been developed for SVM classifiers, Gaussian processes and kernel FDA [306].

3.6.3 Support Vector Machines

When linear classification tools do not provide reliable fault diagnosis, nonlinear techniques are needed. Neural network based classification has been implemented for over a decade for cases where a small number of faults in a closed set are to be diagnosed [252, 63]. A shortcoming of NN-based FD is the possibility of converging to local optima during training. Support Vector Machines (SVM) with kernel-based learning methods provide another powerful alternative. SVMs are learning systems based on statistical learning theory [308] that use a space of linear functions in a high dimensional feature space F for classification problems. Support vectors are representative training data points that provide the best hyperplanes for separating various classes in the data. The aim of support vector (SV) classification is to devise a computationally efficient way of learning 'good' separating hyperplanes in the feature space [42].

To learn nonlinear relations with a linear machine, a set of nonlinear features are selected and the data are 'rewritten' in a new representation. This is achieved by applying a fixed nonlinear mapping of the data to a feature space where the linear machine can be used. The set of hypotheses

considered can be functions

$$f(\mathbf{x}) = \sum_{i=1}^{g} w_i \phi_i(\mathbf{x}) + b \tag{3.76}$$

where w_i are weights, g the dimension of the feature space, b the bias, and $\phi : X \rightarrow F$ is nonlinear map from the input space X to some feature space F [42]. The nonlinear machine can thus be constructed in two steps: (1) a fixed nonlinear mapping transforms data into the feature space F and (2) a linear machine is used to classify them in the feature space.

Consider a system with k pattern classes. The general pattern recognition problem with k classes is to construct a decision function given l independent and identically distributed samples of an unknown function $(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_l, \mathbf{y}_l)$ where $\mathbf{x}_i$ (in the attribute space $\mathbf{X}$) is of length d and $\mathbf{y}_i$ is of length k. The decision function $f(\mathbf{x}, \alpha)$ is chosen from a set of functions selected a priori and is defined by the parameter α for the problem at hand. To select α, the loss $L(\mathbf{y}, f(\mathbf{x}, \alpha))$ is minimized. For example, for the binary pattern recognition problem ($k = 2$), a hyperplane is constructed to separate the two classes labeled $\mathbf{y} \in \{-1, 1\}$ so that the distance between the hyperplane and the nearest point (the margin) is maximized. This yields the following optimization problem:

$$\begin{array}{ll} \min & J_P(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + C \sum_{i=1}^{l} \boldsymbol{\xi}_i \\ \text{such that} & y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \boldsymbol{\xi}_i, \quad i = 1, \cdots, l \\ & \boldsymbol{\xi}_i \geq 0, \quad i = 1, \cdots, l \end{array} \tag{3.77}$$

where $\mathbf{w}$ is the weight vector and $\boldsymbol{\xi}_i$ are the slack variables. The regularization parameter C adjusts the trade-off between the two terms of the objective function J_P in Eq. 3.77. The first term represents model complexity and the second term model accuracy that is related to classification error in the training data. For small values of C, the model does not have enough detail to describe the data. Large values of C cause over-fitting. Methods for selecting optimal values of C have been developed by taking into account the kernel function used, the noise level and the characteristics of the feature space [130].

Linear learning machines can be expressed in a dual representation, enabling expression of the hypotheses as a linear combination of the training point $(\mathbf{x}_i)$ so that the decision rule can be evaluated by using just inner products between the test points $(\mathbf{x})$ and the training points:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \mathbf{y}_i \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle + b \; . \tag{3.78}$$

Kernels are used to compute the inner product $\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle$ directly in the feature space as a function of input points and to merge the two steps of the nonlinear learning machine.

The *dual* solution to this problem is:

$$\begin{aligned} \max \quad & J_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \mathbf{y}_i \mathbf{y}_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{such that} \quad & \sum_{i=1}^{j} \alpha_i \mathbf{y}_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, l \end{aligned} \tag{3.79}$$

where ϕ denotes a mapping $\phi : X \to F$ and the kernel function $K(\mathbf{u}, \mathbf{v}) = \langle \phi(u) \cdot \phi(v) \rangle$ has been introduced instead of the linear relationship in Eq. 3.77. The resulting decision function is

$$f(\mathbf{x}) = sign \left[\left(\sum_{i=1}^{l} \alpha_i \mathbf{y}_i K(\mathbf{x}, \mathbf{x}_i) \right) + b \right] \tag{3.80}$$

In both the dual solution and decision function, only the inner product in the attribute space and the kernel function based on attributes appear, but not the elements of the very high dimensional feature space. The constraints in the dual solution imply that only the attributes closest to the hyperplane, the so-called SVs, are involved in the expressions for weights $\mathbf{w}$. Data points that are not SVs have no influence and slight variations in them (for example caused by noise) will not affect the solution. ξ_i provides a more quantitative leverage against noise in data that may prevent linear separation in feature space [42]. Imposing the requirement that the kernel satisfies Mercer's conditions ($\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)_{i,j=1}^{l}$ must be positive semi-definite) means that the matrix $\mathbf{y}_i \mathbf{y}_j (K(\mathbf{x}_i, \mathbf{x}_j)_{i,j=1}^{l}$ is also positive semi-definite. Consequently, the optimization in Eq. 3.79 is convex and has a unique solution that can be found efficiently, ruling out the problem of local minima encountered in training neural networks [42].

The k-class pattern recognition problem with SVMs was initially solved by using one-against-the-rest and one-against-one classifiers. Recently, k-class SVMs have been proposed [324]. The optimization problem Eq. 3.79 is generalized to yield the decision function

$$f(\mathbf{x}, \alpha) = \arg \max_{n} \left[\sum_{i:y_i=n} A_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle - \sum_{i:y_i=n} \alpha_i^n \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b_n \right] \tag{3.81}$$

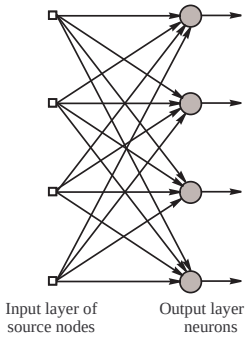
where

$$A_i = \sum_{m=1}^{k} \alpha_i^m$$

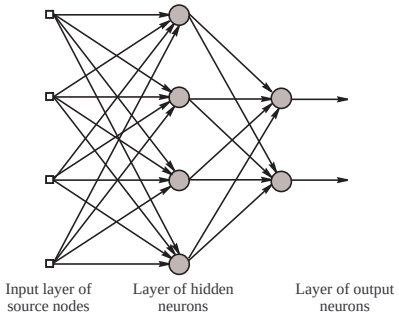
and the inner product $\langle \mathbf{x}_i \cdot \mathbf{x} \rangle$ can be replaced with the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ [324].

3.7 Summary

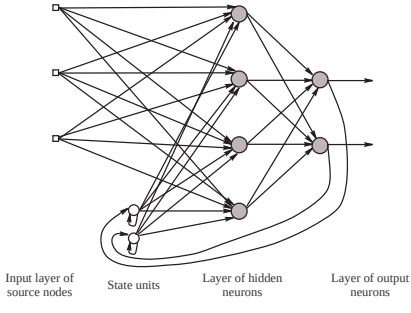
Various statistical methods that provide the foundations for model development, process monitoring and fault diagnosis are presented in this chapter. Linear techniques such as principal components analysis, partial least squares, canonical variates analysis and independent components analysis enable the development of powerful multivariate techniques for detection of abnormal process operation. Once an abnormality is detected and validated, its source cause must be determined. One approach is the use of contribution plots that indicate process variables that have made large contributions to significant changes in monitoring statistics. When these variables are identified, process knowledge is used to pin down the source cause of the abnormality. The other alternative is the use of statistical classification methods such as Fisher's discriminant analysis for diagnosis of source causes directly. Both detection and diagnosis techniques can be developed using a nonlinear approach. Nonlinear methods like neural networks, kernel density estimation and support vector machine are introduced in the last section to provide insight into the deployment of such monitoring and diagnosis tools.



(a) Single-layer feedforward network.



(b) Multi-layer feedforward network.



(c) Recurrent network as in [131].

Figure 3.5. Three fundamentally different network architectures.

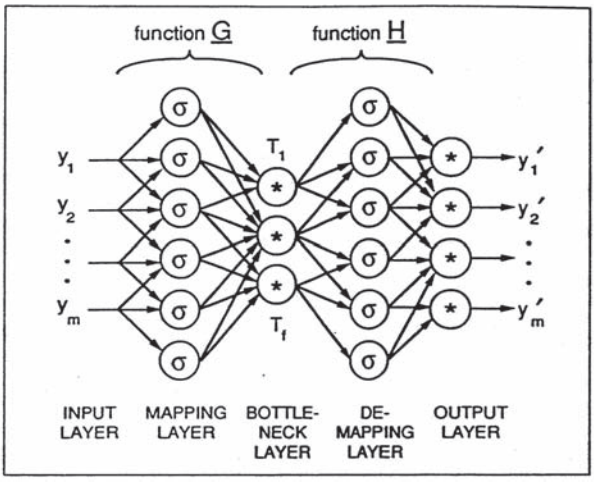


Figure 3.6. Network architecture for determination of f nonlinear factors using an autoassociative neural network. σ indicates nodes with sigmoidal functions, $*$ indicates nodes with sigmoidal or linear functions [150].

4

Empirical Model Development

Process models may be developed by using either first principles such as material and energy balances, or process input and output information (data). *First principles (fundamental) models* describe the internal dynamics of the process based on physical, chemical or biological laws, and *explain* the behavior of the process. But the cost of model development is high. They may be biased by the views and speculations of the model developer, and are limited by the lack of information about specific model parameters. Often, some physical, chemical or transport parameters are computed using empirical relations, or they are derived from experimental data. In either case, there is some uncertainty about their accuracy. As details are added to the model, it may become too complex and too large to run model computations on the computer within an acceptable amount of time. Fundamental models developed may be too large for computations that are fast enough to be used in process monitoring and control activities. These activities require fast update of model predictions so that regulation of process operation can be made in a timely manner.

The alternative model development paradigm is based on developing relations based on process data. *Input-output models* are much less expensive to develop. However, they only *describe* the relationships between the process inputs and outputs, and their utility is limited to features that are included in the available data sets. There are numerous well-established techniques for *linear* input-output model development. Methods for development of linear models are easier to implement and more popular. Since most monitoring and control techniques are based on the linear framework, use of linear models is a natural choice. The design of experiments to collect data and the amount of data available have an impact on the accuracy and predictive capability of the model developed. Data collection experiments should be designed such that all key features of the process are excited in

the frequency ranges of interest. These models can be used for interpolation but they should not be used for extrapolation.

Nonlinear empirical models are more accurate over a wider range of operating conditions and they are more appealing for processes with strong nonlinearities. Various *nonlinear* input-output model development techniques have been proposed during the last fifty years, but they have not been widely accepted. The model structures are dependent on the type of nonlinearities in the data. Since the model may have terms that are composed of combinations of inputs and/or outputs, exciting and capturing the interactions among variables is crucial. Hence, the use of routine operational data for model development, without any consideration of exciting the key features of the model, may yield good fits to the data, but provide models that have poor predictive ability. The amount of data needed for model development is the smallest for first principle models, moderate for linear input-output models, and the largest for nonlinear input-output models.

As manufacturing processes have become increasingly instrumented in recent years, more variables are being measured and data are being recorded more frequently. This yields data overload, and most of the useful information may be hidden in large data sets. The correlated or redundant information in these process measurements must be refined to retain the essential information about the process. Process knowledge must be extracted from measurement information, and presented in a form that is easy to display and interpret. Various methods based on multivariate statistics, systems theory and artificial intelligence are presented in this chapter for data-based input-output model development.

Models are developed to satisfy different types of objectives. One case is the interpretation and modeling of one block of data such as measurements of process variables. Principal components analysis (PCA) may be useful for this to retain essential process information while reducing the size of the data set. A second case is the development of a relationship between two groups of data such as process variables and product variables, i.e., the regression problem. PCA regression or partial least squares (PLS) regression techniques would be good candidates for addressing this problem. Discrimination and classification are activities also related to process monitoring that lead to fault diagnosis. One can consider PCA and PLS based techniques as well as artificial neural networks (ANN) and knowledge-based systems for such problems. Since all these techniques are based on process data, the reliability of data is critical for obtaining dependable results from the implementation of these techniques.

ANNs (Section 3.6.1) provide one framework for nonlinear model development. Extensions of PCA and PLS to develop nonlinear models have

also been proposed. Several nonlinear time series modeling techniques have been reported. Nonlinear system science methods provide a different framework for nonlinear model development and model reduction. This chapter focuses on linear data-based modeling techniques. References are provided for their extensions to the nonlinear framework.

Various multivariate regression techniques are outlined in Section 4.1. Section 4.2 introduces PCA-based regression and its extension to capture dynamic variations in data. PLS regression is discussed in Section 4.3. Input-output modeling of dynamic processes with time series models is introduced in Section 4.4 and state-space modeling techniques are presented in Section 4.5.

4.1 Regression Models

Models between groups of variables such as process measurements $\boldsymbol{x}_{m\times 1}$ and quality variables $\boldsymbol{y}_{q\times 1}$ can be developed by using various regression techniques. Here, the subscripts indicate the vector dimensions (number of variables). If n samples have been collected for each group of variables, the data matrices are $\mathbf{X}_{n\times m}$ and $\mathbf{Y}_{n\times q}$. The existence of a model provides the opportunity to predict process or product variables and compare the measured and predicted values. The residuals between the predicted and measured values of the variables can be used to develop various SPM techniques (residuals-based univariate SPM was discussed in Section 2.3.1) and tools for identification of variables that have contributed to the out-of-control signal.

Consider a process with two measured variables ($m = 2$) and one quality variable ($q = 1$) that are related by a linear model. The term linear is used to indicate that the equation that relates the regressors $\boldsymbol{x} = [x_1 \; x_2]^T$ to the response (dependent) variable y_1 is a linear function of the equation parameters $\boldsymbol{\beta}$. The model equation that can be used for predicting new values of y_1 given values of $\boldsymbol{x}$ are

$$\hat{y}_1 = \beta_0 + \beta_1 \; x_1 + \beta_2 \; x_2 \tag{4.1}$$

where β_0 is the constant (intercept) term. The relationship may be more complex where interactions of the variables ($x_1 x_2$) or polynomial terms of regressors (for example, x_1^2 or x_2^3) can also be included in the model. For example, a second-order model with interaction for the same variables as above is:

$$\hat{y}_1 = \beta_0 + \beta_1 \; x_1 + \beta_2 \; x_2 + \beta_{11} x_1^2 + \beta_{22} \; x_2^2 + \beta_{12} \; x_1 x_2 \tag{4.2}$$

The interaction implies that the effect caused by changing one regressor variable depends on the level of the other regressor variable in the term.

The response surface of the models with such nonlinearities are not linear any more, but as long as the equation is linear in regression coefficients $\boldsymbol{\beta}$, it is considered a linear regression model.

The model equation for multivariable linear regression can be generalized and written in a compact form by using matrices for each of the q dependent variables $\mathbf{y}$ for the n data sets

$$\mathbf{y}_{n\times 1} = \mathbf{Z}_{n\times(m+1)}\boldsymbol{\beta}_{(m+1)\times 1} + \boldsymbol{\epsilon}_{n\times 1} \qquad \boldsymbol{\beta} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y} \tag{4.3}$$

where $\boldsymbol{\epsilon}$ is the random error which accounts for measurement error and effects of other variables not explicitly considered in the model, and

$$\mathbf{Z} = \begin{bmatrix} 1 & z_{11} & z_{12} & \cdots & z_{1m} \\ 1 & z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix} \tag{4.4}$$

with the first column of $\mathbf{Z}$ being a multiplier of the constant term in $\boldsymbol{\beta}$. It is assumed that $E(\boldsymbol{\epsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}$. The value of the dependent variable for a new observation at $\mathbf{z}_o$ is $\hat{y} = \mathbf{z}_o\boldsymbol{\beta}$.

The equations for various dependent variables $\mathbf{y}$ can be developed separately since it is assumed that they do not affect each other. The equations can be expressed in compact form as

$$\mathbf{Y}_{n\times q} = \mathbf{Z}_{n\times(m+1)}\boldsymbol{\beta}_{(m+1)\times q} + \boldsymbol{\epsilon}_{n\times q} \tag{4.5}$$

with

$$E(\boldsymbol{\epsilon}_i) = \mathbf{0} \quad \text{and} \quad Cov(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_j) = \sigma_{ij}\mathbf{I} \quad i,j = 1,2,\cdots,q \tag{4.6}$$

and the covariance matrix $\boldsymbol{\Sigma} = [\sigma_{ij}]$, but the observations from different trials are uncorrelated [126]. Multivariable linear regression is usually used for steady-state data, but by adding lagged values of variables one can extend it for time-varying data. However, time series models and state-space models are more useful for developing dynamic system models.

Colinearity among process variables can have a significant impact on the accuracy of the multivariable regression model and predictions. Colinearity causes numerical difficulties in computing the inverse $(\mathbf{X}^T\mathbf{X})^{-1}$ or $(\mathbf{Z}^T\mathbf{Z})^{-1}$ because some columns of $\mathbf{X}$ are almost identical and consequently the determinant is almost zero. This causes uncertainty and sensitivity in the estimates of $\boldsymbol{\beta}$. The standard errors of the estimates of regression coefficients $\boldsymbol{\beta}$ associated with the colinear regressors become very large. Colinearity can be detected by standardizing all predictor variables (mean-

centered, unit-variance) and computing correlations and coefficients of determination.

$$v_{ij} = \frac{x_{ij} - \bar{x}_j}{d_j} \quad d_j^2 = \sum_{i=1}^{m} (x_{ij} - \bar{x}_j)^2 \ , \quad i = 1, \cdots, n \ , j = 1, \cdots, m \quad (4.7)$$

There is significant degree of colinearity among some predictor variables if the following conditions hold:

1. The correlation between any two predictors exceeds 0.95 (only colinearity between *two* predictors can be assessed).

2. The coefficient of determination R_j^2 of each predictor variable j regressed on all the other predictor variables exceeds 0.90, or the variance inflation factor $VIF_j = (1 - R_j^2)^{-1}$ is less than 10 (variable j is colinear with one or more of the other predictors). VIF_j is the (j, j)th diagonal element of the matrix $\mathbf{V}^T\mathbf{V}^{-1}$ where $\mathbf{V} = [z_{ij}]$. R_j^2 can be computed from the relationship between R_j^2 and VIF_j.

3. Some of the *eigenvalues* of the correlation matrix $\mathbf{V}^T\mathbf{V}$ are less than 0.05. Large elements of the corresponding *eigenvectors* identify the predictor variables involved in the colinearity.

4. The determinant of $\mathbf{X}^T\mathbf{X}$ has a value between 0 and 1. In this case, the smaller the value of the determinant, the higher the degree of colinearity.

5. One or more eigenvalues of $\mathbf{X}^T\mathbf{X}$ having values near 0 implies the presence of colinearity.

Regression techniques that can deal with colinear data include stepwise regression, ridge regression, principal components regression, and partial least squares (PLS) regression. The last two approaches are discussed in Sections 4.2 and 4.3.

Stepwise regression

Stepwise regression is one of the early techniques that can deal with colinear data [108, 203]. Predictor variables are added to or deleted from the prediction (regression) equation one at a time. Stepwise variable selection procedures are useful when a large number of candidate predictors is available. It is expected that only one of the strongly colinear variables will be included in the model. Major disadvantages of stepwise regression are the limitations in identifying alternative candidate subsets of predictors, and the inability to guarantee the optimality of the final model.

Ridge Regression

The regression coefficients are biased by introducing a parameter along the diameter of $\mathbf{Z}^T\mathbf{Z}$ [109]. The computation of regression coefficients $\boldsymbol{\beta}$ in Eq. 4.3 is modified by introducing a ridge parameter κ:

$$\boldsymbol{\beta} = \left[\mathbf{Z}^T\mathbf{Z} + \kappa\mathbf{I}\right]^{-1}\mathbf{Z}^T\mathbf{y} \ . \tag{4.8}$$

Standardized ridge estimates β_j with $j = 1, \cdots, m$ are calculated for a range of values of κ and plotted versus κ. This plot is called a *ridge trace*. The $\boldsymbol{\beta}$ estimates usually change dramatically when κ is initially incremented by a small amount from 0. As κ is increased, the trace stabilizes. A κ value that stabilizes all $\boldsymbol{\beta}$ coefficients is selected and the final values of $\boldsymbol{\beta}$ are estimated.

A good estimate of the κ value is obtained using

$$\kappa = \frac{m\ MSE}{\sum_{j=1}^{m}(\beta_j^*)^2} \tag{4.9}$$

where β_j^*s are the least squares estimates for the standardized predictor variables, and MSE is the least squares mean squared error, $SSE/(n - m - 1)$.

Ridge regression estimators are biased. The trade-off for stabilization and variance reduction in regression coefficient estimators is the bias in the estimators and the increase in the squared error.

Nonlinear regression models are generated when one or more of the coefficients $\boldsymbol{\beta}$ are part of a nonlinear term [12] such as

$$\hat{y} = \beta_0 e^{\beta_1 x} \tag{4.10}$$

Chemical reaction rate terms are a familiar example to most chemists and chemical engineers. Sometimes, it is possible to make the equation linear by using a transformation such as taking the *log*. Otherwise, the computation of regression parameters become more complex.

4.2 PCA Models

Principal components regression (PCR) is one of the techniques to deal with ill-conditioned data matrices by regressing the dependent variables such as quality measurements on the principal components scores of regressor variables such as the measured variables (flow rates, temperature) of the process. The implementation starts by representing the data matrix $\mathbf{X}$ with its scores matrix $\mathbf{T}$ using the transformation $\mathbf{T} = \mathbf{XP}$. The number of principal components to retain in the model is determined as in the

PCA to reduce the effect of noise and to optimize the predictive power of the PCR model. This is generally done by using cross-validation. Then, the regression equation becomes

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\beta} + \mathbf{E} \tag{4.11}$$

where the optimum matrix of regression coefficients $\boldsymbol{\beta}$ is obtained as

$$\hat{\boldsymbol{\beta}} = (\mathbf{T^T T})^{-1}\mathbf{T^T Y} \tag{4.12}$$

In contrast to the inversion of $\mathbf{X}^T\mathbf{X}$ when some of the $\mathbf{x}$ are colinear, the inversion of $\mathbf{T}^T\mathbf{T}$ does not cause any problems due to the mutual orthogonality of the scores. Score vectors corresponding to small eigenvalues can be left out in order to avoid colinearity problems. Since principal components regression is a two-step method, there is a risk that useful predictive information would be discarded with a principal component that is excluded. Hence caution must be exercised while leaving out vectors corresponding to small eigenvalues. If regression based on the original variables $\mathbf{x}$ is preferred, the most important variables can be selected by inspecting the variables that contribute to the first few loadings and avoiding those that provide duplicate information.

To include information about process dynamics, lagged variables can be included in $\mathbf{X}$. The (auto)correlograms of all $\mathbf{x}$ variables should be developed to determine first how many lagged values are relevant for each variable. Then the data matrix should be augmented accordingly and used to determine the principal components that will be used in the regression step.

Nonlinear extensions of PCA have been proposed by using autoassociative neural networks discussed in Section 3.6.1 (an illustrative example is provided in Section 7.7.1) or by using principal curves and surfaces [106, 161].

4.3 PLS Regression Models

Partial least squares (PLS) regression, develops a biased regression model between $\mathbf{X}$ and $\mathbf{Y}$. In the context of chemical process operations, usually $\mathbf{X}$ denotes the process variables and $\mathbf{Y}$ the quality variables. PLS selects latent variables so that variation in $\mathbf{X}$ which is most predictive of the product quality data $\mathbf{Y}$ is extracted. PLS works on the sample covariance matrix $(\mathbf{X}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{X})$ [86, 87, 111, 172, 188, 334, 338]. Measurements of m process variables taken at n different times are arranged into a $(n \times m)$ process data matrix $\mathbf{X}$. The q quality variables are given by the corresponding

$(n \times q)$ matrix $\mathbf{Y}$. PLS modeling works better when the data are fairly symmetrically distributed and have fairly constant 'error variance' [67]. Both $\mathbf{X}$ and $\mathbf{Y}$ data blocks are usually centered and scaled to unit variance because in PLS the influence of a variable on model parameters increases with the variance of the variable. The PLS model can be built by using the nonlinear iterative partial least squares algorithm (NIPALS). The PLS model consists of outer relations ($\mathbf{X}$ and $\mathbf{Y}$ blocks individually) and an inner relation (linking both blocks) (Figure 4.1). The outer relations for the $\mathbf{X}$ and $\mathbf{Y}$ blocks are respectively

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \sum_{i=1}^{a} \mathbf{t}_i \mathbf{p}_i^T + \mathbf{E} \tag{4.13}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F} = \sum_{i=1}^{a} \mathbf{u}_i \mathbf{q}_i^T + \mathbf{F} \tag{4.14}$$

where $\mathbf{E}$ and $\mathbf{F}$ represent the residuals matrices. Linear combinations of $\mathbf{x}$ vectors are calculated from the latent variable scores $\mathbf{t}_i = \mathbf{w}_i^T \mathbf{x}$ and those for the $\mathbf{y}$ vectors from $\mathbf{u}_i = \mathbf{q}_i^T \mathbf{y}$ so that they maximize the covariance between $\mathbf{X}$ and $\mathbf{Y}$ explained at each dimension. $\mathbf{w}_i$ and $\mathbf{q}_i$ are the weight vectors and $\mathbf{p}_i$ are the loading vectors of $\mathbf{X}$. The number of latent variables can be determined by cross-validation [332] or more pragmatic techniques discussed in Section 3.1.

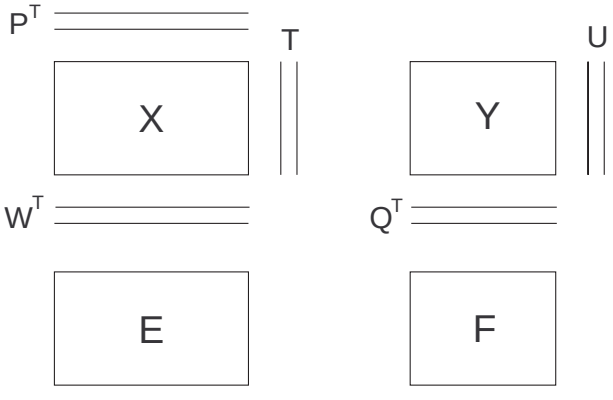


Figure 4.1. The matrix relationships in PLS as shown by [67]. $\mathbf{T}$ and $\mathbf{U}$ are PLS scores matrices of $\mathbf{X}$ and $\mathbf{Y}$ blocks, respectively, $\mathbf{P}$ contains the $\mathbf{X}$ loadings, $\mathbf{W}$ and $\mathbf{Q}$ are weight matrices for $\mathbf{X}$ and $\mathbf{Y}$ blocks, respectively, and $\mathbf{E}$ and $\mathbf{F}$ are residual matrices of $\mathbf{X}$ and $\mathbf{Y}$ blocks.

For the first latent variable, PLS decomposition is started by selecting y_j, an arbitrary column of $\mathbf{Y}$ as the initial estimate for $\mathbf{u}_1$. Usually, the

column of $\mathbf{Y}$ with greatest variance is chosen. Starting in the $\mathbf{X}$ data block, for the first latent variable:

$$\mathbf{w}_1^T = \frac{\mathbf{u}_1^T \mathbf{X}}{\|\mathbf{u}_1^T \mathbf{u}_1\|} \quad , \quad \mathbf{t}_1 = \frac{\mathbf{X}\mathbf{w}_1}{\|\mathbf{w}_1^T \mathbf{w}_1\|} \tag{4.15}$$

In the $\mathbf{Y}$ data block:

$$\mathbf{q}_1^T = \frac{\mathbf{t}_1^T \mathbf{Y}}{\|\mathbf{t}_1^T \mathbf{t}_1\|} \quad , \quad \mathbf{u}_1 = \frac{\mathbf{Y}\mathbf{q}_1}{\|\mathbf{q}_1^T \mathbf{q}_1\|} \tag{4.16}$$

Convergence is checked by comparing $\mathbf{t}_1$ in Eq. 4.15 with the $\mathbf{t}_1$ from the previous iteration. If their difference is smaller than a prespecified threshold, one proceeds to Eq. 4.17 to calculate $\mathbf{X}$ data block loadings $\mathbf{p}_1$ and weights $\mathbf{w}_1$ are rescaled using the converged $\mathbf{u}_1$. Otherwise, $\mathbf{u}_1$ from Eq. 4.16 is used for another iteration. If $\mathbf{Y}$ is univariate, Eqs. 4.16 can be omitted, and $\mathbf{q}_1 = 1$. The loadings of the $\mathbf{X}$ data block and are computed and the scores and weights are rescaled:

$$\mathbf{p}_1^T = \frac{\mathbf{t}_1^T \mathbf{X}}{\|\mathbf{t}_1^T \mathbf{t}_1\|} \quad , \quad \mathbf{p}_{1n} = \frac{\mathbf{p}_{1o}}{\|\mathbf{p}_{1o}\|} \tag{4.17}$$

$$\mathbf{t}_{1n} = \mathbf{t}_{1o}\|\mathbf{p}_{1o}\| \quad , \quad \mathbf{w}_{1n} = \mathbf{w}_{1o}\|\mathbf{p}_{1o}\| \tag{4.18}$$

where the subscript o refers to old and n to new values. The regression coefficient b_i for the inner relation is computed using

$$b_1 = \frac{\mathbf{t}_1^T \mathbf{u}_1}{\|\mathbf{t}_1^T \mathbf{t}_1\|} \tag{4.19}$$

When the scores, weights, and loadings have been determined for a latent variable (at convergence), $\mathbf{X}$- and $\mathbf{Y}$-block matrices are adjusted to exclude the variation explained by that latent variable. Equations 4.20 and 4.21 illustrate the computation of the residuals after the first latent variable and weights have been determined:

$$\mathbf{E}_1 = \mathbf{X} - \mathbf{t}_1 \mathbf{p}_1^T \tag{4.20}$$

$$\mathbf{F}_1 = \mathbf{Y} - b_1 \mathbf{t}_1 \mathbf{q}_1^T \tag{4.21}$$

The entire procedure is repeated for finding the next latent variable and weights starting with Eq. 4.15. The variations in data matrices $\mathbf{X}$ and $\mathbf{Y}$ explained by the earlier latent variables are excluded from $\mathbf{X}$ and $\mathbf{Y}$ by replacing them in the next iteration with their residuals that contain unexplained variation. After the convergence of the first set of latent variables

to their final values, $\mathbf{X}$ and $\mathbf{Y}$ are replaced with the residuals $\mathbf{E}_1$ and $\mathbf{F}_1$, respectively, and all subscripts are incremented by 1.

Several enhancements have been made to the PLS algorithm [48, 93, 169, 184, 336, 333, 339]. Commercial software is available for developing PLS models [328, 269].

Nonlinear PLS Models

To model nonlinear relationships *between* $\mathbf{X}$ and $\mathbf{Y}$, their projections should be nonlinearly related to each other [336]. One alternative is the use of a polynomial function such as

$$\mathbf{u}_i = c_{0i} + c_{1i}\mathbf{t}_i + c_{2i}\mathbf{t}_i^2 + \boldsymbol{\epsilon}_i \tag{4.22}$$

where i represents the model dimension, c_{0i}, c_{1i}, and c_{2i} are constants, and $\boldsymbol{\epsilon}_i$ is a vector of errors (innovations). This quadratic function can be generalized to other nonlinear functions of $\mathbf{t}_i$:

$$\mathbf{u}_i = f(\mathbf{t}_i) + \boldsymbol{\epsilon}_i \tag{4.23}$$

where $f(\cdot)$ may be a polynomial, exponential, or logarithmic function.

Another structure for expressing a nonlinear relationship between $\mathbf{X}$ and $\mathbf{Y}$ is splines [333] or smoothing functions [75]. Splines are piecewise polynomials joined at knots (denoted by z_j) with continuity constraints on the function and all its derivatives except the highest. Splines have good approximation power, high flexibility and smooth appearance as a result of continuity constraints. For example, if cubic splines are used for representing the inner relation:

$$u = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + \sum_{j=1}^{s} b_{j+3}(t - z_j)_+^3 \tag{4.24}$$

where the s knot locations and the model coefficients b_i are the free parameters of the spline function. There are $l + s + 1$ coefficients where l is the order of the polynomial. The term $b_{j+3}(t - z_j)_+^3$ denotes a function with values 0 or $b_{j+3}(t - z_j)^3$ depending on the value of t:

$$b_{j+3}(t - z_j)_+^3 = \begin{cases} b_{j+3}(t - z_j)^3 & : \quad t > z_j \\ 0 & : \quad t < z_j \end{cases} \tag{4.25}$$

The desirable number of knots and degrees of polynomial pieces can be estimated using cross-validation. An initial value for s can be $n/7$ or $\sqrt{(n)}$ for $n > 100$ where n is the number of data points. Quadratic splines can be used for data without inflection points, while cubic splines provide a general approximation for most continuous data. To prevent over-fitting data with

higher-order polynomials, models of lower degree and higher number of knots should be considered for lower prediction errors and improved stability [333]. B splines provide an attractive alternative to quadratic and cubic splines when the number of knots is large [49]. Other nonlinear PLS models that rely on nonlinear inner relations have been proposed [61, 96, 288]. Nonlinear relations within $\mathbf{X}$ or $\mathbf{Y}$ can also be modeled.

4.4 Input-Output Models of Dynamic Processes

Time series models have been popular in many fields ranging from modeling stock prices to climate. They could be cast as a regression problem where the regressor variables are the previous values of the same variable and past values of inputs. They are 'black box' models that describe the relationship of the present value of the output to external variables but do not provide any knowledge about the physical description of the processes they represent. It will be assumed that data are collected using a fixed sampling rate (the sampling time between any two consecutive samples is identical). Time series models relate the current value of the observed variable to

- Past values of the observed variable: **A**uto**r**egressive terms (up to order p) **AR**
- **I**ntegrated AR terms (up to order d) **I**
- Past values of the prediction error or past values of the predicted values: **M**oving **a**verage terms (up to order r) **MA**
- Past values of control signals and known disturbances: E**x**ogenous variables **X**.

The **prediction error** $e(k)$ is the difference between the observed and the predicted values at a specific time $e(k) = y(k) - \hat{y}(k)$. An autoregressive - integrated - moving average model is represented as ARIMA(p, d, r). For example, ARIMA(0,1,1) = IMA(1,1) indicates:

$$\begin{aligned} (y(k) - y(k-1)) &= e(k) + \theta e(k-1) \\ y(k) &= y(k-1) + e(k) + \theta e(k-1) \end{aligned} \tag{4.26}$$

where θ is a parameter for the MA term. Many processes can be approximated by an ARIMA(p, d, r) with $p,\ r \leq 2$ and $d = 0$ or 1. Time series models are often developed by using a data set consisting of individual observations over time.

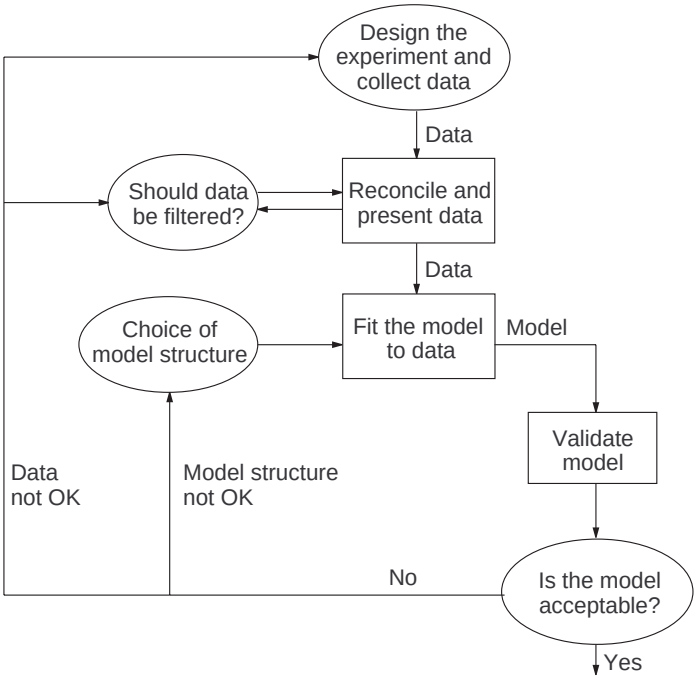


Figure 4.2. Model identification strategy suggested by Ljung [171].

Model development (also called system identification) involves several critical activities including design of experiments and collection of data, data pretreatment, model fitting, model validation and acceptability of the model for its use. A vast literature has been developed over the last 50 years in various aspects of model identification [99, 170, 174, 246, 278]. A schematic diagram in Figure 4.2 where the ovals represent human activities and decision making steps and the rectangles represent computer-based computations and decisions illustrates the links between critical activities.

A short list complements the process outlined in Figure 4.2:

- Design and perform experiments, collect data,
- Plot data and autocorrelation functions, postulate structure of time series model,
- Estimate model parameters:
  - Form one-step-ahead predictor,
  - Compute 'least squares' estimates,

- Validate model.

Model identification is an iterative process. There are several software packages with modules that automate time series model development. When a model is developed to describe data that have stochastic variations, one has to be cautious about the degree of fit. By increasing model complexity (adding extra terms) a better fit can be obtained. But, the model may describe part of the stochastic variation in that particular data which will not occur identically in other data sets. Consequently, although the fit to the 'training' data may be improved, the prediction errors may get worse.

Inputs, outputs and disturbances will be denoted as $\mathbf{u}$, $\mathbf{y}$, and $\mathbf{d}$, respectively. For multivariable processes where $u_1(k)$, $u_2(k)$, $\cdots$, $u_m(k)$ are the m inputs, the input vector $\mathbf{u}(k)$ at time k is written as a column vector. Similarly, the p outputs are defined by a column vector:

$$\mathbf{u}(k) = \begin{pmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{pmatrix}, \quad \mathbf{y}(k) = \begin{pmatrix} y_1(k) \\ \vdots \\ y_p(k) \end{pmatrix} \tag{4.27}$$

Disturbances $\mathbf{d}(k)$ and residuals $\mathbf{e}(k)$ are also represented by column vectors with appropriate dimensions in a similar manner.

A general linear discrete-time model for a variable $y(k)$ can be written as

$$y(k) = \eta(k) + w(k) \tag{4.28}$$

where $w(k)$ is a disturbance term such as measurement noise and $\eta(k)$ is the noise-free output

$$\eta(k) = G(q, \theta)u(k) \tag{4.29}$$

with the rational function $G(q, \theta)$ and input $u(k)$. q is called the shift operator and q^{-1} the backward shift operator such that

$$y(k-1) = q^{-1}y(k) \tag{4.30}$$

and θ represents the model parameters such as f_i and b_i in Eq. 4.31. The function $G(q, \theta)$ relates the inputs to noise-free outputs whose values are not known because the outputs are corrupted by measurement noise. Assume that relevant information for the current value of output $y(k)$ is provided by past values of $y(k)$ with a window length (number of previous sampling times) n_y and past values of $u(k)$ for n_u previous instances. The relationship between these variables is

$$\begin{aligned} &\eta(k) + f_1\eta(k-1) + \cdots + f_{n_y}\eta(k-n_y) \\ &\quad = b_1u(k) + b_2u(k-1) + \cdots + b_{n_u}u(k-(n_u-1)) \end{aligned} \tag{4.31}$$

where $f_i,\ i = 1, 2, \ldots, n_y$ and $b_i,\ i = 1, 2, \ldots, n_u$ are parameters to be determined from data. Writing Eq. 4.31 by using two polynomials in q

$$\begin{aligned}\eta(k)\left(1 + f_1 q^{-1} + \cdots + f_{n_y} q^{-n_y}\right) \\ = u(k)\left(b_1 + b_2 q^{-1} + \cdots + b_{n_u} q^{-(n_u-1)}\right)\end{aligned} \tag{4.32}$$

and defining the polynomials

$$\begin{aligned} F(q) &= (1 + f_1 q^{-1} + \cdots + f_{n_y} q^{-n_y}) \\ B(q) &= (b_1 + b_2 q^{-1} + \cdots + b_{n_u} q^{-(n_u-1)}) \end{aligned} \tag{4.33}$$

Equation 4.31 can be written in a compact form as

$$\eta(k) = G(q, \theta)\ u(k) \quad \text{with} \quad G(q, \theta) = \frac{B(q)}{F(q)} \tag{4.34}$$

If there is a delay in the effects of inputs on the output by n_k sampling times, Eq. 4.31 is modified as

$$\begin{aligned}\eta(k) + f_1\eta(k-1) + \cdots + f_{n_y}\eta(k-n_y) \\ = b_1 u(k-n_k) + b_2 u(k-(n_k+1)) + \cdots + b_{n_u} u(k-(n_u+n_k-1))\end{aligned} \tag{4.35}$$

The disturbance term can be expressed in the same way

$$w(k) = H(q, \theta)\varepsilon(k) \tag{4.36}$$

where $\varepsilon(k)$ is white noise and

$$H(q, \theta) = \frac{C(q)}{D(d)} = \frac{1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d}} \tag{4.37}$$

The model (Eq. 4.28) can be written as

$$y(k) = G(q, \theta)u(k) + H(q, \theta)\varepsilon(k) \tag{4.38}$$

where the parameter vector θ contains the coefficients b_i, c_i, d_i and f_i of the transfer functions $G(q, \theta)$ and $H(q, \theta)$. The model structure is described by five parameters $n_y,\ n_u,\ n_k,\ n_c$, and n_d. Since the model is based on polynomials, its structure is finalized when the parameter values are selected. These parameters and the coefficients are determined by fitting candidate models to data and minimizing some criteria based on reduction of prediction error and parsimony of the model.

The model Eq. 4.38 is known as the *Box-Jenkins (BJ) model* [23]. It has several special cases:

- **Output Error (OE) model.** When the properties of disturbances are not modeled and the noise model $H(q)$ is chosen to be identity ($n_c = 0$ and $n_d = 0$), the noise source $w(k)$ is equal to $e(k)$, the difference (error) between the actual output and the noise-free output.

- **AutoRegressive Moving Average model with eXogenous inputs (ARMAX).** If the same denominator is used for G and H

$$A(q) = F(q) = D(q) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \tag{4.39}$$

  Hence, Eq. 4.38 becomes

$$A(q)y(k) = B(q)u(k) + C(q)\varepsilon(k) \tag{4.40}$$

  where $A(q)y(k)$ is the autoregressive term, $C(q)\varepsilon(k)$ is the moving average of white noise, and $B(q)u(k)$ represents the contribution of external inputs. Use of a common denominator is reasonable if the dominating disturbances enter the process together with the inputs.

- **AutoRegressive model with eXogenous inputs (ARX).** A special case of ARMAX is obtained by letting $C(q) = 1$ ($n_c = 0$).

These models are used for prediction of the output given the values of inputs and outputs in previous sampling times. Since white noise cannot be predicted, its current value $\varepsilon(k)$ is excluded from prediction equations. Predicted values are denoted by a $\hat{}$ (hat) over the variable symbol. To emphasize that predictions are based on a specific parameter set θ, the nomenclature is further extended to $\hat{y}(k \mid \theta)$.

The computation of parameters θ is usually cast as a minimization problem of prediction errors $e(k, \theta) = y(k) - \hat{y}(k \mid \theta)$ for given sets of data over a time period. For n data points

$$\hat{\theta}_n = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} e^2(i, \theta) \tag{4.41}$$

where *argmin* denotes the minimizing argument. This criteria must be modified to prevent over-fitting of data. The objective function in Eq. 4.41 can be reduced by adding more parameters to the model. The resulting model may fit the data used for model development very well including part of the noise in the data. But the over-fitting may cause large prediction errors when new data are used with the model. Several criteria have been proposed to balance model fit and model complexity. Two of them are given here to illustrate how accuracy and parsimony are balanced:

- *Akaike's Information Criterion (AIC)*

$$\min_{l,\theta} \left(1 + \frac{2\,l}{n}\right) \sum_{i=1}^{n} e^2(i,\theta) \tag{4.42}$$

where l is the number of parameters estimated (dimension of θ).

- *Final Prediction Error (FPE)*

$$\min_{l,\theta} \left(\frac{1 + l/n}{1 - l/n}\frac{1}{n}\right) \sum_{i=1}^{n} e^2(i,\theta) \tag{4.43}$$

The merits and limitations of these and other criteria are discussed in the literature [170, 278]

Nonlinear Time Series Models

The linear model structures discussed in this section can handle mild nonlinearities. They can also result from linearization around an operating point. Simple alternatives can be considered for developing linear models with better predictive capabilities than a traditional ARMAX model for nonlinear processes. If the nature of nonlinearity is known, a transformation of the variable can be utilized to improve the linear model. A typical example is the knowledge of the exponential relationship of temperature in reaction rate expressions. Hence, the *log* of temperature with the rate constant can be utilized instead of the actual temperature as a regressor. The second method is to build a recursive linear model. By updating model parameters frequently, mild nonlinearities can be accounted for. The rate of change of the process and the severity of the nonlinearities are critical factors for the success of this approach. Another approach is based on the estimation of nonlinear systems by using multiple linear models [11, 82, 83].

Time series modeling is extended to nonlinear models by using a variety of structures. These models have the capability to describe pathological dynamic behavior and to provide accurate predictions over a wider range of operating conditions compared to linear models. ANNs were introduced in Section 3.6.1. Various other nonlinear model development paradigms include Volterra kernels [185, 315], cascade (block-oriented) models [97, 157, 187, 314], polynomial models, threshold models [297], and models based on spline functions. Polynomial models include bilinear models [72, 201], state-dependent models [233], nonlinear autoregressive moving average models with exogenous inputs (NARMAX) [30, 31, 167, 231], nonlinear polynomial models with exponential [98] and trigonometric functions (NPETM), and multivariate adaptive regression splines (MARS) [76]. A unified nonlinear model development framework is not available, and search

for the appropriate nonlinear structure is part of the model development effort. Use of a nonlinear model development paradigm which is not compatible with the types of nonlinearities that exist in data can have a significant negative effect on model development effort and model accuracy.

A new methodology has been proposed for developing multivariable additive NARX (Nonlinear Autoregressive with eXogenous inputs) models based on subspace modeling concepts [50]. The model structure is similar to that of a Generalized Additive Model (GAM) and is estimated with a nonlinear Canonical Variates Analysis (CVA) algorithm called CANALS. The system is modeled by partitioning the data into two groups of variables. The first is a collection of 'future' outputs, the second is a collection of past input and outputs, and 'future' inputs. Then, future outputs are predicted in terms of past and present inputs and outputs. This approach is similar to linear subspace state-space modeling [159, 211, 307]. The appeal of linear and nonlinear subspace state-space modeling is the ability to develop models with error prediction for a future window of output (window length selected by user) and with a well-established procedure that minimizes trial-and-error and iterations. An illustrative example of such modeling is presented based on a simulated continuous chemical reactor that exhibits multiple steady-states in the outputs for a fixed level of the input [50].

4.5 State-Space Models

State variables are the minimum set of variables that are necessary to describe completely the state of a system. The n state variables of a system at time t is represented as $\mathbf{x}(t) = [x_1(t)\ x_2(t)\ \cdots x_n(t)]^T$. In quantitative terms, given the values of state variables $\mathbf{x}(t)$ at time t_0 and the values of inputs $\mathbf{u}(t)$ (Eq. 4.27) for $t > t_0$, the values of outputs $\mathbf{y}(t)$ can be computed for $t > t_0$. All process variables of interest can be included in a model as state variables while the measured variables can form the set of output variables. This way, the model can be used to compute all process variables based on measured values of output variables and the state-space model.

In this section, classical state-space models are discussed first. They provide a versatile modeling framework that can be linear or nonlinear, continuous- or discrete-time, to describe a wide variety of processes. State variables can be defined based on physical variables, mathematical solution convenience or ordered importance of describing the process. Subspace models are discussed in the second part of this section. They order state variables according to the magnitude of their contributions in explaining the variation in data. State-space models also provide the structure for

developing state estimators where one can estimate corrected values of state variables, given process input and output variables and estimated values of process outputs.

State-space models relate the variation in state variables over time to their values in the immediate past and to inputs with differential or difference equations. Algebraic equations are then used to relate output variables to state variables and inputs at the same time instant. Consider a system of first-order differential equations (Eq. 4.44) describing the change in state variables and a system of output equations (Eq. 4.45) relating the outputs to state variables:

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.44}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.45}$$

If $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are known at time t_0, $\dot{\mathbf{x}}(t_0)$ can be computed using Eq. 4.44. For an infinitesimally small interval δt, one can compute $\mathbf{x}(t_0 + \delta t)$ using Euler's method

$$\mathbf{x}(t_0 + \delta t) = \mathbf{x}(t_0) + \delta t \cdot \mathbf{f}(\mathbf{x}(t_0), \mathbf{u}(t_0)) \tag{4.46}$$

Then, the output $\mathbf{y}(t_0 + \delta t)$ can be computed using $\mathbf{x}(t_0 + \delta t)$ and Eq. 4.45. This computation sequence can be repeated to compute values of $\mathbf{x}(t)$ and $\mathbf{y}(t)$ for $t > t_0$ if the corresponding values of $\mathbf{u}(t)$ are given for subsequent values of time such as $t_0 + 2\delta t, \cdots, t_0 + k\delta t$. The model composed of Eqs. 4.44–4.45 is called the *state-space model*, the vector $\mathbf{x}(t)$, the *state vector*, and its components $x_i(t)$, the *state variables*. The dimension of $\mathbf{x}(t)$, n, is the model order.

State-space models can also be developed for discrete-time systems. Let the current time be denoted as k and the next time instant where input values become available as $k + 1$. The equivalents of Eqs. 4.44–4.45 in discrete time are

$$\mathbf{x}(k + 1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad k = 0, 1, 2, \cdots \tag{4.47}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{u}(k)) \tag{4.48}$$

For the current time k, the state at time $k + 1$ is now computed by the difference equations 4.47–4.48. Usually, the time interval $\delta t = t(k+1) - t(k)$ between the two discrete times is a constant equal to the sampling time.

Linear State-Space Models

The functional relations $\mathbf{f}(\mathbf{x}, \mathbf{u})$ and $\mathbf{h}(\mathbf{x}, \mathbf{u})$ in Eqs. 4.44–4.45 or Eqs. 4.47–4.48 can be restricted to be linear. The linear continuous state-space

model becomes

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{4.49}$$

where the dimensions of the coefficient matrices are $\mathbf{A}_{n\times n}$, $\mathbf{B}_{n\times m}$, $\mathbf{C}_{p\times n}$ and $\mathbf{D}_{p\times m}$, respectively.

The linear discrete-time model for $k = 0, 1, 2, \cdots$ is

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{4.50}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ are related to matrices $\mathbf{F}$ and $\mathbf{G}$ as

$$\mathbf{F} = e^{\mathbf{A}T} \qquad \mathbf{G} = \int_0^T e^{\mathbf{A}\tau}\mathbf{B}d\tau \tag{4.51}$$

where the sampling interval $T = \delta t$ is assumed to be equal for all values of k. Since the coefficient matrices have constant elements, these models are called linear *time-invariant* models. Mild nonlinearities in the process can often be described better by making the matrices in model equations (Eqs. 4.49 and 4.50) time dependent. This is indicated by symbols such as $\mathbf{A}(t)$ or $\mathbf{F}(k)$.

Disturbances

Some disturbances can be measured, but the presence of others is only recognized because of their influence on process and/or output variables. The state-space model needs to be augmented to incorporate the effects of disturbances on state variables and outputs. Following Eq. 4.28, the state-space equation can be written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)) \end{aligned} \tag{4.52}$$

where $\mathbf{w}(t)$ denotes disturbances. It is necessary to describe $\mathbf{w}(t)$ in order to compute how the state variables and outputs behave in presence of disturbances. If the disturbances are known and measured, their description can be appended to the model. For example, the linear state-space model can be written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{W}_1\mathbf{w}_1(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{W}_2\mathbf{w}_2(t) \end{aligned} \tag{4.53}$$

where $\mathbf{w}_1(t)$ and $\mathbf{w}_2(t)$ are disturbances affecting the state variables and outputs, respectively, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are the corresponding coefficient matrices. This model structure can also be used to incorporate modeling uncertainties (represented by $\mathbf{w}_1(t)$) and measurement noise (represented by $\mathbf{w}_2(t)$).

Another alternative is to develop a model for *unknown* disturbances to describe $\mathbf{w}(t)$ as the output from a dynamic system with a known input $\mathbf{u}_w(t)$ that has a simple functional form.

$$\begin{aligned} \dot{\mathbf{x}}_w(t) &= \mathbf{f}_\mathbf{w}(\mathbf{x}_w(t), \mathbf{u}_w(t)) \\ \mathbf{w}(t) &= \mathbf{h}_w(\mathbf{x}_w(t), \mathbf{u}_w(t)) \end{aligned} \tag{4.54}$$

where the subscript w indicates state variables, inputs and functions of the disturbance(s). Typical choices for input forms may be an impulse, white noise or infrequent random step changes. Use of fixed impulse and step changes lead to deterministic models, while white noise or random impulse and step changes yield stochastic models [171]. The disturbance model is appended to the state and output model to build an augmented dynamic model with *known* inputs.

Linearization of Nonlinear Systems

The behavior of a nonlinear process can be approximately described by a linear model in the vicinity of a known operating point developed by linearizing the nonlinear model. The nonlinear terms of the model are expanded by using the linear terms of Taylor series and the equations are written in terms of deviations of process variables (the so-called *deviation variables*) from the operating point to obtain the linear model. The model can then be expressed in state-space form [253].

Consider the state-space model, Eqs. 4.44–4.45, and assume that it has a stable stationary solution (a steady-state) at $\mathbf{x} = \mathbf{x}_{ss}, \mathbf{u} = \mathbf{u}_{ss}$:

$$\mathbf{f}(\mathbf{x}_{ss}, \mathbf{u}_{ss}) = 0 \tag{4.55}$$

If $\mathbf{f}(\mathbf{x}, \mathbf{u})$ has continuous partial derivatives in the neighborhood of the stationary solution $\mathbf{x} = \mathbf{x}_{ss}, \mathbf{u} = \mathbf{u}_{ss}$, then for $\ell = 1, \cdots, n$:

$$\begin{aligned} f_\ell(x,u) &= f_\ell(\mathbf{x}_{ss}, \mathbf{u}_{ss}) + \frac{\partial f_\ell}{\partial x_1}(\mathbf{x}_{ss}, \mathbf{u}_{ss})(x_1 - x_{ss,1}) + \cdots \\ &+ \frac{\partial f_\ell}{\partial x_n}(\mathbf{x}_{ss}, \mathbf{u}_{ss})(x_n - x_{ss,n}) + \frac{\partial f_\ell}{\partial u_1}(\mathbf{x}_{ss}, \mathbf{u}_{ss})(u_1 - u_{ss,1}) \\ &+ \cdots + \frac{\partial f_\ell}{\partial u_m}(\mathbf{x}_{ss}, \mathbf{u}_{ss})(u_m - u_{ss,m}) + r_k(\mathbf{x} - \mathbf{x}_{ss}, \mathbf{u} - \mathbf{u}_{ss}) \end{aligned} \tag{4.56}$$

where $\frac{\partial f_\ell}{\partial x_i}(\mathbf{x}_{ss}, \mathbf{u}_{ss})$ indicates that the partial derivative with respect to x_i is evaluated at $(\mathbf{x}_{ss}, \mathbf{u}_{ss})$ and $\mathbf{r}_k$ denotes the higher order nonlinear terms that are assumed to be negligible. Define Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ that have the partial derivatives in Eq. 4.56 as their elements:

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{pmatrix} \tag{4.57}$$

with the partial derivatives being evaluated at $(\mathbf{x}_{ss}, \mathbf{u}_{ss})$. Using Eq. 4.55, Eq. 4.56 can be written in a compact form as

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}(\mathbf{x} - \mathbf{x}_{ss}) + \mathbf{B}(\mathbf{u} - \mathbf{u}_{ss}) + \mathbf{r}(\mathbf{x} - \mathbf{x}_{ss}, \mathbf{u} - \mathbf{u}_{ss}) \tag{4.58}$$

Neglecting the higher order terms $\mathbf{r}_k(\mathbf{x} - \mathbf{x}_{ss}, \mathbf{u} - \mathbf{u}_{ss})$ and defining the deviation variables

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_{ss}, \qquad \tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_{ss} \ , \tag{4.59}$$

Eq. 4.44 can be written as

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} \tag{4.60}$$

The output equation is developed in a similar manner:

$$\tilde{\mathbf{y}} = \mathbf{C}\tilde{\mathbf{x}} + \mathbf{D}\tilde{\mathbf{u}} \tag{4.61}$$

where the elements of $\mathbf{C}$ and $\mathbf{D}$ are the partial derivatives $\partial h_i / \partial x_j$ with $i = 1, \cdots, p$ and $j = 1, \cdots, n$ and $\partial h_i / \partial u_j$ with $i = 1, \cdots, p$ and $j = 1, \cdots, m$, respectively. Hence, the linearized equations are of the same form as the original state-space equations in Eq. 4.49. Linearization of discrete-time nonlinear models follows the same procedure and yields linear difference equations similar to Eq. 4.50.

Subspace State-Space Models

Subspace state-space models are developed by using techniques that determine the largest directions of variation in the data to build models. Two subspace methods, PCA and PLS have already been introduced in Sections 4.2 and 4.3. Usually, they are used with steady-state data, but they could also be used to develop models for dynamic relations by augmenting the appropriate data matrices with lagged values of the variables. In recent years, dynamic model development techniques that rely on subspace concepts have been proposed [158, 159, 307, 313]. Subspace methods are introduced in this section to develop state-space models for process monitoring and closed-loop control.

Consider a simple state-space model without external inputs $u(k)$

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{H}\boldsymbol{\epsilon}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \boldsymbol{\epsilon}(k) \end{aligned} \tag{4.62}$$

where $\mathbf{x}(k)$ is the state variable vector of dimension n at time k and $\mathbf{y}(k)$ is the observation vector with p output measurements. The stochastic input $\boldsymbol{\epsilon}(k)$ is the serially uncorrelated innovation vector having the same dimension as $\mathbf{y}(k)$ and covariance $\mathbf{E}[\boldsymbol{\epsilon}(k)\boldsymbol{\epsilon}(k+l)^T] = \boldsymbol{\Delta}$ if $l = 0$, and $\mathbf{0}$ otherwise. This representation would be useful for process monitoring activities where 'appropriate' state variables (usually the first few state variables) are used to determine if the process is operating as expected. The statistics used in statistical process monitoring (SPM) charts assume no correlation over time between measurements. If state-space models are developed such that the state variables and residuals are uncorrelated at zero lag, the statistics can be safely applied to these calculated variables instead of measured process outputs. Several techniques, balanced realization [5], PLS realization [210], N4SID [307], and the canonical variate realization [158, 209] can be used for developing these models.

Subspace algorithms generate the process model by successive approximation of the memory or the state variables of the process by determining successively functions of the past that have the most information for predicting the future [159]. In the canonical variate (CV) realization approach, canonical variates analysis (Section 3.2) is used to develop the state-space models [158] where the first state variable contains the largest amount of information about the process dynamics, the second state variable is orthogonal to the first (does not repeat the information explained in the previous state variable) and describes the largest amount of the remaining process variation. The first few significant state variables can often be used to describe the greatest variation in the process. The system order n is determined by inspecting the dominant singular values (SV) of a covariance matrix (the ratio of the specific SV to the sum of all the SVs [5] generated by singular value decomposition (SVD) or an information theoretic approach such as the Akaike Information Criterion (AIC) [158] introduced in Section 4.4.

The data used in subspace state-space model development consists of the time series data of output and input variables. For illustration, assume a case with only output data and the objective is to build a model of the form Eq. 4.62. Since the whole data set is already known, it can be partitioned as past and future with respect to any sampling time. Defining a past data window of length K and a future data window of length J that are shifted from the beginning to the end of the data set, stacked vectors of data are formed. The Hankel matrix (Eq. 4.64) is used to develop subspace

models. It expresses the covariance between future and past stacked vectors of output measurements. Defining the stacked vectors of future ($\mathcal{Y}_{k_J}^{+}$) and past ($\mathcal{Y}_{k_K}^{-}$) data with respect to the current sampling time k as

$$\mathcal{Y}_{k_J}^{+} = \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \\ \vdots \\ \mathbf{y}(k+J-1) \end{bmatrix} \quad \text{and} \quad \mathcal{Y}_{k_K}^{-} = \begin{bmatrix} \mathbf{y}(k-1) \\ \mathbf{y}(k-2) \\ \vdots \\ \mathbf{y}(k-K) \end{bmatrix} \tag{4.63}$$

the Hankel matrix (note that $\mathbf{H}_{KJ}$ is different than the $\mathbf{H}$ matrix in Eq. 4.62) is

$$\mathbf{H}_{KJ} = E\left[\mathcal{Y}_{k_J}^{+}\mathcal{Y}_{k-1_K}^{-^T}\right] = \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{\Lambda}_2 & \cdots & \mathbf{\Lambda}_K \\ \mathbf{\Lambda}_2 & \mathbf{\Lambda}_3 & \cdots & \mathbf{\Lambda}_{K+1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{\Lambda}_J & \mathbf{\Lambda}_{J+1} & \cdots & \mathbf{\Lambda}_{J+K-1} \end{bmatrix} \tag{4.64}$$

where $\mathbf{\Lambda}_q$ is the autocovariance of $\mathbf{y}(k)$'s which are q time periods apart and $E[\cdot]$ denotes the expected value of a stochastic variable. The non-zero singular values of the Hankel matrix determine the order of the system, i.e., the dimension of the state variables vector. The non-zero and dominant singular values of $\mathbf{H}_{JK}$ are chosen by inspection of singular values or metrics such as AIC.

Canonical variate realization requires that covariances of future and past stacked observations be conditioned against any singularities by taking their square roots. The Hankel matrix is scaled by using $\mathbf{R}_K^-$ and $\mathbf{R}_J^+$ defined in Eq. 4.66. The scaled Hankel matrix ($\overline{\mathbf{H}}_{JK}$) and its singular value decomposition is given as

$$\bar{\mathbf{H}}_{JK} = \left[\mathbf{R}_J^+\right]^{-1/2} \mathbf{H}_{JK} \left[\mathbf{R}_K^-\right]^{-1/2} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.65}$$

where

$$\begin{aligned} \left(\mathbf{R}_J^+\right) &= E\left(\mathcal{Y}_{k_J}^{+}\mathcal{Y}_{k_J}^{+^T}\right) \\ \left(\mathbf{R}_K^-\right) &= E\left(\mathcal{Y}_{k-1_K}^{-}\mathcal{Y}_{k-1_K}^{-^T}\right) \end{aligned} \tag{4.66}$$

$\mathbf{U}$ has dimensions $pJ \times a$ and contains the a left eigenvectors of $\bar{\mathbf{H}}_{JK}$. $\mathbf{\Sigma}$ is $a \times a$ and contains the singular values (SV). $\mathbf{V}$ is $Kp \times a$ and contains the a right singular vectors of the decomposition. The SVD matrices in Eq. 4.65 include only the SVs and singular vectors corresponding to the a state variables retained in the model. The full SV matrix $\mathbf{\Sigma}$ has dimension

$Jp \times Kp$ and it contains all SVs in a descending order. If the process noise is small, all SVs smaller than the ath SV are effectively zero and the corresponding state variables are excluded from the model.

The state variables are given as

$$\mathbf{x}_k = \mathbf{\Sigma}^{1/2}\mathbf{V}^T \left(\mathbf{R}_K^-\right)^{-1/2} \mathcal{Y}_{k-1_K}^- \tag{4.67}$$

Once $\mathbf{x}(k)$ (or for the continuous case $\mathbf{x}(t)$) is known, $\mathbf{F}$, $\mathbf{G}$ (or $\mathbf{A}$, $\mathbf{B}$), $\mathbf{C}$ defined in Eq. 4.62, and the stochastic input covariance $\mathbf{\Delta}$ can be constructed [209]. The covariance matrix of the state vector based on CV decomposition $E[\mathbf{x}(k)\mathbf{x}(k)^T] = \mathbf{\Sigma}$ reveals that $\mathbf{x}(k)$ are independent at zero-lag.

The subspace state-space model that includes external inputs is of the form :

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{H}_1\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{H}_2\mathbf{v}(k) \end{aligned} \tag{4.68}$$

where $\mathbf{F}, \mathbf{G}, \mathbf{C}, \mathbf{D}, \mathbf{H}_1$ and $\mathbf{H}_2$ are system matrices, and $\mathbf{w}$ and $\mathbf{v}$ are zero-mean noise vectors that have Normal distribution. The model, Eq. 4.68, can be developed by using CV realization or other methods such as N4SID [307]. When CV realization is used, these models are called canonical variates state-space (CVSS) models.

Extensions to Nonlinear State-Space Models

Various extensions of linear state-space approach have been proposed for developing nonlinear models [227, 274]. An extension of linear CVA for finding nonlinear state-space models was proposed by Larimore [160] where use of alternating conditional expectation (ACE) algorithm [24] was suggested as the nonlinear CVA method. Their examples used linear CVA to model a system by augmenting the linear system with polynomials of past outputs.

Subspace modeling can be cast as a reduced rank regression (RRR) of collections of future outputs on past inputs and outputs after removing the effects of future inputs. CVA performs this regression. In the case of a linear system, an approximate Kalman filter sequence is recovered from this regression. The state-space coefficient matrices are recovered from the state sequence. The nonlinear approach extends this regression to allow for possible nonlinear transformations of the past inputs and outputs, and future inputs and outputs before RRR is performed. The model structure consists of two sub models. The first model is a multivariable dynamic model for a set of latent variables, the second relates these latent variables to outputs. The latent variables are linear combinations of nonlinear transformations of past inputs and outputs. These nonlinear transformations or functions are

found using CANALS [305]. Using nonlinear CVA to fit dynamic models is not new. ACE algorithm was used to visually infer nonlinear functions for single output additive models [29]. DeCicco and Cinar [50] proposed a CANALS-based approach where the nonlinear functions estimated are directly utilized for prediction. Also, a collection of multiple future outputs is considered, which leads to the latent variables model structure. The latent variables are then linked to the outputs using linear projection type nonlinear model structures such as projection pursuit regression [77] or a linear model through least squares regression.

4.6 Summary

Several input-output model development techniques that extract dynamic relations from process data are discussed in this chapter. Methods based on multivariate statistics, systems theory and artificial intelligence are presented. Various multivariate regression techniques are outlined first, to provide the foundation for the discussion on PCA-based regression and its extension to capture dynamic variations in data. Next, PLS regression is introduced, with a similar extension to capture dynamic variations. Then, input-output modeling of dynamic processes with time series models is introduced. The last modeling framework presented is state-space modeling that enables the extraction of arbitrary variables (state variables) that describe the dynamics of the system, while relating the input and output variables. Since most chemical processes are nonlinear, the extensions of these modeling paradigms to the nonlinear frameworks are also introduced. Extensions of PCA and PLS to develop nonlinear models, nonlinear time series modeling techniques and nonlinear state-space modeling techniques are briefly introduced and references are provided for each method.

5

Monitoring of Multivariate Processes

Multivariate SPM (MSPM) methods are gaining acceptance in monitoring continuous processes because multivariate monitoring charts provide more accurate information about the process, give warnings earlier than the signals of univariate charts, and are easy to compute and interpret. MSPM relies on the statistical distance concept which is a generalization of the Student t statistic. First discussed in [226] and later proposed independently in [112] and [179], it provides a useful statistic for representing the deviation of the process from its desired state. If the process has a few variables, the statistical distance statistic T^2 can be computed by using all variables and its charts can be plotted for MSPM [190]. If the number of variables is large and there is significant colinearity among some of them, the PCA or PLS can be used. If the data used for chart development are process variables, MSPM charts are based on principal components (PC). When both process and quality variables are used, and the two blocks of data need to be related as well, the MSPM charts are based on the latent variables (LV) of PLS. Both sets of charts summarize the information about the status of the process by using two statistics, the Hotelling's T^2 and the squared prediction error (SPE). The details are discussed in Sections 5.1 and 5.2. The charts are simply the plots of T^2 or SPE values computed by using the information collected at each sampling time on the time axis. The T^2 chart indicates the distance of the current operation from the desired operation as captured by the PCs or LVs included in the development of the PCA or the PLS model of the process. Since only the first few PCs or LVs that capture most of the variation in the data are used to build the model, the model is a somewhat accurate but incomplete description of the process. The SPE chart captures the magnitude of the error caused by deviations resulting from events that are not described by the model. The T^2 chart indicates a deviation based on process behavior that can be explained by

the model while the SPE chart indicates a significant deviation that can not be explained by the model (the prediction error is inflated). The T^2 and SPE charts must be used as a pair and if either chart indicates a significant deviation from expected operation, the presence of an abnormal process operation must be declared.

If the process is out-of-control, the next step is to find the source cause of the deviation (fault diagnosis) and then to remedy the situation. Fault diagnosis can be conducted by associating process behavior patterns to specific faults or by relating the process variables that have significant deviations from their expected values to various equipment that can cause such deviations as discussed in Chapter 7. If the latter approach is used, univariate charts provide readily the information about process variables with significant deviation. Since multivariate monitoring charts summarize the information from many process variables, the variables that inflate T^2 or SPE statistics must be determined. This is usually done by using contribution plots (Sections 3.4 and 7.4).

To include the information about process dynamics in the models, the data matrix can be augmented with lagged values of data vectors, or model identification techniques such as subspace state-space modeling can be used (Section 4.5). Negiz and Cinar [209] have proposed the use of state variables developed with *canonical variates* based realization to implement SPM to multivariable continuous processes. Another approach is based on the use of Kalman filter residuals [326]. MSPM with dynamic process models is discussed in Section 5.3. The last section (Section 5.4) of the chapter gives a brief survey of other approaches proposed for MSPM.

5.1 SPM Methods Based on PCA

Multivariate SPM methods with PCs can employ various types of monitoring charts. If only a few PCs can describe the process behavior in a satisfactory manner, biplots could be used as visual aids that are easy to interpret. Such biplots can be generated by projecting the data to two dimensional surfaces as PC_1 versus PC_2, PC_1 versus SPE, and PC_2-SPE as illustrated in Figure 5.1.

Data representing normal operation (NO) and various faults are clustered in different regions, providing the opportunity to diagnose source causes as well [153]. Score biplots are used to detect any departure from the in-control region defined by the confidence limits calculated from the reference set. The axis lengths of the confidence ellipsoids in the direction of ith principal component are given by [126]

$$\pm[\mathbf{S}(i,i)F_{a,n-a,\alpha}a(n^2-1)/(n(n-a))]^{1/2} \tag{5.1}$$

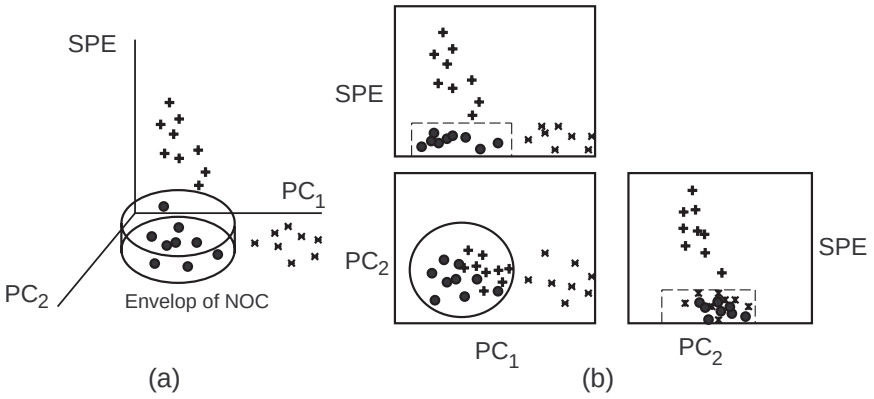


Figure 5.1. The multivariate monitoring space. (a) Three-dimensional representation, (b) Two-dimensional representation.

where $\mathbf{S}$ is the estimated covariance matrix of scores and $F_{a,n-a,\alpha}$ is the F distribution value with a and $n-a$ degrees of freedom in α significance level, n is the number of samples in the reference set, a is the number of PCs retained in the model. Inspection of many biplots becomes inefficient and difficult to interpret when a large number of PCs are needed to describe the process. Monitoring charts based on squared residuals (SPE) and T^2 become more useful. By appending the confidence interval (UCL) to such plots, a multivariate SPM chart as easy to interpret as a Shewhart chart is obtained.

Sometimes, plots of individual PC scores can be used for preliminary analysis of variables that contribute to an out-of-control signal. The control limits for new $\mathbf{t}$ scores under the assumption of Normality at significance level α at any time interval k is given by [100]

$$\pm t_{n-1,\alpha/2} s_{\text{ref}} (1 + 1/n)^{1/2} \tag{5.2}$$

where n and s_{ref} are the number of observations and the estimated standard deviation of the $\mathbf{t}$-score sample at sampling time k (mean is always 0) and $t_{n-1,\alpha/2}$ is the critical value of the Studentized variable with $n-1$ degrees of freedom at significance level $\alpha/2$.

Hotelling's T^2 charts

Hotelling's T^2 plot detects the small shifts and deviations from normal operation defined by the model since it includes contributions of all variables that can become significant faster than the deviation of an individual

variable. The T^2 statistic *based on process variables* at sampling time k is

$$T^2(k) = \left(\mathbf{x}(k) - \bar{\mathbf{x}}\right)^T \mathbf{S}^{-1} \left(\mathbf{x}(k) - \bar{\mathbf{x}}\right) \tag{5.3}$$

where $\bar{\mathbf{x}}$ and $\mathbf{S}$ are estimated from process data. If the individual observation vector $\mathbf{x}(k)$ is independent of $\bar{\mathbf{x}}$ and $\mathbf{S}$, then T^2 follows an F distribution with m and $n-m$ (m measured variables, n sample size) degrees of freedom [190]:

$$T^2 \sim \left[\frac{m(n+1)(n-1)}{n(n-m)}\right] F_{m,n-m} \tag{5.4}$$

If the observation vector $\mathbf{x}$ is not independent of the estimators $\bar{\mathbf{x}}$ and $\mathbf{S}$, but is included in their computation, then T^2 follows a Beta distribution with $m/2$ and $(n-m-1)/2$ degrees of freedom [190]:

$$T^2 \sim \left[\frac{(n-1)^2}{n}\right] B_{m/2,(n-m-1)/2} \tag{5.5}$$

The T^2 charts *based on PCs* use

$$T^2(k) = \mathbf{t}_a^T(k)\mathbf{S}^{-1}\mathbf{t}_a(k) \tag{5.6}$$

and follow an F or a Beta distribution for the same conditions leading to Eqs. 5.4 and 5.5, with a and $n-a$ degrees of freedom for the F distribution, and $a/2$ and $(n-a-1)/2$ degrees of freedom for the Beta distribution, assuming that the data follow a multivariate Normal distribution [121, 120]. As before, a denotes the number of PCs, $\mathbf{t}_a$ is a vector containing the scores from the first a PCs [121] and $\mathbf{S}$ is the $(a \times a)$ estimated covariance matrix, which is diagonal due to the orthogonality of the $\mathbf{t}$ scores [298]. The T^2 based on PCs can also be calculated at each sampling time k as [121]

$$T^2(k) = \sum_{i=1}^{a} \frac{t_i^2(k)}{\lambda_i} = \sum_{i=1}^{a} \frac{t_i^2(k)}{s_i^2} \tag{5.7}$$

where the PC scores t_i have variance λ_i (or estimated variance s_i^2 from the scores of the reference set) which is the ith largest eigenvalue of the covariance matrix $\mathbf{S}$. The term (k) that indicates the explicit dependence on sampling time will be omitted from the T^2 equations in the remainder of the book without loss of generality. If tables for the Beta distribution are not readily available, this distribution can be approximated by using [298]:

$$B_{a/2,(n-a-1)/2,\alpha} = \frac{(a/(n-a-1))F_{a,n-a-1,\alpha}}{1 + (a/(n-a-1))F_{a,n-a-1,\alpha}} \tag{5.8}$$

A variant of T^2 statistic is the D statistic:

$$D(k) = \frac{\mathbf{t}_a^T \mathbf{S}^{-1} \mathbf{t}_a n}{(n-1)^2} \qquad \sim B_{a/2,(n-a-1)/2} \tag{5.9}$$

Squared Prediction Error (SPE) charts

Squared Prediction Error (SPE) charts show deviations from NO based on variations that are not captured by the model. Recall Eq.3.2 that can be rearranged to compute the prediction error (residual) $\mathbf{E}$

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \qquad \mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \tag{5.10}$$

where $\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^T$ denotes the estimates of the data $\mathbf{X}$. The location of the projection of an observation (at sampling time k) on the a-dimensional PC space is given by its score $t_a(k)$. The orthogonal distance of the observation $\mathbf{x}(k)$ from the projection space is the prediction error $\mathbf{e}(k)$ which is squared to compute $SPE(k)$. The $\mathbf{e}(k)$ gives a measure of how close the observation at time k is to the a-dimensional space

$$SPE(k) = \mathbf{e}(k)^T \mathbf{e}(k) \quad = \sum_{j=1}^{m} e_j^2(k) = \sum_{j=1}^{m} [x_j(k) - \hat{x}_j(k)]^2 \tag{5.11}$$

where $\hat{x}_j(k)$ is computed from the PCA model. SPE is also called the Q-statistic.

Statistical limits on the Q-statistic are computed by assuming that the data have a multivariate Normal distribution [120, 121]. The control limits for Q-statistic are given by Jackson and Mudholkar [122] based on Box's [22] formulation (Eq. 5.12) for quadratic forms with significance level of α given in Eqs. 5.12 and 5.13 as

$$Q_\alpha = g\chi^2_{h,\alpha} \tag{5.12}$$

$$Q_\alpha = \theta_1 [1 - \theta_2 h_0 (1 - h_0)/\theta_1^2 + z_\alpha (2\theta_2 h_0^2)^{1/2}/\theta_1]^{1/h_0} \tag{5.13}$$

where χ^2_h is the chi-squared variable with h degrees of freedom and z is the standard normal variable corresponding to the upper $(1-\alpha)$ percentile (z_α has the same sign as h_0). θ values are calculated using the unused eigenvalues of the covariance matrix of observations (eigenvalues that are not retained in the model) as [327]

$$\theta_i = \sum_{j=k+1}^{m} \lambda_j^i, \text{ for } i = 1, 2, \text{ and } 3 \tag{5.14}$$

The other parameters are

$$g = \theta_2/\theta_1 \qquad h = \theta_1^2/\theta_2 \qquad h_0 = 1 - 2\theta_1\theta_3/3\theta_2^2 \tag{5.15}$$

θ_i's can be estimated from the estimated covariance matrix of residuals (residual matrix used in Eq. 5.11) for use in Eq. 5.13 to develop control limits on Q for comparing residuals. A simplified approximation for Q-limits has also been suggested in [68] by rewriting Box's equation (Eq. 5.12) by setting $\theta_2^2 \approx \theta_1\theta_3$

$$Q_\alpha \cong gh[1 - 2/9h + z_\alpha(2/9h)^{1/2}]^3 \tag{5.16}$$

SPE values for new data at time k are calculated using

$$SPE(k) = \sum_{j=1}^{m}(x_j(k) - \hat{x}_j(k))^2 \tag{5.17}$$

These $SPE(k)$ values computed using Eq. 5.17 follow the χ^2 (chi-squared) distribution [22]). This distribution can be well approximated at each time interval using Box's equation in Eq. 5.12 (or its modified version in Eq. 5.16).

Example The performance of univariate and multivariate process monitoring charts are illustrated in Figures 5.2 and 5.3 for the polymerization of vinyl acetate in a CSTR. The simulation uses a model developed by Teymour [291], consisting of four ordinary differential equations for the reactor temperature, solvent volume fraction, monomer volume fraction and the initiator concentration in the reactor, and three differential equations for the molecular weight moments of the reactor. The moments are functions of polymer chain reaction kinetics and probabilities of polymer chain propagation. They are used for calculating various polymer molecular weights, polydispersity and conversion. The 'measured' variables are polydispersity, reactor temperature, conversion and the reactor initiator concentration. The five input variables are the reactor cooling jacket temperature, the initiator concentration in the feed stream, the feed stream temperature, the feed solvent volume fraction and the residence time. The four monitored output variables are assumed to be available via analytical methods at one minute intervals for the physical system. The assumption is valid for the reactor temperature, conversion and initiator concentration, though the polydispersity measurement in a physical system may take up to 30 *min* or more to obtain via analytical monitoring techniques. The manipulated variables are modified by adding random fluctuations to each of the inputs. Disturbances may be added by changing the values of input variables.

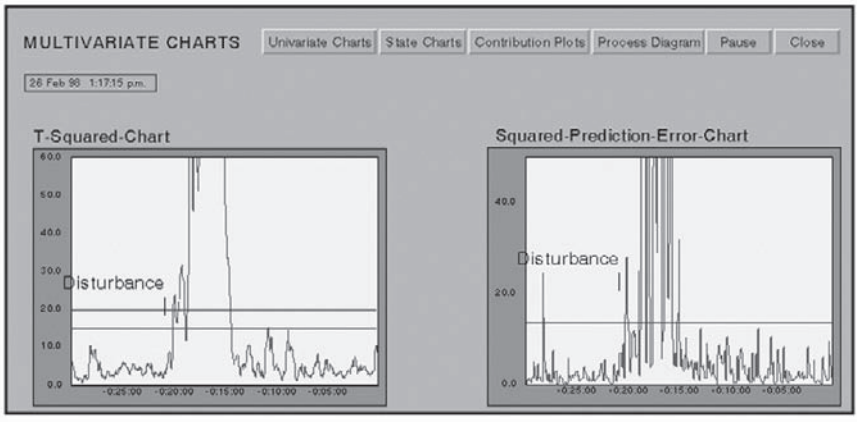


Figure 5.2. T^2 and SPE charts based on PCA for monitoring a continuous polymerization reactor. A 5% increase in reactor feed temperature is introduced for 60 *min* at the time instant indicated by a vertical bar on the plot.

A 5% increase in reactor feed temperature was introduced and maintained for 60 *min* before returning the feed stream to normal operating conditions. The multivariate charts (Figure 5.2) are the first to detect the disturbance to the reactor operation. The T^2 statistic exceeds the 99% confidence interval 25 *min* after the disturbance was introduced, and the SPE statistic 20 *min* after the disturbance, a few minutes earlier than the T^2 chart. The initiator concentration in the reactor exceeds the statistical limits of the Shewhart chart (Figure 5.3) after 35 *min*. Reactor temperature and conversion readings exceed the statistical limits after approximately 40 *min* and the polydispersity measurement exceeds the univariate limit after 44 *min*.

5.2 SPM Methods Based on PLS

Large amounts of process data, such as temperatures and flow rates, are collected at high frequency by process data collection systems. Information on product quality variables is collected less frequently since these measurements are expensive. Although it is possible to measure some quality variables on-line by means of sophisticated devices, measurements are generally made off-line in the quality control laboratory and often involve time lags between data collection and receiving analysis results. Process data

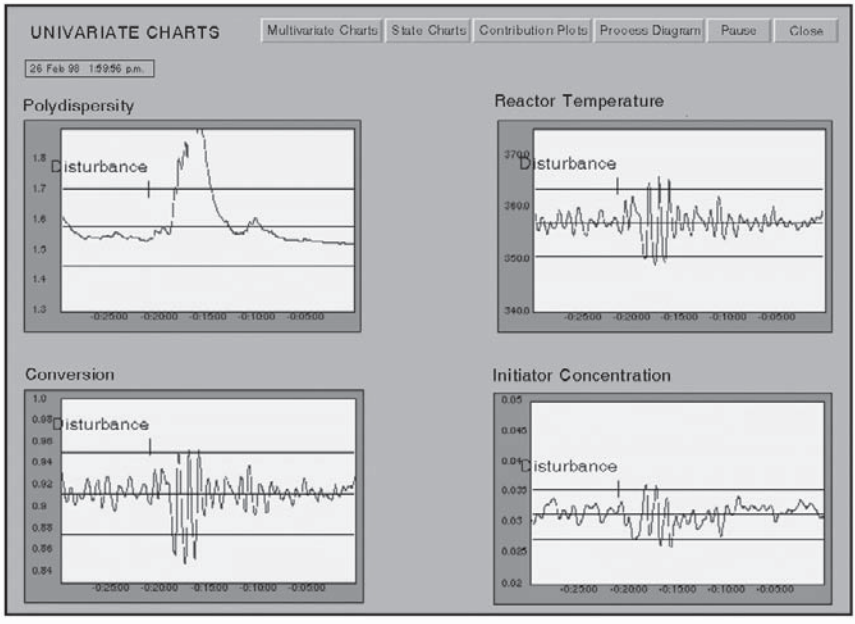


Figure 5.3. Shewhart charts for monitoring a continuous polymerization reactor. A 5% increase in reactor feed temperature is introduced for 60 *min*.

contain valuable information about both the quality of the product and the performance of the process operation. PLS models provide the quantitative relations for estimating product quality from process data. They can also be used used to quickly detect process upsets and unexpected behavior.

Cross correlations and colinearity among process variables severely limit the use of traditional linear regression techniques. PLS, as a projection method, offers a suitable solution for modeling such data.

The first step in the development of a PLS model is to group the process variables as $\mathbf{X}$ and the product quality variables as $\mathbf{Y}$ (Figure 5.4). This selection is dependent on the measurements available and the objectives of monitoring. The reference set used to develop the multivariate monitoring chart will determine the variations considered to be part of normal operation and ideally includes all variations leading to desired process performance. If routine variations in the reference set are too small, the resulting model used for process monitoring will cause frequent alarms, and if it includes a data set that contains large variations, the sensitivity for detecting abnormal operation will be poor. The reference data set selected should

include the range of process variables that yield desired product quality. If the PLS model is developed for monitoring certain process conditions, the reference data set should include data collected under these conditions.

Since PLS technique is sensitive to outliers and scaling, outliers should be removed and data should be scaled prior to modeling. After data pretreatment, the number of latent variables (PLS dimensions) to be retained in the model is determined. Cumulative prediction sum of squares (CUMPRESS) versus the number of latent variables or prediction sum of squares (PRESS) versus the number of latent variables plots are used for this purpose. It is usually enough to consider the first few PLS dimensions for monitoring activities, while more PLS dimensions are needed for prediction in order to improve the accuracy of predictions.

The squared prediction error (SPE) can be calculated for the $\mathbf{X}$ and the $\mathbf{Y}$ block models

$$SPE_{\mathbf{X},k} = \sum_{j=1}^{m} (x_{kj} - \hat{x}_{kj})^2 \tag{5.18}$$

$$SPE_{\mathbf{Y},k} = \sum_{j=1}^{q} (y_{kj} - \hat{y}_{kj})^2 \tag{5.19}$$

where $\hat{x}$ and $\hat{y}$ are predicted observations in $\mathbf{X}$ and $\mathbf{Y}$ using the PLS model, respectively, k and j are the indexes for observations and variables in $\mathbf{X}$ or $\mathbf{Y}$, respectively.

$\hat{x}_{kj}$ and $\hat{y}_{kj}$ in Eqs. 5.18 and 5.19 are calculated for new observations as follows:

$$t_{i,\text{new}} = \sum_{j=1}^{m} x_{\text{new},j} w_{i,j} \tag{5.20}$$

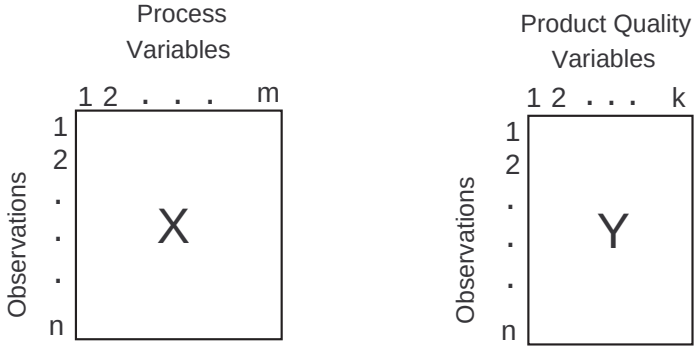


Figure 5.4. Arrangement of data in PLS for SPM as suggested in [41].

$$\hat{x}_{\text{new},j} = \sum_{i=1}^{a} t_{i,\text{new}} p_{i,j} \tag{5.21}$$

$$\hat{y}_{\text{new},j} = \hat{x}_{\text{new},j} b \tag{5.22}$$

where $w_{i,j}$ denotes the weights, $p_{i,j}$ the loadings for the $\mathbf{X}$ block (process variables) of the PLS model, $t_{i,\text{new}}$ the scores of new observations, and b the regression coefficient for the inner relations.

Multivariate monitoring charts based on Hotelling's statistic (T^2) and squared prediction errors (SPE_X and SPE_Y) are constructed using the PLS models. Hotelling's T^2 statistic for a new independent $\mathbf{t}$ vector is [298]

$$T^2 = \mathbf{t}_{\text{new}}^T \mathbf{S}^{-1} \mathbf{t}_{\text{new}} \quad \sim \frac{a(n^2-1)}{n(n-a)} F_{a,n-a} \tag{5.23}$$

where $\mathbf{S}$ is the estimated covariance matrix of PLS model scores, a the number of latent variables retained in the model and $F_{a,n-a}$ the F distribution value. The control limits on SPE charts can be calculated by an approximation of the χ^2 distribution given as $SPE_\alpha = g\chi^2_{h\alpha}$ [22]. This equation is well approximated as [68, 122, 218]

$$SPE_\alpha \cong gh \left[1 - \frac{2}{9h} + z_\alpha \left(\frac{2}{9h} \right)^{1/2} \right]^3 \tag{5.24}$$

where g is a weighting factor and h degrees of freedom for the χ^2 distribution. These can be approximated as $g = v/(2m)$ and $h = 2m^2/v$, where v is the variance and m the mean of the SPE values from the PLS model.

Biplots of scores ($\mathbf{t}_i$ vs $\mathbf{t}_{i+1}$, for $i = 1, \cdots, a$) can also be developed. The control limits at significance level $\alpha/2$ for a new independent t score under the assumption of Normality at any sampling time are

$$\pm t_{n-1,\alpha/2} s_{\text{est}} (1 + 1/n)^{1/2} \tag{5.25}$$

where n, s_{est} are the number of observations and the estimated standard deviation of the score sample at the chosen time interval and $t_{n-1,\alpha/2}$ is the critical value of the Student's t test with $n-1$ degrees of freedom at significance level $\alpha/2$ [100, 218]. The use of PLS models will be illustrated in Section 8.1 for sensor failure detection.

5.3 SPM Using Dynamic Process Models

MSPM techniques rely on the model of the process. If the process has significant dynamic variations, state-space and subspace state-space models

can represent the dynamics of the process. The subspace models can be developed by using the methodology described in Section 4.5. Commercial and open-source software are available for developing subspace state space models using canonical variate (CV) realization and N4SID approaches. MSPM techniques use the state-variables $\mathbf{x}(k)$ of the subspace state-space models to generate the T^2 values and the residuals $\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$ between the measured and estimated values of the model outputs to generate the SPE values at time k. The control limits of the charts are identical to those given in Section 5.1.

The residuals are used in monitoring with the normalized SPE chart (SPE_N) in this example. At time k, $SPE_N(k)$ is

$$SPE_N(k) = (\mathbf{e}(k) - \overline{\mathbf{e}})^T \mathbf{\Sigma}_e^{-1} (\mathbf{e}(k) - \overline{\mathbf{e}}) \tag{5.26}$$

where $\mathbf{\Sigma}_e$ and $\overline{\mathbf{e}}$ are the covariance matrix and the mean vector of residuals respectively, which are determined for in-control data. SPE given here is called as *normalized* since the $SPE(k)$ values are scaled with their in-control mean and variance. SPE_N is distributed as

$$SPE_N \sim \frac{m\ (n^2 - 1)}{n(n - m)} F_{m,n-m} \tag{5.27}$$

The in-control residual mean vector $\mathbf{e}$ is almost zero and in-control residual covariance matrix $\mathbf{\Sigma}_e$ is diagonal.

Example The performance of MSPM charts based on CV state-space models is illustrated by monitoring a high-temperature short-time (HTST) pasteurization system [143, 211]. Pasteurization is a heat treatment process of foods to secure destruction of pathogenic bacteria without markedly affecting the physical and chemical properties of the end product. In HTST pasteurization of milk, the standard time-temperature combination is 72^oC (161^oF) with a residence (holding) time of 15 *sec* before the pasteurized milk is cooled. The process (Figure 5.5) consists of a plate heat exchanger, a centrifugal pump, a flow diversion valve, a boiler and a homogenizer. There are two regulatory valves, the steam injection valve to the boiler and the hot water flow valve in preheater section.

The incoming raw product passing through the regenerator section goes first to the preheater section where it exchanges heat with hot water for controlling raw product temperature entering the homogenizer. After the homogenizer, raw milk flows to the main heat exchanger and follows the same procedure as in the generic pasteurization plant.

The primary source of heat is hot water. The hot water is heated by direct steam injection in the hot water heater. Three PID controllers are used to control product temperature. The first control loop regulates the

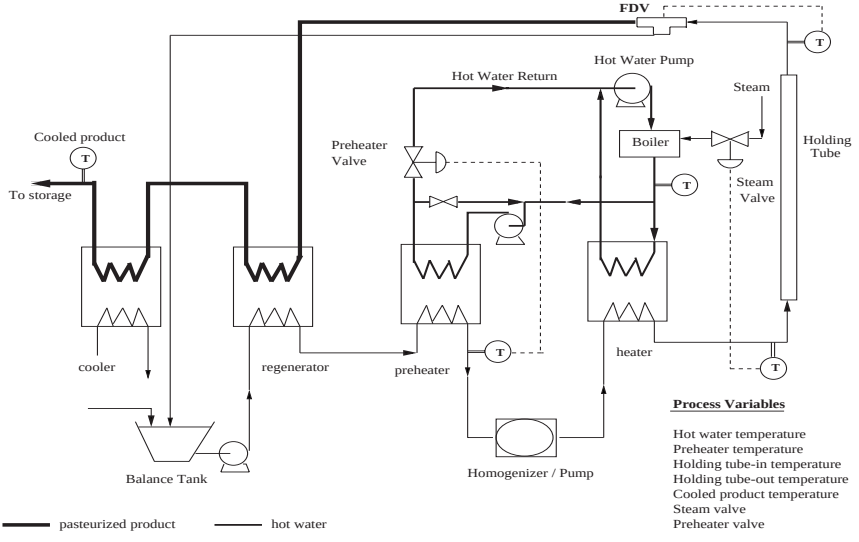


Figure 5.5. Diagram of the NCFST pilot HTST pasteurization plant. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

raw product temperature leaving the preheater. The second loop controls product temperature entering the holding tube. The last loop controls the temperature of the pasteurized product leaving the cooler. The raw product temperature at the exit of the preheater is controlled by manipulating the flow of hot water through the preheat heater exchanger. The product temperature at the holding tube inlet is controlled by manipulating the steam flow rate into the hot water heat exchanger. The cooler product temperature is controlled by manipulating the flow rate of cold water through the cooler heat exchanger. The flow diversion valve is controlled by pasteurized milk temperature at the holding tube exit. The measured variables are hot water, holding tube inlet, holding tube outlet, and preheater exit temperatures and the steam valve and preheater valve signals.

The variables used in process modeling and fault diagnosis implementation are four temperature measurements (^oC) and two PID controller

outputs (mA). Hot water temperature, holding tube inlet temperature of pasteurized product, holding tube outlet temperature of pasteurized product and preheater outlet temperature of raw product are the output variables of the process. PID controller of the steam valve regulates the holding tube inlet temperature of product, and PID controller of the preheater hot water valve regulates the preheater outlet temperature of raw product.

Data for model formation are collected under open-loop conditions by exciting the process with pseudo-random binary sequence (PRBS) signals that are sent to the control valves. PRBS allows the control valves (steam valve and preheater hot water valve) to switch between two different signal levels depending on a switching probability, P. Two different PRBS series are used for two actuators. First a series of random numbers $\mathbf{r}$ with a uniform distribution are generated. The signal that is sent to process at time k changes depending on the value of $r(k)$ and P. At time k, the signal $S(k)$ stays at the same level as the previous signal $S(k-1)$ if the value of the random number $r(k)$ is less than the switching probability P. Otherwise $S(k)$ will switch to the other level. To collect open loop data of the process, uniformly distributed, different PRBS (5000×1) were generated with P of 0.94 for each actuator. For steam control valve and preheater control valve, the actuator command generated by the above procedure changed between $6 - 11$ mA and $10 - 15$ mA, respectively. The number of state variables in the state-space model used for process monitoring was chosen as 12. The design parameters, which are backward and forward time windows to build the time-lagged data matrix, were chosen as 15 in model determination.

Three types of faults were implemented: sensor faults, actuator faults and combination faults (single sensor-single actuator faults and multiple sensors-single actuator faults) [143]. Experiments were conducted with different fault magnitudes and duration. Actuator faults to the steam valve are used for illustration. The faults are caused by keeping the controllers active and sending a constant signal to the actuators instead of the controller signal for a specific time period. The controller output is put in a data file. Therefore, the system will not know about the actuator abnormality until controller notices deviations in the controlled variables.

Table 5.1 shows the time and duration of the faults and the detection times of T^2 and SPE charts. The T^2 chart signals the abnormal situation for last two failures that have larger magnitudes. SPE_N chart shows all the failures (Figure 5.6). The arrows and numbers in the figures indicate the faults (first column of Table 5.1) and their time of occurrence.

Table 5.1. Steam valve fault: Times and magnitudes of faults, performance of SPM charts in terms of sampling times elapsed before detection (NA: **N**o **A**larm generated).

Fault	Time (sec)	Valve Signal (mA)	Duration (sec)	T^2	SPE_N
1	301	7.0	20	7	7
2	521	7.5	20	NA	25
3	741	11.0	20	40	11
4	961	12.0	20	19	1

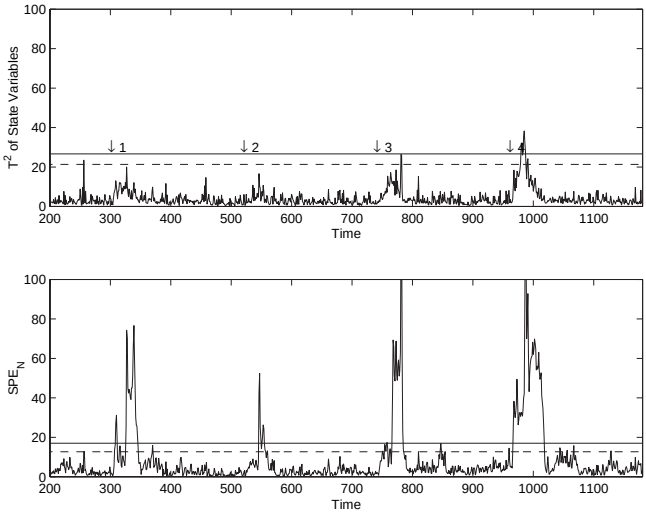


Figure 5.6. Steam valve fault: T^2 of state Variables and SPE_N chart with 99% (-) and 95% (- -) confidence limits. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

5.4 Other MSPM Techniques

MSPM techniques based on PCA and PLS are gaining popularity and replacing traditional process performance assessment activities that rely on univariate tools such as Shewhart charts. PCA techniques have been used to monitor an LDPE reactor operation [145], high speed polyester film production [320], Tennessee Eastman simulated process [168, 242, 342] and sheet forming processes [249]. Several new concepts introduced in the 1990s are being extended. MSPM of strongly autocorrelated processes based on

state equations derived by using subspace model identification [211] is used successfully in abnormal situation management [132, 133] and larger-scale simulation problems such as the Tennessee Eastman process [168]. Multi-scale PCA by using wavelet decomposition (Section 6.1) has been proposed for monitoring processes with multi-scale data [8, 344].

Independent component analysis (ICA) is proposed as an alternative to PCA for MSPM. Various studies indicate that ICA-based MSPM tools are more successful for non-Gaussian data [162]. Several papers have been published recently to illustrate the strengths and limitations of ICA for MSPM [137, 138, 163, 164].

MSPM methods have been extended to processes that operate in multiple modes. Multi-group and multi-block PLS have been proposed to monitor with a single model a number of similar products manufactured across different unit processes [189]. Dynamic PCA (DPCA) is used for a two-step clustering method for process states in agile chemical plants [280]. Process states are first classified into modes corresponding to transitions and steady-states. DPCA is then used to compare different modes and transitions and to cluster them using similarity measures. Support vector machines (SVM) have been used for simultaneous fault detection and operation mode identification in multi-mode operations [40]. SVM is used for classification together with an entropy-based variable selection method to discriminate between data clusters corresponding to multiple operational modes and abnormal data corresponding process faults. Angle-based classification and fault diagnosis techniques introduced by Raich and Cinar [243] have been extended to monitor processes with multiple operating modes [348].

When a process consists of several units that should be monitored individually along with the whole process, multi-block techniques [322] such as consensus PCA (CPCA) [335], hierarchical PCA (HPCA) [337], multi-block PLS (MBPLS) [319, 335] or hierarchical PLS (HPLS) [335] can be used. In multi-block algorithms, the descriptor variables (for PCA) and the response variables (for PLS) are divided into several blocks so as to obtain local information (block scores) as well as global information from process data. The CPCA and MBPLS algorithms normalize the block loadings and super loadings, while the HPCA and HPLS algorithms normalize the block scores and super scores [237, 322].

Moving-window PCA (MWPCA) has been proposed to monitor time-varying processes where both the PCA model and the statistical confidence intervals of the monitoring charts are adapted [316]. MWPCA provides recursive adaptation within the moving window to adapt the mean and variance of process variables, the correlation matrix, and the PCA model by recomputing the decomposition. MWPCA is compared to recursive

PCA and its performance is illustrated using the fluid catalytic cracking unit (FCCU) challenge problem [193].

5.5 Summary

Multivariate SPM (MSPM) methods based on PCS, PLS, and state-space models are presented in this chapter. Multivariate monitoring charts provide more accurate information about the process and give warnings earlier than the signals of univariate charts. MSPM relies on the statistical distance concept T^2 that can be computed by using all variables if the process has a few variables. MSPM techniques based on PCA (Section 5.1) or PLS (Section 5.2) are preferred if the process has a large number of variables and there is significant colinearity among some of them. PCA is used if only process variables are used in the development of MSPM charts. When both process and quality variables are used, and the two blocks of data need to be related as well, the MSPM charts are based on the latent variables of PLS. In both cases, the status of the process is summarized by using two statistics, the Hotelling's T^2 and the squared prediction error (SPE). The charts plot T^2 or SPE values computed by using the information collected at each sampling time on the time axis. The T^2 chart indicates the distance of the current operation from the desired operation. The SPE chart captures the magnitude of the error caused by deviations resulting from events that are not described by the PCA or PLS-based model. The T^2 and SPE charts are used together, and if either chart indicates a significant deviation from the expected operation, the presence of an abnormality in process operation must be declared.

To include the information about process dynamics in the models, the data matrix can be augmented with lagged values of data vectors, or model identification techniques such as subspace state-space modeling can be used (Section 5.3). Other approaches proposed for MSPM are summarized in Section 5.4).

If process monitoring detects an abnormality in process operation, the next steps are to find the source cause of the deviation (fault diagnosis) and then to remedy the situation. Fault diagnosis is achieved by associating process behavior patterns to specific faults (Chapter 7) or by relating the process variables that have significant deviations from their expected values to various equipment that can cause such deviations. If the latter approach is used, the variables that inflate T^2 or SPE statistics must be determined since multivariate monitoring charts summarize the information from many process variables. This is usually done by using contribution plots (Sections 3.4 and 7.4).

6

Characterization of Process Signals

Interpretation of a process signal solely based on its temporal evolution is often risky. Subtle changes in signal characteristics and key transitions may be missed leading to incorrect assessment of process status. In some cases, one can attempt to extract more information from a process signal by transforming it into a domain that might help to accentuate key features of the signal. One such approach is the use of the Fourier transform (FT) to determine the frequency content of a signal. Yet, it would also be interesting to understand if the frequency characteristics of the signal may be changing in time. In the next section (Section 6.1), wavelet transform (WT) will be briefly introduced to show how both frequency and temporal features of a signal can be localized. This will be followed in Section 6.2 by a discussion on signal denoising based on wavelet transforms and a hybrid strategy that can also deal with outliers that are often present in real-world signals. The subsequent sections will introduce methods that help model process signals for later use in monitoring applications. First, in Section 6.3, triangular episodes will be discussed as a means of obtaining a symbolic representation from an otherwise numerical time series data. A more elaborate strategy based on a doubly stochastic model, namely the hidden Markov models (HMMs), will be introduced in Section 6.4.2 and the chapter will conclude in Section 6.5 with the modeling of wavelet coefficients using the HMM paving the way for a trend analysis methodology to be introduced in Chapter 7.

6.1 Wavelets

In recent years, wavelet transform (WT) has been developed as a novel, to use the term somewhat loosely, 'extension' of the traditional Fourier transform (FT) as a means of capturing transitions in the frequency content

of a signal in time. In signal processing, wavelets are used as a major tool to analyze non-stationary signals [44, 182] and also well studied for signal denoising and compression purposes [56, 90, 229]. In process applications, following the first studies summarized in the book by Motard and Joseph [205], there emerged also fine examples of wavelet applications in process monitoring [8, 9], denoising [59] and compression [10, 200]. To follow the historical development of WT, it is best to start with a brief review of FT and its early extensions. Then, continuous and discrete WT are introduced separately and illustrated by examples.

6.1.1 Fourier Transform

An important feature of a process signal is its periodicity, or, in other words, its frequency content. Fourier theory indicates that it is possible to separate individual frequency components from stationary signals (i.e., stochastic signals whose statistical characteristics do not change over time) and make a transformation from the amplitude-time domain to the amplitude-frequency domain. This transformation is known as the *Fourier Transform* (FT). The Fourier transform of a continuous stationary signal $z(t)$ for a given frequency ω in radians is defined as,

$$Z(\omega) = \int_{-\infty}^{+\infty} z(t)e^{-i\omega t}dt \tag{6.1}$$

Often, one uses the frequency representation of FT as follows:

$$Z(f) = \int_{-\infty}^{+\infty} z(t)e^{-i2\pi ft}dt \tag{6.2}$$

In effect, FT expresses a periodic signal in terms of sinusoidal basis functions. The spectrum obtained by the transformation shows the overall strength with which any frequency ω is contained in $z(t)$. When applied to aperiodic signals, it is required that the signal has finite energy, i.e.,

$$\int_{-\infty}^{\infty} |z(t)|^2 dt < \infty \tag{6.3}$$

For periodic signals, the basis functions (exponential building blocks) can be related harmonically, while for aperiodic signals, one can only say that they are infinitesimally close in frequency.

In practical applications, a process signal is sampled to yield a sequence of discrete values, i.e., $z(t) \rightarrow z(k)$ where $k = 0, 1, ..., n-1$. Thus, the discrete signal can be expressed as a sequence, $\{z(k) = z_0, z_1, ..., z_{n-1}\}$.

For this finite sequence, the discrete Fourier transform (DFT) is defined as

$$Z_k = \sum_{j=0}^{n-1} z_j e^{-2\pi ijk/n}, \qquad k = 0, 1, ..., n-1 \tag{6.4}$$

It is noted that Z_k is associated with the frequency $f_k \equiv k/n$. There are efficient algorithms for computing Z_k using the fast Fourier transform (FFT) [220].

When evaluated using real-valued inputs (data), FFT gives outputs (spectrum) whose positive and negative frequencies are redundant. It turns out that they are complex conjugates of each other, meaning that their real parts are equal and their imaginary parts are negatives of each other. Note that the original signal sequence z_j can be reconstructed from its DFT by,

$$z_j = \frac{1}{n} \sum_{k=0}^{n-1} Z_k e^{2\pi ijk/n} \qquad j = 0, 1, ..., n-1 \tag{6.5}$$

Example Two signals are constructed using pure sinusoids with three different frequencies. The first signal is defined as,

$$f_1(t) = sin(2.05 \cdot 2\pi t) + sin(0.1 \cdot 2\pi t) + sin(1.5 \cdot 2\pi t) \tag{6.6}$$

This is a stationary signal because all of the frequencies are present throughout the duration of the signal. The second signal is a non-stationary signal that contains the same frequencies but at different time intervals, leading to discontinuities at the points of transition:

$$f_2(t) = \begin{cases} sin(2.05 \cdot 2\pi t) & 0 \le t \le 20 \\ sin(0.1 \cdot 2\pi t) & 20 \le t \le 35 \\ sin(1.5 \cdot 2\pi t) & 35 \le t \le 50 \end{cases} \tag{6.7}$$

Figure 6.1 displays these signals along with their FT. One can easily observe that while the two signals are vastly different, their FT is quite similar, underscoring the inappropriateness of using FT for non-stationary signals.

The idea of preserving temporal information while obtaining the frequency spectra of any function led to the extension of the standard FT. This extension of FT is known as the Gabor transform or the short-time Fourier transform (STFT) in signal processing [235]. The purpose is to transform non-stationary signals, so that time and frequency information is preserved. Since a non-stationary signal can be viewed as composed of segments of stationary signals of certain length, the idea here is to decompose the non-stationary signal into small segments and perform FT of each

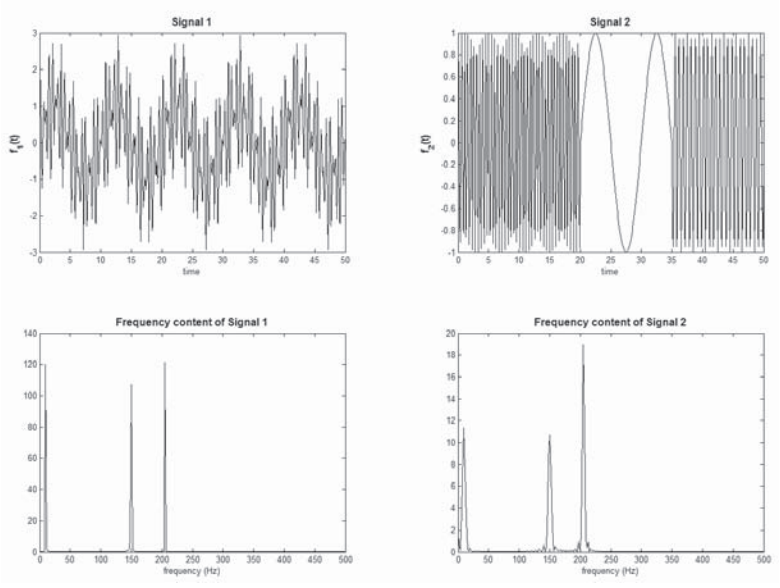


Figure 6.1. Two signals and their corresponding FT.

segment. For this purpose, a window function needs to be chosen. Ideally, the width of this window must be equal to the portion of the signal where it does not violate stationarity conditions. The STFT can be defined as follows:

$$Z(\tau, f) = \int_{-\infty}^{\infty} z(t)w(t-\tau)e^{-i2\pi ft}dt \tag{6.8}$$

where $w(t-\tau)$ is a window function centered at τ (see [182] for various window functions and their comparative merits). It can be seen that STFT is a convolution of the signal with the window function. In STFT, the narrower the window, the better the time resolution, but the poorer the frequency resolution, and vice versa. This problem stems from the Heisenberg's uncertainty principle which states that one cannot know the exact time-frequency representation of a signal as no signal has finite time duration and finite frequency bandwidth simultaneously. One can only know the time intervals in which a certain band of frequencies exist, which turns out to be a resolution problem. The FT does not have any resolution issues in the frequency domain by the fact that window used in its kernel is the $e^{-2\pi ift}$ function which lasts for all time. In STFT, the window is of finite length, thus the frequency resolution becomes poorer. In other words, with a narrow window selection, STFT provides good time resolution but

poor frequency resolution, and with a wide window, good frequency resolution but poor time resolution is achieved. Naturally, the main drawback of STFT is that once the window size is selected, it is fixed for all frequencies.

Example Figure 6.2 depicts the STFT of the two signals introduced in the previous example. A 21-point Hanning window function is used for this example, and the window functions overlapped half of the previous window when translated. One can now observe the difference in the signal characteristics as the non-stationary signal yields a STFT that sharply delineates the transition period.

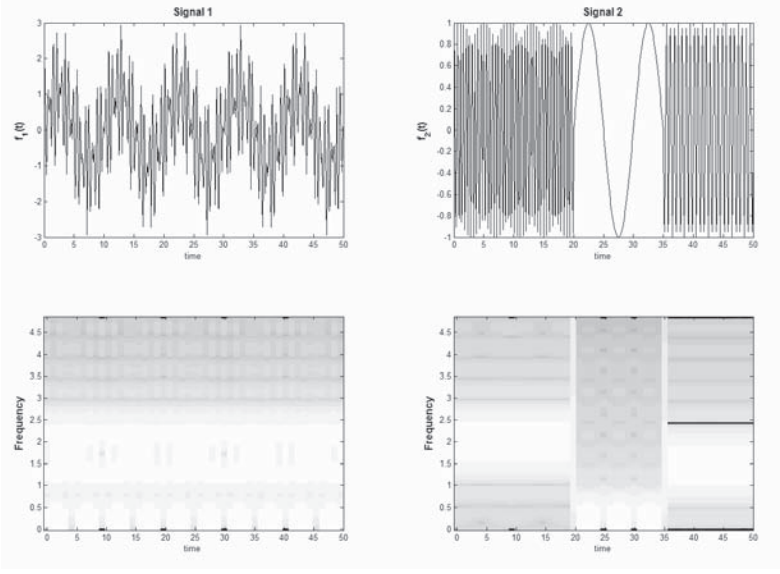


Figure 6.2. Two signals and their corresponding STFT.

6.1.2 Continuous Wavelet Transform

Historically, the first wavelet function is attributed to Haar [95] when he replaced the sinusoidal basis functions of FT with an orthonormal function, $\psi(t)$, given as,

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & t \notin [0, 1] \end{cases} \tag{6.9}$$

The most important difference between the Haar basis and the sinusoids

is that $e^{-j\omega t}$ has infinite support, which means that it stretches out to infinity, while the Haar basis has compact support since it only has nonzero values between 0 and 1.

The *Wavelet Transform* (WT) was formally introduced in the late 1970s by Morlet [45], a geophysicist with Elf-Acquitane in France. Morlet's company was searching for oil by sending impulses into the ground and analyzing their echoes. As sound waves travel through different materials at different speeds, geologists can infer what kind of material lies under the surface. As Morlet analyzed these signals with FT and STFT, he was not satisfied with the constant window sizes in STFT in providing him with the much needed frequency resolution. Morlet proposed a new transform function by taking a cosine wave windowed by a Gaussian (Figure 6.3):

$$\psi(t) = Cexp\left(-\frac{t^2}{2}\right) cos(5t) \tag{6.10}$$

By compressing this function in time, Morlet was able to obtain a higher frequency resolution and spread it out to obtain a lower frequency resolution. To localize time, he shifted these waves in time. He called his transform the 'wavelets of constant shape' and today, after a substantial number of studies in its properties, the transform is simply referred to as the Wavelet transform. The Morlet wavelet is defined by two parameters: the amount of compression, called the scale, and the location in time.

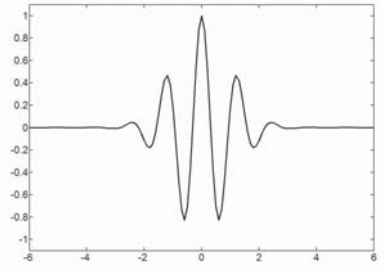


Figure 6.3. The Morlet wavelet.

There are several families of wavelets, proposed by different authors. Those developed by Daubechies [46] are extensively used in engineering applications. Wavelets from these families are orthogonal and compactly supported, they possess different degrees of smoothness and have the maximum number of vanishing moments for a given smoothness. In particular, a function $f(t)$ has ε vanishing moments if

$$\int t^n f(t)dt = 0, \qquad n = 0, 1, ..., \varepsilon - 1 \tag{6.11}$$

These properties are desirable when representing signals through a wavelet series. In addition [44],

- The function should decrease quickly towards 0 as its argument approaches infinity, and
- The function is null outside a segment of the Real line, R.

The equation for the continuous wavelet transform (CWT) can be expressed as,

$$CWT(s,u) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} z(t)\psi(s,u)dt \tag{6.12}$$

where $CWT(s,u)$ are the wavelet coefficients and $\psi(s,u)$ is the family of wavelets where s and u represent the dilation (scaling) and translation (shifting) parameters, respectively. The family of wavelets represent the translations and the dilations of the mother wavelet $\psi(t)$ and can be expressed in the form:

$$\psi_{s,u} = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-u}{s}\right), \qquad s,u \in R, s \neq 0 \tag{6.13}$$

The best known wavelets are the Daubechies wavelets ($db\varepsilon$) and the Coifman wavelets ($coif\varepsilon$). In both cases, ε is the number of vanishing moments of the functions. Daubechies also suggested the 'symlets' as the nearly symmetric wavelet family as a modification of the db family. The family 'Haar' is the well-known Haar basis [95]. Figure 6.4 shows a number of wavelet functions. As can be seen, the Haar functions are discontinuous and may not provide good approximation for smooth functions.

In general, one would be interested in not only analyzing the signal but also in reconstructing (synthesizing) the original signal using the wavelet coefficients. While the mother wavelet can be any function for the former exercise, it has to satisfy more conditions to provide the latter. The wavelet functions are designed to have large number of moments (zero-crossings), thus, the expansion of functions on such wavelet bases needs much fewer terms than the Taylor expansion. This property leads to very sparse decompositions of functions, which facilitates the applications such as filtering and data compression. For ideal signal reconstruction, the wavelets should satisfy the orthogonality condition if the same wavelet is to be used for both analysis and synthesis. For more details on the properties of WT, the reader can consult a number of excellent books [228, 235].

Figure 6.5 depicts the frequency coverage of FT, STFT and WT. While FT provides information on the power of frequencies present in the signal, STFT can follow the signal in fixed windows and show the presence and

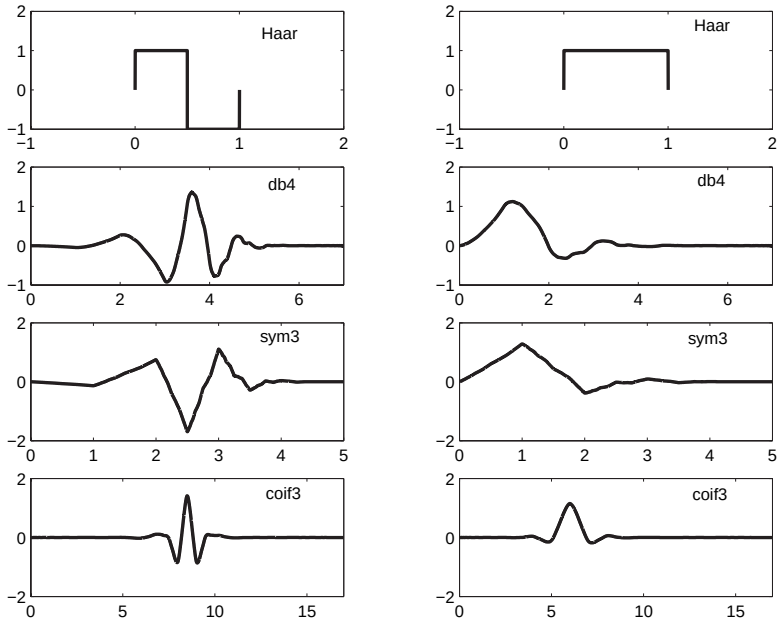


Figure 6.4. The scaling (left) and wavelet (right) functions for four wavelets, Haar, Daubechies 4, Symlet 3 and Coiflet 3.

the power of frequencies present in each window. Furthermore, while the tiling of STFT is linear, the tiling of WT is logarithmic [235], indicating that the building blocks in two decompositions are different, and frequency localization in WT is proportional to the frequency level. Thus, for WT, time localization gets finer in the highest frequencies. The multiresolution decomposition concept that will be discussed next allows for an arbitrary tiling of the time-frequency plane.

Example The *CWT* of the signals defined in Eq. 6.6 and Eq. 6.7 are shown in Figure 6.6. Here, the Daubechies 4 (db4) wavelet is used. The *CWT* yields a three-dimensional representation similar to that of STFT and easily discriminates between the two signals, correctly pinpointing the times at which the signal frequency changes. For $f_1(t)$, since all frequency components are present for the entire duration, two prominent bands are observed spanning the time axis at different scales. Note that since there are two higher-frequency components of the signal, the band at the lower scales appears fuzzy as one frequency masks the other. For $f_2(t)$, the appearance and disappearance of the low-frequency behavior are delineated clearly by

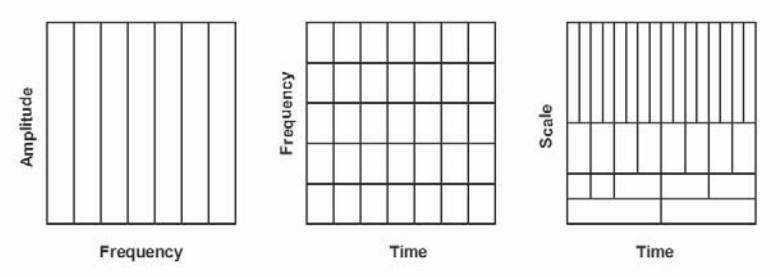


Figure 6.5. The frequency coverage of FT, STFT and WT.

the appearance and disappearance of the band of wavelet coefficients at higher scales. One can also see clearly the difference in the two frequencies at the lower scales.

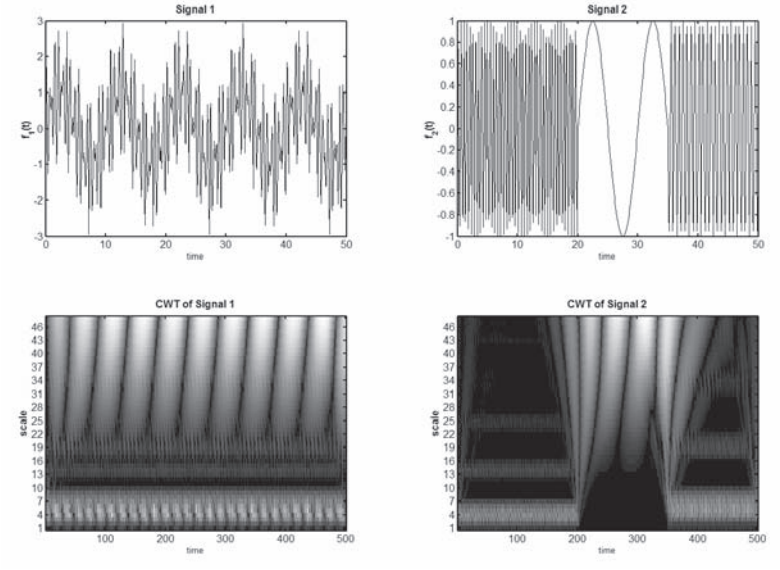


Figure 6.6. Two signals and their corresponding *CWT*.

6.1.3 Discrete Wavelet Transform

The *CWT* results in wavelet coefficients at every possible scale. Thus, there is a significant amount of redundancy in the computation. But there

is an easy way to obtain WT, which is called the discrete wavelet transform (DWT). DWT is a special case of the WT and is based on dyadic scaling and translating. For most practical applications, the wavelet dilation and translation parameters are discretized dyadically ($s = 2^j, u = 2^j k$).

A process signal $z(t)$ can be represented through DWT as follows [44]:

$$z(t) = \sum_k c_k \phi_{j_0,k}(t) + \sum_{j=-\infty}^{j_0} \sum_k d_{j,k} \psi_{j,k}(t) \tag{6.14}$$

with

$$c_k \equiv \int z(t) \phi^*_{j_0,k}(t) dt$$

$$d_{j,k} \equiv \int z(t) \phi^*_{j,k}(t) dt$$

Here, the wavelet function, $\psi(t)$, and the scaling function, $\phi(t)$ (see Figure 6.4), are defined as

$$\psi_{j,k}(t) \equiv 2^{-j/2} \psi(2^{-j} t - k) \tag{6.15}$$

$$\phi_{j_0,k}(t) \equiv 2^{-j/2} \phi(2^{-j_0} t - k), \quad j, k \in Z \tag{6.16}$$

In this representation, integer j indexes the scale or resolution of analysis, i.e., smaller j corresponds to a higher resolution, and j_0 indicates the coarsest scale or the lowest resolution. k indicates the time location of the analysis. For a wavelet $\phi(t)$ centered at time zero and frequency ω_0, the *wavelet coefficient* $d_{j,k}$ measures the signal content around time $2^j k$ and frequency $2^{-j}\omega_0$. The *scaling coefficient* c_k measures the local mean around time $2^{j_0} k$. The DWT represents a function by a countable set of wavelet coefficients, which correspond to points on a 2-D grid of discrete points in the scale-time domain.

Mallat [182] proposed an algorithm, referred to as the multiresolution signal decomposition (MSD), to efficiently perform DWT. Its basic idea is to use a low-pass filter (see Section 6.2.1) and a high-pass filter to decompose a dyadic-length discrete signal (time series) into low frequency and high frequency components, respectively. As shown in Figure 6.7, for a signal S consisting of 128 points, one also performs a down-sampling operation to reduce the number of points in each scale by half. It is noted that, for discrete signals, the upper limit for the scales is bounded by the maximum number of available details in the signal.

One can show that the relationship between high-pass and low-pass finite impulse response (FIR) filters and the corresponding wavelet and

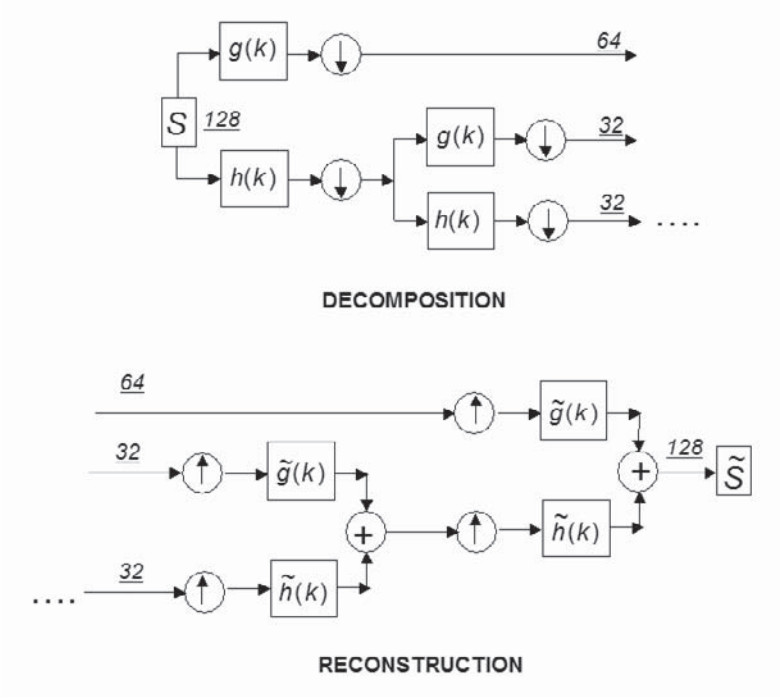


Figure 6.7. The signal decomposition and reconstruction using FIR filters. Note that $\tilde{g}$ and $\tilde{h}$ are the dual filters and $\tilde{S}$ is the reconstructed signal.

scaling functions can be expressed as:

$$\psi(t) = \sqrt{2} \sum_k h(k)\psi(2t-k) \tag{6.17}$$

$$\phi(t) = \sqrt{2} \sum_k g(k)\psi(2t-k) \tag{6.18}$$

This approach greatly facilitates the calculation of wavelet and scaling coefficients as typically implemented in the Matlab® Wavelet Toolbox [199]. One can associate the scaling coefficients with the signal *approximation*, and the wavelet coefficients as the signal *detail*.

Due to the down-sampling procedure during decomposition, the number of resulting wavelet coefficients (i.e., approximations and details) at each level is exactly the same as the number of input points for this level. It is sufficient to keep all detail coefficients and the final approximation coefficient (at the coarsest level) to be able to reconstruct the original data. The

signal reconstruction involves the reverse procedure and up-sampling which inserts zeros in between signal values from the previous level (see Figure 6.7).

Example The *DWT* of the two signals defined in Eq. 6.6 and Eq. 6.7 are shown in Figures 6.8 and 6.9, respectively. Some random noise (zero mean, unit variance) is also added to each signal to distort the original features slightly. Figures show the approximate and the detail coefficients of the MSD for a three level decomposition. Following observations are made:

- The detail coefficients show the strength of the signal component removed at each scale level. Especially in Figure 6.9, one can clearly see how the first level removes the noise components followed by signal components with distinct frequency behavior.
- Each approximate signal level depicts a coarser approximation of the signal, with the last level (level 3) showing the key underlying signal feature (mean).
- As one can see in the detail signals, each level represents a band-pass filtered signal, thus comprising a range of characteristic frequencies.
- While high-frequency noise is expected to be removed at the first decomposition level, one can see that the effect of noise persists in all scales.

There are a few issues that any user should be aware of in applying the WT to the signals of interest. Below, some of these issues are highlighted (the reader is referred to the references mentioned earlier for a more detailed discussion):

- It is not a straightforward task to come up with a procedure that would lead to the best mother wavelet for a given class of signals. Nevertheless, exploiting several characteristics of the wavelet function, one can determine which family of wavelets would be more appropriate for a specific application.
- As a general rule, all orthogonal wavelets lack symmetry. This becomes an issue in applications such as image processing where symmetric wavelets are preferable. The symmetric wavelets also facilitate the handling of image boundaries.
- Dealing with boundaries becomes an issue in wavelet analysis of finite array of data. These edge effects or singularities can be avoided by using adaptive filters near the signal boundaries.

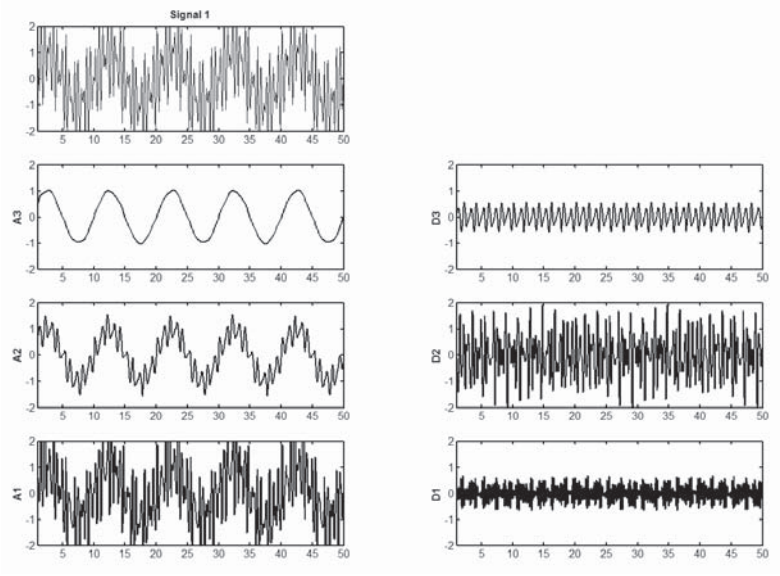


Figure 6.8. The DWT of noisy signal 1 using Daubechies 4 wavelet.

- An important property of wavelet bases is their lack of translational invariance. In other words, when a pattern is translated, its descriptors are not only translated but also modified. This is a direct consequence of the down-sampling procedure and leads to distorted reconstruction of the underlying signal features. A possible solution is to omit down-sampling, resulting in a redundant family of coefficients.

6.2 Filtering and Outlier Detection

The measurements of process signals inherently contain noise that consists of random signal disturbances interfering with the actual signal. The signal noise can be due to variations in voltage, current or the measurement technique itself. If the signal-to-noise ratio (SNR) is small, one may encounter misleading or biased results in subsequent data analysis steps. Thus, denoising (noise filtering) is a crucial step in signal analysis that is aimed at removing random signal behavior and producing a clean signal that contains relevant process characteristics. Numerous techniques have been proposed for filtering process signals, going back to the seminal paper by Kalman [135] and others, [221, 309].

In addition to noise, process data may also contain outliers (gross er-

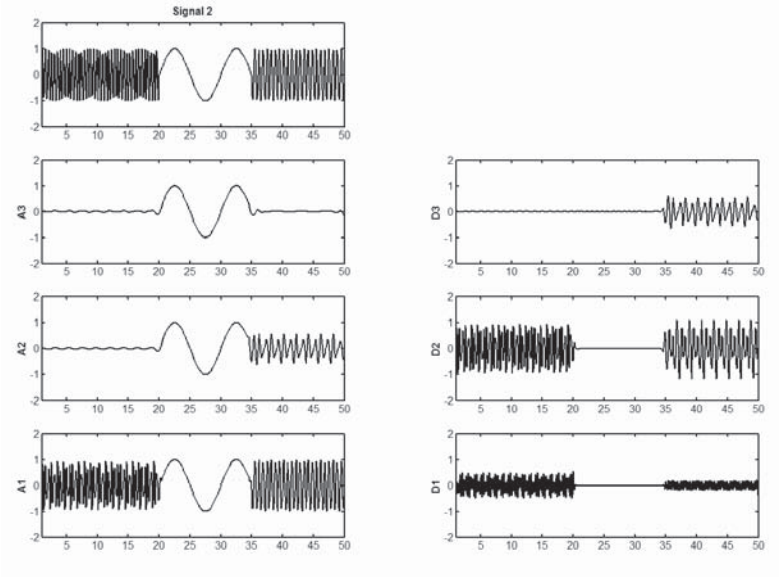


Figure 6.9. The DWT of noisy signal 2 using Daubechies 4 wavelet.

rors) that may comprise up to 10 % of the data points in low-quality data sets [101]. In practical applications, one might consider the observations exceeding five standard deviations as outliers, and in univariate data sets, the outliers can be easily identified by visual inspection. However, for higher dimensional data sets, this task becomes cumbersome and often intractable. Due to their influence on the actual signal characteristics, outliers need to be removed in a systematic manner, often through methodologies that perform this task in an unsupervised manner [26, 57].

6.2.1 Simple Filters

In the signal processing literature, there are a number of filtering techniques that can be adopted for a variety of purposes. Here, only two of them will be considered, chiefly in the context of denoising, namely the low-pass filter and the median filter. The goal is to introduce the reader to the concept of filtering and also prepare the groundwork for wavelet filtering techniques to be discussed next.

The so-called low-pass filter removes the high frequency components of a signal and was referred to earlier in Section 6.1.3 (see also EWMA charts in Section 2.2.4). Suppose that $y(k)$, with $k = 1, \ldots n$, represents the true

signal and $\hat{y}(k)$ is the noisy signal that one observes. The filter equation represents a first-order difference equation given by,

$$\tilde{y}(k) = \beta\hat{y}(k) + (1-\beta)\tilde{y}(k-1) \tag{6.19}$$

Here, $\tilde{y}(k)$ represents the estimate of the true signal. Further, β is the filter constant, or, in other words, the filtering bandwidth. By a judicious choice of β, one can remove high-frequency noise components from the signal and retain the relevant signal characteristics. Figure 6.10 shows the frequency response of a first-order filter and how the bandwidth changes as a function of β.

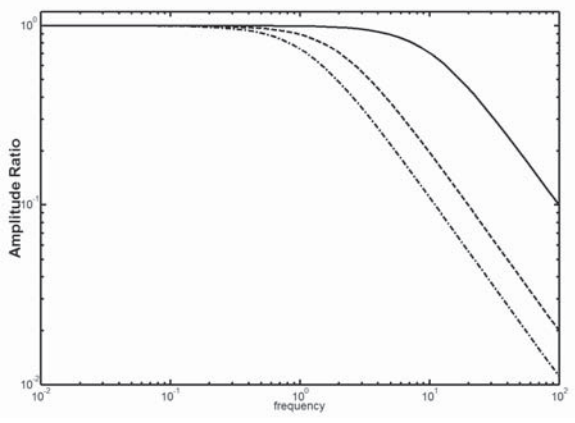


Figure 6.10. The frequency response of a low-pass filter with different filter constants. $\beta = 0.1$ (solid), $\beta = 0.5$ (dash), $\beta = 0.9$ (dashdot)

One drawback of the first-order filter is associated with the slope of the frequency response that indicates how sharp the cut-off is for high frequency components. Since the frequency response attenuates rather slowly at high frequencies, this would create a somewhat ineffective denoising performance if the SNR is relatively low.

Example Figure 6.11 shows a noisy signal and the effect of β on the denoising performance. As one can observe, the closer the value of β is to 1, the less effective the denoising is, since the bandwidth becomes larger, almost reproducing the original signal. Yet, smaller values of β may also be ineffective as the filter tends to remove relevant signal characteristics as the bandwidth gets smaller, resulting in the removal of signal components at moderate to low frequencies.

Since the test signal is known in this case, the performance of the filtering methods can be evaluated by measuring the fidelity of the denoised

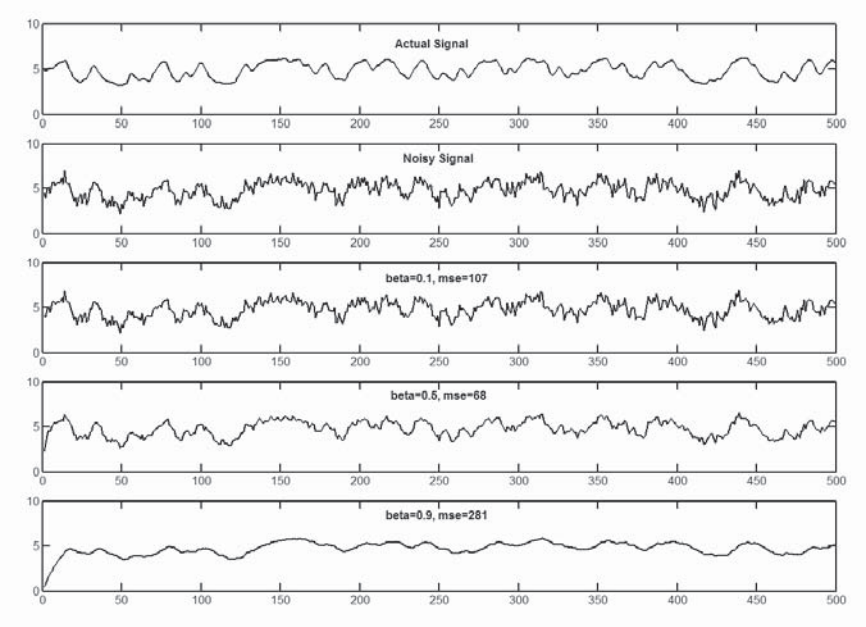


Figure 6.11. The actual test signal, the test signal with noise and its denoised estimates using three different filter constants.

signal to the original signal. The mean square error (MSE) can be calculated as,

$$MSE = \sqrt{\frac{\sum_{k=1}^{n}(\tilde{y}(k) - \hat{y}(k))^2}{n}} \tag{6.20}$$

As shown in Figure 6.11, the lowest MSE is associated with $\beta = 0.5$ and one can judge the quality of denoising visually as well.

Another simple filter is the moving median (MM) filter first developed by Tukey [299]. In this filtering technique, the median of a window containing an odd number of observations is calculated as the window slides over the entire signal. As a result, the original signal is freed from noise as well as from outliers. Davies [47] showed that the MM filter could handle signals that have moderate or high SNR, or contaminated with noise, which comes from asymmetric distributions. The MM filter equation is expressed as,

$$\tilde{y}(k) = med\,(\hat{y}[l - w/2], ..., \hat{y}[l + w/2]) \tag{6.21}$$

with $l = w/2 + 1, w/2 + 2, ...n - w/2$ and $w + 1$ is the window length. Similar to the filter constant of the low-pass filter, the window length is the

adjustable parameter for MM, and defines the quality of denoising as will be illustrated in the example below.

Example Figure 6.12 shows the performance of the MM filter for three different choices of the window length. It can be seen that a smaller window is almost as good as the low-pass filter with $\beta = 0.5$ and longer window lengths actually produce a signal estimate much closer to the actual signal as the lower *MSE*s indicate.

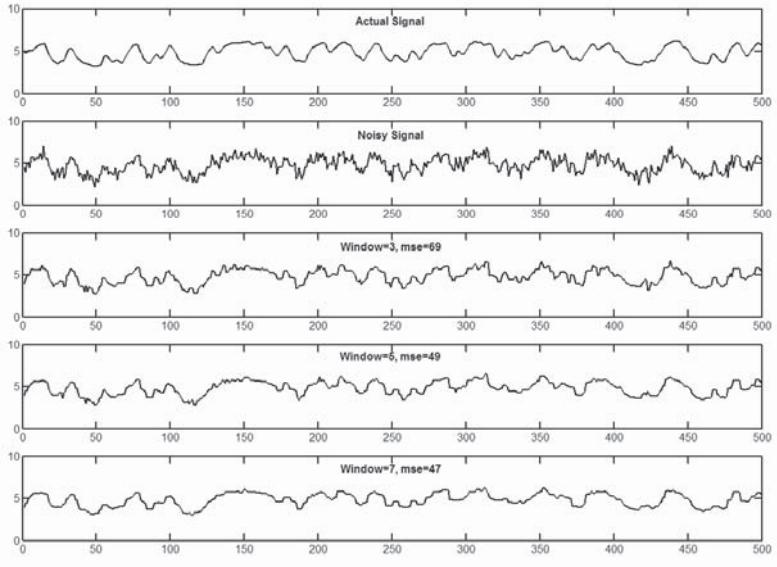


Figure 6.12. The actual test signal, the test signal with noise and its denoised estimates using the moving median filter with different window lengths.

6.2.2 Wavelet Filters

Wavelet-based denoising methods involve taking the discrete wavelet transform (DWT) of a signal, passing the resulting wavelet coefficients through a thresholding step and then taking the inverse DWT (Figure 6.13). If a signal has its energy concentrated in a small number of wavelet dimensions, the magnitude of its coefficients will be relatively large compared to noise components that have their energy spread over a large number of coefficients. This implies that, during thresholding (or shrinkage), the wavelet transform will remove the low amplitude noise or the undesired signal com-

ponents in the wavelet domain, and an inverse wavelet transform will then reconstruct the desired signal with little loss of relevant features.

Figure 6.13. Wavelet-based denoising strategy.

The following thresholding methods can be defined:

1. The *hard-thresholding* filter, F_H, selects wavelet coefficients that exceed a certain threshold and sets the others to zero:

$$F_H(d) = \begin{cases} 1 & |d| \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{6.22}$$

2. The *soft-thresholding* filter, F_L, is similar to the hard-thresholding filter, but it also shrinks the wavelet coefficients above the threshold,

$$F_L(d) = \begin{cases} d - \tau & |d| \geq \tau \\ 0 & |d| < \tau \\ d + \tau & |d| \leq \tau \end{cases} \tag{6.23}$$

The soft-thresholding is often preferred as the hard-thresholding has discontinuities that introduce artifacts to the denoised signal. The next step is to determine the threshold value, τ.

Donoho and Johnstone [56] suggest $\tau = \sqrt{2\sigma_e^2 log(n)}$ for thresholding (also called the universal threshold). Here, σ_e^2 is the estimate of the noise variance and n is the length of the time series. If soft-thresholding is used in conjunction with this threshold, then the estimates with high probability are as smooth as the original ones and with small values for their risks in both the bias and the variance. This approach is referred to as *VisuShrink* by Donoho and Johnston [56], in reference to the good visual quality of the reconstruction obtained by choosing the appropriate threshold and simple 'shrinkage' of wavelet coefficients. However, it may overly smooth the signal for large n. Another shrinkage approach is referred to as *SureShrink* that uses a hybrid of the universal threshold and the SURE threshold along with soft-thresholding, and is derived from minimizing Stein's unbiased risk estimator [282].

Example Figure 6.14 displays the denoising performance of VisuShrink and SureShrink strategies using two different wavelets. Here, the wavelet

coefficients for the first two levels of the wavelet decomposition are selectively denoised. As the number of decomposition levels (scales) increases, the reconstructed signals tend to become smoother, losing some of the relevant features. Thus, the *MSE* significantly improves compared with the performance of the simple filters. One can also see that the SureShrink with db8 performs the best for this application.

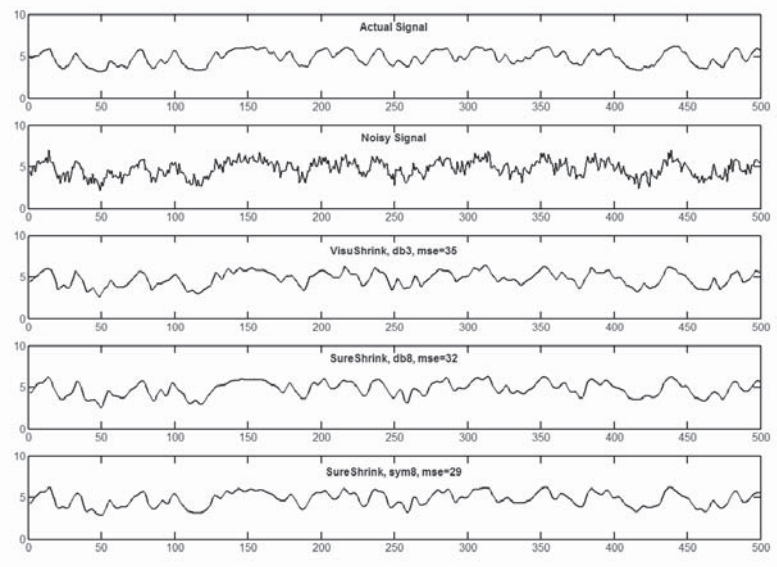


Figure 6.14. The actual test signal, the test signal with noise and its denoised estimates using VisuShrink and SureShrink.

6.2.3 Robust Filter

The performance of denoising methods tends to deteriorate in the presence of outliers. Doymaz *et al.* [59] proposed a robust filtering strategy that uses a median filter (MM) (see Section 6.2.1) in tandem with the coefficient denoising method [7]. Here, this strategy is briefly reviewed and its key benefits for denoising are pointed out.

The robust filtering strategy is depicted in Figure 6.15 in which the primary goal of MM is to remove outliers so that the wavelet denoising step can be more effective. It should be recognized that while MM removes outliers, it also removes some noise elements thus complementing the subsequent denoising step. This is an important issue, even in the absence

of outliers, since the decision regarding the length of the moving window becomes less critical and choosing it as 3 often suffices.

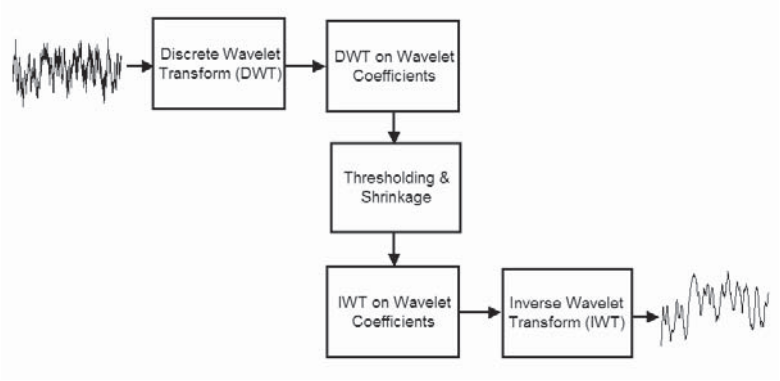


Figure 6.15. The schematic of the robust filtering strategy.

The denoising step in Figure 6.15 uses a novel filtering scheme that uses two wavelet shrinkage stages. In addition to the traditional thresholding and shrinkage, the coefficient denoising method uses a second thresholding and shrinkage step for the wavelet coefficients. It has been proven that this strategy has better denoising performance [7] when the coefficient denoising step uses the Wiener filter which requires the knowledge of the statistics of the signal and the noise. An approximation of the optimal Wiener filter can be obtained using the diagonal elements as,

$$F_W(d) = \frac{d^2}{d^2 + \sigma^2} \tag{6.24}$$

The noise variance σ^2 can be estimated from the wavelet coefficients of the noisy signal and then used in Eq. 6.24 [90]. It is known that the larger the variations in the input data power (relative to noise variance), the greater the loss in performance due to simple thresholding compared to optimal Wiener weighting. In general, if the signal is smooth, it will have a larger energy spread over the scaling coefficients, resulting in a substantial performance loss. Thus, the coefficient denoising approach, by taking double transformation and using Wiener thresholding in the space of coefficients, spreads the energy less over the detail coefficients, resulting in better performance than simply using the Wiener shrinkage. Naturally, this procedure cannot be continued further, because the signal would then start to lose its fundamental features.

Example The Bumps benchmark signal is contaminated with additive

Table 6.1. MSE values for various filtering strategies.

MM filter	Wiener Thresholding	Robust Filtering
0.7474	4.422	0.4768

Gaussian white noise, $N(0,1)$, and 1% outliers were added using Poisson distribution. The results [59] show that the robust filtering approach performs quite well. Figure 6.16 depicts the visual performance of the robust filtering strategy while Table 6.1 demonstrates how MSE is minimized with this tandem approach. Using just a simple MM filter, the signal is freed of outliers and the noise quite satisfactorily, while the wavelet Wiener thresholding suffers from the presence of outliers significantly. The tandem approach appears to achieve the minimum MSE.

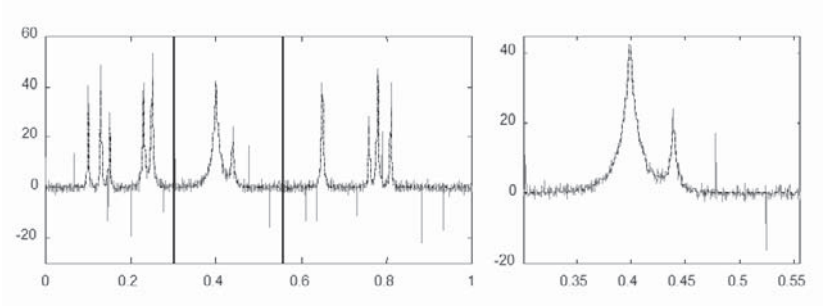


Figure 6.16. The robust filtering technique performed on the Bumps benchmark signal (grey solid) with noise and outliers. The dashed signal is the signal estimate. Reprinted from [59]. Copyright © 2001 with permission from Elsevier.

6.3 Signal Representation by Fuzzy Triangular Episodes

The analysis of process signals may be facilitated if the time series data can be cast into a symbolic form. The relevant trends and generic data features can then be extracted and monitored using this qualitative representation. Such a transformation is often carried out by defining a set of primitives (alphabet) that define a visual characteristic of the signal [78, 142, 247]. Here, the methodology proposed by Stephanopoulos and coworkers is discussed [9, 34, 35]. They treated the problem of trend representation graphically

by utilizing a declarative language based on the notion that at the extrema or inflection points, the first or second derivatives, respectively, are zero. Thus, an *episode*, $E_{[a,b]}$, is described as any part of a signal or process trend given by

$$E_{[a,b]} = \{(t_a, y_a), (t_b, y_b,)\} \tag{6.25}$$

with a constant sign of the first and the second derivatives,

$$sign\left(\frac{\partial y}{\partial t}\right)_{[a,b]} = \text{constant}; \quad sign\left(\frac{\partial^2 y}{\partial t^2}\right)_{[a,b]} = \text{constant} \tag{6.26}$$

Here, the time series segment is defined by the time duration of the episode, $[t_a, t_b]$, and the signal magnitude, $[y_a, y_b]$ (Figure 6.17). For each episode, a triangle is created, where one side of the triangle is constructed by drawing a line between the two end points of the episode. The other sides are drawn by connecting the tangents of these endpoints, up to the point where the slopes intersect. It is noted that this is a semi-qualitative representation because the positions of the endpoints (duration, magnitude) as well as the slopes of the tangents to the curve at the endpoints are also retained.

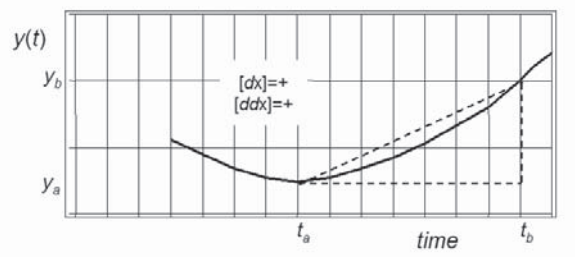


Figure 6.17. Definition of an episode. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

Cheung and Stephanopoulos [34] were able to reduce the time series into a semi-qualitative form using seven primitive shapes that consist of four triangles and three straight lines (Figure 6.18).

A drawback of this method is its sensitivity to high-frequency noise in the time series, thus a filtering step becomes necessary. Cheung and Stephanopoulos [35] overcome this problem by using a filtering process described as qualitative scaling. In this geometric approach, the original sequence of letters is sequentially reduced by approximating a subsequence of letters within the original sequence by a trapezoid. Bakshi and Stephanopoulos [9] point out that this is a heuristic formulation and lacks computational speed. They offer an alternative strategy by utilizing

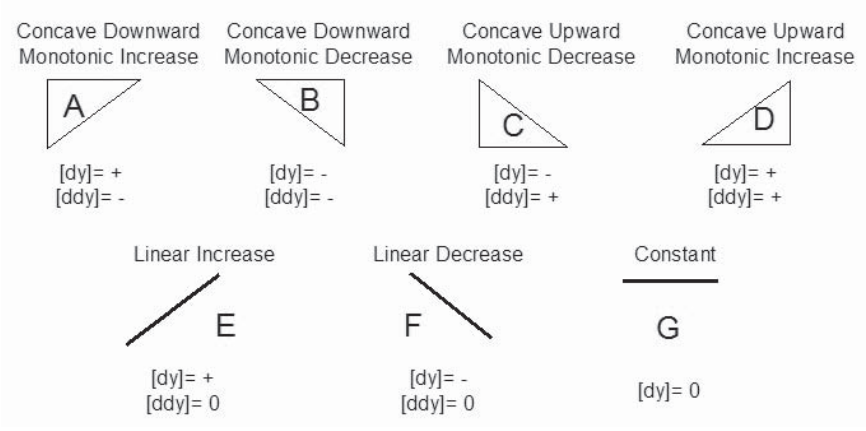


Figure 6.18. Definition of primitive shapes. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

wavelet analysis and scale-space filtering in conjunction with the triangular representation. Wong *et al.* [340] suggest simple wavelet denoising as a prelude to episode construction.

To obtain a fully symbolic representation of the time series, Wong *et al.* [340] propose a fuzzification procedure. In fuzzy logic [346], the basic premise is to characterize the mapping of a set of inputs to a set of outputs by using a set of if-then rules. An attractive feature of this approach is its ability to convert numeric data into linguistic variables. A membership function is used to define how well a variable belongs to the output based on the degree of membership between 0 and 1. When a set of inputs is to be mapped to a set of outputs, a combination of if-then rules, membership functions and logical operators are used in order to create a fuzzy model. Based on the if-then rules and the logical operators (AND, OR, etc.), these inputs can be combined to generate an output for each rule.

In Wong *et al.* [340], the quantitative values of magnitude and duration of a triangular episode are fuzzified using symbolic variables (*small*, *medium*, and *large*) expressed by two membership functions (Figure 6.19). With this technique, each triangle or line in Figure 6.18 can be transformed into nine different qualitative triangles or lines. Figure 6.20 shows the triangle, A, now expressed as nine new triangles. For example, smA represents a symbol with the characteristic shape described by the letter A that is small, s, in magnitude and medium, m, in duration. All letters are similarly fuzzified with the exception of G which represents a straight line with no change in magnitude. Hence, one only has three new fuzzified lines, sG,

mG, and lG. Now, a new alphabet emerges with 57 symbolic characters. The new alphabet is more versatile because it allows the comparison of the sequences based on the size of the characters as well as their shape. This symbolic representation will be the basis for a process trend analysis strategy that will be introduced in Section 7.1.

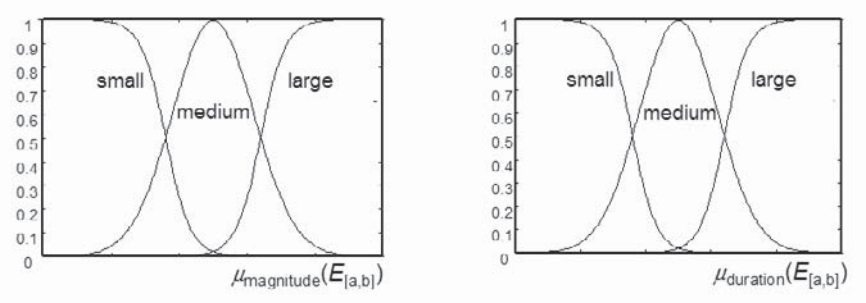


Figure 6.19. Membership functions for the duration and magnitude of primitive shapes. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

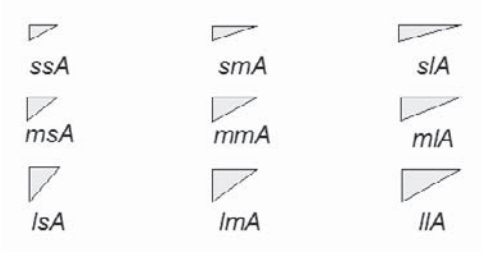


Figure 6.20. Extending the alphabet to include the fuzzified triangle A. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

6.4 Development of Markovian Models

The Hidden Markov Model (HMM) is a powerful statistical tool for modeling a sequence of data elements called the observation vectors. As such, extraction of patterns in time series data can be facilitated by a judicious selection and training of HMMs. In this section, a brief overview will be presented and the interested reader can find more details in numerous tutorials

on this subject. The most notable applications of HMMs are found in the fields of automatic speech recognition (ASR) [117, 240, 239] and bioinformatics [65, 144]. In ASR, the goal is to differentiate among a vocabulary of spoken words while recognizing the same words spoken by different people. In biological sequence matching, one attempts to match unknown sequences of amino acids to a known family of proteins. Given the ability of HMMs to model time series data, a number of studies on fault detection and trend analysis have been reported [277, 286, 340].

Before introducing the HMMs, it is imperative to understand the concept of Markov chains and the probability models. Next subsection provides a brief account of these topics.

6.4.1 Markov Chains

A (first-order) Markov process is defined as a finite-state probability model in which only the current state and the probability of each possible state change is known. Thus, the probability of making a transition to each state of the process, and thus the trajectory of states in the future, depends only upon the current state. A Markov process can be used to model random but dependent events. Given the observations from a sequence of events (Markov chain), one can determine the probability of one element of the chain (state) being followed by another, thus constructing a stochastic model of the system being observed. For instance, a first-order Markov chain can be defined as

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_i, \ldots) = P(q_t = S_j | q_{t-1} = S_i) \tag{6.27}$$

Here, we used the shorthand notation q_t to denote the value of q at time t as $q(t)$ in an attempt to simplify the subsequent expressions. Here, t is used to denote the traditional use of the time instant in the Markovian modeling literature. The notation $P(x|y)$ indicates the conditional probability of observing x, premised on the presence of y. The change of the states is captured by transition probabilities, a_{ij}, given by

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), \qquad 1 \leq i, j \leq M \tag{6.28}$$

where M represents the number of states. The transition probabilities satisfy the following relationships:

$$\begin{aligned} a_{ij} &\geq 0 \\ \sum_{j=1}^{M} a_{ij} &= 1 \end{aligned} \tag{6.29}$$

Figure 6.21 shows a cyclic three-state Markov chain, $\mu = \{S_1, S_2, S_3\}$. Given an initial state and the matrix of transition probabilities, one can not only estimate the state of the chain at any future instant but can also determine the probability of observing a certain sequence, using the state transition matrix. The examples below demonstrate these cases.

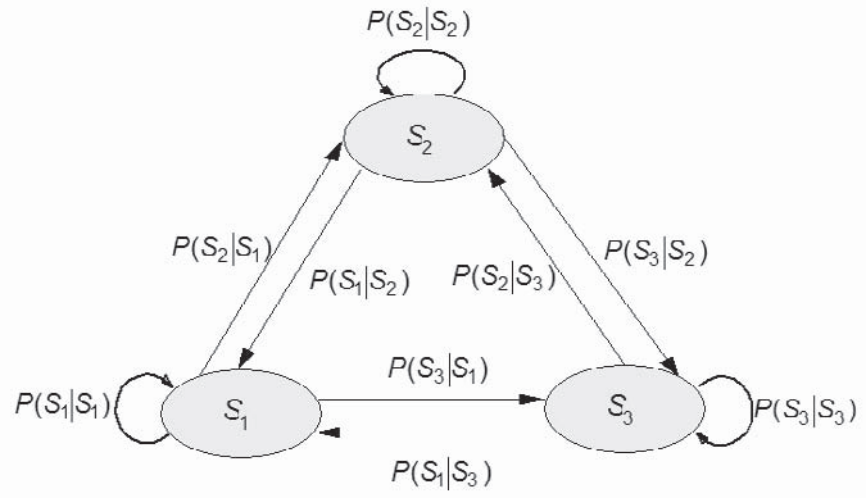


Figure 6.21. A three-state Markov model.

Example An analysis is presented where the consequences of brand switching between three different brands of laundry detergent, X, Y and Z are explored. A market survey is conducted to estimate the following transition matrix for the probability of moving between brands each month:

$$\mathbf{A} = \begin{bmatrix} 0.80 & 0.15 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.25 & 0.70 & 0.05 \end{bmatrix} \tag{6.30}$$

For the first (current) month, market shares are given as 40%, 25% and 35% for brands X, Y and Z respectively. This establishes the initial condition, $S_1 = [0.40, 0.25, 0.35]$. The expected market shares after three months have elapsed will be estimated. Hence, after one month has elapsed, the state of the system is given as $S_2 = S_1\mathbf{A} = [0.38, 0.36, 0.26]$ and after three months have elapsed the state of the system is given as $S_4 = S_3\mathbf{A} = S_2\mathbf{A}^2 = [0.318, 0.5084, 0.1736]$. Note that the elements of S_4 add to one as required. Hence, the market shares after three months have elapsed are given as 31.8%, 50.84% and 17.36% for brands X, Y and Z, respectively.

Example The classic example of a Markov chain is the weather pattern modeling [240]. Again, consider a three-state Markov model in which the states, characterizing the weather on any given day t, are given as follows: State 1: rainy; State 2: cloudy; State 3: sunny. The state transition probability matrix is defined as,

$$\mathbf{A} = \begin{bmatrix} 0.40 & 0.30 & 0.30 \\ 0.20 & 0.60 & 0.20 \\ 0.10 & 0.10 & 0.80 \end{bmatrix} \tag{6.31}$$

The goal is to determine the probability of observing the sequence 'cloudy-rainy-sunny' for the next three days, if the weather today is sunny. In other words, one is interested in the probability of the observation sequence, $O = \{S_3, S_2, S_1, S_3\}$.

$$\begin{aligned} P(O|Model) &= P(S_3, S_2, S_1, S_3|Model) \\ &= P(S_3) \cdot P(S_2|S_3) \cdot P(S_1|S_2) \cdot P(S_3|S_1) \\ &= \pi_3 \cdot a_{23} \cdot a_{12} \cdot a_{31} \\ &= 1 \cdot (0.2) \cdot (0.3) \cdot (0.1) \\ &= 0.006(0.6\%) \end{aligned} \tag{6.32}$$

Here, the notation $\pi_i = P(q_1 = S_i)$ is used for the initial state probability.

6.4.2 Hidden Markov Models

Hidden Markov models (HMMs) are doubly stochastic in nature. In other words, the sequence of states, $S = S_1, S_2, S_3, ..., S_M$, of a Markov chain are unobservable yet still are defined by the state transition matrix. In addition, each state of the Markov chain is associated with a discrete output symbol probability that generates an observable output sequence (outcome), $O = o_1, o_2, \ldots, o_T$ with length T. HMMs are finite because the number of states, M, as well as the number of observable symbols $V = v_1, v_2, \ldots, v_L$ of an output alphabet, i.e., L, remain fixed for a particular model. Since it is only the outcome, not the state visible to an external observer and the states are 'hidden' to an outside observer, such a system is referred to as the Hidden Markov Model.

Example The concept of HMMs can be best explained by the urn-and-ball example discussed by Rabiner [240]. Consider a collection of urns where each urn contains a different proportion of colored balls which defines the probability of drawing a specific colored ball from that urn. The data are generated by drawing a colored ball from an urn, and then based on that selection, a new urn is chosen and a another ball is drawn. The process

is continued until a sequence of balls is generated. In this process, the sequence of the chosen urns is not announced (thus, *hidden*) and only the sequence of balls is known (observed).

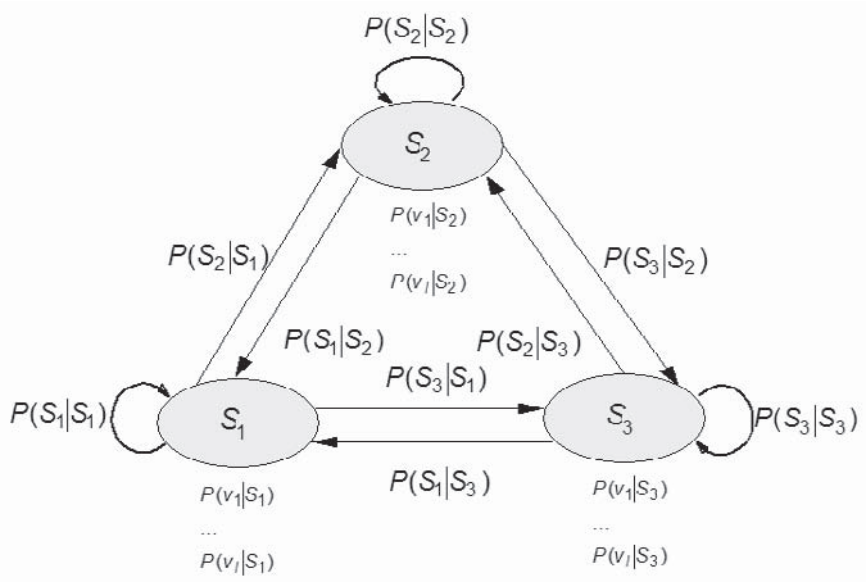


Figure 6.22. A three-state HMM showing the state transitions and the output probabilities. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

To illustrate the construction and properties, the three-state Markov model depicted in Figure 6.21 is extended to express a HMM as given in Figure 6.22. The key difference is that each state now has a set of observation symbols, along with the probability of observing that symbol, o_l, from a given alphabet of symbols, V, in that state i, $P(o_l|S_i)$.

Example For the ball-and-urn example, the observations clearly are the balls that are drawn and each state is represented by an urn. Thus, the observation probability corresponds uniquely to the specific urn from which a ball is drawn. The three-state HMM given in Figure 6.22 can be used to model the observed symbols (balls) by estimating a set of HMM parameters.

A HMM, denoted as λ, can be uniquely described (parameterized) by M, the number of states, L, the number of observation symbols, and three probability measures, π, $\mathbf{A}$, and $\mathbf{B}$.

$$\lambda = (\pi, \mathbf{A}, \mathbf{B}) \tag{6.33}$$

where

$$\begin{aligned} \pi &= \{P(S_i|t=1)\} \\ \mathbf{A} &= \{a_{ij}\} = \{P(q_{t+1} = S_j|q_t = S_i)\} \\ \mathbf{B} &= \{b_j(l)\} = \{P(v_l att|q_t = S_j)\} \end{aligned} \tag{6.34}$$

An initial state distribution, π, defines the probabilities of beginning the observation in each state. The matrices **A** and **B** are the probability density distributions of the state transitions and the observation symbols, respectively.

The number of states is usually unknown, but some physical intuition about the system can provide a basis for defining M. Naturally, a small number of states usually results in poor estimation of the data, while a large number of states improves the estimation but leads to extended training times. The quality of the HMM can be gauged by considering the residuals of the model or the correlation coefficients of observed and estimated values of the variables. The residuals are expected to have a Normal distribution ($N(0, \sigma^2)$) if there is no systematic information left in them. Hence, the normality of the residuals can provide useful information about model performance in representing the data.

The number of observation symbols is more definitive as it corresponds directly to the possible outcomes of the system being observed (e.g., the number of different colors that the balls would have in the urns). There are three key problems that need to be solved: training (learning), evaluation and state estimation.

For the *evaluation problem*, the probability of an observation sequence $O = o_1, o_2, \ldots, o_T$ is determined, given the model λ, $P(O|\lambda)$. This probability can be found using the forward part of the inductive forward-backward algorithm (Baum-Welch algorithm [13]), which is initialized by

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq M \tag{6.35}$$

where $\alpha_t(i)$ is the forward variable,

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = S_i|\lambda) \tag{6.36}$$

Then, using the forward inductive equation, the induction step is performed:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{M} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}) \tag{6.37}$$

where $\alpha_t(j)$ is the probability of being in state j and observing the partial symbol sequence $O = o_1, o_2, \ldots, o_t$ up to time t, given the HMM, λ. Then,

we have

$$P(O|\lambda) = \sum_{i=1}^{M} \alpha_T(i) \tag{6.38}$$

which yields the final result. This algorithm establishes the basis for the classification of faults as will be illustrated in Sections 7.1 and 7.2.

In the *training problem*, the model parameters are estimated that best describe the observation sequence. In other words, the observation sequence is used to train the HMM by adjusting the model parameters. This training is again accomplished through the Baum-Welch algorithm [13] that uses the maximum likelihood estimation approach to adjust the parameters, π, $\mathbf{A}$, and $\mathbf{B}$ in order to maximize $P(O|\lambda)$. The backward part of the forward-backward algorithm is used in the training step and initialized with

$$\beta_T(i) = 1 \qquad\qquad 1 \leq i \leq M \tag{6.39}$$

where $\beta_k(i)$ is the backward variable,

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T, q_t = S_i|\lambda) \tag{6.40}$$

The inductive backward equation is given by

$$\beta_t(i) = \sum_{i=1}^{M} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{6.41}$$

The combination of the two inductive parts are essential in the re-estimation of the parameters of the HMM. By maximizing the auxiliary function

$$Q(\lambda|\bar{\lambda}) = \sum_{Q} P(Q|O,\lambda) log[P(O,\lambda|\bar{\lambda})] \tag{6.42}$$

the re-estimation formulas

$$\bar{\pi} = \alpha_1(i)\beta_1(i) \tag{6.43}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \tag{6.44}$$

$$\bar{b}_j(l) = \frac{\sum_{t-1;s.t.o_t=o_l}^{T-1} \alpha_t(i)\beta_t(j)}{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \tag{6.45}$$

can be derived. The use of three re-estimation formulas guarantees that the new probability $P(O|\bar{\lambda})$, using the estimated parameters, is greater than or equal to the prior probability $P(O|\lambda)$.

In the *state estimation problem*, the aim is to find a state sequence that best explains the real observations. For this, a new variable γ is defined in terms of the forward (α) and backward (β) variables of the Baum-Welch algorithm:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{M} \alpha_k(i)\beta_k(i)} \tag{6.46}$$

where

$$\sum_{i=1}^{M} \gamma_t(i) = 1 \tag{6.47}$$

The new variable, $\gamma_t(i)$ represents the probability of being in state i at time t. The size of the matrix γ is $T \times M$. One can then find the most likely state at time t using $\gamma_t(i)$:

$$q_t = \arg\max_{1 \le i \le T} [\gamma_t(i)], \qquad 1 \le t \le T \tag{6.48}$$

While this provides the most likely states for each t, there may be a problem with the state sequence obtained from this algorithm, as the algorithm ignores the probability of occurrence of sequences of states. This can be remedied by using the Viterbi algorithm. Further details of these algorithms can be found in [240]. Many algorithms are also developed as Matlab® toolboxes, a notable one being the toolbox by Thorvaldsen [295] that focuses on the solution of problems in bioinformatics.

6.5 Wavelet-Domain Hidden Markov Models

In DWT, the scaling coefficients are decomposed iteratively at each scale (Figure 6.7), clearly showing the dependency between adjacent scales. For orthogonal wavelet decomposition, it is expected that the wavelet coefficients are uncorrelated between scales. However, for most practical applications, there is a residual dependency after the signal decomposition, even though the dependency of the wavelet coefficients may be local. This means that, for scaling and wavelet coefficients, there exists a dependency *within* and *across* the scales. This is consistent with the clustering and persistence properties of the wavelet coefficients [43] that state that for a large (small) wavelet coefficient, the adjacent coefficients are also likely to be large (small) and that such values propagate across scales.

Given the dependency of the wavelet coefficients, one still has to find the appropriate framework for modeling their probability density functions. A Gaussian model is not appropriate since the wavelet decomposition tends to produce a large number of small coefficients and a small number of

large coefficients, the very property that one takes advantage of in data compression and denoising. Alternatively, the marginal probability of each coefficient can be represented by a mixture density. Instead of assigning a statistical model to wavelet coefficients, Crouse *et al.* [43] suggest assigning a set of *states* to each coefficient and then associating a probability density function with each state, $f(w|S)$. Here, one can choose a two-state model in which the coefficients can belong either to a high-variance state, $f(w|S = 1)$, or to a low-variance state, $f(w|S = 2)$. This yields a two-state zero-mean Gaussian mixture model. It should be noted that to enhance the fidelity of the fit [279], more complex mixture models (even with nonzero means) can also be used, naturally at the expense of increased computational burden. This framework also allows the use of non-Gaussian densities [245].

Next, the dependencies among the wavelet coefficients need to be defined. Given the persistence and clustering properties alluded to earlier, it appears logical to assume Markovian dependencies between the adjacent state variables (not the wavelet coefficients). Such a structure gives rise to the hidden Markov trees (HMTs). Note that this statistical model, as suggested by Crouse *et al.* [43], can also be used to describe dependencies among the scaling coefficients (albeit with nonzero means). For this latter case, Gaussian mixture models can reasonably explain the salient distributions of the scaling coefficients. Here, the modeling of wavelet coefficient will be considered for simplicity.

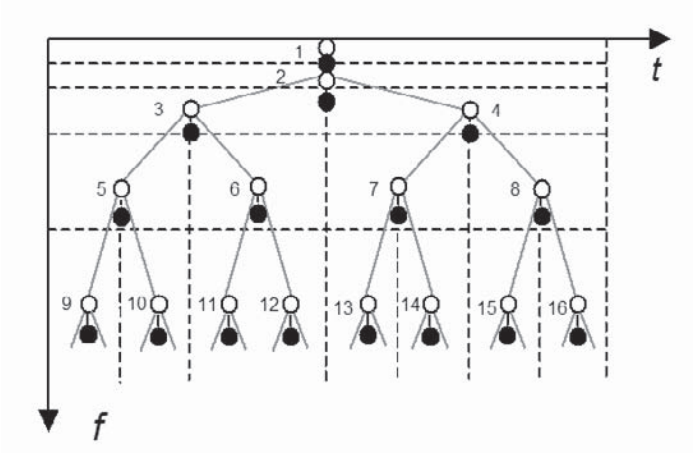


Figure 6.23. Tree structure HMM in wavelet domain as suggested in [43].

As shown in Figure 6.23, Crouse *et al.* [43] proposed a model, where the underlying backbone becomes a tree structure, and the Markov property

exists from the root to the leaves through the branches. The HMT model is specified via the parameters (for the node i), μ_i^m, σ_i^m and the initial, $\pi_i^m = P(S_i = m)$, and the transition probabilities, $a_i^{m,n} = P(S_i = m|S_{\rho(i)} = n)$. Here m and n denote the two states. The subscript $\rho(i)$ refers to the parent node, hence $S_{\rho(i)}$ is the parent state. Consequently, the HMT model is defined via the following parameters,

- $\pi_1^m = P(S_1 = m)$ is the probability mass function for the first node.
- $a_i^{m,n} = P(S_i = m|S_{\rho(i)} = n)$ is the conditional probability that S_i is in state m given its parent $S_{\rho(i)}$ is in state n.
- μ_i^m and σ_i^m are the conditional mean and the standard deviation, respectively, of the wavelet coefficient w_i at the ith node, given S_i is in state m, with $f(w_i|S_i = m)$.

The training problem determines the set of model parameters given above for an observed set of wavelet coefficients. In other words, one first obtains the wavelet coefficients for the time series data that we are interested in and then, the model parameters that best explain the observed data are found by using the maximum likelihood principle. The expectation maximization (EM) approach that jointly estimates the model parameters and the hidden state probabilities is used. This is essentially an upward and downward EM method, which is extended from the Baum-Welch method developed for the chain structure HMM [43, 286].

For a limited amount of training data, to avoid over-fitting, a robust training result can be achieved by assuming an identical distribution for a certain number of nodes, referred to as *tying*. Tying can be applied within and across the scales and increases the number of training data for a certain distribution in the model by simplifying the model structure. Simply put, tying indicates that a certain number of nodes share the same statistical distribution, the same number of states and the same distribution parameters. In signal denoising, one is interested in the shrinkage of noisy components, and the noise components are assumed to be identically distributed, therefore tying can significantly help capture such statistical features. In trend analysis, however, the signal trend plays a more important role in characterizing the process failure, and thus, tying may distort the trend characteristics and will not be employed in the studies presented in this book.

6.6 Summary

In this chapter, several signal characterization and modelling methods have been introduced and discussed. It was shown that the wavelet transforma-

tion provides a time-frequency localization of a signal, allowing for the detection of varying signal characteristics manifested by changing frequency behavior over time. The use of wavelet transformation in signal denoising has been demonstrated, especially in the context of outlier removal and robust filtering. While this chapter focused on one-dimensional signals, wavelet transformation can also be extended to two-dimensional signals (i.e., images) where one can perform similar denoising and feature extraction tasks. In Chapter 10, the denoising implementation will be illustrated in an example concerning the full sheet profile in a paper machine. Wavelet transformation and subsequent feature extraction of image data have been studied for many years [285] and a recent direction is the study of nanoscale features in atomic force microscopy (AFM) images [27, 80, 180]. The use of hidden Markov models allows the representation of process signals via probabilistic models and, when combined with triangular episodes and the discrete wavelet transformation, facilitates the expression of specific signal characteristics. This forms the basis of trend detection and fault diagnosis strategies to be discussed in Chapter 7, next.

7

Process Fault Diagnosis

The widespread availability of Distributed Control Systems (DCS) not only provides the framework for advanced control applications but also greatly facilitates the continuous monitoring of chemical processes to maintain safe and profitable plant operations. In most facilities, the plant operators are asked to manage the operation in such a way as to ensure optimal production levels, while attending to occasional alarm situations that may result from equipment malfunctions. It is critical to identify such abnormal situations in a timely manner as there may be a potential for a safety hazard that may affect not only the plant and its personnel but also the surrounding communities. Most operators traditionally relied on personal expertise for such a task, and in some cases, the events exceeded the capabilities of any human operator, thus leaving the plant vulnerable to costly shutdowns and, in the worst case scenario, to possibly fatal accidents [216]. Today, human expertise is complemented by computerized support systems that comprise various data analysis and interpretation strategies that can provide guidance to the plant personnel for handling abnormal situations. The key component of such a system is *fault detection and diagnosis* (FDD) that monitors the occurrence of process failures and identifies their root causes.

7.1 Fault Diagnosis Using Triangular Episodes and HMMs

This section builds on the techniques described in Sections 6.2 and 6.3 to offer a strategy for process trend analysis (Figure 7.1). The problem can be stated simply as follows: given a set of known models of the process operating conditions, determine the likelihood of a new set of observations.

The first step of the analysis is the training where Hidden Markov Models (HMMs) representing various operating behaviors are trained using labeled historical data from the process. In this section, three broad operat-

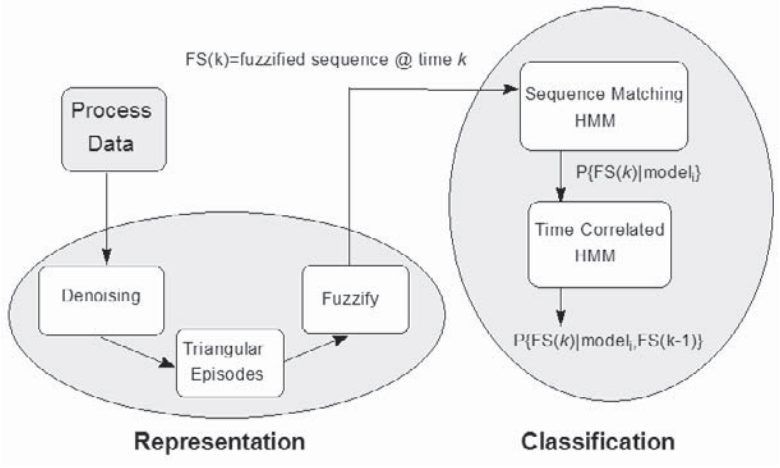


Figure 7.1. The trend analysis strategy using HMMs. The process information (measurement) at a time instant k is first expressed as a fuzzified sequence (FS) and then processed through a classification step. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

ing classes, namely, normal, abnormal and intermediate (transition between normal and abnormal), are used for labeling. The training step starts by segmenting the labeled time series in windows, and an overlapping moving window (slice) is defined for the time series signal that will be analyzed for process trends. The moving window enables the expression of the process trend for a discrete set of windows. The choice of the window length and the overlap period is problem dependent and will be discussed in the case studies. Next, the time series in the selected window is subjected to denoising to eliminate any random behavior and to facilitate the subsequent construction of triangular episodes. Any filtering technique can be used in this step. The smoothed signal is then converted into semi-qualitative form by using the triangular episodes. Finally, this semi-qualitative sequence is transformed into a purely qualitative form by fuzzification of the quantitative descriptors of the triangular episodes. With the data now being purely symbolic in the form of a sequence of letters, two tandem HMM-based classification methods are trained to determine whether the window can be assigned to normal, abnormal or intermediate classes. The first HMM-based step classifies each time series segment (window) as being explained by normal, abnormal or intermediate HMMs. In this step, each window is treated as if it were independent of windows that come before it or that follow it. The second HMM-based classifier accounts for the temporal cor-

relation among the adjacent windows and creates the final assignment.

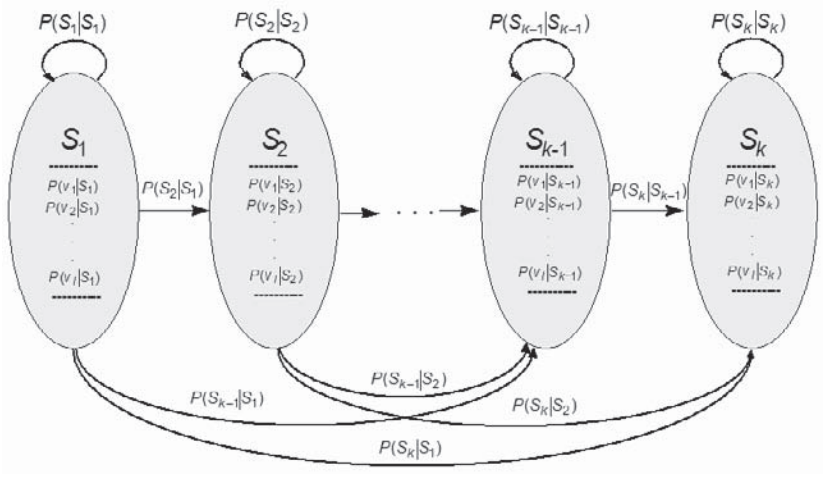


Figure 7.2. A left-to-right HMM. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

The first HMM-based classifier (sequence matching) uses a subclass of HMMs called the left-to-right HMMs [240] (Figure 7.2). These models only allow transitions to themselves or to the states to their right and the model must begin in the first state. The output for each state will be based on the 63-character alphabet of the fuzzy triangular representation. Starting from the first state, a sequence of varying lengths can be modeled using different combinations of state transitions, yielding a set of HMMs to represent the model classes (e.g., normal, abnormal and intermediate).

The structure of the second HMM-based classifier (time-correlated) is the three-state HMM given in Figure 7.3. Instead of using the fuzzy triangular episode alphabet as the output from the state, this classifier directly uses the probability of the sequence that was calculated using the sequence matching HMM and determines the probability of the class based on this current window and the window of data just prior to it. The information from the current window is utilized as well as the temporal information from the past windows to calculate a corrected probability based upon the knowledge of how the entire sequence has propagated up to the current window [277].

Once the relevant HMMs are trained, the trend analysis is carried out on the newly observed time series in real-time. The time series is windowed and smoothed before the signal in the window can be represented in the

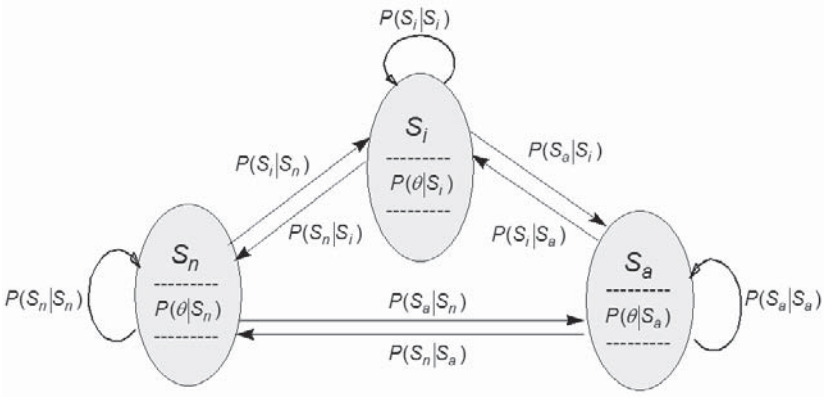


Figure 7.3. The three-state HMM, where the subscripts i, n and a denote intermediate, normal and abnormal conditions, respectively. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

symbolic form. Then, the sequence of letters in the window is classified using the EM algorithm through two classifiers. Consequently, when an unknown sequence is evaluated by each HMM classifier, the class of the sequence will correspond to the model with the greatest probability (Figure 7.1).

The method will be illustrated by two case studies next.

7.1.1 CSTR Simulation

The method has been demonstrated on a continuous stirred tank reactor (CSTR) simulation to identify an abnormal inlet concentration disturbance [340]. The jacketed CSTR, in which an exothermic reaction takes place, is under level and temperature control. An important process variable is the coolant flow rate through the jacket, that is related to the amount of heat produced in the CSTR, and it indirectly characterizes the state of the process. This variable will be monitored in this classification scheme.

The trend analysis strategy will be shown to be able to differentiate between normal and abnormal responses of the coolant flow rate and is similar to the example used in the paper by Whiteley and Davis [325]. Here, three categories of classification are considered: normal, intermediate and abnormal. An intermediate class represents a window of data that can move into the normal or abnormal classes in the next window and no definitive decision between normal and abnormal can be made during that specific time period. For normal operation, the system is able to handle a $\pm 5\%$

fluctuation in the temperature of the inlet feed stream and maintain the outlet concentration at a predetermined quality boundary. An abnormal situation occurs when an unmeasured disturbance in the inlet feed concentration develops in addition to the inlet feed temperature disturbance. The data collected consists of 50 simulations corresponding to 25 different steady-state operating points of normal and abnormal simulations. The simulations were about 1 *hr* long and the data were sampled at 0.1 *min* intervals.

In half of the simulations, a step increase of 5% in the inlet feed temperature was introduced; this change is deemed to be normal and can be easily handled by the feedback control system. In the remaining 25 simulations, this step change was used in addition to a 5% increase in the feed concentration, resulting in an abnormal process trend that cannot be handled by the control system and leading to off-specification product.

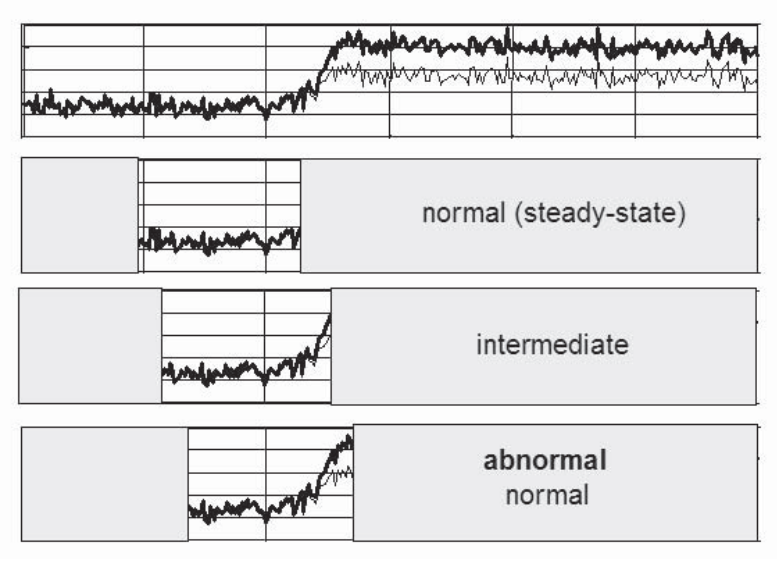


Figure 7.4. A training set for the CSTR example. The first window at the top shows the entire time series while the others indicate the moving window. Thicker line indicates the abnormal trend. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

An example of the training classification is displayed in Figure 7.4 where normal and abnormal situation simulations are also superimposed. The second window shows data progressing at a normal steady-state mode. When a change in the system occurs, the coolant flow rate reacts to compensate

for this deviation from the original steady-state. Initially, because there is no indication of whether this response is normal or abnormal based on the current information (third window), this window is considered to be in an intermediate mode. Not until later can one differentiate between normal and abnormal trends.

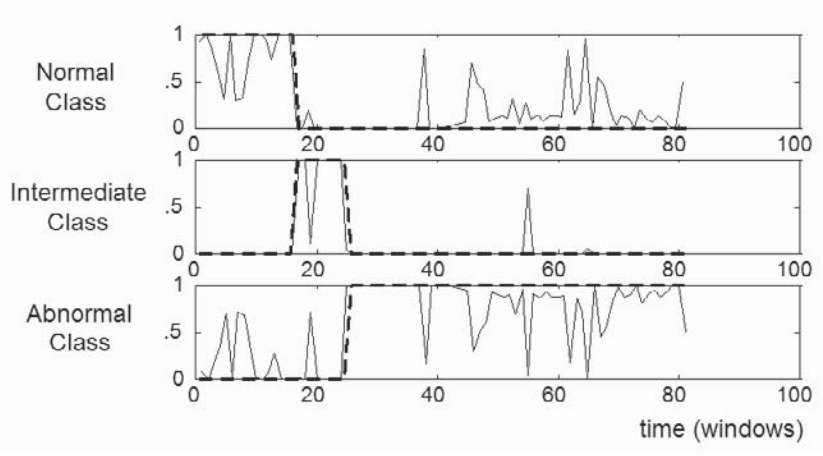


Figure 7.5. The normalized probabilities after the sequence matching HMM step. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

As part of the analysis, each signal was converted into individual windows with a length of 6.4 *min* and the window moves in 36 *sec* intervals. This results in 81 windows to yield a total of 4050 windows for the overall simulation. All 4050 windows were converted into symbolic sequences using fuzzified triangular representation as discussed before; 3159 of the sequences were used to train the HMMs, and the rest for evaluation. Three sequence matching HMMs were created and trained using sequences belonging to the particular event class. The evaluation set of sequences consists of five normal and six abnormal simulations. Figure 7.5 displays the normalized probabilities generated from the three sequence matching HMMs, from one of the abnormal event simulations. The dark solid lines represent the true probability while the dashed lines represent the probability calculated by the HMM. The y-axis for each plot is the probability (0 to 1) of belonging to that class and the x-axis represents the time in terms of the number of windows. The simulation begins in the normal state and the disturbance is introduced at window 17. The abnormal coolant flow rate should be detected at window 26. The erratic behavior of the probability assignments

is expected at this step since the windowed sequences are difficult to distinguish between abnormal and normal situations due to the similarity of the local responses.

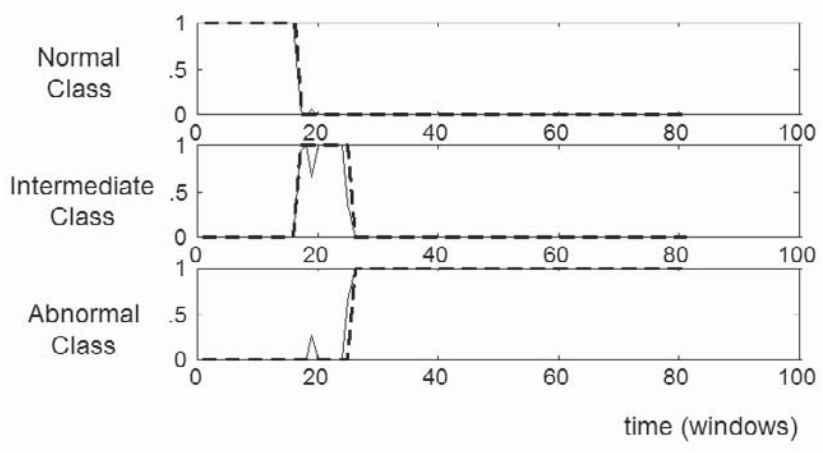


Figure 7.6. The normalized probabilities after the time-correlated HMM step. Reprinted from [340]. Copyright © 1998 with permission from Elsevier.

The time-correlated HMM is utilized to associate the individual sequences and eliminate the ambiguous assignments. When the time-correlated HMM is applied to the sequence, there is a dramatic increase in the prediction time-correlated probabilities of the true classes (Figure 7.6).

7.1.2 Vacuum Column

The vacuum column studied here is associated with the lubrication unit in the Mizushima Refinery of the Japan Energy Corporation [341]. The goal is to identify the weeping condition where the liquid is drained through the perforations due to low gas flow rates and hence causes instability in the operation of the column. The analysis uses temperature measurements from the tray 12 from the bottom of the column, T127, which has been determined by the operators to capture the weeping dynamics. Figure 7.7 depicts this temperature measurement corresponding to the normal condition and the weeping conditions.

The window length for the temperature time series is taken as 64 *min* and the window moves in 4 *min* intervals. For the test case, initially, the process is assumed to be operating normally. At the 54th window, a

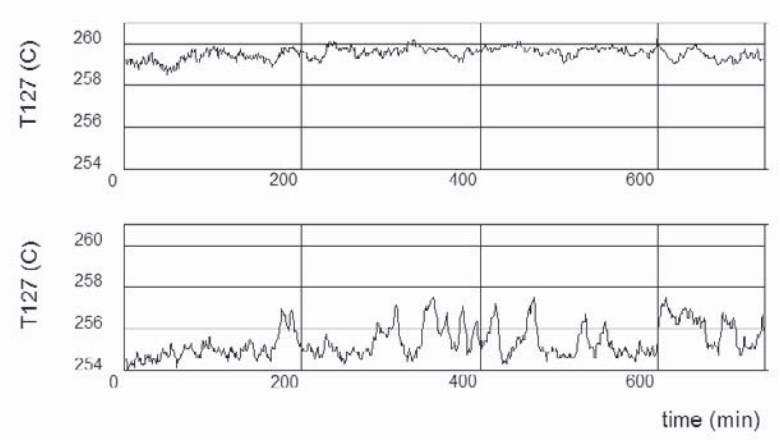


Figure 7.7. The temperature measurement at tray 12 for the vacuum column. The top figure represents the normal conditions and the bottom figure represents the weeping conditions.

weeping condition is detected that lasts until window 218 when the normal operation is recovered. A second weeping condition begins at window 273 and lasts for about 164 window lengths.

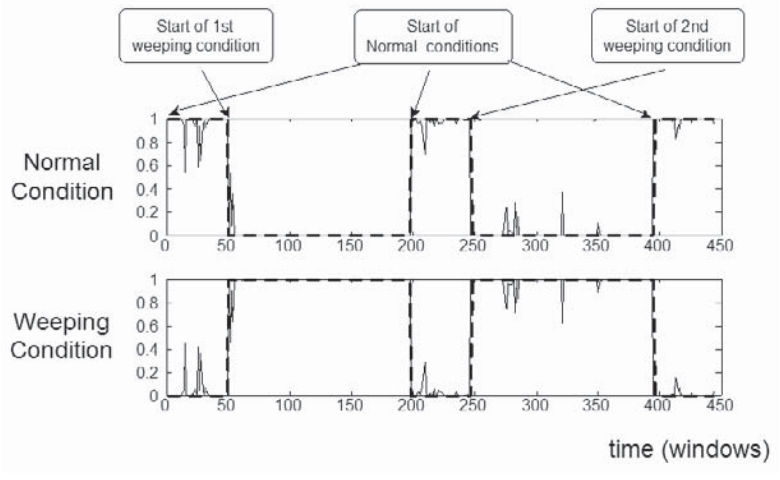


Figure 7.8. The probability of classification after the sequence matching HMM step.

Two sets of HMMs were trained associated with normal and weeping event classes using historical plant data and the test signal is evaluated as before. Figure 7.8 shows the probabilities generated from the sequence matching HMM for the normal and the weeping conditions. Classification after the sequence matching HMMs indicates a 8.5% misclassification of the true class of the process versus the predicted class from the sequence matching HMM. This percentage is calculated by associating a class to each window by choosing the sequence matching HMM with the higher probability and then comparing the results with the known classification.

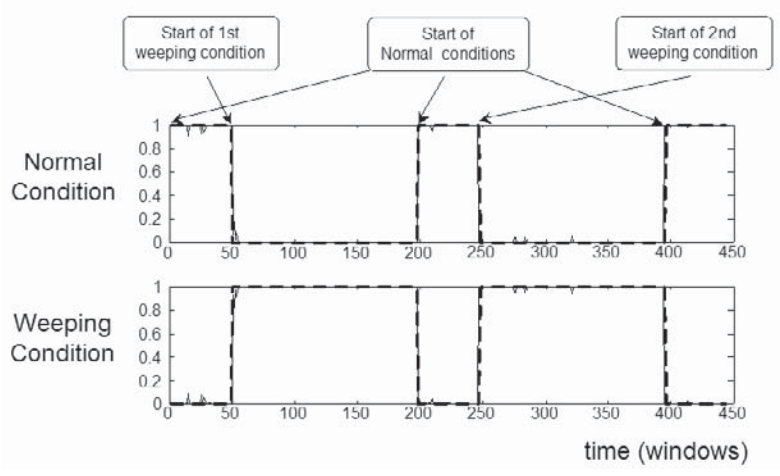


Figure 7.9. The final probability of classification after the temporal correlation HMM step.

When the time correlated HMM is introduced and the probabilities are re-calculated, the results show a significant improvement (Figure 7.9). The misclassification rate is reduced to 3.9%.

7.2 Fault Diagnosis Using Wavelet-Domain HMMs

A trend analysis strategy is proposed that takes advantage of the wavelet-domain hidden Markov trees (HMTs) for constructing statistical models of wavelets (see Section 6.5). Figure 7.10 depicts the strategy that can be used to detect and classify faulty (abnormal) situations. As before, in the training phase, time series data collected under various conditions are

used to develop models. The monitoring phase, then, considers the on-line signal(s) and determines the model that best explains it, thus classifying the event associated with the model.

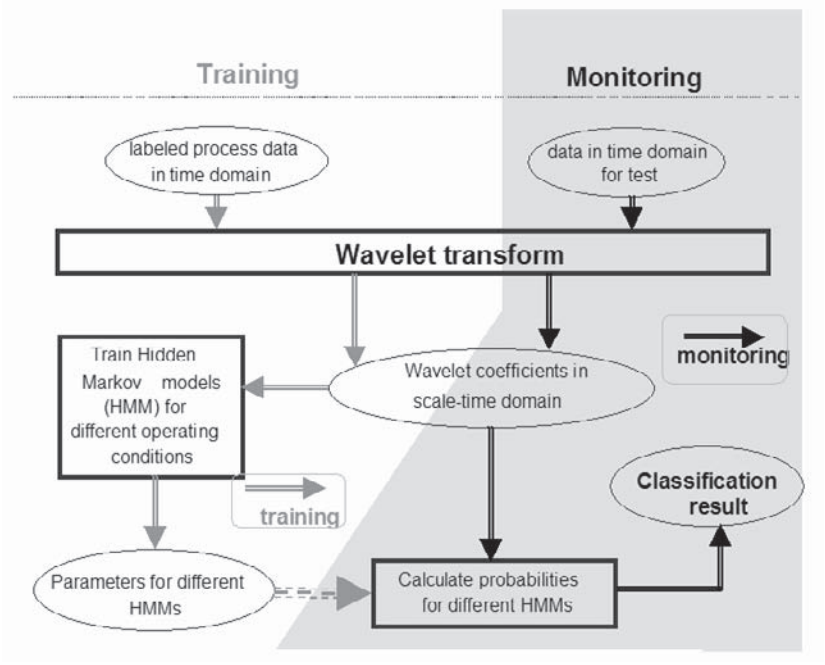


Figure 7.10. The trend analysis strategy using wavelet-domain HMMs. From [286], reproduced with permission. Copyright © 2003 AIChE.

The time series data are represented in the wavelet domain in the form of scaling and wavelet coefficients. Ideally, modeling all these coefficients can glean all the information regarding the observed process operating condition, but will result in a large model tree structure, and increase the computational effort in the training phase. It must be noted that for different operating conditions, the scaling coefficients (approximation) and the wavelet coefficients (detail) play different roles. Thus, for a specific trend analysis application, a different set of coefficients may be chosen, leading to a trade-off between classification accuracy and computational cost. Undoubtedly, such a decision can be made *a priori* based on the nature of the fault.

The HMT model can also be extended from the single-tree structure to the multi-tree structure (MHMT), which is then used in the multivari-

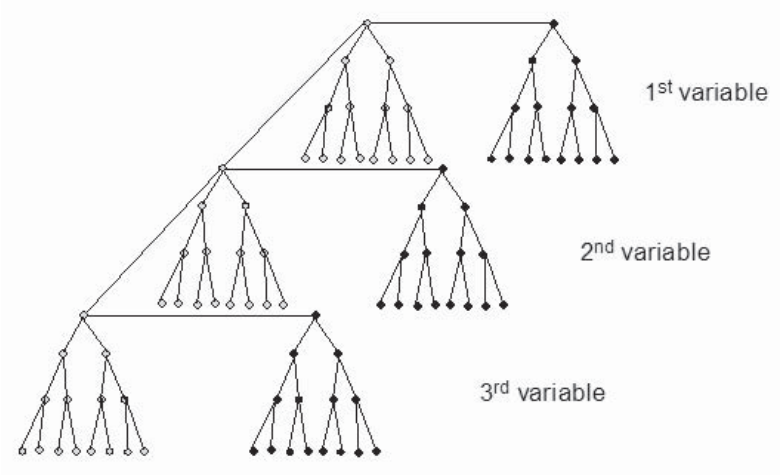


Figure 7.11. A three-variable HMT showing both scaling and wavelet coefficient trees. From [286], reproduced with permission. Copyright © 2003 AIChE.

ate trend analysis. To illustrate the concept, a three variable multi-tree structure is depicted in Figure 7.11. For each measured variable, there are two tree structures joined together, one constructed by the scaling coefficients (light nodes) and the other by the wavelet coefficients (dark nodes). The root nodes of each tree from a single variable are connected together. The joint structure can be used for any single variable modeling, which includes all frequency (scale) components in this specific variable. To limit the computational complexity, only the tree of scaling coefficients is used. In process monitoring, magnitude of a variable contains more information of the process, which is mainly described by its corresponding scaling coefficients. Also scaling coefficients at each scale represent a smooth version of the signal with a different resolution, therefore it is not difficult to argue that the scaling coefficients are sufficient for most of the process monitoring applications.

The root nodes of each tree structure are connected, corresponding to each variable under consideration. In each single tree, the deterministic trend information and the random factors are all accounted for. The rationale behind using the multivariate tree structure is to be able to capture the correlations among variables. Here, the connection among variables is arbitrary, and the apparent parent-child connection does not really imply the parent-child dependence, but it is just a way to model the relation be-

tween two nodes. In principle, the multi-tree structure can be expanded indefinitely, but the computational complexity may restrict the number of variables.

The EM algorithm for the single-variable case also applies in a straightforward manner to the multi-tree structure, since the binary structure remains unchanged in this structure expansion. For on-line process monitoring, moving window is again used in the multivariable case. The complexity of the computation increases by the number of trees compared to the single tree at the same window size, but the multivariable system contains more structural information, which makes it possible to reduce the window size, and therefore, keep the computational complexity the same as or less than the single variable case. In principle, one can use different window sizes for variables, but the same number of data points needs to be used to each variable to keep the same weight of contribution to the MHMT from each variable. Using different window sizes results in having a monitoring delay corresponding to the time of the longest window size used in the model, but it may reduce the computational complexity if a longer window size is needed for some of the variables. Longer window size is usually suggested for the slow dynamics, so fewer data points within a certain time period would be enough to characterize the variable trend. Similarly, a smaller window size is preferred for the fast dynamics, so more data points within a certain time period would be needed. The monitoring strategy is carried out in the same manner as in the single-variable case.

While on-line implementation of the trend detection strategy is indeed feasible, the training of HMT models may be rather time consuming and inefficient. We note the following on the implementation issues:

- The presence of local minima encountered in the EM algorithm limits the complexity of the problems that can be tackled.
- The amount of data required for training is problem dependent, and especially when process events have somewhat similar features, more data would be required for training to ensure the development of models that can capture the subtleties associated with each event.
- A smaller window size reduces the computational burden in training and can improve detection time, yet the classification performance may deteriorate as trends may not have developed distinct features in a short time.
- Moreover, combining detail and approximation coefficients to build HMT models would be a natural next step in process trend detection, but this extension is hampered by computational difficulties as mentioned before.

- Furthermore, in the multivariable problem, while three to five variables can be handled relatively easily, one reaches a computational bottleneck for larger problems. This can be possibly resolved by considering some of the new developments in HMM training algorithms [254, 71].

Two case studies are presented next to illustrate this strategy.

7.2.1 *pH* Neutralization Simulation

The simulation of a *pH* neutralization process has been previously studied by Galán *et al.* [81]. An acid stream (HCl solution) and an alkaline stream ($NaOH$ and $NaHCO_3$ solution) are fed to a 2.5 L constant volume, well-mixed tank, where the *pH* is measured through a sensor located directly in the tank. The *pH* value is maintained at 7.0 by a PI controller.

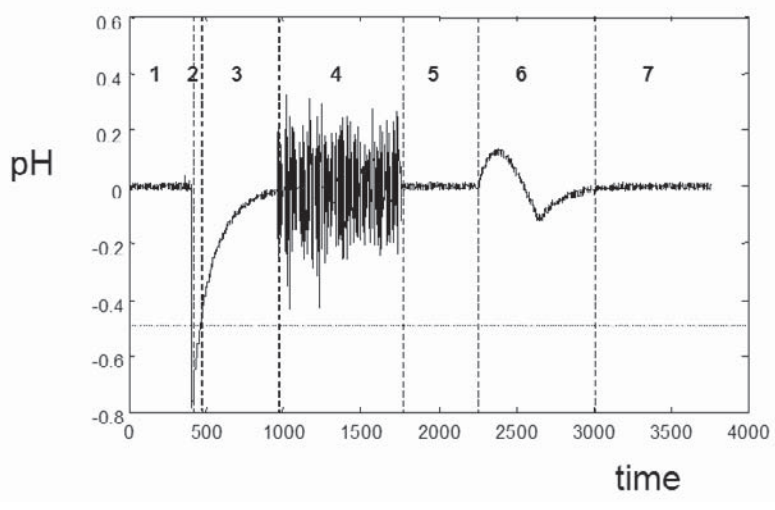


Figure 7.12. The test *pH* response showing regions of operation. From [286], reproduced with permission. Copyright © 2003 AIChE.

When an unexpected event occurs, the controller may not maintain the *pH* value within the allowed range of operation and its performance degrades, thus resulting in an abnormal (faulty) operating condition. Here, four distinct situations, other than the normal operating condition, will be considered as follows:

- Abnormal Condition I (AI): The *pH* value shows a sustained deviation of more than ∓ 0.5 (region 2 in Figure 7.12), which could result from

a large and sudden change in either the flow or the pH value of the acid stream as a result of changes in the upstream process.

- Intermediate (I): The pH value indicates deviations (region 3 in Figure 7.12), which could result from the same source as Abnormal I, but the deviation remains within ∓ 0.5. This region may act as a warning (buffer zone) for an imminent change in normal operating conditions.

- Abnormal II (AII): The pH value exhibits high amplitude, high frequency oscillations (region 4 in Figure 7.12), which could be the outcome of a sensor failure or other equipment malfunction, such as pump cavitation.

- Abnormal III (AIII): The pH value increases slowly and reaches a maximum point, then comes back to 7.0 slowly (region 6 in Figure 7.12), which could be the result of a temporary sensor drift.

All other operation around $pH = 7$ are considered as normal (N).

A moving window is used to analyze the data as before, and especially since only a limited number of data points can be considered at one time for the wavelet decomposition. If a short window size is chosen, one may capture process changes quickly, but the window may not contain enough information to sufficiently reflect the current process state, thus generating ambiguous classifications. Longer window sizes can consider more information, which is helpful to recognize the process trend, but may lead to large time delays for the detection and classification of trends. Here, an adaptive window size is implemented that uses a short window size for rapidly changing data and a long window size for longer lasting phenomena, based on the spectral analysis of the signal. The window is moved every sampling time. Sixteen separate simulation runs are used, fifteen for training purposes and the last one (depicted in Figure 7.12) for testing the methodology. There are 1200 data points in each simulation set for training and 3755 data points in the simulation for testing. The data were sampled every 45 *sec*.

In this example, the Haar wavelet is used with a single scaling tree and the scaling coefficients are modeled using a two-component ($M = 2$) HMT model with nonzero mixture means. The models were trained using multiple observations (without tying). The wavelet coefficients were not modeled since the approximate signal contains more distinguishing features among the studied abnormalities, as the primary goal in this study is to detect if the pH deviation is beyond the tolerable limit ± 0.5. In other words, the decision depends more on the information provided by the scaling coefficients than the wavelet coefficient.

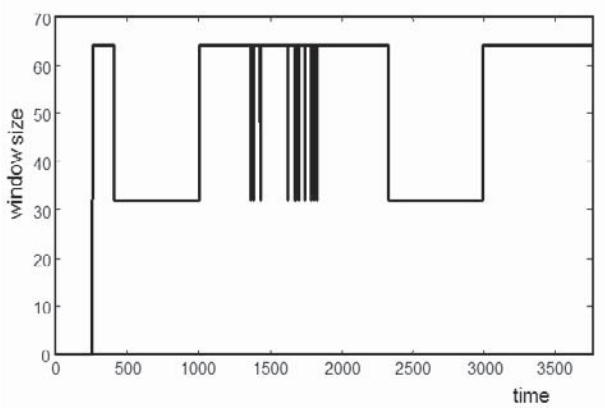


Figure 7.13. The variation of window sizes as a result of the adaptive algorithm. From [286], reproduced with permission. Copyright © 2003 AIChE.

To keep the analysis simple, two window lengths are considered, a 32-point window and a 64-point window. After training, five sets of models for different operating conditions under 32- and 64-point windows are obtained. Spectral analysis based on Thomson's multitaper method [292] was used to differentiate the short- and long-term signal behavior, then the appropriate window size is chosen to analyze the test signal, since the high frequency components are more important in short-term behavior and the low-frequency components dominate in long-term behavior. Window size selection for the test signal is depicted in Figure 7.13.

Figure 7.14 depicts the classification result from likelihood determination, comparing the true and calculated probabilities. It can be observed that the HMT model yields the correct classification for most part of the test signal. Following observations can be made:

- The abnormal condition AI (disturbance) and the abnormal condition AIII (sensor drift) can be recognized clearly (Figures 7.14c, 7.14d).

- The instances of misclassification between sensor noise (AII) and intermediate operating condition (I) (Figures 7.14b, 7.14e) are notable. As the level of noisy signal momentarily matches the level of the signal in the intermediate region, the method results in misclassification. Yet, since these misclassification instances are rather isolated, the overall trend can still be inferred. Nevertheless, the model may need further training to eliminate such instances.

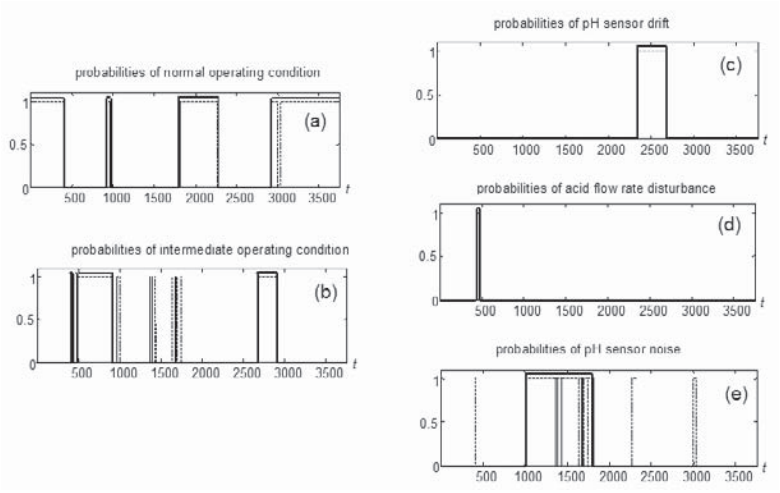


Figure 7.14. The classification results for different classes of operating conditions. Solid lines represent the true probabilities while the light dashed lines are the calculated probabilities. From [286], reproduced with permission. Copyright © 2003 AIChE.

- The brief misclassification between the normal condition (N) and the sensor noise (AII) (Figure 7.14a, 7.14e) is due to the switching of the window size from 32 to 64, and as the number of data points increases suddenly, this causes the method to consider the new signal section as noisy. As the window moves forward, this misclassification error is corrected immediately.

7.2.2 CSTR Simulation

A constant volume, continuous stirred tank reactor (CSTR) is simulated to demonstrate the multivariate trend analysis strategy. In the CSTR, a single irreversible, exothermic reaction, $A \rightarrow B$, is assumed to occur and the model equations are given in [232]. The disturbances are the feed concentration and temperature, c_{Af} and T_f, respectively, and the control outputs are the tank concentration and temperature, c_A and T, respectively. The two outputs are controlled by two PI controllers via feed and coolant flow rates, q_f and q_c, respectively. White Gaussian noise is added to the outputs to simulate the real process signal. The sampling interval is assumed to be $0.1min$. The normal operating condition (N) is taken as the steady-state, $c_{As} = c_A(0) = 8.235 \times 10^{-2} mol/L; T_s = T(0) = 441.81K$.

Any deviations from the assumed normal operating condition are considered as abnormal operating conditions, which would need the operator's attention. Here, four abnormal operating conditions are defined and two manipulated variables, q_f and q_c, are monitored. Tested cases are summarized below.

- In response to a sudden increase in inlet concentration c_{Af} (AI), q_f decreases and q_c increases.
- In response to a sudden decrease in inlet concentration c_{Af} (AII), q_f increases and q_c decreases.
- A sudden decrease in the pre-exponential factor (AIII) due to an unmeasured component variation in the inlet flow, both q_f and q_c decrease.
- The flow sensor for q_f drifts high (AIV) without affecting the process, so other variables, including q_c, remain unchanged.

Three simulations of each operating condition are carried out for model training. One simulation under each operating condition is used to test the monitoring result. Each test data set includes the transient process from normal operating condition to the abnormal operating condition. The results use an 8-point window and the Haar wavelet. The test results are shown in Figures 7.15, 7.16, 7.17, and 7.18. To simplify the model structure and therefore to reduce the computational effort, only scaling coefficients are considered in this case study. Following remarks are in order:

- The method classifies AI correctly (Figure 7.15) with a small delay. During the delay, AI is temporally misclassified as AIV due to the similar response of these two abnormalities in the beginning.
- The method classifies AII correctly when the process nears steady-state (Figure 7.16). There is a brief period of misclassification during the transient between AII and AIII due to the similar responses of the two monitored variables. With more process information, the misclassification can be possibly avoided. As in (a), there is a temporary misclassification between AII and AIV.
- The method classifies AIII correctly when process nears steady-state (Figure 7.17). There is a short period of misclassification between N and AIII during the transient, which is considered as N initially.
- The method classifies AIV correctly (Figure 7.18).

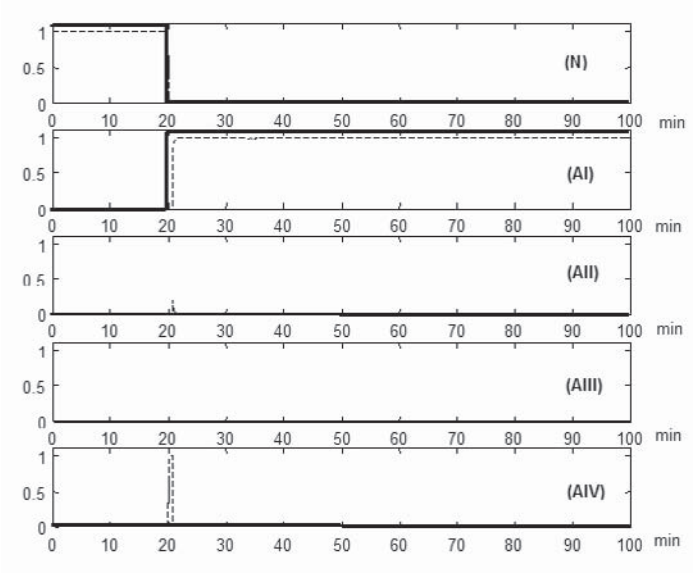


Figure 7.15. The classification of the AI abnormality. Each figure corresponds to the probability assignment made with respect to a model class. The solid line is the expected probability and the light dashed line is the calculated probability. From [286], reproduced with permission. Copyright © 2003 AIChE.

From the results above, one can conclude that the MHMT method for trend analysis correctly classifies the different process operating conditions. There are a few ambiguities in the transient region, which result from the similar responses of manipulated variables to different process events. In other words, the information contained in two variables is not enough to immediately discern the transient part of the process. To eliminate such ambiguities, additional variables may be needed in the HMT model. Sun *et al.* [286] have shown the extension of the method to the multivariate case for the CSTR simulation example.

7.3 Fault Diagnosis Using HMMs

Hidden Markov models (HMMs) provide a powerful framework for recognizing patterns in data and diagnosing process faults as shown in the previous sections. Here, another procedure is introduced that is based on the state estimation problem (see Section 6.4.2). The procedure determines first

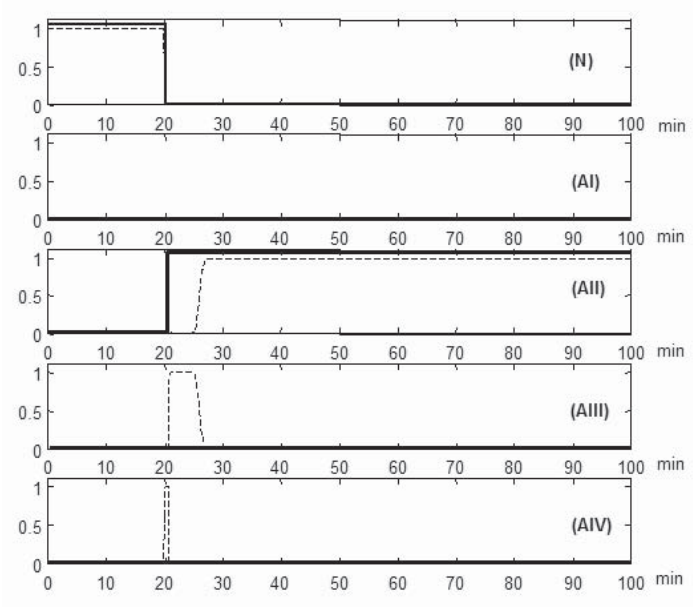


Figure 7.16. The classification of the AII abnormality. From [286], reproduced with permission. Copyright © 2003 AIChE.

whether the mean values of HMM state variables for each measured process variable have changed significantly from their in-control values. Then, by monitoring the changing trends in the HMM states, one can identify the faults that caused the variation in behavior. Here, the whole data set that contains all the faults that need to be diagnosed is used for developing the HMM. The following section demonstrates this strategy.

7.3.1 Case Study of HTST Pasteurization Process

The process of HTST pasteurization (Figure 5.5) is described in detail in Section 5.3. The variables used here are four temperature measurements (^oC) and two PID controller outputs (mA). The hot water temperature, preheater outlet temperature of raw product, holding tube inlet temperature of pasteurized product and holding tube outlet temperature of pasteurized product are the output variables of the process (variables 1-4, respectively). The input variables of the process are the PID controller output to the steam valve (variable 5) that regulates the holding tube inlet temperature of product and the PID controller output to preheater hot wa-

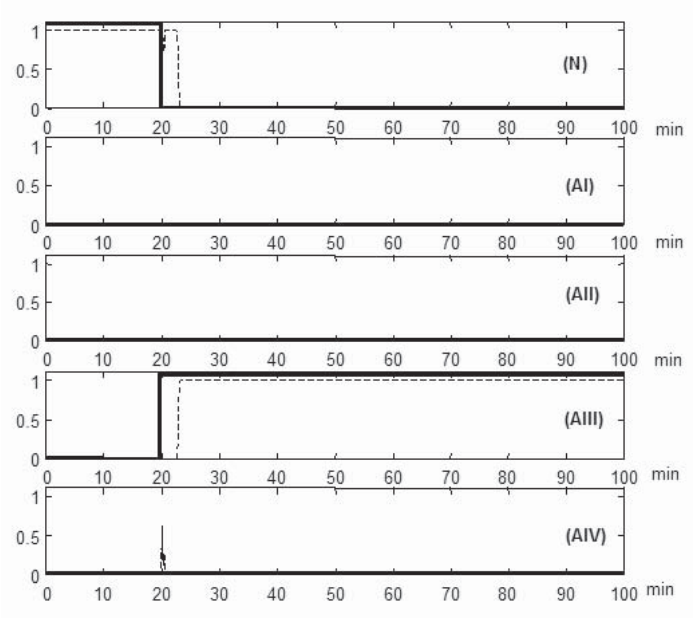


Figure 7.17. The classification of the AIII abnormality. From [286], reproduced with permission. Copyright © 2003 AIChE.

ter valve (variable 6) that regulates the preheater outlet temperature of raw product. A series of sensor and actuator faults were tested on the process, and the experiments were conducted with different fault magnitudes and durations.

Sensor failures are deliberately introduced using the process control computer software. To accomplish this, a real number is introduced to the actual sensor reading, which is transmitted to the computer from the process. Instead of the actual sensor reading, the altered 'reading' is sent to the PID controllers. As the controllers compute the control action based on the false sensor reading, the process receives a false corrective action and the fault originated by the sensors propagates through the system. The magnitude of sensor faults varies between -0.83^oC and 0.83^oC, and their duration changes between 2 *sec* and 30 *sec*.

To implement actuator faults, the controllers are turned off for a specific time period which results in a constant signal being sent to the actuators. During the implementation of this class of faults, when the controlled variables deviate from the set-points, the controller and the actuator cannot respond until the implementation is over.

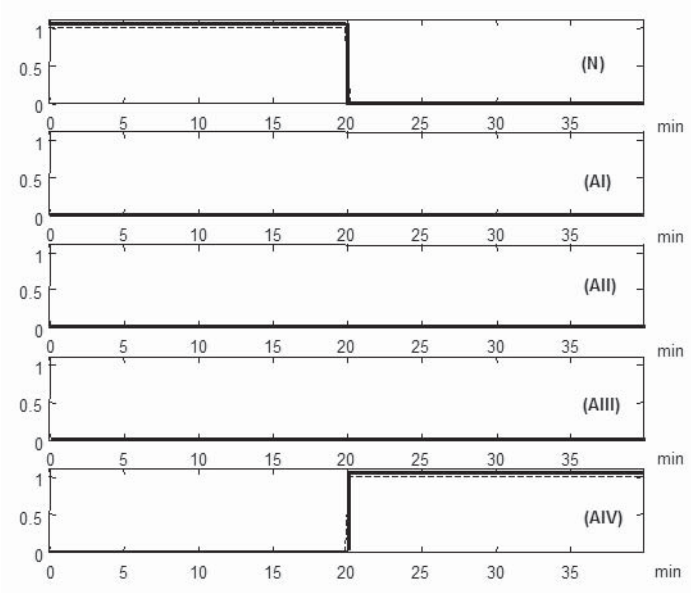


Figure 7.18. The classification of the AIV abnormality. From [286], reproduced with permission. Copyright © 2003 AIChE.

The aim of the fault diagnosis strategy is to determine whether the mean values of HMM state variables for each process variable change significantly compared to in-control HMM states. The increase in probabilities of these states is investigated when corresponding faults occur during operation. After construction of a HMM for the given data sequence, the mean values of M state variables (μ matrix) are checked for an abnormal increase or decrease in the state variables. Then, the probability values which these particular states take during a specific time period are examined using the γ matrix (Eq. 6.46) to determine the time and duration of faults.

First, the performance of HMMs in representing HTST data is assessed using the model residuals and the correlation coefficients of observed and estimated values of process variables. This is done by checking the normality of residuals of some important process variables (e.g., holding tube inlet temperature and steam valve setting, variables 3 and 5, respectively). It appears that the residuals are autocorrelated most of the time. The normality property is affected by the extreme values of faulty signals since the model may not perform well to estimate the measurements at times of fault implementation. If the observation sequence contains many outliers, the residuals will likely not belong to Normal distribution.

Table 7.1. Performance of HMMs with different number of states.

M	SPE	r (holding tube inlet)	r (steam valve)
30	112.87	0.9667	0.9374
50	60.95	0.9890	0.9825
70	33.69	0.9971	0.9969
90	28.57	0.9972	0.9939

As noted before, the number of HMM states (M) is an indicator of model performance. Low M values are not considered for HTST pasteurization data because they possess poor predictive capabilities. On the other hand, large M values lead to increased computational effort and can cause overfitting of the data. Table 7.1 shows the squared prediction errors (SPE) of a data set by HMM with different M values. Table 7.1 also depicts the r values (correlation coefficient) for the holding tube inlet temperature and the steam valve opening. When a large M is used, some states become too specific for certain process behavior. For example, each temperature increase in holding tube inlet sensor can be represented by different states even though there are other faults with the same magnitude causing similar reactions in the system. Real process data may show strong autocorrelation, cross correlation and also include noise. In those cases, HMMs may require high numbers of states to truly capture the process behavior. For the HTST data, 50 states were selected for modeling.

Sensor faults are introduced in the holding tube inlet temperature sensor. This is the controlled variable of the process, thus, any deviation in its measurements causes the controllers to respond to that change and influence process operation. The actuator faults are introduced in the steam control valve. This is the manipulated variable, thus, any fault would cause all process variables to behave abnormally depending on the magnitude and duration of the fault. When there is a fault in the holding tube inlet sensor, the steam valve that supplies the heating medium into the system responds right away. The hot product is then introduced to the holding tube. For consumer health purposes, the temperature at the exit of the holding tube is critical and it is very strongly correlated with the holding tube inlet temperature. Any kind of fault related to both holding tube inlet sensor and steam valve may compromise consumer health and thus needs to be closely monitored.

Figure 7.19 displays the mean values of HMM states of process variables for data collected under normal operating conditions. For example, mean values of states for steam valve signals (variable 5) change between -2 and +2 approximately. Mean values of HMM states of process variables for

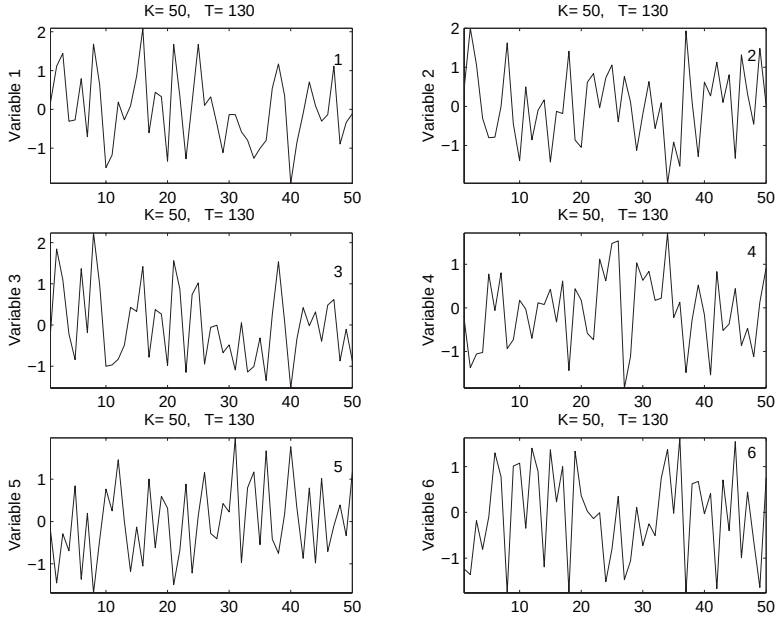


Figure 7.19. Mean values of HMM states for the normal operating condition.

data sequences with steam valve actuator faults are displayed in Figure 7.20 (Case I). Mean values of states for steam valve signals (variable 5) vary between -5 and +7 approximately. Therefore, HMM state variables can be monitored to determine their behavior with respect to their mean values for particular process variables under various operating conditions. Table 7.2 depicts the magnitude, duration and time of occurrence of six different faults implemented to the steam valve for the first case study. The length of data sequence T used for HMM development is 114. Figure 7.20 shows the mean values of HMM states for the data sequence collected under faults implemented to the steam valve as given in Table 7.2. State 36 of steam valve signals (variable 5) in Figure 7.20 has the highest mean value among the HMM states and state 44 has the lowest mean value. Figure 7.21 shows the probability values associated with these particular states. It is clearly seen that state 44 indicates faults in which the steam valve signals are low, i.e., 6 mA. On the other hand, state 36 indicates the faults in which steam valve signals are higher, 12 mA. HMM state 6 has large values among the states of both hot water temperature sensor (variable 1) and holding tube inlet temperature sensor (variable 3) (Figure 7.20). This

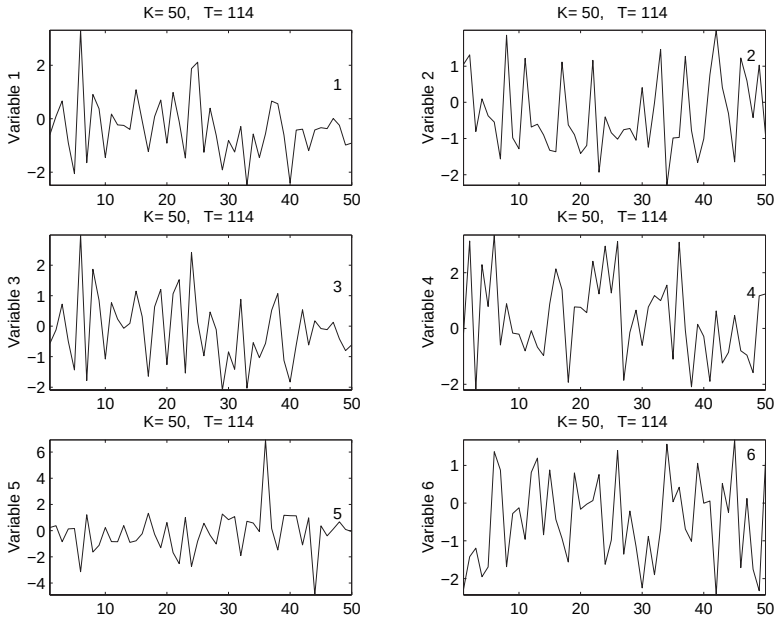


Figure 7.20. Mean values of HMM states for steam valve actuator fault.

state gets a low mean value for steam valve signals (variable 5 in Figure 7.20). As seen in Figure 7.21, state 6 gets high probability value after fault 6 (magnitude of 12 mA with duration of 8 sec in Table 7.2), which causes an increase in both temperature sensors. This is a typical behavior when a fault occurs in the steam valve of the HTST pasteurization system. Since the steam valve opens up to 12 mA and introduces large amounts of steam into the heater, the hot water temperature and consequently the product temperature in holding tube inlet increase.

In the second case study, the faults in holding tube inlet temperature sensor (variable 3) are investigated (Case II, Table 7.3). For the HMM development, the length of data sequence, T, is taken as 131. Figure 7.22 shows the mean values of HMM states for six process variables. State 43 of steam valve signals (variable 5) has the lowest value among all HMM states and the highest value for the holding tube inlet temperature sensor (variable 3). Whenever a fault occurs in the holding tube inlet sensor, the steam valve responds promptly. Consequently, the same HMM state variable represents the changes in both variables. Since the faults in the sensor show positive magnitudes, they cause a reduction in signal magnitudes to the steam valve. Figure 7.23a shows the probability values of state

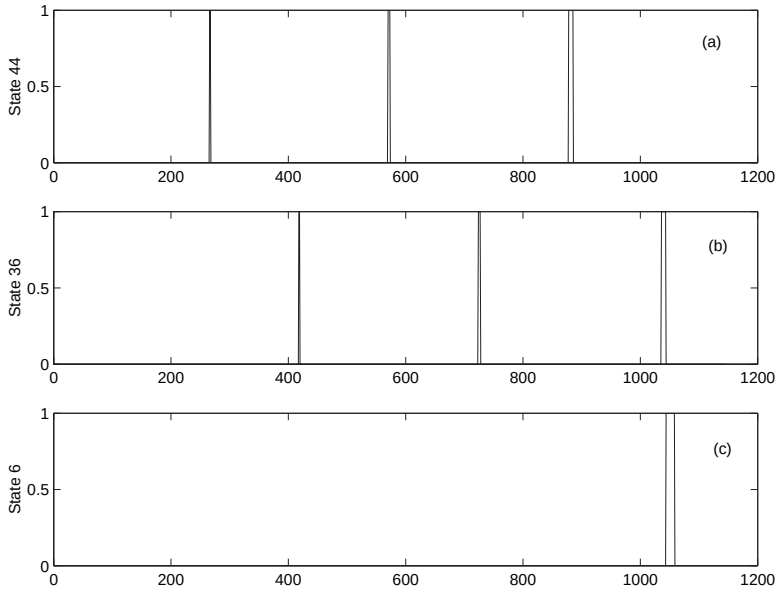


Figure 7.21. Probabilities of states 44, 36 and 6 for steam valve actuator fault.

43 during process operation. State 43 indicates the faults with magnitudes of 0.83^oC, which cause stronger responses than the faults numbered 1, 3, and 5 with fault magnitudes of 0.39^oC in Table 7.3. State 35 of steam valve (variable 5), which has the largest value, indicates all faults except the first one (Figure 7.23b). The probabilities become 1 during the time intervals after the faults occur. This state contributes to situations where the steam valve opens and injects additional steam just after the closing action because of increasing temperature in the holding tube inlet sensor. The second largest mean value belongs to HMM state 13 among the holding tube inlet temperature states. The final plot in Figure 7.23c shows the probability values of state 13 during the operation. This state indicates faults 2, 3, 4, 5 and 6. The first fault in the holding tube inlet is hard to detect since it does not cause significant deviation in any of the process variables. In the case of holding tube inlet sensor faults, the same HMM states contribute to changes in the holding tube inlet temperature measurements and steam valve actuator behavior. This situation is not observed in HMM of data sequences with the faults in steam valve (Case I).

The HMM strategy can be modified by considering a moving window to detect changes with low magnitudes and duration. It has been shown

Table 7.2. Steam valve faults.

Fault	Fault Time (*sec*)	Steam Valve Signal (mA)	Duration (*sec*)
1	266	6.0	2
2	418	12.0	2
3	570	6.0	4
4	724	12.0	4
5	878	6.0	8
6	1036	12.0	8

Table 7.3. Holding tube inlet temperature sensor faults.

Fault	Fault Time (*sec*)	Temperature Signal (^oC)	Duration (*sec*)
1	63	+0.39	2
2	215	+0.83	2
3	367	+0.39	4
4	521	+0.83	4
5	675	+0.39	8
6	833	+0.83	8

by Tokatli and Cinar [296] that such a strategy performs better than fault diagnosis methods that are based on parity-space as well as state-space identification [143].

7.4 Fault Diagnosis Using Contribution Plots

When T^2 or SPE charts exceed their control limits to signal abnormal process operation, variable contributions can be analyzed to determine which variable(s) caused the inflation of the monitoring statistic and initiated the alarm. The variables identified provide valuable information to plant personnel who are responsible for associating these process variables with process equipment or external disturbances that will influence these variables, and diagnosing the source causes for the abnormal plant behavior. The procedure and equations for developing the contribution plots was presented in Section 3.4.

The decomposition technique given in [146] can be extended to the T^2 and SPE_N values of state variables. The state variables are calculated by Eq. 4.67 in which the past data vector is used. When the T^2 or SPE chart of the state variables gives an out-of-control signal, contribution plots can be inspected to find the responsible variable for that signal.

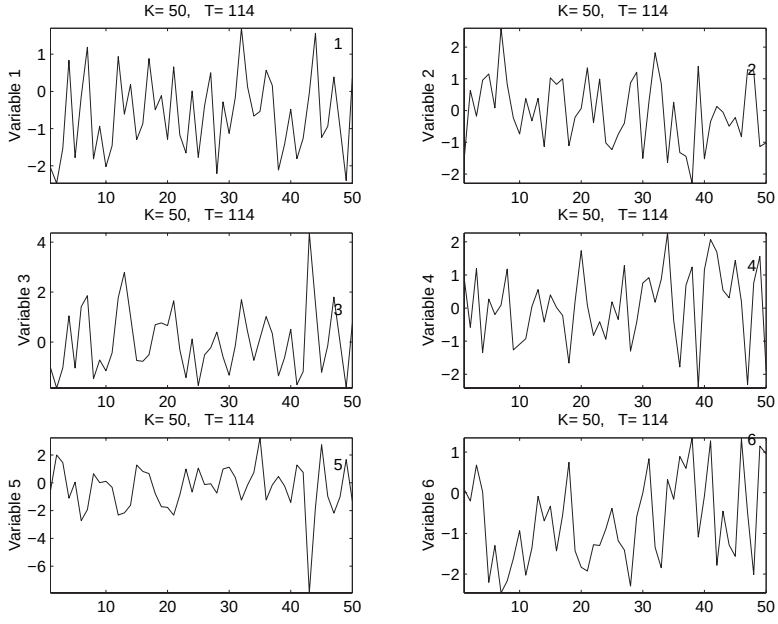


Figure 7.22. Mean values of HMM states for holding tube inlet temperature sensor fault.

The contribution of output (process) variable y_j on state variable x_i at time k is

$$cont_{i,j} = \frac{x_i}{\Sigma_{st_{i,i}}} \mathbf{TRAN}_{i,j} y^-_{j_K} \tag{7.1}$$

TRAN is used to calculate state variable vector $\mathbf{x}$ by utilizing $\mathcal{Y}^-_K$ which is composed of K past values of the process variables. Based on Eq. 4.67, matrix **TRAN** is given as

$$\mathbf{TRAN} = \mathbf{\Sigma}^{1/2} \mathbf{V}^T (\mathbf{R}^-_K)^{1/2} \tag{7.2}$$

The total contribution of variable y_j:

$$CONT^{T^2}_j = \sum_{i=1}^{n} cont_{i,j} \tag{7.3}$$

where $j = 1, \ldots, p$ number of process variables and n is the number of state variables. Unlike the computation of score variables in PCA, the state variables at each time are calculated by using not only the present value of the process variables, but also the past values of the process variables

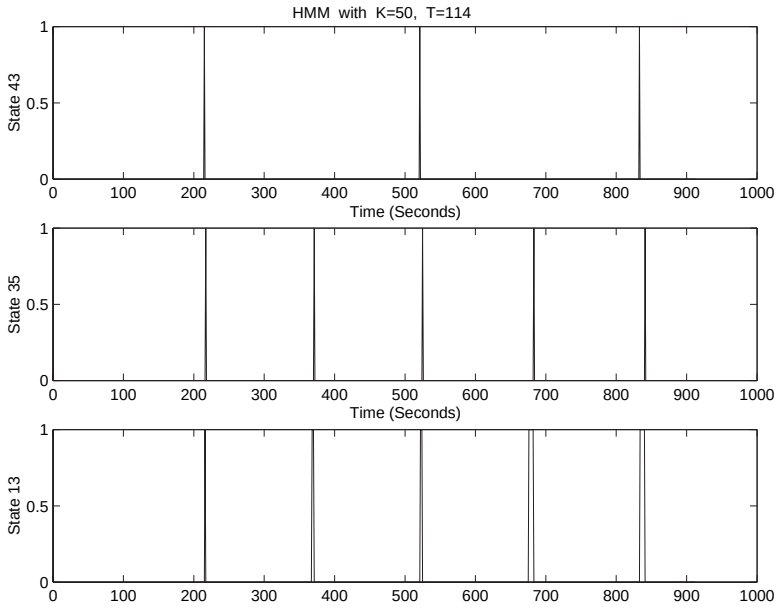


Figure 7.23. Probabilities of state 43, 35, and 13 for holding tube temperature sensor fault.

over the past data window K. The total contribution of each variable on state variable x_j is calculated by summing up all the contribution of the variable coming from its past values. This procedure is repeated for all state variables x_i $(i = 1, \ldots, n)$.

The computed values of the contributions of each process variable and its past values on all the state variables are plotted on a bar chart. The procedure is repeated for all process variables $\mathbf{Y}_j$ $(j = 1, \ldots, p)$. Their contributions are plotted on the same bar plot to decide which variable(s) caused the out-of-control alarm in the multivariate T^2 chart of state variables. Use of state variables in SPM and their contribution plots are introduced and illustrated in [211] and [219], respectively.

The contribution of process variable y_j on the SPE_N statistics at time k is determined by dividing the squared error associated with jth variable by the SPE_N value at time k:

$$CONT_j^{SPE} = \frac{[(e_j - \overline{e}_j)^2 / \Sigma_{e_{j,j}}]}{SPE_N} \tag{7.4}$$

where $\overline{e}_j$ and $\Sigma_{e_{j,j}}$ are the mean and variance of jth error term, respectively.

The contribution of each process variable is determined and plotted on a bar graph to decide which variable(s) caused the inflation of the SPE_N value at that particular time.

Investigating the dynamic pattern of contribution plots is more effective in fault diagnosis rather than the single snapshot of contributions of variables at a particular time. The contributions can be plotted over time, following an alarm signal in MSPM charts. The variation of the contributions over time can also be summarized by plotting the sum of the contributions over a time period [301]. Rapid detection of the variables responsible for inflating the monitoring statistics is necessary because the contributions smear over time as the effects of the abnormality spreads over other variables. On the other hand, inspection of contributions over a period is desirable to filter out instantaneous spurs caused by measurement noise or errors. In the test case summarized below a sequence of contribution plots following the detection of an abnormal situation is given to illustrate the smearing over time.

Example Consider the HTST pasteurization example discussed in Section 5.3. In all contribution plots, the six process variables are hot water, preheater, holding tube inlet and holding tube outlet temperatures, and control signals to steam valve and preheater valve shown as the first, second, third, fourth, fifth and sixth process variable in the horizontal axis, respectively. The vertical axis is the total contribution of each process variable. The same fault is repeated at different times for the same duration but with increasing magnitude. For the steam valve fault, T^2 chart did not alarm the faults or alarmed the faults later than the SPE charts or the alarm signal persisted for shorter periods of time (Table 5.1). The T^2 chart alarmed the third and fourth faults later than the SPE_N chart. The contribution plots of T^2 (Figures 7.24 and 7.25) showed that the holding tube inlet temperature sensor and the hot water temperature sensor (variable 1) caused the alarms. Since the out-of-control situation in T^2 chart is for 2 and 3 sampling times, the information gathered from the contribution plots did not help to diagnose the fault in the steam valve. They did not provide information about the other process variables either. The variable contributions on SPE_N for the first and second faults in the steam valve (Figure 7.26 and 7.27) showed that the contribution of hot water temperature sensor (variable 1) leads for 2 or 3 sampling times, then the holding tube inlet temperature (variable 3) follows it. However, the contribution plots did not show the steam valve (variable 5) as a contributor even in the later sampling times.

In the third fault at time 741 in steam valve fault, the contribution plots of SPE_N showed the holding tube inlet temperature sensor as the cause of the alarms (Figure 7.28). In the fourth fault at time 961 in steam

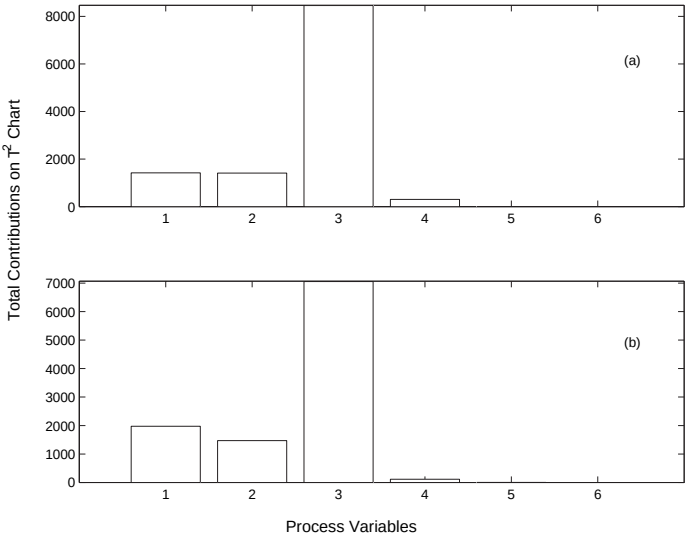


Figure 7.24. Contribution plots of T^2 for the steam valve fault 3 (Table 5.1). Sampling time of the snapshot after the fault is introduced (a) 40, (b) 41. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

valve fault I, just after the fault was introduced, SPE_N chart gave an out-of-control alarm which was caused by holding tube inlet and holding tube outlet temperature sensors according to the contribution plots (Figure 7.29). Obviously, this can not be caused by the fault in the steam valve. After 6 sampling times, contribution plots showed that the reason for the alarms in SPE_N chart after time 961 is the hot water temperature sensor and then the holding tube inlet temperature, which is the expected result of a fault in the steam valve.

Fault diagnosis of the HTST pasteurization system has also been conducted by using parity relations [143], providing a comparative illustration of the use of HMMs (Section 7.3.1), contribution plots and parity space. The parity-space-based diagnosis issued alarms for the faults at the same time or after the multivariate charts in this case study and indicated the reasons behind the out-of-control alarms [143]. A fault diagnosis system that uses several of these techniques simultaneously and integrates their findings by using a decision maker seems more powerful than any single technique used.

Analysis of contribution plots can be automated and linked with fault

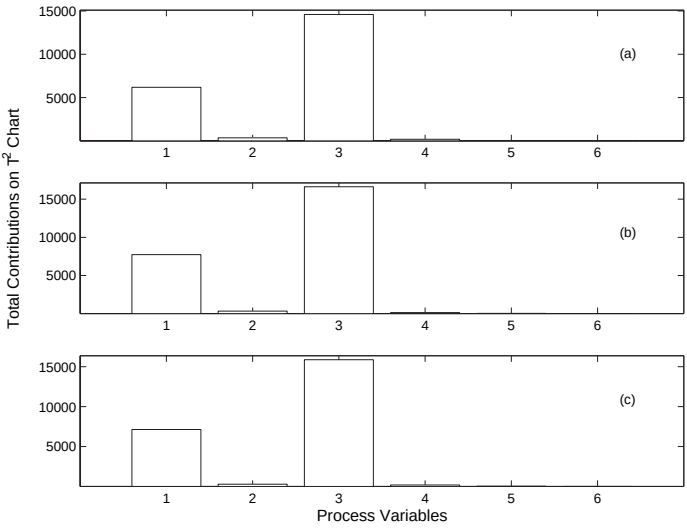


Figure 7.25. Contribution plots of T^2 for the steam valve fault 4. Sampling time of the snapshot after the fault is introduced (a) 19, (b) 20, (c) 21. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

diagnosis by using real-time knowledge-based systems (KBS). The integration of statistical detection tools, contribution plots and fault diagnosis by a supervisory KBS has been illustrated for both continuous [219] and batch processes [302, 303].

7.5 Fault Diagnosis with Statistical Methods

Contribution plots presented in Section 7.4 provide an indirect approach to fault diagnosis by first determining process variables that have inflated the detection statistics. These variables are then related to equipment and disturbances. A direct approach would associate the trends in process data to faults explicitly. HMMs discussed in the first three sections of this chapter is one way of implementing this approach. Use of statistical discriminant analysis and classification techniques discussed in this section and in Section 7.6 provides alternative methods for implementing direct fault diagnosis.

When a process can be represented by a few PCs, the biplots of PCs and SPE provide a visual aid to identify data clusters that indicate normal operation or operation under a specific fault (Figure 5.1). An integrated

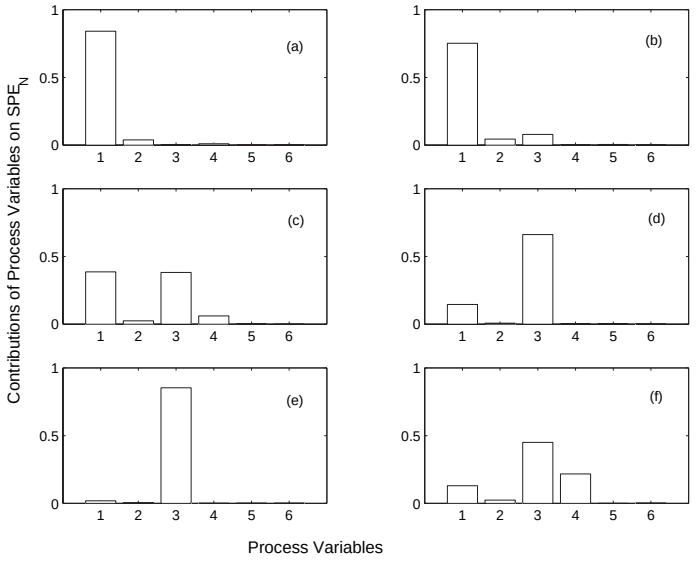


Figure 7.26. Contribution plots of SPE_N for steam valve fault 1 at (Table 5.1). Sampling time of the snapshot after the fault is introduced (a) 7, (b) 8, (c) 9, (d) 10, (e) 11, (f) 15. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

statistical method was developed for processes that need to be described by a higher number of PCs or for automation of diagnosis activities by utilizing PCA and discriminant analysis techniques [242]. PCA is used to develop a model describing normal operation (NO). This PC model is used to detect outliers from NO, as excessive variation from normal target or unusual patterns of variation. Operation under various known upsets is also modeled using PCA provided that sufficient historical data are available. These fault models are then used to isolate source causes of faulty operation based on the proximity of current process operation to one of the data clusters indicating a specific fault. Using PCs for several sets of data under different operating conditions (NO and with various upsets), statistics can be computed to describe distances of the current operating point to regions representing other conditions of operation. Both scores distances and model residuals are used to measure such distance-based statistics. In addition, angle-based criteria can also be used. The FDD system design includes the development of PC models for NO and abnormal operation with specific faults, and the computation of threshold limits using historical data sets collected during normal plant operation and operation under specific faults.

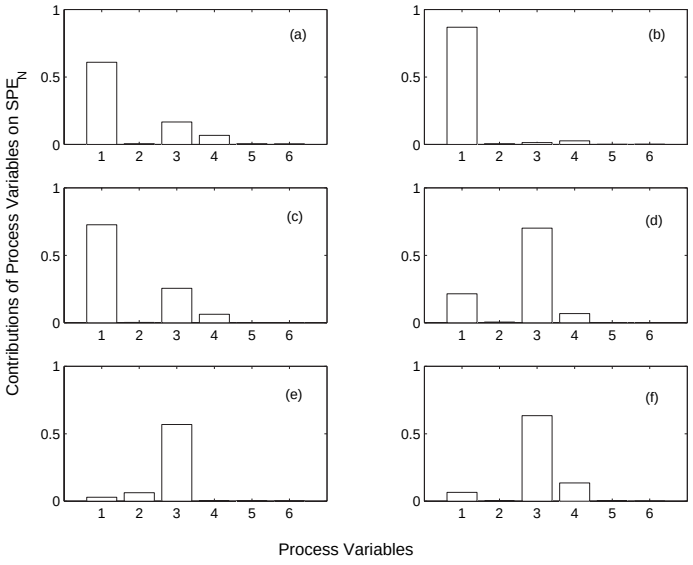


Figure 7.27. Contribution plots of SPE_N for steam valve fault 2. Sampling time of the snapshot after the fault is introduced (a) 25, (b) 26, (c) 27, (d) 28, (e) 29. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

The implementation of the FDD system at each sampling time starts with monitoring. The model describing NO is used with new data to decide if the current operation is in-control. If there is no significant evidence that the process is out-of-control, further analysis is not necessary and the procedure is concluded for that measurement time. If score or residual tests exceed their statistical limits, there is significant evidence that the process is out-of-control. Then, the PC models for all faults are used to carry out the score, residuals, and/or angle tests, and discriminant analysis is performed by using PC models for various faults to diagnose the source cause of abnormal behavior.

The method was developed for monitoring continuous processes deviating from their steady-state operation and determining the most likely source causes from a closed set of candidate causes. Stationarity, ergodicity and lack of significant autocorrelation should be established before utilizing this method. The method does not rely on visual inspection of plots; consequently, it is suitable for processes described by large sets of variables. The method was illustrated by monitoring the Tennessee Eastman industrial challenge problem [58].

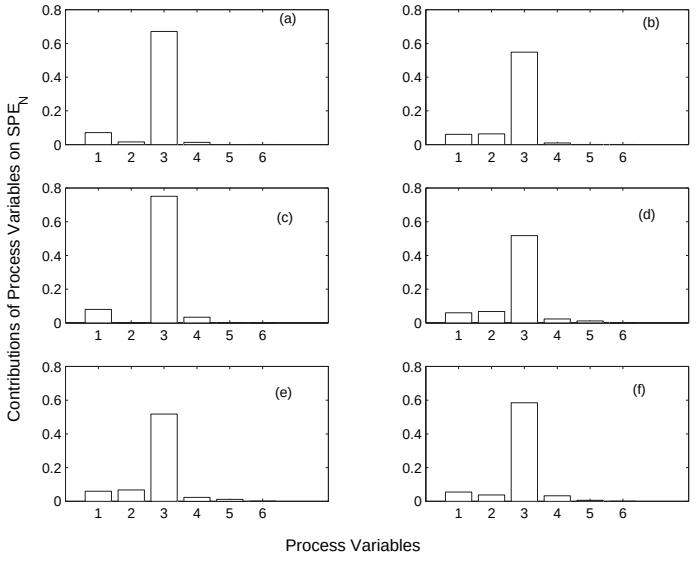


Figure 7.28. Contribution plots of SPE_N for steam valve fault 3. Sampling time of the snapshot after the fault is introduced (a) 11, (b) 14, (c) 15, (d) 16, (e) 17, (f) 18. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

PC models for specific faults can be developed using historical data sets collected when the process was experiencing that fault. When current measurements indicate out-of-control behavior, a likely cause for this behavior is assigned by pattern matching by using scores, residuals, angles or their combination.

Score Discriminant Assuming that PC models retain sufficient variation to discriminate between possible causes in scores that have independent Normal distributions, the maximum likelihood that data $\mathbf{x}$ collected at a specific sampling time are from fault model i is indicated by the minimum distance. This minimum can be determined for example by the maximum of d_i expressed by quadratic discrimination (Eq. 3.41)

$$d_i(\mathbf{t}) = \ln p_i - \frac{1}{2}\ln|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{t}-\bar{\mathbf{t}}_\mathbf{i})^\mathbf{T}\mathbf{\Sigma_i}^{-\mathbf{1}}(\mathbf{t}-\bar{\mathbf{t}}_\mathbf{i}) \tag{7.5}$$

where $\mathbf{t} = \mathbf{xP}_i$ is the location of original observation $\mathbf{x}$ in PC space for fault model i, $\bar{\mathbf{t}}_i$ and $\mathbf{\Sigma_i}$ are the mean and the covariance along PCs for fault model i, and p_i is the adjustment for overall occurrence likelihood of fault i [126].

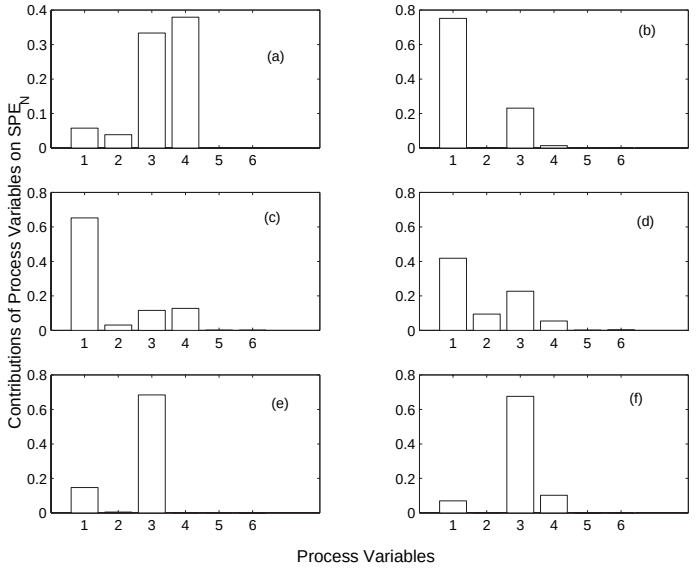


Figure 7.29. Contribution plots of SPE_N for steam valve fault 4. Sampling time of the snapshot after the fault is introduced (a) 1, (b) 6, (c) 7, (d) 8, (e) 9, (f) 10. Reprinted from [143]. Copyright © 2001 with permission from Elsevier.

Residual Discriminant For situations where the data collected are not described well by PC models of other faults but will be within the residual threshold of their own class, it is most likely that $\mathbf{x}$ is from the fault model i with minimum

$$r_i/r_{i,\alpha} \qquad \text{where} \qquad r_i = \mathbf{t}_i^T(\mathbf{I} - \mathbf{P}\mathbf{P}^\mathbf{T})\mathbf{t}_i \tag{7.6}$$

r_i is the residual computed using the PCA model for fault i and $r_{i,\alpha}$ is the residual threshold at level 100α based on the PCA model for fault i.

Combined Distance Discriminant Combining the information available in scores and residuals usually improves the diagnosis accuracy [206]. Comparing the combined information to the confidence limits of each fault model, $\mathbf{x}$ is most likely to be from the fault model i with minimum

$$c_i\left(\frac{r_i}{r_{i,\alpha}}\right) + (1 - c_i)\left(\frac{t_i}{t_{i,\alpha}}\right) \tag{7.7}$$

where t_i and r_i are the score distance and residual for fault i based on the PC model, respectively, $t_{i,\alpha}$ and $r_{i,\alpha}$ are the score distance and residual

thresholds using the PC model, respectively, for fault i, and c_i is a weight between 0 and 1. c_i is set equal to the fraction of total variance explained by scores in order to weigh scores and residuals according to the amount of variation in data explained by each. The combined discriminant value thus calculated gives an indication of the degree of certainty for the diagnosis. A value less than 1 indicates a good fit to the chosen model. If no model results in a statistic less than 1, none of the models provide an adequate match to the observation. When a group of observations fail to fit within any of the known groups, they could be considered as a new group and added to the discrimination scheme.

The statistical distance discrimination schemes described are simple to implement. Relating increasing distance with lower likelihoods, they have an intuitive appeal. They can use a large number of correlated variables to choose between many possible source populations. Disjointedness and overlap of sets can be accommodated. Unlike other diagnosis methods, additional source populations can easily be incorporated into the discrimination scheme without retraining the whole diagnosis system.

Example The test statistics using different types of discriminants can be plotted versus sample number in semilog plots. Figure 7.30 shows (a) residual, (b) score (plotted as the negative of the discriminant) and (c) combined score-residual discriminants at each sampling time during a run with disturbance A (random variations in feed temperature) of the Tennessee Eastman industrial challenge problem. The minimum and maximum (dashed line), and average (solid line) statistics comparing a sample to all possible groups are shown along with the discriminant for the actual disturbance (stars). Correct diagnosis is made when the statistic for the true group coincides with the minimum value. The correct diagnosis is never made for this case with residuals (Figure 7.30a). The combined discriminant (Figure 7.30c) diagnoses the disturbance erratically and score discriminant (plotted as log of its absolute value) (Figure 7.30c) diagnoses the disturbance correctly most of the time.

Figure 7.31 illustrates the fault isolation process when disturbance 3 (step change in feed temperature) is introduced. Score discriminants are calculated using PC models for the various known faults (Figure 7.31c); this semilog plot shows the negative of the discriminant. The most likely fault is chosen over time by selecting the fault corresponding to the maximum discriminant (curve with the lowest magnitude). Figure 7.31d reports the fault selected at each sampling time. Fault 3, which is the correct fault, has been reported consistently after the first 10 sampling times.

Angle-Based Discriminants for Diagnosis

The angles between principal coordinate directions of current data and

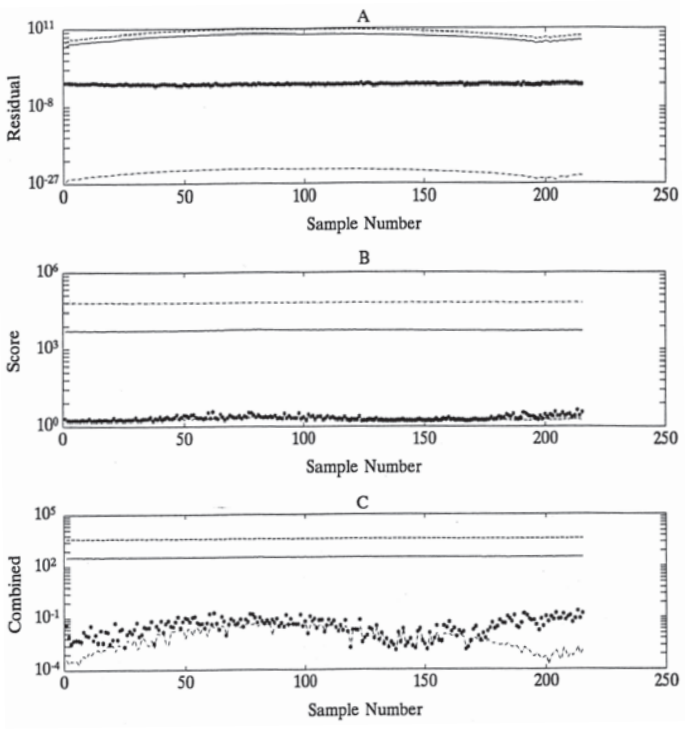


Figure 7.30. Test statistics based on distance discriminants when the process is subjected to disturbance A: (a) Residuals, (b) Scores, and (c) Combined statistics. Minimum and maximum values (dashed lines), average values (solid line), statistics of disturbance A (stars). Reprinted from [243]. Copyright © 1997 with permission from Elsevier.

regions corresponding to operation with different faults can be used for fault diagnosis, to complement distance-based methods [243]. The method uses angles between different coordinate systems and a similarity index defined by using the angle information [154].

Euclidean and Mahalanobis Angles The Euclidean angle θ_E between two points a and b (with coordinates $\mathbf{a}$ and $\mathbf{b}$ and the vertex at the origin) is defined using vector products,

$$cos(\theta_E) = (\mathbf{a}^T\mathbf{b})/(\|\mathbf{a}\|\|\mathbf{b}\|) \qquad \text{where} \qquad \|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}} \tag{7.8}$$

Adjusting the angle definition for a weighted distance, the Mahalanobis

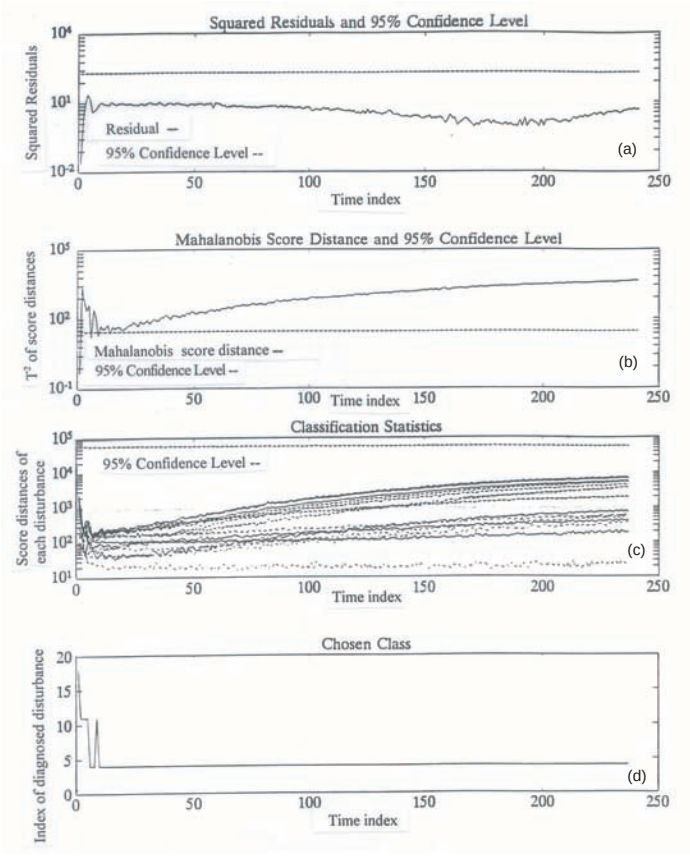


Figure 7.31. Detection and diagnosis of process upsets, (a) Detection of outliers based on residuals, (b) Detection based on T^2 test of scores, (c) Diagnosis statistics considering each possible disturbance, (d) Index of chosen disturbance for each observation. Reprinted from [243]. Copyright © 1997 with permission from Elsevier.

angle (θ_M) between points a and b with the vertex at the origin is

$$cos(\theta_M) = (\mathbf{a}^T\mathbf{S}^{-1}\mathbf{b})/(d(\mathbf{a},0)d(\mathbf{b},0)) \tag{7.9}$$

where $\mathbf{S}$ is the covariance matrix and $d(\mathbf{a},\mathbf{b}) = \sqrt{(\mathbf{a}-\mathbf{b})^T\mathbf{S}^{-1}(\mathbf{a}-\mathbf{b})}$ is the Mahalanobis distance for points a and b. A constant Mahalanobis angle around the line joining point a with the origin is a hyperconical surface, with distortion given by the matrix $\mathbf{S}$.

Angular Discriminant For distance-based discriminants, the diagnosis can be posed as a minimization of distance penalties. For angular information, a suitable discriminant can be stated as:

$$\min_i |\theta_i| \tag{7.10}$$

where θ_i is the angle between the test point and the mean of the ith group, with the vertex positioned at the mean of NO. Looking at the absolute value has the effect of ignoring on which side of the target mean a point may lie, relative to the line joining the mean and the origin. Decision boundaries for angular discriminants describe open-ended conical regions in space.

Choice of Angular Mahalanobis Weighing A major task, as in distance-based discriminants, is finding a suitable dispersion matrix and choice of coordinates or dimensions to retain. In general, distance-based discriminants use a covariance matrix. Estimation of the covariance is done by the method derived for a multivariate Normal distribution which provides the most likely estimate. However, there are some difficulties in using the usual estimate for highly correlated variables. Mahalanobis-style weighing uses the inverse of the covariance matrix, which can be mathematically unstable or physically unsuitable as it amplifies the importance of measurements that have the smallest change. Use of PCA can work around the inversion problem, but are generally also derived from the multivariate Normal distribution.

Residual Mahalanobis Angle The residual Mahalanobis angle ϕ is defined by replacing $\mathbf{S}^{-1}$ with $\mathbf{I} - \mathbf{PP}'$ as the weighing matrix:

$$d_r(\mathbf{a}, \mathbf{b}) = \sqrt{(\mathbf{a} - \mathbf{b})^T (\mathbf{I} - \mathbf{PP}^T)(\mathbf{a} - \mathbf{b})} \tag{7.11}$$

$$cos(\phi) = (\mathbf{a}^T (\mathbf{I} - \mathbf{PP}^T) b) / (d_r(\mathbf{a}, 0) d_e(\mathbf{b}, 0)) \tag{7.12}$$

Example Use of angle-based diagnosis is illustrated by introducing again disturbance A (random noise in feed temperature) of the Tennessee Eastman industrial challenge problem. Figure 7.32a shows the minimum and maximum angles (dashed lines), and the average angle (solid line) for all 21 possible disturbances along with the angle to the correct disturbance (indicated by $\times$). The diagnosis at each sampling time is made by selecting the disturbance with the minimum angle to the observation, as plotted in Figure 7.32b. Most samples are correctly diagnosed as coming from disturbance A (class 10), with a few misclassifications at the beginning and end of the run. A geometric explanation for this behavior could be that the trajectory of data over time is curved, so the samples near the middle of

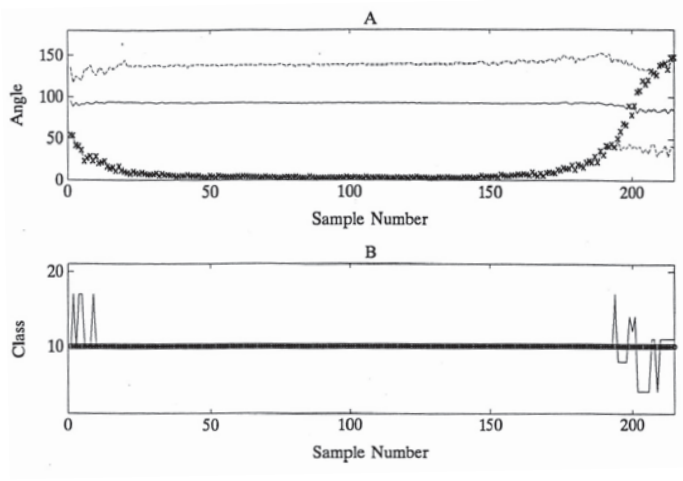


Figure 7.32. (a) Test statistics based on angle discriminants: minimum and maximum values (dashed lines), average values (solid line), statistics of disturbance A (×), (b) Disturbance/fault diagnosed when the process is subjected to disturbance A. Reprinted from [243]. Copyright © 1997 with permission from Elsevier.

the run are within angular bounds while those at the beginning and end of the run are at larger angles.

Table 7.4 lists the average percentage of observations correctly diagnosed using angles, score, residual and combination of scores and residuals discriminants with new data not used in model development. In general, half of the observations were correctly diagnosed, with step and ramp disturbances 10 to 40% better classified than random disturbances. As expected, diagnosis with new data was slightly less successful than diagnosis using observations from training.

A related topic is the comparison of PC models and statistical tests for overlap between disturbance regions. Krzanowski [154] describes the derivation of angles between coordinate axes from different models, and proposes the minimum angle between models as a benchmark for simple analysis. Use of angles to evaluate overlap between regions is discussed in [243]. The similarity index can be used to evaluate discrimination models by selecting a threshold value to indicate where mistakes in classification of data from the two models involved may occur. It can also be used to compare models built from different operating runs of the same process for monitoring systematic changes in process variation during normal opera-

Table 7.4. Percentage success in diagnosis of various disturbances with new data. Reprinted from [243]. Copyright © 1997 with permission from Elsevier.

Disturbance	Type	Score	Residual	Combination	Angle
1	Step	94	0	69	100
2	Step	41	0	0	93
3	Step	0	100	92	29
4	Step	0	0	0	74
5	Step	37	0	48	98
6	Step	70	0	73	93
7	Step	93	0	51	62
8	Random	1	0	0	99
9	Random	14	0	0	12
A	Random	57	0	25	88
B	Random	0	0	0	31
C	Random	7	0	9	0
C and F	Random	33	0	41	0.9
D	Ramp	0.7	0	0	87
E	Random	0	100	62	0
F	Random	0	14	0	44
G	Random	33	0	13	47
H	Ramp	76	0	66	97
I	Ramp	0	0	0	98
J	Random	0	14	0	44
K	Random	73	0	0.9	32

tion. The similarity index has a range from 0 to 1, increasing as models become more similar. It provides a quantitative measure of difference in covariance directions between models and a description of overall geometric similarity in spread.

Discrimination and Diagnosis of Multiple Disturbances

Detection and diagnosis of *multiple simultaneous faults* is an important concern. Most FDD techniques rely on the assumption of a single fault. In a real process, combinations of faults may occur. An intervention policy to improve process operation may need to take into account each of the contributing faults. Diagnosis should be able to identify major contributors and correctly indicate which, if any, secondary faults are occurring [241]. In fault diagnosis, where process behavior due to different faults is described by different models, it is useful to have a quantitative measure of similarity or overlap between models, and to predict the likelihood of successful diag-

nosis. Similarity measures serve as indicators of the success in diagnosing combinations of faults. They can identify combinations of faults that may be masked or falsely diagnosed, and provide information about the success rates of different diagnosis schemes incorporating single and combinations of faults. Using these guidelines, multiple faults occurring in a process can be analyzed *a priori* with respect to their components, and accommodated within the diagnosis framework.

In comparing multivariate models, much work has been reported for testing significant differences between means when covariance is constant. Testing for differences in covariance is more difficult yet crucial; diagnosis can be successfully done, whether or not means are different, as long as there is a difference in covariance [79]. Testing for eigenvalue models of covariance adds new complications, since the statistical characteristics are not well known, even for common distributions. Simplifying assumptions for special cases can be made, with significant loss of generality [194].

Overlap of Means An important statistical test in comparing multivariate models is for differences in means. This corresponds to comparison of origin of coordinates rather than the coordinate directions. Many statistical tests have been developed for testing means, but most of them can become numerically unstable when significant correlation exists between variables. In order to work around the instability, overlap between eigenvalue-based models can be evaluated. Target factor analysis can assign a likelihood on whether a candidate vector is a contributor to the model of a multivariate data set. A statistic is defined to test if a specific vector is significantly inside the confidence region containing the modeled data [181]. For overlap of means, the test can determine whether the mean from one model, μ_1, significantly overlaps the region of data from another (second) model [242]. Mean overlap analysis can be used to test if an existing PC model fits a new set of observations or if two PC models are analogous.

If there is no overlap between regions spanned by two different faults, two alternative schemes might handle multiple faults modeled by PCA. In one method, the combination fault is idealized as being located between the regions of the underlying component faults; allocations of membership to the different independent faults contributing to the combination may provide diagnosis of underlying faults. The second method is based on a more general extension of the discrimination scheme by introducing new models for each multiple-fault combination of interest. The measures of similarity in model center and direction of spread can be useful to determine the independence of the models used in diagnosis.

Masking of Multiple Faults When the region spanned by the model for one (*outer*) fault contains the model for another (*inner*) fault, their combination

will not be perfectly diagnosed. Idealizing the two fault regions as concentric spheres, the *inner* model region is enveloped by the *outer* model. As a result, only the *outer* fault will be diagnosed and the *inner* fault will be masked. Overlap of regions is likely to exist for most processes under closed-loop control, the multiple fault scenario is further complicated for such processes.

Faults causing random variation about a mean value (such as excessive sensor noise) move a process less drastically off-target than step or ramp faults. Similarity measures should indicate that the random variation faults have more overlap with other models, particularly with each other. Ramp or step faults tend to be the *outer* models and mask secondary random variation faults.

7.6 Fault Diagnosis Using SVM

Support vector machines (SVM) have been used for many classification and diagnosis problems in applications such as medical diagnosis, image recognition hand-written character recognition, bioinformatics and text categorization [42]. Their use in chemical process fault diagnosis has been reported in recent years. In one application, the performances of Fisher discriminant analysis, SVM, and proximal SVM for fault diagnosis are investigated [37]. Proximal SVM determines 'proximal' planes that separate the different classes to reduce the computational burden. The fault classification performance was evaluated by using the Tennessee Eastman process simulator. The authors report the data sets had irrelevant information when all variables were used and the classification with SVM and PSVM were poor. When relevant variables were selected by using genetic algorithms and contribution plots, and used for fault classification, the percentage of misclassifications dropped and SVM and PSVM outperformed FDA [37]. The authors report misclassification for the testing data set to drop from 38% to 18% for FDA, and from 44-45% to 6% for SVM and PSVM. By incorporating time lags into SVM and PSVM for auto-correlated data, they reduced the overall misclassification with SVM and PSVM to 3%.

A study that integrates SVM with genetic-quasi-Newton optimization algorithms reported the application of the methodology to rayon yarn data (two classes) and wine data (three classes) with very low misclassification rates (0.1%) [156].

7.7 Fault Diagnosis with Robust Techniques

A number of practical issues arise when a process monitoring strategy is implemented in a real-time environment. Specifically, the data collected from a Distributed Control System (DCS) are high dimensional, noisy, have strongly correlated variables and, in most cases, the correlation structure may be nonlinear. Furthermore, such process data often contain outliers (gross errors) as a result of process characteristics, faulty sensors, equipment failures, transient effects, or during the transference of values acquired by analog/digital converters. When developing an operator support system (OSS), such issues need to be tackled directly to ensure intelligent monitoring capabilities.

The OSS has to provide the means for suppressing noise and outliers, detecting in-control and out-of-control operations, and render sensor reconstruction when some sensors become unavailable. Thus, the elements of a robust monitoring strategy would be (i) robust filtering, (ii) dimensionality reduction, (iii) fault detection and isolation and (iv) sensor reconstruction. This strategy is depicted in Figure 7.33. Each step in this strategy will be reviewed in the next section, followed by an application to a pilot-scale distillation column.

7.7.1 Robust Monitoring Strategy

The elements of the robust monitoring strategy builds on the methods discussed previously (e.g., PCA in 3.1 and signal filtering in 6.2.3). Here, only the key variations will be introduced.

Robust Filtering The robust filtering step uses the tandem filtering approach discussed in Section 6.2.3 where the moving median filter is used along with wavelet coefficient denoising to remove outliers and noise artifacts from the measured signal. Then, the 'clean' process signal is presented to the subsequent steps for monitoring.

Nonlinear PCA To address the nonlinearity in the identity mapping of multivariate data, a nonlinear counterpart of the PCA can be used (see Section 3.6.1). As the versions of NLPCA make use of the neural network (NN) concept to address the nonlinearity, they suffer from the known over-parameterization problem in the case of noise corrupted data. Data with small SNR will also give rise to extensive computations during the training of the network. Shao *et al.* [266] used wavelet filtering to pre-process the data followed by IT-net to detect the non-conforming trends in an industrial spray drier.

The approach presented here is based on Kramer's work [150] where his method uncovers both linear and nonlinear correlations independent of the

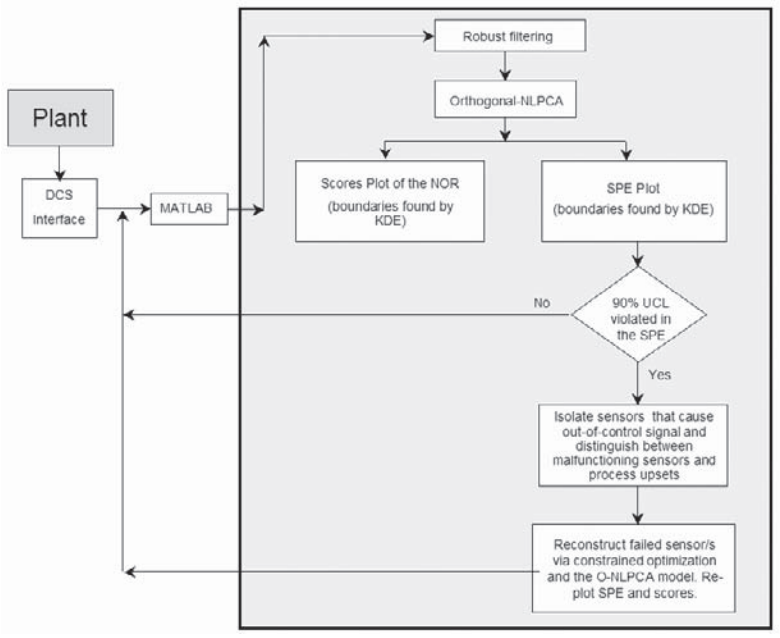


Figure 7.33. The schematic of robust monitoring strategy. Reprinted from [60]. Copyright © 2001 with permission from Elsevier.

structure of nonlinearity present in the data. This is indeed the NLPCA built on autoassociative NNs discussed in Section 3.6.1. This architecture constructs lower dimensional features which are nonlinear combinations of the original process variables, but it does not encourage explicitly the development of principal components which measure distinct dimensions in the data, as in the case for PCA. To provide a strategy in line with the features of linear PCA, an orthogonalization needs to be performed. This is accomplished using the Gram-Schmidt orthogonalization method that can be found in many textbooks on linear algebra [92]. The Gram-Schmidt procedure is performed as follows: Given a non-orthogonal set of vectors, $\{U_1, U_2, \cdots, U_p\}$,

1. Let $T_1 = U_1$
2. Compute vectors $T_1, T_2, \cdots, T_p$ successively using the formula,

$$T_i = U_i - \left(\frac{U_i \cdot T_1}{T_1 \cdot T_1}\right) T_1 - \left(\frac{U_i \cdot T_2}{T_2 \cdot T_2}\right) T_2 - \cdots - \left(\frac{U_i \cdot T_{i-1}}{T_{i-1} \cdot T_{i-1}}\right) T_{i-1} \tag{7.13}$$

where '$\cdot$' denotes the dot product.

The set of vectors $\{T_1, T_2, \cdots, T_p\}$ constitutes an orthogonal set. The procedure requires designation of a vector from which all other vectors are constructed so as to be orthogonal to the initially chosen vector. In this case, there is no restriction on which vector has to be chosen first.

In summary, the orthogonal nonlinear principal component analysis (O-NLPCA) algorithm develops orthogonal components directly from an auto associative neural network using the Gram-Schmidt process. The mechanism by which it incorporates the orthogonalization procedure resembles 'cascade control' where a faster inner loop rejects a disturbance before it affects the outer loop. Figure 7.34 depicts the schematic of the O-NLPCA proposed by Chessari [33]. The procedure is implemented in such a way that the mapping layer of network is designated as the inner loop, and the whole network is regarded as the outer loop. In order for the network to generate orthogonal outputs at the bottleneck layer, one of the outputs is chosen to be an 'anchor' vector so that the remaining outputs are orthogonalized with respect to this anchored vector. Choice of the anchor vector is random so as to eliminate biasing towards one vector. Once the orthogonalized outputs are obtained, the inner loop can map the inputs onto this set of bottleneck outputs. The training should not be carried out until the error is minimized below a certain threshold because these vectors may not satisfy the overall identity mapping objective. Training with a small number of iterations encourages the construction of orthogonal nonlinear principal components (secondary objective) without intervening with the identity mapping (the main objective). The drawback of having such a structure is that the overall training of the network takes longer than the original form of the NLPCA. Also if too many passes are allowed in the inner loop, not only does the overall convergence slows down, but also the secondary objective will not be met.

Fault Detection and Isolation Stork *et al.* [284], and Stork and Kowalski [283] proposed two algorithms to identify multiple sensor disturbances using backward elimination sensor identification (BESI), and to distinguish between process upsets and sensor malfunctions via redundant sensor voting system (RSVS), respectively. In the BESI approach, once the SPE is violated at a given time, every sensor is sequentially removed from the model matrix followed by calculation of the upper control limit. If the ratio of the SPE/SPE_{limit} is less than one, then algorithm terminates and points to the sensor/s that are left out for the out-of-control signal. Otherwise, the procedure is continued until SPE/SPE_{limit} ratio drops below one. This approach is computationally expensive to carry out multiple PCA calculations at each time the SPE is violated. Moreover, it is almost

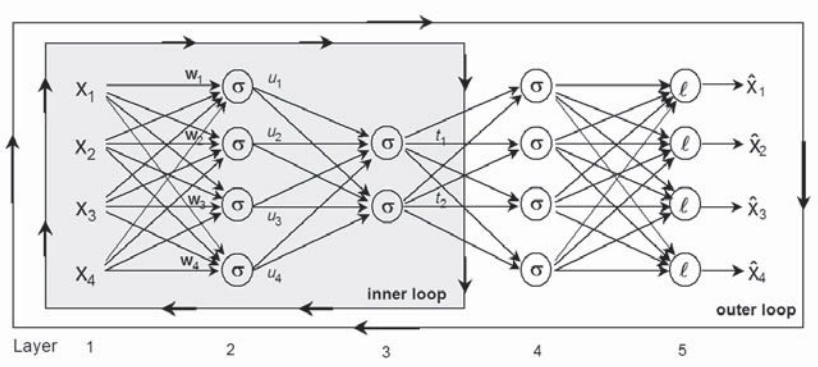


Figure 7.34. The O-NLPCA structure. Reprinted from [60]. Copyright © 2001 with permission from Elsevier.

impossible to incorporate it within the nonlinear PCA framework since every time the SPE_{limit} is exceeded, a new neural network would need to be trained to check the ratio of SPE/SPE_{limit}. On the other hand, the RSVS approach is a three step procedure: (i) identification of the redundancies among process sensors and determination of minimum redundancy bandwidth, (ii) ordering of sensors with redundant sensors in close proximity and (iii) application of probabilistic and/or empirical rules using the disturbance pattern identified for a new process measurement. Prior to applying the RSVS, disturbances need to be identified (via BESI, for instance) correctly and then RSVS can distinguish whether the disturbance is a sensor malfunction or a process upset. Stork and Kowalski point out that [283] false alarms might lead the RSVS to misdiagnose the source of the disturbance. In what follows is a new fault detection and identification technique that reduces computational load, suits both linear and nonlinear PCA, provides reconstructed values for sensors identified as faulty, and potentially eliminates false negative situations.

The technique is referred to as Backward Substitution for Sensor Identification and Reconstruction (BSSIR) and it is based on the principle that process upsets and sensor failures can be identified in the presence of redundancy among sensor arrays. Due to process characteristics, these measurements may have strong correlations among each other, particularly the ones in close proximity and measuring the same variable. Therefore, when a disturbance affects the process, it would be sensed by a group of sensors rather than just by one. However, if a sensor malfunctions (e.g., due to complete failure, bias, precision degradation, or a drift), then this will only affect the individual sensor performance, at least initially. If the malfunc-

tioning sensor is associated with a manipulated (or a controlled) variable of a feedback control system, the information conveyed to the controller will be inaccurate. As a result, such a sensor malfunction may eventually manifest itself in more than one sensor.

Once a calibration model for the process space is built using the linear/nonlinear PCA, over the course of operation, the SPE can be used to monitor the process against any unanticipated disturbances and/or sensor failures. At times when the SPE_{limit} is violated, instead of evaluating the variable contribution to the SPE, one can go one step back in each sensor array and calculate the SPE again. Subsequently, the SPE values are ordered from minimum to maximum. In other words, following vectors are defined first,

$$\begin{aligned} x_{1-} &= [x_1(k-1)\ x_2(k) \cdots\ x_m(k)] \quad \rightarrow \quad \hat{x}_{1-} = x_{1-}aa^T \\ x_{2-} &= [x_1(k)\ x_2(k-1) \cdots\ x_m(k)] \quad \rightarrow \quad \hat{x}_{2-} = x_{2-}aa^T \\ &\vdots \\ x_{m-} &= [x_1(k)\ x_2(k) \cdots\ x_m(k-1)] \quad \rightarrow \quad \hat{x}_{m-} = x_{m-}aa^T \end{aligned} \tag{7.14}$$

where x_{i-} denotes a row vector containing measurements for all sensors at time k, but $k-1$ sample for sensor j; $\hat{x}_{j-}$ is the model estimate. The r_j represents the corresponding SPE:

$$\begin{aligned} r_1 &= \sum (x_{1-} - \hat{x}_{1-})^2 \\ r_2 &= \sum (x_{2-} - \hat{x}_{2-})^2 \\ &\vdots \\ r_m &= \sum (x_{m-} - \hat{x}_{m-})^2 \end{aligned}$$

Then, the sensor index of ordered sum of squared residuals can be expressed as

$$r_{s,index} = index\{sort(r_1, r_2, \cdots, r_m)\} \tag{7.15}$$

The sensor with $r_{s,index}(1)$ is first reconstructed using the calibration model and the constrained optimization algorithm described below in sensor reconstruction. After the first iteration, if the SPE remains above its limit, then $r_{s,index}(1,2)$ are reconstructed together. This procedure continues until either the SPE falls below its limit or the number of reconstructed sensors equals the number of principal components retained for the calibration model. Meanwhile, the reconstructed values are saved for use in the subsequent instant the SPE goes beyond its limit.

Now that the affected sensors are isolated, the root cause for the alarm can be explored. In other words, is the alarm due to a sensor malfunction,

or a disturbance? The criterion is the correlation coefficient (CC) defined as

$$CC = \frac{Cov(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}} \tag{7.16}$$

where $Cov(x_,x_j)$ denotes the covariance between x_i and x_j, and σ_x is the standard deviation for each vector. For the jth sensor, the most correlated sensor pairs can be found by ordering CCs between the jth sensor and other sensors. To calculate the correlation coefficient at time the SPE_{limit} is triggered in the test set, a $Q \times 1$-size moving window, which contains the current sample and $Q-1$ past samples for each sensor array, is formed. The CCs calculated during plant operation are then compared with the ones obtained from the training data. Starting from the first test sample, if a threshold of 10% or more degradation is observed in the CCs between the jth sensor and the two most correlated ones, then sensor j is most likely malfunctioning, because a failure in one sensor should not interfere with other sensor readings unless that sensor is conveying information to one of the controllers in the system. Otherwise, the cause that triggered the SPE to exceed its limit will be due to a disturbance in the process, since a disturbance would typically propagate through the process and affect multiple correlated sensors.

Sensor Reconstruction Following the fault detection procedure discussed above, the maintenance of unavailable sensors is needed as soon as they are detected. However, if the sensor that conveys information to one of the controllers were to be faulty, it is essential that its value be reconstructed from the remaining sensors on-line. Sensor reconstruction can be performed using the calibration model based on the PCA/NLPCA.

Here, after detecting and identifying the failed sensor/s, the unavailable sensor values are reconstructed using the calibration model and a constrained optimization algorithm from the remaining sensors. Each unavailable sensor value can be estimated by solving the following problem:

$$min\|x_i - \hat{x}_i\| \qquad i = 1, 2, \cdots, m \tag{7.17}$$

such that

$$LB \leq \hat{x}_{fs,i} \leq UB, \qquad SPE \leq \Omega_{1-\alpha}$$

where $\hat{x}$ denotes the estimation obtained from the calibration model, x_{fs} represents the failed sensor, and $\hat{x}_{fs,i}$ is the reconstructed value of i^{th} failed sensor. LB and UB are the lower and upper bounds for the missing sensor, and $\Omega_{1-\alpha}$ is the $100(1-a)\%$ upper control limit of the SPE. Equation 7.17 can be solved easily since only the forward evaluations of the trained O-NLPCA network are required. Hence, a one-dimensional search over the

missing values that satisfy the constraints will provide a solution to the problem effortlessly. Meanwhile, the values of the remaining sensors are kept constant while the optimization is carried out. Multiple sensor failures can also be accommodated in an analogous way provided that the dimension of the bottleneck is equal to or less than the number of the available sensors. In this case, however, the problem becomes a multidimensional search for values of the missing sensors that satisfy Eq. 7.17.

7.7.2 Pilot-Scale Distillation Column

A pilot-scale distillation column located at the University of Sydney, Australia is used as the case study [60]. The 12-tray distillation column separates a 36% mixture of ethanol and water. The following process variables are monitored: temperatures at trays 12, 10, 8, 6, 4, and the reflux stream, bottom and top levels (condenser), and the flow rates of bottoms, feed, steam, distillate and reflux streams. The column is operated at atmospheric pressure using feedback control. Three variables are controlled during the operation: top product temperature, condenser level, and bottom level. Temperature at tray 8 is considered as the inferential variable for top product composition. To maintain a desired product composition, PI controllers cascaded on flow were used to manipulate the reflux, top product and bottom product streams.

The column was operated four times at various operating conditions. The first three data sets corresponding to a total of 12.8 *hr* of operation were used to train the O-NLPCA network, and the fourth one was used for model validation. However, prior to building a calibration model, both the training and the testing data were processed through the robust tandem filter to remove noise and suppress possible outliers.

The O-NLPCA network has 8-6-10-12 neurons in each layer, yielding a prototype model with 6 principal components (PCs). For comparison, the linear PCA was also applied to the same data. As a performance criterion, the root mean square of error (RMSE) was evaluated to compare the prediction ability of the developed PCA and O-NLPCA models on the training and validation data. While the linear PCA gave 0.3021 and 0.3227 RMSE on training and validation data sets, respectively, the O-NLPCA provided 0.2526 and 0.2244 RMSE. This suggests that to capture the same amount of information, the linear PCA entails utilization of more principal components than its nonlinear counterpart. As a result, the information embedded in the nonlinear principal components addresses the underlying events more efficiently than the linear ones.

To define the NO region (NOR) of the plant, kernel density estimation (KDE) is used. The joint probability density of the first and second, and

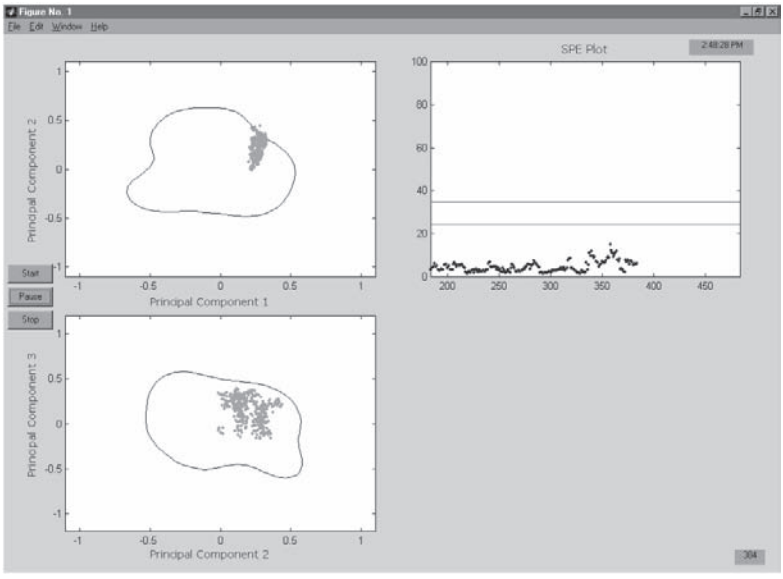


Figure 7.35. The screen shot of the on-line monitoring strategy, indicating normal operation of the column. Reprinted from [60]. Copyright © 2001 with permission from Elsevier.

second and third PCs were estimated. The NOR is defined to be the 95% contour underneath the surface of the joint probability density. In addition, the SPE plot with 90% warning limit, and two-sided 99% individual confidence limits for the filtered process variables were also constructed using KDE to facilitate fault detection and isolation. Any violations of the 90% limit in the SPE will initiate the BSSIR algorithm to isolate the sensors that cause an out-of-control signal, and to distinguish between malfunctioning sensors and process upsets. The next step is to reconstruct those malfunctioning sensor/s using the constrained optimization algorithm and the trained O-NLPCA network. While searching over the values of the failed sensor/s, two criteria are to be met: the value of the SPE should stay below its 90% confidence limit, and the sensor/s values should remain within their previously defined intervals. Following the reconstruction, the scores and the SPE are recalculated and plotted.

To filter incoming process data, a window length of 500 samples was used. The window size is maintained constant by forgetting the first entry of the data vector and appending the new measurement vector to the end. To test the strategy for tracking sensor failures, a complete failure case

is simulated. Sensor 11 (temperature at tray 4) was forced to completely fail between 621 and 877 sampling intervals. The sensor value, which was $\sim 80^oC$, was first decreased to 30 (in time segment 621-700) and then to zero (in time segment 701-877). To test the disturbance monitoring, flooding condition was generated by reducing the steam supply from ~ 1.15 to $\sim 0.74\ kg/min$ for $1.5\ min$ (between 1103-1120 sampling instants), and then increased to $\sim 1.7\ kg/min$ for the rest of the operation.

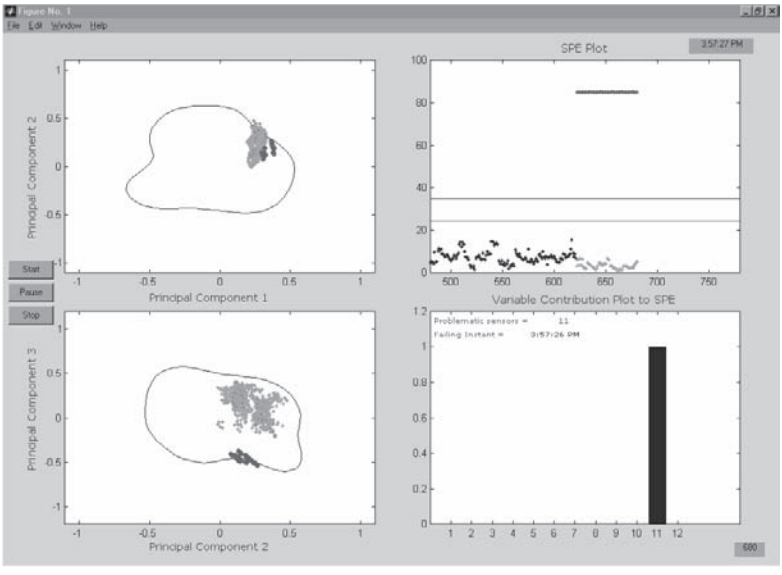


Figure 7.36. The screen shot of the on-line monitoring strategy, indicating sensor failure. Reprinted from [60]. Copyright © 2001 with permission from Elsevier.

In Figure 7.35, the normal operating condition is captured. The subplots in the figure are the scores plot (upper left) between 1st and 2nd PCs, the scores plot (lower left) between 2nd and 3rd PCs, the SPE plot (upper right), and the sensor diagnostic plot (lower right) that shows faulty sensors. The trends show that the process is operating normally, hence, no violations are indicated in the SPE and the scores plot.

Figure 7.36 shows how the monitoring strategy responds when sensor 11 fails. This event is well captured in the scores plot between 2nd and 3rd PCs, and the SPE plot. When the magnitude of SPE is greater than 85, its value is plotted at 85 so that the sum of squared residuals corresponds to normal operation, and the 90% and 95% upper control limits are visible. In addition, the lower right subplot depicts the sensor number and its failing

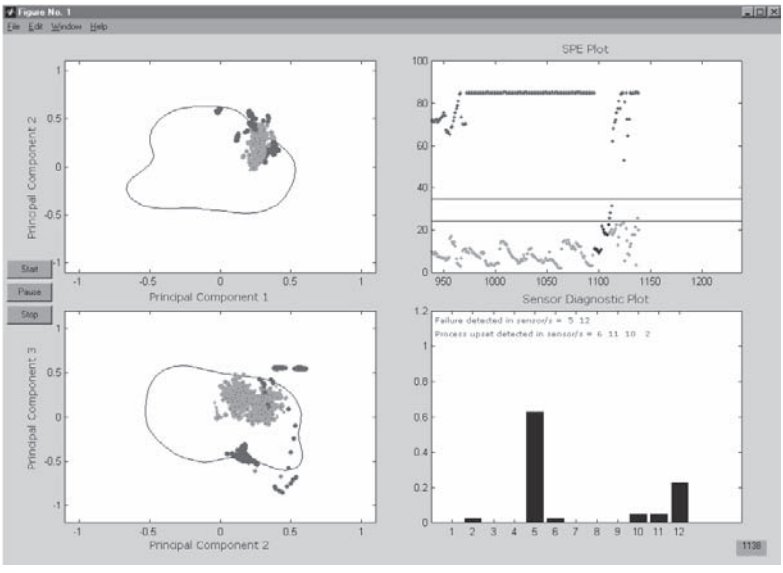


Figure 7.37. The screen shot of the on-line monitoring strategy, indicating the flooding condition. Reprinted from [60]. Copyright © 2001 with permission from Elsevier.

instant. Sensors that are suspected as failed or influenced by a process upset are plotted using a bar chart that shows their normalized squared residuals at the instant of failure. The figure also shows the SPE plot when the faulty sensor was reconstructed using Eq. 7.17.

Next, flooding is introduced. As the flooding progresses through the column stages (Figure 7.37), sensors 5 and 12 were highlighted as faulty, whereas sensors 2, 6, 10, and 11 were isolated as signifying a process upset. Since there was no actual sensor failure, this shows that the BSSIR algorithm could not distinguish correctly between sensor failure and process upset leading to false positive identification. The reasons for this could be that the correlation among sensors is not sufficiently strong, resulting in large deviation in the CC between the sensor labelled as faulty and the most correlated one; or that the CC criterion measures the degree of linearity among variables, hence if some variables are nonlinearly correlated, the CC will again be small. Nevertheless, this ambiguity in the fault detection strategy is noted.

7.8 Summary

Several fault diagnosis methodologies have been demonstrated to stress the variety and availability of techniques that can be deployed in practical applications. First, hidden Markov models (HMMs) have been developed either to solve the state estimation problem for detecting faults or, in conjunction with wavelets and triangular episodes, to solve the maximum likelihood problem for assigning fault classes. Next, multivariate statistical techniques have been used to develop fault diagnosis strategies that are based on PCA and contribution plots that are also extended to robust strategies to deal with measurement noise/outliers and nonlinear correlations among process variables.

8

Sensor Failure Detection and Diagnosis

Sensor auditing is an important component of statistical process monitoring (SPM). The sensors generate a wealth of information. This information is used for monitoring and controlling the process. Misleading information can be generated if there is a bias change, drift or high levels of noise in some of the sensors. Erroneous information often causes control actions that are unnecessary, resulting in the deterioration of product quality, safety and profitability [224]. Identifying failures such as a broken thermocouple is relatively easy since the signal received from the sensor has a fixed and unique value. Incipient sensor failures that cause drift, bias change or additional noise are more difficult to identify and may remain unnoticed for extended periods of time. Consequently, early detection and diagnosis of such faults followed by timely reporting of the analysis can assist plant operators in improving product quality, process safety and profitability.

The fundamental idea is to utilize additional relevant process information for assessing the correctness of information generated by a sensor. This approach is known as the *functional redundancy* and it is more attractive than physical redundancy by duplicating sensors and using a voting logic to select the correct information. Several techniques based on statistics and system theory have been developed for validation of sensor information by functional redundancy. In most of these techniques, it is assumed that detailed process information is available *a priori*. Often, this knowledge is in the form of an accurate state-space model [39, 230]. In many cases, this type of accurate representation of a chemical process based on first principles is not available.

This chapter introduces two sensor audit strategies that can detect and diagnose sensor faults. The first strategy (Section 8.1) focuses on sensor auditing by using calibration and test data sets that are processed either by developing PLS models (for data with low autocorrelation) or canoni-

cal variates state-space (CVSS) models (for strongly autocorrelated data) The software that implements these techniques can be integrated with a knowledge-based system (KBS) to diagnose the cause of the abnormal information received from the sensors and to discriminate between sensor faults and process upsets. The second fault detection and diagnosis (FDD) strategy (Section 8.2) uses PCA models to assess the contribution of each sensor to the T^2 and SPE of the calibration model. This information is used to first isolate the sensors causing out-of-control signal and then distinguish between process upsets and sensor failures. It also reconstructs estimated values for malfunctioning sensors.

8.1 Sensor FDD Using PLS and CVSS Models

Two variants of a technique which relies on input-output models developed from operation data are presented: the first uses PLS and the second CVSS models. PLS regression based on the zero lag covariance of the process measurements was introduced in Section 4.3. A *Multipass PLS* algorithm is developed for detecting simultaneous multiple sensor abnormalities. This algorithm is only suitable for process measurements where the successive measurements are not correlated. The negligible autocorrelation assumption is justified for a continuous process operating at steady-state and having only random noise on measurements.

CVSS models introduced in Section 4.5 for process data with strong autocorrelation and crosscorrelation. One-step ahead residuals generated from the CVSS model will be used for sensor audit. A *multipass* version of the technique is developed to identify submodels by eliminating successively the corrupted measurements from both the calibration and test data sets and fitting a different CV state-space model.

Multipass PLS Algorithm for Sensor Audit

The *Multipass PLS* based sensor monitoring technique was proposed by Negiz and Cinar [208]. The use of residuals obtained from multivariate calibration techniques in sensor audit is due to Wise *et al.* [329]. However, their technique was not suitable for identifying and isolating simultaneous multiple sensor faults. The proposed algorithm addresses this fault isolation problem by eliminating the adverse effects of corrupted measurements of other sensors which can cause false alarms. The technique consists of two main steps. First, a *calibration* model is obtained by using a data set collected over a period of time without known sensor faults and operating under acceptable operating conditions. Then, new (test) data are used with

the *calibration* model to perform the sensor audit.

Assume that p sensors are monitored. The *calibration* data set is collected for n time steps and the measurements are stored in the block matrix $\mathbf{X}$ after being scaled to unit-variance and zero-mean columns. $\mathbf{x}(k)$ is the $p \times 1$ vector of observations at the kth sampling time. The transpose of $\mathbf{x}(k)$ is the kth row of $\mathbf{X}$.

Denote by $\widehat{\mathbf{X}}_{\bullet i}$ the ith $n \times 1$ column vector which contains the predicted values of the ith $(i = 1, ..., p)$ sensor from the $\mathbf{X}$ block excluding itself (the ith column of $\mathbf{X}$) given as

$$\widehat{\mathbf{X}}_{\bullet i} = \mathbf{X}\bar{\boldsymbol{\beta}}_{\mathbf{i}} \tag{8.1}$$

where the $p \times 1$ vector $\bar{\beta}_{\mathbf{i}}$ is the zero augmented PLS regressor vector given as

$$\bar{\boldsymbol{\beta}}_i = [\beta_i(1) \ \cdots \ \beta_i(i-1) \ \ 0 \ \ \beta_i(i) \ \cdots \ \beta_i(p-1)]^T \tag{8.2}$$

The $p-1$ nonzero elements of the regressor vector β_i for each variable i are computed by using PLS algorithm where the predicted variable block $\mathbf{Y}$ contains the measurements of the ith sensor taken from $\mathbf{X}$, and the predictor block denoted by $\mathbf{X}$ contains the observations from the remaining $p-1$ sensors. Equations 8.1 and 8.2 are defined such that the $n \times p$ matrix $\mathbf{X}$ which contains the measurements from all the p sensors is utilized directly. These PLS regressions are repeated for $i = 1, \cdots, p$. For the ith sensor, the $p-1$ elements of the regressor vector β_i are [329]

$$\beta_i = b_1 \mathbf{w}_1 \mathbf{q}_1^T + \sum_{l=2}^{n_{pls}(i)} b_l \left[\prod_{j=1}^{l-1} [\mathbf{I} - \mathbf{w}_j \mathbf{p}_j^T] \right] \mathbf{w}_l \mathbf{q}_l^T \tag{8.3}$$

where the quantities b_l, $\mathbf{w}_l$, $\mathbf{q}_l$, $\mathbf{p}_l$ for $l = 1, \ldots, n_{pls}$ are as defined in the PLS algorithm, and $n_{pls}(i)$ denotes the number of PLS components retained to model of the ith sensor. The vectors $\mathbf{q}$ are scalars with absolute values equal to unity, because the dependent variable block $\mathbf{Y}$ in the PLS algorithm has a single column. Define

$$\bar{\mathbf{B}} = \left[\bar{\boldsymbol{\beta}}_i \ \cdots \bar{\boldsymbol{\beta}}_p\right] = \begin{bmatrix} 0 & \cdots & \beta_p(1) \\ \beta_1(1) & \cdots & \beta_p(2) \\ \vdots & \cdots & \vdots \\ \beta_1(p-1) & \cdots & 0 \end{bmatrix} \tag{8.4}$$

Then, the residual $n \times p$ block matrix $\mathbf{R}$ is given as

$$\mathbf{R} = \mathbf{X} - \hat{\mathbf{X}} = \mathbf{X}(\mathbf{I} - \bar{\mathbf{B}}) = \mathbf{X}\mathcal{M}_{PLS} \tag{8.5}$$

where the $p \times p$ matrix $\mathcal{M}_{PLS}$ represents the transformation from the original variables ($\mathbf{X}$) to the unscaled residuals ($\mathbf{R}$). The residuals can be scaled to unit variance by noting that their 'in-control' covariance is

$$Cov(\mathbf{R}) = \mathcal{M}_{PLS}^T Cov(\mathbf{X})_o \mathcal{M}_{PLS} \tag{8.6}$$

The unit variance residual block $\mathbf{R_s}$ is

$$\mathbf{R_s} = \mathbf{X}\ \mathcal{M}_{PLS}\ \mathrm{diag}(\mathcal{M}_{PLS}^T\ Cov(\mathbf{X})_o\ \mathcal{M}_{PLS})^{-\frac{1}{2}} \tag{8.7}$$

The correlation matrix ($Cov(\mathbf{R_s})$) becomes

$$\begin{aligned} Cov(\mathbf{R_s}) \quad = \quad & \mathrm{diag}(\mathcal{M}_{PLS}^T\ Cov(\mathbf{X})_o\ \mathcal{M}_{PLS})^{-\frac{1}{2}}\ \mathcal{M}_{PLS}^T\ \bullet \\ & Cov(\mathbf{X})_o\ \mathcal{M}_{PLS}\ \mathrm{diag}(\mathcal{M}_{PLS}^T\ Cov(\mathbf{X})_o\ \mathcal{M}_{PLS})^{-\frac{1}{2}} \end{aligned} \tag{8.8}$$

The PLS residual transformation given by Eq. 8.7 is diagonally dominant. Hence, if a change occurs in the mean or variance of any sensor in $\mathbf{X}$, its corresponding residual will be affected the most. This result will be used in resolving multiple sensor faults occurring simultaneously.

The mean and variance of the residuals for each variable is computed using $\mathbf{R}$ in Eq. 8.5. The null distribution of the residuals is assumed to be *iid* and normal. This null hypothesis is tested against significant autocorrelation in the residuals for each variable by using their autocorrelation coefficients at various lags. If 5% of those autocorrelation coefficients are between $\pm 1.96/\sqrt{n}$, where n is the number of data points, then with 95% confidence the distribution is *iid*. To assess the normality of this distribution a standard quantile test is appropriate. The marginal normal distributions for each residual are characterized by computing its mean and variance from the respective column of the residual matrix $\mathbf{R}$. If there is a significant autocorrelation for residuals, then the procedure that generates the residuals should be modified.

After the PLS model is developed with the in-control (calibration) data set, the statistics for the residuals are computed for setting the null hypothesis. A test sample block of size $n_t \times p$ is taken from the process measurements. The residual statistics for the test sample are then generated by using the PLS model developed. The statistical test compares the residual statistics of the test sample with the statistics of the *calibration* for detecting any significant departures.

Denote by $\mathbf{R}_{\bullet i}$ the ith $n \times 1$ residual vector column from the $n \times p$ residual block matrix $\mathbf{R}$. The statistic for testing the null hypothesis of the equality of means from two normal populations with equal and unknown variances is [64]

$$\frac{\bar{\mathbf{R}}_{\bullet \mathbf{i}_{test}} - \bar{\mathbf{R}}_{\bullet \mathbf{i}_{model}}}{\hat{\sigma}_{p_i}\sqrt{1/n + 1/n_t}} \sim t_{n+n_t-2} \tag{8.9}$$

where $\bar{\mathbf{R}}_{\bullet \mathbf{i}_{test}}$ and $\bar{\mathbf{R}}_{\bullet \mathbf{i}_{model}}$ denote the maximum likelihood estimates of the residual means for the variable i in the test sample and the *calibration* set, $\hat{\sigma}_{p_i}$ is the pooled standard deviation of the two residual populations for the ith variable, n and n_t denote the sizes of the *calibration* and testing populations, and t_{n+n_t-2} is the t distribution with the $N + N_t - 2$ degrees of freedom.

The statistic for testing the hypothesis of equality of variances from two normal populations with unknown means is [64]

$$\frac{\hat{\sigma}^2_{i_{test}}}{\hat{\sigma}^2_{i_{model}}} \sim F_{n_t-1,n-1} \tag{8.10}$$

where $F_{n_t-1,n-1}$ is the F distribution with respective degrees of freedom. The level of the test for all the testing statistics is chosen to be 5% and two-sided. This portion of the testing procedure is similar to the algorithm in [329].

Detection of Simultaneous Multiple Sensor Faults with Multipass PLS Algorithm

The flowchart of the multipass sensor audit algorithm is given in Figure 8.1. The model stated in the first block can be a PLS model as discussed so far in this section or a CVSS model. The algorithm checks if the residual mean and/or variance are out of the statistical limits (based on t and F probability distributions) for each sensor variable (Figure 8.1). Since the mathematical structure of PLS allows the corrupted variable to affect the predictions of the remaining ones, false alarms might be generated unless the corrupted variable is taken out from both the *calibration* and the *test* data block. The information loss due to taking the variable out of both the calibration and the test sample set is compensatory since the testing procedures are based on the *iid* assumption of the residuals and not on the minimum prediction error criterion by the model. The variable with the highest corruption level is discarded by comparing the ratios of its residual variance and its residual mean to their statistical limits which are based on Eqs. 8.9 and 8.10.

Example The proposed algorithm is applied to detect multiple sensor failures such as drift, noise, bias and bias plus noise for simulation data obtained from a HTST pasteurization model [211]. The sensors associated with the measurement of each process variable are numbered as listed in Table 8.1.

Calibration data are generated by the simulator by setting the nominal values of process inputs (residence time in the holding tube, and the steam temperature) such that the product has at least 15 *sec* residence time and

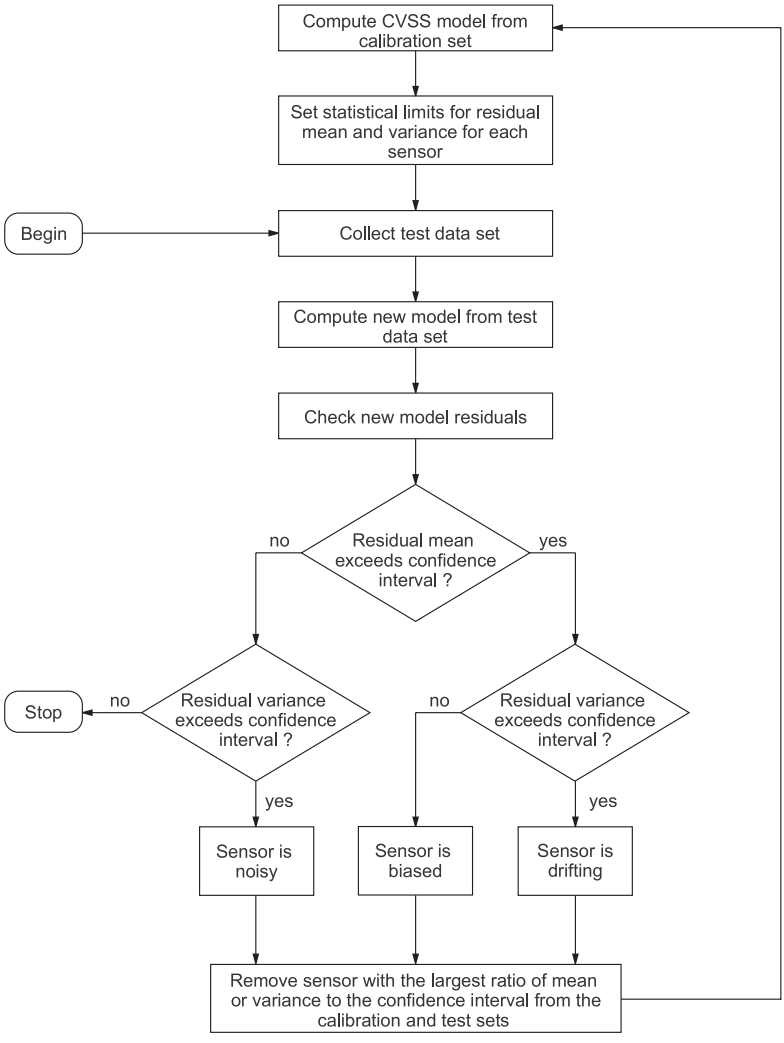


Figure 8.1. The multi-pass PLS or CVSS sensor audit algorithm for detecting simultaneous faults suggested in [290].

the holding tube exit temperature is around 77^oC (170^oF) and by randomly manipulating the nominal inputs around their steady-state values. The calibration data are scaled to zero-mean and unit-variance, and a separate PLS model is obtained for each variable by using the remaining six variables ($\mathbf{X}$) by following the modeling process of Eqs. 8.1–8.5 to obtain the *calibra-*

Table 8.1. Process Variables in the HTST Pasteurization Model

Variable	Description
1	Temperature at the exit of the holding tube (^oC)
2	Temperature at the inlet of the holding tube (^oC)
3	Feed temperature to the tank (unpasteurized product) (^oC)
4	Exit temperature from the tank (unpasteurized product) (^oC)
5	Temperature at the inlet of the steam heater (^oC)
6	Steam temperature (^oC)
7	Residence time in the holding tube (sec)

tion models. The residual means and variances for these *calibration* models which includes all of the seven variables are given with respect to each variable in Figure 8.2, respectively. The dashed lines indicate the upper and lower 95 percentile points of the t and F distributions for the residual means and variances, respectively. The assumption in obtaining those two statistics is checked with the prescribed autocorrelation coefficient test as a function of the lag time. No significant correlation is detected for the residuals of variable 1 since all of the correlation coefficients are within expected bounds (Figure 8.2c). The autocorrelation tests for the other variables also indicate no significant correlation. The order of the PLS model is increased until 95% of the variability in $\mathbf{X}$ is used to predict $\mathbf{Y}$.

Based on the *calibration* PLS model, the 7×7 transformation given in Eq. 8.7 is

$$\begin{bmatrix} 7.91 & -2.91 & -0.04 & -0.03 & -4.18 & -2.29 & -0.37 \\ -2.69 & 8.08 & 0.01 & -0.03 & -1.66 & -4.37 & 0.16 \\ -0.07 & 0.08 & 1.01 & -0.15 & -0.02 & 0.00 & -0.10 \\ -0.15 & -0.08 & -0.16 & 1.02 & 0.16 & 0.10 & 0.11 \\ -2.79 & -1.21 & -0.03 & 0.14 & 5.84 & 0.12 & -0.41 \\ -2.45 & -3.95 & 0.00 & 0.05 & 0.09 & 6.65 & 0.84 \\ -0.09 & 0.00 & -0.12 & 0.15 & -0.02 & 0.16 & 1.04 \end{bmatrix} \tag{8.11}$$

The diagonal entries in the transformation matrix (Eq. 8.11) are the maximum entries across their corresponding column or row. This diagonal dominance ensures that scaled residual of a specific variable will show a change the most if a change occurs in the original variable that corresponds to it. Therefore the ratio of the original residual to its detection limits, which is a linear function of its standard deviation, is a proper statistic to observe this impact without scaling of the original residuals. The hypothesis testing is done for a new data set by comparing its variability with that of the *calibration* set.

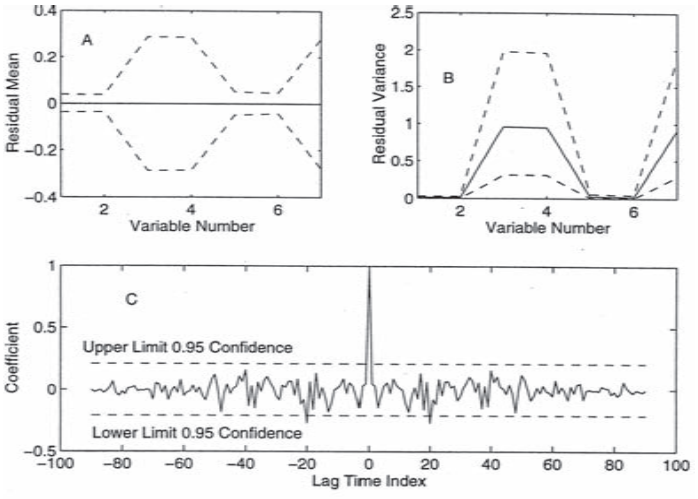


Figure 8.2. The residual statistics for calibration model. (a) Residuals means, (b) residuals variances, (c) autocorrelation coefficients of residuals of variable 1. Variable numbers are based on Table 8.1.

Consider an illustrative example. When the HTST pasteurization data shown in Figure 8.3 are inspected, the operator would decide to take action because the temperature at the exit of the tube is drifting upwards (Figure 8.3a) and the residence time is below the specified limit (Figure 8.3b). In reality these data were artificially produced by adding noise to the flow sensor (variable 7) and a drift to the product temperature sensor (variable 1).

A plot of the residuals means and variances of the corrupted data (Figure 8.4) shows that variables 1, 2, 4, 5, and 6 may be drifting (both the means and variances of residuals are out of bounds); variable 3 may be biased (residual mean out of bounds but residual variance within the bounds); variable 7 may be corrupted by noise (residual variance out of bounds but residual mean within the bounds). In Figure 8.4c, the ratios of the residual variances to the detection limits are plotted for each variable, in order to assess which variable should be excluded from the model. Variable 1 shows the highest ratio and therefore is excluded from the model with a sensor drift diagnosis since both its residual mean and variance is out of detection limits. The *calibration* data set for the remaining variables are used to develop the new PLS regression model and the test data of the remaining variables are used to generate the residuals and their statistics. The new means and variances excluding variable 1 (set to zero) are shown in Figure

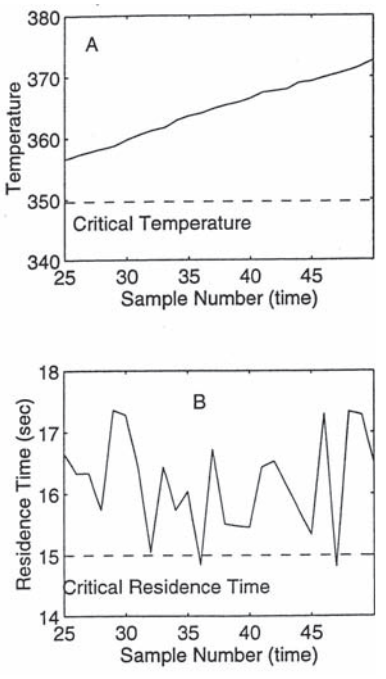


Figure 8.3. Simulated test data set for the HTST pasteurization system. (a) Holding tube outlet temperature, (b) residence time in holding tube.

.

8.5.

All remaining residuals means are within their statistical bounds (Figure 8.5a), whereas the residual variance of variable 7 is out (Figure 8.5b). The ratios of the variances to the detection limits (Figure 8.5c) indicate that variable 7 is corrupted by noise. The plot of the residuals means and variances excluding variables 1 and 7 (Figure 8.6), indicates that they are all within the limits. Consequently, the responses observed in Figure 8.3 for the variables 1 and 7 were the result of a drift in the holding tube exit thermocouple and a noise corruption in the flow rate sensor.

Drift in a sensor or a combination of bias change and noise inflation would affect both the means and variances of the residuals. To determine which one of these sensor faults has occurred, an additional test is performed. The data for the variable under study is filtered by using a moving average filter in order to eliminate the effects of process/instrument noise. If the filtered data have a non-stationary mean (changing over time as a

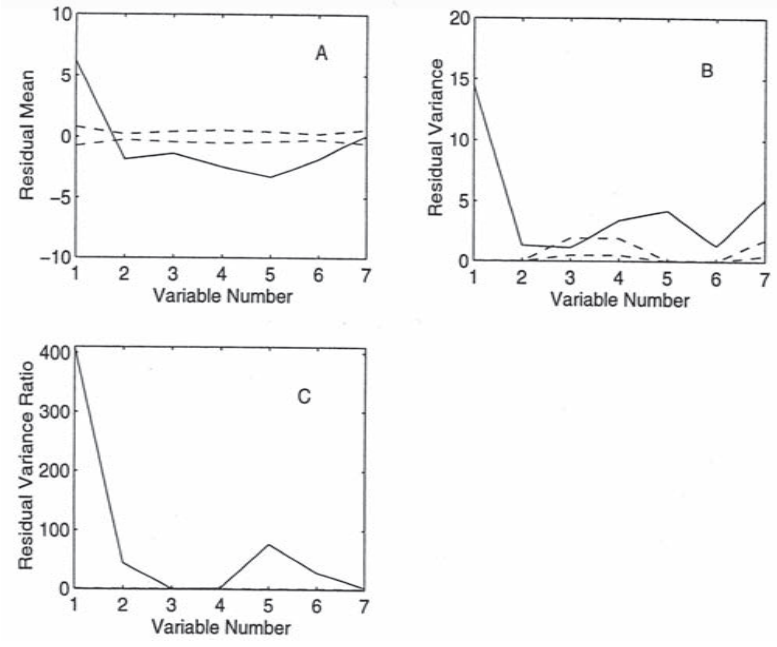


Figure 8.4. Plots of residuals statistics for test data. (a) Residuals means, (b) residuals variances, (c) residuals variance ratios to detection limits. Variable numbers are based on Table 8.1.

ramp), then the sensor is drifting. If the mean of the filtered data is stationary, the sensor has a bias change and increased noise.

The PLS-based sensor fault diagnosis algorithm is based on the zero-lag correlation structure (Gaussian covariance). The PLS model residuals should not have significant autocorrelation. Otherwise, the conditions for the statistical tests given by Eqs. 8.9 and 8.10 are violated and the *Multipass PLS* algorithm should not be used for sensor auditing. Instead, the CVSS model should be used to capture the dynamic behavior of the autocorrelated process measurements and the one-step-ahead residuals based on the CVSS model *calibration* data are used to generate the residuals statistics. The functional redundancy generated by the CVSS model provides residuals which are essentially *iid*. The residual statistics generated from the CVSS model of the *calibration* data set is used to test the hypothesis of no change against the hypothesis of change for a test data set. Sensor auditing equivalent to the PLS-based methodology (consistent with the residuals statistics testing schemes given in Eqs. 8.9 and 8.10) is developed

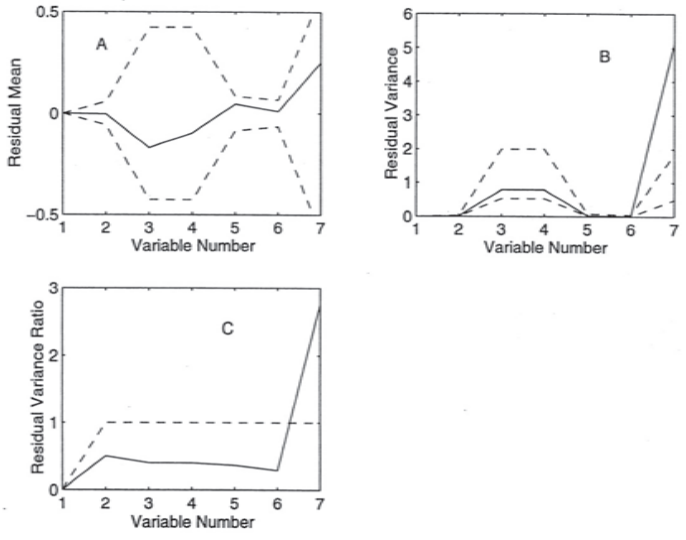


Figure 8.5. Plots of residuals statistics for test data after elimination of Sensor 1 data. (a) Residuals means, (b) residuals variances, (c) residuals variance ratios to detection limits. Variable numbers are based on Table 8.1.

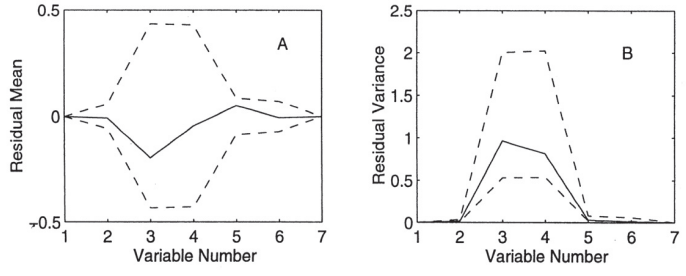


Figure 8.6. Plots of residuals statistics for test data after elimination of Sensor 7 data. (a) Residuals means, (b) residuals variances. Variable numbers are based on Table 8.1.

by inspecting the test data in batches of length n_t. However, the CVSS form can also be utilized to generate the residuals on-line. In this case, the statistical hypothesis tests (Eqs. 8.9 and 8.10) have to be modified. The statistical thresholds obtained with the assumption of normal marginal distributions should be modified, since the central limit theorem is no longer

applicable. Appropriate marginal distribution functions of the residuals can be developed in terms of Lambda distributions. Using thresholds on the means and variances based on the Lambda distributions solves the on-line implementation of the sensor auditing scheme [207]. The details of the CVSS model-based sensor auditing method and a case study based on experimental data are given in [212].

Either method can be embedded in a real-time KBS in order to diagnose the source cause of an abnormal sensor reading to discriminate between sensor failures and process upsets caused by disturbances and equipment failures [290].

Example Monitoring of polymerization of vinyl acetate in a continuous stirred tank reactor illustrates the performance of such an integrated monitoring system. The simulation is based on a model developed by [291] that consists of four ordinary differential equations for the reactor temperature, solvent volume fraction, monomer volume fraction and the initiator concentration in the reactor, as well as three differential equations for the molecular weight moments of the reactor. The moments are functions of the polymer chain reaction kinetics as well as the probabilities of polymer chain propagation, and are used for the calculation of the various polymer molecular weights, polydispersity and conversion. Variables that are 'measured' and displayed by the KBS include the polydispersity, reactor temperature, conversion and the reactor initiator concentration. The five manipulated variables are the reactor cooling jacket temperature, the initiator concentration in the feed stream, the feed stream temperature, the feed solvent volume fraction, and the residence time. The four monitored system output variables are assumed to be available via analytical methods at one minute intervals for the physical system. The assumption is valid for the reactor temperature, conversion and initiator concentration, though the polydispersity measurement in a physical system may take up to 30 min or more to obtain via analytical monitoring techniques. The manipulated variables are modified by adding random fluctuations to each of the inputs. A case study illustrates the synergy between the multivariable monitoring charts and the sensor audit routine.

A bias is added to the reactor conversion measurement with a magnitude of about 5 % of the current reactor conversion (sensor 2). Immediately after the bias change is introduced to the sensor, both the univariate chart for reactor conversion and the multivariate T^2 and SPE charts (Figure 8.7) indicate an abnormality. A CVSS model is developed to generate the residuals for sensor audit. The KBS automatically begins the sensor validation routine to determine the source cause of the inflated T^2 and SPE statistics. Figure 8.8 shows that the residuals mean for sensors 1 (initiator concentration), 2 (reactor conversion), and 4 (polydispersity) have exceeded

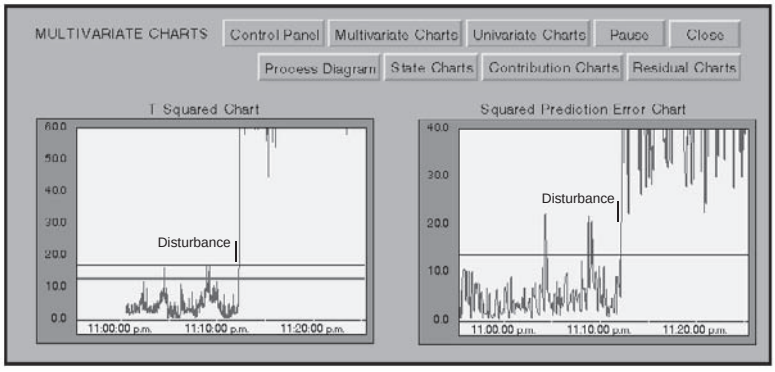


Figure 8.7. T^2 and SPE charts after a bias change is introduced to the reactor conversion sensor. The time of occurrence of the disturbance is indicated by a short vertical bar [290].

their statistical limits. Sensor 2 has the highest ratio of the residual mean to the limit, and is therefore removed from the calibration set and test data sets and a new CVSS model is developed automatically by the software.

After the new CVSS model is constructed, the T^2 and SPE statistics immediately return to within the NO limits (Figure 8.9). The statistical limits for the T^2 and SPE are automatically recalculated by the KBS, depending on the number of sensors monitored by each statistic. The univariate chart for the reactor conversion continues to indicate that the process is operating out of control. However, the sensor audit routine has successfully shown that only the reactor conversion measurement is biased and the remaining sensors are operating correctly as indicated by the in-control residual means and variances after sensor 2 is removed from the model and test data sets (Figure 8.10). The KBS reveals that the reactor conversion (sensor 2) is corrupted by a bias change.

8.2 Real-Time Sensor FDD Using PCA-Based Techniques

In this section, a calibration model will be constructed using the principal components analysis (PCA) (see Chapter 3). A common approach to isolate process disturbances is to check the contribution of each sensor to the sum of squared prediction error (SPE) of the calibration model.

Sensor failures are usually localized in nature however a process upset

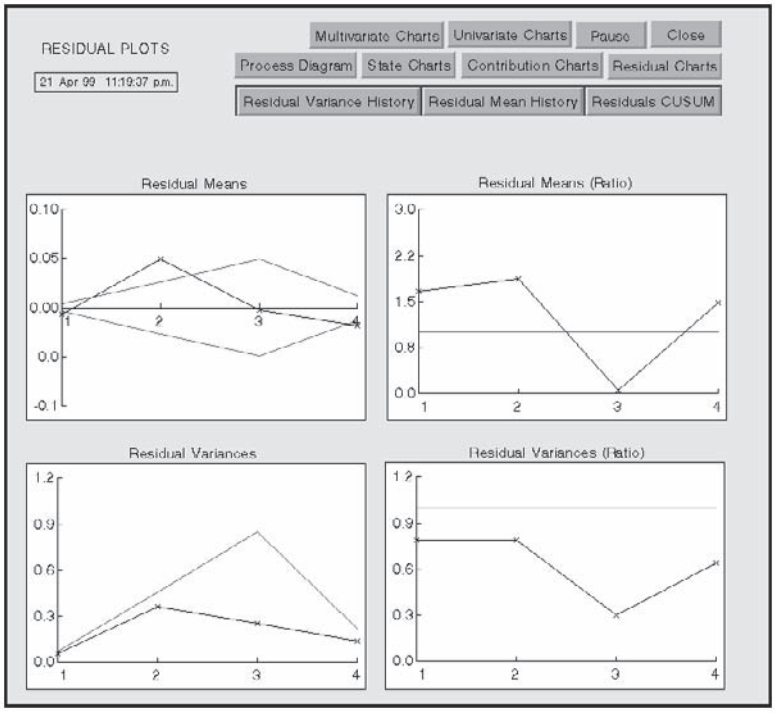


Figure 8.8. The residual means and variances (dotted line) for the test data set and the statistical limits (solid line) after a bias change is introduced to the reactor conversion sensor. Variables shown are reactor initiator concentration (1), reactor conversion (2), reactor temperature (3) and reactor polydispersity (4) [290].

can manifest itself in multiple sensors. To distinguish between process upsets and sensor failures, Stork and Kowalski [283] proposed the redundant sensor voting system (RSVS). This strategy utilizes the work proposed by Stork *et al.* [284], which is called backward elimination for sensor identification (BESI). In the BESI approach, once the SPE_{limit} is violated at a given time, every sensor is sequentially removed from the model matrix followed by calculation of the upper control limit. If the ratio of the SPE/SPE_{limit} is less than one, then algorithm terminates and points to the sensor/s that are left out for the out-of-control signal. Otherwise, the procedure is continued until SPE/SPE_{limit} ratio drops below one. The drawbacks of this approach are that, first, it is computationally expensive, and, second, it is almost impossible to incorporate it within a nonlinear PCA framework s-

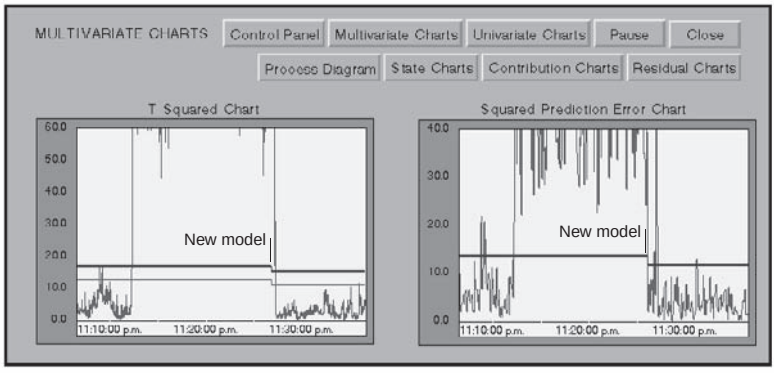


Figure 8.9. T^2 and SPE charts indicating normal operating conditions after Sensor 2 is removed from the calibration and test data sets. The time when the new model was developed is indicated by a short vertical bar [290].

ince every time the SPE_{limit} is exceeded, a new neural network needs to be trained to check the ratio of SPE/SPE_{limit}, and, finally, the SPE space may not reveal all process upsets. On the other hand, the RSVS approach is a three-step procedure:

1. identification of the redundancies among process sensors and determination of minimum redundancy bandwidth,
2. ordering of sensors with redundant sensors in close proximity, and
3. application of probabilistic and/or empirical rules using the disturbance pattern identified for a new process measurement.

Prior to applying the RSVS, disturbances need to be identified correctly (via BESI, for instance) and then RSVS can distinguish whether the disturbance is a sensor malfunction or a process upset. Stork and Kowalski [283] stated that false alarms might lead the RSVS to misdiagnose the source of the disturbance.

In the following, the importance of using the T^2 and the SPE together will be emphasized. This will be followed by a novel strategy, enhanced by sensor reconstruction, that aims to first isolate the sensors causing out-of-control signal and then to distinguish between process upsets and sensor failures. The proposed strategy reduces computational load, and can accommodate both linear and nonlinear PCA, provides reconstructed values for malfunctioning sensors, and potentially eliminates false negative situations.

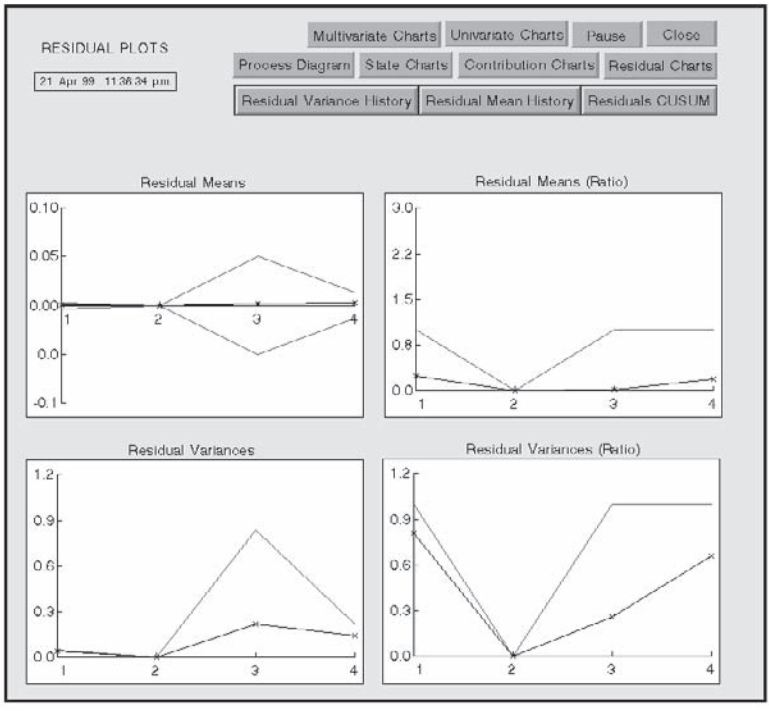


Figure 8.10. The residual means and variances (dotted line) for the test data set and the statistical limits (solid line) after Sensor 2 is removed from the calibration and test data sets. Variables shown are reactor initiator concentration (1), reactor conversion (2), reactor temperature (3), and reactor polydispersity (4) [290].

8.2.1 Methodology

When the process undergoes changes, deviation from the NO region (NOR) is observed as no such features are generalized by the calibration model. Yet, fault detection and isolation based on the information captured by the latent space is somewhat insensitive to changes in the sensor arrays or to small process upsets. Presence of a bias, precision degradation or a drifting failure in a sensor array may cause the measure based on the latent space to remain within the control perimeter, thus, leading to false negative situations. This can be seen by examining the expression for the first latent variable,

$$\mathbf{t}_1 = \mathbf{X}\mathbf{p}_1 \tag{8.12}$$

The ith latent variable can be explicitly written as follows:

$$\mathbf{t}_i = \mathbf{x}_1 p_{i,1} + \mathbf{x}_2 p_{i,2} + \ldots + \mathbf{x}_n p_{i,n} \tag{8.13}$$

Because each latent variable is a linear combination of all variables, a fault in one of the sensors, or small process upsets may not be amplified sufficiently to trigger the alarm and give an indication to out-of-control signal. On the other hand, the SPE measure is more sensitive to such changes compared to the T^2 or the scores plot. This is due to the fact that the error of any type will be propagated to all latent space, thus in the de-mapping part of the PCA, i.e.,

$$\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \ldots + \mathbf{t}_a \mathbf{p}_a^T \tag{8.14}$$

all estimated variables will be influenced by any type of disturbance in the input sequence.

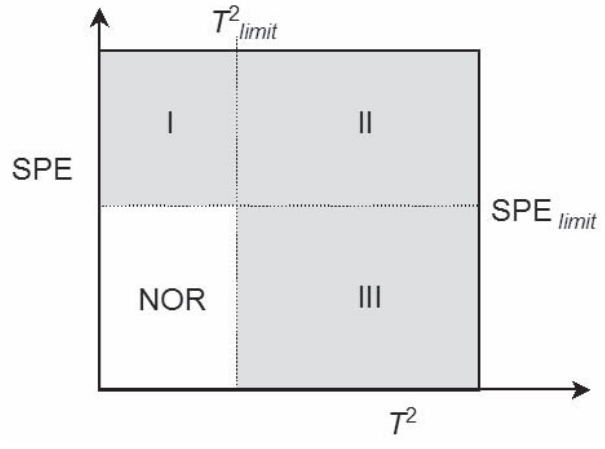


Figure 8.11. SPE vs T^2 plot to illustrate process disturbance detection. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

Hence, at the time instant the disturbance occurs, it is more likely to manifest itself in the SPE by violating the upper control limit than that of the T^2 (Figure 8.11, region I). However, significant changes in the process or in the sensor characteristics can trigger the alarm in both of these measures (Figure 8.11, region II). Nevertheless, there might be process upsets undetected by the SPE due to the extrapolating feature of the calibration model. In such cases, the latent space will capture these changes but no violation in the SPE will be observed (Figure 8.11, region III). This feature

of the PCA model is noticed first and plays an important role in uncovering more of the underlying process upsets that exist in the benchmark example presented next. At any given time, if multiple disturbances occur, depending upon the characteristics of the fault, either statistics will be able to indicate such abnormal conditions.

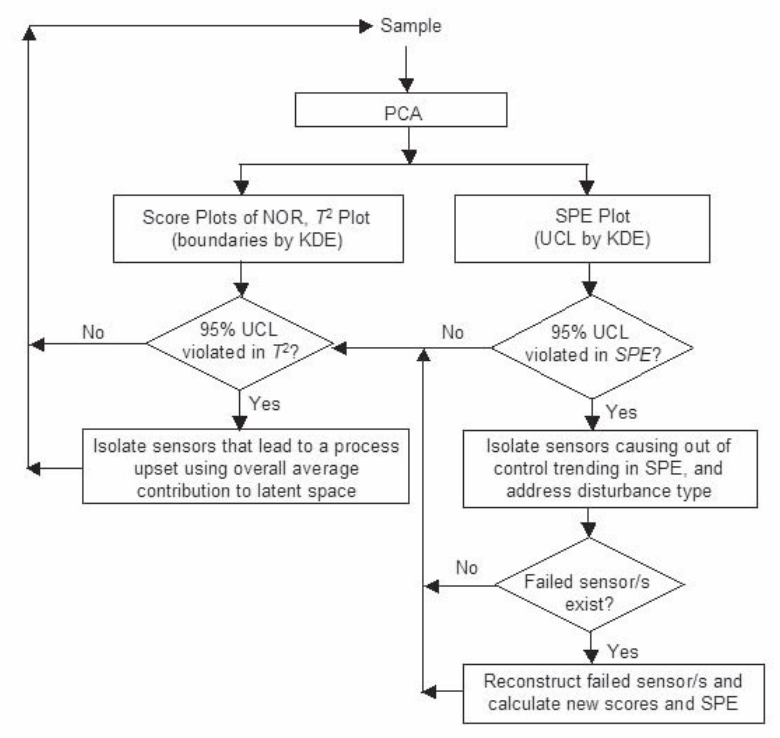


Figure 8.12. The flowchart for the strategy to distinguishing between process upsets and sensor failures. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

It should be noted that it is possible to capture and isolate multiple sensor malfunctions of different types (complete failure, precision degradation, bias, and drifting) at any given time. If a sensor failure and a process upset (pump failure, flooding, or fouling for instance) occur simultaneously, the method is capable to distinguish between these two disturbances as will be seen in the case study (see Section 8.2.2). However, the proposed strategy is incapable of distinguishing among multiple process upsets per sample instant. Angle-based methods of discrimination of samples between

possible groups [243] can be utilized to quantitatively determine the source of multiple process upsets.

The strategy is depicted in Figure 8.12. Once the calibration model is built, the system can be monitored through these steps and in case of abnormal behavior, the algorithm will initiate a search to interrogate the affected sensors. Sensors identified to be causing the problem are checked against a criterion to distinguish whether the root cause of the out-of-control trending is due to a sensor malfunction or a process upset.

Identification of the Source of Disturbances

Process upsets and sensor failures can be identified in the presence of redundancy among sensor arrays [283]. In well-instrumented process settings, there may be many sensors at various locations measuring the same or different quantities such as flow rates, temperatures, pressures, etc. Due to process characteristics, these variables may be cross-correlated with each other and/or autocorrelated in time. Particularly, the ones in close proximity and measuring the same variable may have as low as 0.90 or a higher correlation coefficient. Therefore, when a process upset occurs in the process, its aftereffect will be detected by a group of sensors rather than only by one of them. However, if a sensor malfunction is the case, then this will only appear in the individual sensor response. Nevertheless, if the malfunctioning sensor is measuring a manipulated variable in the control system, the information conveyed to a controller would be inaccurate. As a result, a sensor malfunction may manifest itself in more than one sensor.

Once a calibration model for the process is built using the linear/nonlinear PCA, over the course of operation, the SPE can be used to monitor the process against any unanticipated process changes and/or sensor failures. At times when the SPE_{limit} is violated, a search will be initiated to find out if the underlying cause is due to a sensor malfunction. The correlation coefficient, (CC), will be used as the criterion (see also Section 7.7.1),

$$CC = \frac{Cov(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}} \tag{8.15}$$

For the jth sensor, the most correlated sensor can be found by ordering CCs between the jth sensor and other sensors. To calculate the correlation coefficient when the SPE_{limit} is exceeded in the test set, a $Q \times 1$-size moving window, which contains the current sample and $Q-1$ past samples for each sensor, is formed. The CCs calculated amid plant operation are, then, compared with the ones obtained from the training data. Starting from the first test sample, if a threshold of 10% or more degradation is observed in the CCs between the jth sensor and the most correlated one,

then we will suspect that sensor j is malfunctioning, because a failure in one sensor should not interfere with other sensor readings unless that sensor is conveying information to one of the controllers in the system. If degradation less than the threshold is observed, and no sensor is isolated other than the malfunctioning ones, the search will proceed to the latent space to check if a process upset is present. In Figure 8.11, the points that appear in region I and II may either be due to a process upset or a sensor failure. In addition to malfunctioning sensors revealed by the SPE space, one also needs to identify the ones affected by a process upset, particularly when the points spot in region I in Figure 8.11. The latent space is searched again for a process upset following the reconstruction of the malfunctioning sensors.

Calculation of Variable Contributions

To find variables that triggered the T^2_{limit} or the SPE_{limit}, one needs to find the contribution of each variable the model and the residual spaces. Two approaches will be offered here.

First, the overall average contribution to the modeling space is defined. To detect process upsets in the latent space, the overall average contribution per variable [146] is calculated:

$$C(i,j) = \sum_{\kappa=1}^{a} \{xs(i,j)p(j,\kappa)\} \frac{t(i,\kappa)}{var(t(\kappa))} \tag{8.16}$$

where $xs(i,j)$ denotes scaled values of real observations for ith measurement of jth variable; $p(j,\kappa)$ is loading of variable j in κth principal component; $var(t(:,\kappa))$ is the eigenvalue of the same latent variable; and $C(i,j)$ is the sum of the contribution of variable j to all latent variables. Bracketed summation term in Eq. 8.16 is carried out only if the sign of $\{xs(i,j)p(j,\kappa)\}$ equals the sign of the overall score at observation i. As pointed out by Kourti and MacGregor [146], the way the overall average contribution is calculated is different than the calculation for T^2. If the T^2_{limit} is exceeded at time i, the sign of each weighted variable j should be the same as that of κth latent variable in order for that variable to have any contribution to the overall scores space. This feature makes this approach superior to the method that uses the contribution to T^2 in determining the variables responsible for the out-of-control trending in the modeling space.

The second approach makes use of the contribution to residual space. Process upsets and changes in sensor characteristics can be identified by analyzing the residual space. To investigate variables being affected by such disturbances, the contribution of each variable that ultimately triggers the SPE limit should be calculated as follows:

$$R(i,j) = |x(i,j) - t(i,\kappa)p(j,\kappa)| \quad i = 1,\ldots,n; j = 1,\ldots,m \tag{8.17}$$

where $R(i,j)$ represents the residual of jth variable at time i.

To identify variables causing T^2 and SPE to exceed their thresholds in validation part of the calibration model, the scores contribution index (SCI), and the residual contribution index (RCI) are defined (similar to the one proposed by Raich and Cinar [243]). The SCI and RCI of each variable at any time are defined as the ratio of its contribution values over the contribution upper control limits, which are found by the KDE. Here, the confidence limit for each variable contribution can be defined as

$$\mathbf{C}_{j,\alpha} \approx \Omega_{j,\alpha} \tag{8.18}$$

and

$$\mathbf{R}_{j,\alpha} \approx \Psi_{j,\alpha} \tag{8.19}$$

hence,

$$SCI_{j,\alpha} = \frac{\mathbf{C}_j}{\Omega_{j,\alpha}} \tag{8.20}$$

and

$$RCI_{j,\alpha} = \frac{\mathbf{R}_j}{\Psi_{j,\alpha}} \tag{8.21}$$

where $\Omega_{j,\alpha}$, and $\Psi_{j,\alpha}$ are estimated probability densities of the jth variable contribution to the latent-space and the residual-space, respectively, and α denotes the confidence level. Thus, if the $SCI_{j,\alpha}$ or the $RCI_{j,\alpha}$ at time i exceeds unity, then, variable j is said to be contributing to either of those measures to exceed their limits.

Sensor Reconstruction

In any industrial setting, sensor failure is likely in the midst of plant operation. Maintenance of unavailable sensors is therefore needed as soon as they are detected. However, if the sensor that conveys information to one of the controllers were to be faulty, it is essential that its value be reconstructed from the remaining sensors on-line. Sensor reconstruction can be performed using the calibration model based on the PCA/NLPCA [60, 150, 214].

Here, after identifying the malfunctioning sensor/s, their values are reconstructed using the calibration model. Each unavailable sensor value can be estimated by solving the optimization problem introduced in Section 7.7.1. By minimizing Eq. 7.17, a one-dimensional search is established over the missing values to obtain a solution to the problem. Meanwhile, the values of the remaining sensors are kept constant while the optimization is carried out. Multiple sensor failures can also be accommodated in an analogous way provided that retained number of principal components is equal to or less than the number of the available sensors. In this case, however,

Table 8.2. Known disturbances in the test set. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

Sensor No.	Time Instant	Disturbance Type
1	210-342	Sensor Malfunction
5	50-342	Sensor Malfunction
6-9	33,92-93,156,331-332	Process upset (cold cap)

the problem becomes a multidimensional search for values of the missing sensors that minimizes Eq. 7.17.

8.2.2 Case Study

This study involves a data set collected from an LFCM unit of a slurry-fed ceramic melter (SFCM) [330]. The SFCM vitrifies a mixture of the original wastes, contaminated zeolite and glass forming additives of the nuclear fuel reprocessing campaigns. The data set comprises 792 observations measured over 21 process variables. The melter temperature is monitored at 20 locations, and the resistance and power dissipated between each of the electrode pairs, glass tank level and feed flow rate are also recorded. In all, 29 variables are measured, but among these only the temperatures and glass level recordings are reported [328]. Although the process is sampled at a much faster rate, the recordings were taken at 5 *min* intervals.

A 20-variable process data (temperature recordings only) is considered. This benchmark data set is composed of measurements recorded from temperature sensors. The first 450 measurements are used to build the PCA model and the remaining 342 samples are utilized as a test set. Known disturbances reported for the test set are given in Table 8.2. It was pointed out that not all the disturbances in the test set have been previously identified.

In the model building step, data were first mean-centered. The PCA model utilizes 5 LVs to capture 97% of the variability in the system. To monitor the process, scores plot between the 1st and 2nd LVs, and the SPE vs T^2 plots are considered. KDE with a bivariate Gaussian kernel with a smoothing parameter $h = 45$, is utilized to define the NOR, the SPE_{limit}, and the T^2_{limit}. 95% upper control limit is chosen for these measures, to detect unmodeled disturbances in the system (see Figure 8.13). It is noted that the NOR determined by KDE has almost an ellipsoidal shape, which implies a near Normal distribution of the first two latent variables.

In the validation part of the calibration model, the data set is mean-centered using the mean of the training data. Samples are sequentially

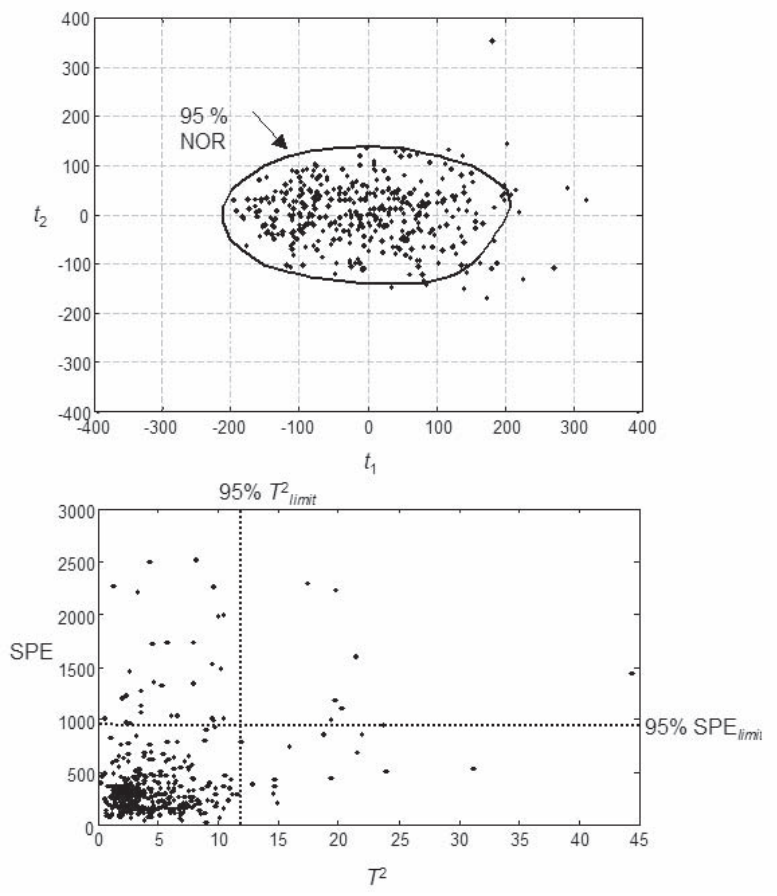


Figure 8.13. The NOR using the scores plot and the SPE/T^2 graph. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

processed following the proposed strategy depicted in Figure 8.12. As long as the incoming samples show no out-of-control trending, the algorithm performs the usual PCA calculation. However, in case the T^2_{limit} or the SPE_{limit} are violated, a search to investigate the underlying cause is initiated. The proposed strategy, thereby, is responsible for detecting the abnormal events, isolate the sensors being affected by its consequences, and differentiate between sensor failures or process upsets. If there is a sensor malfunction, the algorithm minimizes Eq. 7.17 so as to provide reconstructed values. After processing all the test data, results tabulated

in Table 8.3 were obtained with 95% UCL on the SPE/T^2. Bold-faced numbers are the ones matching the instances reported in Table 8.2. The algorithm identified malfunctioning sensors correctly, and uncovered more number of process upsets than listed in Table 8.2. When both sensors were in failure mode, we found that the reconstructed values of few instances also indicate a process upset in the process.

Remark The algorithm diagnosed the first two points (210 and 211) of failing instants in sensor 1 as a process upset instead of a sensor failure. This may be due to the fact that the weight of these two consecutive points in 100-sample size moving window are not affecting the CC between sensor 1 and its closest neighbor (sensor 2) to deteriorate more than the 10% threshold.

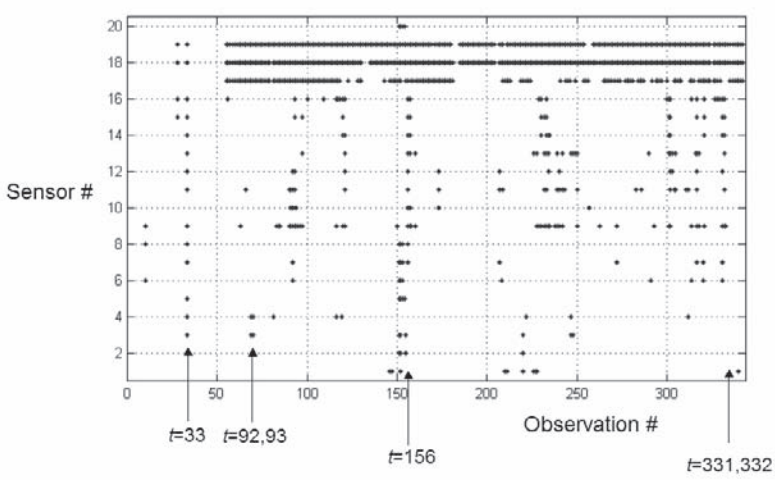


Figure 8.14. Process upsets revealed by sensors over the course of operation. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

The impact of process upsets on the sensors can be visualized by displaying the affected sensors over time as in Figure 8.14. The process upsets are expected to manifest themselves by influencing more sensors at any given instant. For instance, one can observe that at those instants listed in Table 8.2 (e.g., $t = 33$), there were a few more sensors in addition to sensors 6–9 affected by the cause. Figure 8.14 also reveals a much more pronounced disturbance effect, captured by sensors 17–19 in the cold-cap region, which starts at observation no. 55 and lasts till the end of the test set. Although there is no reported incidence to validate the major process

Table 8.3. Disturbances identified by proposed strategy using 95% UCL on SPE/T^2. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

Sensor No.	Fault	Process Upset
1	**212-342**	146-147, 152, 210–211, 220, 226–228, 340
2		151–152, 155, 220
3		**33**, 69–70, 151–152, 155, 220, 247–248
4		**33**, 70, 81, 116, 119, 147, 222, 312
5	**50–342**	**33**, 151-154
6		10, **92**, 151–153, 208, 291, 314, 320, **331**
7		**33**, **92**, 152–153, **156**, 207, 272, 317, 320, **331**
8		10, **33**, 151, 153, **156**
9		10, **33**, 63, 83–85, **90–97**, 116, 120–121, 150, **156**–158, 160, 228-230, 232–235, 238–242, 250, 263, 272, 293, 301-302, 314, 317–318, 321, **331**–333
10		91–**92**, 94, **156**–157, 173, 257
11		**33**, 66, **90–93**, 121, **156**, 173, 207, 209, 232-233, 239–240, 242–243, 250, 283, 286, 302, 304–305, 311–312, 317, **332**
12		**33**, **92–93**,121, **156**, 173, 207, 234, 240, 302–303, 317, **331**
13		**33**, 97, 121, **156**–157, 226, 228, 232-234, 239, 242, 249-250, 290, 301–303, 305, 316–318, **332**
14		**33**, 12–121, **156**–157, 230, 233–235, 301–302, 321, **331–332**
15		28, **33**, 56, **93**, 100, 109, 116–118, 120–121, **156**–157, 229, 230, 233, 300–302, 314, 317, 321, 327–329, **331–332**
16		28, **33**, **93**, 97, 120, **156**–157, 230, 232–233, 301–302, 317, 321, **331–332**
17		55-79, 81-91, **92-93**, 94-118, 123, 128-130, 143, 146-152, 155–**156**, 158-159, 161-181, 209-213, 219-224, 241, 244, 246, 249, 254-256, 265-269, 271, 273-274, 277-287, 292, 294-297, 300, 304-308, 313-315, 318-324, 326-328, 330, 335, 337-342
18		28, **33**, 55-57, 59-91, **92-93**, 94-130, 135-155, **156**, 157-181, 185-204, 207-261, 263-324, 326-342
19		28, **33**, 55-57, 59-91, **92-93**, 94, **156**, 157-169, 171-180, 185-205,207-330,331-332, 333-342
20		151–154

Table 8.4. Source of identification for the sensors affected by the process upset. Results are obtained after reconstruction of faulty sensors, and for those points that belong to region II in Figure 8.15b. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

Sample No.	Latent Space	Residual Space
33	9, 18, 19	3, 4, 5, 7, 8, 9, 11, 12, 13, 14, 15, 16, 18
64	18	17, 18, 19
92	9, 12	6, 7, 9, 10, 11, 17, 18, 19
93	9, 12	6, 7, 9, 10, 11, 17, 18, 19
151	9, 11, 12	9, 15, 16, 17, 18, 19
152	1, 2, 3, 5, 6, 7, 8, 20	17, 18, 19
155	–	2, 3, 17, 18, 19
156	8, 9, 11, 12, 13	7, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19

upset in these sensors, the scores plot after reconstructing the failed sensors and the SPE/T^2 plot (Figure 8.15) consistently indicate the presence of such a process upset. Moreover, similar, but less pronounced, behavior was also observed at $t = 10, 28, 120-121, 151-153, 157, 173, 207, 228-230, 232-235, 238-240, 242, 247, 250, 301-303$ and $320-321$. While these instances also point to the presence of possible process upsets, it should also be recognized that the ones revealed by one or two uncorrelated sensors may be due to small changes in signal characteristics, such as noise.

An interesting realization is the fact that points that fall in region II of the SPE/T^2 (Figure 8.15) were not always caused by the same group of sensors that were affected by the disturbance. Table 8.4 gives few instances when the system was undergoing the disturbances. As one can see, both latent and residual spaces are characterized rarely by the same group of sensors. Furthermore, both the SPE (in region I) and the T^2 (in region III) are capturing different type of disturbances. These findings suggest that fault identification and isolation methods, which utilize the information from the residual or latent space only, will not be able to reveal all the disturbances.

The importance of reconstructing the faulty measurements plays a crucial role in identifying the process upsets inherent in the system. Without reconstruction, these events might go undetected that eventually lead to false negative situations. Thereby, to remedy the masking effect of the faulty measurements that inflate the T^2 and the SPE, reconstruction is vital. As a particular aspect of this example, it was found that the under-

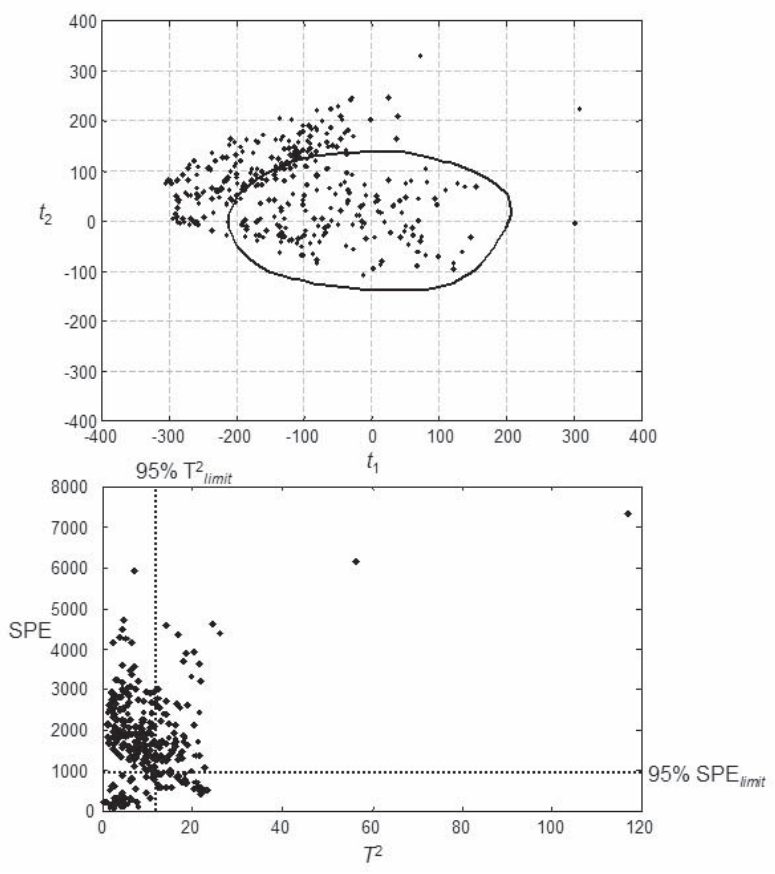


Figure 8.15. Process status for the validation data set. Reprinted from [62]. Copyright © 2001 with permission from Elsevier.

lying reasons that led to a drift from the NOR were not due to the faulty sensors. This realization is a strong indication to present process upsets masked by the failed sensors. It is worth mentioning that due to the masking problem, these disturbances were not correctly identified before in the literature. Therefore, it can be pointed out that the reported list of process upsets may be incomplete.

8.3 Summary

Process sensors are a key element of monitoring strategies as they provide a wealth of information about process status. However, they are also subject to various modes of failure, which can complicate the detection and diagnosis of faults and catastrophic events. In this chapter, two sensor auditing strategies were presented that can aid in the isolation of failed sensors. Based on the concepts of PLS, CVSS and PCA, these sensor audit strategies play a substantial role in discriminating between actual process disturbances and sensor malfunctions, thus helping operators locate the true root cause of process faults. The second method has also shown that the malfunctioning sensors can be reconstructed using measurement information from other sensors.

9

Controller Performance Monitoring

The objective of controller performance monitoring (CPM) is to develop and implement technology that provides information to plant personnel for determining if appropriate performance targets and response characteristics are being met by the controlled process variables. Typical operating targets include limits on deviation from the set-point, limits on manipulated variable moves, variances of controlled and manipulated variables, frequency of soft constraint violations and frequency of reaching hard constraints. These targets can be used as criteria for assessing controller performance. Additional criteria are developed by considering the dynamic response characteristics such as decay ratio, overshoot, response time and response characteristics of the output error and the manipulated variable. Several additional criteria are defined for multivariable systems including the extent of dynamic interactions and loop shaping. Many of these criteria may not be automated easily and various techniques that can compute indexes indicating controller performance have been proposed.

The initial design of control systems includes many uncertainties caused by inaccuracies in process models, estimations of disturbance dynamics and magnitudes, and assumptions concerning the operating conditions [253]. The control algorithm and the tuning parameter values are chosen by using this uncertain information, leading to process performance that can differ significantly from the design specifications. Even if controllers perform well initially, many factors can cause their abrupt or gradual performance deterioration. Sensor or actuator failure, equipment fouling, feedstock variations, product changes and seasonal variations may affect controller performance. It is reported that as many as 60% of all industrial controllers have some kind of performance problem [105]. It is often difficult to effectively monitor the performance and diagnose problems from trends in raw process data [148]. These data show complicated response patterns caused by dis-

turbances, noise, time-varying systems and nonlinearities. In addition, the scarcity of engineers with control expertise to evaluate routinely the large number of control loops in chemical processes makes the analysis of raw data virtually unmanageable. These facts stress the necessity of efficient on-line techniques in controller performance monitoring and diagnosis. Development of on-line tools that can be automated and provide easy to interpret results by plant personnel are desirable.

CPM ensures proper performance of the control systems to enable the process to operate as expected and manufacture products that meet their specifications. CPM and control system diagnosis activities are a subset of the plantwide process monitoring and diagnosis activities. CPM and diagnosis rely on the interpretation of data collected from the process. When an abnormality is detected in process data, it is necessary to determine if it is caused by a control system related cause as opposed to process equipment failure. The sequence of events and interactions can be more complex if for example an equipment failure triggers process variations that are further amplified by the feedback of the control system. This chapter focuses on CPM and diagnosis will be limited to determining if source causes are associated with the controller. Controlled variables should meet their operating targets such as specifications on output variability, effectiveness in constraint enforcement, or closeness to optimal control. A comprehensive approach for assessing the effectiveness of control systems includes: (i) Determination of the capability of the control system; (ii) Development of statistics for monitoring controller performance; (iii) Development of methods for diagnosing the underlying causes of changes in the performance of the control system [105].

Performance criteria must be defined to determine the capability of a control system. A benchmark is established for assessment by using data collected during some period of process operation with acceptable performance. Once these are achieved, controller performance can be monitored over time to detect significant changes. Since control system inputs are random variables, the outputs of the performance measure will be stochastic as well. Therefore, statistical analysis tools should be used to detect statistically significant changes in controller performance. When performance degradation is detected, the underlying root causes have to be identified. Methods for isolating problems associated with the controller from those arising from the process would be very useful. This chapter focuses on CPM of single loop, multivariable and model predictive control (MPC) systems. Diagnosis is illustrated for MPC and is limited to distinguishing between root cause problems associated with the controller and problems that are not caused by the controller [264].

Integration of CPM with diagnosis was reported for single-loop cases

[281]. A recent review [293] summarizes various advances in plantwide CPM for single-loop controllers and integrates CPM with detection of periodic and nonperiodic oscillations in plant operation, valve stiction and root cause of plant disturbances. Diagnostic tools for performance degradation in multivariable model-based control systems have been proposed [141]. Very few uses of KBSs for CPM and diagnosis have been reported [125, 139, 264]. Review papers summarize various approaches for CPM of single-loop, multi-input-multi-output (MIMO), and MPC controllers [236, 238], and detection of valve stiction problems [123, 255]. CPM of MIMO processes by using projections to subspaces [195, 196], and valve stiction by qualitative shape analysis illustrate the diversity of techniques proposed for CPM and diagnosis.

An overview of single-loop CPM is presented in Section 9.1. Section 9.2 surveys CPM tools for multivariable controllers. Monitoring of MPC performance and a case study based on MPC of an evaporator model and a supervisory knowledge-based system (KBS) is presented in Section 9.3 to illustrate the methodology. The extension of CPM to web and sheet processes is discussed in Section 10.3.

9.1 Single-Loop Controller Performance Monitoring

An elegant CPM method based on minimum variance control (MVC) and the variance of the controlled variable computed from routine process data proposed by Harris [102] has initiated the recent interest in CPM. The variance of a controlled variable is an important performance measure, since many process and quality criteria are based on it. The theoretically achievable absolute lower bound on the variability of the output can be an appropriate benchmark to measure the performance of a regulatory control system. This benchmark is achieved by a system under MVC. Using MVC as performance benchmark, one can assess the performance of a control loop and make statements on the potential of improvements resulting from retuning of controller parameters or implementing more sophisticated linear feedback controllers [53]. A good performance relative to MVC indicates that further tuning or re-design of the control algorithm is neither necessary nor helpful. In this case, further reduction of process variability can only be obtained by implementation of feedforward control or re-engineering of the process. A poor performance might result from constraints such as unstable or poorly damped zeros of the process transfer functions or control action limits and indicates the necessity of further analysis such as process identification and controller re-design [115].

Various performance indices have been suggested [54, 53, 149, 20, 148] and several approaches have been proposed for estimating the performance index for SISO systems, including the normalized performance index approach [53], the three estimator approach [175], and the filtering and correlation analysis (FCOR) approach [115]. A model free approach for linear quadratic CPM from closed-loop experiments that uses spectrum analysis of the input and output data has been suggested [136]. Implementation of SISO loop based CPM tools for refinery-wide control loop performance assessment has been reported [294].

The most popular tool for monitoring single-loop feedback and feedforward/feedback controllers is based on relative performance with respect to minimum variance control (MVC) [53, 102]. The idea is not to implement MVC but to use the variance of the controlled output variable that would be obtained if MVC were used as the reference point. The variation of the inflation of the controlled output variance indicates if the process is operating as expected or not. Furthermore, if the variance with a MVC is larger than what could be tolerated, this indicates the need for modification of operating conditions or process.

Following the MVC framework [102, 148], consider a process described by a linear discrete-time transfer function model:

$$y(k) = P(q^{-1})u(k) + \sum_i D_i(q^{-1})d_i(k) + v(k) \tag{9.1}$$

where $y(k)$ is the output, $u(k)$ is the input, $d_i(k)$ is the ith measured disturbance, and $v(k)$ represents the additive effect of noise and unmeasured disturbances at the output. The argument (k) represents discrete time instants. $P(q^{-1})$ and $D_i(q^{-1})$ are stable polynomials corresponding to the transfer functions between the output and the manipulated input or measured disturbance i, respectively. The manipulated input is computed by the controller

$$u(k) = C(q^{-1})e(k) + \sum_i C_{f,i}(q^{-1})d_i(k) \tag{9.2}$$

where $C(q^{-1})$ and $C_{f,i}(q^{-1})$ are the feedback and feedforward controller transfer functions. The output deviation (error) from the set-point $r(k)$ is

$$e(k) = r(k) - y(k) \tag{9.3}$$

By using Eqs. 9.1 and 9.2, the error $e(k)$ can be written as

$$e(k) = \frac{r(k) - \sum_i (D_i(q^{-1}) + P(q^{-1})C_{f,i})d_i(k) - v(k)}{1 + P(q^{-1})C(q^{-1})} \tag{9.4}$$

The dynamic response of $e(k)$ can be expressed as an autoregressive moving average (ARMA) model or a moving average (MA) time series model:

$$e(k) = \frac{\theta(q^{-1})}{\phi(q^{-1})} a(k) = [1 + \psi_1(q^{-1}) + \psi_2(q^{-2}) + \ldots]a(k) \tag{9.5}$$

where $a(k)$ is a random noise sequence with variance σ^2 and ψ_i are the coefficients of the MA model or the impulse weights. Harris and his coworkers [53, 102] have noted that the variance of the closed-loop output is given by

$$\sigma_e^2 = [1 + \psi_1^2 + \psi_2^2 + \ldots + \psi_f^2 + \ldots]\sigma_a^2 \tag{9.6}$$

The output error variance for MVC becomes

$$\sigma_{mv}^2 = (1 + \psi_1^2 + \psi_2^2 + \ldots + \psi_f^2)\sigma_a^2 \tag{9.7}$$

where f denotes the number of time intervals equivalent to the process time delay. Harris [53] defines a performance index

$$\eta(f) = 1 - \frac{\sigma_{mv}^2}{\sigma_y^2} \tag{9.8}$$

The index $\eta(f)$ gives the ratio of the variance in excess of that could be achieved under MVC to the actual variance. If $\eta(f)$ is close to 0 the controller performs closely to the performance of MVC, and $\eta(f)$ values closer to 1 indicate poor controller performance.

Kozub and Garcia [149] point out that in many practical cases, rating of output error characteristics relative to MVC is not practical or achievable. They propose autocorrelation patterns for first-order exponential output error decay trend:

$$e(k) = \frac{1}{1 - \lambda q^{-1}}\, a(k) \quad \text{with} \quad \lambda = exp\,(-\frac{T}{\tau}) \tag{9.9}$$

where T is the sampling interval and τ is the first-order response time constant. The autocorrelation pattern is given by

$$\rho_e(k) = \lambda^k \tag{9.10}$$

which can be compared to the autocorrelation pattern of the error $e(k)$. They define a closed-loop potential (CLP) factor defined as

$$CLP = \frac{\sigma_{mv}^2}{\sigma_e^2} \tag{9.11}$$

For the closed-loop performance bound given in Eq. 9.9, the variance of the output error is

$$\sigma_e^2 = \frac{1}{1-\lambda^2}\sigma_a^2 \tag{9.12}$$

which yields a bound limit for the CLP by noting that $\sigma_{mv}^2 = \sigma_a^2$ if $f = 0$:

$$CLP = 1 - \lambda^2 \tag{9.13}$$

These indexes can be extended to consider the variance ratios of the k-step-ahead forecast error to the variance of $e(k)$. A performance index similar to CLP, CLP_k is defined as [148]:

$$CLP_k = \frac{(1 + \psi_1^2 + \psi_2^2 + \ldots + \psi_k^2)\sigma_a^2}{\sigma_e^2} \tag{9.14}$$

Other enhancements for indexes that originate from the same concepts have been proposed [20, 110, 248] and applications to refinery control loops have been reported [294]. Lynch and Dumont [175] have presented a methodology based on Laguerre networks to model the closed-loop system for computing the minimum achievable variance, an on-line delay estimator, and static input-output estimator for assessing process nonlinearity. Likelihood ratio tests have been proposed to determine if the output error response characteristics are acceptable based on specified dynamic performance bounds [300]. But Kozub [148] warns that this approach is conceptually and computationally too demanding compared to other methods and that reliance on only settling-time specification to construct the likelihood ratio tests [300] may be misleading.

Time series models of the output error such as Eq. 9.5 can be used to identify the dynamic response characteristics of $e(k)$ [148]. Dynamic response characteristics such as overshoot, settling time and cycling can be extracted from the pulse response of the fitted time series model. The pulse response of the estimated $e(k)$ can be compared to the pulse response of the desired response specification to determine if the output error characteristics are acceptable [148].

Cross correlation analysis is proposed for assessing the dynamic significance of measured disturbances and set-point changes with respect to closed-loop error response, and testing the existence of plant-model mismatch for models used in controller design [281].

9.2 Multivariable Controller Performance Monitoring

CPM of *multivariable control systems* has attracted significant attention because of its industrial importance. Several methods have been proposed for performance assessment of multivariable control systems. One approach is based on the extension of minimum variance control performance bounds to multivariable control systems by computing the interactor matrix to estimate the time delay [103, 116]. The interactor matrix [103, 116] can be obtained theoretically from the transfer function via the Markov parameters or estimated from process data [114]. Once the interactor matrix is known, the multivariate extension of the performance bounds can be established. For example, Harris and co-workers [103] propose

$$\eta = 1 - \frac{E[\mathbf{y}_{MV}^T \mathbf{W} \mathbf{y}_{MV}]}{E[\mathbf{y}_t^T \mathbf{W} \mathbf{y}_t]} \tag{9.15}$$

where $\mathbf{W}$ is a positive-definite weighting matrix, $\mathbf{Y}$ is the vector of outputs and $E[.]$ denotes expectation. As an extension of this approach, a filtered optimal H_2 control law with desired closed-loop dynamics has been proposed [114]. Alternatively, multivariate MVC performance might be estimated via multivariate time series analysis [105]. A pass/fail likelihood ratio test was proposed to determine if performance specifications like settling time, decay ratio, minimum variance, or frequency-domain bounds are met [300]. Huang and Shah [115] proposed as benchmark user-specified closed-loop dynamics, like settling time or overshoot. Covariance-based performance indexes and a user-defined benchmark have been presented by Qin and co-workers [195, 196, 238].

Another group of approaches focuses on model-based control systems. The ratio of the desired and achieved controller objective functions, settling time, and constraint violations based criteria have been proposed for a Dynamic Matrix Control (DMC) type model predictive controller [223]. Diagnosis tools for source causes of poor controller performance have also been suggested. A different group of tools for detecting and diagnosing controller performance problems have been suggested by using multivariate statistical tests on the prediction error for detection and casting the diagnosis problem as a state estimation problem [141].

The third class of techniques include a frequency-domain method based on the identification of the *sensitivity function* ($S(s)$) and the *complementary sensitivity function* ($T(s)$) from plant data or CPM of multivariable systems [140]. Robust control system design methods seek to maximize closed-loop performance subject to specifications for bandwidth and peak

magnitude of $S(s)$ and $T(s)$. Estimates of these transfer functions can be obtained by exciting the reference input with a zero-mean, pseudo-random binary sequence, observing the process output and error response, and developing a closed-loop model. Performance assessment is based on the comparison between the observed frequency response characteristics and the design specifications. Selection of appropriate model structures, experimental design and model validation which will ensure reasonable estimates of $S(s)$ and $T(s)$ are discussed in [140]. The method has been automated and embedded in a real-time knowledge-based system for supervisory multivariable control [139]. Since the technique is intrusive, it should be used after one of the nonintrusive techniques discussed earlier indicates a controller performance problem. Because the procedure checks controller performance against design criteria, controller design and tuning via loop shaping techniques provide an automated controller modification opportunity for maximizing performance.

9.3 CPM for MPC

CPM for model predictive control (MPC) systems has been studied in recent years. The availability of a model for MPC offers new alternatives for CPM of MPCs in contrast to multivariable control CPM that is usually data-driven, relying only on routinely collected process data. This section starts with a summary of some CPM techniques proposed in the literature. These techniques are extended and integrated to a comprehensive MPC performance assessment and monitoring methodology and diagnosis of types of causes for poor process performance [264]. Use of real-time KBSs for integrating CPM and diagnosis is also presented. Integration of CPM and diagnosis is illustrated by using an evaporator control case study. MPC calculations in this work are performed using a slightly modified version of the Matlab® MPC Toolbox [204] to allow for nonlinear plant models and a stepwise calculations necessary for on-line monitoring.

Model predictive control is based on real-time optimization of a cost function. Consequently, CPM methods that focus on the values of this cost function can be developed. The MPC cost function $\Phi(k)$ is

$$\begin{aligned}\Phi(k) &= \sum_{j=N_1}^{P} [\widehat{\mathbf{y}}(k+j) - \mathbf{r}(k+j)]^T \mathbf{Q} [\widehat{\mathbf{y}}(k+j) - \mathbf{r}(k+j)] \\ &+ \sum_{j=1}^{M} [\Delta \mathbf{u}(k+j-1)]^T \mathbf{R} [\Delta \mathbf{u}(k+j-1)] \end{aligned} \tag{9.16}$$

where $\mathbf{r}(k)$, $\widehat{\mathbf{y}}(k)$, and $\Delta \mathbf{u}(k)$ are vectors of reference trajectories, predicted

outputs, and change in manipulated variables at time k, respectively. $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices representing the relative importance of each controlled and manipulated variable. Control moves at each sampling time are obtained by calculating a control sequence that minimizes $\Phi(k)$. Therefore, it is reasonable to measure MPC performance by calculating values of $\Phi(k)$ using plant data. A performance measure based on $\Phi(k)$ can be defined as

$$J_{actual}(k) = \mathbf{e}^T(k)\mathbf{Q}\mathbf{e}(k) + \Delta\mathbf{u}^T(k)\mathbf{R}\Delta\mathbf{u}(k) \tag{9.17}$$

where $\mathbf{e}(k) = \widehat{\mathbf{y}}(k) - \mathbf{r}(k)$ is the vector of controlled variable errors and $\Delta\mathbf{u}(k)$ is the vector of control moves at time k. $\Phi(k)$ is a random variable because of measurement noise and disturbances. Consequently, the expected value of the cost function is more suitable for measuring the controller performance achieved:

$$J_{ach} = E[J_{actual}(k)] = E[\mathbf{e}^T(k)\mathbf{Q}\mathbf{e}(k) + \Delta\mathbf{u}^T(k)\mathbf{R}\Delta\mathbf{u}(k)] \tag{9.18}$$

Here $E[.]$ is the expectation operator and $\mathbf{e}(k)$ and $\Delta\mathbf{u}(k)$ are computed from the data set under examination. The *LQG benchmark* [115], the *historical performance benchmark* [222], and the *model-based performance benchmark* [222, 347] are some of the methods that have been proposed in the literature for CPM of MPC.

LQG-Benchmark The achievable performance of a linear system characterized by quadratic costs and Gaussian noise can be estimated by solving the linear quadratic Gaussian (LQG) problem. The solution can be plotted as a trade-off curve that displays the minimal achievable variance of the controlled variable versus the variance of the manipulated variable [115] which is used as a CPM benchmark. Operation close to optimal performance is indicated by an operating point near this trade-off curve. For multivariable control systems, H_2 norms are plotted. The LQG objective function and the corresponding H_2 norms are [115]

$$\Phi_{LQG}(\lambda) = E[\mathbf{e}(k)^T\mathbf{Q}\mathbf{e}(k)] + \lambda E[\Delta\mathbf{u}(k)^T\mathbf{R}\Delta\mathbf{u}(k)] \tag{9.19}$$

$$\|G_Y\|_Q^2 = E[\mathbf{e}(k)^T\mathbf{Q}\mathbf{e}(k)] \qquad \|G_u\|_R^2 = E[\Delta\mathbf{u}(k)^T\mathbf{R}\Delta\mathbf{u}(k)] \tag{9.20}$$

The trade-off curve is obtained by calculating the H_2 norms for different values of λ and plotting $\|G_Y\|_Q^2$ versus $\|G_u\|_R^2$. Once the trade-off curve is calculated, the H_2 norms under the existing control system are computed and compared to the optimal control represented by the trade-off curve.

The LQG benchmark is limited to a special group of MPCs characterized by the equality of control (M) and prediction (P) horizons and lack of feedforward components and constraints. It may be considered as a limit of achievable performance in terms of input and output variance to evaluate

various types of controllers. Since M and P are two independent and important tuning parameters and incorporation of constraints and feedforward control are important advantages of MPC over conventional controllers, alternatives to the LQG benchmark have been developed for monitoring the performance of these more interesting MPC implementations.

Historical Benchmark *A priori* knowledge that the performance was good during a certain time period is necessary to use this approach [222]. For the block of input and output data of this period, the historical benchmark J_{hist} is given by an equation of the same form as Eq. 9.18 where $\mathbf{e}(k)$ and $\Delta\mathbf{u}(k)$ are taken from the historical data set. The objective function for the performance achieved (J_{ach}) is calculated by using again Eq. 9.18 where $\mathbf{e}(k)$ and $\Delta\mathbf{u}(k)$ are taken from data collected during the period of interest. The performance measure is defined as the ratio

$$\gamma_{hist} = \frac{J_{hist}}{J_{ach}} \tag{9.21}$$

Model-based Performance Measure Two alternatives that rely on a process model, the design case and the expected performance, have been proposed:

Design Case Approach. Patwardhan *et al.* [222] have suggested the comparison of the achieved performance with the performance in the design case that is characterized by inputs and outputs given by the model. The design cost function J_{des} has the same form as Eq. 9.18 where $\mathbf{e}^*(k)$ and $\Delta\mathbf{u}(k)^*$ are substituted for $\mathbf{e}(k)$ and $\Delta\mathbf{u}(k)$ to indicate the predicted deviations of model outputs from the set-points (an estimate of the disturbance is included) and the optimal control moves, respectively. J_{ach} is the same as that in historical benchmark Eq. 9.18 and is calculated using plant data. Performance variation between the real plant (J_{ach}) and model (J_{des}) is expressed by

$$\gamma_{des} = \frac{J_{des}}{J_{ach}} \tag{9.22}$$

Expected Performance Approach. Zhang and Henson [347] have proposed an on-line comparison between expected and actual process performance. The expected performance is obtained by implementing controller actions on the process model. The expected performance incorporates estimates of state noise, but no output disturbances. The actual and expected performance are compared on-line over a moving horizon P_C of past data using the ratio [347]:

$$I_{MPC}(k) = \frac{J_{exp}(k)}{J_{act}(k)} \tag{9.23}$$

The actual performance is defined as

$$J_{act}(k) = \sum_{j=1}^{P_C} \mathbf{e}^T(k+j-P_C)\mathbf{Q}\mathbf{e}(k+j-P_C) \tag{9.24}$$

The expected performance uses Eq. 9.24 as well, after replacing $\mathbf{e}$ with $\mathbf{e}^*$. The ratios γ_{des} and I_{MPC} are very similar. In general, they are smaller than 1 due to imperfect models, sensor noise, or other uncertainties.

I_{MPC} is a stochastic variable and statistically significant changes in the controller performance can be detected by statistical analysis. I_{MPC} is assumed to be generated by an ARMA model

$$A(q^{-1})I_{MPC}(k) = C(q^{-1})z(k) \tag{9.25}$$

where $C(q^{-1})$ and $A(q^{-1})$ are monic polynomials and $z(k)$ is a zero-mean, uncorrelated, Gaussian noise signal [347]. Polynomials A and C and the variance of z can be estimated from a sequence of I_{MPC} values computed by using data collected in a time interval in which the controller performs as expected. I_{MPC} is highly serially correlated and the AR part is first-order [347]:

$$(1 - a_1 q^{-1})I_{MPC}(k) = z(k) \tag{9.26}$$

Defining

$$\Delta I_{MPC}(k) \equiv \frac{\hat{A}(q^{-1})}{\hat{C}(q^{-1})} I_{MPC}(k) \tag{9.27}$$

where $\hat{C}(q^{-1})$ and $\hat{A}(q^{-1})$ are estimated polynomials, the estimated noise variance is used to compute 95% confidence intervals on $\Delta I_{MPC}(k)$ [347]. Violation of these control limits indicates a statistically significant change in controller performance. According to Eqs. 9.26 and 9.27, $\Delta I_{MPC}(k)$ is a prediction residual and should have a Normal distribution. Prediction residuals are used to monitor variations in autocorrelated random variables using well-established SPM charts.

A Comprehensive Technique for MPC Performance Monitoring

The essential step in the LQG benchmark is the calculation of various control laws for different values of λ and prediction (P) and control (M) horizons ($P = M$). This is a case study for a special type of MPC (unconstrained, no feedforward) and a special parameter set ($M = P$) to find the optimal value of the cost function and an optimal controller parameter set. Using the same information (plant and disturbance model, covariance matrices of noise and disturbances), studies can be conducted for any type of MPC and the influence of any parameter can be examined. These studies

Table 9.1. Categorization of techniques to be used (ff – feedforward).

Controller Specification	Assessment	Monitoring	Diagnosis
unconstrained, no ff	LQG	$\gamma_{hist}(k)$	$\gamma_{des}(k)$
unconstrained, ff	comparative study	$\gamma_{hist}(k)$	$\gamma_{des}(k)$
constrained, no ff	comparative study	$\gamma_{hist}(k)$	$\gamma_{des}(k)$
constrained, ff	comparative study	$\gamma_{hist}(k)$	$\gamma_{des}(k)$

can be automated and the corresponding value of the cost function can be reported as function of the underlying parameter set [264].

A value of the cost function suitable to be the historical benchmark and a design case that performs acceptably is selected. Two performance measures for on-line monitoring are defined after a benchmark is obtained. $\gamma_{hist}(k)$ is extended for computation at each sampling time to determine controller performance. $\gamma_{des}(k)$ is extended for computation at each sampling time to assist in diagnosis of types of causes for poor performance. CPM is implemented by using the LQG benchmark or a benchmark obtained from case studies and $\gamma_{hist}(k)$. When the controller performance is declared poor, $\gamma_{des}(k)$ is used to make diagnostic decisions.

Tools for controller performance assessment (CPA), CPM, and diagnosis are available for four types of MPCs by obtaining benchmarks for constrained cases and controllers including feedforward components, and establishing statistical analysis to the historical and model-based performance measures $\gamma_{hist}(k)$ and $\gamma_{des}(k)$ (Table 9.1).

The tuning parameters of MPC include P, M, and α that determines the desired speed of approach to the set-point by using a relationship between the set-points and the reference trajectory $\mathbf{r}(k+l) = \alpha \mathbf{s}_p(k+l-1) + (1-\alpha)\mathbf{s}_p(k+l)$. In addition, weight matrices and input constraints can be used to adjust the aggressiveness of the controller. The minimum achievable value of the cost function J can be found by varying M, P, and α if the weight matrices and constraints are fixed to specific values. For $P = M$ (LQG benchmark), the largest value of $P(= M)$ minimizes the cost function. However, $M = 2$ and $P = 20$ seems to be the optimum combination for the parameter ranges under examination for the evaporator control case study. The minimal value of J can be used as a benchmark. A quantitative measure of the performance is given by γ_{hist}. Systematic comparative studies may be computationally too intensive, especially if limits on control moves and weight matrices are considered. Therefore, one might want to select M and P first and then continue to seek the benchmark value by varying other parameters. The absolute optimum may be missed because of the interdependencies of parameters, but the trade-off is

significant reduction in the computational burden.

For on-line monitoring, γ_{hist} is computed at each sampling time. In analogy to the calculation of J_{act} [347], the achieved cost function (J_{ach}) is calculated over a moving horizon P_C of past data

$$J_{ach} = \frac{1}{P_C}\left[\sum_{j=1}^{P_C}(\mathbf{e}^T(k+j-P_C)\mathbf{Q}\mathbf{e}(k+j-P_C)) + \Delta\mathbf{u}^T(k+j-P_C)\mathbf{R}\Delta\mathbf{u}(k+j-P_C)\right] \quad (9.28)$$

where $\mathbf{e}(k)$ is the vector of control errors at time k. The performance measure $\gamma_{hist}(k)$ at sampling time k is

$$\gamma_{hist}(k) = \frac{J_{hist}}{J_{ach}(k)} \quad (9.29)$$

Since γ_{hist} is a random variable, SPM tools can be used to detect statistically significant changes. $\gamma_{hist}(k)$ is highly autocorrelated. Use of traditional SPM charts for autocorrelated variables may yield erroneous results. An alternative SPM method for autocorrelated data is based on the development of a time series model, generation of the residuals between the values predicted by the model and the measured values, and monitoring of the residuals [1]. The residuals should be approximately normally and independently distributed with zero-mean and constant-variance if the time series model provides an accurate description of process behavior. Therefore, popular univariate SPM charts (such as $\overline{x}$-chart, CUSUM, and EWMA charts) are applicable to the residuals. Residuals-based SPM is used to monitor $\gamma_{hist}(k)$. An AR model is used for representing $\gamma_{hist}(k)$:

$$A(q^{-1})\gamma_{hist}(k) = \epsilon(k) \quad (9.30)$$

where $A(q^{-1})$ is monic polynomial with $a_i, i = 1, \cdots, na$ and $\epsilon(k)$ is a zero-mean, uncorrelated, Gaussian noise signal. Equation 9.30 is used to estimate the value of $\hat{\gamma}_{hist}(t)$ at time k, $\gamma_{hist}(k)$. The residuals are

$$e_\gamma(k) = \gamma_{hist}(k) - \hat{\gamma}_{hist}(k) \quad (9.31)$$

The AR model and the variance of $e_\gamma(k)$ can be estimated from an 'in-control' data set using software such as Matlab® System Identification Toolbox [191]. A standard $\overline{x}$-chart is designed using control limits at ± 3 standard deviations (3σ limits) to monitor the residuals $e_\gamma(k)$ and consequently $\gamma_{hist}(t)$.

Table 9.2. Groups of root cause problems.

Group I	Group II
(a) change in controller specifications	change in process dynamics
(b) change in measured disturbances	change in unmeasured disturbance
(b) input saturation	change in noise covariance

The model-based performance measure γ_{des} is used in the proposed method as model-based performance measure after modifying the cost functions for on-line monitoring. $J_{des}(k)$ and $J_{ach}(k)$ are computed using Eq. 9.28 with $\mathbf{e}^*$ and $\mathbf{e}$, respectively.

$$\gamma_{des}(k) = \frac{J_{des}(k)}{J_{ach}(k)} \tag{9.32}$$

Statistical monitoring similar to that for $\gamma_{hist}(k)$ is developed to detect significant changes over time.

Diagnosis

γ_{des} is monitored for diagnosing the causes of performance degradation. Some root causes affect the design case controller while others do not. For instance, increases in unmeasured disturbances, actuator faults, or increase in the model mismatch do not influence the design case performance. Accordingly, J_{des} remains constant while J_{ach} increases, reducing the model-based performance measure. Root cause problems such as input saturation or increase in measured disturbance, on the other hand, affect the design case performance as well. This leads to an approximately constant value of the model-based performance measure, if the effect is quantitatively equal (which happens for a good process model). The three techniques introduced can be classified according to the type of controller and the indexes used for CPA/CPM and diagnosis activities (Table 9.1).

When degradation in performance is indicated, diagnosis can be performed by inspecting $\gamma_{des}(k)$. Assuming that only one source cause occurs, if $\gamma_{des}(k)$ has not changed significantly, the reason for the overall degradation does affect both the design and achieved performance cost function to the same extent. Thus, the cause belongs to Group I (Table 9.2). If the model-based performance measure shows a degradation as well, the cause belongs to group II. If multiple causes can occur simultaneously, then the diagnosis logic becomes more complex.

Subgroups are defined to further distinguish between the root cause problems in Group I. All changes in the controller (e.g., tuning parameters, estimator, constraints) are assumed to be performed manually, since

the action taken is known and the root cause of the effect does not need to be identified by diagnosis tools (Subgroup Ia). Changes in measured disturbances and input saturation make up subgroup Ib. Additional information is needed to distinguish between them. Input saturation can be determined by looking at manipulated variable trajectories. A saturation effect in a manipulated variable indicates input saturation as underlying root cause and rules out the increase in measured disturbances.

Discrimination between performance degradation due to increases in unmeasured disturbances and changes in process parameters is a question of model validation. Consider an idealized case where disturbances can be regarded as white noise. If the model is perfect, the innovation sequence is white noise as well [2]. Imperfect models change the color of the innovation sequence that can be detected using various methods.

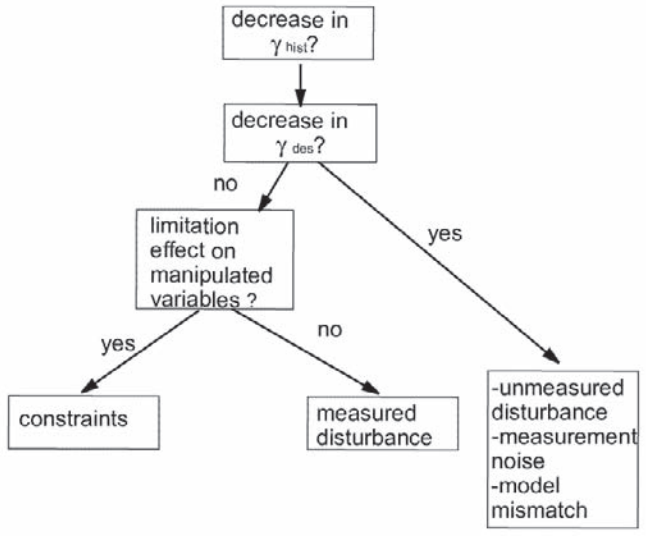


Figure 9.1. Diagnosis logistics.

If it is assumed that changes in controller specifications are done manually and do not need to be identified by the diagnosis tools, the sequence of detection and diagnosis follows the path in Figure 9.1. Performance is monitored over time using the performance measure based on γ_{hist}. Once a degradation is detected, γ_{des} is used to distinguish between root cause problems of Group I and Group II. Information about the trend of manipulated variables is used to distinguish between problems resulting from constraints and increases in measured disturbances.

Example A case study illustrates the application of CPM and diagnosis to MPC of a forced circulation evaporator using a detailed model [264]. First, a historical benchmark is found. Then, performance monitoring and diagnosis are performed simultaneously for two different cases differing by the use of linear and nonlinear plant models. The fundamental assumption of a known plant and disturbance model while assessing the initial performance is perfectly valid for the first case study and questionable for the second. The impact of linearity assumption and other effects resulting from nonlinearity are shown and discussed [264]. A forced circulation evaporator model is used. It is a linear state-space model in deviation variables obtained from linearization around normal operating conditions [215]. The system has three controlled variables (separator level (L_2), product composition (X_2), and operating pressure (P_2)); three manipulated variables (product flow rate (F_2), steam pressure (P_{100}), and cooling water flow rate (F_{200})); and five disturbances (circulation flow rate (F_3), feed flow rate (F_1), feed composition (X_1), feed temperature (T_1), and cooling water inlet temperature (T_{200})). Two cases are summarized to display the performance of the integrated CPM and diagnosis method presented. Details are provided elsewhere [264].

Decrease of the Saturation Limit. The saturation limit of P_{100} is set to zero at $k = 300\ min$. γ_{hist} indicates a performance degradation (Figure 9.2). A linear plant simulation model and a linear MPC model are used. Because γ_{des} does not decrease, the source cause of the degradation belongs to Group I. To distinguish between an increase in measured disturbances, an increase in the measurement noise and an input saturation as the source cause, the trend of the manipulated variables is observed (Figure 9.3). The effect of input saturation can be seen clearly between $k = 300\ min$ and $k = 350\ min$. After $k = 350\ min$ the MPC being aware of this limit tries to stay at the operation point by rearranging the use of the manipulated variables. However, the input saturation is correctly identified to be the root cause problem.

Real-time Diagnosis with G2® – Increase in Measured Disturbance. G2® is a commercial knowledge-based system (KBS) development tool for building real-time KBS [88]. It can be used for developing supervisory KBS for building process models, monitoring and control systems, fault diagnosis algorithms, on-line operator interaction, and integration of these functions. Expert knowledge and reasoning can be represented by rules that can make inferences based on process data. Procedures containing a certain sequence of actions can be programmed, for instance, for automating checks on different variables. Communication between G2® and external systems is handled by the G2® Standard Interface (GSI) that provides the necessary

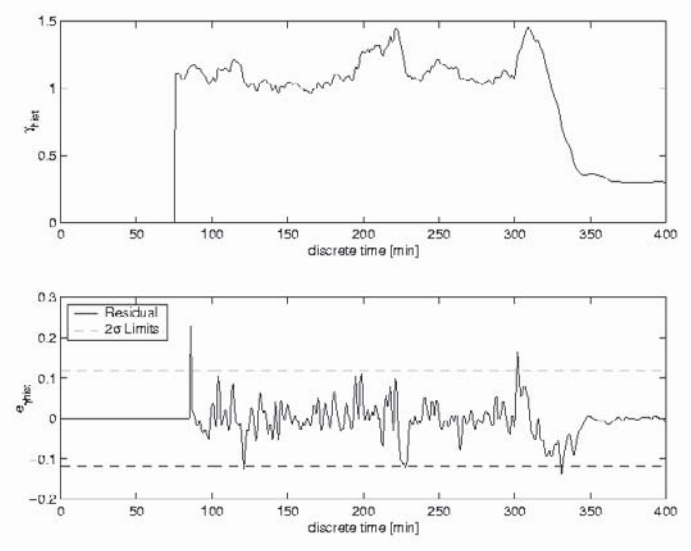


Figure 9.2. Effect of input saturation on γ_{hist}.

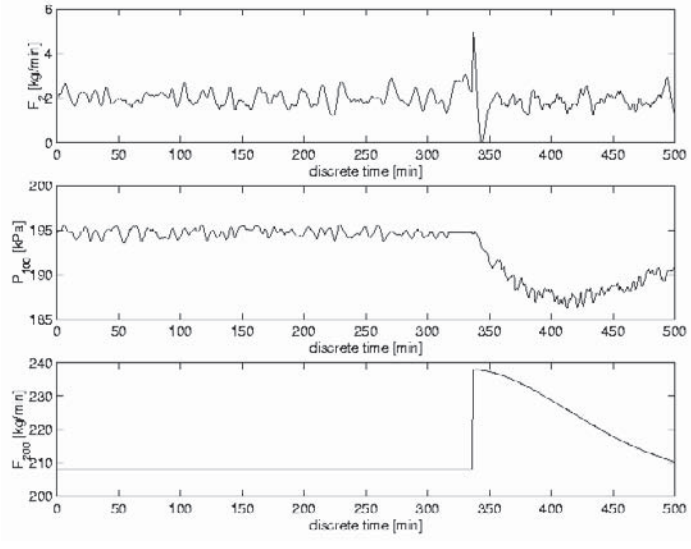


Figure 9.3. Effect of input saturation on the manipulated variables.

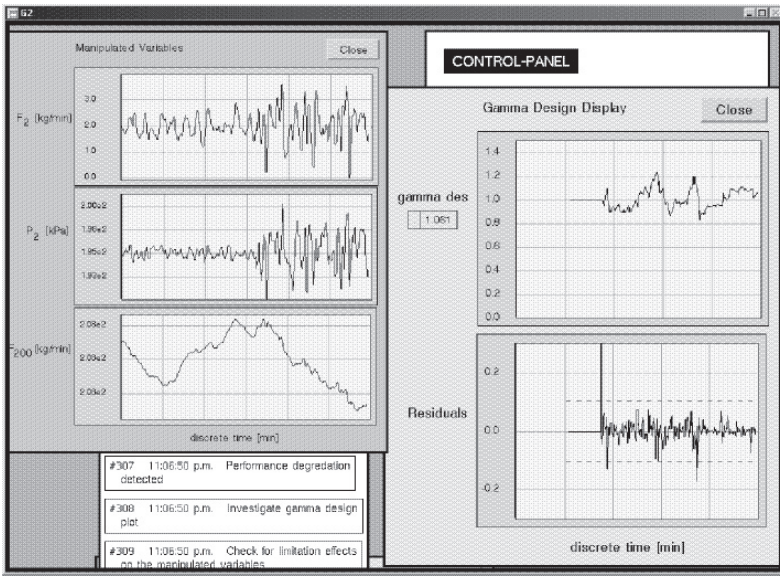


Figure 9.4. Snapshot of G2® screen – Increase in the measured disturbance.

network protocol information for communication between G2® and external functions written in C. In this work, software modules developed in Matlab® are converted to C with the Matlab® C compiler and linked to G2® [289]. The increase in measured disturbance is implemented on F_3 at $k = 300\ min$. The disturbance data sequence of this variable is increased by a factor 4. The increase in the measured disturbance causes performance degradation as indicated by γ_{hist}. Since γ_{des} does not decrease, the actual cause of degradation belongs to Group I. Trends in manipulated variables are observed to distinguish between the possible subgroups. A performance degradation due to constraints is ruled out since the manipulated variables are not saturated. The diagnosis logistics (Figure 9.1) are implemented as a rule base in G2® to support the operator. Figure 9.4 shows the G2® screens with the message box, the manipulated variable trajectories,and the CPM measures. The result of inferencing by G2® and the diagnostic results are displayed in the message box.

9.4 Summary

Controller performance monitoring (CPM) ensures proper performance of the control systems for safe and profitable operation of a process and man-

ufacture of products within specifications. CPM and control system diagnosis activities are a subset of the plantwide monitoring and diagnosis activities and they rely on the interpretation of process data. A comprehensive approach for assessing the effectiveness of control systems includes the determination of control system capability, development of statistics for monitoring its performance, and development of methods for diagnosing the underlying causes of changes in performance. Many new techniques and tools have been developed in recent years to enhance CPM. This chapter focused on CPM of single-loop, multivariable and model predictive control (MPC) systems. Diagnosis was limited to distinguishing between root cause problems associated with an MPC system and problems that are not caused by the controller. Monitoring of MPC performance and a case study based on MPC of an evaporator model and a supervisory knowledge-based system (KBS) is presented to illustrate the methodology. The extension of CPM to web and sheet processes is discussed in Section 10.3.

10

Web and Sheet Processes

Sheet forming processes measure performance data mostly through scanning sensors that traverse in the cross-direction (CD) as the sheet is formed in the machine direction (MD), thus creating a zigzag pattern of discrete data path. However, there are some rare applications that have the capability for full sheet measurement at each sampling time. Figure 10.1 shows the difference between the two and the resulting spatio-temporal form of process data. Representing the scanner generated data as a two-dimensional full matrix $\mathbf{Y}(n, k)$ is a practical approximation that greatly simplifies the necessary calculations for process performance tracking and evaluation. Nature of the process data $\mathbf{Y}(n, k)$ may be the thickness of the sheet, its moisture, basis weight (mass/area), brightness or any other pertinent measure of process performance or product value. The process itself may involve manufacturing of metal sheets, glass, plastic film, fabrics or pulp and paper sheets.

A unique characteristic of the process data for sheet forming processes is the presence of two independent variables, space n and time k. In most cases the target of the process is to maintain the uniformity of $\mathbf{Y}$ for all n and k. For some applications a predefined constant CD profile $\mathbf{y}_{CD}^{target}(n)$ may be the desired target. Therefore, the objective is to analyze the deviation of $\mathbf{Y}$ from its target and extract meaningful information from the results to be used for process control or performance evaluation and diagnostics. While the space variable n is well defined between the front and back ends of the cross-direction the time variable k is flexible in terms of its origin and end. Most sheet forming processes are continuous and thus k may be treated as an indefinite discrete variable. At the same time, for practical reasons, all sheets are cut to finite lengths for packaging and transportation or for post-processing. Therefore, the time index k may also be treated as a finite length temporal variable.

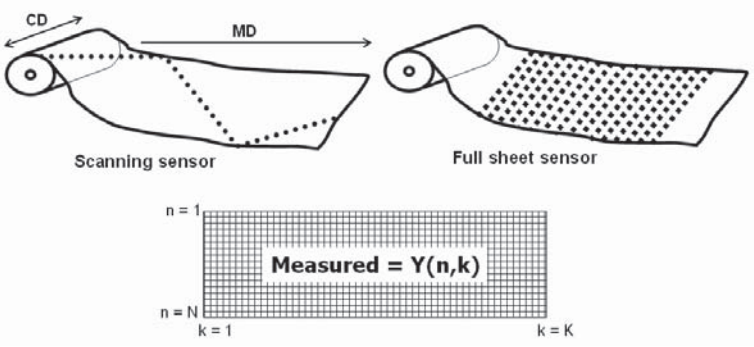


Figure 10.1. Sheet process data measurement.

10.1 Traditional Data Analysis

In terms of orientation to appreciate the traditional basics of sheet forming process data it is useful to briefly review the conventional steps of data analysis for a univariate system. Consider the set of hypothetical data shown in Figure 10.2. There are 40 data points with X as the independent variable and Y as the dependent variable. The means are X_m and Y_m respectively. Clearly there is a trend in Y as a function of X. Let the function $(Y-Y_m) = f(X-X_m)$ represent the trend in an optimal manner and be the correlation model for the segment of observed data. The residual of data when compared to the model is a measure of scatter or variability around the dominant trend. Close observation of the residual indicates that the variability in the range $X < X_m$ is smaller than the variability in the range $X > X_m$. This simple yet useful approach has three basic components: (1) establishing and removing the data means, (2) establishing the correlated trend in dependent variable, (3) analyzing the scatter around the correlation model as a function of the independent variable. Complete evaluation of data and quantifying its pertinence for process improvement may require further effort which may be extensive depending on the complexity of the problem. However, in general these are the three basic steps for initial analysis of process data that directly apply to sheet forming processes as well.

10.1.1 MD/CD Decomposition

As an example consider the basis weight measurement in paper manufacturing where the uniformity of sheet density, tracked as gm/m^2, must be closely maintained at a fixed target. Figure 10.3 is a three-dimensional rep-

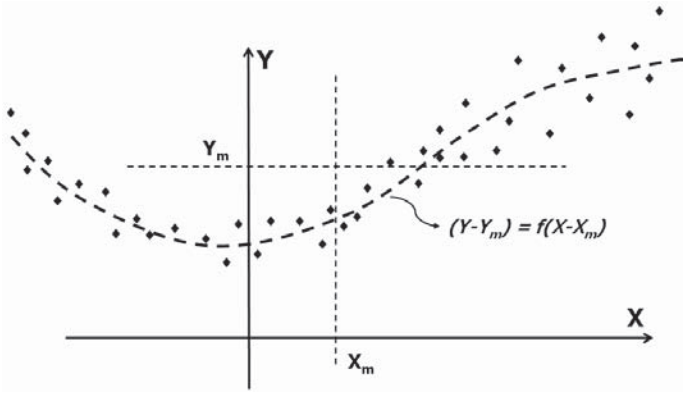


Figure 10.2. General approach to univariate function analysis.

resentation of a typical data set for a roll of sheet showing measurements at 54 cross-direction (CD) locations for 110 machine-direction (MD) scans, $N = 54$ and $K = 110$. Typically total CD distance may be about 5 m and the MD length may correspond to approximately 30 min of production. CD data length increments usually match the size of CD adjustment actuators while each data is the averaged scanner information for the corresponding distance that is also known as the data lane. For a typical application additional sheet properties like thickness and moisture are measured simultaneously. For the basis weight measurements in this example there are a total of $54 \times 110 = 5940$ data points represented in matrix form $\mathbf{Y}(n, k)$. For practical reasons, the original numbers of the data are turned into normalized deviation variables by subtracting the overall mean and then dividing by the standard deviation. For a typical paper roll the overall mean is very close to the target basis weight, so there is not much information loss by the use of deviation variables. However, the product value for the roll is easily degraded by the two-dimensional variability of $\mathbf{Y}(n, k)$ regardless of how close the target is satisfied by the overall average. Figure 10.4 is the histogram of the total data set showing typical similarity to a normal distribution. It is important to recognize that the histogram captures only the collective variability of the individual data points without any regard to trends and correlations that might exist with respect to specific (n, k) locations on the sheet.

MD/CD decomposition separates the two-dimensional means one at a time from the data matrix $\mathbf{Y}(n, k)$ that has N rows and K columns. First, the averages of all spatial locations for each scan are computed to get the MD trend as $\mathbf{y}_{MD}(k)$. Then, the CD profile is computed by subtract-

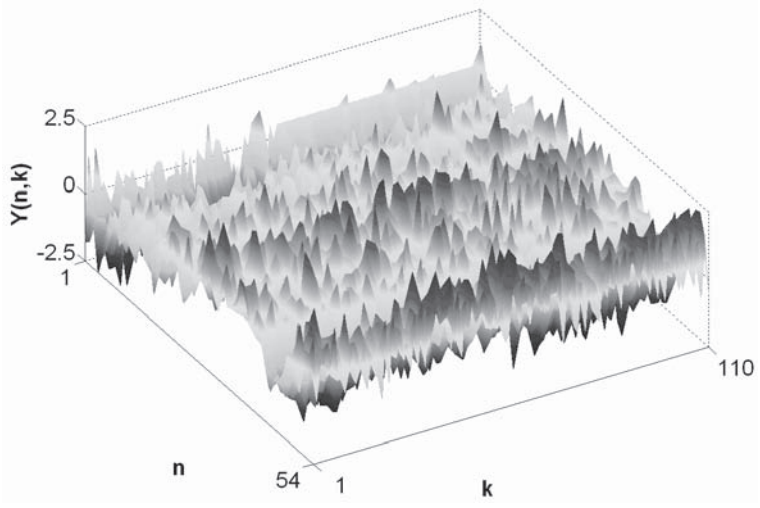


Figure 10.3. Normalized basis weight data for a roll of paper sheet.

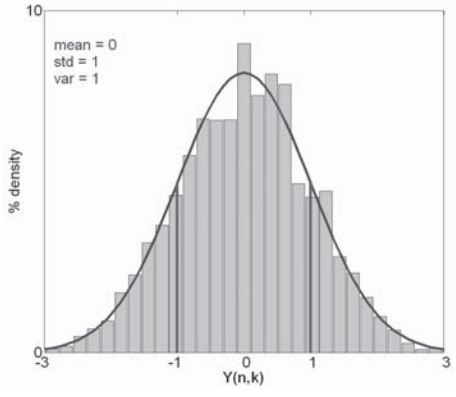


Figure 10.4. Histogram of normalized basis weight data with comparison to normal distribution.

ing $\mathbf{y}_{MD}(k)$ from each element of the corresponding column of $\mathbf{Y}(n,k)$ followed with row-by-row averaging to get $\mathbf{y}_{CD}(n)$. The MD trend is an N-dimensional row vector while the CD profile is a K-dimensional column vector. It is useful to construct corresponding set of matrices $\mathbf{Y}_{MD}(n,k)$ and $\mathbf{Y}_{CD}(n,k)$ where the vectors $\mathbf{y}_{MD}(k)$ and $\mathbf{y}_{CD}(n)$ are repeated to fill

in the NxK dimensions. Data residual is then defined as

$$\mathbf{Y}_R(n,k) = \mathbf{Y}(n,k) - \mathbf{Y}_{MD}(n,k) - \mathbf{Y}_{CD}(n,k) \tag{10.1}$$

or simply $\mathbf{Y}_R = \mathbf{Y} - \mathbf{Y}_{MD} - \mathbf{Y}_{CD}$.

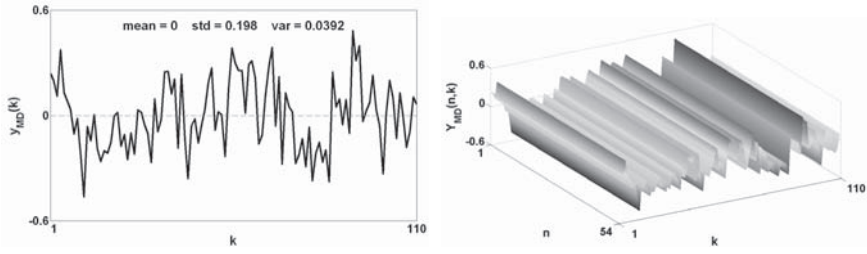


Figure 10.5. Vector and matrix forms of MD trend.

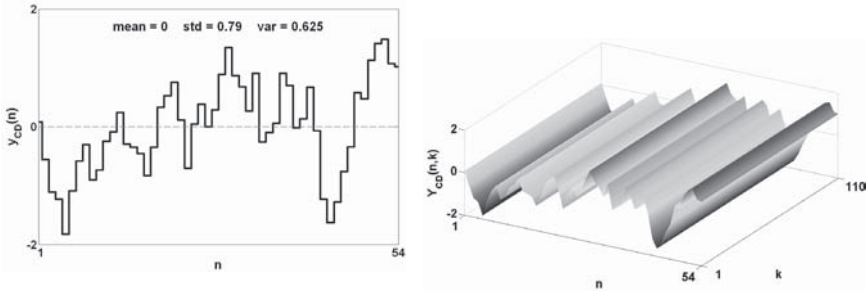


Figure 10.6. Vector and matrix forms of CD profile.

Both vector and matrix forms of MD trend and CD profiles are shown in Figures 10.5 and 10.6. MD/CD decomposition removes the most dominant trends in data through simple averaging along each dimension. MD trending is uniquely defined as it applies exclusively to each time increment. On the other hand CD profile calculation is specifically dependent on the time 'window' or the number of scans used for averaging, in this case $K = 110$. Figure 10.7 shows that the remainder $\mathbf{Y}_R$ is more random than the original data $\mathbf{Y}$ as should be expected.

MD/CD decomposition is a sequence of two practically independent averaging or zero-order filtering operations with resulting variances that are

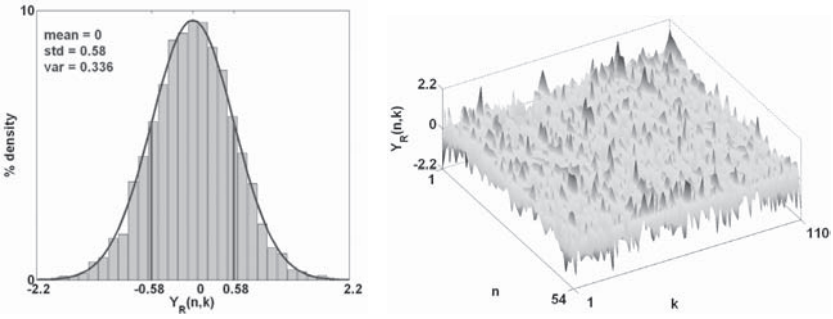


Figure 10.7. Residual of data after removing MD trend and CD profile.

essentially additive, $\sigma_Y^2 = \sigma_{Y_{MD}}^2 + \sigma_{Y_{CD}}^2 + \sigma_{Y_R}^2$. For process monitoring all three variance components $\sigma_{Y_{MD}}^2$, $\sigma_{Y_{CD}}^2$ and $\sigma_{Y_R}^2$ or equivalently the corresponding standard deviations are tracked. Reduction in variability may come from improvements in: (a) process equipment, (b) operating practices including the elimination of disturbances, and (c) automatic control. According to the normalized variances as displayed in the Figures 10.5–10.7 major contributors to process variability for this example are $\mathbf{Y}_{CD}$ and $\mathbf{Y}_R$ implying the presence of effective basis weight automatic control for $\mathbf{y}_{MD}(k)$ but no active feedback correction for CD profile.

10.1.2 Time Dependent Structure of Profile Data

Analysis of full sheet data is useful for process performance evaluations and product value calculations. For feedback control or any other on-line application, it is necessary to continuously convert scanner data into a useful form. Consider the data vector $\mathbf{Y}(:,k)$ for scan number k. It is separated into its MD and CD components as $\mathbf{Y}(:,k) = \mathbf{y}_{MD}(k) + \mathbf{Y}_{CD}(:,k)$ where $\mathbf{y}_{MD}(k)$ is the mean of $\mathbf{Y}(:,k)$ as a scalar and $\mathbf{Y}_{CD}(:,k)$ is the instantaneous CD profile vector. MD and CD controllers correspondingly use these calculated measurements as feedback data for discrete time k. Univariate MD controllers are traditional in nature with only measurement delay as a potential design concern. On the other hand, CD controllers are multivariate in form and must address the challenges of controller design for large dimensional correlated systems.

Control systems ignore short term variabilities through appropriately designed filters. Effective length of the filter window determines how quickly significant variations are actually detected. Defining a CD profile vector $\mathbf{y}_{CD}(n)$ for a complete roll is perhaps the simplest form of a large window

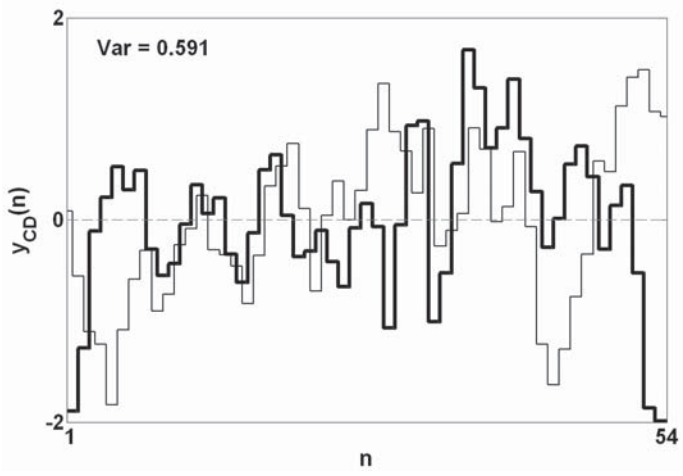


Figure 10.8. Comparison of full roll CD profiles. Light line is a repeat of Fig. 10.6 and heavy line is for another roll produced about 24 hours earlier.

or long time-span filter. Although $\mathbf{y}_{CD}(n)$ effectively and very efficiently captures the gross nature of CD variability it does not imply that the same profile vector is a quasi steady-state property of the sheet. Depending on process conditions and raw material variations CD profile usually changes with time. For the example discussed in Section 10.1.1, a comparison of its CD profiles with a roll manufactured approximately 24 hours earlier is shown in Figure 10.8. In contrast, an example of typical changes in the CD profile within the span of the same roll can be tracked through consecutive four shorter-window averages sequentially displayed in Figure 10.9. The 110 scans, shown as a single roll average in Figure 10.6, are divided into approximately four equal segments to capture and demonstrate the short-term time dependence of the CD profile.

10.2 Orthogonal Decomposition of Profile Data

In Section 10.1, it was stated that the basic steps of data evaluation are (a) removing of mean, (b) correlating the dominant trend, and (c) analyzing the residual scatter around the correlation. For the two-dimensional sheet process data MD/CD decomposition is essentially the implementation of step (a) in both the spatial and temporal modes. The resulting data com-

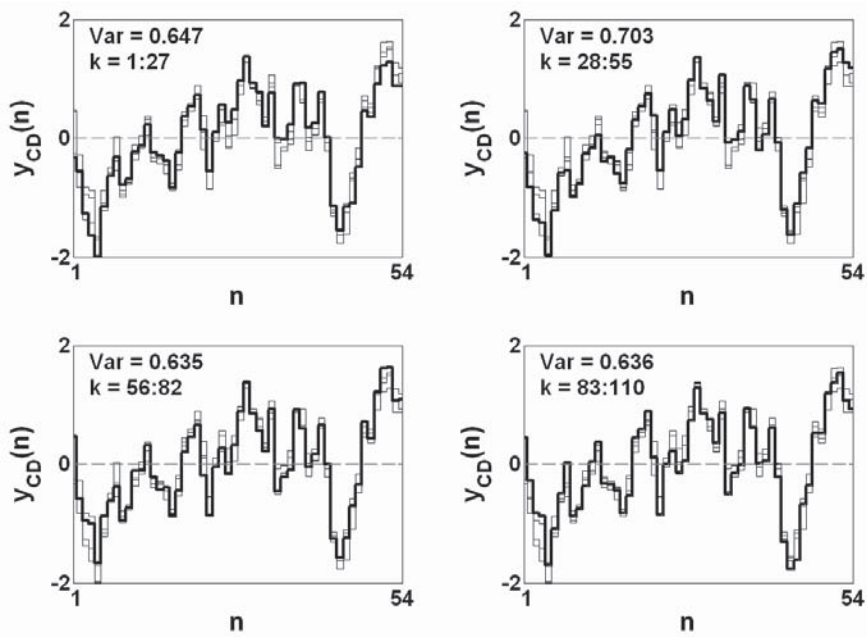


Figure 10.9. Sequence of changes in short-term CD profile through the length of a roll.

ponents $\mathbf{y}_{MD}$, $\mathbf{y}_{CD}$ and $\mathbf{Y}_R$ maintain significant information about process performance and improvement potentials, which can be evaluated through identification and analysis of dominant trends as suggested in step (b).

Process data can be correlated or de-trended using a variety of functions. A computationally reliable approach is the use of orthogonal functions. Least squares fit of data with a simple reduced order function can provide valuable information about process performance in terms of dominant contributions to variability.

Let $\mathbf{y} = [y_1; y_2; ...; y_N] = [y_1...y_N]^T$ be an N-dimensional mean-centered observation vector where $\bar{\mathbf{y}} = 0$. Let $\mathbf{z}_1...\mathbf{z}_M$ be M orthogonal basis functions where $M < N$, each basis vector $\mathbf{z}_m = [z_{m,1}...z_{m,N}]^T$ is N-dimensional, and $\mathbf{z}_i^T\mathbf{z}_j = 0$ for $i \neq j$. Define the $N \times M$ dimensional bases matrix $\mathbf{\Phi} = [\mathbf{z}_1\mathbf{z}_2...\mathbf{z}_M]$ through which $\mathbf{y}$ can be approximated as $\mathbf{y} \approx \mathbf{\Phi c}$, where $\mathbf{c} = [c_1...c_M]^T$ is the score vector measuring the projection magnitudes of $\mathbf{y}$ onto the lower dimensional bases $\mathbf{z}_1...\mathbf{z}_M$. Least squares approximation of $\mathbf{c}$ is obtained by $\mathbf{c} = \mathbf{\Psi y}$ where $\mathbf{\Psi} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T$ is called the transition matrix . Now $\mathbf{y}$ can be expressed as $\mathbf{y} = \mathbf{y}_M + \mathbf{y}_R$

where $\mathbf{y}_M = \mathbf{\Phi c}$ is the M-dimensional low-order approximation of $\mathbf{y}$ and $\mathbf{y}_R$ is the residual. Due to the orthogonal nature of the decomposition $\mathbf{y}_M$ and $\mathbf{y}_R$ are independent and therefore $\sigma_y^2 = \sigma_{y_M}^2 + \sigma_{y_R}^2$.

Separating measured data vectors or matrices into independent lower order approximations and residual terms is useful both in process performance evaluation, as variance contributions can be clearly separated, and in feedback process control, as the number of decision variables can be significantly reduced while the adverse effects of autocorrelation are eliminated. In the following two sections orthogonal decomposition approaches using Gram polynomials and principal components analysis (PCA) will be introduced.

10.2.1 Gram Polynomials

Gram polynomials are orthogonal and defined uniquely for discrete data at equidistant positions much like the spatial data collected in sheet forming processes. For N data positions, discrete-point scalar components of the mth-order polynomial vector $\mathbf{p}_m = [p_{m,1}...p_{m,n}...p_{m,N}]^T$ are defined as

$$p_{m,n} = \sum_{j=0}^{m} \frac{(-1)^j (m+j)^{2j}}{(j!)^2} \left(\frac{n}{N}\right)^j \tag{10.2}$$

$$\begin{aligned} p_{0,n} &= 1 \\ p_{1,n} &= 1 - 2(\frac{n-1}{N-1}) \\ p_{m,n} &= \frac{(N-1)(2m-1)}{m(N-m)} \left[1 - \frac{2(n-1)}{N-1}\right] p_{(m-1),n} \\ &- \frac{(m-1)(N-1+m)}{m(N-m)} p_{(m-2),n} \end{aligned} \tag{10.3}$$

Recursive formulations with respect to both polynomial order and data position are given in Eq. 10.3. Zero-order is only used to account for the data mean if needed. For a mean-centered data vector, the effective polynomials are $\mathbf{p}_1$ through $\mathbf{p}_{N-1}$. First five of these are plotted in Figure 10.10 for $N = 50$. Note that the polynomials are explicitly defined through the data length.

Example Consider the CD profile examined in Figure 10.6. The measurement vector $\mathbf{y}_{CD}$ has $N = 54$ data positions with the corresponding Gram polynomials $\mathbf{p}_1$ through $\mathbf{p}_{53}$ that form $\mathbf{\Phi} = [\mathbf{p}_1\mathbf{p}_2...\mathbf{p}_{53}]$ and $\mathbf{\Psi} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T$ as defined earlier. Computationally $\mathbf{\Psi}$ can be easily

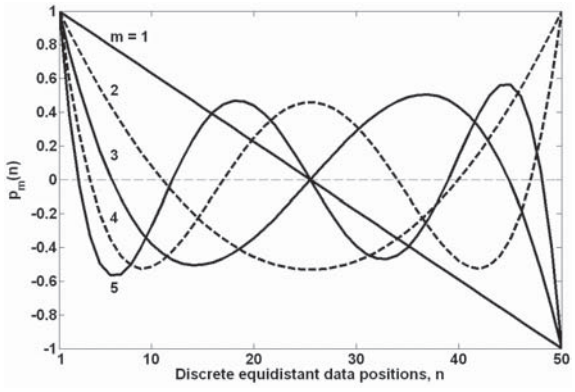


Figure 10.10. Gram polynomials as basis functions, first-order through fifth-order.

constructed row-by-row owing to the simplification arising from orthogonality, where the mth row is $\mathbf{\Psi}_m = \mathbf{p}_m^T/(\mathbf{p}_m^T\mathbf{p}_m)$, which is the normalized form of the corresponding polynomial. Projection magnitudes of $\mathbf{y}_{CD}$ on the Gram polynomial basis vectors are $\mathbf{c} = \mathbf{\Psi}\mathbf{y}_{CD}$. Although all 53 components of $\mathbf{c}$ are needed to duplicate $\mathbf{y}_{CD}$ only a few of the low-order coefficients may be sufficient to capture the dominant trends in the mean profile. Consider the full representation as $\mathbf{y}_{CD} = \mathbf{\Phi}\mathbf{c}$ and its partitioned form $\mathbf{y}_{CD} = \mathbf{\Phi}[\mathbf{c}_M; \mathbf{c}_{(N-1-M)}]$ leading to $\mathbf{y}_{CD} = \mathbf{\Phi}_M\mathbf{c}_M + \mathbf{\Phi}_R\mathbf{c}_R$ where the subscript M is used to designate the selection of the first M polynomial orders and R to designate the residual. Dimensions of $\mathbf{\Phi}_M$ and $\mathbf{c}_M$ are $N \times M$ and $M \times 1$ respectively.

Orthogonal decomposition of the CD profile $\mathbf{y}_{CD} = \mathbf{y}_{CD(M)} + \mathbf{y}_{CD(R)}$ using $M = 4$ and the corresponding variance contributions $\sigma^2_{y_{CD}} = \sigma^2_{y_{CD(M)}} + \sigma^2_{y_{CD(R)}}$ reveal that approximately 50% of the variability can be attributed to a low-frequency wave captured by 4th-order Gram polynomials. Figure 10.11 shows that the low-order approximation is in fact very effective and that the residual profile is more balanced compared to the original CD profile. Cumulative contributions of individual polynomial orders towards total variance are plotted in Figure 10.12. For this particular case even the first-order polynomial, which is simply a straight line, accounts for more than 20% of the variance. As it is visible in Figure 10.11, the CD profile has a distinct slant increasing from left (front of machine) to right (back), which is captured by $\mathbf{p}_1$. Also plotted in Figure 10.12 are the cumulative power spectra of $\mathbf{y}_{CD}$ and $\mathbf{y}_{CD(R)}$ as functions of frequency based on incremental measurement length. It is confirmed again that the fourth-order

Gram polynomial approximation has essentially removed all low-frequency variations from the profile.

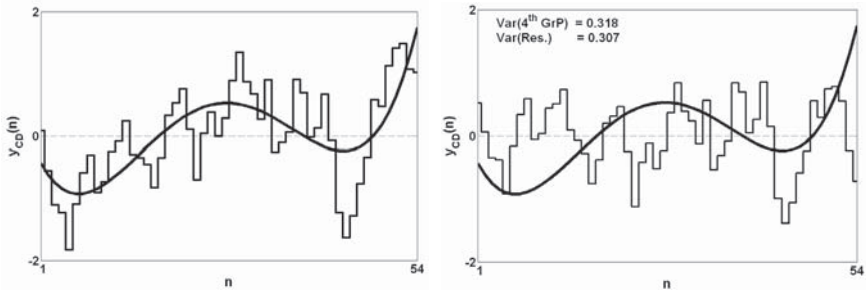


Figure 10.11. Fourth-order Gram polynomial approximation of mean CD profile and comparison with the residual measurement signal, $\mathbf{y}_{CD(R)} = \mathbf{y}_{CD} - \mathbf{y}_{CD(M)}$.

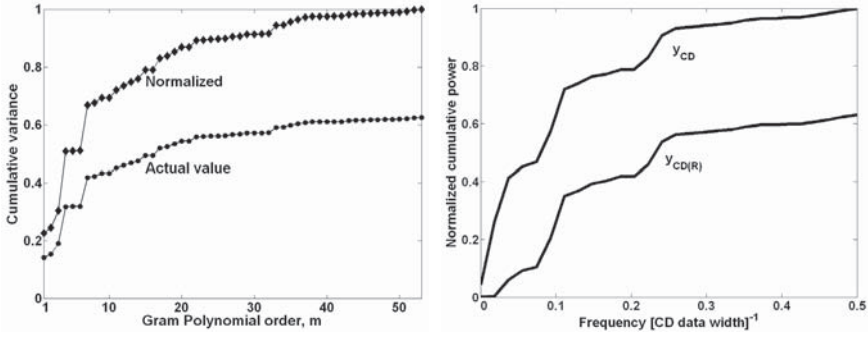


Figure 10.12. Accounting for total CD profile variability through Gram polynomial approximations and the cumulative power spectra for $\mathbf{y}_{CD}$ and $\mathbf{y}_{CD(R)}$.

A reasonable observation after these results may be that an effective CD controller should be able to reduce CD variability by at least a factor of 2. Another way of stating the same observation from a performance monitoring point of view would be that as long as a CD controller is functioning properly both plots in Figure 10.12 should indicate insignificant contributions from Gram polynomials up to order 4 or 5, which would also mean essentially similar power spectra for $\mathbf{y}_{CD}$ and $\mathbf{y}_{CD(R)}$.

10.2.2 Principal Components Analysis

Principal components analysis (PCA) (see Section 3.1) provides a technique to define orthogonal basis functions that are directly constructed from process data, unlike Gram polynomials which are dependent on the data length only. PCA is also uniquely suitable for extracting the dominant features of two-dimensional data like the residual profile obtained after MD/CD decomposition, $\mathbf{Y}_R$.

Let $\mathbf{Y}$ be $N \times K$ dimensional data ($N < K$) with mean-centered rows and columns as it is for $\mathbf{Y}_R$. Although the only requirement for PCA application is mean-centering of columns, having the rows mean-centered as well due to CD profile removal provides better scaling to remaining data. Define a covariance or scatter matrix $\mathbf{Z} = \mathbf{Y}\mathbf{Y}^T$ and let $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2...\mathbf{u}_N]$ with $\mathbf{u}_i = [u_{i,1}u_{i,2}...u_{i,N}]^T$ be the orthonormal eigenvectors of $\mathbf{Z}$ such that $\mathbf{ZU} = \mathbf{U\Lambda}$. As $\mathbf{Z}$ is symmetric and $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_N$ it follows that $\mathbf{Z} = \mathbf{U\Lambda U}^T$. Both $\mathbf{U}$ and $\mathbf{\Lambda}$ are easily computed through singular value decomposition (SVD). $\mathbf{\Lambda}$ is the diagonal eigenvalue matrix containing elements $\lambda_1...\lambda_N$ that are sequenced in descending order $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$. The basis matrix $\mathbf{U}$ is optimal in the sense that the largest contribution to data variance is captured through the first eigenvector, and of the residual the largest contribution to variance is then captured by the second eigenvector, and so on. Projection of original data $\mathbf{Y}$ onto the new basis vectors is calculated by $\mathbf{A} = \mathbf{U}^T\mathbf{Y}$ and equivalently the data are represented through the eigenvector bases as $\mathbf{Y} = \mathbf{UA}$. Corresponding to their directional roles $N \times N$ matrix $\mathbf{U}$ and $N \times K$ matrix $\mathbf{A}$ are referred to as spatial modes and temporal scores respectively. Both $\mathbf{A}$ and the eigenvalue matrix $\mathbf{\Lambda}$ provide a measure of data variability along each eigenvector, $\mathbf{AA}^T = \mathbf{U}^T\mathbf{Y}\mathbf{Y}^T\mathbf{U} = \mathbf{U}^T\mathbf{ZU} = \mathbf{\Lambda}$. For eigenvector $\mathbf{u}_i$ the corresponding variance contribution of data is $\sigma_i^2 = (\mathbf{a}_i\mathbf{a}_i^T)/(K-1)$ where $(\mathbf{a}_i\mathbf{a}_i^T) = \lambda_i$ and $\mathbf{a}_i$ is the ith row-vector of $\mathbf{A}$. Similar to λ_i variance contributions are also in descending order $\sigma_1^2 \geq \sigma_2^2 \geq ... \geq \sigma_N^2$.

Dominant correlations of data are usually captured by a small number of initial eigenvectors. A simple orthogonal decomposition is accomplished by partitioning $\mathbf{U} = [\mathbf{U}_M\mathbf{U}_R]$ and $\mathbf{A} = [\mathbf{A}_M; \mathbf{A}_R]$ where M designates the number of initial dominant modes to be used for approximation while R stands for the remaining $(N - M)$ modes or the residual. Data matrix becomes $\mathbf{Y} = \mathbf{U}_M\mathbf{A}_M + \mathbf{U}_R\mathbf{A}_R = \mathbf{Y}_M + \mathbf{Y}_R$. For a successful approximation, $\mathbf{Y}_M$ captures significant variability trends and $\mathbf{Y}_R$ simply represents residual random noise. Transformation in the form $\mathbf{Y} \approx \mathbf{Y}_M$ uses $M(N + K)$ data entries and provides $[1 - M(N + K)/(NK)]100$ % data compression.

For the paper machine data considered earlier, the PCA analysis of the residual matrix $\mathbf{Y}_R$ generates eigenvalues that are plotted in Figure 10.13.

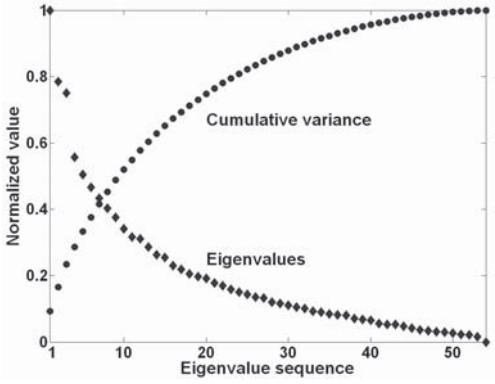


Figure 10.13. Normalized eigenvalues of $\mathbf{Y}_R$ and the cumulative contributions of modes towards total variance.

For practical purposes, the eigenvalues are normalized with respect to λ_1. Last eigenvalue is zero as the rank of the scatter matrix $\mathbf{Z}$ is $N-1$ due to mean centering of $\mathbf{Y}_R$ rows with CD profile removal. Variance contributions of each mode associated with the eigenvalues are also plotted. There are various methods of choosing the number of modes to be used for PCA approximation. One common method is to select the point of transition where eigenvalues start decreasing gradually after the initial faster drop. For this case it happens at approximately $M = 4$. The choice of M is not absolute as $M = 6$ would have provided similar results; however, smaller M is always preferred for parsimony unless there is a practical reason to choose a larger M. Figure 10.14 shows a few examples of the eigenvectors and the temporal scores for $\mathbf{Y}_R$. First two eigenvectors are capturing dominant slant and parabolic wave behaviors while one of the latter eigenvectors (45th) is practically high frequency noise. Eigenvectors are constructed to have normalized magnitude while their contributions for $\mathbf{Y}_R$ reconstruction for each temporal position are captured through associated score vectors. Correspondingly, magnitudes of first two scores are much higher than the 45th. For process control and monitoring, the goal is to have all eigenvectors and scores resemble the high-frequency nature of the 45th mode whether the PCA analysis is done on $\mathbf{Y}$ or $\mathbf{Y}_R$.

PCA approximation of $\mathbf{Y}_R$ with $M = 4$ gives $\mathbf{Y}_R = \mathbf{Y}_{RM} + \mathbf{Y}_{RR}$ where the last matrix is the residual profile recreated through $N - M$ modes. Combining $\mathbf{Y}_{RM}$ with the averaging results of MD/CD decomposition generates an overall filtered approximation for the full sheet profile as $\mathbf{Y} = \mathbf{Y}_{MD} + \mathbf{Y}_{CD} + \mathbf{Y}_{RM} + \mathbf{Y}_{RR} = \mathbf{Y}_M + \mathbf{Y}_{RR}$. Last two profiles are shown

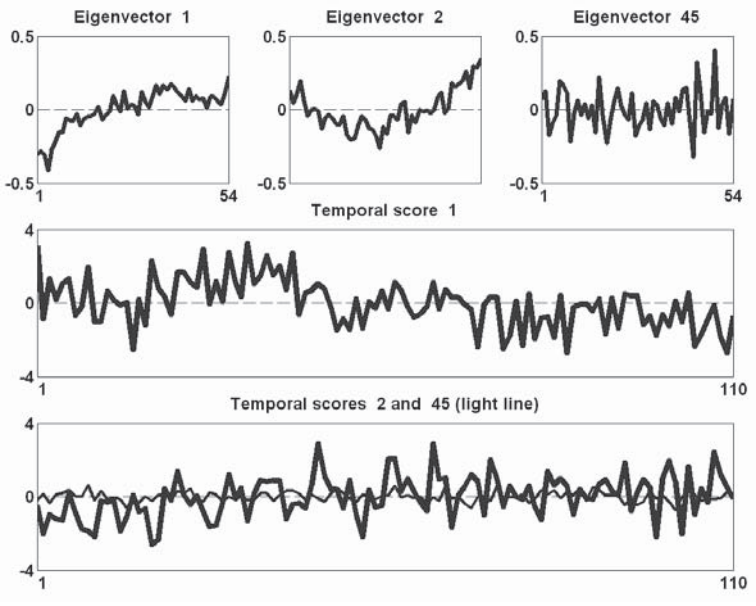


Figure 10.14. First two and the 45th eigenvectors and scores of $\mathbf{Y}_R$.

in Figure 10.15. $\mathbf{Y}_{RR}$ retains only high frequency random contributions in $\mathbf{Y}$ with $S^2_{RR} = 0.24$ while the filtered profile retains the rest $S^2_M = 0.76$. Process control and monitoring target would be to minimize overall process variability with the majority of variance captured in $\mathbf{Y}_{RR}$.

An alternative orthogonal decomposition of two-dimensional profile data is denoising through wavelet transforms as discussed in Section 6.2.2. To demonstrate, hard-thresholding with the wavelet function db8, as defined in Eq. 6.22, is used on the profile $\mathbf{Y}_R$ to approximate the full profile as $\mathbf{Y}_W \approx \mathbf{Y}_{MD} + \mathbf{Y}_{CD} + \mathbf{Y}_{RW}$. Figure 10.16 shows the results of denoising carried out using two levels of decomposition. Corresponding variances are $S^2_{W1} = 0.78$ and $S^2_{W2} = 0.71$ compared to the PCA result at $S^2_M = 0.76$.

10.2.3 Flatness of Scanner Data

Performance objective for sheet forming processes is to maintain uniformity or flatness in $(\mathbf{Y} - \mathbf{Y}^{target}_{CD})$. The target matrix represents $\mathbf{y}^{target}_{CD}$ for those applications where the CD set-point is not uniform, like in metal and plastic sheets that may have a 'frown' thickness target almost parabolic gradually increasing front and back ends towards the middle with relatively uniform mid section. Given the full matrix form of sheet data $\mathbf{Y}$, in

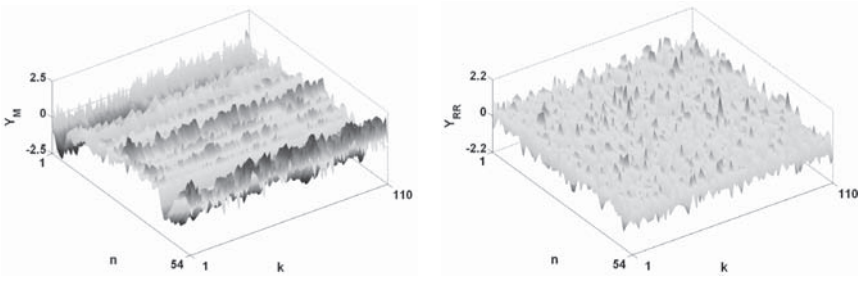


Figure 10.15. 4th-order PCA approximation of $\mathbf{Y}$ and the residual, $\mathbf{Y} = \mathbf{Y}_M + \mathbf{Y}_{RR}$.

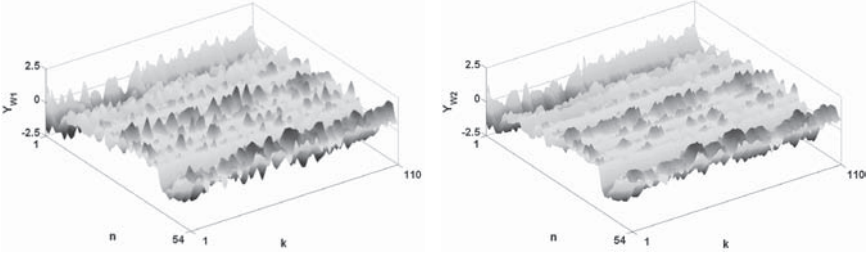


Figure 10.16. Filtered approximation of $\mathbf{Y}$ through wavelet denoising using hard-thresholding with one level of decomposition (left) and two levels of decomposition.

deviation form if the CD target is not uniform, MD/CD decomposition $\mathbf{Y} = \mathbf{Y}_{MD} + \mathbf{Y}_{CD} + \mathbf{Y}_R$ clearly indicates the importance of maintaining constant average behavior, $\mathbf{y}_{MD} = 0$ and $\mathbf{y}_{CD} = 0$, in keeping $\mathbf{Y}$ on target. $\mathbf{Y}_R$ contains scan-by-scan residual of data that measures point-wise deviations with respect to $\mathbf{y}_{MD}$ trend and $\mathbf{y}_{CD}$ profile. Each column vector of $\mathbf{Y}_R$ has deviation data that originate from local variabilities that are either random or structured in MD and/or CD directions. Due to the traversing nature of the scanner it is not possible to differentiate the origin of locally structured variations between MD and CD. Regardless of this limitation it is still informative to explore and quantify the local data trends in $\mathbf{Y}_R$ in

order to expose the potential margin of improvement.

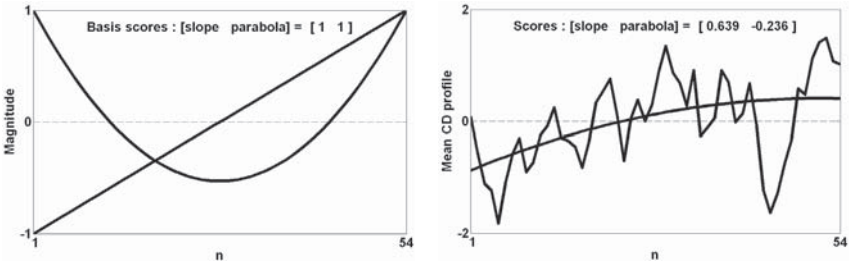


Figure 10.17. [Slope Parabola] as bases and the approximation of $\mathbf{y}_{CD}$.

First- and second-order Gram polynomials provide simple but powerful orthogonal bases to test the 'flatness' of a profile measurement. For a profile to be flat, it should at least have approximately 0 as the magnitudes for its second-order Gram polynomial approximation. Consider the modified version of the basis functions as shown in Figure 10.17 where $\mathbf{p}_1$ has a positive slope for convenience (opposite of the formal definition of Gram polynomials, see Eq. 10.3) and the basic scores are [1 1] at full scale. Approximation of sheet data $\mathbf{y}_{CD}$ indicates scores of $[0.639 \ -0.236]$, which are significant. Clearly, requiring only these scores to be 0 is not sufficient for $\mathbf{y}_{CD}$ uniformity, but it is a necessary first step.

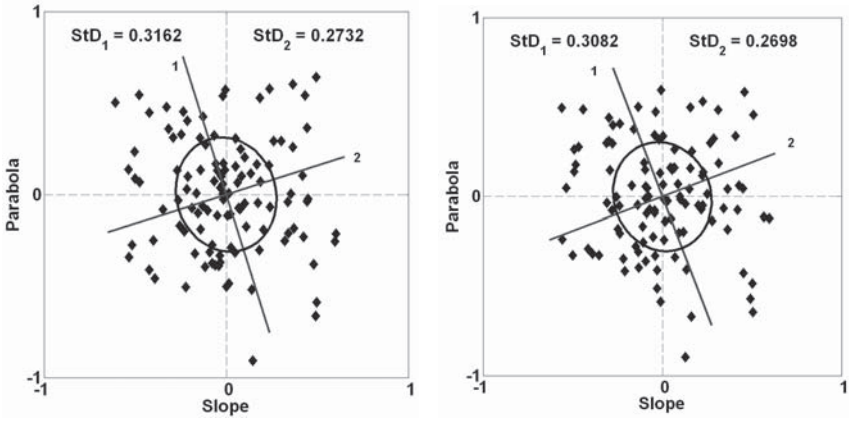


Figure 10.18. PCA coordinates $[\mathbf{u}_1\mathbf{u}_2]$ of the [Slope Parabola] projections of $\mathbf{Y}_R$ (left plot) and $\mathbf{Y}_{RM}$ scans (columns) with $M = 4$.

A similar transformation of all scan data residuals, i.e., columns of $\mathbf{Y}_R$, is accomplished by $\mathbf{Y}_R \approx \mathbf{\Phi B}$ and $\mathbf{B} = \mathbf{\Psi Y}_R$ where $\mathbf{\Phi} = [\mathbf{p}_1\mathbf{p}_2]$ and $\mathbf{\Psi} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T$. $\mathbf{B}$ is the $2 \times K$ scores matrix measuring the magnitudes of slope and parabola bases for each residual scan contained in $\mathbf{Y}_R$. A phase-plane scatter plot of $\mathbf{B}$ provides a concise view of flatness in terms of deviations from the target [0 0]. Further characteristics of the scores are measured through a PCA decomposition by establishing equivalent dominant eigenvectors $\mathbf{U}$. Through SVD of $\mathbf{Z} = \mathbf{BB}^T$, the two basis vectors $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2]$ are identified with a measure of transformed scores $\mathbf{A} = \mathbf{U}^T\mathbf{B}$. Figure 10.18 shows the scatter plots for $\mathbf{Y}_R$ and $\mathbf{Y}_{RM}$ $(M = 4)$ with the corresponding PCA alignments and the standard deviation contours as reference. Both $\mathbf{Y}_R$ and $\mathbf{Y}_{RM}$ are very similar indicating the effectiveness of fourth-order PCA approximation. Significant scatter of the scores show that the residual scans contain structured variabilities measurable as first- and second-order polynomials indicating non-flat behavior. Further evidence of $\mathbf{Y}_R$ PCA approximation accuracy is displayed in Figure 10.19 where the phase-plane plots for $\mathbf{Y}_{RR} = \mathbf{Y}_R - \mathbf{Y}_{RM}$ are shown for fourth- and sixth-order approximations. Increasing the number of PCA modes from 4 to 6 adds marginal improvement for [Slope Parabola] projections. As a process performance objective, Figure 10.18 plots for $\mathbf{Y}_R$ and $\mathbf{Y}_{RM}$ should look similar to that of $\mathbf{Y}_{RR}$ in Figure 10.19. During continuous process improvement a logical followup to the minimization of $[\mathbf{p}_1\mathbf{p}_2]$ projections is to do the same with $[\mathbf{p}_3\mathbf{p}_4]$ projections, and so on until all dominant trends are eliminated and $\mathbf{Y}_R$ columns contain only high-frequency random signals.

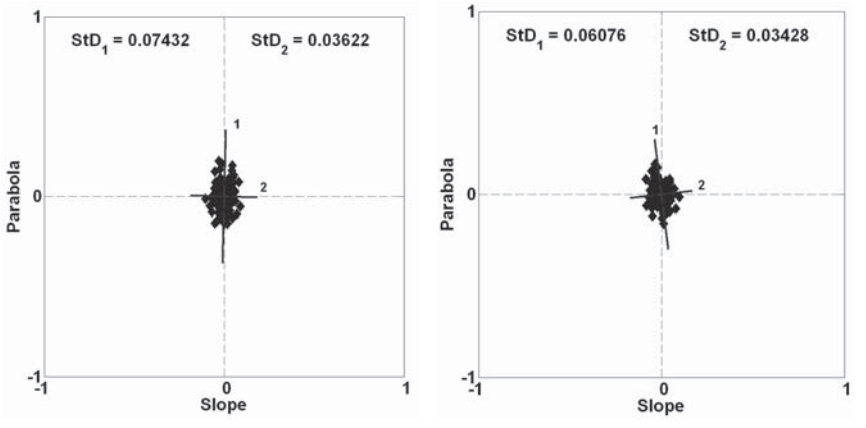


Figure 10.19. PCA coordinates of the [Slope Parabola] projections of $\mathbf{Y}_{RR} = \mathbf{Y}_R - \mathbf{Y}_{RM}$ with $M = 4$ (left plot) and $M = 6$.

The quadrants of the phase-plane plots divide projection scores into different characteristic patterns. For example, quadrant 1 is uphill slope and smiling parabola, quadrant 3 is downhill slope and frowning parabola, etc. Figure 10.20 shows the larger scores of each quadrant with a distinct shade of gray while allowing a central elliptic region to be the lower limit for shape classification. Slope and parabola limits of the plot are arbitrarily assigned for demonstration purposes. Creating five distinct classifications for scores that can be used as a simple filter is a form of masking for additional data mining. Data points of scores representing $K = 110$ scans provide an overall accounting of different shape trends while differentiating almost flat from the significantly shaped scans. Another view of the same information is also included in Figure 10.20 as a top view of the sheet with scan colors reflecting the designated score shade. The lower magnitude scores within the ellipsoid limit are neutral (white) in the sheet view, though they were gray circles in the score plot for visibility. Sheet view of the scores contain temporal information indicating frequency of changes between shape patterns. For example, first and second halves of the sheet do not have similar patterns. The switch from lighter to darker shades imply a change in process disturbance behavior. Obviously, the performance objective is to have all scores within the ellipsoid limit and a sheet view without any stripes.

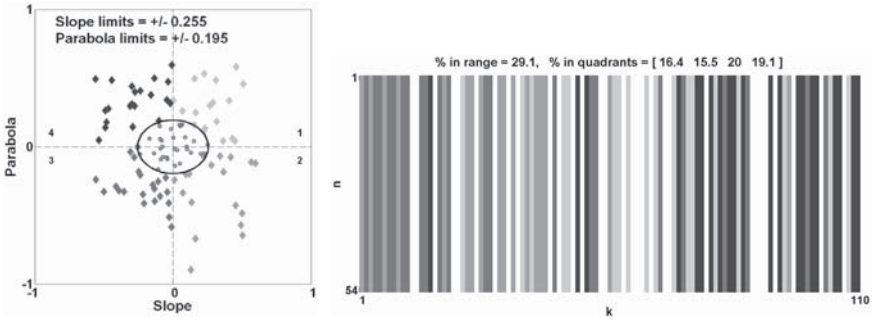


Figure 10.20. Masking of $\mathbf{Y}_{RM}$ [Slope Parabola] projections to highlight significant deviations and characteristic patterns.

10.3 Controller Performance

Sheet forming processes have univariate MD and multivariate CD controllers. Process dynamics for both are dominated by gain and time delay. Most of the appreciable dynamics arise from the design of signal filters and

feedback loop interactions. Performance evaluation of MD control basically follows the same principals as any other univariate control system. On the other hand CD control performance evaluation is more involved reflecting the difficulties that arise in CD controller design due to high dimensionality and strong correlations of decision variables. In the following sections, MD and CD control performance evaluations will be discussed. The methodologies presented for CD controller design and performance evaluation are both model-based and reflect recent developments in the technology.

10.3.1 MD Control Performance

MD control performance is evaluated directly from the closed-loop time series data $\mathbf{y}_{MD}$. Let deviation variable y_t represent the measurement signal at time t. One way of representing y_t in terms of previous measurements is through a moving average (MA) correlation model

$$y(k) = (1 + \sum_{j=1}^{\infty} \psi_j q^{-1}) a(k) \tag{10.4}$$

where $a(k)$ is the noise signal, q^{-1} is the backward difference operator and ψ_j is a model constant. For a process with an effective time delay of h sampling intervals, Eq. 10.4 can be restated in two parts as

$$\begin{aligned} y(k) = (1 + \psi_1 q^{-1} + ... \quad &+ \quad \psi_{h-1} q^{-h+1}) a(k) + \\ &+ (\psi_h \quad + \quad \psi_{h+1} q^{-1} + ... + \psi_{h+l} q^{-l} + ...) a(k-h) \end{aligned} \tag{10.5}$$

The second term is an h-step-ahead prediction of $y(k)$ while the first term is the prediction error. The best a controller can do is to eliminate the deviation represented by the second term. Thus, theoretical minimum variance is $S^2_{min} = (1 + \sum_{j=1}^{h-1} \psi_j^2) S^2_a$, which notably requires the calculations of only $h-1$ constants ψ_j and variance of noise, traditionally done through Yule-Walker equations using the autocorrelation coefficients of $\mathbf{y}$. For process control purposes, total variance of measurement signal is $S^2 = {S_y}^2 + \bar{y}^2$ where the contribution of offset from target is also accounted. Normalized performance index (NPI) is $\eta(h) = 1 - S^2_{min}/S^2$, which measures the margin of improvement opportunity for the controller.

NPI for sheet data $\mathbf{y}_{MD}$ is calculated for a range of measurement delays $h = 1, \ldots, 20$ and plotted in Figure 10.21. In most sheet processes where data are collected with a traversing scanner the measurement delay is 2 or 3. The plot shows that there is essentially no significant margin for additional MD control improvement. Confirmation of MD controller performance can be seen in Figure 10.22 where the fifth-order auto regressive (AR) model of $\mathbf{y}_{MD}$ is compared to the residual noisy signal. The

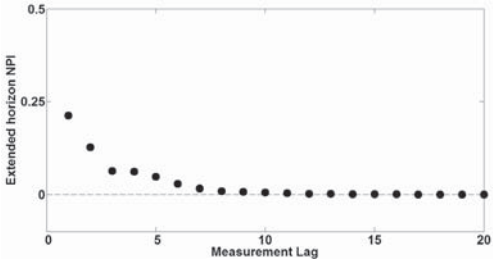


Figure 10.21. Normalized performance index NPI of $\mathbf{y}_{MD}$ to measure MD control improvement potential.

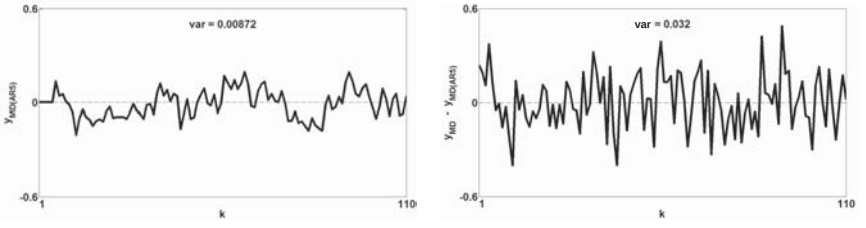


Figure 10.22. Fifth-order AR approximation of $\mathbf{y}_{MD}$ and the residual random signal.

latter captures more than 75% of the total variance. Autocorrelation coefficients of $\mathbf{y}_{MD}$ as a function of measurement lag are plotted in Figure 10.23 with the corresponding 95% confidence limits. Within process time delay of lag 3, the magnitude of the coefficients is reduced below desired limit, again confirming the general satisfactory behavior of the controller. However, in the same plot, it is also clear that there is a cycling trend of the autocorrelation coefficients with increasing measurement lag that suggests possibility of a tightly tuned controller. Similar plot for the AR(5) model shows the cycling nature of $\mathbf{y}_{MD}$ signal correlations more clearly. For a well-tuned and efficiently performing controller the autocorrelation bars should be randomly varying within the confidence limits starting shortly after the process measurement time lag.

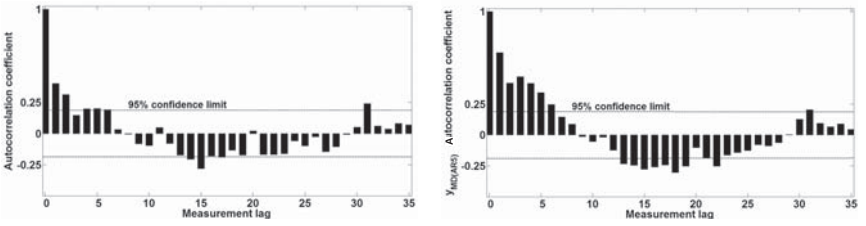


Figure 10.23. Autocorrelation coefficients of $\mathbf{y}_{MD}$ and its AR(5) model approximation.

10.3.2 Model-Based CD Control Performance

Realistic CD control performance calculation is not possible from direct computations only using measured process output data. Although traditionally it has been suggested that minimum variance of CD profile should be linked to the Nyquist frequency of measurement data length, such an approach results in a rather aggressive and unrealistic estimation of what can actually be achieved through a well designed CD controller. The reason is the high dimensionality and correlated interactions of the control elements or the slice actuators that render 'perfect' CD control impossible. Instead, for a given process, an 'optimal' CD control performance can be estimated through simulation, which can be compared to actual process output data to calculate improvement potential in terms of a normalized performance index. The model-based simulation approach for control performance evaluation is a general method that can be used for other applications which may require more accuracy beyond methods that use direct process output data only.

NPI for the model-based approach is defined as $\eta = 1 - S^2_{Y(opt)}/S^2_Y$ where $S^2_{Y(opt)}$ is calculated through simulation using process related data as shown in Figure 10.24. There are two parts to the calculations, disturbance estimation and achievable performance estimation. Process disturbance is estimated from the difference between actual process output data $\mathbf{Y}$ and model prediction from control decisions $\mathbf{U}$. In turn, for the optimal performance estimation, $\mathbf{D}_{(est)}$ is used as the input of a closed-loop simulated control system. The process model (Model A) in the simulated control system is the same model used in disturbance estimation, while the optimal

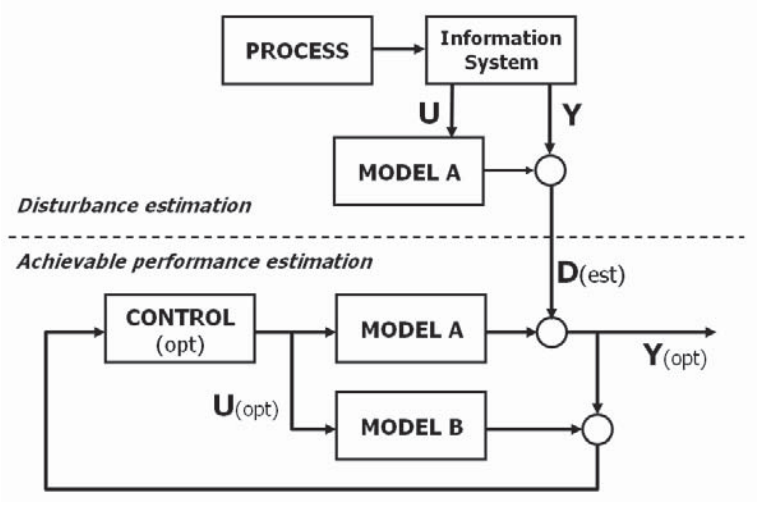


Figure 10.24. Block diagram of the general approach to calculate best achievable (optimal) controller performance through simulation.

controller reference model (Model B) can be different and simpler. It is important to emphasize that disturbance estimation calculations use only the output of the actual implemented control system and not the algorithmic details. Thus, it is possible to carry out these calculation without requiring proprietary information from control vendors. For CD control application, the design of the simulated optimal controller does not imply aggressive behavior to achieve minimum variance. The control tuning should be realistic resulting in mild actions to avoid picketing while satisfying all necessary constraints imposed by slice physical characteristics.

The paper machine example first introduced in Section 10.1.1 reflects open-loop in CD control data. Application of model-based CD control performance calculations is simplified for this case as $\mathbf{U} = \mathbf{0}$ while the results establish an upper limit of improvement expectations from a potential CD control implementation. Following the procedure summarized in the block diagram of Figure 10.24 simulation calculations provide $S^2_{Y(opt)} = 0.0644$ or $\eta = 1 - 0.0644/0.625 \approx 0.9$. This is of course a theoretical target and the practical reality may be in the range 0.5 to 0.7, which would still be a significant improvement.

A unique advantage of the model-based CD control performance calculations is the two-dimensional information detail for improvement potential. Figure 10.25 shows the process variability reduction for each measuremen-

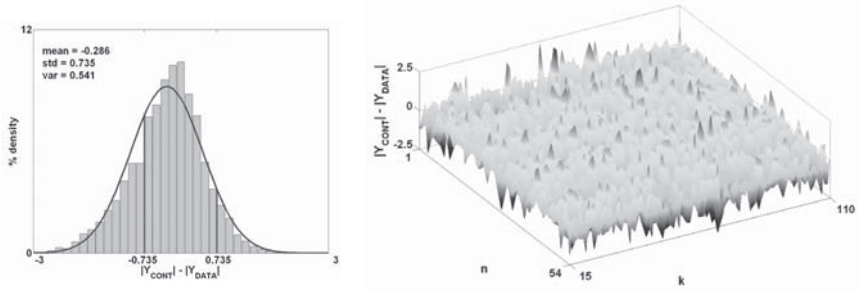


Figure 10.25. Improvement in profile data as a result of simulated CD control implementation. Differences in absolute values of local data show improvements as reductions in deviation magnitudes.

t location as a difference in deviation magnitudes ($|\mathbf{Y}_{OptCont}| - |\mathbf{Y}_{Data}|$), which is a negative number at any position where original process deviation is reduced through simulated CD control implementation. Histogram of the collective data has a negative mean (-0.286) confirming overall improvement potential. Performance analysis is based on the computed results for the last 96 scans as the first 14 were part of the controller filter initiation, which is a natural artifact of working with a batch data file. However, the procedure is equally valid for on-line applications where η and the corresponding visualization diagrams can be tracked in real time in terms of moving windows of preselected scan lengths.

Another informative visualization of the results is through a 2D plot showing CD controller effects at each location as one of three categories: (a) significant reduction in deviation magnitude from target (dark), (b) mild reduction or amplification (gray), or (c) significant amplification (light). For demonstration purposes the standard deviation limits [$\pm(S = 0.735)$] of the ($|\mathbf{Y}_{OptCont}| - |\mathbf{Y}_{Data}|$) data are used as color masking boundaries and the results are shown in Figure 10.26. Corresponding color masking of the fourth-order PCA approximation clearly shows that the significant improvement areas are correlated and match the strong deviation locations of the original profile shown in Figures 10.3 and 10.15. This is a confirmation that the CD controller improvement potential calculations are based on appropriately targeted variability reduction. A simple implementation of the masked PCA approximation as an on-line monitoring metric is to

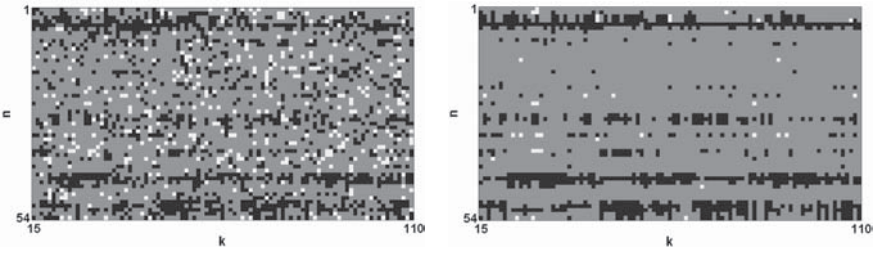


Figure 10.26. CD control implementation improvements showing significant differences of $(|\mathbf{Y}_{OptCont}| - |\mathbf{Y}_{Data}|)$ on a 2D plot by masking the results into three categories: $[< -0.735] = dark$, $[-0.735$ to $0.375] = gray$ and $[> 0.735] = light$. Similar masking of the fourth-order PCA approximation (right) emphasizes correlated improvement locations.

expect the complete 2D plot to be in gray color within quality limits that are reasonable for the specific product. Calculated dark colors, showing significant improvement potential locations, that cover more than $2 - 5\%$ of the plot would flag reduction in CD control performance.

10.4 Summary

Process data for web and sheet forming processes are in two-dimensional form describing spatio-temporal properties that are practically described as cross direction (CD) and machine direction (MD) for space and time, respectively. Mean data values at discrete CD increments describe average property profile which needs to be kept on target for optimum product value. Both process and controller performance analyses focus on degree of data variability in MD and CD averages as well as the residual of data after the removal of MD/CD trends. Extraction and quantitative analysis of structured correlations in the two-dimensional residual profile can be done using orthogonal decomposition methods like Gram polynomials, principal components analysis (PCA), and wavelet denoising. These methods can identify any significant process variability information hidden within the otherwise seemingly random nature of residual data. Rigorous CD control application of sheet forming processes has unique complications arising from the strong correlations of its large scale and constrained decision variables.

Accordingly, CD control performance evaluation requires a special method of model-based approach to compute improvement potential based on a realistically achievable target for that particular process.

Bibliography

[1] LC Alwan and HV Roberts. Time series modeling for statistical process control. *J. Business and Economic Statistics*, 6:87–95, 1988.

[2] BDO Anderson and JB Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] TW Anderson. *Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, New York, NY, 2nd edition, 1984.

[4] D Antis, JL Slutsky, and CM Creveling. *Design for Six Sigma in Technology and Product Development*. Prentice-Hall PTR, Upper Saddle River, NJ, 2003.

[5] M Aoki. *State Space Modeling of Time Series*. Springer-Verlag, New York, NY, 2nd edition, 1990.

[6] M Bagshaw and RA Johnson. The effect of serial correlation on the performance of CUSUM tests. *Technometrics*, 17:73–80, 1975.

[7] A Bakhtazad, A Palazoglu, and JA Romagnoli. Process data denoising using wavelet transform. *Intelligent Data Analysis*, 4:267–285, 1999.

[8] BR Bakshi. Multiscale PCA with application to multivariate statistical monitoring. *AIChE J.*, 44(7):1596–1610, 1998.

[9] BR Bakshi and G Stephanopoulos. Representation of process trends, part iii. Multiscale extraction of trends from process data. *Comput. & Chem. Engg.*, 18:267–302, 1994.

[10] BR Bakshi and G Stephanopoulos. Compression of chemical process data by functional approximation and feature extraction. *AIChE J.*, 42:477–492, 1996.

[11] A Banerjee, Y Arkun, B Ogunnaike, and R Pearson. Estimation of non-linear systems using linear multiple models. *AIChE J.*, 43:1204–1226, 1997.

[12] DB Bates and DG Watts. *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons, New York, NY, 1988.

[13] LE Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

[14] S Beaver and A Palazoglu. A cluster aggregation scheme for ozone episode selection in the San Francisco, CA Bay Area. *Atmospheric Environment*, 40:713–725, 2006.

[15] S Beaver and A Palazoglu. Cluster analysis for autocorrelated and cyclic chemical process data. *Ind. & Engg. Chem. Research*, 2006. Submitted.

[16] S Beaver and A Palazoglu. Cluster analysis of hourly wind measurements to reveal synoptic regimes affecting air quality. *J. Applied Meteorology and Climatology*, 2006. In press.

[17] S Becker. Unsupervised learning procedures for neural networks. *Int. J. Neural Systems*, 2:17–33, 1991.

[18] KR Beebe and BR Kowalski. An introduction to multivariate calibration and analysis. *Anal. Chem.*, 59:1007A–1015A, 1987.

[19] DP Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 2nd edition, 2000.

[20] S Bezergianni and C Georgakis. Controller performance assessment based on minimum and open-loop output variance. *Chem. Engg. Practice*, 8:791–797, 2000.

[21] AW Bowman. Altrenative method of cross-validation for the smoothing of density estimate. *Biometrika*, 76:353–360, 1984.

[22] GEP Box. Some theorems on quadratic forms applied in the study of analysis of variance problems: Effect of inequality of variance in one-way classification. *The Annals of Mathematical Statistics*, 25:290–302, 1954.

[23] GEP Box, GM Jenkins, and GC Reinsel. *Time Series Analysis – Forecasting and Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 3rd edition, 1994.

[24] L Breiman and JH Friedman. Estimating optimal transformations for multiple regression and correlation. *J. Amer. Statist. Assoc.*, 80:580–598, 1985.

[25] R Bro and AK Smilde. Centering and scaling in component analysis. *J. Chemometrics*, 17:16, 2003.

[26] AG Bruce, DL Donoho, HY Gao, and RD Martin. Denoising and robust nonlinear wavelet analysis. In *SPIE Proceedings – Wavelet Applications*, page 2242, Orlando, FL, 1994.

[27] M Carmichael, R Vidu, A Maksumov, A Palazoglu, and P Stroeve. Using wavelets to analyze AFM images of thin films: Surface micelles and supported lipid bilayers. *Langmuir*, 20:11557–11568, 2004.

[28] B Chen and A Westerberg. *Proceedings of Process Systems Engineering (PSE)*. Elsevier, New York, NY, 2003.

[29] R Chen and RS Tsay. Nonlinear additive ARX models. *J. Amer. Statist. Assoc.*, 88(423):955–967, 1993.

[30] S Chen and SA Billings. Modeling and analysis of nonlinear time series. *Int. J. Control*, 49:2151–2171, 1989.

[31] S Chen and SA Billings. Representations of nonlinear systems: The NARMAX model. *Int. J. Control*, 49:1013–1032, 1989.

[32] Y Cheng, W Karjala, and DM Himmelblau. Resolving problems in closed loop nonlinear process identification using IRN. *Comput. & Chem. Engg.*, 20(10):1159–1176, 1996.

[33] CJ Chessari. *Studies in Modeling and Advanced Control*. PhD thesis, The University of Sydney, Australia, 1995.

[34] JTY Cheung and G. Stephanopoulos. Representation of process trends, Part I. A formal representation framework. *Comp. & Chem. Engg.*, 14:495–510, 1990.

[35] JTY Cheung and G Stephanopoulos. Representation of process trends, Part II. The problem of scale and qualitative scaling. *Comp. & Chem. Engg.*, 14:511–539, 1990.

[36] LH Chiang and RD Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag, London, UK, 2001.

[37] LH Chiang, ME Kotanchek, and AK Kordon. Fault diagnosis based on Fisher's discriminant analysis and support vector machines. *Comput. & Chem. Engg.*, 28(8):1389–1401, 2004.

[38] LH Chiang, EL Russell, and RD Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag, London, UK, 2001.

[39] K Choe and H Baruh. Sensor failure detection in flexible structures using modal observers. *J. Dynamic Systems Measurement and Control*, 115:411–418, 1993.

[40] Y-H Chu, SJ Qin, and C Han. Fault detection and operation mode identification based on pattern classification with variable selection. *Ind. & Engg. Chem. Research*, 43:1701–1710, 2004.

[41] A Cinar, S Parulekar, C Undey, and G Birol. *Batch Fermentation: Modeling, Monitoring, and Control*. Marcel Dekker, New York, NY, 2003.

[42] N Cristianini and J Shawe-Taylor. *Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[43] MS Crouse, RD Nowak, and RG Baraniuk. Wavelet-based statistical signal pocessing using hidden Markov models. *IEEE Trans. on Signal Processing*, 46:886–902, 1998.

[44] I Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Pennsylvania, 1992.

[45] I Daubechies. Where do wavelets come from? A personal point of view. *Proc. of IEEE*, 84(4):510–513, 1996.

[46] I Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. on Pure and Applied Math. X*, 51:909–996, 1998.

[47] ER Davies. The relative effects of median and mean filters on noisy signals. *J. Modern Optics*, 39:103–113, 1992.

[48] BS Dayal and JF MacGregor. Improved PLS algorithms. *J. Chemometrics*, 11:73–85, 1997.

[49] M deBoor. *A Practical Guide to Splines*. Springer-Verlag, New York, NY, 1978.

[50] J DeCicco and A Cinar. Empirical modeling of systems with output multiplicities by multivariate additive NARX models. *Ind. & Engg. Chem. Research*, 39(6):1747–1755, 2000.

[51] N Delfosse and P Loubaton. Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45:59–83, 1995.

[52] WE Deming. *Out of the Crisis*. MIT Press, Cambridge, MA, 1982.

[53] L Desborough and TJ Harris. Performance assessment measures for univariate feedback control. *Canadian J. of Chem. Engg.*, 70:1186–1197, 1992.

[54] WR DeVries and SM Wu. Evaluation of process control effectiveness and diagnosis of variation in paper basis weight via multivariate time series analysis. *IEEE Trans. on Automatic Control*, 23:702–708, 1978.

[55] D Dong and TJ McAvoy. Nonlinear principal components analysis based on principal curves and neural networks. *Comput. & Chem. Engg.*, 20(1):65–78, 1996.

[56] DL Donoho and IM Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[57] DL Donoho and TPY Yu. Nonlinear pyramid transforms based on median interpolation. *SIAM J. Math. Analysis*, 31:1030–1061, 2000.

[58] JJ Downs and EF Vogel. A plant-wide industrial control problem. In *AIChE Annual Meeting*, Chicago, IL, 1990.

[59] F Doymaz, A Bakhtazad, JA Romagnoli, and A Palazoglu. Wavelet-based robust filtering of process data. *Comput. & Chem. Engg.*, 25:1549–1559, 2001.

[60] F Doymaz, J Chen, JA Romagnoli, and A Palazoglu. A robust strategy for real-time process monitoring. *J. Process Control*, 11:343–359, 2001.

[61] F Doymaz, A Palazoglu, and JA Romagnoli. Orthogonal nonlinear partial least-squares. *Ind. & Eng. Chem. Research*, 42:5836–5849, 2003.

[62] F Doymaz, JA Romagnoli, and A Palazoglu. A strategy for detection and isolation of sensor faults and process upsets. *Chemometrics & Intell. Lab. Sys.*, 55:109–123, 2001.

[63] RO Duda, PE Hart, and DG Stork. *Pattern Classification*. John Wiley & Sons, New York, NY, 2nd edition, 2001.

[64] EJ Dudewicz and SN Mishra. *Modern Mathematical Statistics*. John Wiley & Sons, New York, NY, 1988.

[65] R Durbin, S Eddy, A Krogh, and G Mitchinson. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK, 1998.

[66] JR English, M Krishnamurthi, and T Sastri. Quality monitoring of continuous flow processes. *Comput. & Chem. Engg.*, 20:251–260, 1991.

[67] L Eriksson, E Johansson, N Kettaneh-Wold, and S Wold. *Multi- and Megavariate Data Analysis*. Umetrics Academy, Umeå, Sweden, 2001.

[68] M Evans, N Hastings, and B Peacock. *Statistical Distributions*. John Wiley & Sons, New York, NY, 1993.

[69] BS Everitt. *Cluster Analysis*. Heinemann Education, London, UK, 3rd edition, 1993.

[70] FW Faltin and WH Woodall. Some statistical process control methods for autocorrelated data – discussion. *J. Quality Technology*, 23:194–197, 1991.

[71] G Fan and XG Xia. Improved hidden Markov models in the wavelet-domain. *IEEE Trans. on Signal Processing*, 49:115–120, 2001.

[72] W Favoreel, B De Moor, and P Van Overschee. Subspace identification of bilinear systems subject to white inputs. Technical Report ESAT-SISTA/TR 1996-53I, Dept. Elektrotechniek Katholieke Universiteit Leuven, 1999.

[73] AP Featherstone and RD Braatz. Model–based cross–directional control. *Tappi J.*, 83(3):203–207, 1999.

[74] RA Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[75] I Frank. A nonlinear PLS model. *Chemometrics & Intell. Lab. Sys.*, 8:109–119, 1990.

[76] JH Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19:1–144, 1991.

[77] JH Friedman and W Stuetzel. Projection pursuit regression. *J. Amer. Statist. Assoc.*, 76:817–823, 1981.

[78] T Fujiwara, M Koyama, and H Nishitani. Extraction of operating signatures by episodic representation. In *Advanced Control of Chemical Processes: IFAC Symposium*, pages 333–338, Kyoto, Japan, 1994.

[79] K Fukunaga. *Statistical Pattern Recognition.* Academic Press, San Diego, CA, 1990.

[80] C Gackenheimer, L Cayon, and R Relfenberger. Analysis of scanning probe microscope images using wavelets. *Ultramicroscopy*, 106:389–397, 2006.

[81] O Galán, A Palazoglu, and JA Romagnoli. Robust H_∞ control of nonlinear plants based on multi-linear models – An application to a bench scale pH neutralization reactor. *Chem. Engg. Sci.*, 55:4435–4450, 2000.

[82] O Galán, JA Romagnoli, and A Palazoglu. Real-time implementation of multi-linear model-based control strategies. An application to a bench-scale pH neutralization reactor. *J. Process Control*, 14:571–579, 2004.

[83] O Galán, JA Romagnoli, A Palazoglu, and Y Arkun. The gap metric concept and implications for multi-linear model-based controller design. *Ind. & Eng. Chem. Research*, 42:2189–2197, 2003.

[84] F Gao. *Proceedings of Advanced Control of Chemical Processes (ADCHEM).* Elsevier, New York, NY, 2004.

[85] P Geladi. Wold, Herman, the father of PLS. *Chemometrics & Intell. Lab. Sys.*, 15:R7–R8, 1992.

[86] P Geladi and BR Kowalski. An example of 2-block predictive partial least-squares regression with simulated data. *Analytica Chimica Acta*, 185:19–32, 1986.

[87] P Geladi and BR Kowalski. Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[88] Gensym. *G2® Reference Manual.* Gensym Corporation, Cambridge, MA, 1997.

[89] *www.gensym.com.* [Accessed 11 July 2006].

[90] S Ghael, AM Sayeed, and RG Baraniuk. Improved wavelet denoising via empirical Wiener filtering. In AF Laine, MA Unser, and A Aldroubi, editors, *SPIE Technical Conference on Wavelet Applications in Signal Processing VI*, volume 3458, San Diego, CA, 1997.

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

[92] GH Golub and CF van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 2nd edition, 1989.

[93] C Goutis. A fast method to compute orthogonal loadings partial least squares. *J. Chemometrics*, 11:33–38, 1997.

[94] SP Gurden, JA Westerhuis, R Bro, and AK Smilde. A comparison of multiway regression and scaling methods. *Chemometrics & Intell. Lab. Sys.*, 59(1-2):121–136, 2001.

[95] A Haar. Zur theorie der orthogonalen funktionen-systeme. *Math. Annals*, 69:331–371, 1910.

[96] H Haario and V-M Taavitsainen. Nonlinear data analysis. II. Examples on new link functions and optimization aspects. *Chemometrics & Intell. Lab. Sys.*, 23:51–64, 1994.

[97] R Haber and H Unbehauen. Structure identification of nonlinear dynamic systems – A survey on input/output approaches. *Automatica*, 26:651–677, 1990.

[98] V Haggan and T Ozaki. Modeling nonlinear random vibrations using an amplitude dependent autoregressive time series model. *Biometrika*, 68:186–196, 1981.

[99] AC Hahn. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, New York, NY, 1989.

[100] GJ Hahn and WQ Meeker. *Statistical Intervals. A Guide to Practitioners*. John Wiley & Sons, New York, NY, 1991.

[101] FR Hampel, EM Ronchetti, PJ Rousseeuw, and WA Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, New York, NY, 1986.

[102] TJ Harris. Assessment of control loop performance. *Can. J. Chem. Engg.*, 67:856–861, 1989.

[103] TJ Harris, F Boudreau, and JF MacGregor. Performance assessment of multivariate feedback controllers. *Automatica*, 32:1505–1518, 1996.

[104] TJ Harris and WH Ross. Statistical process control procedures for correlated observations. *Canadian J. Chem. Engg.*, 69:48–57, 1991.

[105] TJ Harris, CT Seppala, and LD Desborough. A review of performance monitoring and assessment techniques for univariate and multivariate control systems. *J. Process Control*, 9:1–17, 1999.

[106] T Hastie and W Stuetzle. Principal curves. *J Amer. Statist. Assoc.*, 84(406):502–516, 1989.

[107] S Haykin. *Neural Networks.* Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[108] RR Hocking and RN Leslie. Selection of the best subset in regression analysis. *Technometrics*, 9(4):531–540, 1967.

[109] AE Hoerl and RW Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[110] A Horch and AJ Isaksson. A modified index for control performance assessment. In *Proc. of ACC98.* IEEE, 1998.

[111] A Hoskuldsson. PLS regression methods. *J. Chemometrics*, 2:211–228, 1988.

[112] H Hotelling. The generaliztion of Student's ratio. *Ann. Math. Statist.*, 2:360–378, 1931.

[113] H Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, 24:417, 1933.

[114] B Huang and SL Shah. Practical issues in multivariable feedback control performance assessment. *J. Process Control*, 8(5-6):421–430, 1998.

[115] B Huang and SL Shah. *Performance Assessment of Control Loops.* Springer-Verlag, London, UK, 1999.

[116] B Huang, SL Shah, and EK Kwok. Good, bad, or optimal? Performance assessment of multivariable processes. *Automatica*, 33:1175–1183, 1997.

[117] XD Huang, Y Ariki, and MA Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, UK, 1990.

[118] R Hudlet and R Johnson. Linear discrimination and some further results on best lower dimensional representations. In V Rzyin, editor, *Classification and Clustering*, pages 371–394. Academic Press, Inc., New York, NY, 1977.

[119] A Hyvärinen, J Karhunen, and E Oja. *Independent Component Analysis.* John Wiley & Sons, New York, NY, 2001.

[120] JE Jackson. Principal components and factor analysis : Part I - principal components. *J. Quality Technology*, 12(4):201–213, 1980.

[121] JE Jackson. *A Users Guide to Principal Components*. John Wiley & Sons, New York, NY, 1991.

[122] JE Jackson and GS Mudholkar. Control procedures for residuals associated with principal components analysis. *Technometrics*, 21:341–349, 1979.

[123] M Jelali. An overview of controller performance assessment technology and industrial applications. *Control Engg. Practice*, 14:441–466, 2006.

[124] XJ Jiao, MS Davies, and GA Dumont. Wavelet packet analysis of paper machine data for control assessment and trim loss optimization. *Pulp & Paper Canada*, 105(9):T208–211, 2004.

[125] P Jofriet, C Seppala, M Harvey, B Surgenor, and TJ Harris. An expert system for control loop performance. *Pulp & Paper Canada*, 97:207–211, 1996.

[126] RA Johnson and DW Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 4th edition, 1998.

[127] LPM Johnston and MA Kramer. Probability density estimation using elliptical basis functions. *AIChE J.*, 40:1639–1649, 1994.

[128] IT Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, 1986.

[129] IT Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, 2nd edition, 2002.

[130] EM Jordaan and GF Smits. Estimation of the regularization parameter for support vector regression. In *Proc. World Conf. Computational Intelligence*, pages 2785–2791, Honolulu, Hawaii, 2002.

[131] M Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. Eighth Annual Conf. of the Cognitive Science Society*, Amherst, MA, 1986.

[132] BC Juricek, DE Seborg, and WE Larimore. Predictive monitoring for abnormal situation management. *J. Process Control*, 11:111–128, 2001.

[133] BC Juricek, DE Seborg, and WE Larimore. Fault detection using canonical variate analysis. *Ind. & Engg. Chem. Research*, 43:458–474, 2004.

[134] C Jutten and J Herault. Blind separation of sources: I. An adaptive algorithm based on neuromimetric architecture. *Signal Process.*, 24:1, 1991.

[135] RE Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME – J. Basic Engineering*, 82:34–45, 1960.

[136] LC Kammer, RR Bitmead, and PL Bartlett. Optimal controller properties from closed-loop experiments. *Automatica*, 34:83–91, 1998.

[137] M Kano, S Hasebe, I Hashimoto, and H Ohno. Evolution of multivariate statistical process control: Independent component analysis and external analysis. *Comput. & Chem. Engg.*, 28(6-7):1157–1166, 2004.

[138] M Kano, S Tanaka, S Hasebe, I Hashimoto, and H Ohno. Monitoring independent components for fault detection. *AICHE J.*, 49:969–976, 2003.

[139] SJ Kendra, MR Basila, and A Cinar. Intelligent process control with supervisory knowledge-based systems. *IEEE Control Systems*, 14:37–47, 1994.

[140] SJ Kendra and A Cinar. Controller performance assessment by frequency domain techniques. *J. Process Control*, 7(3):181–194, 1997.

[141] P Kesavan and JH Lee. Diagnostic tools for multivariable model-based control systems. *Ind. & Engg. Chem. Research*, 36:2725–2738, 1997.

[142] KB Konstantinov and T Yoshida. Real-time qualitative analysis of the temporal shapes of (bio)process variables. *AIChE J.*, 38(11):1703–1715, 1992.

[143] F Kosebalaban and A Cinar. Integration of multivariate SPM and FDD by parity space technique for a food pasteurization process. *Comput. & Chem. Engg.*, 25:473–391, 2001.

[144] T Koski. *Hidden Markov Models for Bioinformatics*. Prentice-Hall, Boston, MA, 1999.

[145] T Kourti and JF MacGregor. Process analysis, monitoring and diagnosis using multivariate projection methods. *Chemometrics & Intell. Lab. Sys.*, 28:3–21, 1995.

[146] T Kourti and JF MacGregor. Multivariate SPC methods for process and product monitoring. *J. Quality Technology*, 28(4):409–428, 1996.

[147] BR Kowalski. *Chemical Process Control – V Conference Proceedings*, chapter Process Analytical Chemical Engineering, pages 97–101. AIChE Symposium Series 316. CACHE–AIChE, 1997.

[148] DJ Kozub. Controller performance monitoring and diagnosis : Experiences and challenges. In *CPC V Proceedings*, pages 83–96, Lake Tahoe, NV, 1997.

[149] DJ Kozub and CE Garcia. Monitoring and diagnosis of automated controllers in the chemical process industry. In *AICHE Annual Meeting, St. Louis, MO*, 1993.

[150] MA Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.*, 37:233–243, 1991.

[151] MA Kramer. Autoassociative neural networks. *Comput. & Chem. Engg.*, 16(4):313–328, 1992.

[152] MA Kramer and JA Leonard. Diagnosis using backpropagation neural networks – Analysis and criticism. *Comput. & Chem. Engg.*, 14:1323–1338, 1990.

[153] JV Kresta, JF MacGregor, and TE Marlin. Multivariate statistical monitoring of process operating performance. *Canadian J. Chem. Engg.*, 69:35–47, 1991.

[154] WJ Krzanowski. Between-groups comparison of principal components. *J. Amer. Statist. Assoc.*, 74:703–707, 1979.

[155] WJ Krzanowski. Cross-validation choice in principal component analysis. *Biometrics*, 43:575–584, 1987.

[156] A Kulkarni, VK Jayaraman, and BD Kulkarni. Support vector classification with parameter tuning assisted by agent-based systems. *Comput. & Chem. Engg.*, 28:311–318, 2004.

[157] S Lakshminarayanan, SL Shah, and K Nandakumar. Identification of Hammerstein models using multivariate statistical tools. *Chem. Engg. Science*, 50(22):3599–3613, 1995.

[158] WE Larimore. System identification, reduced-order filtering and modeling via canonical variate analysis. In *Proc. of Automatic Control Conf.*, page 445, 1983.

[159] WE Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *Proc. of IEEE Conf. on Decision and Control*, page 596, 1990.

[160] WE Larimore. Identification and filtering of nonlinear systems using canonical variate analysis. In *Nonlinear Modeling and Forecasting: Proc of the Workshop on Nonlinear Modeling and Forecasting, Santa Fe, NM, Vol 12*. Addison-Wesley, 1990.

[161] M LeBlanc and R Tibshirani. Adaptive principal surfaces. *J. Amer. Statist. Assoc.*, 89(425):53–64, 1994.

[162] J-M Lee, SJ Qin, and I-B Lee. Fault detection and diagnosis of multivariate processes based on modified independent components analysis. *AIChE J.*, 2006. Submitted.

[163] J-M Lee, CK Yoo, and I-B Lee. New monitoring technique with ica algorithm in wastewater treatment process. *Water Science and Technology*, 47:49–56, 2003.

[164] J-M Lee, CK Yoo, and I-B Lee. Statistical process monitoring with independent components analysis. *J. Process Control*, 14:467–485, 2004.

[165] J Leonard and MA Kramer. Improvement of the backpropagation algorithm for training neural networks. *Comput. & Chem. Engg.*, 14(3):337–341, 1990.

[166] J Leonard, MA Kramer, and LH Ungar. A neural network architecture that computes its own reliability. *Comput. & Chem. Engg.*, 16(9):819–835, 1992.

[167] IJ Leontaritis and SA Billings. Input-output parametric models for nonlinear systems. *Int. J. Control.*, 41:303–344, 1985.

[168] D Lieftucht, U Kruger, L Xie, T Littler, Q Chen, and S-Q Wang. Statistical monitoring of dynamic multivariate processes – Part 2. Identifying fault magnitude and signature. *Ind. & Engg. Chem. Research*, 45:1677–1688, 2006.

[169] F Lindgren, P Geladi, S Rännar, and S Wold. Interactive variable selection (IVS) for PLS. Part I. Theory and algorithms. *J. Chemometrics*, 8:349–363, 1994.

[170] L Ljung. *System Identification: Theory for the user*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 1999.

[171] L Ljung and T Glad. *Modeling of Dynamic Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[172] A Lorber, L Wangen, and B Kowalski. A theoretical foundation for the PLS algorithm. *J. Chemometrics*, 1:19–31, 1987.

[173] C-W Lu and MR Reynolds, Jr. *Control charts based on residuals for monitoring autocorrelated processes*. Technical Report 94-8, Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1994.

[174] H Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, Germany, 1991.

[175] CB Lynch and GA Dumont. Control loop performance monitoring. *IEEE Trans. on Control System Technology*, 4:185–192, 1996.

[176] JF MacGregor. Some statistical process control methods for autocorrelated data – discussion. *J. Quality Technology*, 23:198–199, 1991.

[177] JF MacGregor, C Jaeckle, C Kiparissides, and M Koutoudi. Process monitoring and diagnosis by multiblock PLS methods. *AIChE J.*, 40(5):826–838, 1994.

[178] JB MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, 1967. Univesity of California Press.

[179] PC Mahalanobis. On tests and measures of group divergence. *J. Proc. Asiatic Soc. Bengal*, 26:541–588, 1930.

[180] A Maksumov, R Vidu, A Palazoglu, and P Stroeve. Enhanced feature analysis using wavelets for scanning probe microscopy images of surfaces. *J. Colloid and Interface Science*, 272:365–377, 2004.

[181] ER Malinowski. Statistical F-tests for abstract factor analysis and target testing. *J. Chemometrics*, 3:49–60, 1988.

[182] SG Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[183] EC Malthouse. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Trans. on Neural Networks*, 9(1):165–173, 1998.

[184] EC Malthouse, AC Tamhane, and RSH Mah. Nonlinear partial least squares. *Comput. & Chem. Engg.*, 21(8):875–890, 1997.

[185] B Maner, FJ Doyle III, B Ogunnaike, and R Pearson. Nonlinear model predictive control of a multivariable polymerization reactor using second-order Volterra series. *Automatica*, 32:1285–1302, 1996.

[186] HD Maragah and WH Woodall. The effect of autocorrelation on the retrospective x-chart. *J. Statist. Comput. Simul.*, 40:29–42, 1992.

[187] PZ Marmarelis and VZ Marmarelis. *Analysis of Physiological Systems*. Plenum Press, New York, NY, 1978.

[188] H Martens and T Næs. *Multivariate Calibration*. John Wiley & Sons, New York, NY, 1989.

[189] EB Martin and AJ Morris. Monitoring performance in flexible process monitoring. In *Preprints IFAC ADCHEM 7*, pages 47–54, Hong Kong, 2004.

[190] RL Mason and JC Young. *Multivariate Statistical Process Control with Industrial Applications*. ASA-SIAM, Philadelphia, 2002.

[191] MathWorks. *Matlab® System Identification Toolbox*. The MathWorks, Inc., Natick, MA, 2001.

[192] WS McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Mathematical Biophysics*, 5:115–133, 1943.

[193] RC McFarlane, RC Reineman, JF Bartee, and C Georgakis. Dynamic simulator for a model IV fluid catalytic cracking unit. *Comput. & Chem. Engg.*, 17(3):275–300, 1993.

[194] GJ McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, New York, NY, 1992.

[195] CA McNabb and SJ Qin. Projection based MIMO control performance monitoring – I. Covariance monitoring in state space. *J. Process Control*, 13:739–759, 2003.

[196] CA McNabb and SJ Qin. Projection based mimo control performance monitoring – II. Measured disturbances. *J. Process Control*, 15:89–102, 2005.

[197] P Miller and RE Swanson. Contribution plots: The missing link in multivariate quality control. In *37th Annual Fall Technical Conf., ASQC*, Rochester, NY, 1993.

[198] P Miller, RE Swanson, and CF Heckler. Contribution plots: The missing link in multivariate quality control. *Int. J. App. Math. & Comp. Science*, 8(4):775–792, 1998.

[199] M Misiti, G Oppenheim, J-M Poggi, and Y Misiti. *Wavelet Toolbox User's Guide (For Use with Matlab®)*. Mathworks, Natick, MA, 1996.

[200] M Misra, S Kumar, SJ Qin, and D Seemann. Error based criterion for on-line wavelet data compression. *J. Process Control*, 11(6):717–731, 2001.

[201] RR Mohler. *Bilinear Control Processes*. Academic Press, New York, NY, 1973.

[202] DC Montgomery and CM Mastrangelo. Some statistical process control methods for autocorrelated data. *J. Quality Technology*, 23:179–193, 1991.

[203] DC Montgomery and GC Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, New York, NY, 1st edition, 1994.

[204] M Morari and L Ricker. *Model Predictive Control Toolbox for use with Matlab®*. The Mathworks Inc., Natick, MA, 1998.

[205] RL Motard and B Joseph. *Wavelet Applications in Chemical Engineering*. Kluwer Academic Publishers, Boston, MA, 1994.

[206] T Naes and T Isaksson. Splitting of calibration data by cluster analysis. *J. Chemometrics*, 5:49–65, 1991.

[207] A Negiz. *Statistical dynamic modeling and monitoring methods for multivariable continuous processes*. PhD thesis, Illinois Institute of Technology, Department of Chemical and Environmental Engineering, Chicago, IL, 1995.

[208] A Negiz and A Cinar. On the detection of multiple sensor abnormalities in multivariable processes. In *Proc. American Control Conference*, pages 2364–2369, 1992.

[209] A Negiz and A Cinar. A parametric approach to statistical monitoring of processes with autocorrelated observations. In *AIChE Annual Meeting*, Miami, FL, 1995.

[210] A Negiz and A Cinar. PLS, balanced and canonical variate realization techniques for identifying varma models in state space. *Chemometrics & Intell. Lab. Sys.*, 38:209–221, 1997.

[211] A Negiz and A Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE J.*, 43(8):2002–2020, 1997.

[212] A Negiz and A Cinar. Monitoring of multivariable dynamic processes and sensor auditing. *J. Process Control*, 8(5-6):375–380, 1998.

[213] A Negiz, ES Lagergren, and A Cinar. Mathematical models for concurrent spray drying. *Ind. & Engg. Chem. Research*, 34:3289–3302, 1995.

[214] PRC Nelson, PA Taylor, and JF MacGregor. Missing data methods in PCA and PLS: score calculations with incomplete observations. *Chemometrics & Intell. Lab. Sys.*, 35:45–65, 1996.

[215] RB Newell and PL Lee. *Applied Process Control: A Case Study.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[216] I Nimmo. Adequately addressing abnormal operations. *Chem. Engg. Progress*, 91:36–45, 1995.

[217] P Nomikos. Detection and diagnosis of abnormal batch operations based on multiway principal components analysis. *ISA Trans.*, 35:259–266, 1996.

[218] P Nomikos and JF MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37:41–59, 1995.

[219] A Norvilas, A Negiz, J DeCicco, and A Cinar. Intelligent process monitoring by interfacing knowledge-based systems and multivariate statistical monitoring. *J. Process Control*, 10(4):341–350, 2000.

[220] AV Oppenheim and RW Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[221] A Papoulis. *Signal Analysis.* McGraw-Hill, New York, NY, 1977.

[222] RS Patwardhan and SL Shah. Issues in performance diagnostics of model-based controllers. *J. Process Control*, 12(3):413–427, 2002.

[223] RS Patwardhan, SL Shah, G Emoto, and H Fujii. Performance analysis of model-based predictive controllers: An industrial case study. In *Proc. of AIChE Annual Meeting*, Miami Beach, FL, 1998.

[224] R Payne. Predictive sensor diagnostics reduce downtime and costs. *I&CS*, 14:59–63, 1993.

[225] K Pearson. *Mathematical contributions to the theory of evolution XIII. On the theory of contingency and its relation to association and normal correlation.* Drapers Co. Res. Mem. Biometric series I, Cambridge University Press, London, UK, 1901.

[226] K Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2:559, 1901.

[227] RK Pearson and BA Ogunnaike. *Nonlinear Process Control*, chapter Nonlinear Process Identification. Prentice-Hall PTR, Upper Saddle River, NJ, 1997.

[228] DB Percival and AT Walden. *Wavelet Methods for Time Series Analysis.* Cambridge University Press, Cambridge, UK, 2000.

[229] AA Petrosian and FG Meyer. *Wavelets in Signal and Image Analysis: From Theory to Practice.* Kluwer Academic, Boston, MA, 2001.

[230] NP Piercy. Sensor failure estimators for detection filters. *IEEE Trans. on Automatic Control*, 37:1553–1558, 1992.

[231] M Pottmann and R Pearson. Block-oriented NARMAX models with output multiplicities. *AIChE J.*, 44(1):131–140, 1998.

[232] M Pottmann and DE Seborg. Identification of nonlinear processes using reciprocal multiquadric functions. *J. Process Control*, 2:189–203, 1992.

[233] MB Priestley. *Nonlinear and Nonstationary Time Series Analysis.* Academic Press, London, UK, 1988.

[234] DC Psichogios and LH Ungar. SVD-NET:An algorithm that automatically selects network structure. *IEEE Trans. on Neural Networks*, 5(3):513–515, 1994.

[235] S Qian. *Introduction to Time-Frequency and Wavelet Transforms.* Prentice-Hall, Upper Saddle River, NJ, 2002.

[236] SJ Qin. Controller performance monitoring. A review and assessment. *Comput. & Chem. Engg.*, 23:178–186, 1998.

[237] SJ Qin, S Valle, and MJ Piovoso. On unifying multiblock analysis with application to decentralized process monitoring. *J. Chemometrics*, 15:715–742, 2001.

[238] SJ Qin and J Yu. Multivariable controller performance monitoring. In *Prep. IFAC ADCHEM 2006*, pages 593–600, Gramado, Brazil, 2006.

[239] L Rabiner and BH Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[240] LR Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[241] A Raich and A Cinar. Multivariate statistical methods for monitoring continuous processes: Assessment of discrimination power of disturbance models and diagnosis of multiple disturbances. *Chemometrics & Intell. Lab. Sys.*, 30:37–48, 1995.

[242] A Raich and A Cinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE J.*, 42(4):995–1009, 1996.

[243] A Raich and A Cinar. Diagnosis of process disturbances by statistical distance and angle measures. *Comput. & Chem. Engg.*, 21(6):661–673, 1997.

[244] JB Rawlings and I Chien. Gage control of sheet and film forming processes. *AIChE J.*, 42(3):753–766, 1996.

[245] R Redner and H Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.*, 26:195–239, 1994.

[246] GC Reinsel. *Elements of Multivariate Time Series Analysis*. Springer-Verlag, New York, NY, 2nd edition, 1997.

[247] R Rengaswamy and V Venkatasubramanian. A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engg. App. of Artificial Intelligence*, 8:35–51, 1995.

[248] RR Rhinehart. A watchdog for controller performance monitoring. In *Proceedings of American Control Conference*, Seattle, WA, 1995.

[249] A Rigopoulos, Y Arkun, and F Kayihan. Full CD profile control of sheet forming processes using adaptive PCA and reduced order MPC design. In *Proceedings of ADCHEM'97*, page 396, 1997.

[250] A Rigopoulos, Y Arkun, and F Kayihan. Identification of full profile disturbance models for sheet forming processes. *AIChE J.*, 43(3):727–739, 1997.

[251] A Rigopoulos, Y Arkun, and F Kayihan. A novel approach to full CD profile control of sheet forming processes using adaptive PCA and reduced order IMC design. *Comput. & Chem. Engg.*, 22(7–8):945–962, 1998.

[252] BD Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, New York, NY, 1996.

[253] JA Romagnoli and A Palazoglu. *Introduction to Proces Control*. CRC Press / Taylor & Francis, Boca Raton, FL, 2005.

[254] JK Romberg, H Choi, and RG Baraniuk. Bayesian tree structured image modeling using wavelet-domain hidden Markov models. *IEEE Trans. on Image Processing*, 10:1056–1068, 2001.

[255] M Rossi and C Scali. A comparison of techniques for automatic detection of stiction: Simulation and application to industrial data. *J. Process Control*, 15:505–514, 2005.

[256] M Rudemo. Empirical choice of histograms and kernel density estimators. *Scand. J. Statistics*, 9:65–78, 1982.

[257] DE Rumelhart and JL McClelland, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, MA, 1986.

[258] GC Runger and FB Alt. Choosing principal components for multivariate statistical process control. *Commun. Statist. – Theory & Methods*, 25(5):909–922, 1996.

[259] GC Runger, TR Willemain, and S Prabhu. Average run lengths for CUSUM control charts applied to residuals. *Commun. Statist. – Theory & Methods*, 24(1):273–282, 1995.

[260] EL Russell, LH Chiang, and RD Braatz. *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Springer-Verlag, London, UK, 2000.

[261] TP Ryan. Some statistical process control methods for autocorrelated data – discussion. *J. Quality Technology*, 23:200–202, 1991.

[262] AA Safavi, J Chen, and JA Romagnoli. Wavelets-based density estimation and application to process monitoring. *AIChE J.*, 43:1227–1241, 1997.

[263] T Sastri. A recursive estimation algorithm for adaptive estimation and parameter change detection of time series models. *J. Op. Res. Soc.*, 37:987–999, 1986.

[264] J Schaefer and A Cinar. Multivariable MPC system performance assessment, monitoring, and diagnosis. *J. Process Control*, 14(2):113–129, 2004.

[265] DW Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley & Sons, New York, NY, 1992.

[266] R Shao, F Jia, EB Martin, and AJ Morris. Wavelets and nonlinear principal components analysis for process monitoring. *Control Engg. Practice*, 7:865–879, 1999.

[267] WA Shewhart. *Economic Control of Quality of Manufactured Product*. Van Nostrand, New York, NY, 1931.

[268] BW Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.

[269] SIMCA-P (Version 11.0), 2006. UMETRICS AB, Umeå, Sweden, (*www.umetrics.com*).

[270] A Singhal and DE Seborg. Pattern matching in historical batch data using PCA. *IEEE Control Systems Magazine*, 22:53–63, 2002.

[271] A Singhal and DE Seborg. Pattern matching in multivariate time series databases using a moving window approach. *Ind. & Engg. Chem. Research*, 41:3822–3838, 2002.

[272] A Singhal and DE Seborg. Effect of data compression on pattern matching in historicals data. *Ind. & Engg. Chem. Research*, 44:3203–3212, 2005.

[273] A Singhal and DE Seborg. Evaluation of pattern matching method for the Tennessee Eastman challenge problem. *J. Process Control*, 16:601–613, 2006.

[274] J Sjoberg, Q Zhang, L Ljung, A Benveniste, B Delyon, P Glorennec, H Hjalmarsson, and A Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691–1721, 1995.

[275] JC Skelton, PE Wellstead, and SR Duncan. Distortion of web profiles by scanned measurements. *Pulp & Paper Canada*, 104(12):T316–319, 2003.

[276] AK Smilde, R Bro, and P Geladi. *Multiway Analysis: Applications in the Chemical Sciences*. John Wiley & Sons, New York, NY, 2004.

[277] P Smyth. Hidden Markov models for fault detection in dynamic systems. *Pattern Recognition*, 27:149–164, 1994.

[278] T Södersrtöm and P Stoica. *System Identification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[279] HW Sorenson and DL Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.

[280] R Srinivasan, C. Wang, WK Ho, and KW Lim. Dynamic principal component analysis based methodology for clustering process states in agile chemical plants. *Ind. & Engg. Chem. Research*, 43:2123–2139, 2004.

[281] N Stanfelj, TE Marlin, and JF MacGregor. Monitoring and diagnosis of process control performance: The single-loop case. *Ind. & Engg. Chem. Research*, 32:301–314, 1993.

[282] CM Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Statistics*, 9:1135–1151, 1981.

[283] CL Stork and BR Kowalski. Distinguishing between process upsets and sensor malfunctions using sensor redundancy. *Chemometrics & Intell. Lab. Sys.*, 46:117–131, 1999.

[284] CL Stork, DJ Veltcamp, and BR Kowalski. Identification of multiple sensor disturbances during process monitoring. *Analytical Chemistry*, 69:5031–5036, 1997.

[285] G Strang and T Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.

[286] W Sun, A Palazoglu, and JA Romagnoli. Detecting abnormal process trends by wavelet-domain hidden Markov models. *AIChE J.*, 749:140–150, 2003.

[287] JA Suykens, TV Gestel, J de Brabanter, B De Moor, and J Vanderwalle. *Least Squares Support Vector Machines*. World Scientific Publishing Co., Singapore, 2002.

[288] V-M Taavitsainen and P Korhonen. Nonlinear data analysis with latent variables. *Chemometrics & Intell. Lab. Sys.*, 14:185–194, 1992.

[289] E Tatara. *An Integrated Knowledge-Based System for Automated System Identification, Monitoring, and Sensor Audit for Multivariate Processes*. Master's thesis, Illinois Institute of Technology, Chicago, IL, 1999.

[290] E Tatara and A Cinar. An intelligent system for multivariate statistical process monitoring and diagnosis. *ISA Trans.*, 41:255–270, 2002.

[291] F Teymour. *The Dynamic Behavior of Free-Radical Solution Polymerization in Continuous Stirred Tank Reactors*. PhD thesis, University of Wisconsin, Madison, 1989.

[292] DJ Thomson. Spectrum estimation and harmonic analysis. *Proceedings of IEEE*, 70:1055–1096, 1982.

[293] NF Thornhill and A Horch. Advances and new directions in plant-wide controller performance assessment. In *Prep. IFAC ADCHEM 2006*, pages 29–36, Gramado, Brazil, 2006.

[294] NF Thornhill, M Oettinger, and P Fedenczuk. Refinery-wide control loop performance assessment. *J. Process Control*, 9:109–124, 1999.

[295] S Thorvaldsen. A tutorial on Markov models based on Mendel's classic experiments. *J. Bioinformatics and Comp. Biology*, 3:1441–1460, 2005.

[296] F Tokatli-Kosebalaban and A Cinar. Fault detection and diagnosis in a food pasteurization process with hidden Markov models. *Canadian J. Chem. Engg.*, 82:1–11, 2004.

[297] H Tong. *Threshold Models in Nonlinear Time Series Analysis*. Springer-Verlag, New York, NY, 1983.

[298] ND Tracy, JC Young, and RL Mason. Multivariate control charts for individual observations. *J. Quality Control*, 24(2):88–95, 1992.

[299] JW Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1970.

[300] ML Tyler and M Morari. Performance monitoring of control systems using likelihood methods. *Automatica*, 32:1145–1162, 1996.

[301] C Undey, S Ertunc, and A Cinar. Online batch/fed-batch process performance monitoring, quality prediction, and variable contribution analysis for diagnosis. *Ind. & Engg. Chem. Research*, 42:4645–4658, 2003.

[302] C Undey, E Tatara, and A Cinar. Real-time batch process supervision by integrated knowledge-based systems and multivariate statistical methods. *Engg. App. Artificial Intelligence*, 16:555–566, 2003.

[303] C Undey, E Tatara, and A Cinar. Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations. *J. Biotechnology*, 108(1):61–77, 2004.

[304] C Undey, E Tatara, BA Williams, G Birol, and A Cinar. A hybrid supervisory knowledge-based system for monitoring penicillin fermentation. In *Proc. American Control Conf.*, volume 6, pages 3944–3948, Chicago, IL, 2000.

[305] E van der Burg and J de Leeuw. Nonlinear canonical correlation. *British J. Math. Statist. Psychol.*, 36:54–80, 1983.

[306] T Van Gestel, J Suykens, G Lanckriet, A Lambrechts, B De Moor, and J Vandewalle. Bayesian framework for least squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis. *Neural Computation*, 15:1115–1148, 2002.

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

[308] V Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, NY, 1995.

[309] SV Vaseghi. *Advanced Signal Processing and Digital Noise Reduction.* John Wiley & Sons, New York, NY, 1996.

[310] A Vasilache, B Dahhou, G Roux, and G Goma. Classification of fermentation process models using recurrent neural networks. *Int. J. Systems Science*, 32(9):1139–1154, 2001.

[311] V Venkatasubramanian and K Chan. A neural network methodology for process fault diagnosis. *AIChE J.*, 35(12):1993–2002, 1989.

[312] V Venkatasubramanian, R Rengaswamy, SN Kavuri, and K Yin. A review of process fault detection and diagnosis Part III: Process history based methods. *Comput. & Chem. Engg.*, 27:327–346, 2003.

[313] M Verhaegen and P Dewilde. Subspace model identification. Part I: The output error state space model identification class of algorithms. *Int. J. Control*, 56:1187–1210, 1992.

[314] M Verhaegen and D Westwick. Identifying MIMO Wiener systems using subspace model identification methods. In *Proc. of 34th Conf. on Decision and Control*, number FP14, 1995.

[315] V Volterra. *Theory of Functionals and Integro-Differential Equations*. Dover, New York, NY, 1959.

[316] X Wang, U Kruger, and GW Irwin. Process monitoring approach using fast moving window PCA.

[317] Z Wang, C Di Massimo, MT Tham, and AJ Morris. Procedure for determining the topology of multilayer feedforward neural networks. *Neural Networks*, 7(2):291–300, 1994.

[318] Z Wang, MT Tham, and AJ Morris. Multilayer feedforward neural networks: A canonical form approximation of nonlinearity. *Int. J. Control*, 56(3):655–672, 1992.

[319] LE Wangen and BR Kowalski. A multiblock partial least squares algorithm for investigating complex chemical systems. *J. Chemometrics*, 3(1):3–20, 1989.

[320] M Weighell, EB Martin, M Bachmann, AJ Morris, and J Friend. Multivariate statistical process control applied to an industrial production facility. In *Proc. of ADCHEM'97*, pages 359–364, 1997.

[321] PJ Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, in Applied Mathematics, 1984.

[322] JA Westerhuis, T Kourti, and JF MacGregor. Analysis of multiblock and hierarchical PCA and PLS models. *J. Chemometrics*, 12:301–321, 1998.

[323] Western Electric Company. *Statistical Quality Control Handbook*. AT&T Technologies, Indianapolis, 1984.

[324] J Weston and C Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, University of London, London, UK, 1998.

[325] JR Whiteley and JF Davis. Qualitative interpretation of sensor patterns. *IEEE Expert*, 8:54–63, 1993.

[326] A Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

[327] BM Wise and NB Gallagher. The process chemometrics approach to process monitoring and fault detection. *J. Process Control*, 6(6):329–348, 1996.

[328] BM Wise, NB Gallagher, R Bro, JM Shaver, W Windig, and RS Koch. *PLS Toolbox 3.5 for use with Matlab®*. Eigenvector Research, Inc., Manson, WA, 2004.

[329] BM Wise, NL Ricker, and DJ Veltkamp. Upset and sensor fault detection in multivariable processes. In *AIChE Annual Meeting, Paper 164b*, San Francisco, CA, 1989.

[330] BM Wise, DJ Veltkamp, NL Ricker, BR Kowalski, SM Barnes, and V Arakali. Application of multivariate statistical process control (MSPC) to the West Valley slurry-fed ceramic melter process. In *Proceedings of Waste Management '91*, pages 169–176, Tucson, AZ, 1991.

[331] H Wold. *Multivariate Analysis*, chapter Estimation of principal components and related models by iterative least squares, pages 391–420. Academic Press, New York, NY, 1966.

[332] S Wold. Cross-validatory estimation of the number of components in factor and principal components analysis. *Technometrics*, 20(4):397–405, 1978.

[333] S Wold. Nonlinear partial least squares modelling: II. Spline inner relation. *Chemometrics & Intell. Lab. Sys.*, 14:71–84, 1992.

[334] S Wold, P Geladi, K Esbensen, and J Ohman. Multi-way principal component and PLS analysis. *J. Chemometrics*, 1:41–56, 1987.

[335] S Wold, S Hellberg, T Lundstedt, M Sjostrom, and H Wold. PLS modeling with latent variables in two or more dimensions. In *Proc. Symp. on PLS Model Building: Theory and Application*, Frankfurt, Germany, Sept. 1987.

[336] S Wold, N Kettaneh-Wold, and B Skagerberg. Nonlinear PLS modeling. *J. Chemometrics*, 7:53–65, 1989.

[337] S Wold, N Kettaneh-Wold, and K Tjessem. Hierarchical multiblock PLS and PC models, for easier model interpretation, and as an alternative to variable selection. *J. Chemometrics*, 10:463–482, 1996.

[338] S Wold, A Ruhe, H Wold, and WJ Dunn. The collinearity problem in linear regression. Partial least squares PLS approach to generalized inverses. *SIAM J. Sci. Stat. Comput.*, 3(5):735–743, 1984.

[339] S Wold, M Sjostrom, and L Eriksson. PLS-regression: A basic tool of chemometrics. *Chemometrics & Intell. Lab. Sys.*, 58:109–130, 2001.

[340] JC Wong, KA McDonald, and A Palazoglu. Classification of process trends based on fuzzified symbolic representation and hidden Markov models. *J. Process Control*, 8:395–408, 1998.

[341] JC Wong, KA McDonald, A Palazoglu, and T Wada. Application of a fuzzy triangular representation and hidden Markov models classification in the detection of abnormal situations in refining processes. In *Proceedings of CONTROL 97*, pages 566–571, Sydney, Australia, 1997.

[342] L Xie, U Kruger, D Lieftucht, T Littler, Q Chen, and S-Q Wang. Statistical monitoring of dynamic multivariate processes – Part 1. Modeling autocorrelation and cross-correlation. *Ind. & Engg. Chem. Research*, 45:1659–1676, 2006.

[343] E Yashchin. Performance of CUSUM control schemes for serially correlated observations. *Technometrics*, 35:37–52, 1993.

[344] S Yoon and JF MacGregor. Principal-component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE J.*, 50(11):2891–2903, 2004.

[345] Y You and M Nikolaou. Dynamic process modeling with recurrent neural networks. *AIChE J.*, 39(10):1654–1667, 1993.

[346] L Zadeh. Fuzzy sets. *Inf. Control*, 8:338–353, 1965.

[347] Y Zhang and MA Henson. A performance measure for constrained model predictive controllers. In *European Control Conference*, Karlsruhe, Germany, 1999.

[348] SJ Zhao, J Zhang, and YM Xu. Monitoring of processes with multiple operating modes through multiple principal components analysis models. *Ind. & Engg. Chem. Research*, 43:7025–7035, 2004.

Index

α error, 10
β error, 10

Akaike's information criterion, 88, 95
Artificial neural networks, 58
  activation function, 59
  autoassociative networks, 63, 79, 193
  back-propagation, 58
  connections, 59
  learning paradigms, 62
    error back-propagation, 62
    reinforcement, 62
    supervised, 62
    unsupervised, 63
  limitations, 59
  multi-layer feedforward networks, 61
  neurons, 59
  recurrent networks, 62
  sigmoid function, 61
  topologies, 61
Autocorrelated data, 22
  parameter change detection, 27
  residuals charts, 26
Autocorrelation coefficient, 24
Autocovariance, 95
Average run length, 17, 19

Basis functions, 116, 119, 262
Beta distribution, 102
Biplots, 100, 179
Box's equation, 104

Canonical variates, 43
  multipass CVSS for sensor auditing, 212
  state-space (CVSS) models, 96
Canonical variates analysis, 43, 89, 100
  Hankel matrix, 95
CD control performance, 271
Classification, 50
  with Fisher's discriminant analysis, 56
  with HMMs, 144, 157
Cluster analysis, 48
Colinearity, 76
Confidence limits, 100
Contribution plots, 46, 100, 174
Control
  linear quadratic Gaussian (LQG), 239
  model predictive, 238
Control charts, *see* Monitoring charts
Control limit
  lower, 12
  of R chart, 15
  of S chart, 16
  of $\bar{x}$ chart, 15
  on SPE, 108
  selection of, 13

upper, 12
warning, 13
Controller performance monitoring, 231
closed-loop potential, 235
CPM using minimum variance control, 233
diagnosis of MPC performance, 242, 244
for model predictive controllers, 238
comprehensive technique, 241
expected performance approach, 240
historical benchmark , 240
LQG-Benchmark, 239
model-based performance measure, 240
frequency-domain method, 237
interactor matrix, 237
minimum variance control, 237
multivariable control systems, 237
single-loop, 233
valve stiction, 233
Correlation function, 23
Correlogram, 24, 79
Cost
function for MPC, 238
of misclassification, 50
Cross-direction (CD), 251
Cross-validation, 40
CSTR, 152, 164
Cumulative sum (CUSUM) charts, 11, 18
one-sided, 18
two-sided, 19

Decomposition
orthogonal, 38, 262
singular value, 39, 262
spectral, 39
Denoising, 127, 150, 193, 264
Discriminant
angular, 186
combined, 183
Euclidian angle, 185
Fisher's, 53
Mahalanobis angle, 185
residual, 182
Mahalanobis angle, 187
score, 182
linear, 53
quadratic, 52
Distance
Euclidian, 48
Mahalanobis, 49
statistical, 49
Distribution
F, 101
χ^2, 103, 108
Beta, 102
chi-squared, 103
Lambda, 214
Normal, 8, 15, 34, 96, 102
Disturbances, 91, 219
discrimination from sensor faults, 195, 220
multiple simultaneous, 189
overlap of means, 190
sensors, 195

Eigenvalues, 39, 262
Eigenvectors, 39, 262
Episode, 136
Estimated
covariance matrix, 108
of residuals, 104
of scores, 101
variance, 102
Exponentially weighted moving average (EWMA) charts, 11, 22

Fault diagnosis, 100

angle-based discriminants, 184
combined distance discriminant, 183
knowledge-based systems, 178
parity relations, 178
residual discriminant, 182
robust, 191
score discriminant, 182
sensor auditing, 203
sensor faults, 204
using contribution plots, 174
using discriminant analysis, 179
using PLS, 204
using statistical methods, 179
using SVM, 191
Faults
actuator, 111, 177
incipient, 203
masking of multiple faults, 190
multiple simultaneous faults, 189
sensor, 111, 170, 195, 223
Feature space, 66
Filter
low-pass, 128
median, 128, 130, 133, 193
robust, 133
Filtering, 127, 136
Final prediction error, 88
Fisher's discriminant analysis, 53
kernel-based, 65
Flatness, 266
Forced circulation evaporator, 246
Fourier transform
definition, 116
discrete, 117
fast, 117
short-time, 117
Functional redundancy, 203
Fuzzification, 137
Fuzzy logic, 137

Gram polynomials, 259

Hidden Markov model (HMM), 138, 141, 149, 166
state variables, 168
states, 169
training, 143
Hidden Markov tree, 147, 157, 162
Hotelling's statistic, *see* Monitoring charts
HTST pasteurization, 109, 167, 177, 207
Hypothesis testing, 9, 12
Type I error, 10, 13, 14
Type II error, 10

Independent component analysis, 43
mixing matrix, 44
process monitoring, 112
separating matrix, 44
sphering matrix, 44
Inner product, 64
Input-output models, 83

k-means clustering, 49
Kernel, 64
Mercer's theorem, 64
Kernel density estimation, 64, 198
Knowledge-based systems (KBS), 178, 204, 214, 238, 246
Kurtosis, 44

Linearization of nonlinear systems, 92
Jacobian matrices, 93

Machine direction (MD), 251
Markov process, 139
Masking, 273
MD control performance, 269
MD/CD decomposition, 253
Mean, 8

Minimum variance control, 233
Mode, 263
Model predictive control, 238
  control horizon, 239
  prediction horizon, 239
  tuning parameters, 242
Model-based control performance, 271
Models
  ARMA, 241
  Box-Jenkins, 86
  first principles, 73
  input-output, 73
    linear, 73
    nonlinear, 74
  linear discrete-time transfer function, 234
  nonlinear, 96
  nonlinear ARMAX, 88
  nonlinear ARX, 89
  nonlinear PCA, 79
  nonlinear PLS, 82
  output error, 87
  regression, 75
  state-space, 89
  subspace state-space, 93
  time series, 83
Monitoring charts
  cumulative sum (CUSUM), 18
  exponentially weighted moving average (EWMA), 22
  for CPM, 243
  moving average (MA), 19
  multivariate, 108
    Q-statistic, 103
    Hotelling's T^2, 101
    score biplots, 100
  Shewhart, 11
    assumptions of, 13
    mean ($\bar{x}$), 14, 15, 17
    range (R), 12, 14
    standard deviation, 12, 15
Moving average (MA) charts, 11, 19
  estimation of S, 19
  process level monitoring, 20
  spread monitoring, 21
Multivariate statistical process monitoring (MSMP), 99
  SPE, 103
  angle-based, 113
  charts, 99
    D, 103
    Q, 103
    SPE, 99, 103, 108, 109
    T^2, 99, 102
  PC scores biplots, 100
  PC scores charts, 101
  PLS scores biplots, 108
  with state variables, 109

NIPALS, 42
Normal operation (NO), 37, 100, 180
Normalized performance index, 269

O-NLPCA, 194, 198
Orthogonal decomposition, 260
Orthogonality, 102
Outliers, 128

Parameter change detection, 27
Partial least squares, 42, 79
  convergence, 81
  inner relations, 81
  multi-block, 113
  multipass PLS for sensor auditing, 204
  nonlinear iterative algorithm (NIPALS), 80
  nonlinear PLS, 82
  outer relations, 80
  residuals matrices, 80, 82
  weight vectors, 80
PCA, 262

pH process, 161
Phase-plane, 267
Population, 8
Prediction error, 83
Prediction error sum of squares (PRESS), 40
Principal components analysis, 37
  consensus, 113
  dynamic, 113
  hierarchical, 113
  loadings, 39
  moving-window, 113
  multi-block, 113
  multiscale, 112
  scores, 263
    matrix, 39
    vector, 258, 263
Projection to latent structures, *see* Partial least squares
Pseudo-random binary sequence, 111

Quadratic discrimination score, 56

Range, 8, 14
Reference set, 101
Regression
  coefficients, 76
  multivariable linear, 76
  nonlinear, 78
  nonlinear PCA, 79
  partial least squares, 79
  principal components, 78
  ridge, 78
  stepwise, 77
  with lagged variables, 79
Residuals charts, 27
  CUSUM charts, 31
  for CPM, 243
Ridge parameter, 78
Run rules, 14

Sample, 8
Scatter
  between-class, 53
  matrix
    between-class, 55
    total, 55
    within-class, 55
  within-class, 53
Sensor
  auditing, 203
  reconstruction, 197, 200, 223
Sheet, 251
Singular value decomposition, 39, 262
Singular values, 95
Slurry-fed ceramic melter (SFCM), 224
Small process shifts, 18
Spectral decomposition, 39
Splines, 82
Spray drier, 30
Squared prediction error (SPE), 103, 107
Standard deviation, 8
State vector, 90
State-space models
  discrete-time, 90
  disturbance, 91
  linear, 90
  linearization of nonlinear models, 92
  nonlinear, 96
  state variables, 89
  subspace, 93
Statistical discrimination, 50
Statistical process control (SPC), 2, 7
Subspace state-space models, 93, 100
  canonical variate realization, 94, 108
  future data window J, 94
  Hankel matrix, 94

N4SID, 94, 108
past data window K, 94
Sum of squares
cumulative prediction (CUMPRESS), 107
prediction (PRESS), 107
Support vector machines (SVM), 66, 191
k-class pattern recognition, 68
decision function, 68
dual solution, 67
proximal, 191

Temporal, 251, 257
Tennessee Eastman industrial challenge problem, 181, 184, 187
Time series models, 83
autoregressive (AR), 83
autoregressive integrated moving average (ARIMA), 83
autoregressive moving average with exogenous inputs (ARMAX), 87
autoregressive with exogenous inputs (ARX), 87
exogenous variables, 83
moving average (MA), 83
NARMAX, 88
nonlinear ARX, 89
Transition matrix, 258
Triangular episodes, 135, 150
Type I error, 10
Type II error, 10

Variables
deviation, 92
predictor, 76
state, 89, 90
Variance inflation factor, 77
Variation
between samples, 12
within samples, 12
Vinyl acetate polymerization, 104, 214

Wavelet filter, 131
coefficient denoising, 133
hard-thresholding, 132, 264
soft-thresholding, 132
Wavelet transform, 264
continuous, 121
discrete, 124
Multiresolution Signal Decomposition, 124
Wavelets, 145, 157
Coiflet, 121
Daubechies, 121
Haar, 120, 162
Morlet, 120
Symlet, 121
Web, 251

OTHER RELATED TITLES OF INTEREST

Batch Fermentation: Modeling: Monitoring, and Control
Ali Cinar, Satish J. Parulekar, Cenk Undey, and Gulnur Birol
ISBN: 0824740343

Engineering Economics and Economic Design for Process Engineers
Thane Brown
ISBN: 0849382122

Instrument Engineers' Handbook, Fourth Edition, Volume One: Process Measurement and Analysis
Béla Lipták
ISBN: 0849310830

Introduction to Process Control
José Romagnoli and Ahmet Palazoglu
ISBN: 0849334969

Materials Processing Handbook
Joanna R. Groza, James F. Shackelford, Enrique J. Lavernia, and Michael T. Powers
ISBN: 0849332168