Other titles published in this Series:

Supervision and Control for Industrial Processes
Björn Sohlberg

Modelling and Simulation of Human Behaviour in System Control
Pietro Carlo Cacciabue

Modelling and Identification in Robotics
Krzysztof Kozlowski

Spacecraft Navigation and Guidance
Maxwell Noton

Robust Estimation and Failure Detection
Rami Mangoubi

Adaptive Internal Model Control
Aniruddha Datta

Price-Based Commitment Decisions in the Electricity Market
Eric Allen and Marija Ilic

Compressor Surge and Rotating Stall: Modeling and Control
Jan Tommy Gravdahl and Olav Egeland

Radiotheraphy Treatment Planning: New System Approaches
Olivier Haas

Feedback Control Theory for Dynamic Traffic Assignment
Pushkin Kachroo and Kaan Özbay

Control and Instrumentation for Wastewater Treatment Plants
Reza Katebi, Michael A. Johnson & Jacqueline Wilkie

Autotuning of PID Controllers
Cheng-Ching Yu

Robust Aeroservoelastic Stability Analysis
Rick Lind & Marty Brenner

Performance Assessment of Control Loops:Theory and Applications
Biao Huang & Sirish L. Shah

Data Mining and Knowledge Discovery for Process Monitoring and Control
Xue Z. Wang

Advances in PID Control
Tan Kok Kiong, Wang Quing-Guo & Hang Chang Chieh with Tore J. Hägglund

Advanced Control with Recurrent High-order Neural Networks: Theory and Industrial Applications
George A. Rovithakis & Manolis A. Christodoulou

Structure and Synthesis of PID Controllers
Aniruddha Datta, Ming-Tzu Ho and Shankar P. Bhattacharyya

Bounded Dynamic Stochastic Systems
Hong Wang

Evan Russell, Leo H. Chiang
and Richard D. Braatz

Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes

With 49 Figures

Springer

Evan L. Russell, PhD
Exxon Production Research Company, PO Box 2189, Room C-328, Houston, TX 77252-2189, USA

Leo H. Chiang, MS
Department of Chemical Engineering, University of Illinois at Urbana-Champaign, 600 S. Matthews Avenue, Urbana, Illinois 61801-3792, USA

Richard D. Braatz, PhD
Department of Chemical Engineering, University of Illinois at Urbana-Champaign, 600 S. Matthews Avenue, Urbana, Illinois 61801-3792, USA

ISSN 1430-9491

ISBN 978-1-4471-1133-7

British Library Cataloguing in Publication Data
Russell, Evan
Data-driven methods for fault detection and diagnosis in chemical processes. - (Advances in industrial control)
1.Chemical process control
I.Title II.Chiang, Leo H. III.Braatz, Richard D.
660.2'815
ISBN 978-1-4471-1133-7 ISBN 978-1-4471-0409-4 (eBook)
DOI 10.1007/978-1-4471-0409-4

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

Originally published by Springer-Verlag London Berlin Heidelberg in 2000
Softcover reprint of the hardcover 1st edition 2000

Typesetting: Camera ready by authors

69/3830-543210 Printed on acid-free paper SPIN 10746315

Advances in Industrial Control

Professor Dr -Ing M. Thoma
Institut für Regelungstechnik
Universität Hannover
Appelstr. 11
30167 Hannover
Germany

Professor H. Kimura
Department of Mathematical Engineering and Information Physics
Faculty of Engineering
The University of Tokyo
7-3-1 Hongo
Bunkyo Ku
Tokyo 113
Japan

Professor A.J. Laub
College of Engineering - Dean's Office
University of California
One Shields Avenue
Davis
California 95616-5294
United States of America

Professor J.B. Moore
Department of Systems Engineering
The Australian National University
Research School of Physical Sciences
GPO Box 4
Canberra
ACT 2601
Australia

Dr M.K. Masten
Texas Instruments
2309 Northcrest
Plano
TX 75075
United States of America

Professor Ton Backx
AspenTech Europe B.V.
De Waal 32
NL-5684 PH Best
The Netherlands

SERIES EDITORS' FOREWORD

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies..., new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

A key objective in industrial plant is to maintain continuous operation and produce outputs meeting the desired specifications. Over recent decades, complex plant has become more and more instrumented in an attempt to improve process monitoring. The consequence has been that process plant data is widely and readily available. The problem arising is what to do with this data. A growing band of experts and engineers have been looking at this problem. The outcome has been a growth in black-box and grey-box data-based methods for modelling and related activities. Among such activities are those of fault detection and diagnosis methods, which have enjoyed considerable expansion over recent years.

This timely *Advances in Industrial Control* monograph by E.L. Russell, L.H. Chiang and R.D. Braatz contributes to this activity in using plant data for fault detection and diagnosis. The monograph presents an application-orientated text on methods based on a statistical framework. In fact the presentation of this approach is comprehensive and systematic; the statistical background, the derivation of the methods and a detailed look at applications are all covered. A valuable feature of the monograph is a final chapter reviewing current alternative methods for the fault-detection problem.

It is useful to note that this text by Russell *et al.* complements a recent *Advances in Industrial Control* monograph by X.Z. Wang entitled Data Mining and Knowledge Discovery for Process Monitoring and Control (ISBN 1-852333-137-2). Both books have a strong applications flavour and should be of special interest to the engineering practitioner in the process and chemical engineering domains.

M.J. Grimble and M.A. Johnson
Industrial Control Centre
Glasgow, Scotland, UK

PREFACE

Modern chemical plants are large scale, highly complex, and operate with a large number of variables under closed loop control. Early and accurate fault detection and diagnosis for these plants can minimize downtime, increase the safety of plant operations, and reduce manufacturing costs. Chemical processes are becoming more heavily instrumented, resulting in large quantities of data becoming available for use in detecting and diagnosing faults. Univariate control charts (e.g., Shewhart charts) have a limited ability to detect and diagnose faults in such processes due to large correlations in the process data. This has led to a surge of academic and industrial effort concentrated towards developing more effective process monitoring methods.

While techniques based on first-principles models have been around for more than two decades, their contribution to industrial practice has not been pervasive due to the huge cost and time required to develop a sufficiently accurate process model for a complex chemical plant. The process monitoring techniques that have dominated the literature for the past decade and have been most effective in practice are based on models constructed almost entirely from process data. The purpose of this book is to bring these data-driven process monitoring techniques to practicing engineers, and to engineering undergraduate or graduate students. Besides describing the state-of-the-art on methods based on chemometrics, pattern classification, and system identification, the methods are compared through application to the Tennessee Eastman Chemical plant simulator. This gives the readers an understanding of the strengths and weaknesses of various approaches, as well as some realistic homework problems.

Although much effort has been devoted to process monitoring by both academics and industrially employed engineers, books on this subject are very limited. The most closely related to this book is "Fault Detection and Diagnosis in Chemical and Petrochemical Processes," written by David M. Himmelblau and published in 1978. It was a ground-breaking book, but is now dated by two decades of significant advances in process monitoring theory and practice. Also, beyond providing some basic background on statistics and univariate process control charts, Himmelblau's book focused primarily on the use of detailed mathematical models. In contrast, as discussed above, this text focuses almost entirely on data-driven processing monitoring techniques, as these are the most promising in process applications.

The goal of the book is to present the theoretical background and practical techniques for data-driven process monitoring. The intended audience is engineering students and practicing engineers. The book is appropriate for a first-year graduate or advanced undergraduate course in process monitoring. As the most effective method for learning the techniques is by applying them, the Tennessee Eastman Chemical plant simulator used in this text has been made available at http://brahms.scs.uiuc.edu. Readers are encouraged to collect process data from the simulator, and then apply a range of process monitoring techniques to detect, isolate, and diagnose various faults. The process monitoring techniques can be implemented using commercial software packages such as the MATLAB PLS Toolbox and ADAPTx.

The authors thank International Paper, DuPont, and the National Center for Supercomputing Applications for funding over the past three years this book was being written.

Urbana, Illinois E.L.R., L.H.C., R.D.B

CONTENTS

Part I. INTRODUCTION

1. Introduction 3
- 1.1 Process Monitoring Procedures 4
- 1.2 Process Monitoring Measures 5
- 1.3 Data-driven Process Monitoring Methods 7
- 1.4 Book Organization 9

Part II. BACKGROUND

2. Multivariate Statistics 13
- 2.1 Introduction 13
- 2.2 Data Pretreatment 14
- 2.3 Univariate Statistical Monitoring 15
- 2.4 T^2 Statistic 19
- 2.5 T^2 Statistic Thresholds 20
- 2.6 Data Requirements 22
- 2.7 Homework Problems 23

3. Pattern Classification 25
- 3.1 Introduction 25
- 3.2 Discriminant Analysis 26
- 3.3 Feature Extraction 28
- 3.4 Homework Problems 29

Part III. METHODS

4. Principal Component Analysis 33
- 4.1 Introduction 33
- 4.2 Principal Component Analysis 34
- 4.3 Reduction Order 37
- 4.4 Fault Detection 39
- 4.5 Fault Identification 42

4.6 Fault Diagnosis 45
4.7 Dynamic PCA 49
4.8 Other PCA-based Methods 51
4.9 Homework Problems 52

5. Fisher Discriminant Analysis 53
5.1 Introduction 53
5.2 Fisher Discriminant Analysis 54
5.3 Reduction Order 56
5.4 Fault Detection and Diagnosis 58
5.5 Comparison of PCA and FDA 59
5.6 Dynamic FDA 64
5.7 Homework Problems 65

6. Partial Least Squares 67
6.1 Introduction 67
6.2 PLS Algorithms 68
6.3 Reduction Order and PLS Prediction 73
6.4 Fault Detection, Identification, and Diagnosis 74
6.5 Comparison of PCA and PLS 75
6.6 Other PLS Methods 77
6.7 Homework Problems 79

7. Canonical Variate Analysis 81
7.1 Introduction 81
7.2 CVA Theorem 83
7.3 CVA Algorithm 85
7.4 State Space Model and System Identifiability 87
7.5 Lag Order Selection and Computation 88
7.6 State Order Selection and Akaike's Information Criterion 90
7.7 Subspace Algorithm Interpretations 91
7.8 Process Monitoring Statistics 93
7.9 Homework Problems 94

Part IV. APPLICATION

8. Tennessee Eastman Process 99
8.1 Introduction 99
8.2 Process Flowsheet 100
8.3 Process Variables 100
8.4 Process Faults 100
8.5 Simulation Program 103
8.6 Control Structure 105
8.7 Homework Problems 105

9. Application Description 109
9.1 Introduction 109
9.2 Data Sets 109
9.3 Sampling Interval 110
9.4 Sample Size 111
9.5 Lag and Order Selection 113
9.6 Fault Detection 114
9.7 Fault Identification 115
9.8 Fault Diagnosis 115

10. Results and Discussion 117
10.1 Introduction 117
10.2 Case Study on Fault 1 117
10.3 Case Study on Fault 4 120
10.4 Case Study on Fault 5 125
10.5 Case Study on Fault 11 127
10.6 Fault Detection 129
10.7 Fault Identification 138
10.8 Fault Diagnosis 142
10.9 Homework Problems 162

Part V. OTHER APPROACHES

11. Overview of Analytical and Knowledge-based Approaches 169
11.1 Parameter and State Estimation 169
11.2 Analytical Redundancy 170
11.3 Causal Analysis and Expert Systems 171
11.4 Pattern Recognition 173
11.5 Combinations of Various Techniques 174

References 175

Index 189

Part I

INTRODUCTION

CHAPTER 1
INTRODUCTION

In the chemical and other related industries, there has been a large push to produce higher quality products, to reduce product rejection rates, and to satisfy increasingly stringent safety and environmental regulations. Process operations that were at one time considered acceptable are no longer adequate. To meet the higher standards, modern chemical processes contain a large number of variables operating under closed loop control. The **standard process controllers** (PID controllers, model predictive controllers, *etc.*) are designed to maintain satisfactory operations by compensating for the effects of disturbances and changes occurring in the process. While these controllers can compensate for many types of **disturbances**, there are changes in the process which the controllers cannot handle adequately. These changes are called **faults**. More precisely, a fault is defined as an unpermitted deviation of at least one characteristic property or variable of the system [95].

The types of faults occurring in chemical processes include **process parameter changes**, **disturbance parameter changes**, **actuator problems**, and **sensor problems** [109]. Catalyst poisoning and heat exchanger fouling are examples of process parameter changes. A disturbance parameter change can be an extreme change in the concentration of a process feed stream or in the ambient temperature. An example of an actuator problem is a sticking valve, and a sensor producing biased measurements is an example of a sensor problem. To ensure that the process operations satisfy the performance specifications, the faults in the process need to be detected, diagnosed, and removed. These tasks are associated with **process monitoring**. **Statistical process control** (SPC) addresses the same issues as process monitoring, but to avoid confusion with standard process control, the methods mentioned in this text will be referred to as **process monitoring methods**.

The goal of process monitoring is to ensure the success of the planned operations by providing information recognizing and indicating anomalies of the behavior. The information not only keeps the plant operator and maintenance personnel better informed of the status of the process, but also assists them to make appropriate remedial actions to remove the abnormal behavior from the process. As a result of proper process monitoring, downtime is minimized, safety of plant operations is improved, and manufacturing costs are reduced. As chemical processes have become more highly integrated and complex, the

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

faults occurring in modern processes present monitoring challenges that are not readily addressed using univariate control charts (e.g., Shewhart charts, see Section 2.3). The weaknesses of univariate control charts for detecting faults in multivariate processes have led to a surge of research literature concentrated on developing better methods for process monitoring. This growth of research activity can also be explained by the fact that chemical processes are becoming more heavily instrumented, resulting in large quantities of data becoming available for use in process monitoring, and that modern computers are becoming more powerful. The availability of data collected during various operating and fault conditions is essential to process monitoring. The storage capacity and computational speed of modern computers enable process monitoring algorithms to be computed when applied to large quantities of data.

1.1 Process Monitoring Procedures

The four procedures associated with process monitoring are: **fault detection**, **fault identification**, **fault diagnosis**, and **process recovery**. There appears to be no standard terminology for these procedures as the terminology varies across disciplines; the terminology given by Raich and Cinar [189] is adopted here. **Fault detection** is determining whether a fault has occurred. Early detection may provide invaluable warning on emerging problems, with appropriate actions taken to avoid serious process upsets. **Fault identification** is identifying the observation variables most relevant to diagnosing the fault. The purpose of this procedure is to focus the plant operator's and engineer's attention on the subsystems most pertinent to the diagnosis of the fault, so that the effect of the fault can be eliminated in a more efficient manner. **Fault diagnosis** is determining which fault occurred, in other words, determining the cause of the observed out-of-control status. Isermann [94] more specifically defines fault diagnosis as determining the type, location, magnitude, and time of the fault. The fault diagnosis procedure is essential to the counteraction or elimination of the fault. **Process recovery**, also called **intervention**, is removing the effect of the fault, and it is the procedure needed to close the **process monitoring loop** (see Figure 1.1). Whenever a fault is detected, the fault identification, fault diagnosis, and process recovery procedures are employed in the respective sequence; otherwise, only the fault detection procedure is repeated.

While all four procedures may be implemented in a process monitoring scheme, this is not always necessary. For example, a fault may be diagnosed (fault diagnosis) without identifying the variables immediately affected by the fault (fault identification). Additionally, it is not necessary to automate all four procedures. For instance, an automated fault identification procedure may be used to assist the plant operators and engineers to diagnose the fault (fault diagnosis) and recover normal operation. Often the goal of process

monitoring is to efficiently incorporate the plant operators and engineers into the process monitoring loop rather than to automate the monitoring scheme entirely.

After a fault occurs, the in-control operations can often be recovered by reconfiguring the process, repairing the process, or retuning the controllers. Once a fault has been properly diagnosed, the optimal approach to counteract the fault may not be obvious. A feasible approach may be to retune the standard process controllers. Several methods have been developed to evaluate controller performance [33, 72, 109, 190, 201, 212], and these can be used to determine which controllers in the process need to be retuned to restore satisfactory performance. In the case of a sensor problem, a sensor reconstruction technique can be applied to the process to restore in-control operations [43]. Even though process recovery is an important and necessary component of the process monitoring loop, process recovery is not the focus of this book.

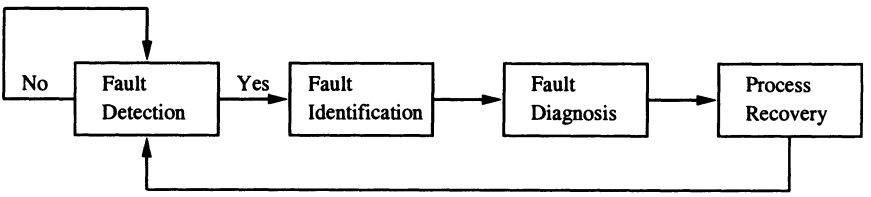

Fig. 1.1. A schemata of the process monitoring loop

1.2 Process Monitoring Measures

A typical process monitoring scheme contains one or more measures, based on developments from statistical theory, pattern classification theory, information theory, and/or systems theory. These measures are calculated directly from the process data, which in some way represent the state or behavior of the process. The idea is to convert the large amount of on-line data collected from the process into a few meaningful measures, and thereby assist the operators in determining the status of the operations and if necessary in diagnosing the faults. For fault detection, limits may be placed on some of the measures, and a fault is detected whenever one of the measures is evaluated outside the limits. In this way, the measures are able to define the in-control process behavior and accordingly the out-of-control status. By developing measures that accurately characterize the behavior of each observation variable, the measure value of one variable can be compared against the measure

values for other variables to determine the variable most affected by the fault. Faults can also be diagnosed by developing and comparing measures that accurately represent the different faults of the process.

The goal of process monitoring is to develop measures that are maximally **sensitive** and **robust** to *all* faults. Faults are manifested in several ways, however, and it is highly unlikely that all faults occurring in a process can be effectively detected and diagnosed with only a few measures. Since each measure characterizes a fault in a different manner, one measure will be more sensitive to certain faults and less sensitive to other faults relative to other measures. This motivates using multiple process monitoring measures, with the proficiency of each measure determined for the particular process and the possible faults at hand.

The measures of a process monitoring scheme are mainly derived based on three approaches; namely, **data-driven**, **analytical**, and **knowledge-based**. The data-driven measures are derived directly from process data. In contrast to the data-driven approach, the analytical approach uses mathematical models often constructed from first-principles while the knowledge-based approach uses qualitative models. The analytical approach is applicable to **information rich** systems, where satisfactory models and enough sensors are available, while the knowledge-based approach is better applied to **information poor** systems, where few sensors or poor models are available [54]. Examples of the analytical approach include parameter and state estimation (see Section 11.1), and residual-based methods (see Section 11.2). Examples of the knowledge-based approach include cause-effect graphs and expert systems (see Section 11.3).

Analytical and knowledge-based approaches have been studied extensively in the literature [53, 78], and several surveys are available [53, 91, 227]. An overview of analytical and knowledge-based approaches is provided in Chapter 11. The fact that analytical measures require accurate detailed models to be effective [40, 96, 238] has motivated a large amount of effort devoted to developing analytical measures that are more robust to model uncertainties [27, 47, 54, 138]. It has been shown that when uncertain models are used, control performance must be traded off against diagnostic performance [167] and the fault detection and control schemes should be designed together [211]. Fundamental issues associated with the identification and control of uncertain systems [18, 17, 48, 143, 193, 191, 194, 218, 217] further complicate the design of robust fault detection and diagnosis measures.

Most applications of the analytical and knowledge-based measures have been to systems with a relatively small number of inputs, outputs, and states [44, 93, 97, 109, 124]. It is difficult to apply the analytical approach to **large scale systems** (*i.e.*, systems containing a large number of inputs, outputs, and/or states) because it requires accurate detailed models in order to be effective [40, 96, 238]. Accurate models for large scale systems are difficult to obtain given all the crosscouplings associated with a multivariable system

[93]. It is challenging to apply the knowledge-based approach to large scale systems because constructing the fault models demands a large amount of effort [238] and requires skills beyond those of the typical engineer [5].

Because accurate detailed models are difficult to develop, most of the process monitoring methods applied to industrial processes are based on data-driven measures. The data-driven monitoring methods use the process data collected during normal operating conditions to develop the measures for detecting and identifying faults, and the data collected during specific faults to develop the measures for diagnosing faults. Because these methods are data-driven, the proficiency of these methods is highly dependent on the *quantity* and *quality* of the process data. While a large quantity of data is available from most processes, only a small portion typically is useful, *i.e.*, where it can be determined with confidence that the data were not somehow corrupted and no unknown faults occurred in the process. This book focuses on data-driven measures, more specifically, on extracting information useful for process monitoring from the data collected from the process.

1.3 Data-driven Process Monitoring Methods

Data-driven process monitoring statistics are based on a handful of methods. These methods not only differ in their objectives, but also in the number of parameters that needs to be estimated to develop the appropriate statistics. The proficiency of the process monitoring methods depends on their objectives and the number of required independent parameters, and these aspects along with the advantages and disadvantages of various methods are discussed in this book.

Traditional monitoring methods consisted of **limit sensing** and **discrepancy detection**. Limit sensing raises an alarm when observations cross predefined thresholds, and has been applied traditionally because it is easy to implement and understand. Limit sensing, however, lacks sensitivity to some process upsets because it ignores interactions between the process variables for the various sensors [40, 94]. Discrepancy detection raises an alarm by comparing simulated to actual observed values. Discrepancy detection highly depends on model accuracy, and model inaccuracies are unavoidable in practice. Since it is difficult to distinguish genuine faults from errors in the model, discrepancy detection can lack robustness [40]. As discussed in Section 1.2, robust discrepancy detection statistics have been studied, however, effective statistics are difficult to obtain, especially for large scale systems.

Limit sensing determines thresholds for each observation variable without using any information from the other variables, and in this way is identical to the univariate statistical techniques discussed in Section 2.3. These methods ignore the correlations among the observation variables (**spacial correlations**) and the correlations among measurements of the same variable taken at different times (**serial correlations**). (Note that spacial correlations also

refer to correlations between different measurements taken at essentially the same physical location.) Process data are spacially correlated because there is often a large number of sensor readings taken throughout the process and the variability of the process variables is restricted to a lower dimension (for example, due to phase equilibria or conservation laws, such as the material and energy balances) [42]. Also, process data are serially correlated because the sampling intervals are relatively small and the standard process controllers are unable to remove all the systematic trends due to inertial components, such as tanks, reactors, and recycle streams. Because limit sensing does not take into account the spacial correlations, it lacks sensitivity to many faults occurring in chemical processes [98, 99], and because limit sensing also ignores the serial correlations, it lacks robustness [73].

The need to handle spacial correlations has led to the development and employment of process monitoring statistics based on **Principal Component Analysis** (PCA) for monitoring chemical processes. PCA is a dimensionality reduction technique for process monitoring which has been heavily studied and applied to chemical processes over the past decade. PCA is an *optimal* dimensionality reduction technique in terms of *capturing the variance* of the data, and it accounts for correlations among variables [98, 99]. The lower dimensional representations of the data produced by PCA can improve the proficiency of detecting and diagnosing faults using multivariate statistics. The structure abstracted by PCA can be useful in identifying either the variables responsible for the fault and/or the variables most affected by the fault. In cases where most of the information in the data can be captured in only two or three dimensions, which can be true for some processes [144], the dominant process variability can be visualized with a single plot (for example, see Figure 4.2). Irrespective of how many dimensions are required in the lower dimensional space, other plots (e.g., T^2 and Q charts) can be used which look similar to univariate control charts but are based on multivariate statistics. These control charts can help the operators and engineers to interpret significant trends in the process data [120].

Fisher Discriminant Analysis (FDA) is a dimensionality reduction technique developed and studied within the **pattern classification** community [41]. FDA determines the portion of the observation space that is most effective in *discriminating amongst several data classes.* Discriminant analysis is applied to this portion of the observation space for fault diagnosis. The dimensionality reduction technique is applied to the data in *all* the classes simultaneously. Thus, all fault class information is utilized when the discriminant function is evaluated for each class and better fault diagnosis performance is expected. The theoretical developments for FDA suggest that it should be more effective than PCA for diagnosing faults.

Partial Least Squares (PLS) are data decomposition methods for *maximizing covariance* between predictor (independent) block and predicted (dependent) block for each component. PLS attempts to find loading and score

vectors that are correlated with the predicted block X while describing a large amount of the variation in the predictor block Y [228]. A popular application of PLS is to select X to contain sensor data and Y to contain only product quality data [144]. Similar to PCA, such inferential models (also known as soft sensors) can be used for detecting, identifying, and diagnosing faults [144, 181, 182]. Another application of PLS primarily focusing on fault diagnosis is to define Y as class membership [26]. This PLS method is known as discriminant Partial Least Squares.

The process monitoring statistics based on PCA, PLS, and FDA can be extended to include serial correlations by augmenting the data collected at a particular time instant to the data collected during several of the previous consecutive sampling intervals. This is an *ad hoc* procedure to include serial correlations or process dynamics into the dimensionality reduction methods. An alternative method to address serial correlations is to average the measurements over many data points (this method has the similar philosophy of CUSUM and EWMA charts, see Section 2.3 for a brief discussion). Another simple approach is to use a larger sampling interval. However, these approaches do not utilize the useful developments made in system identification theory for quantifying serial correlation. A class of system identification methods that produces state variables *directly* from the data are called **subspace algorithms**. The subspace algorithm based on **Canonical Variate Analysis** (CVA) is particularly attractive because of its close relationship to PCA, FDA, and PLS. These relationships motivate the deviation of CVA-based statistics for fault detection, identification, and diagnosis that take serial correlations into account.

1.4 Book Organization

Modern industrial processes, whether an entire chemical plant or a single paper machine, are large scale systems. With the heavy instrumentation typical of modern processes, large scale processes produce an exceptionally large amount of data. Even though much information is available from these processes, it is beyond the capabilities of an operator or engineer to effectively assess process operations simply from observing the data. By computing some meaningful statistics for the process operators and engineers, process monitoring scheme for a large scale system can be improved significantly.

A good process monitoring scheme employs multiple statistics or methods for fault detection, identification, and diagnosis [40]. To maximize the efficiency of employing multiple statistics, however, the confidence of each statistic needs to be assessed for the different operating conditions and types of faults. The effectiveness of various process monitoring statistics has been investigated on simulations of processes [125, 133, 189]. After introducing the data-driven process monitoring techniques, this book illustrates and com-

pares them on a simulation of a realistic large scale process, the **Tennessee Eastman process** [39].

The book is organized into five parts. Part I (this chapter) is an introduction to process monitoring approaches. Part II provides the background necessary to understand the data-driven process monitoring methods in Part III. Chapter 2 provides an introduction to multivariate statistics, and Chapter 3 covers pattern classification. The process monitoring methods described in Part III are PCA, FDA, PLS, and CVA. The methods as described in the literature are extended in cases where the process monitoring statistics are incomplete or inadequate. Part IV describes the application of the process monitoring methods to the Tennessee Eastman process. The Tennessee Eastman process is described in Chapter 8, while Chapter 9 states how the methods are applied to the Tennessee Eastman problem. The results of the methods applied to the simulated data are discussed in Chapter 10. Part V provides an overview of analytical and knowledge-based approaches.

Part II

BACKGROUND

CHAPTER 2
MULTIVARIATE STATISTICS

2.1 Introduction

The effectiveness of the data-driven measures depends on the characterization of the process data variations. There are two types of variations for process data: **common cause** and **special cause** [171]. The common cause variations are those due entirely to random noise (e.g., associated with sensor readings), whereas special cause variations account for all the data variations not attributed to common cause. Standard process control strategies may be able to remove most of the special cause variations, but these strategies are unable to remove the common cause variations, which are inherent to process data. Since variations in the process data are inevitable, statistical theory plays a large role in most process monitoring schemes.

The application of statistical theory to monitor processes relies on the assumption that the characteristics of the data variations are relatively unchanged unless a fault occurs in the system. By the definition of a fault as an abnormal process condition (see Chapter 1), this is a reasonable assumption. It implies that the properties of the data variations, such as the mean and variance, are repeatable for the same operating conditions, although the actual values of the data may not be very predictable. The repeatability of the statistical properties allows thresholds for certain measures, effectively defining the out-of-control status, to be determined automatically. This is an important step to automating a process monitoring scheme.

The purpose of this chapter is to illustrate how to use statistical methods for monitoring processes, in particular methods using the multivariate T^2 statistic. This chapter begins in Section 2.2 by describing the data pretreatment procedure, which is typically performed before determining the statistical parameters (mean, covariance, *etc.*) for the data. The traditional approach to statistical process monitoring using univariate statistics is discussed in Section 2.3. Then in Section 2.4, the T^2 statistic is described along with its advantages over univariate statistics for process monitoring. It is shown in Section 2.5 how to apply the T^2 statistic with statistically-derived thresholds, in order to automate the fault detection procedure and to remove outliers from the training data. In Section 2.6, the applicability of the T^2 statistic is determined in terms of the amount of data available to calculate the statistical parameters.

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

2.2 Data Pretreatment

To effectively extract the information in the data relevant to process monitoring, it is often necessary to pretreat the data in the training set. The **training set** contains off-line data available for analysis prior to the on-line implementation of the process monitoring scheme and is used to develop the measures representing the in-control operations and the different faults. The pretreatment procedures consist of three tasks: **removing variables**, **autoscaling**, and **removing outliers**.

The data in the training set may contain variables that have no information relevant to monitoring the process, and these variables should be removed before further analysis. For instance, it may be known *a priori* that certain variables exhibit extremely large measurement errors, such as those due to improper sensor calibrations, or some of the variables may be physically separate from the portion of the process that is being monitored. In these instances, the proficiency of the process monitoring method can be improved by removing the inappropriate variables.

Process data often need to be scaled to avoid particular variables dominating the process monitoring method, especially those methods based on dimensionality reduction techniques, such as PCA and FDA. For example, when performing an unscaled dimensionality reduction procedure on temperature measurements varying between 300K and 320K and concentration measurements varying between 0.4 and 0.5, the temperature measurements would dominate even though the temperature measurements may be no more important than the concentration measurements for monitoring the process.

Autoscaling standardizes the process variables in a way that ensures each variable is given equal weight before the application of the process monitoring method. It consists of two steps. The first step is to subtract each variable by its sample mean because the objective is to capture the variation of the data from the mean. The second step is to divide each variable of the mean-centered data by its standard deviation. This step scales each variable to unit variance, ensuring that the process variables with high variances do not dominate. When autoscaling is applied to new process data, the mean to be subtracted and the standard deviation to be divided are taken from the training set.

Outliers are isolated measurement values that are erroneous. These values may significantly influence the estimation of statistical parameters and other parameters related to a given measure. Removing the outliers from the training set can significantly improve the estimation of the parameters and should be an essential step when pretreating the data [178]. Obvious outliers can be removed by plotting and visually inspecting the data for outlying points. More rigorous methods based on statistical thresholds can be employed for removing outliers, and a method for doing this using the T^2 statistic is discussed in Section 2.5. For simplicity of presentation only, it is

assumed in the remainder of this book that the data has been pretreated, unless otherwise stated.

2.3 Univariate Statistical Monitoring

A univariate statistical approach to limit sensing can be used to determine the thresholds for each **observation variable** (a process variable observed through a sensor reading), where these thresholds define the boundary for in-control operations and a violation of these limits with on-line data would indicate a fault. This approach is typically employed using a **Shewhart chart** [159, 37, 7] (see Figure 2.1) and has been referred to as **limit sensing** [40] and **limit value checking** [94]. The values of the upper and lower control limits on the Shewhart chart are critical to minimizing the rate of **false alarms** and the rate of **missed detections**. A **false alarm** is an indication of a fault, when in actuality a fault has not occurred; a **missed detection** is no indication of a fault, though a fault has occurred. For fault detection, there is an inherent tradeoff between minimizing the false alarm and missed detection rates. Tight threshold limits for an observation variable result in a high false alarm and low missed detection rate, while limits which are too spread apart result in a low false alarm and a high missed detection rate.

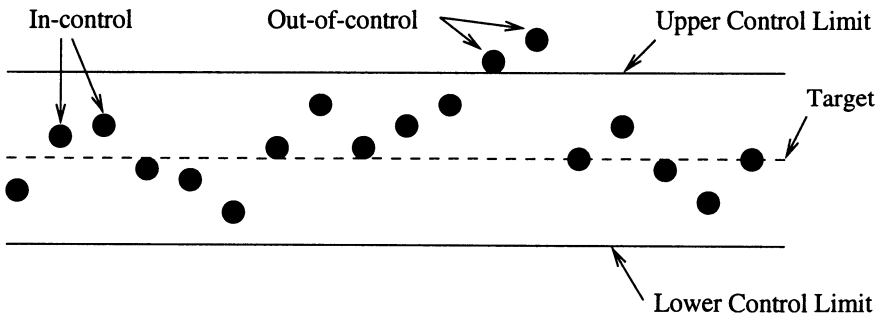

Fig. 2.1. An illustration of the Shewhart chart. The black dots are observations.

Given certain threshold values, statistical hypothesis theory can be applied to predict the false alarm and missed detection rates based on the statistics of the data in the training sets. Consider the case where there can potentially be a single fault i (the more general case of multiple fault classes will be treated thoroughly in the next chapter). Let ω represents the *event* of an in-control operation and ω_i represents the *event* of a specific fault, i. Consider a single observation x with the null hypothesis (assign x as ω) and

the alternative hypothesis (assign x as ω_i), the false alarm rate is equal to the type I error, and the missed detection rate for fault i is equal to the type II error [159]. This is illustrated graphically in Figure 2.2.

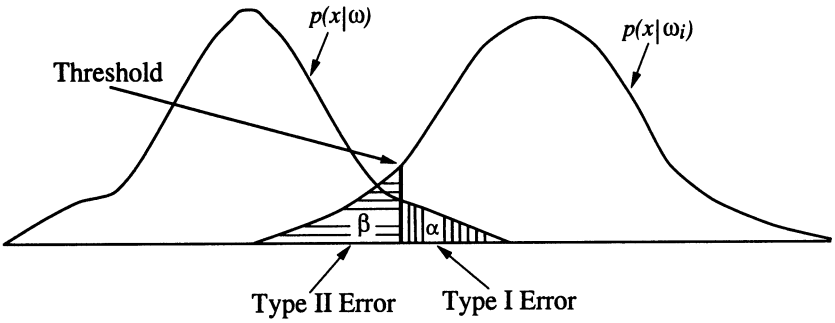

Fig. 2.2. The type I and type II error regions for the null hypothesis (assign x as ω) and the alternative hypothesis (assign x as ω_i). The probability density function for x conditioned on ω is $p(x|\omega)$; the probability density function for x conditioned on ω_i is $p(x|\omega_i)$. The probability of a type I error is α and the probability of a type II error is β. Using Bayesian decision theory [41], these notions can be generalized to include *a priori* probabilities of ω and ω_i.

Increasing the threshold (shifting the vertical line to the right in Figure 2.2) decreases the false alarm rate but increases the missed detection rate. Attempts to lower the false alarm rate are usually accompanied with an increase in the missed detection rate, with the only ways to get around this tradeoff being to collect more data, or to reduce the normal process variability (e.g., through installation of sensors of higher precision). The value of the type I error, also called the **level of significance** α, specifies the degree of tradeoff between the false alarm rate and the missed detection rate.

As a specific example, assume for the null hypothesis that any deviations of the process variable x from a desired value μ are due to inherent measurement and process variability described by a normal distribution with standard deviation σ:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{2.1}$$

The alternative hypothesis is that $x \neq \mu$. Assuming that the null hypothesis is true, the probabilities that x is in certain intervals are

$$\Pr\{x < (\mu - c_{\alpha/2}\sigma)\} = \alpha/2 \tag{2.2}$$

$$\Pr\{x > (\mu + c_{\alpha/2}\sigma)\} = \alpha/2 \tag{2.3}$$

$$\Pr\{(\mu - c_{\alpha/2}\sigma) \leq x \leq (\mu + c_{\alpha/2}\sigma)\} = 1 - \alpha \tag{2.4}$$

where $c_{\alpha/2}$ is the standard normal deviate corresponding to the $(1 - \alpha/2)$ percentile. The standard normal deviate is calculated using the cumulative standard normal distribution [81]; the standard normal deviates corresponding to some common α values are listed in Table 2.1.

Table 2.1. Some typical standard normal deviate values

$\alpha/2$	$c_{\alpha/2}$
0.00135	3.00
0.0025	2.81
0.005	2.58
0.01	2.33
0.025	1.96

The lower and upper thresholds for the process variable x are $\mu - c_{\alpha/2}\sigma$ and $\mu + c_{\alpha/2}\sigma$, respectively. Figure 2.3 illustrates the application of Shewhart chart to monitor the Mooney viscosity of an industrial elastomer [171]. The desired value μ is 50.0; a standard deviation value of $\sigma = 0.5$ is known to characterize the intrinsic variability associated with the sampling procedure. Since all the data points fall inside the upper and lower control limit lines corresponding to $c_{\alpha/2} = 3.0$, the process is said to be "in control".

As long as the sample mean and standard deviation of the training set accurately represent the true statistics of the process, the thresholds using (2.2) and (2.3) should result in a false alarm rate equal to α when applied to on-line data. If 20,000 data points were collected during "in control" operation defined by $c_{\alpha/2} = 3.0$, 27 data points would be expected to fall above the upper control limit, while 27 data points would be expected to fall below the lower control limit. Some typical α values for fault detection are 0.005, 0.01, and 0.05. It has been suggested that even if x does not follow a normal distribution, the limits derived from (2.2) and (2.3) are effective as long as the data in the training set are an accurate representation of the variations during normal operations [113].

Process monitoring schemes based on Shewhart charts may not provide adequate false alarm and missed detection rates. These rates can be improved by employing measures that incorporate observations from multiple consecutive time instances, such as the **cumulative sum** (CUSUM) and **exponentially-weighted moving average** (EWMA) **charts** [159, 171].

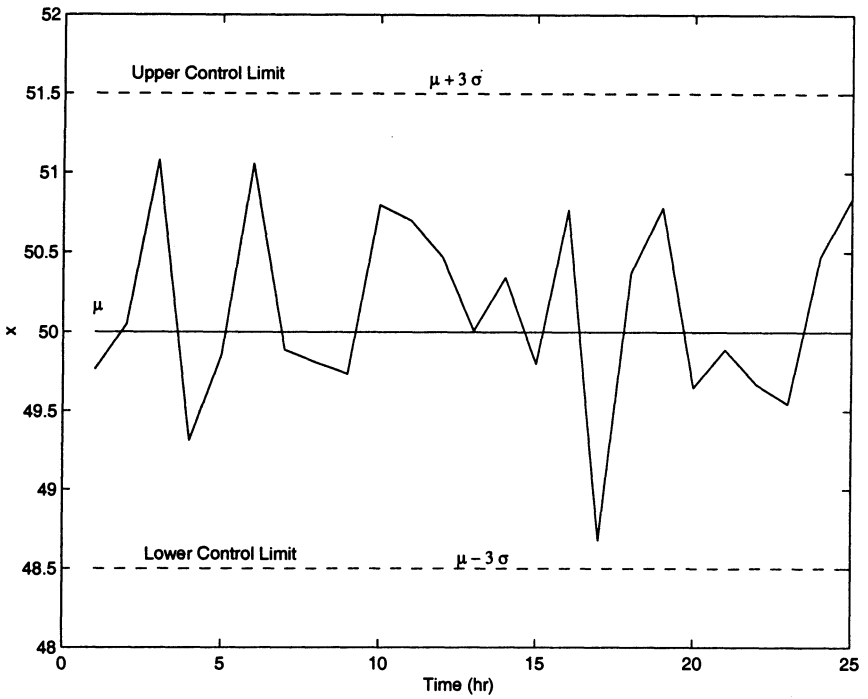

Fig. 2.3. Shewhart chart for the Mooney viscosity data taken from [171]

For a given false alarm rate, these methods can increase the sensitivity to faults over the measures using the Shewhart charts and accordingly decrease the missed detection rate, but at the expense of increasing the **detection delay**, which is the amount of time expended between the start of the fault and time to detection. This suggests that the CUSUM and EWMA charts are better suited for faults producing small persistent process shifts, and the Shewhart charts are better for detecting faults producing sudden large process shifts.

The univariate statistical charts (Shewhart, CUSUM, and EWMA) determine the thresholds for each observation variable individually without considering the information contained in the other variables. As discussed in Section 1.3, because these methods ignore the correlation between variables, they do not accurately characterize the behavior of most modern chemical processes. The next section describes the multivariate T^2 statistic, which takes into account the correlations between the variables.

2.4 T^2 Statistic

Let the data in the training set, consisting of m observation variables and n observations for each variable, be stacked into a matrix $X \in \mathcal{R}^{n \times m}$, given by

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}, \tag{2.5}$$

then the sample covariance matrix of the training set is equal to

$$S = \frac{1}{n-1} X^T X. \tag{2.6}$$

An eigenvalue decomposition of the matrix S,

$$S = V \Lambda V^T, \tag{2.7}$$

reveals the correlation structure for the covariance matrix, where Λ is diagonal and V is orthogonal ($V^T V = I$, where I is the identity matrix) [66]. The projection $\mathbf{y} = V^T \mathbf{x}$ of an observation vector $\mathbf{x} \in \mathcal{R}^m$ decouples the observation space into a set of uncorrelated variables corresponding to the elements of $\mathbf{y}$. The variance of the i^{th} element of $\mathbf{y}$ is equal to the i^{th} eigenvalue in the matrix Λ. Assuming S is invertible and with the definition

$$\mathbf{z} = \Lambda^{-1/2} V^T \mathbf{x}, \tag{2.8}$$

the Hotelling's T^2 statistic is given by [99]

$$T^2 = \mathbf{z}^T \mathbf{z}. \tag{2.9}$$

The matrix V rotates the major axes for the covariance matrix of $\mathbf{x}$ so that they directly correspond to the elements of $\mathbf{y}$, and Λ scales the elements of $\mathbf{y}$ to produce a set of variables with unit variance corresponding to the elements of $\mathbf{z}$. The conversion of the covariance matrix is demonstrated graphically in Figure 2.4 for a two-dimensional observation space ($m = 2$).

The T^2 statistic is a scaled squared 2-norm of an observation vector $\mathbf{x}$ from its mean. The scaling on $\mathbf{x}$ is in the direction of the eigenvectors and is inversely proportional to the standard deviation along the eigenvectors. This allows a scalar threshold to characterize the variability of the data in the entire m-dimensional observation space. Given a level of significance, appropriate threshold values for the T^2 statistic can be determined automatically by applying the probability distributions discussed in the next section.

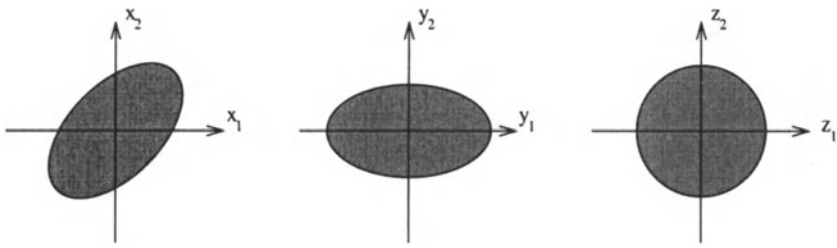

Fig. 2.4. A graphical illustration of the covariance conversion for the T^2 statistic

2.5 T^2 Statistic Thresholds

Appropriate thresholds for the T^2 statistic based on the level of significance, α, can be determined by assuming the observations are randomly sampled from a multivariate normal distribution. If it is assumed additionally that the sample mean vector and covariance matrix for normal operations are equal to the actual mean vector and covariance matrix, respectively, then the T^2 statistic follows a χ^2 distribution with m degrees of freedom [146],

$$T_\alpha^2 = \chi_\alpha^2(m). \tag{2.10}$$

The set $T^2 \leq T_\alpha^2$ is an elliptical confidence region in the observation space, as illustrated in Figure 2.5 for two process variables $m = 2$. Applying (2.10) to process data produces a confidence region defining the in-control status whereas an observation vector projected outside this region indicates that a fault has occurred. Given a level of significance α, Figure 2.5 illustrates the conservatism eliminated by employing the T^2 statistic versus the univariate statistical approach outlined in Section 2.3. As the degree of correlation between the process variables increases, the elliptical confidence region becomes more elongated and the amount of conservatism eliminated by using the T^2 statistic increases.

When the actual covariance matrix for the in-control status is not known but instead estimated from the sample covariance matrix (2.6), faults can be detected for observations taken outside the training set using the threshold given by

$$T_\alpha^2 = \frac{m(n-1)(n+1)}{n(n-m)} F_\alpha(m, n-m) \tag{2.11}$$

where $F_\alpha(m, n-m)$ is the upper $100\alpha\%$ critical point of the F-distribution with m and $n-m$ degrees of freedom [146]. For a given level of significance, the upper in-control limit in (2.11) is larger (more conservative) than the

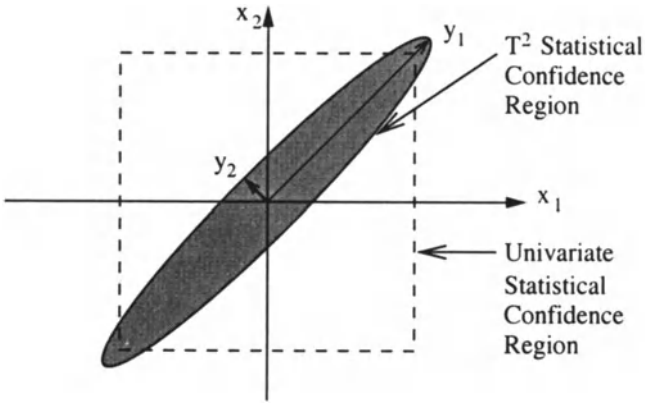

Fig. 2.5. A comparison of the in-control status regions using the T^2 statistic (2.9) and the univariate statistics (2.2) and (2.3) for two process variables [189, 208]

limit in (2.10), and the two limits approach each other as the amount of data increases ($n \to \infty$) [209].

When the sample covariance matrix (2.6) is used, the outliers in the training set can be detected using the threshold given by

$$T_\alpha^2 = \frac{(n-1)^2(m/(n-m-1))F_\alpha(m, n-m-1)}{n(1+(m/(n-m-1))F_\alpha(m, n-m-1)}. \tag{2.12}$$

For a given level of significance, the upper in-control limit in (2.12) is smaller (less conservative) than the limit in (2.10), and the two limits approach each other as the amount of data increases ($n \to \infty$) [209]. Equation (2.12) is also appropriate for detecting faults during process startup, when the covariance matrix is determined recursively on-line because no data are available *a priori* to determine the in-control limit.

The upper control limits in (2.10), (2.11), and (2.12) assume that the observation at one time instant is statistically independent to the observations at other time instances. This can be a bad assumption for short sampling intervals. However, if there are enough data in the training set to capture the normal process variations, the T^2 statistic can be an effective tool for process monitoring even if there are mild deviations from the normality or statistical independence assumptions [16, 113].

There are several extensions that are usually not studied in the process control literature, but for which there are rigorous statistical formulations. In particular, lower control limits can be derived for T^2 [209] which can detect shifts in the covariance matrix (although the upper control limit is usually

used to detect shifts in mean, it can also detect changes in the covariance matrix) [75].

The above T^2 tests are multivariable generalizations of the Shewhart chart used in the scalar case. The single variable CUSUM and EWMA charts can be generalized to the multivariable case in a similar manner [113, 140, 200, 225]. As in the scalar case, the multivariable CUSUM and EWMA charts can detect small persistent changes more readily than the multivariable Shewhart chart, but with increased detection delay.

2.6 Data Requirements

The quality and quantity of the data in the training set have a large influence on the effectiveness of the T^2 statistic as a process monitoring tool. An important question concerning the training set is, "How much data is needed to statistically populate the covariance matrix for m observation variables?" This question is answered here by determining the amount of data needed to produce a threshold value sufficiently close to the threshold obtained by assuming infinite data in the training set.

For a given level of significance α, a threshold based on infinite observations in the training set, or equivalently an exactly known covariance matrix, can be computed using (2.10), and the threshold for n observations in the training set is calculated using (2.11). The relative error produced by these two threshold values,

$$\epsilon = \frac{\frac{m(n-1)(n+1)}{n(n-m)} F_\alpha(m, n-m) - \chi^2_\alpha(m)}{\chi^2_\alpha(m)}, \tag{2.13}$$

indicates the sufficiency of the data amount n, where a large ϵ implies that more data should be collected. Table 2.2 shows the data requirements using (2.13) for various numbers of observation variables, where $\epsilon = 0.10$ and $\alpha = 0.5$; this implies that the medians of the T^2 statistic using (2.10) and (2.11) differ by less than 10%. The table indicates that the required number of observations is approximately 10 times the dimensionality of the observation space. The data requirements given in Table 2.2 do not take into account sensitivities that occur when some diagonal elements of Λ in (2.8) are small. In such cases the accuracy of the estimated values of the corresponding diagonal elements of the inverse of Λ will be poor, which will give erratic values for T^2 in (2.9). This motivates the use of the dimensionality reduction techniques described in Part III of this book.

Table 2.2. The amount of data n required for various number of observation variables m where $\epsilon = 0.10$ and $\alpha = 0.5$

Number of Observation Variables m	Data Requirement n
1	19
2	30
3	41
4	52
5	63
10	118
25	284
50	559
100	1110
200	2210

2.7 Homework Problems

1. Read the original article by Hotelling on the T^2 statistic [86]. How much of the results of this chapter were anticipated by Hotelling? Suggest reasons why these ideas took so long to work their way into industrial process applications.
2. Write a short report on the lower control limits for the T^2 statistic discussed by [209]. For what type of chemical processes and faults will such limits be useful? Give a specific process example (list process, sensors, actuators, *etc.*). Suggest reasons why most of the chemical process control and statistics literature ignores the lower control limit. Justify your statements.
3. Write a short report on the single variable CUSUM and EWMA control charts, including the mathematical expressions for the upper control limits in terms of a distribution function and assumptions on the noise statistics. You are welcome to use any books or journal articles on statistical quality control.
4. Extend the report in Problem 3 to the case of multivariate systems.
5. Consider the photographic process with the covariance matrix given in Table 1 of Jackson and Mulholdkar [101]. Reproduce as much as possible the results reported in the subsequent tables. Discuss the relative merits of the multivariate T^2 compared to scalar Shewhart charts for that process.

CHAPTER 3
PATTERN CLASSIFICATION

3.1 Introduction

Today's processes are heavily instrumented, with a large amount of data collected on-line and stored in computer databases. Much of the data are usually collected during out-of-control operations. When the data collected during the out-of-control operations have been previously diagnosed, the data can be categorized into separate *classes* where each class pertains to a particular fault. When the data have not been previously diagnosed, cluster analysis may aid the diagnoses of the operations during which the data were collected [203], and the data can be categorized into separate classes accordingly. If hyperplanes can separate the data in the classes as shown in Figure 3.1, these **separating planes** can define the boundaries for each of the fault regions. Once a fault is detected using on-line data observations, the fault can be diagnosed by determining the fault region in which the observations are located. Assuming the detected fault is represented in the database, the fault can be properly diagnosed in this manner.

This assignment of data to one of several categories or classes is the problem addressed by **pattern classification** theory [41]. The typical pattern classification system assigns an observation vector to one of several classes via three steps: **feature extraction**, **discriminant analysis**, and **maximum selection** (see Figure 3.2). The objective of the feature extraction step is to increase the robustness of the pattern classification system by reducing the dimensionality of the observation vector in a way that retains most of the information discriminating amongst the different classes. This step is especially important when there is a limited amount of quality data available. Using the information in the reduced dimensional space, the discriminant calculator computes for each class the value of the **discriminant function**, a function quantifying the relationship between the observation vector and a class. By selecting the class with the maximum discriminant function value, the discriminant functions indirectly serve as the separating planes shown in Figure 3.1; however, in general the decision boundaries will not be linear.

The objective of this chapter is to provide an overview of the statistical approach to pattern classification. The focus of this chapter is on parametric approaches to pattern classification. Assuming the statistical distributions of

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

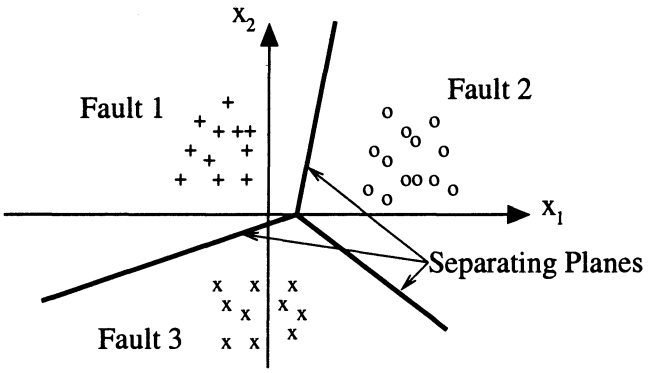

Fig. 3.1. A graphical illustration of the separating plane approach to pattern classification

the classes are known, an optimal pattern classification system can be developed using a parametric approach, while nonparametric approaches, such as the nearest neighbor rule [28], are suboptimal [41]. Pattern classification theory has been a key factor in developing fault diagnosis methods [187, 189], and the background in this chapter is important to understanding the fault diagnosis methods discussed in Part III. This chapter proceeds in Section 3.2 by presenting the optimal discriminant analysis technique for normally distributed classes. Section 3.3 discusses the feature extraction step.

3.2 Discriminant Analysis

The pattern classification system assigns an observation to class i with the maximum discriminant function value

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \forall j \neq i \tag{3.1}$$

where $g_j(\mathbf{x})$ is the discriminant function for class j given a data vector $\mathbf{x} \in \mathcal{R}^m$. The statistics of the data in each class can provide analytical measures to determine the optimal discriminant functions in terms of minimizing the *error rate*, the average probability of error. With ω_i being the event of class i (for example, a fault condition), the error rate can be minimized by using the discriminant function [41]

$$g_i(\mathbf{x}) = P(\omega_i|\mathbf{x}) \tag{3.2}$$

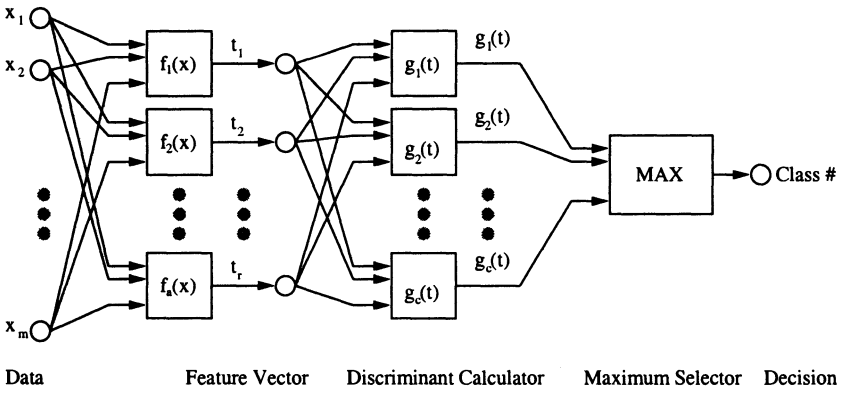

Fig. 3.2. A schemata of a typical pattern classification system, where $f_i(x)$ are the feature extraction functions and $g_i(t)$ are the discriminant analysis functions

where $P(\omega_i|\mathbf{x})$ is the *a posteriori* probability of $\mathbf{x}$ belonging to class i. This is equivalent to choosing the separating curves to be the points at which the *a posteriori* probabilities are equal.

Using Bayes' rule,

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})} \tag{3.3}$$

where $P(\omega_i)$ is the *a priori* probability for class ω_i, $p(\mathbf{x})$ is the probability density function for $\mathbf{x}$, and $p(\mathbf{x}|\omega_i)$ is the probability density function for $\mathbf{x}$ conditioned on ω_i. It can be shown that identical classification occurs when (3.2) is replaced by [41]

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|\omega_i) + \ln P(\omega_i). \tag{3.4}$$

If the data for each class is normally distributed, $p(\mathbf{x}|\omega_i)$ is given by

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{m/2}[\det(\Sigma_i)]^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right] \tag{3.5}$$

where m is the number of measurement variables, and μ_i and Σ_i are the mean vector and covariance matrix for class i, respectively [41]. Substituting (3.5) into (3.4) gives

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i) - \frac{m}{2}\ln 2\pi - \frac{1}{2}\ln[\det(\Sigma_i)] + \ln P(\omega_i) \tag{3.6}$$

This equation assumes that the mean vector and covariance matrix are known. In process monitoring applications, the true mean and covariance are not known. If the mean vector and covariance matrix are estimated and the sufficient data are available for each class to obtain highly accurate estimates, then using the estimated mean vector and covariance matrix in (3.6) will result in nearly optimal classification. Assuming that the *a priori* probability for each class is the same, the discriminant function (3.6) can be replaced by

$$g_i(\mathbf{x}) = -(\mathbf{x} - \bar{\mathbf{x}}_\mathbf{i})^T S_i^{-1}(\mathbf{x} - \bar{\mathbf{x}}_\mathbf{i}) - \ln[\det(S_i)] \tag{3.7}$$

where $\bar{\mathbf{x}}_\mathbf{i}$ is the mean vector for class i and $S_i \in \mathcal{R}^{m \times m}$ is the sample covariance matrix for class i. Using this discriminant function for classification will be referred to as **multivariate statistics** (MS) when it uses the entire data dimensionality for classification. If sufficient data are not available to accurately estimate the mean vector and covariance matrix for each class, then (3.6) will result in suboptimal classifications. In this case dimensionality reduction can be used to improve classification, as described in the next section.

Assuming that the *a priori* probability for each class is the same and the total amount of variability in each class is the same, an identical classification occurs when (3.6) is replaced by

$$g_i(\mathbf{x}) = -T_i^2 = -(\mathbf{x} - \mu_\mathbf{i})^T S_i^{-1}(\mathbf{x} - \mu_\mathbf{i}) \tag{3.8}$$

where T_i^2 is the T^2 statistic for class i (see last chapter). By using the threshold T_α^2 in (2.11), the values for each $g_i(\mathbf{x})$ in (3.8) can be converted to levels of significance which implicitly account for the uncertainties in the mean vector and covariance matrix for each class.

3.3 Feature Extraction

The objective of the pattern classification system is to minimize the *misclassification rate*, the number of incorrect classifications divided by the total number of classifications, whenever it is applied to *testing data*, data independent of the training set. The dimensionality reduction of the feature extraction step can play a key role in minimizing the misclassification rate for observations outside the training set, especially when the dimensionality of the observation space m is large and the number of observations in each class n is small. If the statistical parameters such as the mean and covariance of the classes are known exactly, from an information point of view the entire observation space should be maintained for the discriminant analysis step. In reality, inaccuracies in the statistical parameters of the classes exist. Consequently, the amount of information obtained in some directions of the observation space, specifically those that do not add much information in discriminating the data in the training set, may not outweigh the inaccuracies

in the statistical parameters, and the elimination of these directions in the feature extraction step can decrease the misclassification rate when applied to data independent of the training set.

The dimensionality reduction of the feature extraction step can also be motivated using system identification theory [137]. In system identification, it is shown that the accuracy of a model can be improved by decreasing the number of independent model parameters. This is due to the fact that the mean-squared error of the parameter estimates is reduced by decreasing the number of independent model parameters. By decreasing the number of independent parameters, the variance contribution of the parameter estimates on the mean-squared error is decreased more than the bias contribution is increased. These same arguments can be applied to the feature extraction step. For normally distributed classes, the covariance matrix has $m(m+1)/2$ independent parameters. Reducing the data dimensionality reduces the number of independent parameters in the covariance matrix. This increases the bias of the estimate of the covariance matrix, but decreases the variance. When the decrease in the variance contribution to the parameter error outweighs the increase in the bias contribution, the dimensionality reduction results in better covariance estimates and possibly lower misclassification rates when applied to data outside the training set.

Once the dimensionality reduction has been performed, classification is performed by applying discriminant analysis to the reduced dimensional space. Applications of discriminant analysis to various reduced dimensional spaces will be described in Part III. In particular, Chapter 5 describes a procedure for optimally reducing the dimensionality in terms of pattern classification.

3.4 Homework Problems

1. Derive Equation 3.4.
2. Derive Equation 3.6.
3. Derive Equation 3.8.
4. Explain in detail how to use (3.8) to compute levels of significance for each class i.
5. Consider the case where all the class covariance matrices in (3.5) are equal, $\Sigma_i = \Sigma$. Show that the discriminant function (3.6) can be replaced by a discriminant function which is linear in $\mathbf{x}$ without changing the classification. With this linear discriminant function, show that the equations $g_i(\mathbf{x}) = g_j(\mathbf{x})$ define separating planes as shown in Figure 3.1. Derive the equations for the separating curves when the class covariance matrices are not equal. What are the shapes of these separating curves?

Part III

METHODS

CHAPTER 4
PRINCIPAL COMPONENT ANALYSIS

4.1 Introduction

By projecting the data into a lower dimensional space that accurately characterizes the state of the process, dimensionality reduction techniques can greatly simplify and improve process monitoring procedures. **Principal Component Analysis** (PCA) is such a dimensionality reduction technique. It produces a lower dimensional representation in a way that preserves the correlation structure between the process variables, and is optimal in terms of capturing the variability in the data.

The application of PCA as a dimensionality reduction tool for monitoring chemical processes has been studied by several academic and industrial engineers [120, 182]. Applications of PCA to plant data have been conducted at DuPont and other companies, with much of the results published in conference proceedings and journal articles [111, 182, 181, 228]. Several academics have performed similar studies based on data collected from computer simulations of processes [77, 105, 125, 144, 186, 187, 189, 208]. For some applications, most of the variability in the data can be captured in two or three dimensions [144], and the process variability can be visualized with a single plot. This one-plot visualization and the structure abstracted from the multidimensional data assist the operators and engineers in interpreting the significant trends of the process data [120].

For the cases when most of the data variations cannot be captured in two or three dimensions, methods have been developed to automate the process monitoring procedures [146, 182, 189]. The application of PCA in these methods is motivated by one or more of three factors. First, PCA can produce lower dimensional representations of the data which better generalize to data independent of the training set than that using the entire dimensionality of the observation space, and therefore, improve the proficiency of detecting and diagnosing faults. Second, the structure abstracted by PCA can be useful in identifying either the variables responsible for the fault and/or the variables most affected by the fault. Third, PCA can separate the observation space into a subspace capturing the systematic trends of the process and a subspace containing essentially the random noise. Since it is widely accepted that certain faults primarily affect one of the two subspaces [43, 230, 231], applying one measure developed for one subspace and another measure developed for

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

the other subspace can increase the sensitivity of the process monitoring scheme to faults in general. The three aforementioned attributes of PCA are further discussed later in this chapter.

The purpose of this chapter is to describe the PCA methods for process monitoring. It begins in Section 4.2 by defining PCA and in Section 4.3 by discussing the different methods which can be used to automatically determine the order of the PCA representation. Sections 4.4, 4.5, and 4.6 discuss the PCA developments for fault detection, identification, and diagnosis, respectively. In Section 4.7 is a discussion of **dynamic PCA** (DPCA), which takes into account serial correlations in the process data. Section 4.8 discusses other PCA-based process monitoring methods.

4.2 Principal Component Analysis

PCA is a linear dimensionality reduction technique, optimal in terms of capturing the variability of the data. It determines a set of orthogonal vectors, called **loading vectors**, ordered by the amount of variance explained in the loading vector directions. Given a training set of n observations and m process variables stacked into a matrix X as in (2.5), the loading vectors are calculated by solving the stationary points of the optimization problem

$$\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^T X^T X \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \tag{4.1}$$

where $\mathbf{v} \in \mathcal{R}^m$. The stationary points of (4.1) can be computed via the singular value decomposition (SVD)

$$\frac{1}{\sqrt{n-1}} X = U \Sigma V^T \tag{4.2}$$

where $U \in \mathcal{R}^{n\times n}$ and $V \in \mathcal{R}^{m\times m}$ are unitary matrices, and the matrix $\Sigma \in \mathcal{R}^{n\times m}$ contains the nonnegative real **singular values** of decreasing magnitude along its main diagonal ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$), and zero offdiagonal elements. The loading vectors are the orthonormal column vectors in the matrix V, and the variance of the training set projected along the i^{th} column of V is equal to σ_i^2. Solving (4.2) is equivalent to solving an eigenvalue decomposition of the sample covariance matrix S,

$$S = \frac{1}{n-1} X^T X = V \Lambda V^T \tag{4.3}$$

where the diagonal matrix $\Lambda = \Sigma^T \Sigma \in \mathcal{R}^{m\times m}$ contains the nonnegative real **eigenvalues** of decreasing magnitude ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$) and the i^{th} eigenvalue equals the square of the i^{th} singular value (*i.e.*, $\lambda_i = \sigma_i^2$).

In order to optimally capture the variations of the data while minimizing the effect of random noise corrupting the PCA representation, the loading vectors corresponding to the a largest singular values are typically retained. The motivation for reducing the dimensionality of the PCA representation is analogous to the arguments given in Section 3.3 for pattern classification. Selecting the columns of the loading matrix $P \in \mathcal{R}^{m \times a}$ to correspond to the loading vectors associated with the first a singular values, the projections of the observations in X into the lower dimensional space are contained in the **score matrix**,

$$T = XP, \tag{4.4}$$

and the projection of T back into the m-dimensional observation space,

$$\hat{X} = TP^T. \tag{4.5}$$

The difference between X and $\hat{X}$ is the residual matrix E:

$$E = X - \hat{X}. \tag{4.6}$$

The **residual matrix** captures the variations in the observation space spanned by the loading vectors associated with the $m - a$ smallest singular values. The subspaces spanned by $\hat{X}$ and E are called the **score space** and **residual space**, respectively. The subspace contained in the matrix E has a small signal-to-noise ratio, and the removal of this space from X can produce a more accurate representation of the process, $\hat{X}$.

Defining $\mathbf{t_i}$ to be i^{th} column of T in the training set, the following properties can be shown (see Homework Problem 5) [181]

1. $\text{Var}(\mathbf{t_1}) \geq \text{Var}(\mathbf{t_2}) \geq \cdots \geq \text{Var}(\mathbf{t_a})$.
2. $\text{Mean}(\mathbf{t_i}) = 0;\ \forall i$.
3. $\mathbf{t_i}^T \mathbf{t_k} = 0;\ \forall i \neq k$.
4. There exists no other orthogonal expansion of a components that captures more variations of the data.

A new observation (column) vector in the testing set, $\mathbf{x} \in \mathcal{R}^m$, can be projected into the lower dimensional score space $t_i = \mathbf{x}^T \mathbf{p_i}$ where $\mathbf{p_i}$ is the i^{th} loading vector (see Figure 4.1). The transformed variable t_i is also called the i^{th} **principal component** of $\mathbf{x}$ [102]. To distinguish between the transformed variables and the transformed observation, the transformed variables will be called **principal components** and the individual transformed observations will be called **scores**. The statistical properties listed above allow each of the scores to be monitored separately using the univariate statistical procedures discussed in Section 2.3. With the vectors projected into the lower dimensional space using PCA, only a variables needed to be monitored, as compared with m variables without the use of PCA. When enough data are collected in the testing set, the score vectors $\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_a}$ can be formed. If

these score vectors do not satisfy the four properties listed above, the testing set is most likely collected during different operating conditions than for the training set. This abstraction of structure from the multidimensional data is a key component of the score contribution method for fault identification discussed in Section 4.5.

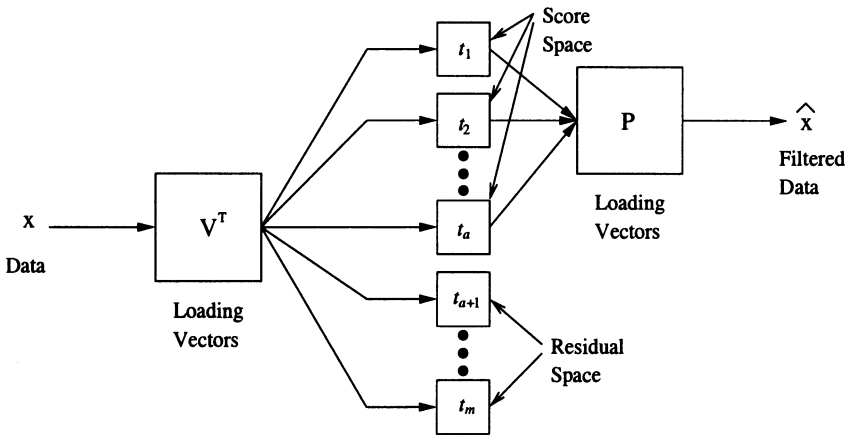

Fig. 4.1. The projection of the observation vector $\mathbf{x}$ into the score and residual spaces, and the computation of the filtered observation $\hat{\mathbf{x}}$

To illustrate the application of PCA, experimental data from [25, 50] are used. The data set consists of three classes, with each class containing 4 measurements and 50 observations. Class 3 data are used to construct X as in (2.5), where $n = 50$ and $m = 4$. After autoscaling X and solving (4.3), we have

$$\Lambda = \begin{bmatrix} 1.92 & 0 & 0 & 0 \\ 0 & 0.96 & 0 & 0 \\ 0 & 0 & 0.88 & 0 \\ 0 & 0 & 0 & 0.24 \end{bmatrix}, \tag{4.7}$$

and

$$V = \begin{bmatrix} 0.64 & -0.29 & 0.052 & -0.71 \\ 0.64 & -0.23 & 0.25 & 0.69 \\ 0.34 & 0.33 & -0.88 & 0.11 \\ 0.25 & 0.87 & 0.41 & -0.09 \end{bmatrix}. \tag{4.8}$$

The total variance for X projected along V is equal to the trace of Λ, which is 4.0. The i^{th} value in the diagonal of Λ indicates the amount of variance

captured by the i^{th} principal component. If only one principal component is retained (*i.e.*, $a = 1$), $(1.92/4.0)100\% = 48.0\%$ of the total variance is captured. For $a = 2$, 72% of the total variance is captured. For $a = 2$, the loading matrix P is equal to the first two columns of V:

$$P = \begin{bmatrix} 0.64 & -0.29 \\ 0.64 & -0.23 \\ 0.34 & 0.33 \\ 0.25 & 0.87 \end{bmatrix}. \tag{4.9}$$

The score matrix T is calculated according to (4.4). The advantage of retaining only two principal components is that the process variability can be visualized by plotting t_2 versus t_1 (see Figure 4.2).

It is easy to verify that $\text{Var}(\mathbf{t_1}) \geq \text{Var}(\mathbf{t_2})$ by observing that the variation along the horizontal axis is much greater than that of the vertical axis for the Class 3 data in Figure 4.2. The ellipsoid and the data for Class 3 are centered at the origin, which indicates that $\text{Mean}(\mathbf{t_1}) = \text{Mean}(\mathbf{t_2}) = 0$. It is straightforward to verify that $\mathbf{t_1}$ and $\mathbf{t_2}$ are orthogonal to each other.

A threshold defines an elliptical confidence region for data belonging to Class 3 (the calculation of the threshold will be described in Section 4.4). In this example, statistics predicts that there is a 95% probability that a Class 3 data point should fall inside the ellipsoid. It is clearly shown in Figure 4.2 that PCA is able to separate Class 3 data from Classes 1 and 2, except for the apparent outlier located at $(t_1, t_2) = (2.5, 5.6)$.

4.3 Reduction Order

It is commonly accepted and with certain assumptions theoretically justified [230] that the portion of the PCA space corresponding to the larger singular values describes most of the *systematic* or *state variations* occurring in the process, and the portion of the PCA space corresponding to the smaller singular values describe the *random noise*. By appropriately determining the number of loading vectors, a, to maintain in the PCA model, the systematic variations can be decoupled from the random variations, and the two types of variations can be monitored separately, as discussed in Section 4.4. Several techniques exist for determining the value of the reduction order a [100, 77], but there appears to be no dominant technique. The methods for determining a described here are:

1. the percent variance test,
2. the scree test,
3. parallel analysis, and
4. the PRESS statistic.

The **percent variance** method determines a by calculating the smallest number of loading vectors needed to explain a specific minimum percentage

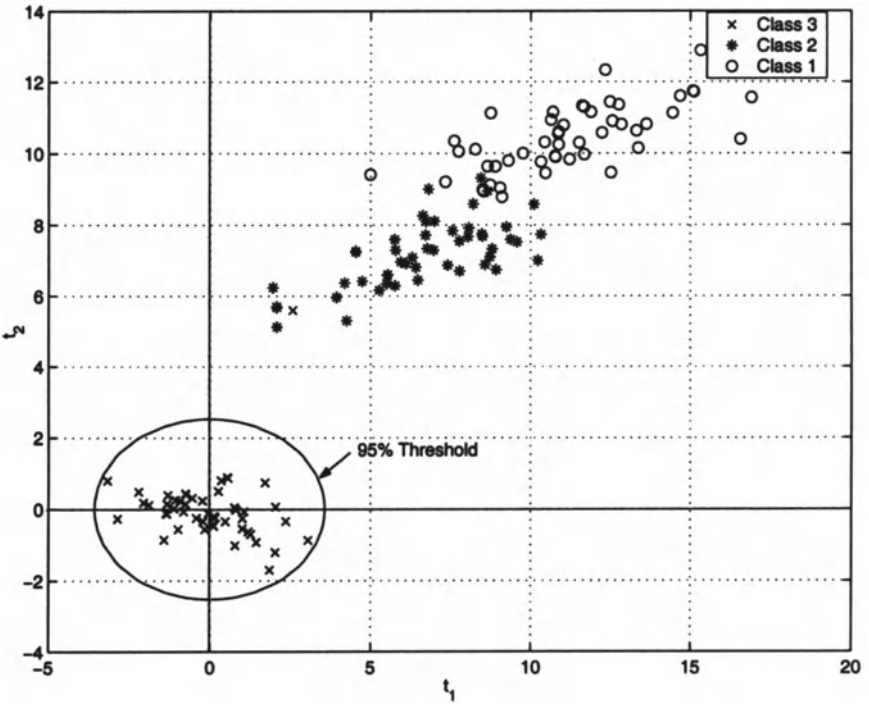

Fig. 4.2. The projections of experimental data [50, 25] for three classes onto the first two PCA loading vectors

of the total variance. (Recall that the variance associated with the i^{th} loading vector is equal to the square of the singular value, σ_i^2.) Because this minimum percentage is chosen arbitrarily, it may be too low or too high for a particular application.

The **scree test** assumes that the variance, σ_i^2, corresponding to the random noise forms a linear profile. The dimension of the score space a is determined by locating the value of σ_i^2 where the profile is no longer linear. The identification of this break can be ambiguous, and thus, this method is difficult to automate. It is especially ambiguous when several breaks from linearity occur in the profile.

Parallel analysis determines the dimensionality by comparing the variance profile to that obtained by assuming independent observation variables. The reduction order is determined as the point at which the two profiles cross. This approach ensures that the significant correlations are captured in the score space, and it is particularly attractive since it is intuitive and easy to

automate. Ku, Storer, and Georgakis [125] recommend the parallel analysis method, because in their experience, it performs the best overall.

The dimension of the score space can also be determined using a **cross-validation** procedure with the **PREdiction Sum of Squares** (PRESS) statistic [232],

$$PRESS(i) = \frac{1}{mn}||X - \hat{X}||_F^2 \tag{4.10}$$

where i is the number of loading vectors retained to calculate $\hat{X}$ and $||\cdot||_F$ is the Frobenius norm (the square root of the sum of squares of all the elements). For the implementation of this technique, the training set is divided into groups. The PRESS statistic for one group is computed based on various dimensions of the score space, i, using all the other groups. This is repeated for each group, and the value i associated with the minimum average PRESS statistic determines the dimension of the score space.

4.4 Fault Detection

As discussed in Section 2.4, the T^2 statistic can be used to detect faults for multivariate process data. Given an observation vector $\mathbf{x}$ and assuming that $\Lambda = \Sigma^T\Sigma$ is invertible, the T^2 statistic in (2.9) can be calculated directly from the PCA representation (4.2)

$$T^2 = \mathbf{x}^T V(\Sigma^T\Sigma)^{-1}V^T\mathbf{x}. \tag{4.11}$$

This follows from the fact that the V matrix in (2.7) can be computed to be identical to the V matrix in (4.2), and the σ_i^2 are equal to the diagonal elements of Λ. When the number of observation variables is large and the amount of data available is relatively small, the T^2 statistic (4.11) tends to be an inaccurate representation of the in-control process behavior, especially in the loading vector directions corresponding to the smaller singular values. Inaccuracies in these smaller singular values have a huge effect on the calculated T^2 statistic because the square of the singular values are inverted in (4.11). Additionally, the smaller singular values are prone to errors because these values contain small signal-to-noise ratios and the associated loading vector directions often suffer from a lack of excitation. Therefore, in this case the loading vectors associated only with the larger singular values should be retained in calculating the T^2 statistic.

By including in the matrix P the loading vectors associated only with the a largest singular values, the T^2 statistic for the lower dimensional space can be computed [99]

$$T^2 = \mathbf{x}^T P\Sigma_a^{-2}P^T\mathbf{x}. \tag{4.12}$$

where Σ_a contains the first a rows and columns of Σ. The T^2 statistic (4.12) measures the variations in the score space only. If the actual mean and covariance are known, the T^2 statistic threshold derived from (2.10) is

$$T_\alpha^2 = \chi_\alpha^2(a). \tag{4.13}$$

When the actual covariance matrix is estimated from the sample covariance matrix, the T^2 statistic threshold derived from (2.11) is

$$T_\alpha^2 = \frac{a(n-1)(n+1)}{n(n-a)} F_\alpha(a, n-a). \tag{4.14}$$

To detect outliers in the training set, the threshold derived from (2.12) is

$$T_\alpha^2 = \frac{(n-1)^2(a/(n-a-1))F_\alpha(a, n-a-1)}{n(1+(a/(n-a-1))F_\alpha(a, n-a-1)}. \tag{4.15}$$

Because the T^2 statistic in (4.12) is not affected by the inaccuracies in the smaller singular values of the covariance matrix, it is able to better represent the normal process behavior and provides a more robust fault detection measure when compared to the T^2 statistic in (4.11). Using the arguments in Section 4.3, the T^2 statistic (4.12) can be interpreted as measuring the systematic variations of the process, and a violation of the threshold would indicate that the systematic variations are out-of-control.

For the example in the last section, we have $n = 50$ and $a = 2$. According to an $F-$distribution table [81], $F_{0.05}(2, 48) = 3.19$. The threshold T_α^2 is equal to 6.64 according to (4.14). The elliptical confidence region, as shown in Figure 4.2, is given by

$$T^2 = \mathbf{x}^T P \Sigma_a^{-2} P^T \mathbf{x} \leq 6.64, \tag{4.16}$$

with

$$\Sigma_a^2 = \begin{bmatrix} 1.92 & 0 \\ 0 & 0.96 \end{bmatrix}. \tag{4.17}$$

The equation

$$\mathbf{t} = P^T \mathbf{x} \tag{4.18}$$

converts this region into the ellipse in Figure 4.2. Inserting (4.18) into (4.16) gives

$$\mathbf{t}^T \Sigma_a^{-2} \mathbf{t} \leq 6.64 \tag{4.19}$$

or

$$\frac{t_1^2}{1.92} + \frac{t_2^2}{0.96} \leq 6.64 \tag{4.20}$$

where $\mathbf{t} = [t_1 \ t_2]^T$.

Data from Classes 1 and 2 are used to illustrate that the PCA model is able to detect data that do not come from Class 3. The data sets for Classes 1 and 2 are first autoscaled according to the mean and standard deviation of Class 3. Equation 4.4 is used to calculate the score matrices for Classes 1 and 2. As shown in Figure 4.2, the mean of each score vector for Classes 1 and 2 is not equal to zero. Indeed, all the data points for Classes 1 and 2 fall outside the elliptical confidence region, indicating data from Classes 1 and 2 are indeed different from the Class 3 data.

The T^2 statistic in (4.11) is overly sensitive to inaccuracies in the PCA space corresponding to the smaller singular values because it directly measures the variation along *each* of the loading vectors. In other words, it *directly* measures the scores corresponding to the smaller singular values. The portion of the observation space corresponding to the $m - a$ smallest singular values can be monitored more robustly by using the Q statistic [101, 100, 103, 119, 233]

$$Q = \mathbf{r}^T\mathbf{r}, \qquad \mathbf{r} = (I - PP^T)\mathbf{x}, \tag{4.21}$$

where $\mathbf{r}$ is the residual vector, a projection of the observation $\mathbf{x}$ into the residual space. Since the Q statistic does not directly measure the variations along each loading vector but measures the total sum of variations in the residual space, the Q statistic does not suffer from an over-sensitivity to inaccuracies in the smaller singular values [101]. The Q statistic, also known as the **squared prediction error** (SPE), is a squared 2-norm measuring the deviation of the observations to the lower dimensional PCA representation.

The distribution for the Q statistic has been approximated by Jackson and Mudholkar [101]

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \tag{4.22}$$

where $\theta_i = \sum_{j=a+1}^{n} \sigma_j^{2i}$, $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$, and c_α is the normal deviate corresponding to the $(1 - \alpha)$ percentile. Given a level of significance, α, the threshold for the Q statistic can be computed using (4.22) and be used to detect faults.

Within the context of Section 4.3, the Q statistic measures the random variations of the process, for example, that associated with measurement noise. The threshold (4.22) can be applied to define the normal variations for the random noise, and a violation of the threshold would indicate that the random noise has significantly changed. The T^2 and Q statistics along with their appropriate thresholds detect different types of faults, and the advantages of both statistics can be utilized by employing the two measures together. When these two statistics are utilized along with their respective

thresholds, it produces a cylindrical in-control region, as illustrated for $a = 2$ in Figure 4.3. The figure indicates that the 'x' data was collected during in-control operations, the 'o' data represents a T^2 statistic violation, and the '+' data represents a Q statistic violation.

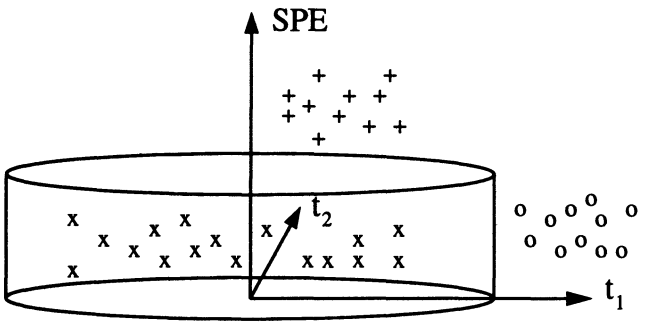

Fig. 4.3. A graphical illustration for fault detection using the Q and T^2 statistics

4.5 Fault Identification

Once a fault has been detected, the next step is to determine the cause of the out-of-control status. Diagnosing the fault can be a challenging task for the plant operators and engineers in view of the fact that usually a large number of process variables are monitored, many of the variables go out-of-control in a short time period when a fault occurs, and chemical processes are highly integrated and complex. The objective of fault identification is to determine which observation variables are most relevant to diagnosing the fault, thereby focusing the plant operators and engineers on the subsystem(s) most likely where the fault occurred. This assistance provided by the fault identification scheme in locating the fault can effectively incorporate the operators and engineers in the process monitoring scheme and significantly reduce the time to recover in-control operations.

Traditionally, univariate statistical techniques were employed for fault identification. Given an observation vector $\mathbf{x}$, the normalized errors for each variable x_j were calculated as

$$e_j = (x_j - \mu_j)/s_j \tag{4.23}$$

where μ_j is the mean and s_j is the standard deviation of the j^{th} variable. These normalized errors were plotted on the same graph, and thresholds based on the level of significance were used to detect the out-of-control variables, as discussed in Section 2.3. However, univariate statistical techniques for fault identification can leave out variables that are responsible for the fault because the techniques do not account for correlations among the process variables, or can give alarm readings for so many variables that the engineer has little guidance on the main variables of concern [113].

Contribution plots are a PCA approach to fault identification that takes into account the spacial correlations, thereby improving upon the univariate statistical techniques [113, 157]. The approach is based on quantifying the contribution of each process variable to the individual scores of the PCA representation, and for each process variable summing the contributions *only* of those scores responsible for the out-of-control status. The procedure is applied in response to a T^2 violation, and it is summarized as follows.

1. Check the normalized scores $(t_i/\sigma_i)^2$ for the observation $\mathbf{x}$ and determine the $r \leq a$ scores responsible for the out-of-control status. For instance, those scores with $(t_i/\sigma_i)^2 > (T_\alpha^2)^{1/a}$. (Recall that t_i is the score of the observation projected onto the i^{th} loading vector, and σ_i is the corresponding singular value.)
2. Calculate the *contribution* of each variable x_j to the out-of-control scores t_i

$$cont_{i,j} = \frac{t_i}{\sigma_i^2} p_{i,j}(x_j - \mu_j) \tag{4.24}$$

   where $p_{i,j}$ is the $(i,j)^{th}$ element of the loading matrix P.
3. When $cont_{i,j}$ is negative, set it equal to zero.
4. Calculate the *total contribution* of the j^{th} process variable, x_j,

$$CONT_j = \sum_{i=1}^{r} (cont_{i,j}). \tag{4.25}$$

5. Plot $CONT_j$ for all m process variables, x_j, on a single graph.

The variables responsible for the fault can be prioritized or ordered by the total contribution values $CONT_j$, and the plant operators and engineers can immediately focus on those variables with high $CONT_j$ values and use their process knowledge to determine the cause of the out-of-control status. While the overall variable contribution approach can be applied to the portion of the observation space corresponding to the $m - a$ smallest singular values, it is not practical because the total contribution values $CONT_j$ would be overly sensitive to the smaller singular values.

Wise *et al.* [231] developed a PCA approach to fault identification which is based on quantifying the total variation of each of the process variables in the residual space. Assuming that the $m - a$ smallest singular values are

all equal, the variance for each variable x_j inside the residual space can be estimated as [231]

$$\hat{s}_j^2 = \sum_{i=a+1}^{p} p_{i,j}\sigma_i^2. \tag{4.26}$$

Given q new observations, the variance of the j^{th} variable outside the PCA model space can be tested where

$$s_j^2/\hat{s}_j^2 > F_\alpha(q-a-1, n-a-1) \tag{4.27}$$

would indicate an out-of-control variable, where s_j^2 and $\hat{s}_j^2$ are the variance estimates of the j^{th} variable for the new and training set observations, respectively, and $F_\alpha(q-a-1, n-a-1)$ is the $(1-\alpha)$ percentile limit using the F distribution [81]. Equation 4.27 is testing the null hypothesis, with the null hypothesis being $s_j = \hat{s}_j$ and the one-sided alternative hypothesis being $s_j > \hat{s}_j$. The one-sided alternative hypothesis is accepted (*i.e.*, the null hypothesis is rejected) if (4.27) holds [81]. In most of the times, the variable that is responsible for a fault has a larger variance than it has in the training set (*i.e.*, $s_j > \hat{s}_j$). However, this is not always true. For example, a broken sensor may give constant reading, indicating that $s_j < \hat{s}_j$. This motivates the use of two-sided hypothesis testing, with the null hypothesis being $s_j = \hat{s}_j$ and the two-sided alternative hypothesis being $s_j \neq \hat{s}_j$. We conclude $\hat{s}_j \neq s_j$ if [81]

$$s_j^2/\hat{s}_j^2 > F_{\alpha/2}(q-a-1, n-a-1) \tag{4.28}$$

or

$$\hat{s}_j^2/s_j^2 > F_{\alpha/2}(n-a-1, q-a-1). \tag{4.29}$$

In addition, a large shift in the mean inside the residual space occurs if [231, 81]

$$\frac{\mu_j - \hat{\mu}_j}{\hat{s}_j\sqrt{\frac{1}{q-a}+\frac{1}{n-a}}} > t_{\alpha/2}(q+n-2a-2) \tag{4.30}$$

or

$$\frac{\mu_j - \hat{\mu}_j}{\hat{s}_j\sqrt{\frac{1}{q-a}+\frac{1}{n-a}}} < -t_{\alpha/2}(q+n-2a-2), \tag{4.31}$$

where μ_j and $\hat{\mu}_j$ are the means of x_j for the new and training set observations, respectively, and $t_{\alpha/2}(q+n-2a-2)$ is the $(1-\alpha/2)$ percentile limit using the t distribution. Equations 4.30 and 4.31 are testing the null hypothesis, with the null hypothesis being $\mu_j = \hat{\mu}_j$ and the alternative hypothesis being $\mu_j \neq \hat{\mu}_j$. The alternative hypothesis is accepted if (4.30) or (4.31) holds [81].

The variables responsible for the out-of-control status, detected by the Q statistic, can be identified using (4.27), (4.30), and (4.31). In addition, the variables can be prioritized using the expression values (4.27), (4.30), and (4.31) where the variable with the largest expression value is given priority. In [231], sensor failures are detected and identified using (4.27), (4.30), and (4.31). Other PCA-based methods developed specifically for detecting sensor failures are discussed elsewhere [43, 164].

The fault identification approaches using (4.27), (4.30), and (4.31) require a group of $q >> 1$ observations. As discussed in Section 2.3, measures based on several consecutive observations are able to increase the robustness and sensitivity over measures based on only a single observation, but result in a slower response time for larger process shifts. A fault identification measure based on an observation vector at a single time instant is the normalized error

$$RES_j = r_j / \hat{s}_j \tag{4.32}$$

where r_j is the j^{th} variable of the residual vector. The values of (4.32) can be used to prioritize the variables where the variable with the highest normalized error is given priority. The measure (4.32), when compared to (4.27), (4.30), and (4.31), is able to more accurately indicate the current status of the process immediately after a large process shift.

4.6 Fault Diagnosis

The previous section discussed fault identification methods, which identify the variables associated with the faulty subsystem. Although these methods assist in diagnosing the faults, it may take a substantial amount of time and process expertise on behalf of the plant operators and engineers before the fault is properly diagnosed. Much of this time and expertise can be eliminated by employing an automated fault diagnosis scheme. One approach is to construct separate PCA models for each process unit [77]. A fault associated with a particular process unit is assumed to occur if the PCA model for that unit indicates that the process is out-of-control. While this approach can narrow down the cause of abnormal process operations, it will not unequivocally diagnose the cause. This distinguishes these *fault isolation* techniques (which are based on non-supervised classification) from the fault diagnosis techniques (which are based on supervised classification) described below.

Several researchers have proposed techniques to use principal component analysis for fault diagnosis. The simplest approach is to construct a single PCA model and define regions in the lower dimensional space which Classifies whether a particular fault has occurred [231]. This approach is unlikely to be effective when a significant number of faults can occur [238].

It was described in Chapter 3 how a pattern classification system can be applied to diagnose faults automatically. The feature extraction step was

shown to be important especially when the dimensionality of the data is large and the quantity of quality data is relatively small (see Section 3.3). A PCA approach which can handle a larger number of faults than using a single PCA model is to develop a separate PCA model based on data collected during each specific fault situation, and then apply the Q [123], T^2 [186], or other statistics [186, 187, 189, 238] to each PCA model to predict which fault or faults most likely occurred. This approach is a combination of Principal Component Analysis and discriminant analysis [187]. Various discriminant functions for diagnosing faults, and these are discussed in the following.

One way to use PCA for fault diagnosis is to derive *one* model based on the data from *all* fault classes. Stacking the data for all fault classes into matrix X, the loading matrix P can be calculated based on (4.2) or (4.3). The maximum likelihood classification for an observation $\mathbf{x}$ is fault class i with the maximum **score discriminant**, which is derived from (3.6) to be

$$\begin{aligned} g_i(\mathbf{x}) = &-\tfrac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_\mathbf{i})^T P\left(P^T S_i P\right)^{-1} P^T(\mathbf{x}-\bar{\mathbf{x}}_\mathbf{i}) + \ln(p_i) \\ &-\tfrac{1}{2}\ln\left[\det\left(P^T S_i P\right)\right] \end{aligned} \tag{4.33}$$

where $\bar{\mathbf{x}}_\mathbf{i}$ is the mean vector for class i,

$$\bar{\mathbf{x}}_\mathbf{i} = \frac{1}{n_i}\sum_{\mathbf{x_j}\in\mathcal{X}_i} \mathbf{x_j}, \tag{4.34}$$

n_i is the number of data points in fault class i, $\mathcal{X}_i$ is the set of vectors $\mathbf{x_j}$ which belong to the fault class i, and $S_i \in \mathcal{R}^{m\times m}$ is the sample covariance matrix for fault class i, as defined in (2.6).

If P is selected to include all of the dimensions of the data (*i.e.*, $P = V \in \mathcal{R}^{m\times m}$) and the overall likelihood for all fault classes is the same, Equation 4.33 reduces to the discriminant function for multivariate statistics (MS) as defined in (3.7). MS selects the most probable fault class based on maximizing the discriminant function (3.7). MS also serves as a benchmark for the other statistics, as the dimensionality should only be reduced if it decreases the misclassification rate for a testing set.

The **score discriminant**, **residual discriminant**, and **combined discriminant** are three discriminant analysis techniques used with *multiple* PCA models [186]. Assuming the PCA models retain the important variations in discriminating between the faults, an observation $\mathbf{x}$ is classified as being in the fault class i with the maximum score discriminant

$$g_i(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T P_i \Sigma_{a,i}^{-2} P_i^T \mathbf{x} - \frac{1}{2}\ln[\det(\Sigma_{a,i}^2)] + \ln(p_i) \tag{4.35}$$

where P_i is the loading matrix for fault class i, $\Sigma_{a,i}$ is the diagonal matrix Σ_a as shown in (4.12) for fault class i ($\Sigma_{a,i}^2$ is the covariance matrix of $P_i\mathbf{x}$), and p_i is the overall likelihood of fault class i [103, 189]. Note that (4.35)

assumes that the observation vector $\mathbf{x}$ has been autoscaled according to the mean and standard deviation of the training set for fault class i. Equation 4.35 is based on the discriminant function (3.6).

The matrices P_i, $\Sigma_{a,i}$, and p_i in (4.35) depend solely on fault class i, that is, the discriminant function for *each* fault class is derived *individually*. A weakness of this approach is that useful information for other classes is not utilized when each model is derived. In general, the reduction order a for each fault class is different. This indicates that the discriminant function (4.35) for each fault class i is evaluated based on different dimensions of the projected data $P_i^T\mathbf{x}$. This inconsistency can result in relatively high misclassification rates.

In contrast to (4.35), the projection matrix P in (4.33) not only utilizes information from all fault classes, but also projects the data onto the same dimensions for each class. Because of these properties, the discriminant function (4.33) can significantly outperform (4.35) for diagnosing faults. To distinguish the *one-model* PCA with the *multi-model* PCA, we will refer to the one-model PCA as **PCA1** and the multi-model PCA as **PCAm** throughout the book.

Assuming that the overall likelihood for all fault classes is the same and the sample covariance matrix of $P_i\mathbf{x}$ for all classes is the same, the use of the score discriminant (4.35) reduces to use of the T_i^2 statistic, where

$$T_i^2 = \mathbf{x}^T P_i \Sigma_{a,i}^{-2} P_i^T \mathbf{x} \tag{4.36}$$

(similarly as shown in Section 3.2). In this case, the score discriminant will select the fault class as that which corresponds to the minimum T_i^2 statistic.

Assuming that the important variations in discriminating between the faults are contained in the residual space for each fault class, it is most likely that an observation is represented by the fault class i with the minimum **residual discriminant**

$$Q_i/(Q_\alpha)_i \tag{4.37}$$

where the subscript i indicates fault class i. If the important variations in discriminating between the faults are contained both within the score and residual space, then an observation is most likely to be represented by the fault class i with the minimum **combined discriminant**

$$c_i[T_i^2/(T_\alpha^2)_i] + (1 - c_i)[Q_i/(Q_\alpha)_i] \tag{4.38}$$

where c_i is a weighting factor between 0 and 1 for fault class i. Assuming an out-of-control observation does not represent a new fault, each of these discriminant analysis techniques (4.35), (4.37), and (4.38) can be used to diagnose the fault.

When a fault is diagnosed as fault i, it is *not* likely to represent a new fault when

$$[T_i^2/(T_\alpha^2)_i] << 1 \tag{4.39}$$

and

$$[Q_i/(Q_\alpha)_i] << 1. \tag{4.40}$$

These conditions indicate that the observation is a good match to fault model i. If either of these conditions is not satisfied (for example, $[T_i^2/(T_\alpha^2)_i]$ or $[Q_i/(Q_\alpha)_i]$ is greater than 1), then the observation is not accurately represented by fault class i and it is likely that the observation represents a new fault.

Before the application of a pattern classification system to a fault diagnosis scheme, it is useful to assess the likelihood of successful diagnosis. In [187, 189], Raich and Cinar describe a quantitative measure of similarity between the covariance structures of two classes. The measure, referred to as the **similarity index**, for Classes 1 and 2 is calculated as

$$f = \frac{1}{m}\sum_{j=1}^{m} \tilde{\sigma}_j \tag{4.41}$$

where $\tilde{\sigma}_j$ is the j^{th} singular value of $V_1^T V_2$ and the matrices V_1 and V_2 contain all m loading vectors for Classes 1 and 2, respectively. The value of f ranges between 0 and 1, where a value near 0 indicates a *lack of similarity* and a value equal to 1 indicates an *exact similarity* [121]. While a high similarity does not guarantee misdiagnosis, a low similarity does generally indicate a low probability of misdiagnosis. The similarity index can be applied to PCA models by replacing V_1 and V_2 with the loading matrix P_1 for Class 1 and the loading matrix P_2 for Class 2, respectively.

In [187, 189], a measure of class similarity using the overlap of the mean for one class into the score space of another class is developed from [147]. Define $\mu_1 \in \mathcal{R}^m$ and $\mu_2 \in \mathcal{R}^m$ to be the means of Classes 1 and 2, respectively, $P \in \mathcal{R}^{m \times a}$ as the projection matrix containing the a loading vectors for Class 2, ρ as the fraction of the explained variance in the data used to build the second PCA model, and $\bar{\Sigma} \in \mathcal{R}^{a \times a}$ as the covariance in a model directions for the second PCA model. The test statistic, referred to as the **mean overlap**, for Classes 1 and 2 is

$$m = \frac{\mathbf{r}^T\mathbf{r}}{(1-\rho)\mathbf{t}^T\bar{\Sigma}^{-1}\mathbf{t}} \tag{4.42}$$

where $\mathbf{t} = P^T(\mu_1 - \mu_2)$ is the approximation of μ_1 by the second model and $\mathbf{r} = P\mathbf{t} - \mu_1$ is the residual error in μ_1 unexplained by the second model. The threshold for (4.42) can be determined from the following distribution

$$m_\alpha = F_\alpha(m-a, n-a) \tag{4.43}$$

where n is the number of model observations for Class 2. In simulations, Raich and Cinar found that the mean overlap was not as successful as the similarity index for indicating pairwise misdiagnosis [187, 189].

Multiple faults occurring within the same time window are likely to happen for many industrial processes. The statistics for *detecting* a single fault are directly applicable for detecting multiple faults because the threshold in (4.14) depends only on the data from the normal operating conditions (Fault 0). The task of *diagnosing* multiple faults is rather challenging and the proficiencies of the fault diagnosis statistics depend on the nature of the combination of the faults. A straightforward approach for diagnosing multiple faults is to introduce new models for each combination of interest; this approach could describe combinations of faults that produce models that are not simply the consensus of component models [187, 189]. The disadvantage of this approach is that the number of combinations grows exponentially with the number of faults. For a detailed discussion of diagnosing multiple faults, refer to the journal articles [187, 189].

4.7 Dynamic PCA

The previously discussed PCA monitoring methods implicitly assume that the observations at one time instant are statistically independent to observations at past time instances. For typical chemical processes, this assumption is valid only for long sampling times, *i.e.*, 2 to 12 hours, and suggests that a method taking into account the serial correlations in the data is needed in order to implement a process monitoring method with fast sampling times. A simple method to check whether correlations are present in the data is through the use of an autocorrelation chart of the principal components [189, 224]. If significant autocorrelation is shown in the autocorrelation chart, the following approaches can be used. One approach to address this issue is to incorporate EWMA/CUSUM charts with PCA (see Section 4.8). Another approach is to average the measurements over a number of data points. Alternatively, PCA can be used to take into account the serial correlations by augmenting each observation vector with the previous h observations and stacking the data matrix in the following manner,

$$X(h) = \begin{bmatrix} \mathbf{x}_t^T & \mathbf{x}_{t-1}^T & \cdots & \mathbf{x}_{t-h}^T \\ \mathbf{x}_{t-1}^T & \mathbf{x}_{t-2}^T & \cdots & \mathbf{x}_{t-h-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{t+h-n}^T & \mathbf{x}_{t+h-n-1}^T & \cdots & \mathbf{x}_{t-n}^T \end{bmatrix} \tag{4.44}$$

where $\mathbf{x}_t^T$ is the m-dimensional observation vector in the training set at time interval t. By performing PCA on the data matrix in (4.44), a multivariate

autoregressive (AR), or ARX model if the process inputs are included, is extracted directly from the data [125, 228]. To see this, consider a simple example of a single input single output (SISO) process, which is described by the ARX(h) model

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_h y_{t-h} + \beta_0 u_t + \beta_1 u_{t-1} + \cdots + \beta_h u_{t-h} + e_t \tag{4.45}$$

where y_t and u_t are the output and input at time t, respectively, $\alpha_1, \ldots, \alpha_h, \beta_1, \ldots, \beta_h$ are constant coefficients, and e_t is a white noise process with zero mean [224, 228]. Mathematically, the ARX(h) model states that the output at time t is linearly related to the past h inputs and outputs. With $\mathbf{x}_t^T = [y_t \;\; u_t]$, the matrix $X(h)$ in (4.44) becomes:

$$X(h) = \begin{bmatrix} y_t & u_t & y_{t-1} & u_{t-1} & \cdots & y_{t-h} & u_{t-h} \\ y_{t-1} & u_{t-1} & y_{t-2} & u_{t-2} & \cdots & y_{t-h-1} & u_{t-h-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{t+h-n} & u_{t+h-n} & y_{t+h-n-1} & u_{t+h-n-1} & \cdots & y_{t-n} & u_{t-n} \end{bmatrix}. \tag{4.46}$$

The ARX(h) model indicates that the first column of $X(h)$ is linearly related to the remaining columns. In the noise-free case the matrix formed in (4.46) would be rank deficient (*i.e.*, not full rank). When PCA is applied to $X(h)$ using (4.3), the eigenvector corresponding to the zero eigenvalue would reveal the ARX(h) correlation structure [125]. In the case where noise is present, the matrix will be nearly rank deficient. The eigenvector corresponding to a nearly zero eigenvalue will be an approximation of the ARX(h) correlation structure [125, 165].

Note that the Q statistic is then the squared prediction error of the ARX model. If enough lags h are included in the data matrix, the Q statistic is statistically independent from one time instant to the next, and the threshold (4.22) is theoretically justified. This method of applying PCA to (4.44) is referred to as **dynamic PCA** (DPCA). When multi-model PCAm is used with (4.44) for diagnosing faults, it will be referred to as **DPCAm**. Note that a statistically justified method can be used for selecting the number of lags h to include in the data for our studies (see Section 7.5). The method for automatically determining h described in [125] is not used here. Experience indicates that $h = 1$ or 2 is usually appropriate when DPCA is used for process monitoring. The fault detection and diagnosis measures for static PCA generalize directly to DPCA. For fault identification, the measures for each observation variable can be calculated by summing the values of the measures corresponding to the previous h lags.

It has been stated that in practice the presence of serial correlations in the data does not compromise the effectiveness for the static PCA method when there are enough data to represent all the normal variations of the

process [113]. Irrespective of this claim, including lags in the data matrix as in (4.44) can result in the PCA representation correlating more information. Therefore, as long as there are enough data to justify the added dimensionality of including h lags, DPCA is expected to perform better than PCA for detecting faults from serially correlated data, and this has been confirmed by testing PCA and DPCA on the Tennessee Eastman problem [125].

4.8 Other PCA-based Methods

The EWMA and CUSUM charts have been generalized to the multivariate case [29, 139, 144, 180, 236, 76], and these generalizations can be applied to the PCA-based T^2 statistic in (4.12). Applying these methods can result in increased sensitivity and robustness of the process monitoring scheme, as discussed in Section 2.3. EWMA and CUSUM charts use data from consecutive observations. If a large number of observations is required, an increase in the detection delay can be expected.

The process monitoring measures discussed so far are for continuous processes. Process monitoring measures for batch processes have been developed with the most heavily studied being **multiway PCA**, [169, 228, 24]. Multiway PCA is a three dimensional extension of the PCA approach. The three dimensions of the array represent the observation variables, the time instances, and the batches, respectively, whereas PCA methods for continuous processes contain only two dimensions, the observation variables and the time instances. Details and applications of multiway PCA are provided in the references [169, 228, 24].

PCA is a linear dimensionality reduction technique, which ignores the nonlinearities that may exist in the process data. Chemical processes are inherently nonlinear; therefore, in some cases nonlinear methods for process monitoring may result in better performance compared to the linear methods. Kramer [114] has generalized PCA to the nonlinear case by using autoassociative neural networks (this is called **nonlinear Principal Component Analysis**). Dong and McAvoy [38] have developed a nonlinear PCA approach based on principal curves and neural networks that produce independent principal components. It has been shown that for certain data nonlinearities these nonlinear PCA neural networks are able to capture more variance in a smaller dimension compared to the linear PCA approach. A comparison of three neural network approaches to process monitoring has been made [42]. Neural networks can also be applied in a pattern classification system to capture the nonlinearities in the data. A text on using neural networks as a pattern classifier is **Neural Networks for Pattern Recognition** by Bishop [15]. Although neural networks potentially can capture more information in a smaller dimensional space than the linear dimensionality reduction techniques, an accurate neural network typically requires much more data and computational time to train, especially for large scale systems.

4.9 Homework Problems

1. Read an article on the use of multiway PCA (e.g., [169, 228, 24]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how the statistics are computed. Formulate both fault detection and diagnosis versions of the algorithm. For what types of processes are these algorithms suited? Provide some hypothetical examples.
2. Describe in detail how to blend PCA with CUSUM and EWMA, including the equations for the thresholds.
3. Read an article on the use of PCA for diagnosing sensor faults (e.g., [43, 164]) and write a report describing in detail how the technique is implemented and applied. Compare and contrast the techniques described in the paper with the techniques described in this book.
4. Read an article on the application of nonlinear PCA (e.g., [114, 38]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how the statistics are computed. For what types of processes are these algorithms suited? Provide some hypothetical examples.
5. Prove the properties 1-4 given below Equation 4.6.
6. Section 5 of [101] describes several alternatives to the Q statistic for quantifying deviations outside of those quantified by the T^2 statistic. Describe these statistics in detail, including their thresholds, advantages, and disadvantages. [Note: one of the statistics is closely related to the T_r^2 statistic in Chapter 7.]
7. Apply PCA to the original Class 3 data set reported by Fisher [50], and construct Figure 4.2 including the confidence ellipsoid. Now reapply PCA and reconstruct the figure for the case where the outlier at $(t_1, t_2) = (2.5, 5.6)$ is removed from the Class 3 data set. Compare the confidence ellipsoids obtained in the two cases. Comment on the relative importance of removing the outlier from the Class 3 data set before applying PCA.
8. Read the article [63] which describes the use of structured residuals and PCA to isolate and diagnose faults, and write a report describing in detail how the technique is implemented and applied. Compare and contrast the approach with the techniques described in this book.

CHAPTER 5
FISHER DISCRIMINANT ANALYSIS

5.1 Introduction

In the pattern classification approach to fault diagnosis outlined in Chapter 3, it was described how the dimensionality reduction of the feature extraction step can be a key factor in reducing the misclassification rate when a pattern classification system is applied to new data (data independent of the training set). The dimensionality reduction is especially important when the dimensionality of the observation space is large while the numbers of observations in the classes are relatively small. A PCA approach to dimensionality reduction was discussed in the previous chapter. Although PCA contains certain optimality properties in terms of fault detection, it is not as well-suited for fault diagnosis because it does not take into account the information between the classes when determining the lower dimensional representation. **Fisher Discriminant Analysis** (FDA), a dimensionality reduction technique that has been extensively studied in the pattern classification literature, takes into account the information between the classes and has advantages over PCA for fault diagnosis.

FDA provides an optimal lower dimensional representation in terms of discriminating among classes of data [41]. Although FDA is only slightly more complex than PCA, it has not yet found extensive use in the process industries for diagnosing faults [26, 192]. This is interesting, since FDA has advantages over PCA, when the primary goal is to discriminate among faults. We suspect that part of the reason that FDA is less popular than PCA in the process industries is that more chemical engineers read the statistics literature (where PCA is dominant) than the pattern classification literature (where FDA is dominant).

This chapter begins in Section 5.2 by defining FDA and presenting some of its optimality properties for pattern classification. An information criterion for FDA is developed in Section 5.3 for automatically determining the order of dimensionality reduction. In Section 5.4, it is described how FDA can be used for fault detection and diagnosis. PCA and FDA are compared in Section 5.5 both theoretically and in application to some data sets. Section 5.6 describes **dynamic FDA** (DFDA), an approach based on FDA that takes into account serial (temporal) correlations in the data.

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

5.2 Fisher Discriminant Analysis

For fault diagnosis, data collected from the plant during specific faults are categorized into classes, where each class contains data representing a particular fault. FDA is a linear dimensionality reduction technique, optimal in terms of maximizing the separation amongst these classes [41]. It determines a set of projection vectors, ordered in terms of maximizing the scatter between the classes while minimizing the scatter within each class.

Define n as the number of observations, m as the number of measurement variables, p as the number of classes, and n_j as the number of observations in the j^{th} class. Represent the vector of measurement variables for the i^{th} observation as $\mathbf{x_i}$. If the training data for all classes have already been stacked into the matrix $X \in \mathcal{R}^{n \times m}$ as in (2.5), then the transpose of the i^{th} row of X is the column vector $\mathbf{x_i}$.

To understand Fisher Discriminant Analysis, first we need to define various matrices that quantifying the total scatter, the scatter within classes, and the scatter between classes. The **total-scatter matrix** is [41, 88]

$$S_t = \sum_{i=1}^{n} (\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^T \tag{5.1}$$

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

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i}. \tag{5.2}$$

With $\mathcal{X}_j$ defined as the set of vectors $\mathbf{x_i}$ which belong to the class j, the **within-scatter matrix** for class j is

$$S_j = \sum_{\mathbf{x_i} \in \mathcal{X}_j} (\mathbf{x_i} - \bar{\mathbf{x}}_\mathbf{j})(\mathbf{x_i} - \bar{\mathbf{x}}_\mathbf{j})^T \tag{5.3}$$

where $\bar{\mathbf{x}}_\mathbf{j}$ is the mean vector for class j:

$$\bar{\mathbf{x}}_\mathbf{j} = \frac{1}{n_j} \sum_{\mathbf{x_i} \in \mathcal{X}_j} \mathbf{x_i}. \tag{5.4}$$

The **within-class-scatter matrix** is

$$S_w = \sum_{j=1}^{p} S_j \tag{5.5}$$

and the **between-class-scatter matrix** is

$$S_b = \sum_{j=1}^{p} n_j (\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{j} - \bar{\mathbf{x}})^T. \tag{5.6}$$

The total-scatter matrix is equal to the sum of the between-scatter matrix and the within-scatter matrix [41],

$$S_t = S_b + S_w. \tag{5.7}$$

The objective of the first FDA vector is to maximize the scatter between classes while minimizing the scatter within classes:

$$\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^T S_b \mathbf{v}}{\mathbf{v}^T S_w \mathbf{v}} \tag{5.8}$$

assuming invertible S_w where $\mathbf{v} \in \mathcal{R}^m$. The second FDA vector is computed so as to maximize the scatter between classes while minimizing the scatter within classes among all axes perpendicular to the first FDA vector, and so on for the remaining FDA vectors. It can be shown that the projection vectors for FDA can be calculated by computing the stationary points of the optimization problem (5.8) [41, 88]. The FDA vectors are equal to the eigenvectors $\mathbf{w_k}$ of the generalized eigenvalue problem

$$S_b \mathbf{w_k} = \lambda_k S_w \mathbf{w_k} \tag{5.9}$$

where the eigenvalues λ_k indicate the degree of overall separability among the classes by projecting the data onto $\mathbf{w_k}$. Any software package that does matrix manipulations, such as MATLAB [70, 71] or IMSL [89], has subroutines for computing the generalized eigenvalues and eigenvectors. Because the direction and not the magnitude of $\mathbf{w_k}$ is important, the Euclidean norm (square root of the sum of squares of each element) of $\mathbf{w_k}$ can be chosen to be equal to 1 ($||\mathbf{w_k}|| = 1$).

The FDA vectors can be computed from the generalized eigenvalue problem as long as S_w is invertible. This will almost always be true provided that the number of observations n is significantly larger than the number of measurements m (the case in practice). Since S_w is expected to be invertible for applications of FDA to fault diagnosis, methods to calculate the FDA vectors for the case of non-invertible S_w are only cited here [25, 84, 207].

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

It is useful to write the goal of FDA more explicitly in terms of a projection. Define the matrix $W_p \in \mathcal{R}^{m\times(p-1)}$ with the $p-1$ FDA vectors as columns. Then the projection of the data from m-dimensional space to $(p-1)$-dimensional space is described by

$$\mathbf{z_i} = W_p^T \mathbf{x_i} \tag{5.10}$$

where $\mathbf{z_i} \in \mathcal{R}^{(p-1)}$. FDA computes the matrix W_p such that data $\mathbf{x_1}, \ldots, \mathbf{x_n}$ for the p classes are optimally separated when projected into the $p-1$ dimensional space. In the case where p is equal to 2, this is equivalent to projecting the data onto a line in the direction of the vector $\mathbf{w}$, for which the projected data are the best separated.

5.3 Reduction Order

No reduction of dimensionality would be needed if the covariance matrix and mean vector were known exactly (see Section 3.3). Errors in the sample covariance matrix (2.6) occur in practice, however, and the dimensionality reduction provided by FDA may be necessary to reduce the misclassification rate when the pattern classification system is applied to new data (data independent of the training set). A popular method for selecting the reduction order for dimensionality reduction methods is to use **cross-validation** [228, 61]. This approach separates the data into multiple sets: the training set, and the testing (or validation) set. The dimensionality reduction procedure is applied to the data in the training set, and then its performance is evaluated by applying the reduced dimension model to the data in the testing set for each reduction order. The reduction order is selected to optimize the performance based on the testing set. For example, if the goal is fault diagnosis, the order of the reduced model would be specified by minimizing the misclassification rate of the testing set.

Cross-validation is not always practical in fault diagnosis applications because there may not be enough data to separate into two sets. In this situation, it is desirable to determine the order of the dimensionality reduction using all the data in the training set. Variations on cross-validation that split the data into larger numbers of sets (such as "leave-one-out" cross-validation [229]) are computationally expensive.

As discussed in Section 3.3, the error of a model can be minimized by choosing the number of independent parameters so that it optimally trades off the bias and variance contributions on the mean-squared error. In an effort to minimize the mean-squared error, criteria in the form

$$\text{(prediction error term)} + \text{(model complexity term)} \tag{5.11}$$

have been minimized to determine the appropriate model order [137]. The **Akaike's information criterion** (AIC), popularly applied in system identification for optimally selecting the model order (for an example, see Section 7.6), can be derived in the form (5.11) [137]. In (5.11), the **prediction error term** is a function of the estimated model parameters and the data in the training set, and the **model complexity term** is a function of the number

of independent parameters and the amount of data in the training set. In system identification, the prediction error term is usually chosen as the average squared prediction-error for the model, but in general, the choice of the complexity term is subjective [137].

A strength of the AIC is that it relies only on information in one set of data (the training data), unlike cross-validation which requires either additional data or a partitioning of the original data set. A criteria in the form (5.11) can be developed for automatically selecting the order for FDA using the information only in the training set [26, 192]. The order can be determined by computing the dimensionality a that minimizes the information criterion

$$f_m(a) + \frac{a}{\tilde{n}} \tag{5.12}$$

where $f_m(a)$ is the misclassification rate (the proportion of misclassifications, which is between 0 and 1) for the training set by projecting the data onto the first a FDA vectors, and $\tilde{n}$ is the average number of observations per class. The misclassification rate of the training set, $f_m(a)$, indicates the amount of information contained in the first a FDA vectors beneficial for pattern classification. While the misclassification rate of the training set typically decreases as a increases, for new data (data independent of the training set), the misclassification rate initially decreases and then increases above a certain order due to overfitting the data. The model complexity term $a/\tilde{n}$ is added in (5.12) to penalize the increase of dimensionality.

The scaling of the reduction order a by the average number of observations per class, $\tilde{n}$, has some intuitive implications. To illustrate this, consider the case where the number of observations in each class is the same, $n_j = \tilde{n}$. It can be shown using some simple algebra that the inclusion of the $a/\tilde{n}$ term in (5.12) ensures that the order selection procedure produces a value for a less than or equal to $\tilde{n}$. In words, this constraint prevents the lower dimensional model from having a higher dimensionality than justified by the number of observations in each class.

The model complexity term $a/\tilde{n}$ can also be interpreted in terms of the total number of misclassifications per class. Defining $m(a)$ as the total number of misclassifications in the training set for order a and assuming that $n_j = \tilde{n}$, the information criterion (5.12) can be written as

$$\frac{m(a)}{p\tilde{n}} + \frac{a}{\tilde{n}} \tag{5.13}$$

where $n = p\tilde{n}$ is the total number of observations. Let us consider the case where it is to be determined whether a reduction order of $a+1$ should be preferred over a reduction order of a. Using the information criterion (5.13) and recalling that a smaller value for the information criterion is preferred, a reduction order of $a+1$ is preferred if

$$\frac{m(a+1)}{p\tilde{n}} + \frac{a+1}{\tilde{n}} < \frac{m(a)}{p\tilde{n}} + \frac{a}{\tilde{n}}. \tag{5.14}$$

This is equivalent to

$$\frac{m(a)}{p} - \frac{m(a+1)}{p} > 1. \tag{5.15}$$

The complexity term does not allow the reduction order to be increased merely by decreasing the number of misclassifications, but only if the decrease in the total number of misclassifications *per class* is greater than 1.

The above analyses indicate that the scaling of a in the model complexity term $a/\tilde{n}$ in the information criterion (5.12) is reasonable. This is confirmed by application in Chapter 10 (for example, see Figure 10.21, where the information criterion correctly captures the shape and slope of the misclassification rate curves for the testing sets).

5.4 Fault Detection and Diagnosis

When FDA is applied for pattern classification, the dimensionality reduction technique is applied to the data in *all* the classes simultaneously. More precisely, denote $W_a \in \mathcal{R}^{m \times a}$ as the matrix containing the eigenvectors $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_a}$ computed from (5.9). The discriminant function can be derived from (3.6) to be [60]

$$\begin{aligned} g_j(\mathbf{x}) = &-\tfrac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_\mathbf{j})^T W_a \left(\tfrac{1}{n_j - 1} W_a^T S_j W_a\right)^{-1} W_a^T (\mathbf{x} - \bar{\mathbf{x}}_\mathbf{j}) + \ln(p_i) \\ &-\tfrac{1}{2} \ln\left[\det\left(\tfrac{1}{n_j - 1} W_a^T S_j W_a\right)\right] \end{aligned} \tag{5.16}$$

where S_j, $\bar{\mathbf{x}}_\mathbf{j}$, and n_j are defined in (5.3) and (5.4). In contrast to PCA1 (see Section 4.6), FDA uses the class information to compute the reduced dimensional space, so that the discriminant function (5.16) exploits that class information to a far greater degree than can be done by PCA. In contrast to PCAm, FDA utilizes *all* p fault class information when evaluating the discriminant function or each class.

FDA can also be applied to *detect* faults by defining an additional class of data, that collected during in-control operations, to the fault classes. However, since this information will be unable to detect faults which occur outside of the lower dimensional space defined by the FDA vectors, this method can be insensitive to faults not represented in the training set. As discussed in Chapter 4, faults not represented in the training set that significantly affect the observations (measurements) can be detected with either the PCA Q or T^2 statistics.

As mentioned in Section 5.2, only the first $p - 1$ eigenvectors in FDA maximize the scatter between the classes while minimizing the scatter within each class. The rest of the $m - p + 1$ eigenvectors corresponding to the zero eigenvalues are not ordered by the FDA objective (5.8). The ranking of these

generalized eigenvectors is determined by the particular software package implementing the eigenvalue decomposition algorithm, which does not order the eigenvectors in a manner necessarily useful for classification. However, more than $p-1$ dimensions in a lower dimensional space may be useful for classification, and a procedure to select vectors beyond the first $p-1$ FDA vectors can be useful. Here two methods are described which use PCA to compute additional vectors for classification.

One method is to use FDA for the space defined by the first $p-1$ eigenvectors, and to use the PCA1 vectors for the rest of the $m-p+1$ vectors, ordered from the PCA vectors associated with the highest variability to the vectors associated with the lower variability. If the reduction order $a \leq p-1$, Equation 5.16 is used directly. If $a \geq p$, the alternative discriminant function is used:

$$g_j(\mathbf{x}) = -\tfrac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_j)^T W_{mix,a}\left(\tfrac{1}{n_j-1} W_{mix,a}^T S_j W_{mix,a}\right)^{-1} W_{mix,a}^T(\mathbf{x}-\bar{\mathbf{x}}_j) - \tfrac{1}{2}\ln\left[\det\left(\tfrac{1}{n_j-1} W_{mix,a}^T S_j W_{mix,a}\right)\right] + \ln(p_i) \tag{5.17}$$

where $W_{mix,a} = [W_{p-1}\ P_{a-p+1}]$, and P_{a-p+1} is the first $a-p+1$ columns of the PCA1 loading matrix P (defined in Section 4.6). When this method is used for diagnosing faults, it will be referred to as the **FDA/PCA1 method.** Recall from Section 4.2 that the variances associated with the loading vectors in PCA are ranked in descending order. Given that the vectors from PCA1 can be useful in a classification procedure (see Section 4.6), incorporating the first $a-p+1$ PCA1 loading vectors into the FDA/PCA1 method may provide additional information for discriminating amongst classes.

Another method to define an additional $m-p+1$ vectors is to apply PCA1 to the residual space of FDA, defined by

$$R = X(I - W_{p-1}W_{p-1}^T). \tag{5.18}$$

As before, if the reduction order $a \leq p-1$, Equation 5.16 is used directly. If $a \geq p$, then the alternative discriminant function (5.17) is used with $W_{mix,a} = [W_{p-1}\ \bar{P}_{a-p+1}]$, where $\bar{P}_{a-p+1}$ is the first $a-p+1$ columns of the PCA1 loading matrix when PCA is applied to R. This method for diagnosing faults will be referred to as the **FDA/PCA2 method.**

5.5 Comparison of PCA and FDA

Here the PCA and FDA dimensionality reduction techniques are compared via theoretical and graphical analyses for the case where PCA is applied to all the data in all the classes together (PCA1 in Section 4.6). This highlights the geometric differences between the two dimensionality reduction procedures. It is also shown how using FDA can result in superior fault diagnosis than when PCA is applied.

The optimization problems for PCA and FDA have been stated mathematically in (4.1) and (5.8), respectively. It can be shown that the PCA loading vectors and FDA vectors can also be calculated by computing the stationary points of the optimization problems

$$\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^T S_t \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \tag{5.19}$$

and

$$\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^T S_t \mathbf{v}}{\mathbf{v}^T S_w \mathbf{v}}, \tag{5.20}$$

respectively. Equations 5.19 and 5.20 indicate that the PCA and FDA vectors are identical for the case when $S_w = \sigma I$ where $\sigma > 0$. One case in which this situation occurs if the data in each class can be described by a uniformly distributed ball (*i.e.*, circle in 2-D space and sphere in 3-D space), even if the balls are of distinct sizes. Differences between the two techniques can occur only if there is elongation in the data used to describe any one of the classes. These elongated shapes occur for highly correlated data sets (see Figure 4.2), typical for data collected from chemical processes. Therefore, when PCA and FDA are applied in the same manner to process data, the PCA loading vectors and FDA vectors are expected to be significantly different, and the differing objectives, (5.19) and (5.20), suggest that FDA will be significantly better for discriminating among classes of faults.

Figure 5.1 illustrates a difference between PCA and FDA that can occur when the distribution of the data in the classes are somewhat elongated. The first FDA vector and PCA loading vector are nearly perpendicular, and the projection of the data onto the first FDA vector is much better able to separate the data in the two classes than the projection of the data onto the first PCA loading vector.

The projection of the experimental data taken from [50, 25] onto the first two PCA and FDA loading vectors are shown in Figure 5.2. The within-class-scatter matrix and between-class-scatter matrix are calculated as

$$S_w = \begin{bmatrix} 56.8 & 37.3 & 16.4 & 9.17 \\ 37.3 & 88.4 & 10.1 & 17.1 \\ 16.4 & 10.1 & 8.75 & 4.64 \\ 9.17 & 17.1 & 4.64 & 22.8 \end{bmatrix} \tag{5.21}$$

and

$$S_b = \begin{bmatrix} 92.2 & -55.7 & 113 & 108 \\ -55.7 & 60.6 & -75.3 & -65.6 \\ 113 & -75.2 & 140 & 133 \\ 108 & -65.6 & 132 & 126 \end{bmatrix} \tag{5.22}$$

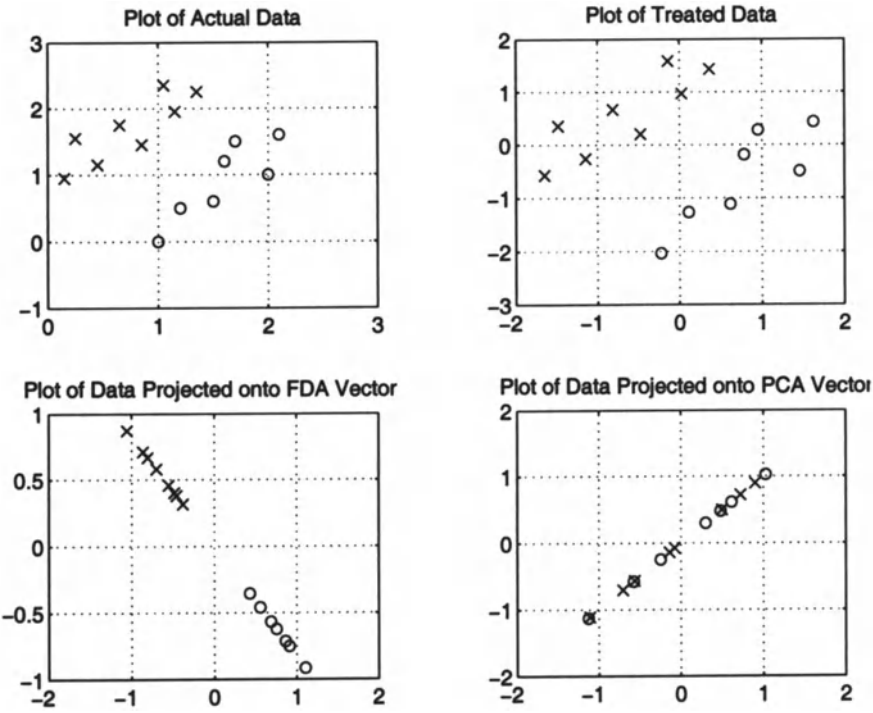

Fig. 5.1. A comparison of PCA and FDA for the projection of the data in classes 'x' and 'o' onto the first FDA vector and PCA loading vector

respectively. Solving (5.9), we have $p - 1 = 2$ eigenvectors associated with nonzero eigenvalues, which are

$$\mathbf{w_1} = \begin{bmatrix} 0.15 \\ 0.12 \\ -0.96 \\ -0.18 \end{bmatrix} \tag{5.23}$$

and

$$\mathbf{w_2} = \begin{bmatrix} -0.13 \\ -0.70 \\ -0.15 \\ 0.68 \end{bmatrix}, \tag{5.24}$$

and the corresponding eigenvalues are $\lambda_1 = 27$ and $\lambda_2 = 0.24$, respectively. The large λ_1 value indicates that there is a large separation of the class means

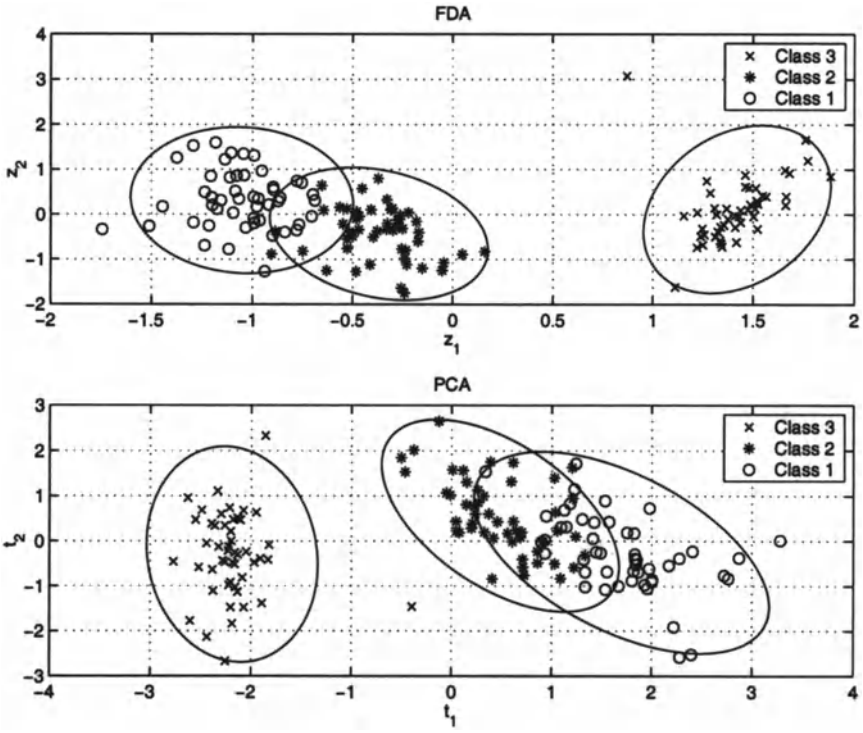

Fig. 5.2. The projections of experimental data [50, 25] for three classes onto the first two FDA and PCA loading vectors, respectively

relative to the class variances on z_1 (see Figure 5.2). Indeed the average values of z_1 for the 3 classes are -1.0, -0.37, and 1.42. The small λ_2 value indicates that the overall separation of the class means relative to the class variances is small in the z_2 direction. The average values of z_2 for the 3 classes are 0.30, -0.43, and 0.12.

The 95% elliptical confidence region for each class can be approximated by solving (3.8) with T_i^2 set to 6.64. The T_i^2 threshold is the same as in the example we showed in Chapter 4. Data falling in the intersection of the two elliptical confidence regions can result in misclassification. The degree of overlap between the confidence regions for Classes 1 and 2 is greater for PCA than for FDA (49 points vs. 17 points), indicating that the misclassification rates for PCA would be higher.

While the elliptical confidence region can be used to illustrate the qualitative classification performance, the discriminant function (5.16) can be used to determine the exact misclassification rates for the experiment data [50, 25]. The results are illustrated in Table 5.1 for different FDA reduction

orders. Although Class 1 and Class 2 data overlap to some extent (see Figure 5.2), the discriminant function (5.16) is able to correctly classify most of the data points. Indeed, no more than 3 out of 50 data points are misclassified regardless of the order selection (see Table 5.1).

Table 5.1. The misclassification rates of the experimental data [50, 25] for FDA

Order (a)	1	2	3	4
Class 1 Misclassifications	0	0	0	0
Class 2 Misclassifications	0.06	0.06	0.04	0.06
Class 3 Misclassifications	0.06	0.02	0.02	0.02
Overall Misclassifications	0.04	0.027	0.02	0.027
AIC	0.06	0.067	0.08	0.11

For the training data, FDA produced the minimum overall misclassification rate when the reduction order is 3. Although this order is optimal for the training set, it may not be the best order when the FDA model is applied to new data. The AIC can be used to estimate the optimal reduction order for FDA when applied to new data. The minimum value for the AIC (as reported in Table 5.1) is for the reduction order $a = 1$.

This example is effective at illustrating the difference in the objectives between PCA and FDA. By comparing the limits of the horizontal and vertical axes and visually inspecting the data, it is clear that the span of the PCA projection is larger than the FDA projection. While PCA is better able to separate the data as a whole, FDA is better able to separate the data among the classes (*, o, x). This is evident in the degree of overlap between '*' and 'o' data regions in the two plots, in which the data points '*' and 'o' barely overlap for the FDA projection, while there is a clear intermingling of data for the PCA projection.

The overall misclassification rates of the experimental data using the FDA, FDA/PCA1, FDA/PCA2, PCA1, PCAm, and MS classification methods are shown in Table 5.2 for various reduction orders. The overall misclassification rates for FDA, FDA/PCA1, and FDA/PCA2 were the same except at $a = 3$. The FDA vectors corresponding to the two nonzero eigenvalues are very effective in discriminating the three classes. At $a = 2$, the overall misclassification rate is 0.027 (*i.e.,* 146 out of 150 data points were correctly classified). When the $p - 1$ FDA vectors are effective in discriminating classes, FDA/PCA1 and FDA/PCA2 will not decrease the overall misclassification rates further. Although the potential benefit of using FDA/PCA1 and FDA/PCA2 over FDA is not shown in the example, FDA/PCA1 and FDA/PCA2 can produce lower overall misclassification rates than FDA as shown in Section 10.8.

For any reduction order, the FDA and mixed FDA/PCA methods had a lower overall misclassification rate than either PCA method. This agrees with

Table 5.2. Overall misclassification rates of the experimental data [50, 25] using PCA1, PCAm, FDA, and FDA/PCA methods

Order (a)	1	2	3	4
FDA	0.040	0.027	0.020	0.027
FDA/PCA1	0.040	0.027	0.027	0.027
FDA/PCA2	0.040	0.027	0.027	0.027
PCA1	0.080	0.087	0.033	0.027
PCAm	0.17	0.15	0.11	0.11
MS	–	–	–	0.027

earlier comments that FDA can do a much better job at diagnosing faults than PCA. At any reduction order, PCA1 gave lower overall misclassification rates than PCAm. This supports our discussion in Section 4.6 that PCA1 will usually produce a better PCA representation for diagnosing faults. For $a = 4$, PCA1, FDA1, FDA/PCA1, and FDA/PCA2 gave the same overall misclassification rates as MS. As discussed in Section 4.6, MS is the same as PCA1 when all orders are included. This does not generally hold for the FDA methods.

For this particular example, dimensionality reduction was not necessary for providing low misclassification rates. This is because the classification methods were only being applied to training data. The benefit of dimensionality reduction is most apparent for the classification of new data. Applications of the methods to simulated plant data in Chapter 10 illustrate this point.

5.6 Dynamic FDA

As mentioned in Section 4.8, CUSUM and EWMA charts can be used to capture the serial correlations in the data for PCA. CUSUM and EWMA charts can also be generalized for FDA. The pattern classification method for fault diagnosis discussed in Chapter 3 and Section 5.4 can be extended to take into account the serial (temporal) correlations in the data, by augmenting the observation vector and stacking the data matrix in the same manner as (4.44), this method will be referred to as **dynamic FDA** (DFDA). This enables the pattern classification system to use more information in classifying the observations. Since the information contained in the augmented observation vector is a superset of the information contained in a single observation vector, it is expected from a theoretical point of view that the augmented vector approach can result in better performance. However, the dimensionality of the problem is increased by stacking the data, where the magnitude of the increase depends on the number of lags h. This implies that more data may be required to determine the mean vector and covariance matrix to the same level of accuracy for each class. In practice, augmenting the observation

vector is expected to perform better when there is both significant serial correlation and there are enough data to justify the larger dimensionality. Since the amount of data n is usually fixed, performing dimensionality reduction using FDA becomes even more critical to the pattern classification system when the number of lags h is large. The application of FDA/PCA1 to (4.44) will be referred to as **DFDA/DPCA1**, and the developments in this chapter for FDA readily apply to DFDA and DFDA/DPCA1.

5.7 Homework Problems

1. In Sections 5.4 and 5.5 it was discussed how the best use of the PCA techniques (T^2 and Q statistics) can outperform FDA for fault detection, while the best use of FDA techniques should outperform PCA for fault diagnosis. Construct data sets (in which you apply both PCA and FDA) to illustrate the key reasoning underlying these conclusions.
2. Define a residual-based statistic for FDA similar to the Q statistic used in PCA. Would the FDA-based Q statistic be expected to outperform the PCA-based Q statistic for fault detection? Construct data sets (in which you apply both PCA and FDA) to illustrate the key reasoning underlying these conclusions. How does this answer depend on the reduction order for FDA?
3. Derive Equations 5.19 and 5.20.
4. Describe in detail how to blend FDA with CUSUM and EWMA, including the equations for the thresholds.
5. Write a one page technical summary of the classic paper by Fisher on discriminant analysis [50]. Compare the equations derived by Fisher to the equations in this chapter. Explain any significant differences.
6. Peterson and Mattson [179] consider more general criteria for dimensionality reduction. Compare their criteria to the Fisher criterion. What are the advantages and disadvantages of each? For what types of data would you expect one criterion to be preferable over the others?
7. Show that the FDA vectors are not necessarily orthogonal (hint: the easiest way to show this is by example). Compare FDA with PLS and PCA in this respect.

CHAPTER 6
PARTIAL LEAST SQUARES

6.1 Introduction

Partial Least Squares (PLS), also known as **Projection to Latent Structures**, is a dimensionality reduction technique for maximizing the covariance between the predictor (independent) matrix X and the predicted (dependent) matrix Y for each component of the reduced space [61, 235]. A popular application of PLS is to select the matrix Y to contain only product quality data which can even include off-line measurement data, and the matrix X to contain all other process variables [144]. Such inferential models (also known as soft sensors) can be used for the on-line prediction of the product quality data [149, 155, 156], for incorporation into process control algorithms [106, 181, 182], as well as for process monitoring [144, 181, 182]. Discriminant PLS selects the matrix X to contain all process variables and selects the Y matrix to focus PLS on the task of fault diagnosis [26].

PLS computes loading and score vectors that are correlated with the predicted block while describing a large amount of the variation in the predictor block [228]. If the predicted block has only one variable, the PLS dimensionality reduction method is known as PLS1; if the predicted block has multiple variables, the dimensionality reduction method is known as PLS2. PLS requires calibration and prediction steps. The most popular algorithm used in PLS to compute the parameters in the calibration step is known as **Non-Iterative Partial Least Squares** (NIPALS) [61, 228]. Another algorithm, known as SIMPLS, can also be used [31]. As mentioned, the predicted blocks used in discriminant PLS and in other applications of PLS are different. In chemometrics and process control applications, where PLS is most commonly applied, the predicted variables are usually measurements of product quality variables. In pattern classification, where discriminant PLS is used, the predicted variables are dummy variables (1 or 0) where '1' indicates an in-class member while '0' indicates a non-class member [6, 32, 170]. In the prediction step of discriminant PLS, discriminant analysis is used to determine the predicted class [170].

Section 6.2 defines the PLS1 and PLS2 algorithms in enough detail to allow the reader to implement these techniques. Section 6.3 discusses the selection of the reduction order. Section 6.4 discusses fault detection, identification, and diagnosis using PLS. The PLS and PCA techniques are compared in

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

Section 6.5. Section 6.6 summarizes several variations of the PLS algorithms for process monitoring.

6.2 PLS Algorithms

PLS requires a matrix $X \in \mathcal{R}^{n \times m}$ and a matrix $Y \in \mathcal{R}^{n \times p}$, where m is the number of predictor variables (the number of measurements in each observation), n is the total number of observations in the training set, and p is the numbers of observation variables in Y. When Y is selected to contain only the product quality variables, then p is the number of product quality variables. When Y is selected as done in discriminant PLS, p is the number of fault classes.

In discriminant PLS, diagnosed data are needed in the calibration. To aid in the description of discriminant PLS, the data in X will be ordered in a particular way. With p fault classes, suppose that there are $n_1, n_2, \ldots, n_p$ observations for each variable in Classes 1, 2, ... , p respectively. Collect the training set data into the matrix $X \in \mathcal{R}^{n \times m}$, as shown in (2.5), so that the first n_1 rows contain data from Fault 1, the second n_2 rows contain data from Fault 2, and so on. Altogether, there are $n_1 + n_2 + \cdots + n_p = n$ rows. There are two methods, known as PLS1 and PLS2, to model the predicted block. In PLS1, each of the p predicted variables is modeled separately, resulting in one model for each class. In PLS2, all predicted variables are modeled simultaneously [150].

In PLS2, the predicted block $Y \in \mathcal{R}^{n \times p}$ contains p product quality variables; in discriminant PLS2, the predicted block $Y \in \mathcal{R}^{n \times p}$ is

$$Y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{p \text{ columns}} \tag{6.1}$$

where each column in Y corresponds to a class. Each element of Y is filled with either *one* or *zero*. The first n_1 elements of Column 1 are filled with

a '1', which indicates that the first n_1 rows of X are data from Fault 1. In discriminant PLS1, the algorithm is run p times, each with the same X, but for each individual column of Y in (6.1).

As mentioned in Section 2.2, data pretreatment is applied first, so that X and Y are mean-centered and scaled. The matrix X is decomposed into a score matrix $T \in \mathcal{R}^{n \times a}$ and a loading matrix $P \in \mathcal{R}^{m \times a}$, where a is the PLS component (reduction order), plus a residual matrix $E \in \mathcal{R}^{n \times m}$:

$$X = TP^T + E. \tag{6.2}$$

The matrix product TP^T can be expressed as the sum of the product of the **score vectors** $\mathbf{t_j}$ (the j^{th} column of T) and the **loading vectors** $\mathbf{p_j}$ (the j^{th} column of P) [61, 228, 105]:

$$X = \sum_{j=1}^{a} \mathbf{t_j p_j}^T + \mathbf{E}. \tag{6.3}$$

Similarly, Y is decomposed into a score matrix $U \in \mathcal{R}^{n \times a}$, a loading matrix $Q \in \mathcal{R}^{p \times a}$, plus a residual matrix $\tilde{F} \in \mathcal{R}^{n \times p}$:

$$Y = UQ^T + \tilde{F}. \tag{6.4}$$

The matrix product UQ^T can be expressed as the sum of the product of the score vectors $\mathbf{u_j}$ (the j^{th} column of U) and the loading vectors $\mathbf{q_j}$ (the j^{th} column of Q):

$$Y = \sum_{j=1}^{a} \mathbf{u_j q_j}^T + \tilde{\mathbf{F}}. \tag{6.5}$$

The decompositions in (6.3) and (6.5) have the same form as that used in PCA (see (4.5)). The matrices X and Y are represented as the sum of a series of rank one matrices. If a is set equal to $\min(m, n)$, then E and $\tilde{F}$ are zero and PLS reduces to ordinary least squares. Setting a less than $\min(m, n)$ reduces noise and collinearity. The goal of PLS is to determine the loading and score vectors which are correlated with Y while describing a large amount of the variation in X.

PLS regresses the estimated Y score vector $\hat{\mathbf{u}}_\mathbf{j}$ to the X score vector $\mathbf{t_j}$ by

$$\hat{\mathbf{u}}_\mathbf{j} = b_j \mathbf{t_j} \tag{6.6}$$

where b_j is the regression coefficient. In matrix form, this relationship can be written

$$\hat{U} = TB \tag{6.7}$$

where $B \in \mathcal{R}^{a \times a}$ is the diagonal regression matrix with $B_{jj} = b_j$, and $\hat{U}$ has $\hat{\mathbf{u}}_\mathbf{j}$ as its columns. Substituting $\hat{U}$ from (6.7) in for U in (6.4), and taking into account that this will modify the residual matrix, gives

$$Y = TBQ^T + F \tag{6.8}$$

where F is the prediction error matrix. The matrix B is selected such that the induced 2-norm of F (the maximum singular value of F [66]), $\|F\|_2$, is minimized [105]. The score vectors $\mathbf{t_j}$ and $\hat{\mathbf{u}}_\mathbf{j}$ are calculated for each PLS factor ($j = 1, 2, ..., a$) such that the covariance between X and Y is maximized at each factor. In PLS1, similar steps are performed, resulting in

$$\mathbf{y_i} = T_i B_i \mathbf{q_i}^T + \mathbf{f_i} \tag{6.9}$$

where $\mathbf{y_i} \in \mathcal{R}^n$ is the i^{th} column of Y, $T_i \in \mathcal{R}^{n \times a}$ is the score matrix, $B_i \in \mathcal{R}^{a \times a}$ is the regression matrix, $\mathbf{q_i} \in \mathcal{R}^a$ is the loading vector, and $\mathbf{f_i} \in \mathcal{R}^n$ is the prediction error vector. Since there are p columns in Y, the range of i is from 1 to p.

Now if the score and loadings matrices for X and Y were calculated separately, then their successive score vectors could be weakly related to each other, so that the regression (6.6) which relates X and Y would result in a poor reduced dimension relationship. The NIPALS algorithm is an iterative approach to computing modified score vectors so that rotated components result which lead to an improved regression in (6.6). It does this by using the score vectors from Y in the calculation of the score vectors for X, and *vice versa.*

For the case of PLS2, the NIPALS algorithm computes the parameters using (6.10) to (6.20) [61, 105, 228]. The first step is the cross regression of X and Y, which are scaled so as to have zero mean and unit variance for each variable. Initialize the NIPALS algorithm using $E_0 = X$ and $F_0 = Y$, $j = 1$, and $\mathbf{u_j}$ equal to any column of F_{j-1}. Equations (6.10)-(6.13) are iteratively computed until convergence, which is determined by comparing $\mathbf{t_j}$ with its value from a previous iteration (the nomenclature $\|\cdot\|$ refers to the vector 2-norm, also known as the Euclidean norm).

$$\mathbf{w_j} = \frac{E_{j-1}^T \mathbf{u_j}}{\|E_{j-1}^T \mathbf{u_j}\|} \tag{6.10}$$

$$\mathbf{t_j} = E_{j-1} \mathbf{w_j} \tag{6.11}$$

$$\mathbf{q_j} = \frac{F_{j-1}^T \mathbf{t_j}}{\|F_{j-1}^T \mathbf{t_j}\|} \tag{6.12}$$

$$\mathbf{u_j} = F_{j-1} \mathbf{q_j} \tag{6.13}$$

Proceed to (6.14) if convergence; return to (6.10) if not. Mathematically, determining $\mathbf{t_1}$, $\mathbf{u_1}$, and $\mathbf{w_1}$ from (6.10) to (6.13) is the same as iteratively

determining the eigenvectors of XX^TYY^T, YY^TXX^T, and X^TYY^TX associated with the largest eigenvalue, respectively [184, 229].

In the second step, $\mathbf{p_j}$ is calculated as

$$\mathbf{p_j} = \frac{E_{j-1}^T \mathbf{t_j}}{\mathbf{t_j}^T \mathbf{t_j}} \tag{6.14}$$

The final values for $\mathbf{p_j}$, $\mathbf{t_j}$, and $\mathbf{w_j}$ are scaled by the norm of $\mathbf{p_{j,old}}$:

$$\mathbf{p_{j,new}} = \frac{\mathbf{p_{j,old}}}{\|\mathbf{p_{j,old}}\|} \tag{6.15}$$

$$\mathbf{t_{j,new}} = \mathbf{t_{j,old}} \|\mathbf{p_{j,old}}\| \tag{6.16}$$

$$\mathbf{w_{j,new}} = \mathbf{w_{j,old}} \|\mathbf{p_{j,old}}\| \tag{6.17}$$

Although it is common to apply the scalings (6.15) to (6.17) in the algorithm [228, 229, 61], the scalings are not absolutely necessary [149]. In particular, the score vectors $\mathbf{t_j}$ used to relate X to Y in (6.6) are orthogonal in either case.

Now that $\mathbf{u_j}$ and $\mathbf{t_j}$ are computed using the above expressions, the regression coefficient b_j that relates the two vectors can be computed from

$$b_j = \frac{\mathbf{u_j}^T \mathbf{t_j}}{\mathbf{t_j}^T \mathbf{t_j}} \tag{6.18}$$

The residual matrices E_j and F_j needed for the next iteration are calculated from

$$E_j = E_{j-1} - \mathbf{t_j} \mathbf{p_j}^T \tag{6.19}$$

and

$$F_j = F_{j-1} - b_j \mathbf{t_j} \mathbf{q_j}^T. \tag{6.20}$$

This removes the variance associated with the already calculated score and loading vectors before computing the score and loading vectors for the next iteration. The entire procedure is repeated for the next factor (commonly called as latent variable [228, 229]) $(j + 1)$ starting from (6.10) until $j = \min(m, n)$.

As discussed in the next section, predictions based on the PLS model can be computed directly from the observation vector and $\mathbf{p_j}, \mathbf{q_j}, \mathbf{w_j}$, and b_j for $j = 1, 2, \ldots, \min(m, n)$. We will also see an alternative approach where the predictions are obtained from the regression matrix $B2_j$ [229, 150]

$$B2_j = W_j (P_j^T W_j)^{-1} (T_j^T T_j)^{-1} T_j^T F_0 \tag{6.21}$$

where the matrices $P_j \in \mathcal{R}^{\min(m,n)\times j}$, $T_j \in \mathcal{R}^{n\times j}$, and $W_j \in \mathcal{R}^{\min(m,n)\times j}$ are formed by stacking the vectors $\mathbf{p_j}, \mathbf{t_j}$, and $\mathbf{w_j}$, respectively. This matrix is saved for $j = 1, 2, \ldots, \min(m, n)$.

The NIPALS algorithm for PLS1 is calculated using (6.22) to (6.27). Initialize the NIPALS algorithm using $E_0 = X$, $j = 1$, and set $i = 1$. The following equations are used:

$$\mathbf{w_{i,j}} = \frac{E_{j-1}^T \mathbf{y_i}}{\|E_{j-1}^T \mathbf{y_i}\|} \tag{6.22}$$

$$\mathbf{t_{i,j}} = E_{j-1}\mathbf{w_{i,j}} \tag{6.23}$$

$$\mathbf{p_{i,j}} = \frac{E_{j-1}^T \mathbf{t_{i,j}}}{\mathbf{t_{i,j}}^T \mathbf{t_{i,j}}} \tag{6.24}$$

After rescaling of $\mathbf{p_{i,j}}, \mathbf{t_{i,j}}$, and $\mathbf{w_{i,j}}$ similarly as in (6.15) to (6.17), the regression coefficient $b_{i,j}$ is computed from

$$b_{i,j} = \frac{\mathbf{y_i}^T \mathbf{t_{i,j}}}{\mathbf{t_{i,j}}^T \mathbf{t_{i,j}}} \tag{6.25}$$

The residuals for the next iteration are calculated as follows

$$E_j = E_{j-1} - \mathbf{t_{i,j}}\mathbf{p_{i,j}}^T \tag{6.26}$$

$$\mathbf{f_{i,j}} = \mathbf{f_{i,j-1}} - b_{i,j}\mathbf{t_{i,j}}q_{i,j} \tag{6.27}$$

where $\mathbf{f_{0,i}} = \mathbf{y_i}$ and $q_{i,j} = 1$. The entire procedure is repeated for the next latent variable $(j + 1)$ starting from (6.22) until $j = \min(m, n)$. After all the parameters for $i = 1$ are calculated, the algorithm is repeated for $i = 2, 3, \ldots, p$.

As discussed in the next section, predictions based on the PLS model can be computed directly from the observation vector and the $\mathbf{p_{i,j}}$, $\mathbf{w_{i,j}}$, and $b_{i,j}$. Alternatively, the predictions are obtained from the regression matrix $B1_j$ [6, 150]

$$B1_j = [\mathbf{b_{1,j}}\, \mathbf{b_{2,j}} \cdots \mathbf{b_{p,j}}] \tag{6.28}$$

where

$$\mathbf{b_{i,j}} = W_{i,j}(P_{i,j}^T W_{i,j})^{-1}(T_{i,j}^T T_{i,j})^{-1} T_{i,j}^T \mathbf{f_{0,j}} \tag{6.29}$$

the matrices $P_{i,j} \in \mathcal{R}^{\min(m,n)\times j}$, $W_{i,j} \in \mathcal{R}^{\min(m,n)\times j}$, and $T_{i,j} \in \mathcal{R}^{n\times j}$ are formed by stacking the vectors $\mathbf{p_{i,j}}$, $\mathbf{w_{i,j}}$, and $\mathbf{t_{i,j}}$, respectively.

6.3 Reduction Order and PLS Prediction

It is important to have a proper number a of PLS factors selected in order to obtain a good prediction, since too high of a number (the maximum theoretical value for a is the rank of X) will cause a magnification of noise and poor process monitoring performance. A standard way to determine the proper reduction order, denoted as c, is to apply cross-validation using the Prediction Residual Sum of Squares (PRESS). The order c is set to be the order at which PRESS is minimum [61]. As discussed previously, the weakness of this approach is that it requires that the data be split into two parts (the training and the testing sets), with the PLS vectors computed based only on the data from the testing set.

In the case of fault diagnosis, an alternative approach is to select the value of c which minimizes the information criterion (5.12). To determine c, the PLS vectors are constructed using all of the data, and then the PLS vectors are applied to all of the data to calculate the misclassification rates for each choice of the reduction order, where the misclassification rate is defined to be the ratio of the number of incorrectly assigned classes to the total number of classifications made (the number of observations in the training set).

For each factor $j = 1, 2, \ldots, \min(m, n)$, the estimated score vector $\hat{\mathbf{t}}_\mathbf{j}$ and matrix residual E_j are

$$\hat{\mathbf{t}}_\mathbf{j} = E_{j-1}\mathbf{w_j} \tag{6.30}$$

$$E_j = E_{j-1} - \hat{\mathbf{t}}_\mathbf{j}\mathbf{p}_\mathbf{j}^T \tag{6.31}$$

where $E_0 = X$. To compute a prediction of the predicted block $Y_{train2,a}$ of the training set using PLS2 with a PLS components:

$$Y_{train2,a} = F_j = \sum_{j=1}^{a} b_j \hat{\mathbf{t}}_\mathbf{j}\mathbf{q}_\mathbf{j}^T. \tag{6.32}$$

For PLS1, the prediction of the predicted block $Y_{train1,a}$ of the training set using PLS1 with a PLS components is computed by

$$Y_{train1,a} = \left[\mathbf{y_{train1,a}}\ \mathbf{y_{train2,a}}\ \cdots\ \mathbf{y_{trainp,a}}\right] \tag{6.33}$$

where

$$\mathbf{y_{traini,a}} = \mathbf{f_{i,j}} = \sum_{j=1}^{a} \mathbf{b_{i,j}}\hat{\mathbf{t}}_{\mathbf{i,j}} q_{i,j} \tag{6.34}$$

Alternatively, a prediction of PLS2 with a PLS components is given by the regression equation [6]:

$$Y_{train2,a} = XB2_a \tag{6.35}$$

The above equation is also used for the alternative prediction of PLS1 by replacing $B2_a$ with $B1_a$.

6.4 Fault Detection, Identification, and Diagnosis

An approach investigated in the chemical engineering community is to apply PLS in the same manner as PCA, selecting the Y matrix to be the product quality variables. Monitoring the PLS scores in this way has the advantage over the PCA scores in that the PLS scores will only monitor variations in X which are known to be related to the product quality variables. All the fault detection, identification, and diagnosis techniques for PCA can be applied in exactly the same way for PLS (e.g., including the Q and T^2 statistics, contribution plots, and discriminant analysis) [113, 228].

The use of discriminant PLS for fault diagnosis requires significantly more explanation. In discriminant PLS, the rows of Y_{train} will not have the form $[0, 0, 0, \ldots, 1, \ldots, 0, 0]$, which requires a method for assigning the class c_k to each observation k. One method is to assign c_k to correspond to the column index whose element is the closest to one [170]. A second method is to assign c_k to correspond to the column whose element has the maximum value.

The term **overestimation** refers to the case where the element of Y_{train} for an in-class member > 1 or the element of Y_{train} for a non-class member > 0. **Underestimation** is where the element of Y_{train} for an in-class member < 1 or the element of Y_{train} for a non-class member < 0. Both assignment methods give accurate classifications in the ideal case, that is, when none of the elements of Y_{train} are overestimated nor underestimated, and in the case where all of the elements of Y_{train} are underestimated. If all of the elements of Y_{train} are overestimated, then the first assignment method can give high misclassification rates, while the second assignment method will still tend to give good classifications [170]. The second assignment method is preferred because of this wider usefulness.

If some of the elements of Y_{train} are underestimated while others are overestimated, either of the above assignment methods can perform poorly. A method to resolve this problem is to take account of the underestimation and overestimation of Y into a second cycle of PLS algorithm [170]. The NIPALS algorithm is run for the second time for PLS1 and PLS2 by replacing $\mathbf{y_i}$ by $\mathbf{y_{train1,i}}$ and Y by Y_{train2} respectively. To distinguish between the *normal* PLS method and this *adjusted* method, PLS1 and PLS2 are denoted as PLS1_{adj} and PLS2_{adj} respectively. The predicted Y of the training set using PLS1_{adj} and PLS2_{adj}, denoted as $Y_{train1,adj}$ and $Y_{train2,adj}$, are obtained in the similar fashion as PLS1 and PLS2 respectively.

The effectiveness of the algorithm can be determined by applying it to a testing set $X_{test} \in \mathcal{R}^{r \times m}$. The predicted block Y_{test1} of the testing set using PLS1 is calculated using (6.30) to (6.31) and (6.33) to (6.34) by replacing X with X_{test} while the predicted block Y_{test2} of the testing set using PLS2 is calculated using (6.30) to (6.32) by replacing X with X_{test}. The predicted blocks $Y_{test1,adj}$ and $Y_{test2,adj}$ using PLS1_{adj} and PLS2_{adj} respectively are obtained similarly.

To illustrate the application of discriminant PLS2, the same experimental data set [50, 25] is used as in Chapter 4. The predictor matrix X is formed by using data from all three classes, where $n = 150$ and $m = 4$; the corresponding predicted matrix Y is formed as in (6.1), where $p = 3$. The matrices X and Y are first autoscaled. The NIPALS algorithm is initialized using $E_0 = X$, $F_0 = Y$, and $\mathbf{u_1}$ arbitrarily set to the third column of Y. After 12 iterations of (6.10)-(6.13), the score vector $\mathbf{t_1}$ converges with an error of less than 10^{-10}. The following vectors are then obtained:

$$\begin{aligned} \mathbf{w_1} &= [0.48\ -0.32\ 0.60\ 0.56]^T, \\ \mathbf{p_1} &= [0.52\ -0.29\ 0.58\ 0.56]^T. \end{aligned} \tag{6.36}$$

The same procedure are done for E_1 and F_1, which results in

$$\begin{aligned} \mathbf{w_2} &= [-0.28\ -0.93\ 0.023\ -0.28]^T, \\ \mathbf{p_2} &= [-0.37\ -0.91\ -0.045\ -0.16]^T. \end{aligned} \tag{6.37}$$

Since the rank of X is four, the procedure can be repeated until $j = 4$. Since only two factors are retained in the example as shown in Chapter 4, we will stop the calibration here and form the regression matrix $B2_2$ as

$$B2_2 = \begin{bmatrix} -0.21 & -0.051 & 0.26 \\ 0.36 & -0.46 & 0.096 \\ -0.33 & 0.078 & 0.25 \\ -0.26 & -0.038 & 0.30 \end{bmatrix}. \tag{6.38}$$

The matrix $Y_{train2,2}$ is formed using (6.35). With the i^{th} column of $Y_{train2,2}$ denoted by y_i, the three dimensional plot of y_1 vs. y_2 vs. y_3 is illustrated in Figure 6.1. The data are reasonably well-separated. Observe that all the 'x' points have large y_3 values and small y_2 and y_1 values, so all Class 3 data would be correctly assigned. Some of the 'o' and '*' points overlap, which indicates that a small portion of the Class 2 data may be misclassified as Class 1 and *vice versa.*

It was discussed above how to diagnose faults based on the rows of Y. An alternative fault diagnosis approach based on discriminant PLS is to apply discriminant analysis to the PLS scores for classification [108]. In the terminology introduced in Chapter 5, for classifying p classes, the $p-1$ PLS directions can have substantially non-zero between-groups variance. This method can also provide substantially improved fault diagnosis over PCA [108].

6.5 Comparison of PCA and PLS

For fault diagnosis, a predicted block Y is not used in PCA, instead a linear transformation is performed in X such that the highest ranked PCA vectors

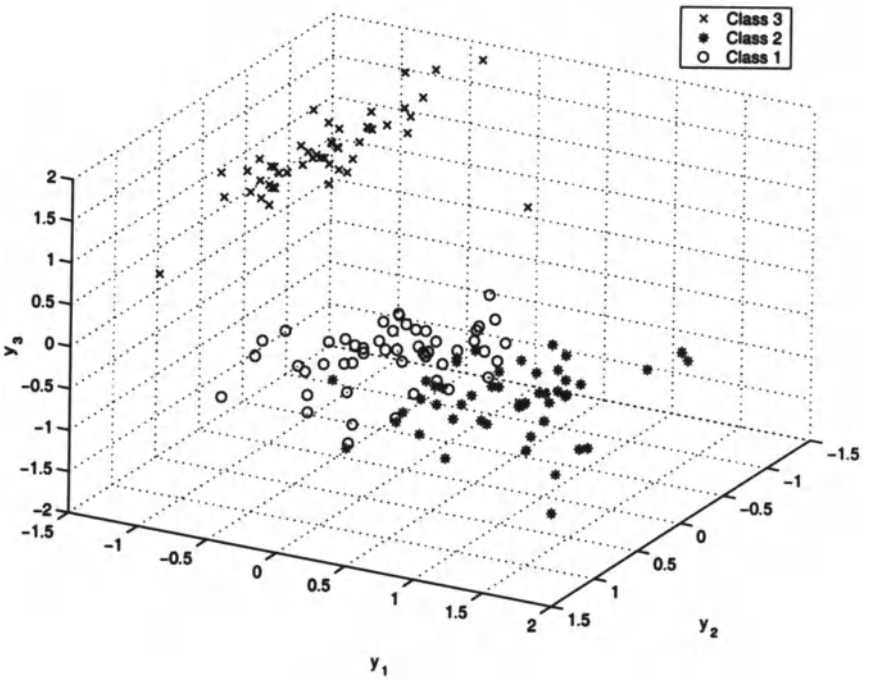

Fig. 6.1. The discriminant PLS predicted matrix plot for the data from [50, 25]

retain most of the variation in X. As described in Chapter 4, the retained scores can be used with discriminant analysis for classification. The disadvantage of the PCA approach is that the highest ranked PCA vectors may not contain the discriminatory power needed to diagnose faults.

PCA maximizes the variance in X while PLS maximizes the covariance between X and Y. By specifying Y to include the fault information as done in discriminant PLS, the PLS vectors are computed so as to provide a lower dimensional representation which is correlated with differences in fault class. Thus fewer of the discriminant PLS vectors should be required and lower misclassification rates obtained. As discriminant PLS exploits fault information when constructing its lower dimensional model, it would be expected that discriminant PLS can provide better fault diagnosis than PCA. However, this is not always true, as will be demonstrated in application in Chapter 10.

The projection of the experimental data taken from [50, 25] onto the first two PCA and discriminant PLS loading vectors is shown in Figure 6.2. Recall that the PCA model is built based on the data from all three classes. The two plots look similar indicating that PCA and discriminant PLS give similar separability of the data when two score vectors are used. For data of

high dimension, our experience is that similarity between the first few PCA and PLS score vectors is often observed [105]. For score vectors of higher orders, the difference between PCA and discriminant PLS usually becomes more apparent. In this example, the loading matrices corresponding to all four loading vectors for PCA and discriminant PLS are

$$P_{PCA} = \begin{bmatrix} 0.5255 & -0.3634 & 0.6686 & -0.3804 \\ -0.2695 & -0.9266 & -0.1869 & 0.1842 \\ 0.5837 & -0.0081 & -0.0013 & 0.8119 \\ 0.5572 & -0.0969 & -0.7197 & -0.4027 \end{bmatrix} \tag{6.39}$$

and

$$P_{PLS} = \begin{bmatrix} 0.5167 & -0.3709 & 0.7510 & -0.2896 \\ -0.2885 & -0.9136 & -0.0275 & 0.2084 \\ 0.5836 & -0.0449 & 0.0024 & 0.8001 \\ 0.5561 & -0.1607 & -0.6597 & -0.4823 \end{bmatrix}, \tag{6.40}$$

respectively.

Note that the first PCA and discriminant PLS loading vectors are very closely aligned and the fourth loading vectors are much less so. Recall that the loading vectors for PCA are orthogonal. In PLS, the loading vectors are rotated slightly in order to capture a better relationship between the predicted and predictor blocks (*i.e.*, maximize the covariance between X and Y) [105]. As a result of this rotation, the PLS loading vectors are rarely orthogonal. In general, the rotation for the first PLS loading vector is usually small. As the order increases, the deviation from orthogonality for the discriminant PLS loading vectors usually increases. Although the discriminant PLS loading vectors are not orthogonal, their score vectors are indeed orthogonal. Readers are urged to verify this property.

6.6 Other PLS Methods

The PLS methods described in this chapter can be extended to take into account the serial correlations in the data, by augmenting the observation vector and stacking the data matrix in the same manner as (4.44). The matrix Y has to be changed correspondingly. Implementation of this approach is left as an exercise for the readers.

The PLS approaches can be generalized to nonlinear systems using nonlinear **Partial Least Squares** (NPLS) algorithms [234, 51, 148]. In NPLS, the relationship between $\hat{\mathbf{u}}_j$ and $\mathbf{t}_j$ in (6.6) is replaced by

$$\hat{\mathbf{u}}_j = f(\mathbf{t}_j) \tag{6.41}$$

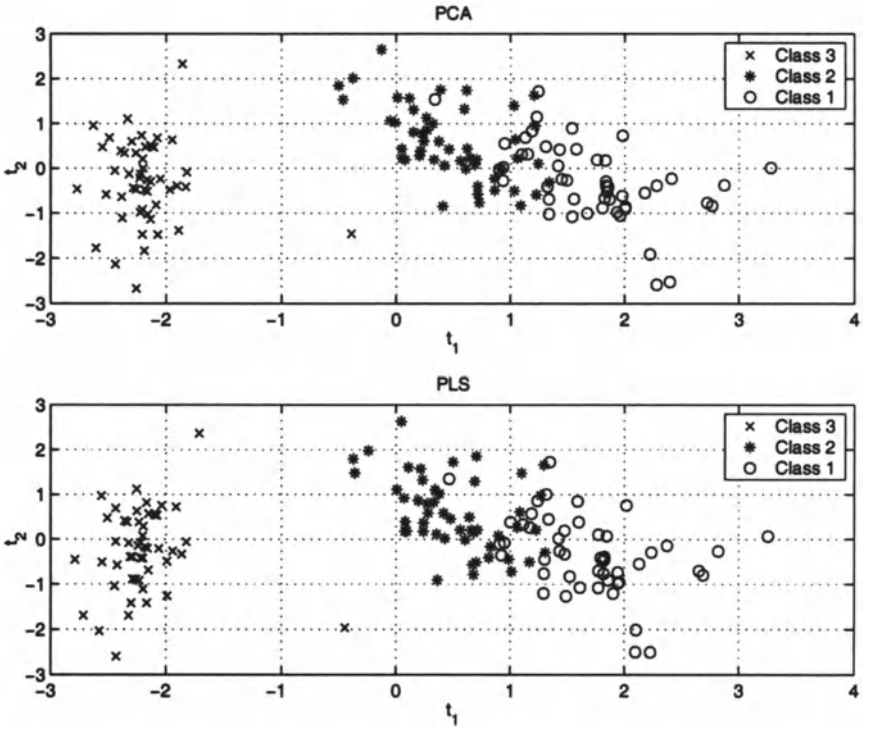

Fig. 6.2. The projections of experimental data [50, 25] for three classes onto the first two discriminant PLS and PCA loading vectors, respectively

where $f(\mathbf{t_j})$ is a nonlinear, continuous, and differentiable function in $\mathbf{t_j}$. The simplest nonlinear relationship for NPLS is a quadratic function

$$f(t_{j,k}) = a_j + b_j t_{j,k} + c_j t_{j,k}^2 \tag{6.42}$$

and $f(\mathbf{t_j}) = [f(t_{j,1})\ f(t_{j,2}) \ldots f(t_{j,n})]^T$. This NPLS model is commonly known as Quadratic Partial Least Squares (QPLS). At each iteration of QPLS, the ordinary PLS steps are applied to $\mathbf{t_j}$, $\mathbf{q_j}$, and $\mathbf{u_j}$, and ordinary least squares are used to estimate the coefficients a_j, b_j, and c_j (see [234] for the detailed procedure). The nonlinearities can also be based on sigmoidal functions as used in artificial neural networks [83, 185].

For systems with mild nonlinearities, the same degree of fit can usually be obtained by a linear model with several factors, or by a nonlinear model with fewer dimensions [234]. In cases where the systems display strong nonlinearities (*i.e.*, if the nonlinearities have maxima, minima, or have significant curvature), a nonlinear model is appropriate and NPLS can perform better than linear PLS especially when the systems are well-determined and with

high observation/variable ratio. However, for an underdetermined system, the models cannot be fitted with acceptable variance using NPLS because of the small number of degrees of freedom in the data sets [51].

Other PLS methods in the literature that have been applied to either simulations or actual process applications are recursive Partial Least Squares (RPLS) [184], multiblock Partial Least Squares [145, 228], and multiway Partial Least Squares [169, 228]. The multiway technique is especially useful for the monitoring of batch processes, in which the predictor X is usually selected to be a three-dimensional array ($i \times j \times k$). A straightforward generalization of the PLS technique to the multiway technique provides a strategy for the detection and diagnosis of faults in batch processes.

6.7 Homework Problems

1. Describe in some detail how to formulate the Q and T^2 statistics for detecting faults using PLS, where Y is the matrix of product quality variables. Compare and contrast this fault detection approach with the PCA-based Q and T^2 statistics. Describe in detail how to generalize the discriminant-based PCA methods for fault diagnosis to PLS, where Y is the matrix of product quality variables. How would you expect the performance of this approach to compare with the performance of discriminant PLS?
2. Generalize PLS as described in Problem 1 to EWMA and CUSUM versions, and to dynamic PLS.
3. Show that the PCA loading vectors for the experimental data from [50, 25] are orthogonal (hint: compute $P_{PCA}^T P_{PCA}$ using P_{PCA} in (6.39)). Show that the PLS loading vectors for the data are not orthogonal. Calculate the angle between the j^{th} PCA and j^{th} PLS loading vector for the data for $j = 1, \ldots, 4$. How does the angle change as a function of j?
4. Generalize discriminant PLS to dynamic discriminant PLS.
5. Provide a detailed comparison of FDA and discriminant PLS. Which method would be expected to do a better job diagnosing faults? Why?
6. Read an article on the use of multiway PLS (e.g., [112, 169]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how the statistics are computed. Formulate a discriminant multiway PLS algorithm. For what types of processes are these algorithms suited? Provide some hypothetical examples.
7. Read an article on the application of multiblock PLS (e.g., [145, 52]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how the statistics are computed. Formulate a discriminant multiblock PLS algorithm. For what types of processes are these algorithms suited? Provide some hypothetical examples.

8. Read an article on the application of nonlinear PLS (e.g., [234, 51, 148]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how the statistics are computed. For what types of processes are these algorithms suited? Provide some hypothetical examples.

CHAPTER 7
CANONICAL VARIATE ANALYSIS

7.1 Introduction

In Section 4.7, it was shown how DPCA can be applied to develop an autoregressive with input ARX model and to monitor the process using the ARX model. The weakness of this approach is the inflexibility of the ARX model for representing linear dynamical systems. For instance, a low order **autoregressive moving average** ARMA (or autoregressive moving average with input ARMAX) model with relatively few estimated parameters can accurately represent a high order ARX model containing a large number of parameters [137]. For a single input single output (SISO) process, an ARMAX(h) model is:

$$y_t = \sum_{i=1}^{h} \alpha_i y_{t-i} + \sum_{i=0}^{h} \beta_i u_{t-i} + \sum_{i=1}^{h} \gamma_i e_{t-i} + e_t \tag{7.1}$$

where y_t and u_t are the output and input at time t, respectively, $\alpha_1, \ldots, \alpha_h$, $\beta_1, \ldots, \beta_h$, and $\gamma_1, \ldots, \gamma_h$ are constant coefficients, and e_t is a white noise process with zero mean [224]. For an invertible process, the ARMAX(h) model can be written as an infinite order ARX model [224]:

$$y_t = \sum_{i=1}^{\infty} \pi_i y_{t-i} + \sum_{i=0}^{\infty} \rho_i u_{t-i} + e_t. \tag{7.2}$$

The constant coefficients $\pi_1, \pi_2, \ldots$ and $\rho_1, \rho_2, \ldots$ are determined from the coefficients in (7.1) via the backshift and division operations [224].

The classical approach to identifying ARMAX processes requires the *a priori* parameterization of the ARMAX model and the subsequent estimation of the parameters via the solution of a least squares problem [137]. To avoid over-parameterization and identifiability problems, the structure of the ARMAX model needs to be properly specified; this is especially important for multivariable systems with a large number of inputs and outputs. This structure specification for ARMAX models is analogous to specifying the observability (or controllability) indices and the state order for state space models, and is not trivial for higher order multivariable systems [215]. Another problem with the classical approach is that the least squares problem

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

requires the solution of a nonlinear optimization problem. The solution of the nonlinear optimization problem is iterative, can suffer from convergence problems, can be overly sensitive to small data fluctuations, and the required amount of computation to solve the optimization problem cannot be bounded [131].

To avoid the problems of the classical approach, a class of system identification methods for generating state space models called **subspace algorithms** has been developed in the past few years. The class of state space models is equivalent to the class of ARMAX models [8, 137]. That is, given a state space model, an ARMAX model with an identical input-output mapping can be determined, and *vice versa*. The subspace algorithms avoid *a priori* parameterization of the state space model by determining the states of the system directly from the data, and the states along with the input-output data allow the state space and covariance matrices to be solved directly via *linear* least squares [215] (see Figure 7.1). These algorithms rely mostly on the singular value decomposition (SVD) for the computations, and therefore do not suffer from the numerical difficulties associated with the classical approach.

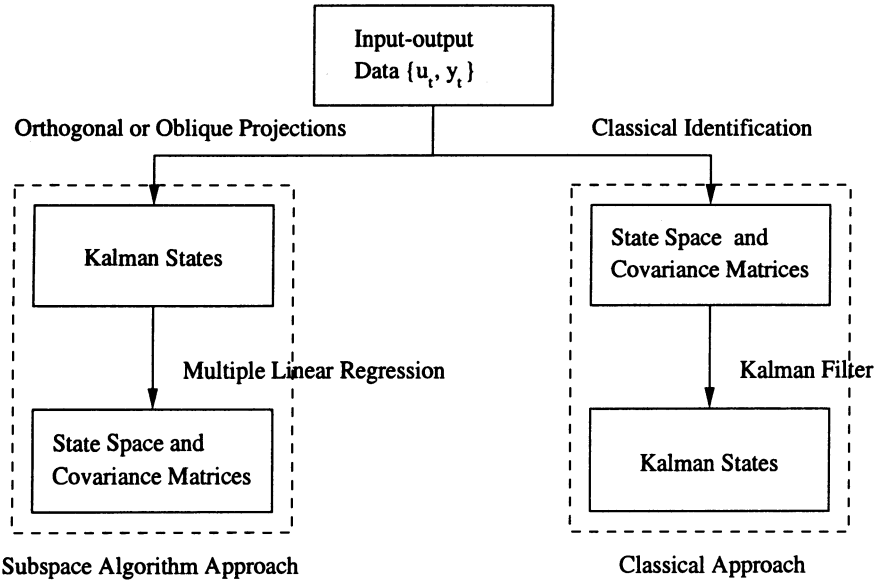

Fig. 7.1. A comparison of the subspace algorithm approach to the classical approach for identifying the state space model and extracting the Kalman states [216]

Three popular subspace algorithms are **numerical algorithms for subspace state space system identification** (N4SID), **multivariable output-error state space** (MOESP), and **Canonical Variate Analysis** (CVA) [216]. Although the subspace algorithm based on CVA is often referred to as "CVA", CVA is actually a dimensionality reduction technique in multivariate statistical analysis involving the selection of pairs of variables from the *inputs* and *outputs* that maximize a correlation measure [131]. For clarity of presentation, "CVA" in this book refers to the dimensionality reduction technique, and the subspace algorithm based on CVA is called the **CVA algorithm**. The philosophy of CVA shares many common features to PCA, FDA, and PLS (see Section 7.2), which makes it a natural subspace identification technique for use in developing process monitoring statistics. The CVA-based statistics described in in this chapter can be readily generalized to the other subspace identification algorithms.

To fully understand all aspects of CVA requires knowledge associated with materials outside of the scope of this book. Enough information is given in this chapter for the readers to gain some intuitive understanding of how CVA works and to implement the process monitoring techniques. Section 7.2 describes the CVA Theorem and an interpretation of the theorem indicating the optimality of CVA for dimensionality reduction. Section 7.3 describes the CVA algorithm with a statistical emphasis. Determination of the state space model and the issues of system identifiability are discussed in Section 7.4. Section 7.5 addresses the computational issues of CVA. A procedure for automatically and optimally selecting the state order of the state space model is presented in Section 7.6. Section 7.7 presents a systems theory interpretation for the CVA algorithm and the other subspace algorithms. Section 7.8 discusses the process monitoring measures developed for the states extracted by the CVA algorithm.

7.2 CVA Theorem

CVA is a linear dimensionality reduction technique, optimal in terms of maximizing a correlation measure between two sets of variables. The **CVA Theorem** states that given a vector of variables $\mathbf{x} \in \mathcal{R}^m$ and another vector of variables $\mathbf{y} \in \mathcal{R}^n$ with covariance matrices Σ_{xx} and Σ_{yy}, respectively, and cross covariance matrix Σ_{xy}, there exist matrices $J \in \mathcal{R}^{m \times m}$ and $L \in \mathcal{R}^{n \times n}$ such that

$$J\Sigma_{xx}J^T = I_{\bar{m}}, \qquad L\Sigma_{yy}L^T = I_{\bar{n}}, \tag{7.3}$$

and

$$J\Sigma_{xy}L^T = D = \text{diag}(\gamma_1, \cdots, \gamma_r, 0, \cdots, 0), \tag{7.4}$$

where $\gamma_1 \geq \cdots \geq \gamma_r$, $\bar{m} = \text{rank}(\Sigma_{xx})$, $\bar{n} = \text{rank}(\Sigma_{yy})$, D contains the **canonical correlations** γ_i, $I_{\bar{m}} \in \mathcal{R}^{m \times m}$ is a diagonal matrix containing

the first $\bar{m}$ diagonal elements as one and the rest of the diagonal elements as zero, and $I_{\bar{n}} \in \mathcal{R}^{n \times n}$ is the diagonal matrix containing the first $\bar{n}$ diagonal elements as one and the rest of the diagonal elements as zero [131]. The vector of **canonical variables** $\mathbf{c} = J\mathbf{x}$ contains a set of uncorrelated random variables and has the covariance matrix

$$\Sigma_{cc} = J\Sigma_{xx}J^T = I_{\bar{m}}, \tag{7.5}$$

and the vector of **canonical variables** $\mathbf{d} = L\mathbf{y}$ contains a set of uncorrelated random variables and has the covariance matrix

$$\Sigma_{dd} = L\Sigma_{yy}L^T = I_{\bar{n}}. \tag{7.6}$$

The cross covariance matrix between $\mathbf{c}$ and $\mathbf{d}$ is diagonal

$$\Sigma_{cd} = J\Sigma_{xy}L^T = D = \text{diag}(\gamma_1, \cdots, \gamma_r, 0, \cdots, 0), \tag{7.7}$$

which indicates that the two vectors are only pairwise correlated. The degree of the pairwise correlations is indicated and can be ordered by the canonical correlations γ_i.

CVA is equivalent to a **generalized singular value decomposition** (GSVD) [126, 131]. When Σ_{xx} and Σ_{yy} are invertible, the projection matrices J and L and the matrix of canonical correlations D can be computed by solving the SVD

$$\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2} = U\Sigma V^T \tag{7.8}$$

where $J = U^T\Sigma_{xx}^{-1/2}$, $L = V^T\Sigma_{yy}^{-1/2}$, and $D = \Sigma$ [127]. It is easy to verify that J, L, and D computed from (7.8) satisfy (7.3) and (7.4). The weightings $\Sigma_{xx}^{-1/2}$ and $\Sigma_{yy}^{-1/2}$ ensure that the canonical variables are uncorrelated and have unit variance, and the matrices U^T and V^T rotate the canonical variables so that $\mathbf{c}$ and $\mathbf{d}$ are only pairwise correlated. The degree of the pairwise correlations are indicated by the diagonal elements of Σ. Note that the GSVD mentioned above is not the same as the GSVD described in most of the mathematics literature [66, 214].

A CVA-related approach in the multivariate statistics literature [161, 34, 135, 204, 122] is known as **Canonical Correlation Analysis** (CCA), which can be generalized into the CVA Theorem [161, 122]. While both CCA and CVA are suitable for correlating two sets of variables, CVA has been applied on time series data (see Section 7.3). To emphasis the application of the process monitoring algorithm on time series data, we prefer to use the terminology CVA over CCA.

Several dimensionality reduction techniques have been interpreted in the framework of the GSVD [131, 135]. For example, consider the case where the left hand side of (7.8) is replaced by $\Sigma_{xx}^{1/2}$. Then

$$\Sigma_{xx}^{1/2} = U\Sigma V^T. \tag{7.9}$$

Using the fact that $U = V$ (Since $\Sigma_{xx}^{1/2}$ is symmetric), squaring both sides give

$$\Sigma_{xx} = U\Sigma^2 V^T. \tag{7.10}$$

The corresponding equation (4.3) for PCA is

$$\Sigma_{xx} = U\Lambda V^T. \tag{7.11}$$

We see that the diagonal elements of Σ in (7.9) is equal to the diagonal elements of Σ in (4.2).

CVA can be reduced to FDA. The generalized eigenvalue problem for FDA (5.9) can be written as a function of x and y as defined in (7.3), where x contains the measurement variables and y contains dummy variables which represent class membership similarly to (6.1) [135].

PLS is also related with CVA, where both methods are equivalent to a GSVD on the covariance matrix. The difference is that CVA uses a weighting so as to maximize *correlation*, whereas PLS maximizes *covariance* [195]. CVA simultaneously obtains all components (J, L, and D) in one GSVD, whereas the PLS algorithm is sequential in selecting the important components, working with the residuals from the previous step.

7.3 CVA Algorithm

In Section 7.2, the optimality and the structure abstraction of CVA were presented via the CVA Theorem. While the CVA concept for multivariate statistical analysis was developed by Hotelling [85], it was not applied to system identification until Akaike's work on the ARMA model [131, 1, 2, 3]. Larimore developed CVA for state space models [131, 127, 126]. This section describes the linear state space model and the CVA algorithm for identifying state space models directly from the data.

Given time series input data $\mathbf{u_t} \in \mathcal{R}^{m_u}$ and output data $\mathbf{y_t} \in \mathcal{R}^{m_y}$, the linear state space model is given by [129]

$$\mathbf{x_{t+1}} = \Phi\mathbf{x_t} + G\mathbf{u_t} + \mathbf{w_t} \tag{7.12}$$

$$\mathbf{y_t} = H\mathbf{x_t} + A\mathbf{u_t} + B\mathbf{w_t} + \mathbf{v_t} \tag{7.13}$$

where $\mathbf{x_t} \in \mathcal{R}^k$ is a k-order state vector and $\mathbf{w_t}$ and $\mathbf{v_t}$ are white noise processes that are independent with covariance matrices Q and R, respectively. The state space matrices Φ, G, H, A, and B along with the covariance matrices Q and R specify the state space model. It is assumed here that the state space matrices are constant (*time-invariance*) and the covariance matrices are constant (*weakly stationary*). The term $B\mathbf{w_t}$ in (7.13) allows the

noise in the *output equation* (7.13) to be correlated with the noise in the *state equation* (7.12). Omitting the term Bw_t, typically done for many state space models, may result in a state order that is not minimal [127]. Time-varying trends in the data can be fitted by augmenting polynomial functions of time to the state space model; a software package that implements this is *ADAPTx Version 3.03* [129].

An important aspect of the CVA algorithm is the separation of *past* and *future*. At a particular time instant $t \in (1, \cdots, n)$ the vector containing the information from the past is

$$\mathbf{p_t} = \left[\mathbf{y}_{t-1}^T, \mathbf{y}_{t-2}^T, \cdots, \mathbf{u}_{t-1}^T, \mathbf{u}_{t-2}^T, \cdots\right]^T, \tag{7.14}$$

and the vector containing the output information in the present and future is

$$\mathbf{f_t} = \left[\mathbf{y}_t^T, \mathbf{y}_{t+1}^T, \cdots\right]^T. \tag{7.15}$$

Assuming the data is generated from a linear state space model with a finite number of states k, the elements of the state vector $\mathbf{x_t}$ is equal to a set of k linear combinations of the past,

$$\mathbf{x_t} = J_k \mathbf{p_t} \tag{7.16}$$

where $J_k \in \mathcal{R}^{k \times m_p}$ is a constant matrix with $m_p < \infty$. The state vector $\mathbf{x_t}$ has the property that the conditional probability of the future $\mathbf{f_t}$ conditioned on the past $\mathbf{p_t}$ is equal to the conditional probability of the future $\mathbf{f_t}$ conditioned on the state $\mathbf{x_t}$

$$P(\mathbf{f_t}|\mathbf{p_t}) = P(\mathbf{f_t}|\mathbf{x_t}). \tag{7.17}$$

In other words, the state provides as much information as past data does as to the future values of the output. This also indicates that only a finite number of linear combinations of the past affect the future outputs. This property of the state vector can be extended to include future inputs [129]

$$P((\mathbf{f_t}|\mathbf{q_t})|\mathbf{p_t}) = P((\mathbf{f_t}|\mathbf{q_t})|\mathbf{x_t}) \tag{7.18}$$

where $\mathbf{q_t} = [\mathbf{u}_t^T, \mathbf{u}_{t+1}^T, \cdots]^T$. In the process identification literature, a process satisfying (7.18) is said to be a **controlled Markov process** of order k.

Let the k-order memory, $\mathbf{m_t} \in \mathcal{R}^k$, be a set of k linear combinations of the past $\mathbf{p_t}$

$$\mathbf{m_t} = C_k \mathbf{p_t} \tag{7.19}$$

where $C_k \in \mathcal{R}^{k \times m_p}$. The term "memory" is used here instead of "state" because the vector $\mathbf{m_t}$ may not necessarily contain all the information in the past (for instance, the dimensionality of k may not be sufficient to capture all

the information in the past). The goal of process identification is to provide the optimal prediction of the future outputs based on the past and current state. Now in a real process the true state order k is unknown, so instead the future outputs are predicted based on the current memory:

$$\hat{f}_t(\mathbf{m_t}) = \Sigma_{fm}\Sigma_{mm}^{-1}\mathbf{m_t} \tag{7.20}$$

where $\hat{\mathbf{f}}_\mathbf{t}(\mathbf{m_t})$ is the optimal linear prediction of the future $\mathbf{f_t}$ based on the memory $\mathbf{m_t}$ [129]. The CVA algorithm computes the optimal matrix for C_k in (7.19), that is, the matrix C_k which minimizes the average prediction error:

$$E\{(\mathbf{f_t} - \hat{\mathbf{f}}_\mathbf{t})\Lambda^\dagger(\mathbf{f_t} - \hat{\mathbf{f}}_\mathbf{t})\} \tag{7.21}$$

where E is the expectation operator and $\Lambda^\dagger$ is the pseudo inverse of Λ, which is a positive semidefinite symmetric matrix used to weigh the relative importance of the output variables over time. The choice $\Lambda = \Sigma_{ff}$ results in nearly maximum likelihood estimates [126, 195].

The optimal value for C_k in (7.19) is computed via the GSVD by substituting the matrix Σ_{xx} with Σ_{pp}, Σ_{yy} with Σ_{ff}, and Σ_{xy} with Σ_{pf} in (7.3) and (7.4) [129]. The optimal estimate for matrix C_k is equal to J_k, where J_k is the first k rows of the matrix J in (7.3) [131]. The optimal k-order memory is

$$\mathbf{m}_\mathbf{t}^{opt} = J_k\mathbf{p_t}. \tag{7.22}$$

The structure of the solution indicates that the optimal memory for order k is a subset of the *optimal* memory for order $k+1$. The optimal memory for a given order k corresponds to the first k states of the system [129], and these states are referred to as the **CVA states**.

7.4 State Space Model and System Identifiability

The process monitoring statistics described in Section 7.8 are based on the matrix J which is used to construct the CVA states, and do not require the construction of an explicit state space model (7.12)-(7.13). The calculation of the state space matrices in (7.12)-(7.13) is described here for completeness.

Assuming the order of the state space model, k, is chosen to be greater than or equal to the order of the minimal state space realization of the actual system, the state vectors $\mathbf{x_t}$ in (7.12) and (7.13) can be replaced by the state estimate $\mathbf{m_t}$:

$$\begin{bmatrix} \mathbf{m_{t+1}} \\ \mathbf{y_t} \end{bmatrix} = \begin{bmatrix} \Phi & G \\ H & A \end{bmatrix} \begin{bmatrix} \mathbf{m_t} \\ \mathbf{u_t} \end{bmatrix} + \begin{bmatrix} I & 0 \\ B & I \end{bmatrix} \begin{bmatrix} \mathbf{w_t} \\ \mathbf{v_t} \end{bmatrix} \tag{7.23}$$

Since $\mathbf{u_t}$ and $\mathbf{y_t}$ are known, and $\mathbf{m_t}$ can be computed once J_k in (7.22) is known, this equation's only unknowns (Φ, G, H, A, and B) are linear in the

parameters. The state space matrices can be estimated by multiple linear regression (see Figure 7.1)

$$\begin{bmatrix} \hat{\Phi} & \hat{G} \\ \hat{H} & \hat{A} \end{bmatrix} = \hat{\Sigma}_{my,mu} \hat{\Sigma}_{mu,mu}^{-1} \tag{7.24}$$

where

$$\mathbf{mu} = \begin{bmatrix} \mathbf{m_t} \\ \mathbf{u_t} \end{bmatrix}, \ \mathbf{my} = \begin{bmatrix} \mathbf{m_{t+1}} \\ \mathbf{y_t} \end{bmatrix}, \tag{7.25}$$

and $\hat{\Sigma}_{i,j}$ represents the sample covariance matrix for variables i and j. The error of the multiple regression has the covariance matrix

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \hat{\Sigma}_{my,my} - \hat{\Sigma}_{my,mu} \hat{\Sigma}_{mu,mu}^{-1} \hat{\Sigma}_{my,mu}^{T}, \tag{7.26}$$

and the matrices $\hat{B} = S_{21} S_{11}^{\dagger}$, $\hat{Q} = S_{11}$, and $\hat{R} = S_{22} - S_{21} S_{11}^{\dagger} S_{12}$ where $\dagger$ signifies the pseudo-inverse [66]. With the matrices $\hat{A}$, $\hat{B}$, $\hat{H}$, $\hat{G}$, $\hat{\Phi}$, $\hat{Q}$, and $\hat{R}$ estimated, the state space model as shown in (7.12) and (7.13) can be used for various applications such as multistep predictions and forecasts, for example, as needed in model predictive control [195, 107].

There is a significant advantage in terms of identifiability of state space identification approaches over classical identification based on polynomial transfer functions. For polynomial transfer functions, it is always possible to find particular values of the parameters that produce arbitrarily poor conditioning [129, 65], and hence a loss in identifiability of the model [222, 183]. The simplest example of this is when a process pole nearly cancels a process zero.

The state space model estimated using (7.24) and (7.26) is globally identifiable, so that the method is statistically well-conditioned [131]. The CVA algorithm guarantees the choice of a well conditioned parameterization.

7.5 Lag Order Selection and Computation

The discussion in Section 7.3 assumes that an infinite amount of data is available. For the computational problem, there is a finite amount of data available, and the vectors $\mathbf{p_t}$, $\mathbf{f_t}$, and $\mathbf{q_t}$ are truncated as

$$\mathbf{p_t} = \left[\mathbf{y_{t-1}^T}, \mathbf{y_{t-2}^T}, \cdots, \mathbf{y_{t-h}^T}, \mathbf{u_{t-1}^T}, \mathbf{u_{t-2}^T}, \cdots, \mathbf{u_{t-h}^T}\right]^T, \tag{7.27}$$

$$\mathbf{f_t} = \left[\mathbf{y_t^T}, \mathbf{y_{t+1}^T} \cdots, \mathbf{y_{t+l-1}^T}\right]^T, \tag{7.28}$$

$$\mathbf{q_t} = \left[\mathbf{u}_t^T, \mathbf{u}_{t+1}^T, \cdots, \mathbf{u}_{t+l-1}^T\right]^T \tag{7.29}$$

where h and l are the number of lags included in the vectors. Note that $\mathbf{p_t}$ with h lags directly corresponds to the observation vector for (4.44) with $h-1$ lags. Theoretically, the CVA algorithm does not suffer when $h = l > k$, where k is the state order of the system generating the data (actually, h and l just need to be larger than the largest observability index [216]). However, the state order of the system is not known *a priori*. The first step of computing of CVA is to determine the number of lags h. Assuming there are n observations in the training set and the maximum number for the lag order is max, Larimore suggests fitting autoregressive models with several different numbers of lags to the *training data*:

$$Y = C_j X_j + E_j \tag{7.30}$$

where the predicted matrix $Y \in \mathcal{R}^{(m_u+m_y)\times(n-max)}$ is given as:

$$Y = \begin{bmatrix} \mathbf{y_{max+1}} & \mathbf{y_{max+2}} & \cdots & \mathbf{y_n} \\ \mathbf{u_{max+1}} & \mathbf{u_{max+2}} & \cdots & \mathbf{u_n} \end{bmatrix} \tag{7.31}$$

and the predictor matrix $X_j \in \mathcal{R}^{j(m_u+m_y)\times(n-max)}$ with j lags is given as the first $j(m_u + m_y)$ rows of

$$X = \begin{bmatrix} \mathbf{y_{max}} & \mathbf{y_{max+1}} & \cdots & \mathbf{y_{n-1}} \\ \mathbf{u_{max}} & \mathbf{u_{max+1}} & \cdots & \mathbf{u_{n-1}} \\ \mathbf{y_{max-1}} & \mathbf{y_{max}} & \cdots & \mathbf{y_{n-2}} \\ \mathbf{u_{max-1}} & \mathbf{u_{max}} & \cdots & \mathbf{u_{n-2}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{y_1} & \mathbf{y_2} & \cdots & \mathbf{y_{n-max}} \\ \mathbf{u_1} & \mathbf{u_2} & \cdots & \mathbf{u_{n-max}} \end{bmatrix} \tag{7.32}$$

and $E_h \in \mathcal{R}^{(m_u+m_y)\times(n-max)}$ is the residual matrix for lag order j. The regression matrix for C_j is determined via least squares:

$$C_j = \Sigma_{YX_j}\Sigma_{X_jX_j}^{-1} \tag{7.33}$$

where the covariance matrix Σ_{YX_j} is equal to $\frac{1}{n-max}YX_j^T$. The residual matrix E_j is calculated for $j = 1, 2, \ldots, max$. The lag order h is selected to be the lag minimizing the small sample AIC criterion (7.37) discussed in Section 7.6. This ensures that large enough lags are used to capture all the statistically significant information in the data. The selection of the state order k is described in next section.

The computational requirements are known *a priori* for the GSVD computation. The number of flop counts grows by order $(nh+h^3)$, and the required storage space is on the order $(n + h^2)$ [131].

The near optimality of the state space model produced by the CVA algorithm has been observed in Monte Carlo simulations. The estimated **Kullback-Leibler information distances** (see Section 7.6) for both open and closed loop simulations were close to the information distances, related to the Cramer-Rao bound, corresponding to the minimum possible parameter estimation error for any unbiased estimation procedure [131]. Simulations have also verified the robustness of the CVA algorithm for systems involving feedback [131].

7.6 State Order Selection and Akaike's Information Criterion

The selection of the state order is an important step in identifying a state space model. The existence of a *true* state order is highly suspect when dealing with real process data; however, the state order can be utilized as a tradeoff parameter for the model complexity, similar to the order of model reduction, a, described for PCA, FDA, and PLS in Chapters 4, 5, and 6, respectively. For instance, choosing the state order too large results in the model overfitting the data, and choosing the state order too small results in the model underfitting the data. This section presents a method for state order selection based on **Akaike's information criterion** (AIC).

The agreement between two probability density functions can be measured in terms of the **Kullback-Leibler information distance** (KLIB) [137]

$$I(p_*(x), \hat{p}(x)) = \int p_*(x) \ln \frac{p_*(x)}{\hat{p}(x)} dx \tag{7.34}$$

where x contains the random variables, $p_*(x)$ is the true probability density function, and $\hat{p}(x)$ is the estimated probability density function. The KLIB is based on the statistical principles of sufficiency and repeated sampling in a predictive inference setting, and is invariant to model reparameterization [130]. If the true probability density function of the process data is known, then the information distance (7.34) could be computed for various state orders and the optimal state order would correspond to the minimum information distance.

For large samples, the optimal estimator of the information distance (7.34) for a given order k is the AIC,

$$AIC(k) = -2 \ln p(y^n, u^n; \hat{\theta}_k) + 2M_k \tag{7.35}$$

where p is the likelihood function [9], the vectors u^n and y^n contain n observations for the input and output variables, respectively, and $\hat{\theta}_k$ are the M_k independent parameters estimated for state order k. The order k is selected

such that the AIC criterion (7.35) is minimized. The number of independent parameters in the state space model (7.12) and (7.13) is

$$M_k = k(2m_y + m_u) + m_u m_y + \frac{m_y(m_y+1)}{2}. \tag{7.36}$$

The number of independent parameters is far less than the actual number of parameters in the state space model [137], and the result (7.36) was developed by considering the size of the equivalence class of state space models having the same input-output and noise characteristics [129].

For small samples, the AIC can be an inaccurate estimate of the KLIB. This has led to the development of the small sample correction to the AIC [129]

$$AIC_C(k) = -2\ln p(y^n, u^n; \hat{\theta}_k) + 2fM_k \tag{7.37}$$

where the correction factor for small samples is

$$f = \frac{\bar{n}}{\bar{n} - \left(\frac{M_k}{m_u + m_y} + \frac{m_u + m_y + 1}{2}\right)} \tag{7.38}$$

where $\bar{n}$ is the number of one-step ahead predictions used to develop the model. The small sample correction to the AIC approaches the AIC ($f \to 1$) as the sample size increases ($\bar{n} \to \infty$). It has been reported to produce state order selections that are close to the optimal prescribed by the KLIB [131]. Within the context of Section 3.3, the selection of the optimal state order results in an optimal tradeoff between the bias and variance effects on the model error.

7.7 Subspace Algorithm Interpretations

The book *Subspace Identification of Linear Systems* by Van Overschee and De Moor [216] presents a unified approach to the subspace algorithms. It shows that the three subspace algorithms (N4SID, MOESP, and CVA) can be computed with essentially the same algorithm, differing only in the choice of weights. Larimore [131] states that the other algorithms differ from the CVA algorithm only in the choice of the matrices Σ_{xx} and Σ_{yy} used in (7.3), and claims accordingly that the other algorithms are statistically suboptimal.

It has been proven under certain assumptions that the subspace algorithms can be used to produce asymptotically unbiased estimates of the state space matrices [216]. However, the state space matrices estimated by the three algorithms can be significantly different when the amount of input and output data is relatively small.

Van Overschee and De Moor also show that the state sequences generated by the subspace algorithms are the outputs of non-steady state Kalman filter

banks. The basis for the states is determined by the weights used by the various algorithms, and the state space realizations produced by the algorithms are balanced realizations under certain frequency-weightings. Therefore, reducing the dimensionality of the memory in the subspace algorithms can be interpreted in the framework of the frequency-weighted balanced truncation techniques developed by Enns [46], with the exception that the subspace algorithms truncate the state space model before the model is estimated (see Figure 7.2). The amount of model error introduced by reducing the order is minimized by eliminating only those states with the smallest effect on the input-output mapping, and for the CVA algorithm, the amount of model error is proportional to the canonical correlations [129]. The model reduction approach of the CVA algorithm has the advantage in that truncating the memory vector prior to the estimation of the state space model instead of truncating the state vector based on a full order state space model is much more computationally and numerically robust (see Figures 7.1 and 7.2). The degree of model reduction, or equivalently the selection of the state order, is an important step in the identification process, and a statistically optimal method was discussed in Section 7.6.

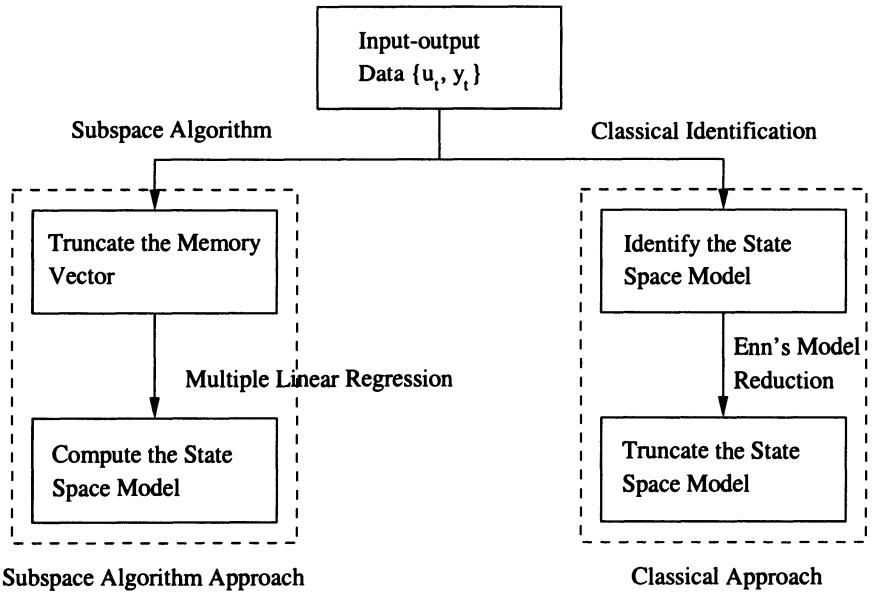

Fig. 7.2. A comparison of the approaches to model reduction using Enn's model reduction technique and the subspace algorithm [216]

7.8 Process Monitoring Statistics

The GSVD for the CVA algorithm produces a set of canonical variables, $\mathbf{c} = J\mathbf{p_t}$ (where $\mathbf{c} \in \mathcal{R}^{h(m_u+m_y)}$), that are uncorrelated and have unit variance. The T^2 statistic for the canonical variables is

$$T^2 = \mathbf{p_t}^T J^T J \mathbf{p_t}. \tag{7.39}$$

The T^2 statistic (7.39), however, may contain a large amount of noise and may not be very robust for monitoring the process. Reducing the order a for DPCA can increase the effectiveness of the T^2 statistic, and allows the process noise to be monitored separately via the Q statistic. An analogous approach is taken here for monitoring the process using the CVA states:

$$\mathbf{x_t} = J_k \mathbf{p_t} = U_k^T \hat{\Sigma}_{pp}^{-1/2} \mathbf{p_t} \tag{7.40}$$

where U_k contains the first k columns of U in (7.8)

A process monitoring statistic based on quantifying the variations of the CVA states has been applied by Negiz and Cinar to a milk pasteurization process [165, 166]. The measure is the T_s^2 statistic

$$T_s^2 = \mathbf{p_t}^T J_k^T J_k \mathbf{p_t}, \tag{7.41}$$

and assuming normality, the T_s^2 statistic follows the distribution

$$T_{s,\alpha}^2 = \frac{k(n^2-1)}{n(n-k)} F_\alpha(k, n-k) \tag{7.42}$$

where n is the number of observations (see 2.11). The T_s^2 statistic measures the variations *inside* the state space, and the process faults can be detected, as shown in Section 2.4, by choosing a level of significance and solving the appropriate threshold using $T_{s,\alpha}^2$.

The variations *outside* the state space can be measured using the statistic

$$T_r^2 = \mathbf{p_t}^T J_q^T J_q \mathbf{p_t} \tag{7.43}$$

where J_q contains the last $q = h(m_u + m_y) - k$ rows of J in (7.8). Assuming normality, the T^2 statistic (7.43) follows the distribution

$$T_{r,\alpha}^2 = \frac{q(n^2-1)}{n(n-q)} F_\alpha(q, n-q). \tag{7.44}$$

A weakness of this approach is that T_r^2 can be overly sensitive because of the inversion of the small values of Σ_{xx} in (7.8) [101]. This can result in a high false alarm rate. To address this concern, the threshold should be readjusted before applying the statistics for process monitoring (see Section 10.6 for an example).

The residual vector of the state space model in terms of the past $\mathbf{p_t}$ can be calculated

$$\mathbf{r_t} = (I - J_k^T J_k)\mathbf{p_t}, \tag{7.45}$$

and the variation in the residual space can be monitored using the Q statistic similar to the (D)PCA approaches

$$Q = \mathbf{r_t}^T \mathbf{r_t}. \tag{7.46}$$

The statistics of T_r^2 and Q essentially measure the noise of the process. The T^2 statistic (7.39) is equal to $T_s^2 + T_r^2$, and by extracting the CVA states from the data, the variations in the state and measurement noise space can be decoupled and measured separately using T_s^2 and T_r^2, respectively. A violation of the T_s^2 statistic indicates that the states are out-of-control, and a violation of the T_r^2 statistic indicates that the characteristic of the measurement noise has changed and/or new states have been created in the process. This is similar to the PCA approach to fault detection outlined in Section 4.4, with the exception that the states of the system are extracted in a different manner. The flexibility of the state space model and the near optimality of the CVA approach suggest that the CVA states more accurately represent the status of the operations compared to the scores using PCA or DPCA. Other CVA-based fault detection statistics are reported in the literature [132, 223].

The correlation structure of the CVA states allows the PCA-based statistics in Chapter 4 for fault identification and diagnosis to be applicable to the CVA model. It is straightforward to extend the PCA-based statistics to CVA. The total contribution statistic (4.25) can be computed for the CVA model by replacing the scores with the CVA estimated states, $\mathbf{m_t} = J_k \mathbf{p_t}$. The statistic (4.32) can be applied for fault identification using the residual vector in (7.45). A pattern classification system for fault diagnosis can be employed using the discriminant function (3.6) based on $(T_s^2)_i$, $(T_r^2)_i$, or Q_i for each class i. These discriminant functions can improve the classification system upon using the discriminant function (3.6) based on the entire observation space, $\mathbf{p_t}$, when most of the discriminatory power is contained in the state space or the residual space.

7.9 Homework Problems

1. Verify that the matrices J, L, and D computed from (7.8) satisfy (7.3) and (7.4).
2. Describe in some detail how to formulate the *CONT* and *RES* statistics for identifying faults using CVA. Name advantages and disadvantages of this approach to alternative methods for identifying faults. Would *CONT* or *RES* expected to perform better? Why?

3. Describe in detail how to formulate CVA for fault diagnosis. Name advantages and disadvantages of this approach to alternative methods for diagnosing faults.
4. Compare and contrast the CVA-based Q and T_r^2 statistics. Which statistic would you expect to perform better for fault detection? Why?
5. Read the following materials [195, 131, 135] and formulate PCA, PLS, FDA, and CVA in the framework of the generalized singular value decomposition. Based on the differences between the methods as represented in this framework, state the strengths and weaknesses of each method for applying process monitoring statistics.
6. Read a chapter in a book on the application of Canonical Correlation Analysis (CCA) [135, 161, 34]. Compare and contrast CCA with FDA and CVA.
7. Compare and contrast the CVA-based statistics described in this chapter with the CVA-based process monitoring statistics reported in these papers [132, 223].
8. Read an article on the application of nonlinear CVA (e.g., [128]) and write a report describing in detail how the technique is implemented and applied. Describe how the computations are performed and how process monitoring statistics can be computed. For what types of processes are these algorithms suited? Provide some hypothetical examples.

Part IV

APPLICATION

CHAPTER 8
TENNESSEE EASTMAN PROCESS

8.1 Introduction

In Part IV the various data-driven process monitoring statistics are compared through application to a simulation of a chemical plant. The methods would ideally be illustrated on data collected during specific known faults from an actual chemical process, but this type of data is not publicly available for any large scale chemical plant. Instead, many academics in process monitoring perform studies based on data collected from computer simulations of a chemical process. The process monitoring methods in this book are tested on the data collected from the process simulation for the **Tennessee Eastman process** (TEP). The TEP has been widely used by the process monitoring community as a source of data for comparing various approaches [10, 24, 62, 63, 74, 77, 125, 133, 187, 189, 188].

The TEP was created by the Eastman Chemical Company to provide a realistic industrial process for evaluating process control and monitoring methods [39]. The test process is based on a simulation of an actual chemical process where the components, kinetics, and operating conditions have been modified for proprietary reasons. The process consists of five major units: a reactor, condenser, compressor, separator, and stripper; and, it contains eight components: A, B, C, D, E, F, G, and H.

Chapter 8 describes the Tennessee Eastman process (TEP) in enough detail to interpret the application of the process monitoring statistics in Chapters 9 and 10. Sections 8.2 to 8.6 describe the process flowsheet, variables, faults, and simulation program. In reality, processes are operated under closed loop control. To simulate realistic conditions, the second plant-wide control structure described in [141] was implemented to generate the data for demonstrating and comparing the various process monitoring methods. The control structure is described in Section 8.6. Detailed discussions on control structures for the TEP are available [152, 151, 163, 220].

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

8.2 Process Flowsheet

Figure 8.1 is a flowsheet for the chemical plant. The gaseous reactants A, C, D, and E and the inert B are fed to the reactor where the liquid products G and H are formed. The reactions in the reactor are:

$$\begin{aligned} \mathrm{A(g)} + \mathrm{C(g)} + \mathrm{D(g)} &\rightarrow \mathrm{G(liq)}, \\ \mathrm{A(g)} + \mathrm{C(g)} + \mathrm{E(g)} &\rightarrow \mathrm{H(liq)}, \\ \mathrm{A(g)} + \mathrm{E(g)} &\rightarrow \mathrm{F(liq)}, \\ 3\mathrm{D(g)} &\rightarrow 2\mathrm{F(liq)}. \end{aligned} \tag{8.1}$$

The species F is a byproduct of the reactions. The reactions are irreversible, exothermic, and approximately first-order with respect to the reactant concentrations. The reaction rates are Arrhenius functions of temperature where the reaction for G has a higher activation energy than the reaction for H, resulting in a higher sensitivity to temperature.

The reactor product stream is cooled through a condenser and then fed to a vapor-liquid separator. The vapor exiting the separator is recycled to the reactor feed through a compressor. A portion of the recycle stream is purged to keep the inert and byproduct from accumulating in the process. The condensed components from the separator (Stream 10) is pumped to a stripper. Stream 4 is used to strip the remaining reactants from Stream 10, which are combined with the recycle stream via Stream 5. The products G and H exiting the base of the stripper are sent to a downstream process which is not included in the diagram.

8.3 Process Variables

The process contains 41 measured and 12 manipulated variables. The manipulated variables are listed in Table 8.1. The 22 measured variables which are sampled every 3 minutes, XMEAS(1) through XMEAS(22), are listed in Table 8.2. The 19 composition measurements, XMEAS(23) through XMEAS(41), are described in Table 8.3. The composition measurements are taken from Streams 6, 9, and 11. The sampling interval and time delay for Streams 6 and 9 are both equal to 6 minutes, and for Stream 11 are equal to 15 minutes. All the process measurements include Gaussian noise.

8.4 Process Faults

The Tennessee Eastman Process simulation contains 21 preprogrammed faults (see Table 8.4). Sixteen of these faults are known, and five are unknown. Faults 1-7 are associated with a step change in a process variable,

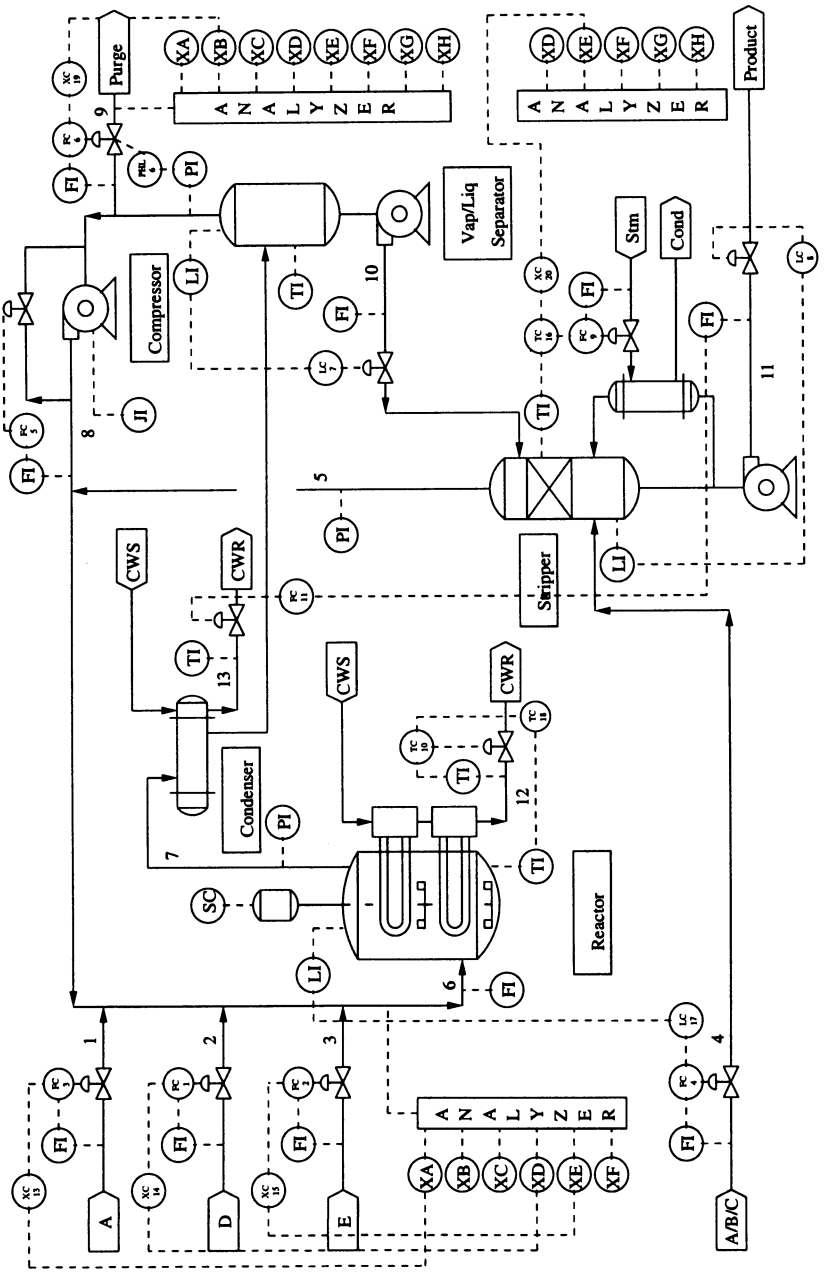

Fig. 8.1. A process flowsheet for the TEP with the second control structure in [141]

Table 8.1. Manipulated variables

Variable	Description
XMV(1)	D Feed Flow (Stream 2)
XMV(2)	E Feed Flow (Stream 3)
XMV(3)	A Feed Flow (Stream 1)
XMV(4)	Total Feed Flow (Stream 4)
XMV(5)	Compressor Recycle Valve
XMV(6)	Purge Valve (Stream 9)
XMV(7)	Separator Pot Liquid Flow (Stream 10)
XMV(8)	Stripper Liquid Product Flow (Stream 11)
XMV(9)	Stripper Steam Valve
XMV(10)	Reactor Cooling Water Flow
XMV(11)	Condenser Cooling Water Flow
XMV(12)	Agitator Speed

Table 8.2. Process measurements (3 minute sampling interval)

Variable	Description	Units
XMEAS(1)	A Feed (Stream 1)	kscmh
XMEAS(2)	D Feed (Stream 2)	kg/hr
XMEAS(3)	E Feed (Stream 3)	kg/hr
XMEAS(4)	Total Feed (Stream 4)	kscmh
XMEAS(5)	Recycle Flow (Stream 8)	kscmh
XMEAS(6)	Reactor Feed Rate (Stream 6)	kscmh
XMEAS(7)	Reactor Pressure	kPa gauge
XMEAS(8)	Reactor Level	%
XMEAS(9)	Reactor Temperature	Deg C
XMEAS(10)	Purge Rate (Stream 9)	kscmh
XMEAS(11)	Product Sep Temp	Deg C
XMEAS(12)	Product Sep Level	%
XMEAS(13)	Prod Sep Pressure	kPa gauge
XMEAS(14)	Prod Sep Underflow (Stream 10)	m^3/hr
XMEAS(15)	Stripper Level	%
XMEAS(16)	Stripper Pressure	kPa gauge
XMEAS(17)	Stripper Underflow (Stream 11)	m^3/hr
XMEAS(18)	Stripper Temperature	Deg C
XMEAS(19)	Stripper Steam Flow	kg/hr
XMEAS(20)	Compressor Work	kW
XMEAS(21)	Reactor Cooling Water Outlet Temp	Deg C
XMEAS(22)	Separator Cooling Water Outlet Temp	Deg C

Table 8.3. Composition measurements

Variable	Description	Stream	Sampling Interval (min.)
XMEAS(23)	Component A	6	6
XMEAS(24)	Component B	6	6
XMEAS(25)	Component C	6	6
XMEAS(26)	Component D	6	6
XMEAS(27)	Component E	6	6
XMEAS(28)	Component F	6	6
XMEAS(29)	Component A	9	6
XMEAS(30)	Component B	9	6
XMEAS(31)	Component C	9	6
XMEAS(32)	Component D	9	6
XMEAS(33)	Component E	9	6
XMEAS(34)	Component F	9	6
XMEAS(35)	Component G	9	6
XMEAS(36)	Component H	9	6
XMEAS(37)	Component D	11	15
XMEAS(38)	Component E	11	15
XMEAS(39)	Component F	11	15
XMEAS(40)	Component G	11	15
XMEAS(41)	Component H	11	15

Units are mole %. Dead time is equal to the sampling interval

e.g., in the cooling water inlet temperature or in feed composition. Faults 8-12 are associated with an increase in the variability of some process variables. Fault 13 is a slow drift in the reaction kinetics, and Faults 14, 15, and 21 are associated with sticking valves.

The sensitivity and robustness of the various process monitoring methods will be investigated in Chapter 10 by simulating the process under various fault conditions. The simulation program allows the faults to be implemented either individually or in combination with one another.

8.5 Simulation Program

The simulation code for the process is available in FORTRAN, and a detailed description of the process and simulation is available [39]. There are six modes to the process operation corresponding to various G/H mass ratios and production rates of Stream 11. Only the base case will be used here. The program is implemented with 50 states in open loop and a 1 second interval for integration. This integration interval is reasonable since the largest negative eigenvalue of the process is about 1.8 seconds. The simulation code for the process in open loop can be downloaded from http://brahms.scs.uiuc.edu.

Table 8.4. Process faults

Variable	Description	Type
IDV(1)	A/C Feed Ratio, B Composition Constant (Stream 4)	Step
IDV(2)	B Composition, A/C Ratio Constant (Stream 4)	Step
IDV(3)	D Feed Temperature (Stream 2)	Step
IDV(4)	Reactor Cooling Water Inlet Temperature	Step
IDV(5)	Condenser Cooling Water Inlet Temperature	Step
IDV(6)	A Feed Loss (Stream 1)	Step
IDV(7)	C Header Pressure Loss - Reduced Availability (Stream 4)	Step
IDV(8)	A, B, C Feed Composition (Stream 4)	Random Variation
IDV(9)	D Feed Temperature (Stream 2)	Random Variation
IDV(10)	C Feed Temperature (Stream 4)	Random Variation
IDV(11)	Reactor Cooling Water Inlet Temperature	Random Variation
IDV(12)	Condenser Cooling Water Inlet Temperature	Random Variation
IDV(13)	Reaction Kinetics	Slow Drift
IDV(14)	Reactor Cooling Water Valve	Sticking
IDV(15)	Condenser Cooling Water Valve	Sticking
IDV(16)	Unknown	
IDV(17)	Unknown	
IDV(18)	Unknown	
IDV(19)	Unknown	
IDV(20)	Unknown	
IDV(21)	The valve for Stream 4 was fixed at the steady state position	Constant Position

8.6 Control Structure

The simulation of the TEP is made available by the Eastman Chemical Company in open loop operation. Since the process is open loop unstable and chemical processes in reality are operated under closed loop, a plant-wide control scheme was employed when applying the process monitoring methods in Chapter 10. In [141, 142], four different plant-wide control structures using only Proportional (P) and Proportional-Integral (PI) controllers were investigated for the TEP. The second control structure listed in [141, 142] was chosen for this book because this structure provided the best performance according to the authors.

The control structure implemented to obtain the results in Chapter 10 is shown schematically in Figure 8.1. The control structure consists of nineteen loops, and the values of the control parameters and other details of the control structure are listed in Table 8.5. The exact values for the controller gains implemented by the author of [141] could not be determined because the controller gains were scaled to be dimensionless and the scalings on the controller inputs and outputs were not presented. However, we estimated the controller parameters based on the values from [141], and these parameters are reported in Table 8.5 with units consistent with the manipulated and measurement variables [39]. Some closed loop simulations with the control parameters from Table 8.5 are shown in Figures 8.2 and 8.3. A comparison of these plots with those in [141] indicates that relatively similar values for the control parameters were employed for both sets of simulations. The simulation code for the process in closed loop can be downloaded from http://brahms.scs.uiuc.edu.

8.7 Homework Problems

1. Plot the manipulated and measured variables over time for one of the process faults in Table 8.4 using the closed loop controllers described in this chapter (the code can be downloaded from http://brahms.scs.uiuc.edu). Explain how the effect of the process fault propagates through the plant, as indicated by the process variables. What is the physical mechanism for each of the process variable changes? Does each variable change in the way you would expect? Explain. For each variable, explain how its time history is affected by the closed loop controllers. Which controllers mask the effect of the fault on the process variables? [Note to instructor: consider assigning a different fault to each student in the class.]
2. Describe the step-by-step procedure used to arrive at the plant-wide control structure used in this chapter (hint: read [142]).

Table 8.5. Control structure and parameter description

Loop Number	Manipulated Variable	Control Variable	Primary (P) Secondary (S)	Gain	Integral Time (sec.)	Sampling Interval (sec.)
1	XMV(1)	XMEAS(2)	S	0.172	∞	3
2	XMV(2)	XMEAS(3)	S	0.120	∞	3
3	XMV(3)	XMEAS(1)	S	98.3	∞	3
4	XMV(4)	XMEAS(4)	S	6.56	∞	3
5	XMV(5)	XMEAS(5)	–	−0.157	1	3
6*	XMV(6)	XMEAS(10)	S	122	∞	3
7	XMV(7)	XMEAS(12)	–	−2.94	∞	3
8	XMV(8)	XMEAS(15)	–	−2.31	∞	3
9	XMV(9)	XMEAS(19)	S	0.0891	∞	3
10	XMV(10)	XMEAS(21)	S	−1.04	1452	3
11	XMV(11)	XMEAS(17)	–	2.37	2600	3
12	–	–	–	–	–	–
13	XMEAS(1)	XMEAS(23)	P	1770	3168	360
14	XMEAS(2)	XMEAS(26)	P	0.143	3168	360
15	XMEAS(3)	XMEAS(27)	P	0.0284	5069	360
16	XMEAS(19)	XMEAS(18)	P/S	0.0283	236	3
17	XMEAS(4)	XMEAS(8)	P	14.6	3168	3
18	XMEAS(21)	XMEAS(9)	P	12.6	982	3
19	XMEAS(10)	XMEAS(30)	P	−2133	6336	360
20	XMEAS(18)	XMEAS(38)	P	−157	12408	900
21	–	–	–	–	–	–

The valve for control loop 6 is completely open for pressures above 2950 kPa gauge and closed for pressures below 2300 kPa gauge

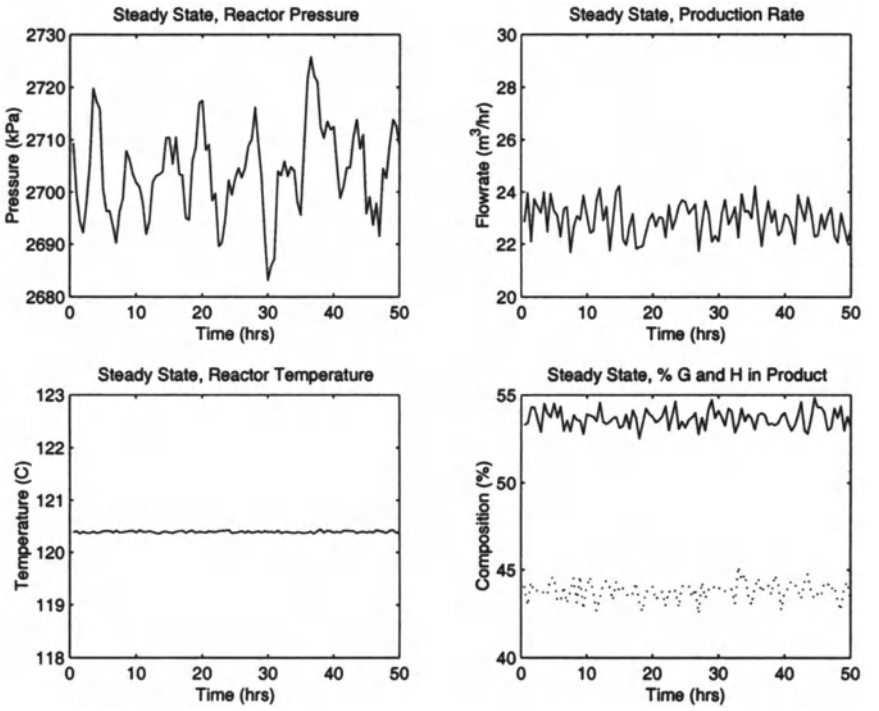

Fig. 8.2. Closed loop simulation for the steady state case with no faults. The solid and dotted lines in the lower right plot represent the compositions of G and H, respectively.

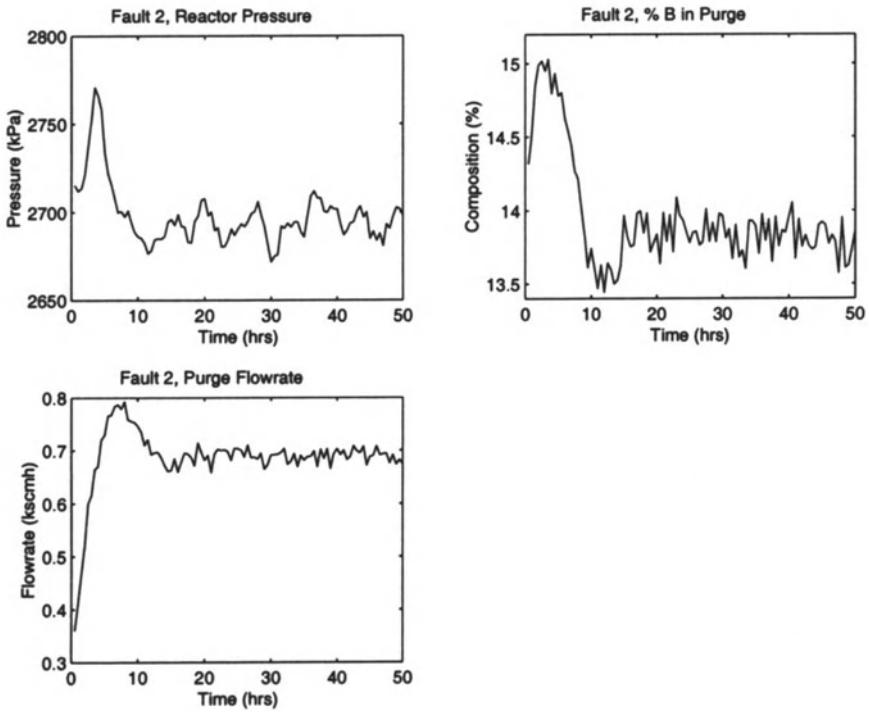

Fig. 8.3. Closed loop simulation for a step change in the composition of the inert B (IDV(2) in Table 8.4)

CHAPTER 9
APPLICATION DESCRIPTION

9.1 Introduction

Chapter 8 describes the process, the control system, and the type of faults for the Tennessee Eastman plant simulator. In Chapter 10, this simulator will be used to demonstrate and compare the various process monitoring methods presented in Part III. The process monitoring methods are tested on data generated by the TEP simulation code, operating under closed loop with the plant-wide control structure discussed in Section 8.6. The original simulation code allows 20 preprogrammed faults to be selectively introduced to the process [39]. We have added an additional fault simulation, which results in a total of 21 faults as shown in Table 8.4. In addition to the aforementioned aspects of the process, the process monitoring performance is dependent on the way in which the data are collected, such as the sampling interval and the size of the data sets.

The purpose of this chapter is to describe the data sets and to present the process monitoring measures employed for comparing the process monitoring methods. Section 9.2 describes how the data in the training and testing sets were generated by the TEP. A discussion on how the selection of the sampling interval and sample size of the data sets affects the process monitoring methods follows in Sections 9.3 and 9.4, respectively. Section 9.5 discusses the selection of the lag and order for each method. Sections 9.6, 9.7, and 9.8 present the measures investigated for fault detection, identification, and diagnosis, respectively. The process monitoring methods (covered in Parts II and III) used for these purpose are collected into Tables 9.2-9.4 which show how the methods are related.

9.2 Data Sets

The data in the training and testing sets included all the manipulated and measured variables (see Tables 8.1-8.3), except the agitation speed of the reactor's stirrer for a total of $m = 52$ observation variables. (The agitation speed was not included because it was not manipulated.) An observation vector at a particular time instant is given by

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

$$\mathbf{x} = [\text{XMEAS}(1), \cdots, \text{XMEAS}(41), \text{XMV}(1), \cdots, \text{XMV}(11)]^T. \quad (9.1)$$

The observations were simulated with an integration step size of 1 second, and this did not produce any numerical inaccuracies. Although some of the observations are sampled continuously while other variables contain time delays (see Section 8.3), it simplifies the implementation to employ the same sampling interval for each variable when the data are collected for calculating multivariate process monitoring measures. A sampling interval of 3 minutes was used to collect the simulated data for the training and testing sets.

The data in the training set consisted of 22 different simulation runs, where the random seed was changed between each run. One simulation run (Fault 0) was generated with no faults; another simulation run (Fault 21) was generated by fixing the position of the valve for Stream 4 at the steady state position; and, each of the other 20 simulation runs (Faults 1-20) was generated under a different fault, each corresponding to a fault listed in Table 8.4. The simulation time for each run was 25 hours. The simulations started with no faults, and the faults were introduced 1 simulation hour into the run. The total number of observations generated for each run was $n = 500$, but only 480 observations were collected after the introduction of the fault. It is only these 480 observations actually used to construct the process monitoring measures.

The data in the testing set also consisted of 22 different simulation runs, where the random seed was changed between each run. These simulation runs directly correspond to the runs in the training set (Faults 0-21). The simulation time for each run was 48 hours. The simulation started with no faults, and the faults were introduced 8 simulation hours into the run. The total number of observations generated for each run was $n = 960$.

9.3 Sampling Interval

The amount of time in which quality data are collected from chemical processes during either in-control or out-of-control operations is usually limited in practice. Typically, only a small portion of the operation time exists where it can be determined with confidence that the data were not somehow corrupted and no faults occurred in the process. Also, the process supervisors do not generally allow faults to remain in the process for long periods of time for the purpose of producing data used in fault diagnosis algorithms.

Typically data collected during faulty operations are stored in historical databases in which engineers or operators diagnose the faults sometime after the fault occurs, and then enter that information into the historical database. The amount of such data available in the historical database is typically fixed and the sampling interval for the process monitoring methods needs to be determined.

It is desirable to detect, identify, and diagnose faults as soon as possible. This suggests a fast sampling rate. Also, given a fixed time $T = n\Delta t$, it is beneficial from an information point-of-view to sample as fast as possible ($\Delta t \to 0$, $n \to \infty$). There are in terms of process monitoring, however, three possible problems with sampling as fast as possible. For the amount of data produced, the computational requirements may exceed the computational power available. Additionally, the model fit may be concentrated to the higher frequencies, where measurement noise is predominant. When identifying an ARX model via a least squares approach, Ljung [137] shows how the bias is shifted when sampling with higher frequencies. This bias shift for fast sampling rates may be undesirable, especially if the faults primarily affect the lower frequency dynamics of the process. Finally, statistics that ignore serial correlation will generally perform more poorly for short sampling times.

The choice of the sampling interval for process monitoring is usually selected based on engineering judgment. For system identification, a rule of thumb is to set the sampling interval to one-tenth the time constant of the process [137]. Considering that many of the time constants of the Tennessee Eastman problem under closed loop appear to be about 2 hours (see Figure 8.2), it is advisable from a system identification point of view to sample at an interval of 12 minutes. This does not, however, take advantage of the instrumentation of the process, which allows much faster sampling rates (see Section 8.3). A sampling interval of 3 minutes was selected here to allow fast fault detection, identification, and diagnosis, and to allow a good comparison between techniques that either take into account or ignore serial correlations. In addition, the same sampling interval has been used in other applications of process monitoring to the TEP [125, 26, 74].

An alternative approach would be to average each measurement over a period of time before using the data in the process monitoring algorithms. This and similar "moving window" techniques will generally reduce normal process variability and hence produce a more sensitive process monitoring method. However, this comes at a cost of delaying fault detection. Wise and co-workers [230] pointed out that the width of the windows (*i.e.*, the number of data points used to compute the average) had an important effect on the performance. In general, a "wide" window allows the detection of smaller changes, but does not respond as quickly to changes as "narrow" windows.

9.4 Sample Size

As mentioned in the previous section, the total time spanned by the training set is generally limited. In the cases where the total time $T = n\Delta t$ is fixed, the selection of the sampling interval Δt and the sample size n cannot be decoupled. Therefore, the effect of the sampling interval on the sample size should be considered when selecting the sampling interval, and *vice versa*.

An important consideration for the sample size is the total number of independent parameters contained in the model being identified. It is desirable to have the number of model parameters be much smaller than the total number of process variables m multiplied by the total number of observations n.

Because the data for this book are simulated by the TEP, the sample size is not limited by T and can be considered separately from the sampling interval. Downs and Vogel [39] recommend a simulation time between 24 and 48 hours to realize the full effect of the faults. With a sampling interval equal to 3 minutes, 24 to 48 hours of simulation time contain $n = 480$ to 960 observations. Simulations (see Figure 8.3) suggest that a run containing 24 simulation hours sufficiently captures the significant shifts in the data produced by the fault.

The sufficiency of the sample size for the training set $n = 480$ can be determined by examining the total number of independent parameters associated with the orders of the various process monitoring methods (see Table 9.1). The total number of states in the closed loop process is $k = 61$; 50 states from the open loop process plus 11 states from the PI controllers. For a state space model of state order $k = 61$ with 11 inputs and 41 outputs, the number of independent parameters M_k is equal to 6985 according to (7.36). For fault detection using the PCA-based T^2 statistic (4.12), the number of estimated parameters M_a is equal to the number of independent degrees of freedom of the matrix product of $P\Sigma_a^{-2}P^T$ in (4.12), which is calculated from

$$M_a = \frac{a + 2am - a^2}{2}. \tag{9.2}$$

For $a = 51$, the number of independent parameters is 1377. For fault detection using the CVA-based T_s^2 statistic (7.41), the number of estimated parameters M_k is equal to the number of independent degrees of freedom of $J_k^T J_k$ in (7.41), which is calculated from

$$M_k = \frac{k + 2kmh - k^2}{2}. \tag{9.3}$$

For $h = 2$ and $k = 61$, the number of independent parameters is 4029. The total number of data points in the training set is equal to $nm = (480)(52) = 24,960$. The absolute minimum requirement to apply the PCA, CVA, or state space model at a given order is that the number of data points is greater than the number of independent parameters in the model. The ratio of the number of data points to the number of independent parameters is $nm/M_k = (480)(52)/6985 = 3.57$ for the state space model, $nm/M_a = 18.1$ for the PCA-based model, and $nm/M_k = 5.53$ for the CVA-based model. With all other variables being equal (e.g., the noise level), the larger the ratio is greater than one, the higher the accuracy of the model. For this data set, all ratios are greater than one, indicating that the size of the training set ($n = 480$) is sufficient to apply the PCA, CVA, and state space model.

Reducing the order may still result in a higher quality model, depending on the noise level. As shown in Table 9.1, the state space model requires the largest number of independent parameters, followed by CVA, and PCA. A PCA model of a given order has significantly less independent parameters, but does not take into account serial correlations.

Table 9.1. The number of independent parameters estimated for the various models and orders

Order†	Inputs	Outputs	Parameters State Space††	Parameters PCA†††	Parameters CVA††††
1	11	41	1405	52	104
11	11	41	2335	517	1089
21	11	41	3265	882	1974
31	11	41	4195	1147	2759
41	11	41	5125	1312	3444
51	11	41	6055	1377	4029
61	11	41	6985	–	4514

† The order is equal to a for PCA and the state order k for the state space model and CVA
†† The number of parameters is based on (7.36)
††† The number of parameters is based on (9.2)
†††† The number of parameters is based on (9.3), using $h = 2$ lags

9.5 Lag and Order Selection

The number of lags included in the DPCA, DFDA, and CVA process monitoring methods can substantially affect the monitoring performance. It is best to choose the number of lags as the minimum needed to accurately capture the dynamics of the process. Choosing the number of lags larger than necessary may significantly decrease the robustness of the process monitoring measures, since the extra dimensionality captures additional noise, which may be difficult to characterize with limited data. The procedure used for this book follows Larimore's suggestion of selecting the number of lags h as that minimizing the small sample AIC criterion using an ARX model (see Section 7.5). This ensures that the number of lags is large enough to capture all the statistically significant information in the data.

As described in Part III, the selection of the reduction order is critical to developing efficient measures for process monitoring. The order selection methods described in Part III will be used. The parallel analysis method (see Section 4.3) is applied to select a in PCA and DPCA. The information criterion (5.12) is used to determine a for FDA and DFDA. The small sample AIC (7.37) is applied to CVA to determine the state order k.

Although it is popularly referred to in the literature, the cross-validation method is not used here for any of the process monitoring methods. Cross-

validation is computationally expensive when dealing with several large data sets. More importantly, there can be problems with cross-validation when serial correlations in the data exist [125].

9.6 Fault Detection

The proficiencies of PCA, DPCA, and CVA for detecting faults were investigated on the TEP. The measures applied for each method, the corresponding equation numbers, and the distributions used to determine the thresholds for the measures are listed in Table 9.2. For instance, the first row indicates that PCA is used to generate the T^2 statistic according to (4.12) and the threshold is calculated according to (4.14). The distribution listed as "TR" means that the threshold is set to be the tenth highest value for Fault 0 of the *testing* set, in which the number of observations $n = 960$. The threshold corresponds to a level of significance $\alpha = 0.01$ by considering the probability distribution of the statistics for Fault 0. A thorough discussion of the measures is available in the respective chapters, and more information related to applying these measures to the TEP is contained in Section 10.6.

Table 9.2. The measures employed for fault detection

Method	Basis	Equation	Distribution
PCA	T^2	4.12	4.14
PCA	Q	4.21	4.22
DPCA	T^2	4.12†	4.14†
DPCA	Q	4.21†	4.22†
CVA	T_s^2	7.41	7.42
CVA	T_r^2	7.43	7.41
CVA	Q	7.46	TR††

† Applied to the data matrix with lags

†† TR - Threshold set based on testing data for Fault 0

There exist techniques to increase the sensitivity and robustness of the PCA and DPCA process monitoring measures as described in Section 4.8, for example, through the use of the CUSUM or EWMA version of the measures. However, these techniques compromise the response time of the measures. Although such techniques can be highly useful in practice, the process monitoring methods applied in Chapter 10 do not employ them because it would complicate the comparison of the process monitoring methods. The measures investigated for each process monitoring method are designed to detect

and diagnose the faults with the smallest delay. Applying the CUSUM and EWMA versions of PCA and DPCA is left as a homework problem.

9.7 Fault Identification

The proficiencies of PCA, DPCA, and CVA for identifying faults were investigated on the TEP. The measures applied for each method and the corresponding equation numbers are presented in Table 9.3. A discussion on how to apply the measures based on PCA, DPCA, and CVA can be found in Sections 4.5, 4.7, and 7.8, respectively. A thorough discussion of the measures is available in the respective chapters, and more information related to applying these measures to the TEP is contained in Section 10.7.

Table 9.3. The measures employed for fault identification

Method	Basis	Equation
PCA	*CONT*	4.25
PCA	*RES*	4.32
DPCA	*CONT*	4.25 with 4.44
DPCA	*RES*	4.32 with 4.44
CVA	*CONT*	4.25 with 7.22
CVA	*RES*	4.32 with 7.45

9.8 Fault Diagnosis

The proficiencies of the fault diagnosis methods described in Part III were investigated on the TEP. Fault diagnosis measures based on discriminant analysis that use no dimensionality reduction are given in (3.7). When this multivariate statistic (MS) is applied to data with no lags, it will be referred to as the T_0^2 statistic. When the multivariate statistic is applied to data with 1 lag, it will be referred to as the T_1^2 statistic. These are considered in Chapter 10 to serve as a benchmark for the other measures, as the dimensionality should only be reduced if it decreases the misclassification rate for a testing set. The fault diagnosis measures and the corresponding equation or section numbers are presented in Table 9.4. The statistic(s) which each measure is based upon is also listed in the table. A thorough discussion of the measures are available in the respective chapters, and more information related to applying these measures to the TEP is contained in Section 10.8.

Table 9.4. The measures employed for fault diagnosis

Method	Basis	Equation/Section
PCAm	T^2	Equation 4.35†
PCA1	T^2	Equation 4.33†
PCAm	Q	Equation 4.37
PCAm	T^2 & Q	Equation 4.38††
DPCAm	T^2	Equations 4.35† and 4.44
DPCAm	Q	Equations 4.37 and 4.44
DPCAm	T^2 & Q	Equations 4.38†† and 4.44
FDA	T^2	Equation 5.16†
FDA/PCA1	T^2	Equations 5.17† and 5.16
FDA/PCA2	T^2	Equations 5.17† and 5.16
DFDA/DPCA1	T^2	Equations 5.17†, 5.16, and 4.44
CVA	T_s^2	Equations 4.35† and 7.41
CVA	T_r^2	Equations 4.35 and 7.43
CVA	Q	Equations 4.37 and 7.46
PLS1	–	Section 6.3
PLS2	–	Section 6.3
$PLS1_{adj}$	–	Section 6.4
$PLS2_{adj}$	–	Section 6.4
MS	T_0^2	Equation 3.7
MS	T_1^2	Equation 3.7

† Applied to the score space only

†† $c_i = 0.5$ and $\alpha = 0.01$

CHAPTER 10
RESULTS AND DISCUSSION

10.1 Introduction

In this chapter, the process monitoring methods in Part III are compared and contrasted through application to the Tennessee Eastman plant simulator (TEP). The proficiencies of the process monitoring statistics listed in Tables 9.2-9.4 are investigated for fault detection, identification, and diagnosis. The evaluation and comparison of the statistics are based on criteria that quantify the process monitoring performance. To illustrate the strengths and weaknesses of each statistic, Faults 1, 4, 5, and 11 are selected as specific case studies in Sections 10.2, 10.3, 10.4, and 10.5, respectively. Sections 10.6, 10.7, and 10.8 present and apply the quantitative criteria for evaluating the fault detection, identification, and diagnosis statistics, respectively. The *overall* results of the statistics are evaluated and compared. Results corresponding to the case studies are highlighted in boldface in Tables 10.6 to 10.20.

10.2 Case Study on Fault 1

In the normal operating condition (Fault 0), Stream 4 in Figure 8.1 contains 0.485, 0.005, and 0.510 mole fraction of A, B, and C, respectively [39]. When Fault 1 occurs, a step change is induced in the A/C feed ratio in Stream 4, which results in an increase in the C feed and a decrease in the A feed in Stream 4. This results in a decrease in the A feed in the recycle Stream 5 and a control loop reacts to increase the A feed in Stream 1 (see Figure 10.1). These two effects counteract each other over time, which results in a constant A feed composition in Stream 6 after enough time (see Figure 10.2).

The variations the flowrates and compositions of Stream 6 to the reactor causes variations in the reactor level (see Figure 8.1), which affects the flowrate in Stream 4 through a cascade control loop (see Figure 10.3). The flowrate of Stream 4 eventually settles to a steady state value lower than its value at the normal operating conditions.

Since the ratio of the reactants A and C changes, the distribution of the variables associated with material balances (*i.e.*, level, pressure, composition) changes correspondingly. Since more than half of the variables monitored deviate significantly from their normal operating behavior, this fault is expected

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

to be easily detected. Process monitoring statistics that show poor performance on Fault 1 are likely to perform poorly on other faults as well.

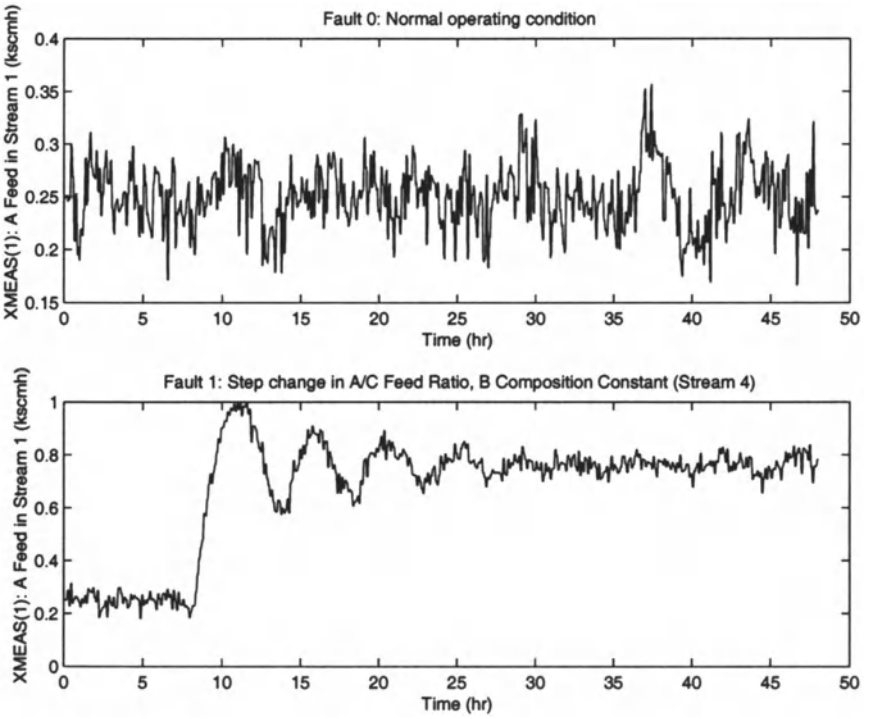

Fig. 10.1. Comparison of XMEAS(1) for Faults 0 and 1

The (D)PCA-based and CVA-based statistics for fault detection are shown in Figures 10.4 and 10.5, respectively. The dotted line in each figure is the threshold for the statistic, the statistic above its threshold indicates that a fault is detected (the statistic is shown as a solid line). The first eight hours were operated under normal operating conditions. Thus, all statistics are expected to fall below the thresholds for the first eight hours, which they did. The quantitative fault detection results are shown in Table 10.1. All of the statistics produced nearly zero missed detection rates. For a fault that significantly changes the distribution of the variables monitored, all fault detection statistics perform very well.

Assuming that process data collected during a fault are represented by a previous fault class, the objective of the fault diagnosis statistics in Table 9.4 is to classify the data to the *correct* fault class. That is, a highly proficient fault diagnosis statistic produces small misclassification rates when applied

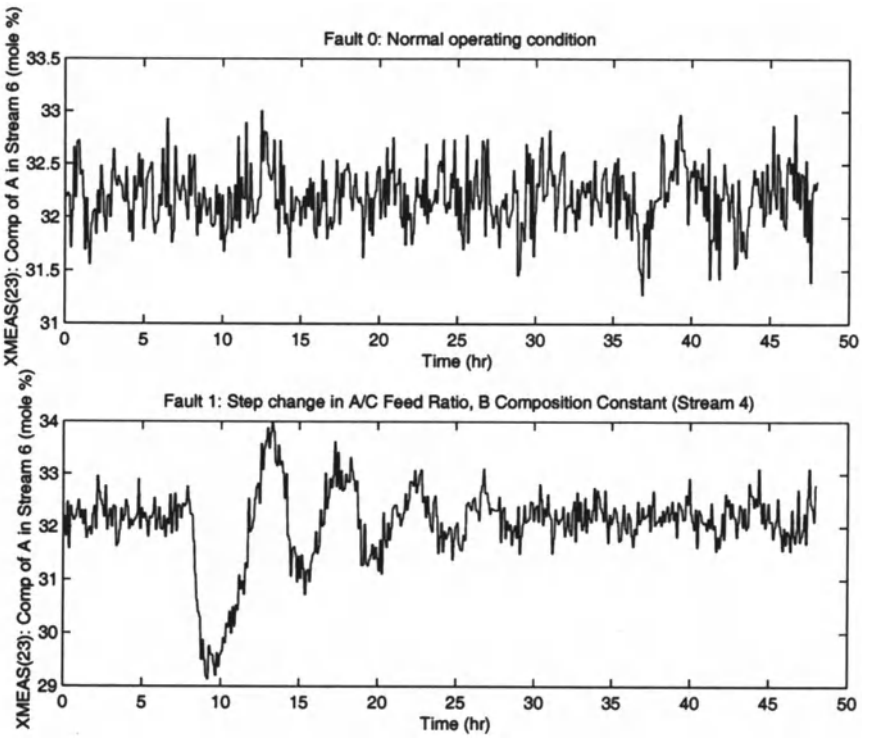

Fig. 10.2. Comparison of XMEAS(23) for Faults 0 and 1

Table 10.1. Missed detection rates for Faults 1, 4, 5, and 11

Method	Fault Basis	1	4	5	11
PCA	T^2	0.008	0.956	0.775	0.794
PCA	Q	0.003	0.038	0.746	0.356
DPCA	T^2	0.006	0.939	0.756	0.801
DPCA	Q	0.005	0	0.748	0.193
CVA	T_s^2	0.001	0.688	0	0.515
CVA	T_r^2	0	0	0	0.195
CVA	Q	0.003	0.975	0	0.669

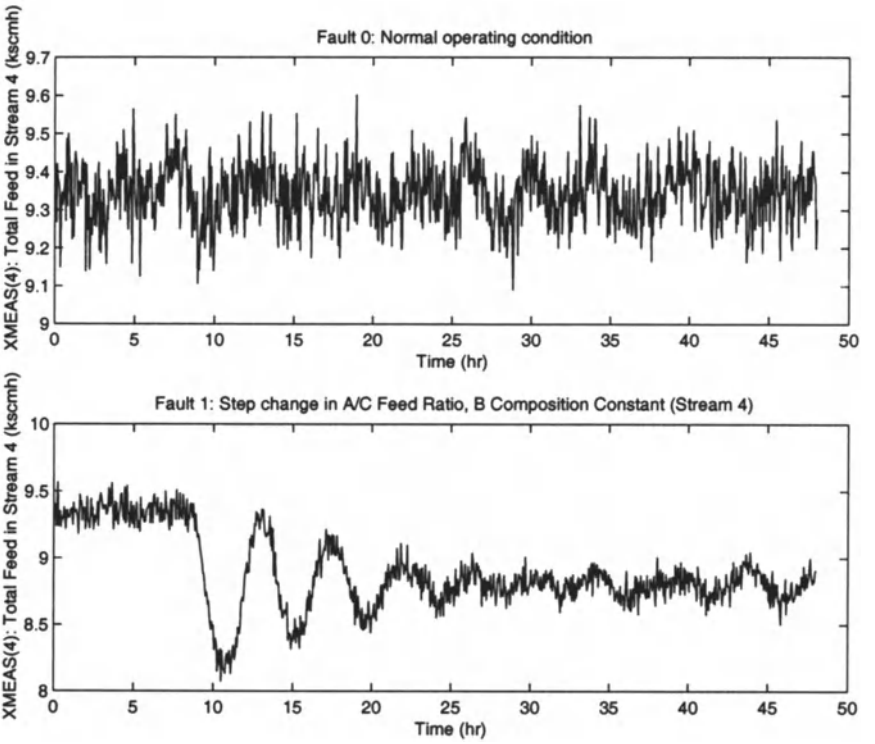

Fig. 10.3. Comparison of XMEAS(4) for Faults 0 and 1

to data independent of the training set. As shown in Table 10.2, most of the fault diagnosis statistics performed very well (Fault 1 being correctly diagnosed $> 96\%$ of the time).

10.3 Case Study on Fault 4

Fault 4 involves a step change in the reactor cooling water inlet temperature (see Figure 8.1). A significant effect of Fault 4 is to induce a step change in the reactor cooling water flowrate (see Figure 10.6). When the fault occurs, there is a sudden temperature increase in the reactor (see Figure 10.7 at time = 8 hr), which is compensated by the control loops. The other 50 measurement and manipulated variables remain steady after the fault occurs; the mean and standard deviation of each variable differ less than 2% between Fault 4 and the normal operating condition. This makes the fault detection and diagnosis tasks more challenging than for Fault 1.

Table 10.2. The overall misclassification rates for Faults 1, 4, 5, and 11

Method	Fault Basis	1	4	5	11
PCAm	T^2	0.680	0.810	0.956	0.989
PCA1	T^2	0.024	0.163	0.021	0.234
PCAm	Q	0.028	0.951	0.913	0.859
PCAm	T^2&Q	0.041	1.000	0.973	0.968
DPCAm	T^2	0.880	0.720	0.874	0.948
DPCAm	Q	0.035	0.964	0.856	0.843
DPCAm	T^2&Q	0.038	1.000	1.000	0.983
PLS1	–	0.013	0.170	0.006	0.989
PLS2	–	0.013	0.119	0.008	0.979
$PLS1_{adj}$	–	0.019	0.364	0.044	0.859
$PLS2_{adj}$	–	0.019	0.320	0.043	0.886
CVA	T_s^2	0.028	0.981	0.061	0.904
CVA	T_r^2	0.026	0.358	0.040	0.139
CVA	Q	0.245	0.890	0.174	0.901
FDA	T^2	0.025	0.176	0.020	0.245
FDA/PCA1	T^2	0.024	0.163	0.020	0.244
FDA/PCA2	T^2	0.025	0.176	0.020	0.245
DFDA/DPCA1	T^2	0.026	0.159	0.023	0.118
MS	T_0^2	0.025	0.178	0.020	0.245
MS	T_1^2	0.035	0.427	0.040	0.121

The extent to which the (D)PCA-based and CVA-based statistics are sensitive to Fault 4 can be examined in Figure 10.8 and Figure 10.9 respectively. The quantitative fault detection results are shown in Table 10.1. The variation in the residual space was captured by T_r^2, but not by the CVA-based Q statistic. The potential advantage of applying T_r^2 to capture variation in the residual space is clearly shown. It is interesting to see that the PCA and DPCA-based Q statistics were able to detect Fault 4, but the CVA-based Q statistic did not. The CVA-based T_s^2 statistic passes the threshold much of time after the fault occurs, but does not have the persistence of the CVA-based T_r^2 statistic (see Figure 10.9). Although the PCA and DPCA-based Q statistics both are able to detect the fault, the DPCA-based Q statistic outperformed the PCA-based statistic in terms of exceeding the threshold by a greater degree. This indicates the potential advantage of taking serial correlation into account when developing fault detection procedures.

For this fault the PCA and DPCA-based Q statistics were more sensitive than the PCA and DPCA-based T^2 statistics, and the CVA-based T_r^2

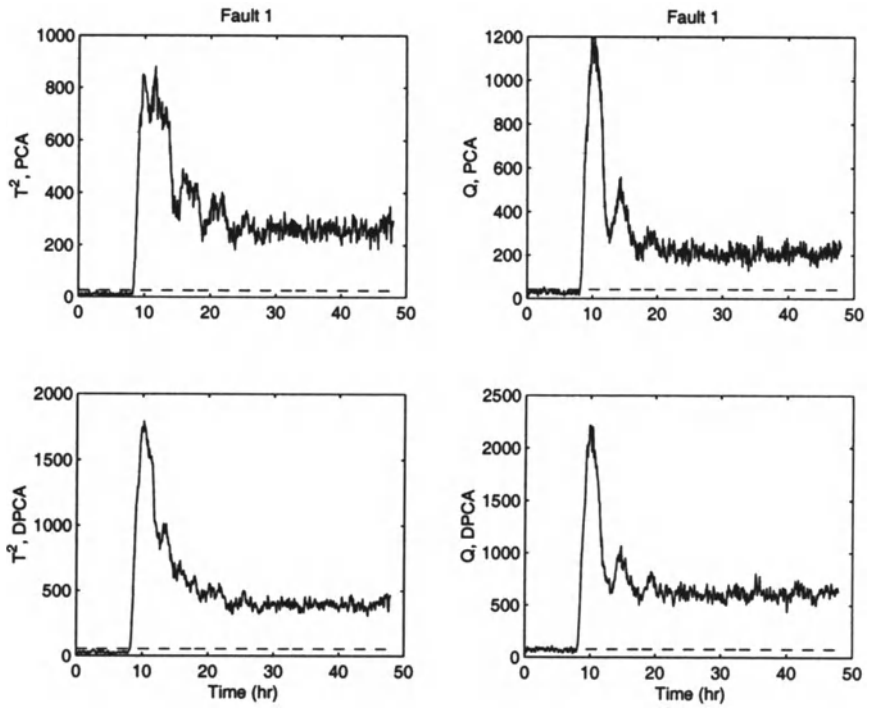

Fig. 10.4. The (D)PCA multivariate statistics for fault detection for Fault 1

statistic was more sensitive than the CVA-based T_s^2 statistic (see Table 10.1). These statistics quantifying variations in the residual space were overall more sensitive to Fault 4 than the statistics quantifying the variations in the score or state space. In other words, the fault created new states in the process rather than magnifying the states based on in-control operations. Although this conclusion does not hold for all faults, it certainly is true for a large portion of them.

Recall that Fault 4 is associated with a step change in the reactor cooling water inlet temperature (see Table 8.4), which is unmeasured. Engineering judgment and an examination of Figure 8.1 and Tables 8.1-8.3 indicate that the most closely related observation variable is the reactor cooling water flowrate. The fault identification statistics in Table 9.3 provide a rank ordering of the observation variables from most relevant to least relevant in terms of being associated with the fault. For Fault 4, the third column of Table 10.3 lists where the reactor cooling water flowrate was ranked by the various fault identification methods. All of the methods correctly ranked the

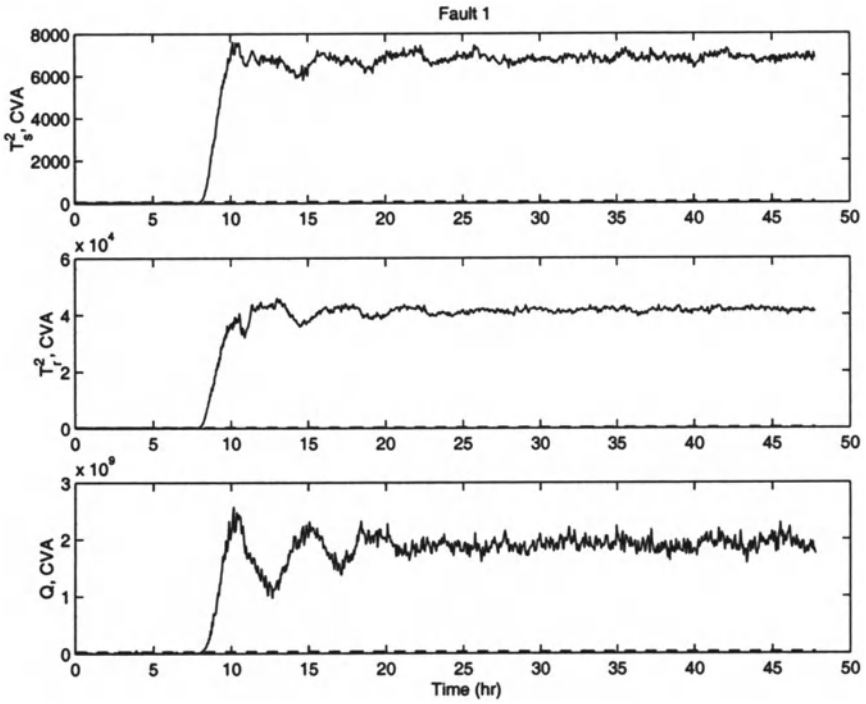

Fig. 10.5. The CVA multivariate statistics for fault detection for Fault 1

reactor cooling water flowrate as most closely related to Fault 4 except for the CVA-based *CONT* statistic.

Table 10.3. The overall rankings for Faults 4 and 11

Method	Fault Basis	4	11
PCA	*CONT*	1	1
PCA	*RES*	1	1
DPCA	*CONT*	1	1
DPCA	*RES*	1	1
CVA	*CONT*	11	13
CVA	*RES*	1	1

The CVA-based *CONT* statistic did not perform well because the inverse of the matrix $\hat{\Sigma}_{pp}$ in (7.40) allowed certain observation variables to dominate the statistic. In particular, the maximum values of the J_k matrix corresponding to the observation variables x_{12}, x_{15}, x_{17}, x_{48}, x_{49}, and x_{52} are above 50

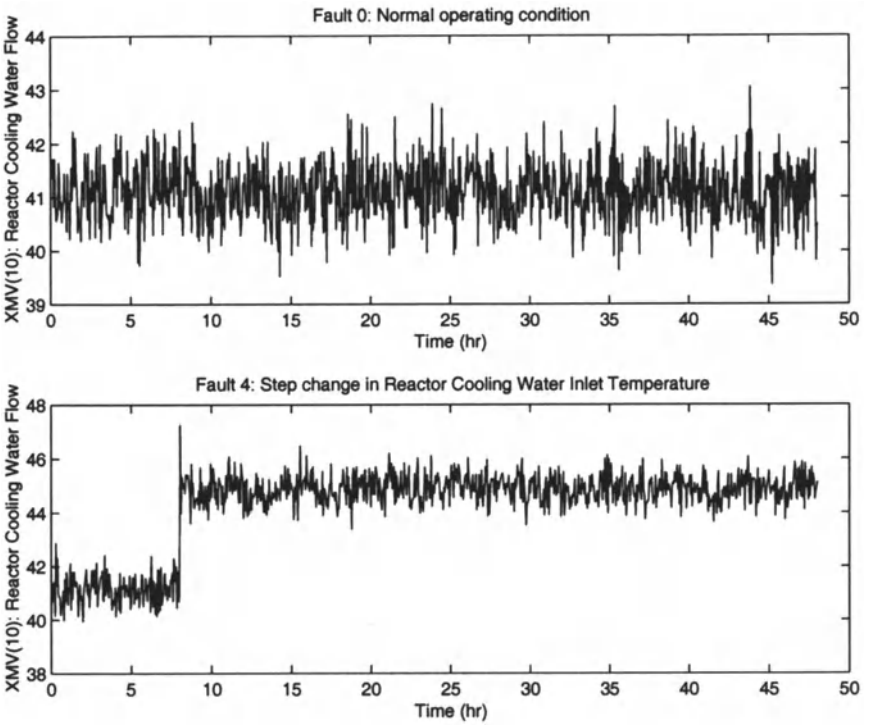

Fig. 10.6. Comparison of XMV(10) for Faults 0 and 4

while the elements of J_k corresponding to all the other variables are less than 3 (see Figure 10.10). The dominance of the observation variables x_{12}, x_{15}, x_{17}, x_{48}, x_{49}, and x_{52} in J_k was observed for all of the other faults investigated as well.

For fault diagnosis, many of the statistics performed poorly for Fault 4 (see Table 10.2). PLS2 gave the lowest misclassification rates. This indicates that discriminant PLS can outperform FDA for some faults although it would be expected theoretically that FDA should be better in most cases. PLS1 had a similar misclassification rate as all the FDA-based statistics, PCA1, and MS T_0^2. PLS1 and PLS2 gave significantly lower misclassification rates than PLS1_{adj} and PLS2_{adj}. This makes the point that the adjustment procedure described in Section 6.4 does not always improve fault diagnosis.

DFDA/DPCA1 produced similar misclassification rates as the static FDA methods. However, including lagged variables actually degraded the performance of the MS statistic.

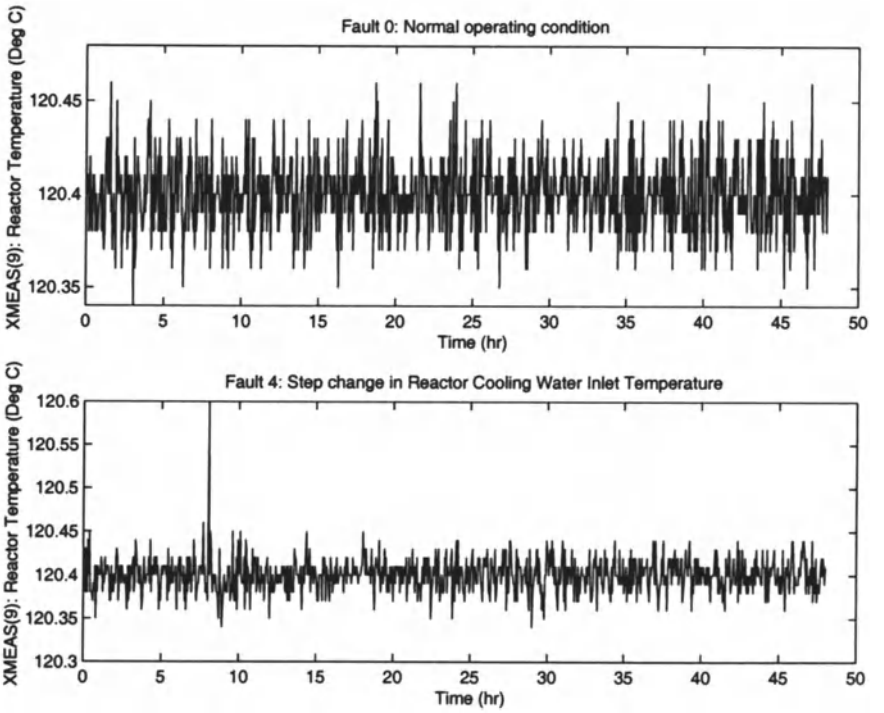

Fig. 10.7. Comparison of XMEAS(9) for Faults 0 and 4

10.4 Case Study on Fault 5

Fault 5 involves a step change in the condenser cooling water inlet temperature (see Figure 8.1). The significant effect of the fault is to induce a step change in the condenser cooling water flowrate (see Figure 10.11). When the fault occurs, the flowrate of the outlet stream from the condenser to the vapor/liquid separator also increases, which results in an increase in temperature in the vapor/liquid separator, and thus the separator cooling water outlet temperature (see Figure 10.12). Similar to Fault 4, the control loops are able to compensate for the change and the temperature in the separator returns to its setpoint. The time it takes to reach the steady state is about 10 hours. For the rest of the 50 variables that are being monitored, 32 variables have similar transients that settle in about 10 hours. Detecting and diagnosing such a fault should not be a challenging task.

The (D)PCA-based and CVA-based statistics for fault detection are shown in Figures 10.13 and 10.14, respectively. The quantitative fault detection results are shown in Table 10.1, where it is seen that the (D)PCA-based

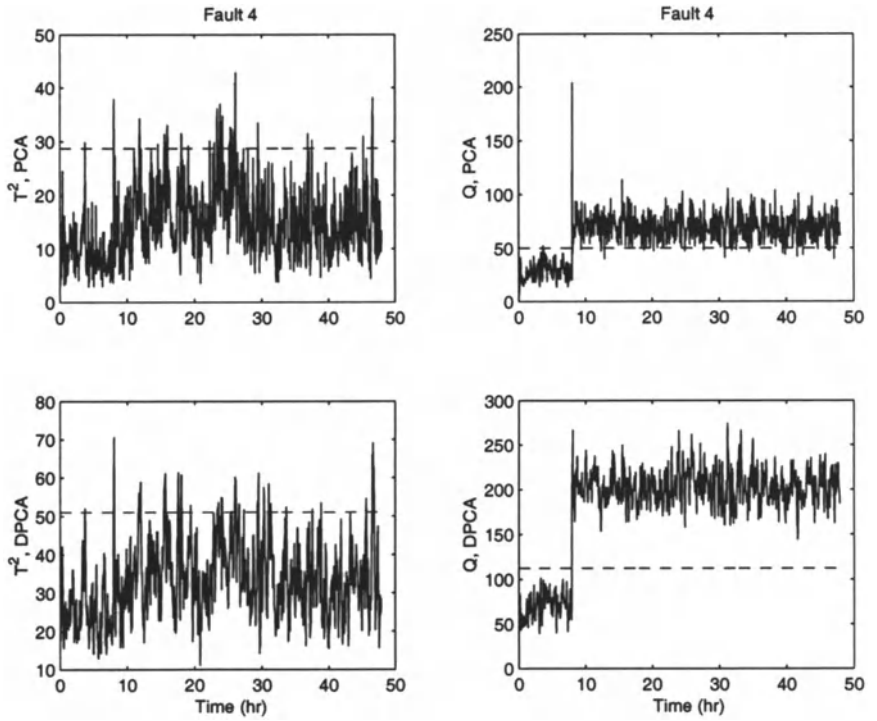

Fig. 10.8. The (D)PCA multivariate statistics for fault detection for Fault 4

statistics had a high missed detection rate, and all the CVA statistics had a zero missed detection rate. The reason for the apparent poor behavior of the (D)PCA-based statistics is clear from plotting the observation variables over time. Most variables behaved similarly to Figure 10.12—they returned to their setpoints 10 hours after the fault occurred. The (D)PCA-based statistics fail to indicate a fault 10 hours after the fault occurs (see Figure 10.13). On the other hand, all the CVA statistics stayed above their thresholds (see Figure 10.14).

The persistence of a fault detection statistic (the CVA statistic in this case) is important in practice. At any given time a plant operator has several simultaneous tasks to perform and typically does not focus on all tasks with the same degree of attentiveness. Also, it usually takes a certain amount of time to track down the cause of abnormal process operation. When the time to locate the source of a fault is longer than the persistence of the fault detection statistic, a plant operator may conclude that the fault has "corrected itself" and assume that the process is again operating in normal operating conditions. In contrast, a persistent fault detection statistic will

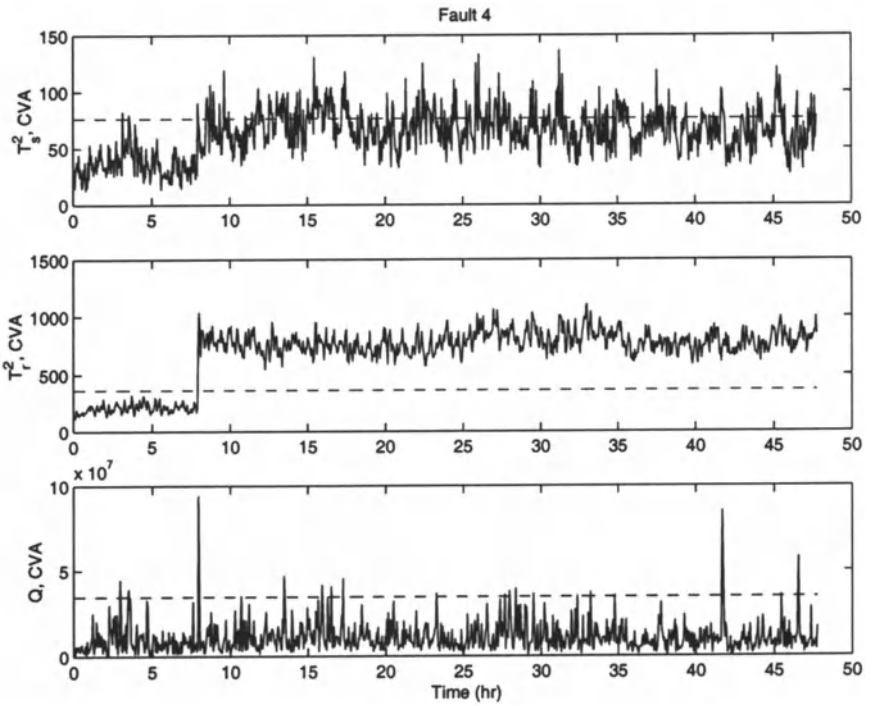

Fig. 10.9. The CVA multivariate statistics for fault detection for Fault 4

continue to inform the operator of a process abnormality although all the process variables will appear to have returned to their normal values.

It is somewhat interesting that examination of the canonical variables ($J\mathbf{p_t}$) for Fault 5 reveals that the canonical variable corresponding to the 99th generalized singular value is solely responsible for the out-of-control T_r^2 values between 10-40 hours after the fault occurred.

10.5 Case Study on Fault 11

Similar to Fault 4, Fault 11 induces a fault in the reactor cooling water inlet temperature. The fault in this case is a random variation. As seen in Figure 10.15, the fault induces large oscillations in the reactor cooling water flowrate, which results in a fluctuation of reactor temperature (see Figure 10.16). The other 50 variables are able to remain around the setpoints and behave similarly as in the normal operating conditions.

The extent to which the (D)PCA-based and CVA-based statistics are sensitive to Fault 11 can be examined in Figure 10.17 and Figure 10.18,

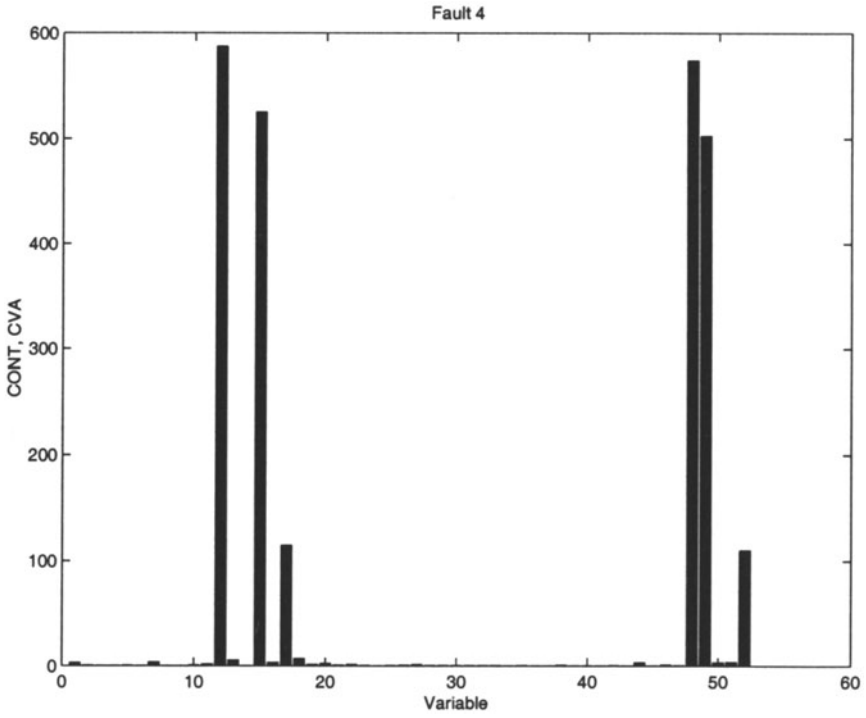

Fig. 10.10. The average contribution plot for Fault 4 for the CVA-based *CONT*

respectively. The quantitative fault detection results are shown in Table 10.1. The (D)PCA-based Q statistics performed better than the (D)PCA-based T^2 statistics. Similarly to Fault 4, the variation in residual space was captured better by T_r^2 than the CVA-based Q statistic. Overall, the DPCA-based Q statistic gave the lowest missed detection rate (see Table 10.1).

As Fault 11 and Fault 4 affect the same process variable, the fault was expected to influence the reactor cooling water flow the most. Similarly to Fault 4, the CVA-based *RES* and the (D)PCA-based statistics gave superior results, in terms of correctly identifying the reactor cooling water flow as the variable responsible for this fault (see Table 10.3). The improper dominance of the observation variables x_{12}, x_{15}, x_{17}, x_{48}, x_{49}, and x_{52} was again responsible for the poor performance of the CVA-based *RES* (see Figure 10.19).

Some fault diagnosis techniques more easily diagnosed Fault 4 while others did better diagnosing Fault 11 (see Table 10.2). The lowest misclassification rates were provided by the MS T_1^2, DFDA/DPCA1 T^2, and CVA T_r^2 statistics, all of which take serial correlation into account. It is interesting that 'dynamic' versions of PCA which are designed to take serial correlation into account did

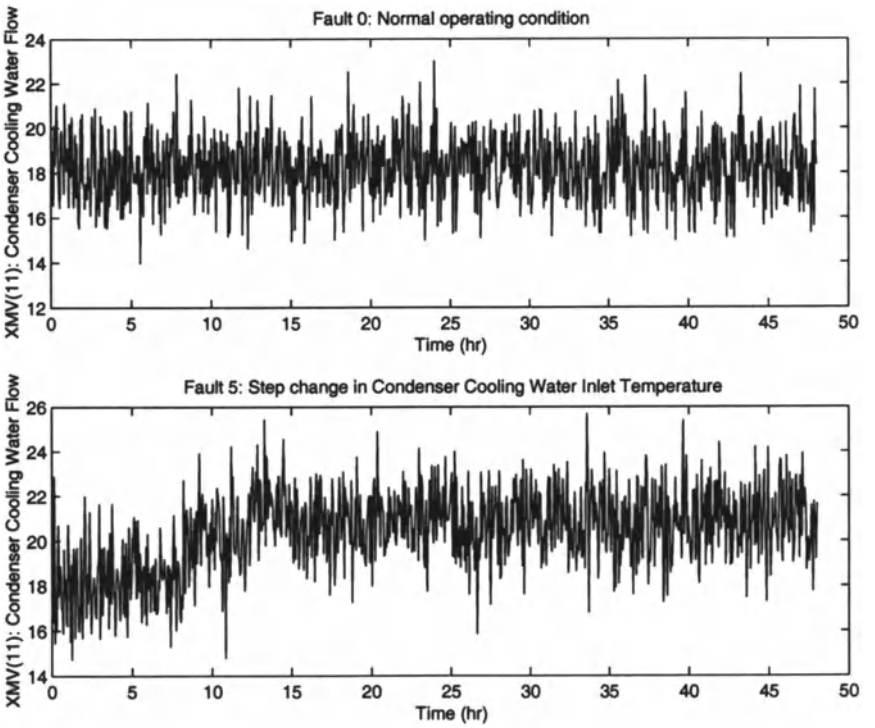

Fig. 10.11. Comparison of XMV(11) for Faults 0 and 5

not provide significantly improved fault diagnosis over their static versions for Fault 11.

10.6 Fault Detection

The objectives of a fault detection statistic are to be *robust* to data independent of the training set, *sensitive* to all the possible faults of the process, and *prompt* to the detection of the faults. The robustness of each statistic in Table 9.2 is determined by calculating the false alarm rate for the normal operating condition of the testing set and comparing it against the level of significance upon which the threshold is based. The sensitivity of the statistics is quantified by calculating the missed detection rates for Faults 1-21 of the testing set. The promptness of the statistics is based on the detection delays for Faults 1-21 of the testing set.

Prior to applying each of the statistics to the testing set, the parameter values associated with each statistic need to be specified. The orders deter-

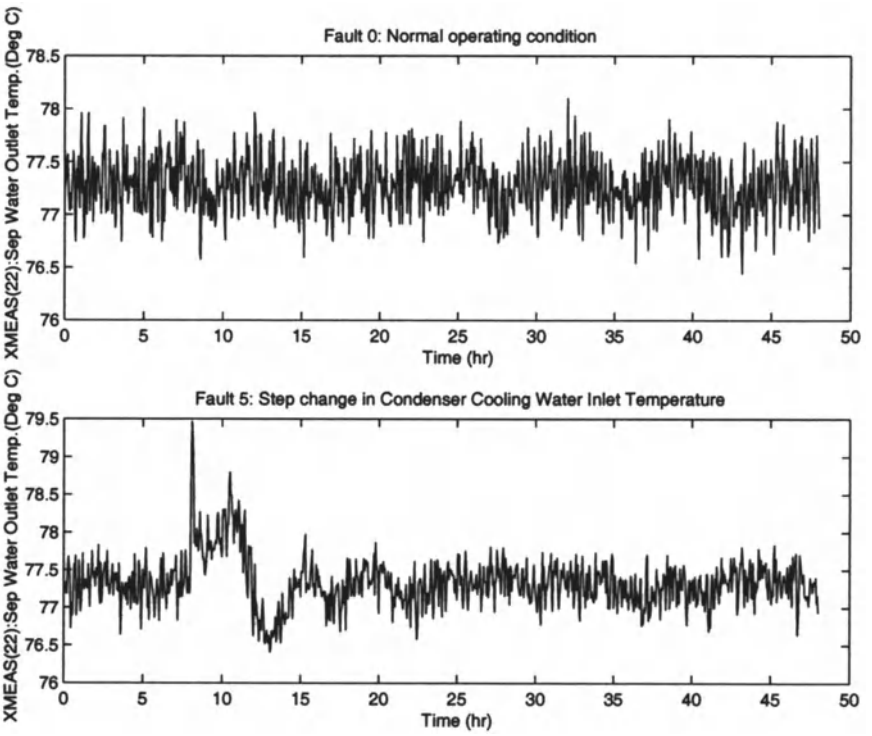

Fig. 10.12. Comparison of XMEAS(22) for Faults 0 and 5

mined for PCA, DPCA, PLS, and CVA and the number of lags h determined for DPCA and CVA are listed in Table 10.4. The orders and the number of lags were determined by applying the procedures described in Section 9.5 to the pretreated data for the normal operating condition of the training set.

The probability distributions used to determine the threshold for each statistic are listed in Table 9.2. Using a level of significance $\alpha = 0.01$, the false alarm rates of the training and testing sets were computed and tabulated in Table 10.5. The false alarm rates for the PCA and DPCA-based T^2 statistics are comparable in magnitude to $\alpha = 0.01$. The CVA-based statistics and the DPCA-based Q statistic resulted in relatively high false alarm rates for the testing set compared to the other multivariate statistics. The lack of robustness for T_s^2 and T_r^2 can be explained by the inversion of $\hat{\Sigma}_{pp}$ in (7.40). The high false alarm rate for the DPCA-based Q statistic may be due to a violation of the assumptions used to derive the threshold (4.22) (see Homework Problem 12 for a further exploration of this issue).

It would not be fair to directly compare the fault detection statistics in terms of missed detection rates when they have such widely varying false

Table 10.4. The lags and orders for the various models. The PLS models are all based on discriminant PLS.

Fault	ARX	PCAm	PCA1	DPCAm	PLS1	PLS2	$PLS1_{adj}$	$PLS2_{adj}$	FDA	FDA/ PCA1	FDA/ PCA2	DFDA/ DPCA1	CVA
	h	a	a	a	a	a	a	a	a	a	a	a	k
0	3	11	47	29	13	45	16	41	52	50	52	51	29
1	2	8	47	13	13	45	16	41	52	50	52	51	25
2	2	8	47	19	13	45	16	41	52	50	52	51	28
3	2	12	47	27	13	45	16	41	52	50	52	51	26
4	2	12	47	26	13	45	16	41	52	50	52	51	26
5	2	8	47	15	13	45	16	41	52	50	52	51	27
6	6	6	47	11	13	45	16	41	52	50	52	51	3
7	2	6	47	10	13	45	16	41	52	50	52	51	27
8	2	8	47	13	13	45	16	41	52	50	52	51	24
9	3	11	47	26	13	45	16	41	52	50	52	51	30
10	3	10	47	24	13	45	16	41	52	50	52	51	28
11	3	9	47	26	13	45	16	41	52	50	52	51	28
12	2	6	47	7	13	45	16	41	52	50	52	51	25
13	2	8	47	12	13	45	16	41	52	50	52	51	24
14	3	13	47	27	13	45	16	41	52	50	52	51	25
15	3	11	47	27	13	45	16	41	52	50	52	51	30
16	3	12	47	27	13	45	16	41	52	50	52	51	26
17	2	9	47	24	13	45	16	41	52	50	52	51	27
18	2	4	47	4	13	45	16	41	52	50	52	51	27
19	3	16	47	32	13	45	16	41	52	50	52	51	29
20	3	11	47	25	13	45	16	41	52	50	52	51	32
21	2	14	47	25	13	45	16	41	52	50	52	51	26

The ARX model is used to determine the lag orders for CVA, DPCA, and FDA as described in Section 7.5

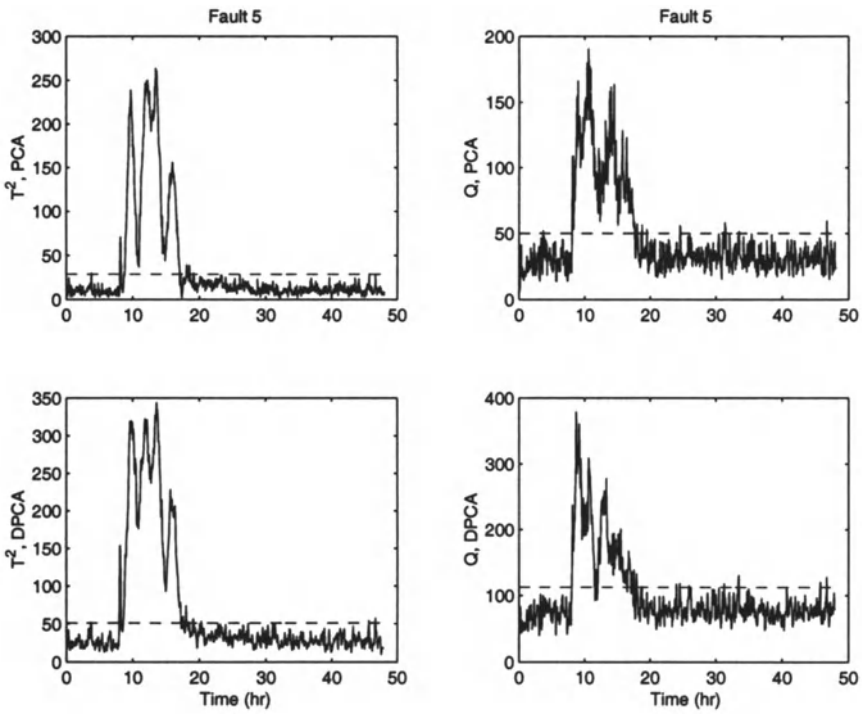

Fig. 10.13. The (D)PCA multivariate statistics for fault detection for Fault 5

Table 10.5. False alarm rates for the training and testing sets

Method	Measures	Training Set	Testing Set
PCA	T^2	0.002	0.014
PCA	Q	0.004	0.016
DPCA	T^2	0.002	0.006
DPCA	Q	0.004	0.281
CVA	T_s^2	0.027	0.083
CVA	T_r^2	0	0.126
CVA	Q	0.009	0.087

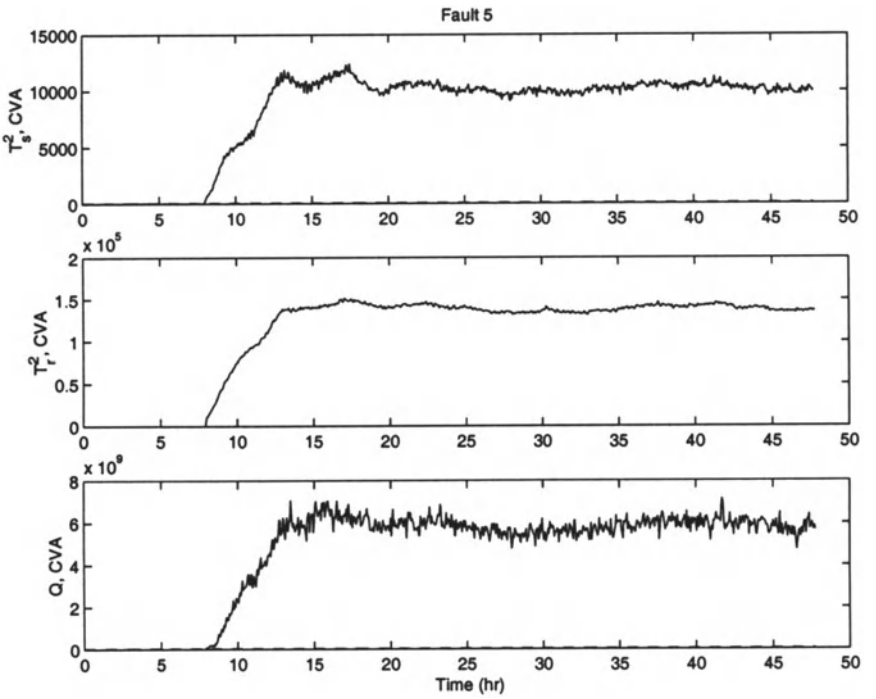

Fig. 10.14. The CVA multivariate statistics for fault detection for Fault 5

alarm rates. In computing the missed detection rates for Faults 1-21 of the testing set, the threshold for each statistic was adjusted to the tenth highest value for the normal operating condition of the *testing* set. The adjusted thresholds correspond to a level of significance $\alpha = 0.01$ by considering the probability distributions of the statistics for the normal operating condition. For statistics which showed low false alarm rates, the adjustment only shifted the thresholds slightly. For each statistic which showed a high false alarm rate, the adjustment increased the threshold by approximately 50%. Numerous simulation runs for the normal operating conditions confirmed that the adjusted thresholds indeed corresponded to a level of significance $\alpha = 0.01$. It was felt that this adjustment of thresholds provides a fairer basis for the comparison of the sensitivities of the statistics. For each statistic, the missed detection rates for all 21 faults were computed and tabulated in Table 10.6.

The missed detection rates for Faults 3, 9, and 15 are very high for all the fault detection statistics. No observable change in the mean or the variance can be detected by visually comparing the plots of each observation variable associated with Faults 3, 9, and 15 to the plots associated with the normal operating condition (Fault 0). It is conjectured that any statistic will result

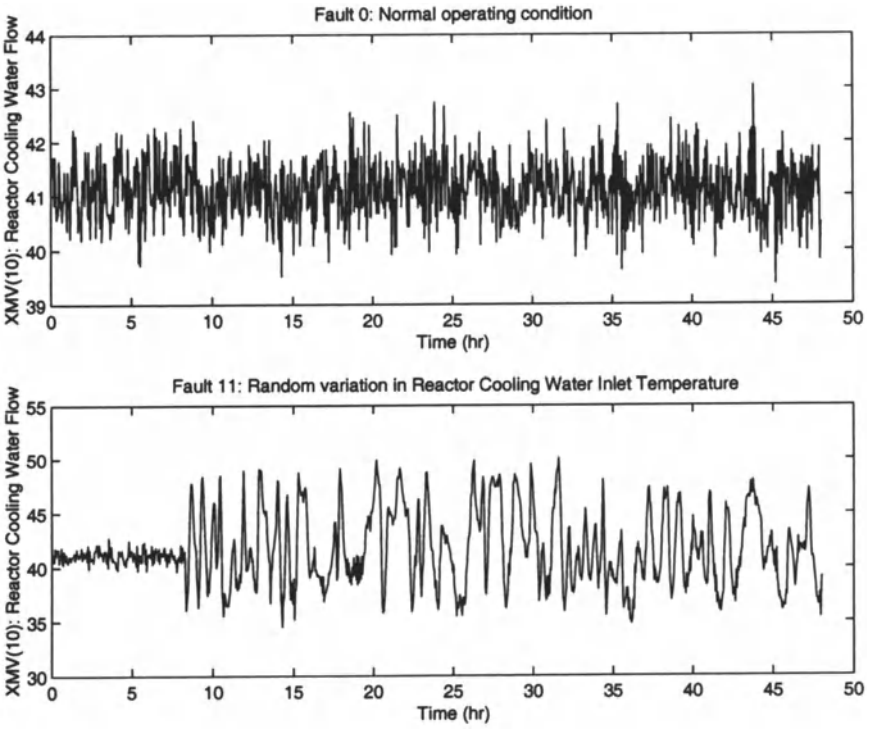

Fig. 10.15. Comparison of XMV(10) for Faults 0 and 11

in high missed detection rates for those faults, in other words, Faults 3, 9, and 15 are *unobservable* from the data. Including the missed detection rates for these faults would skew the comparison of the statistics, and therefore these faults are not analyzed when comparing the overall performance of the statistics.

The minimum missed detection rate achieved for each fault except Faults 3, 9, and 15 is contained in a box in Table 10.6. The T_r^2 statistic with the threshold rescaled as described above had the lowest missed detection rate except for the unobservable Faults 3 and 9. The conclusion that the T_r^2 statistic with a scaled threshold will *always* give lower missed detection rates than the other statistics would be *incorrect*, since another method may be better for a different amount of data or a different process. In particular, a fault that does not affect the states in the T_r^2 statistic will be invisible to this statistic. Since many of the statistics have comparable missed detection rates for many of the faults, it seems to have an advantage to incorporate the T_r^2 statistics with other statistics for fault detection.

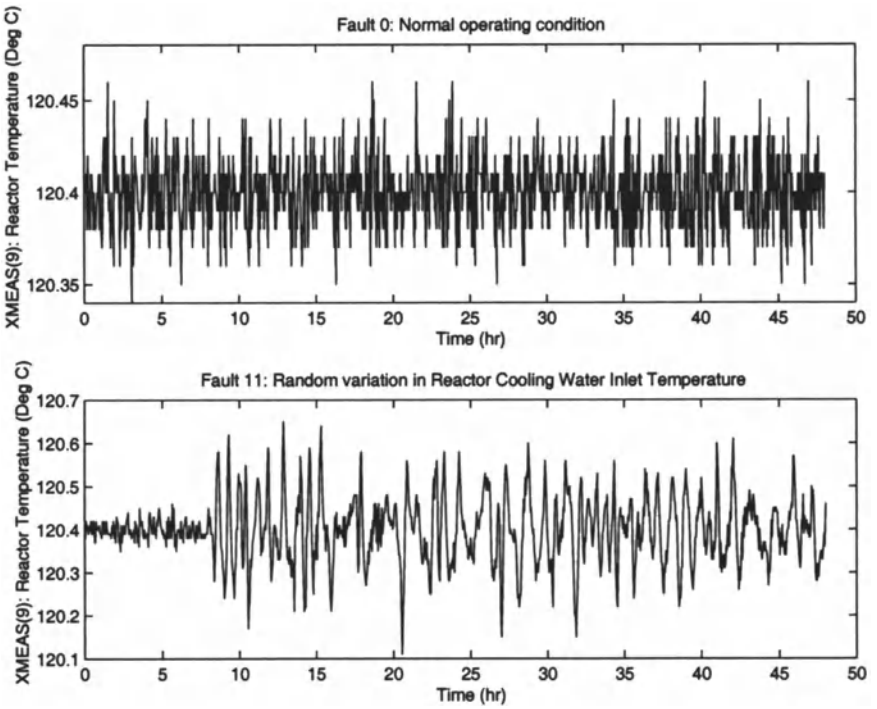

Fig. 10.16. Comparison of XMEAS(9) for Faults 0 and 11

The CVA-based Q statistic gave similar missed detection rates as the T_r^2 statistic for some faults, but performed more poorly for others. Other results, not shown here for brevity, showed that a slight shift in the lag order h or state order k can result in a large variation of the CVA-based Q statistic. Tweaking these parameters may improve the CVA-based Q statistic enough to give fault detection performance more similar to the T_r^2 statistic.

The number of minimums achieved with the residual-based statistics is far more than the number of minimums achieved with state or score-based statistics. Residual-based multivariate statistics tended to be more sensitive to the faults of the TEP than the state or score-based statistics. The better performance of residual-based statistics supports the claims in the literature, based on either theoretical analysis [230] or case studies [125], that residual-based statistics tend to be more sensitive to faults. A comparison of *all* the fault detection statistics revealed that the residual-based T_r^2 statistic was overall the most sensitive to the faults of the TEP. However, the T_r^2 statistic was found not to be very robust compared to most of the other statistics, due to the inversion of the matrix $\hat{\Sigma}_{pp}$ in (7.40). Also, recall that the threshold

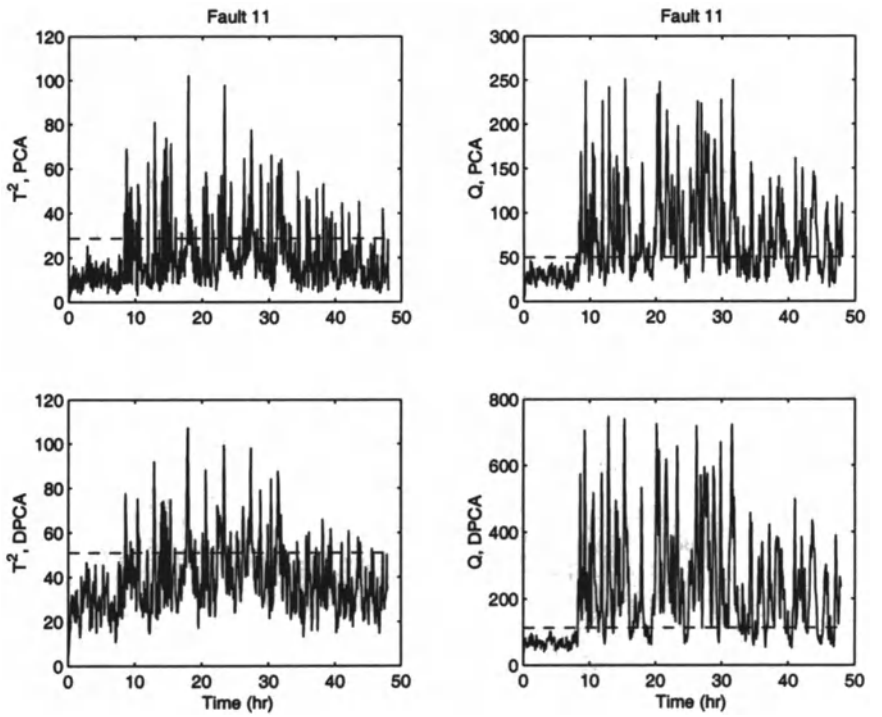

Fig. 10.17. The (D)PCA multivariate statistics for fault detection for Fault 11

used here was rescaled based on the testing set to give a false alarm rate of 0.01, as described in Section 10.6. The behavior of the T_r^2 statistic with the threshold (7.44) can give large false alarm rates, as was discussed earlier.

On average, the DPCA-based statistics were somewhat more sensitive to the faults than the PCA-based statistics, although the overall difference was not very large. The high false alarm rates found for the DPCA-based Q statistic (see Table 10.5) indicate that the threshold (4.22) may need to be rescaled based on an additional set of data as was done here.

Most statistics performed well for the faults that affect a significant number of observation variables (Faults 1, 2, 6, 7, 8, 14, and 18). In these cases, most variables deviated significantly from their distribution in the normal operating conditions. The other faults had a limited number of the observation variables deviate from their distribution in the normal operating conditions. Detecting such faults is relatively more challenging.

Since false alarms are inevitable, it is often difficult to determine whether the out-of-control value of a statistic is the result of a fault or of a false alarm. In order to decrease the rate of false alarms, it is common to show an

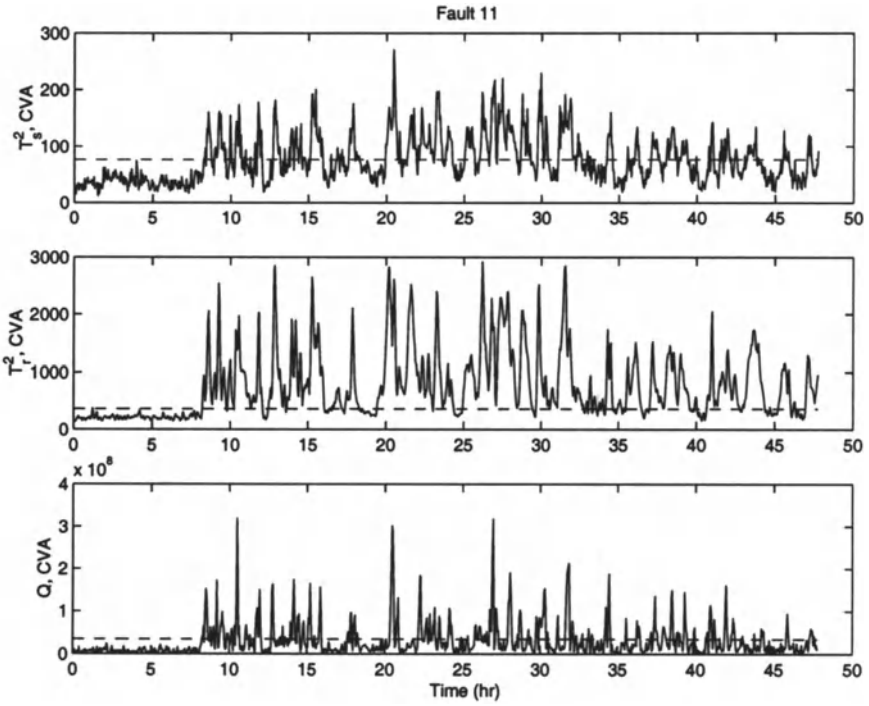

Fig. 10.18. The CVA multivariate statistics for fault detection for Fault 11

alarm only when several consecutive values of a statistic have exceeded the threshold. In computing the detection delays for the statistics in Table 10.7, a fault is indicated only when six consecutive measure values have exceeded the threshold, and the detection delay is recorded as the first time instant in which the threshold was exceeded. Assuming independent observations and $\alpha = 0.01$, this corresponds to a false alarm rate of $0.01^6 = 1 \times 10^{-12}$. The detection delays for all 21 faults listed in Table 10.7 were obtained by applying the same thresholds as used to determine the missed detection rates.

For the multivariate statistics, a close examination of Tables 10.6 and 10.7 reveals that the statistics exhibiting small detection delays tend to exhibit small missed detection rates and *vice versa*. Since the detection delay results correlate well with the missed detection rate results, all of the conclusions for missed detection rates apply here.

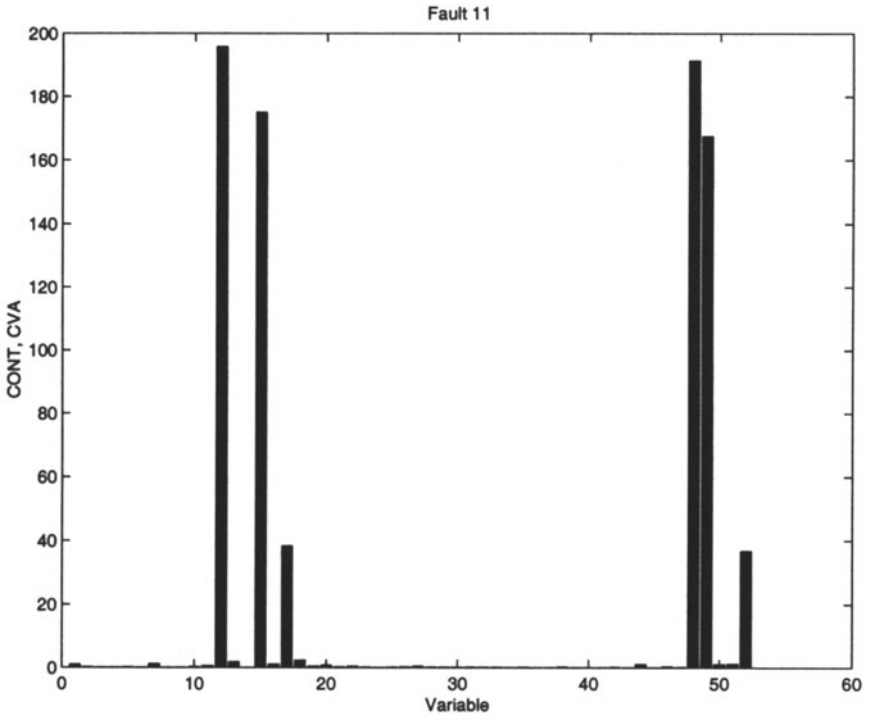

Fig. 10.19. The average contribution plot for Fault 11 for the CVA-based *CONT*

10.7 Fault Identification

The objective of a fault identification statistic is to identify the observation variable(s) most closely related to the fault. The challenge in developing a good criterion for comparing the different statistics is choosing which observation variable(s) is most relevant to diagnosing the fault. This, of course, depends on the knowledge and expertise of the plant operators and engineers. The only faults investigated here for fault identification are those in which a direct and clear link between the fault and an observation variable could be determined. The faults investigated in this section for fault identification and the observation variables directly related to each fault are listed in Table 10.8. The ranking of these observation variables for each fault is the criterion used to compare the different statistics listed in Table 9.3.

The statistics investigated in this section are listed in Table 9.3, and the parameter values associated with the statistics are listed in Table 10.4. The rankings of the observation variables listed in Table 10.8 for each statistic and fault are contained in Tables 10.9, 10.10, and 10.11. These tables list the

Table 10.6. Missed detection rates for the testing set

Fault	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q
1	**0.008**	**0.003**	**0.006**	**0.005**	**0.001**	**0**	**0.003**
2	0.020	0.014	0.019	0.015	0.011	0.010	0.026
3	0.998	0.991	0.991	0.990	0.981	0.986	0.985
4	**0.956**	**0.038**	**0.939**	**0**	**0.688**	**0**	**0.975**
5	**0.775**	**0.746**	**0.758**	**0.748**	**0**	**0**	**0**
6	0.011	0	0.013	0	0	0	0
7	0.085	0	0.159	0	0.386	0	0.486
8	0.034	0.024	0.028	0.025	0.021	0.016	0.486
9	0.994	0.981	0.995	0.994	0.986	0.993	0.993
10	0.666	0.659	0.580	0.665	0.166	0.099	0.599
11	**0.794**	**0.356**	**0.801**	**0.193**	**0.515**	**0.195**	**0.669**
12	0.029	0.025	0.010	0.024	0	0	0.021
13	0.060	0.045	0.049	0.049	0.047	0.040	0.055
14	0.158	0	0.061	0	0	0	0.122
15	0.988	0.973	0.964	0.976	0.928	0.903	0.979
16	0.834	0.755	0.783	0.708	0.166	0.084	0.429
17	0.259	0.108	0.240	0.053	0.104	0.024	0.138
18	0.113	0.101	0.111	0.100	0.094	0.092	0.102
19	0.996	0.873	0.993	0.735	0.849	0.019	0.923
20	0.701	0.550	0.644	0.490	0.248	0.087	0.354
21	0.736	0.570	0.644	0.558	0.440	0.342	0.547

rankings for the average statistic values over the time periods 0-5 hours, 5-24 hours, and 24-40 hours, after the fault occurred. A ranking of 1 in the tables indicates that the observation variable listed in Table 10.8 had the largest average statistic value, and a ranking of 52 indicates that the observation variable listed in Table 10.8 had the smallest average statistic value. The best ranking for each fault is contained in a box. The results are divided into three tables because it is useful to analyze how the proficiencies of the statistics change with time. It is best to properly identify the fault as soon as it occurs, and therefore the results during the time period 0-5 hours after the fault are tabulated separately. The results for the time period between 5-24 and 24-40 hours after the fault occurred were tabulated separately, because this is useful in determining the robustness of the statistics.

As shown in Tables 10.9-10.11, the (D)PCA-based *CONT* performed well. The better performance of the (D)PCA-based *CONT* to the (D)PCA-based *RES* suggests that the abstraction of structure provided by PCA was even more critical to fault identification than fault detection. For the faults where fault propagation occurred, the performance of the data-driven statistics de-

Table 10.7. Detection delays (minutes) for the testing set

Fault	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q
1	**21**	**9**	**18**	**15**	6	**9**	6
2	51	36	48	39	39	45	75
3	–	–	–	–	–	–	–
4	–	**9**	**453**	3	**1386**	3	–
5	**48**	**3**	**6**	**6**	**3**	**3**	0
6	30	3	33	3	3	3	0
7	3	3	3	3	3	3	0
8	69	60	69	63	60	60	63
9	–	–	–	–	–	–	–
10	288	147	303	150	75	69	132
11	**912**	**33**	**585**	21	**876**	**33**	**81**
12	66	24	9	24	6	6	0
13	147	111	135	120	126	117	129
14	12	3	18	3	6	3	3
15	–	2220	–	–	2031	–	–
16	936	591	597	588	42	27	33
17	87	75	84	72	81	60	69
18	279	252	279	252	249	237	252
19	–	–	–	246	–	33	–
20	261	261	267	252	246	198	216
21	1689	855	1566	858	819	1533	906

Table 10.8. The variables assumed to be most closely related to each disturbance

Fault	Process Variable	Data Variable	Variable Description
2	XMV(6)	x_{47}	Purge Valve (Stream 9)
4	XMV(10)	x_{51}	Reactor Cooling Water Flow
5	XMEAS(22)	x_{22}	Sep. Cooling Water Outlet Temp
6	XMV(3)	x_{44}	A Feed Flow (Stream 1)
11	XMV(10)	x_{51}	Reactor Cooling Water Flow
12	XMEAS(22)	x_{22}	Sep. Cooling Water Outlet Temp
14	XMV(10)	x_{51}	Reactor Cooling Water Flow
21	XMV(4)	x_{45}	A, B, and C Feed Flow (Stream 4)

Table 10.9. The rankings for the time period 0-5 hours after the fault occurred

Fault	PCA *CONT*	PCA *RES*	DPCA *CONT*	DPCA *RES*	CVA *CONT*	CVA *RES*
2	[2]	4	[2]	5	10	[2]
4	[1]	[1]	[1]	[1]	**10**	[1]
5	**12**	**21**	**11**	[8]	**15**	**17**
6	[1]	6	3	2	6	6
11	[1]	[1]	[1]	[1]	**10**	[1]
12	[1]	6	[1]	3	10	14
14	2	2	[1]	2	11	[1]
21	52	40	48	48	52	52

Table 10.10. The rankings for the time period 5-24 hours after the fault occurred

Fault	PCA *CONT*	PCA *RES*	DPCA *CONT*	DPCA *RES*	CVA *CONT*	CVA *RES*
2	[2]	5	[2]	7	10	3
4	[1]	[1]	[1]	[1]	**12**	[1]
5	**31**	**34**	**30**	**31**	[18]	**14**
6	5	52	8	45	8	[3]
11	[1]	[1]	[1]	[1]	**13**	[1]
12	[1]	12	[1]	3	13	24
14	2	2	[1]	2	10	[1]
21	52	46	51	51	52	52

Table 10.11. The rankings for the time period 24-40 hours after the fault occurred

Fault	PCA *CONT*	PCA *RES*	DPCA *CONT*	DPCA *RES*	CVA *CONT*	CVA *RES*
2	[2]	5	3	12	10	4
4	[1]	[1]	[1]	[1]	**11**	[1]
5	[9]	**35**	**14**	**30**	**16**	**16**
6	7	51	11	45	[1]	3
11	[1]	[1]	[1]	[1]	**13**	[1]
12	10	21	[4]	36	17	26
14	2	2	[1]	2	11	[1]
21	52	48	52	52	52	50

teriorated as the effect of the fault evolved. Robustness may be achieved by applying model-based fault identification statistics that are able to take into account the propagation of the fault (see Chapter 11).

All fault identification statistics performed poorly for Fault 21 (see Tables 10.9-10.11). The A/B/C feed flow valve for Stream 4 was fixed at the steady state position (see Figure 8.1). The valve was stuck, indicating that the signals from this valve were constant, which corresponds to zero variance. The *RES* and *CONT*-based statistics had great difficulty identifying the A/B/C feed flow as the variable associated with the fault because these statistics are designed to detect *positive shift in variance* only. This illustrates the importance in such cases of implementing statistics such as Equation 4.29 which can detect a *negative shift in variance.* This type of statistic implemented in the appropriate manner would have detected Fault 21 rather easily. In general it is suggested that such a statistic should be applied to each process variable, with the α level set to keep the false alarm rate low.

The performance of a fault identification statistic can significantly deteriorate over time for faults whose effects on the process variables change over time. For instance, the effect of Fault 12 propagates over the interval 5 to 40 hours after the fault occurred. As a result, there is only one statistic producing a ranking below 10 in Table 10.11 while all but one statistic produced a ranking at or above 10 in Table 10.9. For Fault 6, the performance of the (D)PCA-based fault identification statistics substantially degraded over time, while the performance of the CVA-based statistics actually improved.

10.8 Fault Diagnosis

Assuming that process data collected during a fault are represented by a previous fault class, the objective of the fault diagnosis statistics in Table 9.4 is to classify the data to the *correct* fault class. That is, a highly proficient fault diagnosis statistic produces small misclassification rates when applied to data independent of the training set. Such a statistic usually has an accurate representation of each class, more importantly such a statistic separates each class from the others very well. Recall that all the methods listed in Table 9.4 are based on supervised classification. For the discriminant PLS, PCA1, MS, and FDA methods, one model is built for all fault classes. For the other methods listed in Table 9.4, a separate model is built for each fault class. The proficiencies of the statistics in Table 9.4 are investigated in this section based on the misclassification rates for Faults 1-21 of the testing set. The parameters for each statistic were determined from Faults 1-21 of the training set. The lags and orders associated with the statistics are listed in Table 10.4.

The overall misclassification rate for each statistic when applied to Faults 1-21 of the testing set is listed in Table 10.12. For each statistic, the misclassification rates for all 21 faults were computed and tabulated in Tables

10.13-10.20. The minimum misclassification rate achieved for each fault except Faults 3, 9, and 15 is contained in a box.

Table 10.12. The overall misclassification rates

Method	Basis	Misclassification Rate
PCAm	T^2	0.742
PCA1	T^2	0.212
PCAm	Q	0.609
PCAm	T^2 & Q	0.667
DPCAm	T^2	0.724
DPCAm	Q	0.583
DPCAm	T^2 & Q	0.662
PLS1	–	0.565
PLS2	–	0.567
PLS1_{adj}	–	0.576
PLS2_{adj}	–	0.574
CVA	T_s^2	0.501
CVA	T_r^2	0.213
CVA	Q	0.621
FDA	T^2	0.195
FDA/PCA1	T^2	0.206
FDA/PCA2	T^2	0.195
DFDA/DPCA1	T^2	0.192
MS	T_0^2	0.214
MS	T_1^2	0.208

When applying the fault diagnosis statistics, it was assumed that the *a priori* probability for each class i was equal to $P(\omega_i) = 1/p$ where $p = 21$ is the number of fault classes. DFDA/DPCA1 produced the lowest overall misclassification rate (0.192), followed by the rest of the FDA-based methods, as shown in Table 10.12. The CVA-based T_r^2, PCA1, and MS statistics produced comparable overall misclassification rates.

To compare the FDA/PCA1 and FDA/PCA2 methods for diagnosing faults, the overall misclassification rates for the training and testing sets and the information criterion (5.12) are plotted for various orders using FDA, FDA/PCA1, and FDA/PCA2 (see Figures 10.20, 10.21, and 10.22), respectively. The overall misclassification rates for the testing set using FDA/PCA1 and FDA/PCA2 was lower than that of the FDA for most orders $a \geq p$. The performance of FDA/PCA1 and FDA/PCA2 was very similar, indicating that

Table 10.13. The misclassification rates for 0-40 hours after the Faults 1-21 occurred

Fault	PCAm T^2	PCA1 T^2	PCAm Q	PCAm T^2&Q	DPCAm T^2	DPCAm Q	DPCAm T^2&Q	CVA T_s^2	CVA T_r^2	CVA Q
1	**0.680**	**0.024**	**0.028**	**0.041**	**0.880**	**0.035**	**0.038**	**0.028**	**0.026**	**0.245**
2	0.410	0.018	0.024	0.035	0.441	0.060	0.034	0.010	0.090	0.155
3	0.939	0.783	0.991	1.000	0.701	0.995	1.000	0.940	0.821	0.978
4	**0.810**	**0.163**	**0.951**	**1.000**	**0.720**	**0.964**	**1.000**	**0.981**	**0.358**	**0.890**
5	**0.956**	**0.021**	**0.913**	**0.973**	**0.874**	**0.856**	**1.000**	**0.061**	**0.040**	**0.174**
6	0.100	0	0.050	0.076	0.049	0.063	0.089	0.001	0.001	0.014
7	0.978	0	0.405	0.496	0.868	0.336	0.633	0.638	0.001	0.578
8	0.998	0.030	0.270	0.409	1.000	0.170	0.398	0.518	0.055	0.670
9	0.993	0.779	0.995	1.000	0.988	0.998	1.000	0.969	0.848	0.969
10	0.849	0.126	0.988	1.000	0.743	0.995	1.000	0.745	0.098	0.816
11	**0.989**	**0.234**	**0.859**	**0.968**	**0.948**	**0.843**	**0.983**	**0.904**	**0.139**	**0.901**
12	0.850	0.021	0.216	0.204	0.700	0.203	0.215	0.009	0.020	0.294
13	1.000	0.235	0.501	0.754	1.000	0.441	0.721	0.495	0.328	0.591
14	0.244	0.036	0.273	0.438	0.564	0.110	0.153	0.203	0.001	0.450
15	0.963	0.768	0.994	1.000	0.964	0.996	1.000	0.964	0.666	0.984
16	0.841	0.200	0.984	1.000	0.801	0.989	1.000	0.568	0.145	0.859
17	0.563	0.193	0.415	0.413	0.648	0.320	0.403	0.218	0.638	0.217
18	0.360	0.410	0.393	0.324	0.294	0.395	0.298	0.540	0.134	0.829
19	0.401	0.124	0.651	0.876	0.789	0.564	0.956	0.470	0.005	0.929
20	0.761	0.143	0.916	1.000	0.708	0.948	1.000	0.306	0.090	0.588
21	0.899	0.138	0.979	1.000	0.529	0.953	1.000	0.948	0.611	0.924
overall	0.742	0.212	0.609	0.667	0.724	0.583	0.662	0.501	0.213	0.621

Table 10.14. The misclassification rates for 0-40 hours after the Faults 1-21 occurred

Fault	PLS1 –	PLS2 –	$PLS1_{adj}$ –	$PLS2_{adj}$ –	FDA T^2	FDA/PCA1 T^2	FDA/PCA2 T^2	DFDA/DPCA1 T^2	MS T_0^2	MS T_1^2
1	**0.013**	**0.013**	**0.019**	**0.019**	**0.025**	**0.024**	**0.025**	**0.026**	**0.025**	**0.035**
2	0.014	0.024	0.024	0.024	0.019	0.019	0.019	0.019	0.019	0.033
3	0.961	0.970	0.869	0.876	0.780	0.734	0.780	0.735	0.780	0.886
4	**0.170**	**0.119**	**0.364**	**0.320**	**0.176**	**0.163**	**0.176**	**0.159**	**0.176**	**0.427**
5	**0.006**	**0.008**	**0.044**	**0.043**	**0.020**	**0.020**	**0.020**	**0.023**	**0.020**	**0.040**
6	0.435	0.778	0.834	0.831	0	0	0	0	0	0
7	0	0	0	0.001	0	0	0	0	0	0
8	0.851	0.789	0.848	0.850	0.003	0.004	0.003	0.026	0.030	0.019
9	0.981	0.981	0.899	0.915	0.773	0.780	0.773	0.801	0.773	0.872
10	0.661	0.591	0.586	0.569	0.131	0.158	0.131	0.101	0.131	0.098
11	**0.989**	**0.979**	**0.859**	**0.886**	**0.245**	**0.244**	**0.245**	**0.118**	**0.245**	**0.121**
12	0.988	0.953	0.869	0.866	0.018	0.016	0.018	0.030	0.018	0.005
13	0.646	0.625	0.751	0.738	0.239	0.246	0.239	0.229	0.239	0.208
14	0.995	0.998	0.931	0.930	0.013	0.013	0.013	0.004	0.013	0.001
15	0.988	0.981	0.926	0.925	0.764	0.780	0.764	0.784	0.764	0.725
16	0.894	0.660	0.658	0.558	0.193	0.184	0.193	0.218	0.193	0.255
17	0.146	0.164	0.388	0.378	0.150	0.145	0.150	0.043	0.150	0.038
18	0.775	0.843	0.839	0.796	0.315	0.399	0.315	0.154	0.750	0.431
19	0.913	0.945	0.800	0.778	0.039	0.055	0.039	0.142	0.039	0.003
20	0.334	0.274	0.509	0.525	0.126	0.125	0.126	0.176	0.126	0.158
21	0.098	0.096	0.068	0.066	0.044	0.198	0.030	0.261	0.004	0.003
overall	0.565	0.568	0.576	0.574	0.195	0.206	0.195	0.192	0.214	0.208

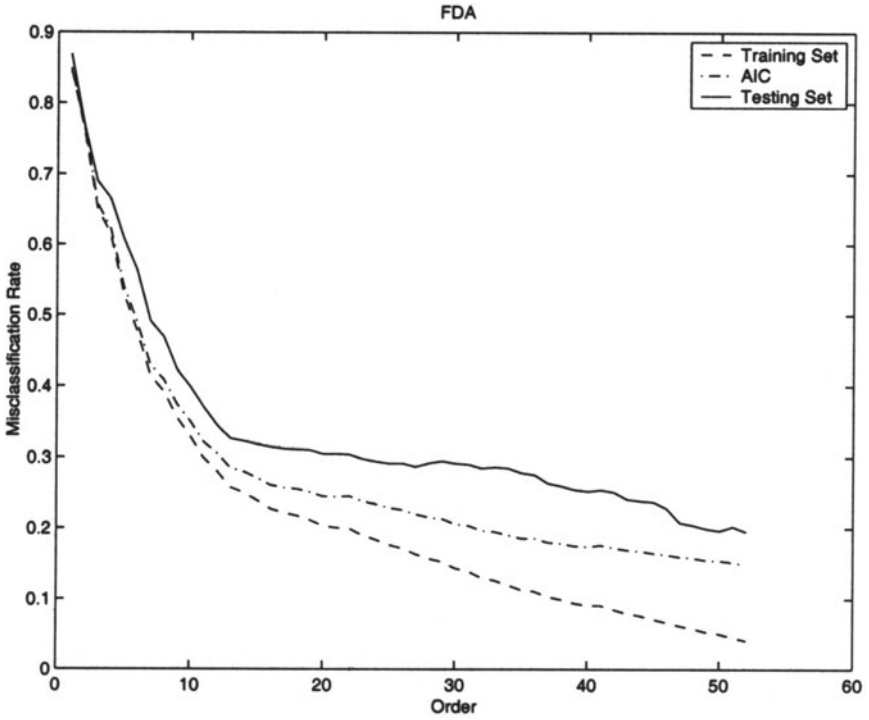

Fig. 10.20. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using FDA

using PCA1 to rank the $m - p + 1$ eigenvectors corresponding to the zero eigenvalues in FDA is a reasonable approach. A close comparison of Figures 10.21 and 10.22 indicates that for $20 \leq a \leq 48$, the overall misclassification rate for the testing set using FDA/PCA1 is lower than FDA/PCA2. Because of this advantage of using FDA/PCA1 over FDA for this problem, lag variables will be included only on the data for FDA/PCA1 when investigating the proficiency of the methods for removing serial correlations of the data.

To evaluate the potential advantage of including lagged variables in FDA/PCA1 to capture correlations, the overall misclassification rates for the training and testing sets and the information criterion (5.12) are plotted for various orders using FDA/PCA1 and DFDA/DPCA1 (see Figures 10.21 and 10.23), respectively. FDA/PCA1 and DFDA/DPCA1 select excellent vectors for projecting to a lower dimensional space for small a. Figures 10.21 and 10.23 show that most of the separation between the fault classes occurs in the space provided by the first 13 generalized eigenvectors. The misclassification rate with $a = 13$ for FDA/PCA1 is 0.33 and DFDA/DPCA1 is 0.34.

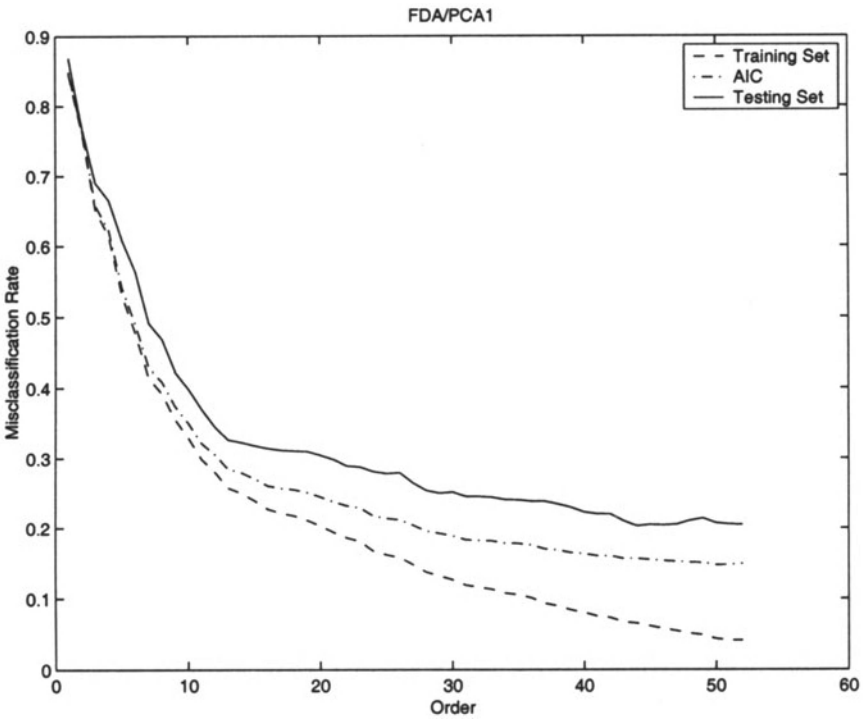

Fig. 10.21. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using FDA/PCA1

The FDA/PCA and DFDA/DPCA1-based statistics were able to separate the fault classes well for the space spanned by the first $p-1$ generalized eigenvectors. The proficiency was slightly increased as the dimensionality was increased further for FDA/PCA1 and DFDA/DPCA1. DFDA/DPCA1 produced the lowest overall misclassification rate among all of the fault diagnosis methods investigated in this chapter. Including lagged variables in FDA/PCA1 can give better fault diagnosis performance. The advantage becomes especially clear when DFDA/DPCA1 is applied to a system with a short sampling time (see Homework Problem 11).

The information criterion performed relatively well, as the slope of the misclassification rate of the testing set is fairly equivalent to the slope of the information criterion for $a = 15$ to 50 in Figures 10.20-10.23. The AIC captures the shape and slope of the misclassification rate curve for the testing data. The AIC weighs the prediction error term and the model complexity term fairly. If one desires to have a lower dimensional FDA model for diag-

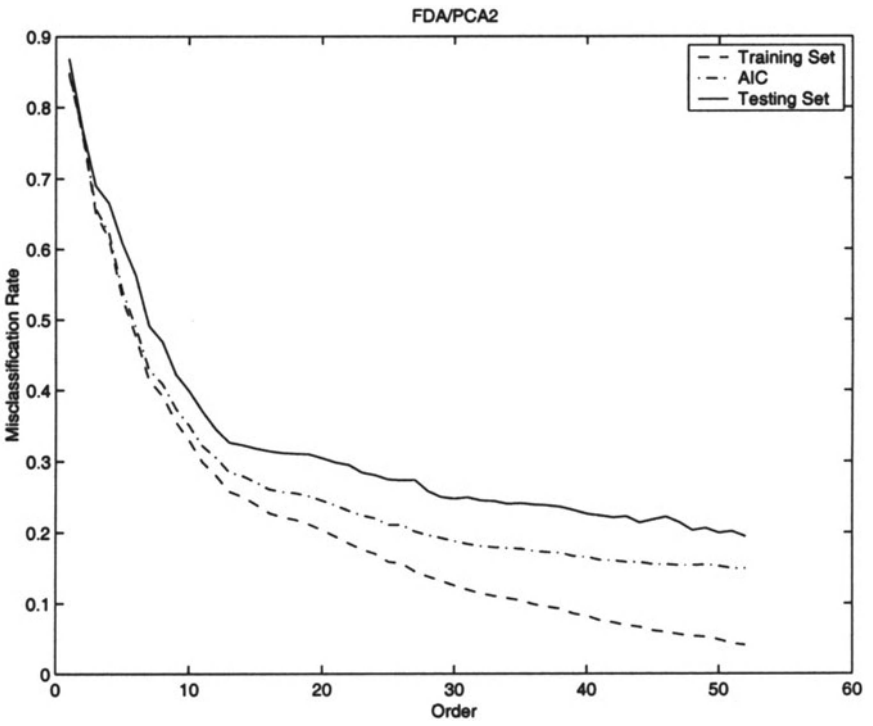

Fig. 10.22. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using FDA/PCA2

nosing faults, the model complexity term can be weighed more heavily (see Homework Problem 5).

Figure 10.24 plots the overall misclassification rates for the training and testing sets and the information criterion (5.12) for various orders using PLS1 and PLS2. The reduction order c is the point at which the information criterion is minimized. The reduction order for each class in PLS1 is $c_1 = 13$ and the reduction order for PLS2 $c_2 = 45$. In general, the overall misclassification rate of PLS1 is lower than that of PLS2 for a fixed order, especially when $a < c_1$. Also, the performance of PLS1 is less sensitive to order selection than PLS2. The misclassification rate on average is the same for the best reduction orders for PLS1 and PLS2, as shown in Table 10.12.

Figure 10.25 plots the overall misclassification rates for the training and testing sets and the information criterion (5.12) for various orders using PLS1_{adj} and PLS2_{adj}. Figures 10.24 and 10.25 show similar trends. Regardless of order selected, PLS1_{adj} performs better than PLS2_{adj} in terms of lower overall misclassification rates. The reduction orders that minimize the AIC

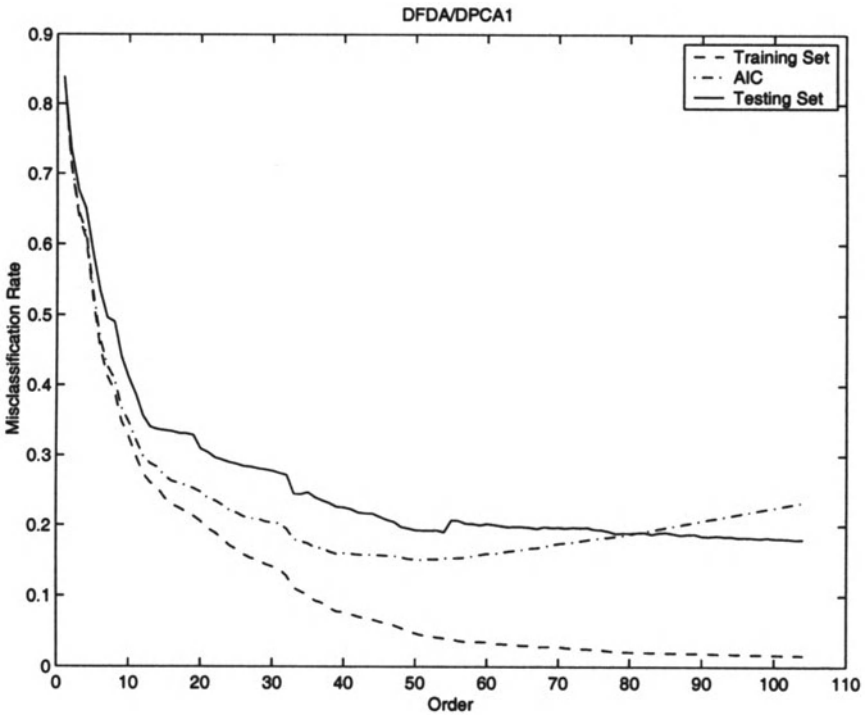

Fig. 10.23. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using DFDA/DPCA1

(5.12) for PLS1_{adj} and PLS2_{adj} are 16 and 41 respectively, which are close to the orders for PLS1 and PLS2 (c_1 and c_2), respectively. In terms of overall misclassification rates, PLS1_{adj} and PLS2_{adj} have similar performance to PLS1 and PLS2, respectively. For a fixed model order, the PLS1 methods almost always gave better fault diagnosis than the PLS2 methods. The performance of the PLS1 methods was also less sensitive to order selection than the PLS2 methods, and with the AIC resulting in lower model orders (see Table 10.4).

The information criterion worked fairly well for all discriminant PLS methods. The overall misclassification rate for the testing set with the reduction order using the information criterion for PLS1_{adj} is 0.58 while that for the other three PLS methods is 0.57. The minimum overall misclassification rate for the testing set is 0.56 for PLS1_{adj} and PLS2_{adj} and 0.55 for PLS1 and PLS2. The AIC curves (see Figures 10.24 and 10.25) nearly overlap the misclassification rate curves for PLS2 and adjusted PLS2, which indicates that the AIC will give similar model orders as cross-validation in these cases.

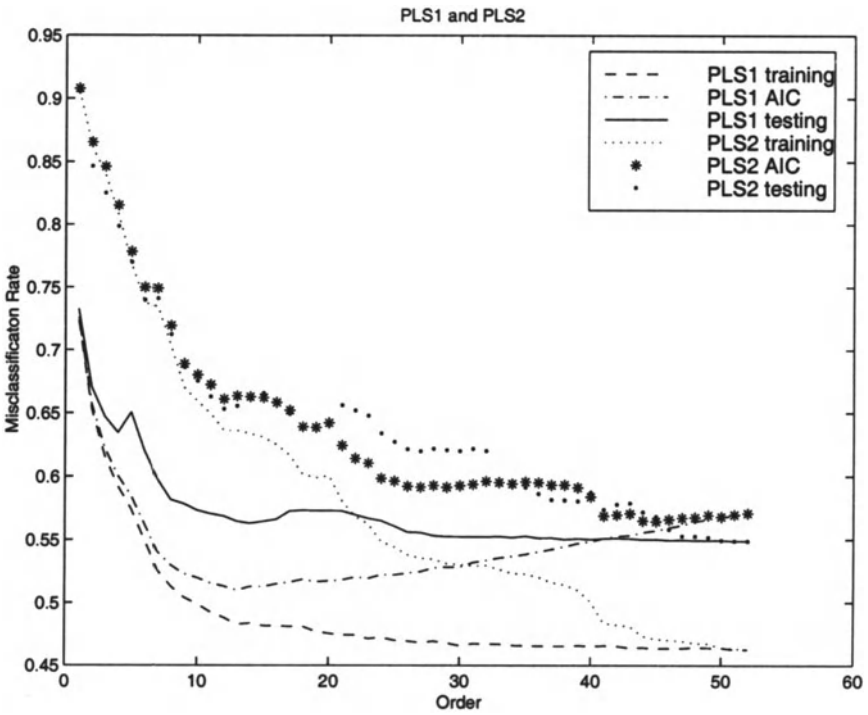

Fig. 10.24. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using PLS1 and PLS2

For PLS1 and adjusted PLS1, the AIC does not overlap with the classification rate curves, but does have a minimum at approximately the same order as where the misclassification rate curves for the testing data flatten out. This indicates that the AIC provided good model orders for the PLS1 methods.

Figure 10.26 plots the overall standard deviation of misclassification rates for the training and testing sets for various orders using PLS1, PLS2, $PLS1_{adj}$, and $PLS2_{adj}$. The standard deviations for $PLS1_{adj}$ and $PLS2_{adj}$ were 10-25% lower than that of PLS1 and PLS2 (respectively) for most orders. This indicates that $PLS1_{adj}$ and $PLS2_{adj}$ provided a more consistent prediction quality than PLS1 and PLS2. For example, 7 of 21 classes had misclassification rates between 0.90 to 1.00 using PLS1 and PLS2, respectively (see Table 10.14). However, only 2 of 21 classes were between 0.90 and 1.00 using $PLS1_{adj}$ and $PLS2_{adj}$ and the highest misclassification rate was 0.93. This also means that when PLS1 and PLS2 produced low misclassification rates, $PLS1_{adj}$ and $PLS2_{adj}$ tended to produce higher misclassification rates. There was an

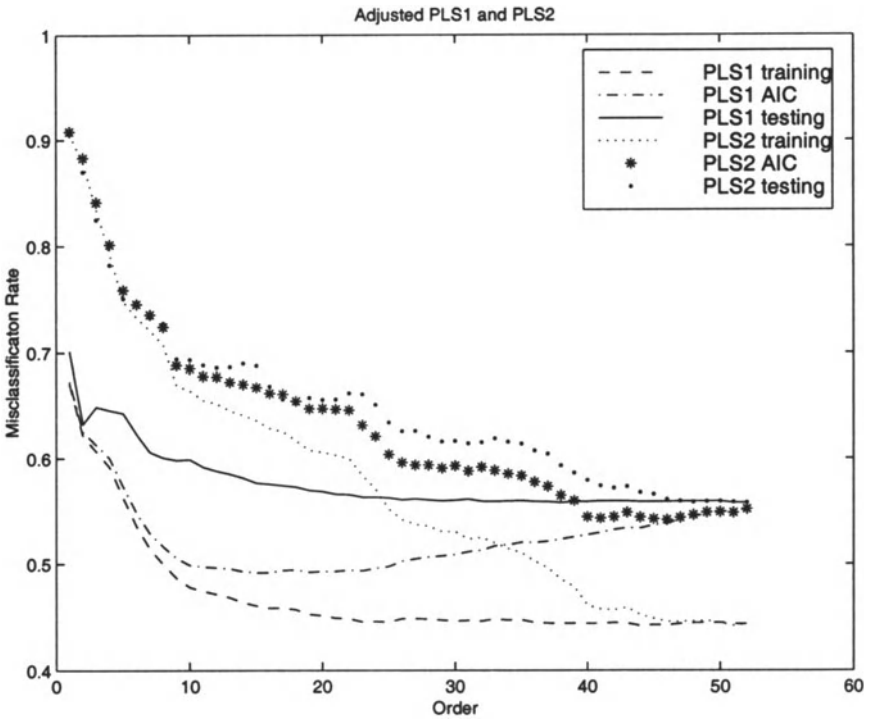

Fig. 10.25. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using PLS1_{adj} and PLS2_{adj}

advantage to apply PLS1_{adj} and PLS2_{adj} when PLS1 and PLS2 performed poorly.

Although PLS1 was able to capture a large amount of variance using only a few factors, it does require more computation time. Recall that in the calibration steps, PLS1 needs to run the NIPALS p times whereas PLS2 only needs to run the NIPALS one time, and that NIPALS runs from (6.10) to (6.20) for each PLS component. Since iteration from (6.10) to (6.13) is needed for PLS2, NIPALS requires a longer computation time in PLS2. Assume that it takes t_1 computation time to run from (6.22) to (6.27) for PLS1, and that it takes PLS2 $t_1 + \epsilon$ computation time. The total computation time t_{train} in the calibration steps is equal to pat_1 and $a(t_1 + \epsilon)$ for PLS1 and PLS2, respectively, where $a = \min(m, n)$. In the prediction steps, assume it takes t_2 computation time unit to run from (6.30) to (6.32), and that the total computation time t_{test} in the prediction step is equal to pc_1t_2 and c_2t_2 for PLS1 and PLS2, respectively. The ratio r_t of the total computation time between PLS1 and PLS2 is

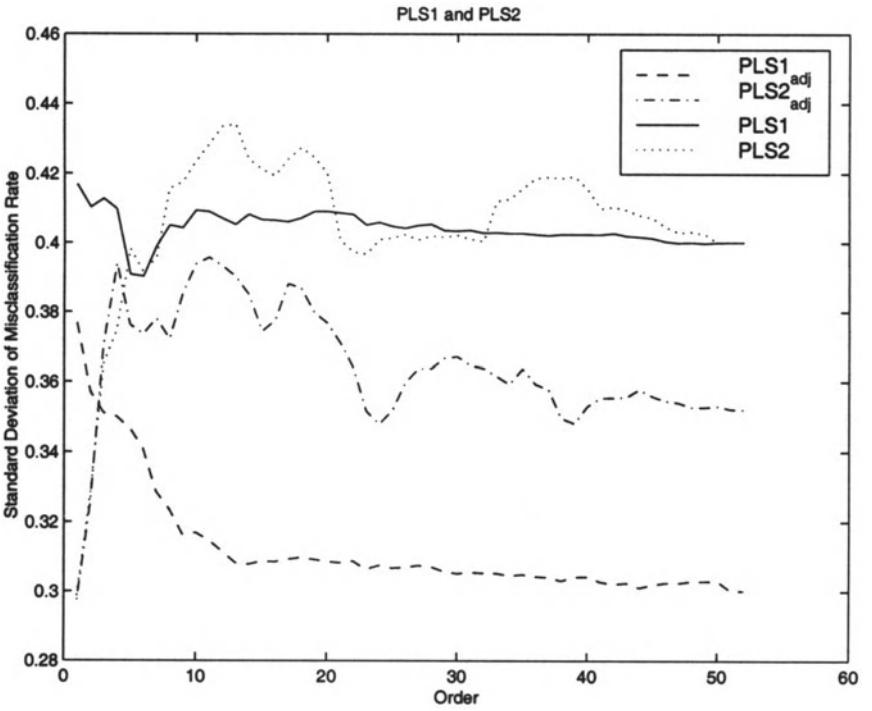

Fig. 10.26. The standard deviation of misclassification rates for the testing set for various orders using PLS1, PLS2, PLS1_{adj}, and PLS2_{adj}

$$r_t = \frac{pat_1 + pc_1t_2}{a(t_1 + \epsilon) + c_2t_2} \tag{10.1}$$

This ratio is much greater than 1 when p is large.

The overall misclassification rates for the training and testing sets and the information criterion (5.12) for various orders using PCA1 are plotted in Figure 10.27. At $a = 52$, the overall misclassification rates for the T^2 statistics based on PCA1 and MS were the same (0.214). This verifies the discussion in Section 4.6 that PCA1 reduces to MS when $a = m$. Regardless of order selected, all FDA methods always gave a lower overall misclassification rate than PCA1 (see Figure 10.20, 10.21, and 10.27). This suggests that FDA model has an advantage over PCA model for diagnosing faults.

It is interesting to see that when all of the factors are included in the FDA methods, the overall misclassification rates were about 0.20, which were different than the overall misclassification rate produced by MS. This is because, when $a = m$, the matrices W_a in (5.16) and $W_{mix,a}$ in (5.17) are not

necessarily orthogonal, and so may not project the data into an orthogonal space.

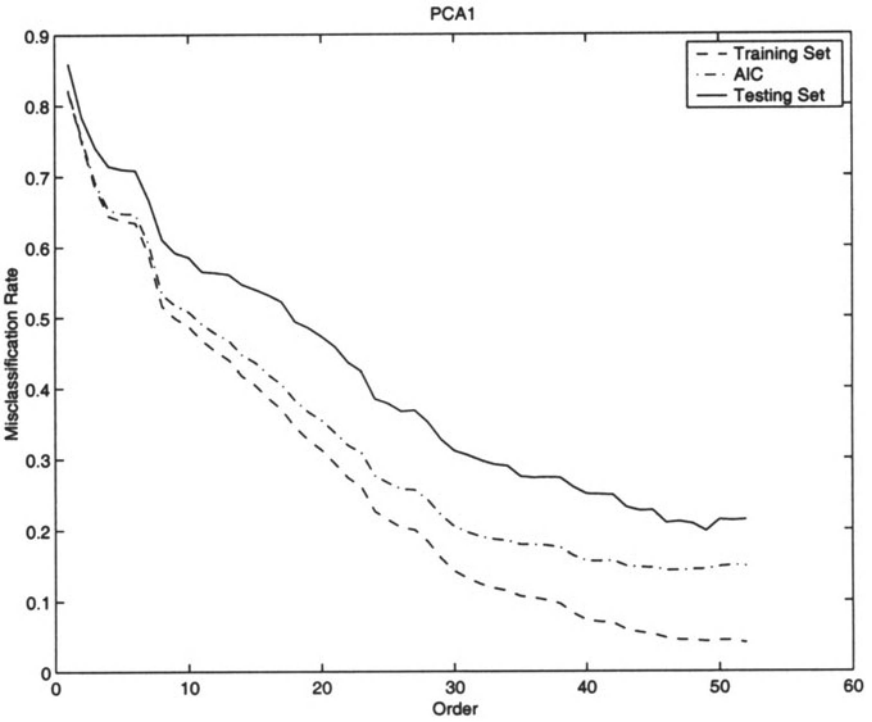

Fig. 10.27. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using PCA1

The PCAm-based and DPCAm-based statistics produced high overall misclassification rates (see Table 10.12). A weakness of the PCAm-based statistics is that PCAm reduces the dimensionality of each class by using the information in only one class but not the information from all the classes. As shown in Tables 10.13, the T^2 statistic based on PCA1 gave a much lower misclassification rate than the statistic based on PCAm for almost all faults.

Now let us consider the PCA, DPCA, and CVA fault diagnosis statistics, all of which separate the dimensionality into a state or score space, and a residual space. For some faults the state or score space version of the statistic gave lower misclassification rates; in other cases the residual space statistics gave lower misclassification rates. Hence, a complete fault diagnosis approach should contain score/state space and residual statistics.

The misclassification rates for the 21 faults were separated into three time periods after the occurrence of the fault (0-5, 5-24, and 24-40 hours), and have been tabulated in Tables 10.15 to 10.20. These tables indicate that each fault diagnosis statistic gives the lowest misclassification rate for some choice of fault and time period. There is no single fault diagnosis statistic that is optimal for all faults or all time periods.

Fault 6 is one of the more interesting faults, so it will be investigated in more detail here. For the time period 0-5 hours after the fault occurred, only the (D)PCAm-based statistics had high misclassification rates (see Table 10.15). For the time period 5-24 hours after the fault occurred, the (D)PCAm-based statistics have low misclassification rates, while the discriminant PLS methods have high misclassification rates (see Table 10.17). For the time period 24-40 hours after the fault occurred, each fault diagnosis technique has a zero misclassification rate except for the discriminant PLS methods, which have nearly 100% misclassification.

The very poor behavior of the discriminant PLS method for Fault 6 after $t = 5$ hours is somewhat surprising when studying the extreme process behavior caused by the fault. For Fault 6, there is a feed loss of A in Stream 1 at $t = 8$ hours (see Figures 8.1 and 10.28), the control loop on Stream 1 reacts to fully open the A feed valve. Since there is no reactant A in the feed, the reaction will eventually stop. This causes the gaseous reactants D and E build up in the reactor, and hence the reactor pressure increases. The reactor pressure continues to increase until it reaches the safety limit of 2950 kPa, at this point the valve for control loop 6 is fully open. Clearly, it is very important to detect this fault promptly before the fault upsets the whole process. While the discriminant PLS methods were able to correctly diagnose Fault 6 shortly after the fault, its diagnostic ability degraded nearly to zero once the effects of the fault worked their way through the system (which occurs approximately at $t = 8 + 5 = 13$ hours, see Figure 10.28).

For these data sets it was found that the FDA-based methods gave the lowest misclassification rates averaged over all fault classes (see Table 10.12), and that the MS, PCA1, and CVA T_r^2 statistics gave comparable overall misclassification rates as the FDA methods. Based only on this information, one might hypothesize that dimensionality reduction techniques are not useful for fault diagnosis as their performance is very similar to MS. However, this conclusion would be *incorrect*, even for this particular application. For particular faults and particular time periods, substantially lower misclassification rates were provided by the statistics that used dimensionality reduction (see Tables 10.15 to 10.20). For example, 24-40 hours after Fault 18 occurred, two dimensionality reduction statistics resulted in a zero misclassification rate while one MS statistic had a 70% misclassification rate and the other had a 100% misclassification rate (see Table 10.20).

There are several general reasons that fault diagnosis statistics based on dimensionality reduction are useful in practice. First, there are inherent lim-

Table 10.15. The misclassification rates for 0-5 hours after the Faults 1-11 occurred

Method	Fault Basis	1	2	3	4	5	6	7	8	9	10	11
PCAm	T^2	**1**	1	0.980	**0.830**	**0.910**	0.720	1	1	1	0.870	**1**
PCA1	T^2	**0.190**	0.140	0.790	**0.110**	**0.170**	0	0	0.160	0.880	0.240	**0.360**
PCAm	Q	**0.210**	0.160	1	**0.890**	**0.900**	0.400	0.480	0.250	1	0.980	**0.860**
PCAm	T^2&Q	**0.330**	0.280	1	**1**	**0.990**	0.610	0.870	0.400	1	1	**0.970**
DPCAm	T^2	**1**	0.960	0.690	**0.740**	**0.470**	0.380	0.800	1	0.990	0.860	**0.970**
DPCAm	Q	**0.240**	0.340	1	**0.950**	**0.930**	0.500	0.370	0.240	1	1	**0.870**
DPCAm	T^2&Q	**0.300**	0.270	1	**1**	**1**	0.710	0.840	0.440	1	1	**0.980**
PLS1	–	**0.090**	0.090	0.940	**0.160**	0	0	0	0.840	0.950	0.810	**0.990**
PLS2	–	**0.090**	0.100	0.950	**0.120**	**0.010**	0	0	0.850	0.940	0.790	**0.990**
$PLS1_{adj}$	–	**0.140**	0.180	0.830	**0.330**	**0.110**	0.100	0	0.840	0.810	0.790	**0.860**
$PLS2_{adj}$	–	**0.140**	0.180	0.870	**0.310**	**0.110**	0.070	0	0.840	0.850	0.770	**0.870**
CVA	T_s^2	**0.200**	0.080	0.950	**0.990**	**0.430**	0.010	0.530	0.270	0.980	0.740	**0.910**
CVA	T_r^2	**0.210**	0.140	0.900	**0.330**	**0.280**	0.010	0.010	0.210	0.950	0.210	**0.200**
CVA	Q	**0.570**	0.460	0.970	**0.910**	**0.480**	0.110	0.330	0.620	0.950	0.830	**0.890**
FDA	T^2	**0.200**	0.150	0.800	**0.140**	**0.160**	0	0	0.160	0.910	0.240	**0.370**
FDA/PCA1	T^2	**0.190**	0.150	0.750	**0.110**	**0.160**	0	0	0.160	0.910	0.280	**0.340**
FDA/PCA2	T^2	**0.200**	0.150	0.800	**0.140**	**0.160**	0	0	0.160	0.910	0.240	**0.370**
DFDA/DPCA1	T^2	**0.210**	0.150	0.740	**0.130**	**0.180**	0	0	0.160	0.930	0.210	**0.150**
MS	T_0^2	**0.200**	0.150	0.800	**0.140**	**0.160**	0	0	0.160	0.910	0.240	**0.370**
MS	T_1^2	**0.280**	0.260	0.950	**0.350**	**0.320**	0	0	0.130	0.970	0.220	**0.190**

Table 10.16. The misclassification rates for 0-5 hours after the Faults 12-21 occurred

Method	Fault Basis	12	13	14	15	16	17	18	19	20	21	Avg.
PCAm	T^2	0.810	1	0.200	0.970	0.870	0.670	0.990	0.500	0.790	0.620	0.844
PCA1	T^2	0.010	0.360	0.040	0.780	0.190	0.320	0.840	0.040	0.740	0.240	0.314
PCAm	Q	0.190	0.440	0.330	0.990	0.980	0.510	0.750	0.570	0.930	0.990	0.658
PCAm	$T^2\&Q$	0.003	0.700	0.500	1	1	0.470	0.820	0.890	1	1	0.754
DPCAm	T^2	0.640	1	0.480	0.950	0.870	0.650	1	0.900	0.850	0.190	0.780
DPCAm	Q	0.240	0.440	0.130	1	0.990	0.390	0.800	0.410	0.960	0.970	0.656
DPCAm	$T^2\&Q$	0.050	0.650	0.160	1	1	0.490	0.710	0.940	1	1	0.740
PLS1	–	0.970	0.840	0.990	0.970	0.940	0.270	0.770	0.940	0.620	0.550	0.606
PLS2	–	0.970	0.830	0.990	0.990	0.870	0.270	0.800	0.970	0.600	0.880	0.663
$PLS1_{adj}$	–	0.950	0.940	0.920	0.910	0.740	0.490	0.840	0.820	0.770	0.530	0.614
$PLS2_{adj}$	–	1.000	0.950	0.930	0.890	0.710	0.470	0.850	0.810	0.790	0.520	0.616
CVA	T_s^2	0	0.610	0.160	0.980	0.620	0.280	0.930	0.210	0.730	1	0.553
CVA	T_r^2	0.010	0.390	0	0.560	0.160	0.200	0.830	0	0.540	0.990	0.340
CVA	Q	0.240	0.690	0.450	0.980	0.960	0.320	0.730	0.880	0.820	0.950	0.673
FDA	T^2	0.010	0.360	0.020	0.760	0.180	0.280	0.840	0.020	0.640	0.100	0.302
FDA/PCA1	T^2	0.010	0.370	0.020	0.840	0.200	0.320	0.830	0.040	0.630	0.230	0.311
FDA/PCA2	T^2	0.010	0.360	0.020	0.760	0.180	0.280	0.840	0.020	0.640	0.099	0.301
DFDA/DPCA1	T^2	0.020	0.360	0	0.800	0.300	0.200	0.790	0.090	0.610	0.480	0.310
MS	T_0^2	0.010	0.360	0.020	0.760	0.180	0.280	0.840	0.020	0.640	0.020	0.298
MS	T_1^2	0	0.320	0	0.640	0.220	0.170	0.770	0	0.610	0.030	0.306

Table 10.17. The misclassification rates for 5-24 hours after the Faults 1-11 occurred

Method	Fault Basis	1	2	3	4	5	6	7	8	9	10	11
PCAm	T^2	**0.645**	0.345	0.939	**0.832**	**0.934**	0.021	0.953	0.995	0.995	0.845	**0.984**
PCA1	T^2	0	0	0.721	**0.189**	0	0	0	0.021	0.782	0.121	**0.226**
PCAm	Q	**0.003**	0.008	0.992	**0.963**	**0.903**	0	0.416	0.321	0.997	0.989	**0.863**
PCAm	T^2&Q	0	0	1	**1**	**0.979**	0	0.511	0.466	1	1	**0.974**
DPCAm	T^2	**0.834**	0.352	0.732	**0.737**	**0.876**	0.003	0.792	1	0.984	0.771	**0.916**
DPCAm	Q	**0.011**	0.037	0.997	**0.966**	**0.853**	0	0.318	0.208	0.997	0.995	**0.850**
DPCAm	T^2&Q	0	0	1	**1**	**1**	0	0.634	0.466	1	1	**0.979**
PLS1	–	0	0	0.955	**0.158**	**0.008**	0.076	0	0.840	0.950	0.810	**0.990**
PLS2	–	0	0	0.974	0.095	**0.008**	0.797	0	0.803	0.995	0.518	**0.984**
$PLS1_{adj}$	–	0	0	0.840	**0.355**	**0.034**	0.887	0	0.879	0.940	0.505	**0.868**
$PLS2_{adj}$	–	0	0	0.850	**0.324**	**0.034**	0.890	0	0.879	0.961	0.492	**0.890**
CVA	T_s^2	**0.005**	0	0.929	**0.984**	**0.016**	0	0.553	0.582	0.963	0.742	**0.887**
CVA	T_r^2	0	0.003	0.789	**0.400**	**0.008**	0	0	0.050	0.842	0.079	**0.100**
CVA	Q	**0.213**	0.150	0.979	**0.890**	**0.121**	0	0.582	0.620	0.950	0.830	**0.890**
FDA	T^2	0	0	0.716	**0.213**	0	0	0	0.021	0.771	0.118	**0.221**
FDA/PCA1	T^2	0	0	0.716	**0.184**	0	0	0	0.029	0.787	0.142	**0.226**
FDA/PCA2	T^2	0	0	0.716	**0.213**	0	0	0	0.021	0.771	0.118	**0.221**
DFDA/DPCA1	T^2	0	0	0.726	**0.174**	0	0	0	0.013	0.824	0.092	**0.097**
MS	T_0^2	0	0	0.716	**0.213**	0	0	0	0.021	0.771	0.118	**0.221**
MS	T_1^2	0	0	0.866	**0.468**	0	0	0	0.005	0.874	0.063	**0.095**

Table 10.18. The misclassification rates for 5-24 hours after the Faults 12-21 occurred

Method	Fault Basis	12	13	14	15	16	17	18	19	20	21	Avg.
PCAm	T^2	0.813	1	0.247	0.958	0.871	0.461	0.018	0.397	0.758	0.900	0.710
PCA1	T^2	0.005	0.211	0.018	0.700	0.187	0.145	0.124	0.084	0.061	0.095	0.176
PCAm	Q	0.200	0.621	0.242	0.995	0.992	0.416	0.124	0.587	0.900	0.989	0.596
PCAm	$T^2\&Q$	0.126	0.850	0.397	1	1	0.405	0.063	0.839	1	1	0.648
DPCAm	T^2	0.697	1	0.555	0.971	0.868	0.616	0.103	0.808	0.658	0.403	0.699
DPCAm	Q	0.161	0.558	0.100	0.995	0.995	0.308	0.116	0.489	0.934	0.976	0.565
DPCAm	$T^2\&Q$	0.147	0.818	0.126	1	1	0.387	0.058	0.942	1	1	0.646
PLS1	–	0.979	0.942	0.995	0.982	0.874	0.097	0.587	0.916	0.266	0.050	0.541
PLS2	–	0.947	0.942	1.000	0.979	0.882	0.129	0.561	0.945	0.200	0.053	0.562
$PLS1_{adj}$	–	0.866	0.982	0.929	0.913	0.634	0.358	0.711	0.795	0.458	0	0.569
$PLS2_{adj}$	–	0.882	0.976	0.934	0.921	0.640	0.345	0.700	0.761	0.468	0	0.569
CVA	T_s^2	0	0.516	0.218	0.953	0.526	0.205	0.155	0.421	0.253	0.982	0.471
CVA	T_r^2	0.003	0.324	0.003	0.650	0.132	0.045	0.063	0.003	0.024	0.966	0.214
CVA	Q	0.229	0.571	0.413	0.992	0.816	0.184	0.711	0.929	0.542	0.940	0.602
FDA	T^2	0.005	0.213	0.003	0.716	0.184	0.116	0.132	0.024	0.063	0.013	0.168
FDA/PCA1	T^2	0.003	0.221	0.003	0.732	0.163	0.090	0.129	0.032	0.068	0.192	0.177
FDA/PCA2	T^2	0.005	0.213	0.003	0.716	0.184	0.116	0.132	0.024	0.063	0.005	0.168
DFDA/DPCA1	T^2	0.003	0.195	0.003	0.737	0.174	0.013	0.116	0.100	0.092	0.416	0.180
MS	T_0^2	0.005	0.213	0.003	0.716	0.184	0.116	0.516	0.024	0.063	0	0.186
MS	T_1^2	0	0.168	0.003	0.661	0.240	0.018	0.108	0.003	0.097	0	0.175

Table 10.19. The misclassification rates for 24-40 hours after the Faults 1-11 occurred

Method	Fault Basis	1	2	3	4	5	6	7	8	9	10	11
PCAm	T^2	**0.622**	0.303	0.925	**0.779**	**1**	0	1	1	0.988	0.847	**0.991**
PCA1	T^2	0	0	0.853	**0.147**	0	0	0	0	0.744	0.097	**0.203**
PCAm	Q	0	0	0.988	**0.956**	**0.928**	0	0.369	0.216	0.991	0.988	**0.853**
PCAm	$T^2\&Q$	0	0	1	**1**	**0.959**	0	0.363	0.344	1	1	**0.959**
DPCAm	T^2	**0.897**	0.384	0.669	**0.694**	**0.997**	0	0.978	1	0.991	0.672	**0.978**
DPCAm	Q	0	0	0.991	**0.966**	**0.837**	0	0.347	0.103	0.997	0.994	**0.825**
DPCAm	$T^2\&Q$	0	0	1	**1**	**1**	0	0.566	0.303	1	1	**0.988**
PLS1	–	**0.003**	0.003	0.972	**0.188**	**0.006**	0.997	0.003	0.806	0.969	0.684	**0.984**
PLS2	–	**0.003**	0.003	0.969	**0.147**	**0.006**	0.997	0.003	0.753	0.975	0.619	**0.969**
$PLS1_{adj}$	–	**0.003**	0.003	0.916	**0.384**	**0.034**	0.997	0.003	0.879	0.940	0.505	**0.868**
$PLS2_{adj}$	–	**0.003**	0.003	0.909	**0.319**	**0.031**	0.997	0.003	0.816	0.881	0.597	**0.891**
CVA	T_s^2	0	0	0.950	**0.975**	0	0	0.772	0.519	0.972	0.750	**0.922**
CVA	T_r^2	0	0	0.834	**0.316**	**0.003**	0	0	0.013	0.822	0.084	**0.166**
CVA	Q	**0.181**	0.066	0.978	**0.884**	**0.141**	0	0.650	0.650	0.972	0.834	**0.900**
FDA	T^2	0	0	0.850	**0.144**	0	0	0	0	0.731	0.113	**0.234**
FDA/PCA1	T^2	0	0	0.800	**0.153**	0	0	0	0	0.731	0.138	**0.234**
FDA/PCA2	T^2	0	0	0.850	**0.144**	0	0	0	0	0.731	0.113	**0.234**
DFDA/DPCA1	T^2	0	0	0.743	**0.151**	0	0	0	0	0.734	0.078	**0.132**
MS	T_0^2	0	0	0.850	**0.144**	0	0	0	0	0.731	0.113	**0.234**
MS	T_1^2	0	0	0.890	**0.401**	0	0	0	0	0.841	0.100	**0.132**

Table 10.20. The misclassification rates for 24-40 hours after the Faults 12-21 occurred

Method	Fault Basis	12	13	14	15	16	17	18	19	20	21	Avg.
PCAm	T^2	0.906	1	0.253	0.966	0.797	0.403	0.569	0.375	0.756	0.984	0.736
PCA1	T^2	0.044	0.225	0.056	0.844	0.219	0.209	0.616	0.197	0.053	0.156	0.222
PCAm	Q	0.244	0.378	0.291	0.994	0.975	0.384	0.600	0.753	0.931	0.962	0.610
PCAm	$T^2\&Q$	0.350	0.656	0.466	1	1	0.403	0.478	0.916	1	1	0.662
DPCAm	T^2	0.722	1	0.600	0.959	0.700	0.684	0.300	0.731	0.722	0.784	0.736
DPCAm	Q	0.241	0.303	0.116	0.997	0.981	0.313	0.600	0.700	0.959	0.919	0.580
DPCAm	$T^2\&Q$	0.347	0.597	0.181	1	1	0.394	0.600	0.978	1	1	0.664
PLS1	–	0.997	0.244	0.997	0.991	0.894	0.163	1.000	0.903	0.316	0.006	0.577
PLS2	–	0.953	0.181	0.997	0.988	0.903	0.172	1.000	0.947	0.253	0.006	0.564
$PLS1_{adj}$	–	0.847	0.419	0.938	0.944	0.669	0.391	0.997	0.803	0.488	0.003	0.571
$PLS2_{adj}$	–	0.856	0.388	0.925	0.938	0.666	0.388	0.997	0.790	0.509	0.003	0.567
CVA	T_s^2	0.022	0.434	0.197	0.972	0.600	0.213	0.875	0.609	0.237	0.890	0.519
CVA	T_r^2	0.044	0.313	0	0.719	0.156	0.044	0	0.009	0.028	0.072	0.173
CVA	Q	0.388	0.584	0.494	0.975	0.878	0.225	1.000	0.944	0.569	0.897	0.629
FDA	T^2	0.034	0.231	0.022	0.822	0.206	0.150	0.369	0.063	0.041	0.063	0.194
FDA/PCA1	T^2	0.034	0.238	0.022	0.819	0.203	0.156	0.584	0.088	0.034	0.194	0.211
FDA/PCA2	T^2	0.034	0.231	0.022	0.822	0.206	0.150	0.369	0.063	0.041	0.047	0.193
DFDA/DPCA1	T^2	0.066	0.229	0.006	0.834	0.245	0.028	0	0.207	0.141	0.009	0.172
MS	T_0^2	0.034	0.231	0.022	0.822	0.206	0.150	1.000	0.063	0.041	0.031	0.222
MS	T_1^2	0.013	0.219	0	0.828	0.285	0.019	0.709	0.003	0.088	0	0.216

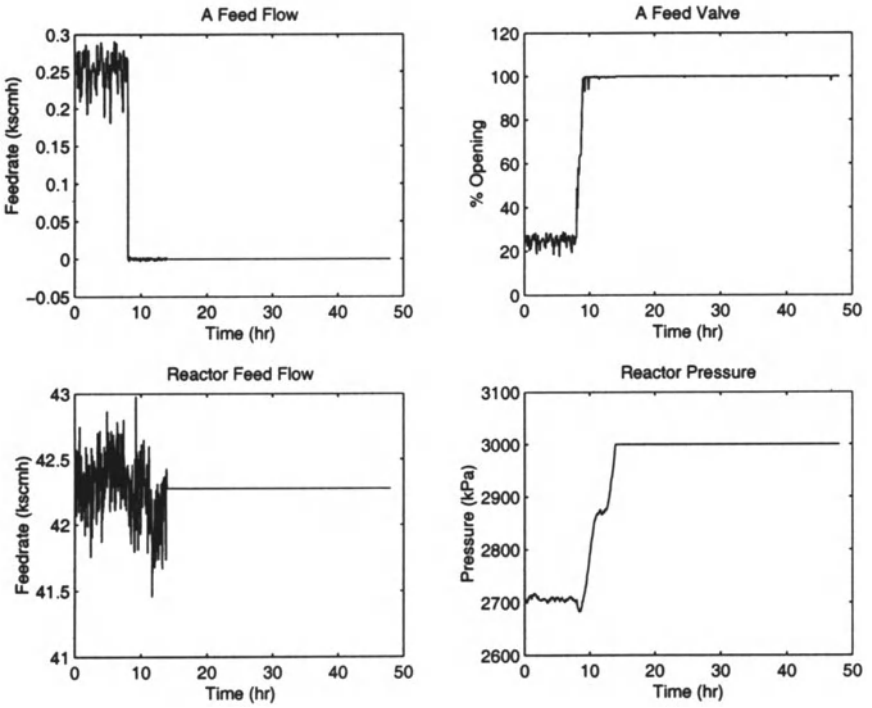

Fig. 10.28. Closed loop simulation for a step change of A feed loss in Stream 1 (Fault 6)

itations due to roundoff errors that usually prevent the construction of full dimensional models for large scale systems such as chemical plants. Second, there can be limitations on the size of the models used by process monitoring methods that can be implemented in real time on the computer hardware connected to a particular process. While this limitation is becoming less of an issue over time, the authors are aware of industrial control systems still using older control computers.

The main reason for dimensionality reduction is based on the amount of data usually available in practice that has been sufficiently characterized for use in process monitoring. This data, for example, should be cleaned of all outliers caused by computer or database programming errors [178]. For the application of fault diagnosis methods it is required to label each observation as being associated with normal operating conditions or with a particular fault class. These requirements can limit the available training data, especially for the purposes of computing fault diagnosis statistics, to less than what was used in this chapter.

To illustrate the relationship between data dimensionality and the size of the training set, 100 data points were collected for each fault class in the training set (for all other simulations shown in this chapter, 500 data points were collected in the training sets). The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using PCA1 are plotted in Figure 10.29. Although the misclassification rates reduced nearly to zero as a goes to 52 for the training set, the overall misclassification rates for the testing set were very high as compared to Figure 10.27. Recall that PCA1 reduces to the MS statistic when $a = 52$, this shows that the MS statistic gives a higher overall misclassification rate for many reduction orders ($a = 20$ to 45, as seen in Figure 10.29). In the case where the number of data points in the training set is insufficient (the usual case in practice), errors in the sample covariance matrix will be significant. In such cases there is an advantage to using dimensionality reduction techniques. The relationship between reduction order and the size of the training set is further investigated in Homework Problem 11.

The purpose of dimensionality reduction techniques (PCA, FDA, PLS, and CVA) is to reduce the dimensions of the data while retaining the most useful information for process monitoring. In most cases, the lower dimensional representations of the data will improve the proficiency of detecting and diagnosing faults.

10.9 Homework Problems

1. A co-worker at a major chemical company suggested that false alarms were not an issue with fault identification and that it may be useful to apply all the scores (not just the first a scores) for the PCA, DPCA, and CVA-based $CONT$ as shown in Section 4.5. Evaluate the merits of the proposal. Apply this idea to the data collected from the Tennessee Eastman plant simulator (http://brahms.scs.uiuc.edu). What are your conclusions?
2. Apply the similarity index (4.41) and mean overlap (4.42) to the data collected from the Tennessee Eastman plant simulator. Relate your results with these two measures with the misclassification rates of the fault diagnosis statistics as reported in this chapter. Do the similarity index and mean overlap assess the likelihood of successful diagnosis? Explain in detail why one measure performs better than the other.
3. As discussed in Chapter 5, (D)FDA only ranks the eigenvectors associated with the nonzero eigenvalues. Propose a method other than PCA1 to rank the eigenvectors associated with the zero eigenvalues. Evaluate your proposal using the data collected from the Tennessee Eastman plant simulator.
4. In addition to the original 21 faults for the TEP, simulate 39 additional multiple faults (combination of two faults) of your choice. Apply FDA,

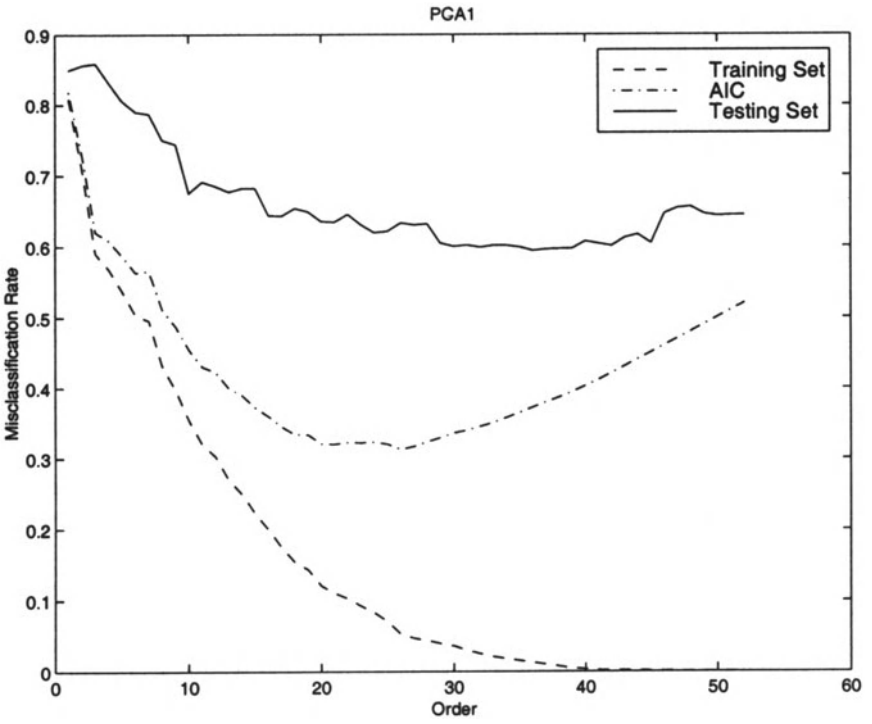

Fig. 10.29. The overall misclassification rates for the training and testing sets and the information criterion (AIC) for various orders using PCA1 with 100 data points in the training set

FDA/PCA1, FDA/PCA2, and their corresponding dynamic version to diagnose these 60 faults and comment on your findings.

5. A co-worker at a major chemical company proposed to modify the model complexity term in the information criterion (5.12) to $1.5a/\tilde{n}$. Based only on the performance as given by Figure 10.23 which was obtained by an application of the original information criterion (5.12) to a simulated chemical plant, evaluate the relative merits of the co-worker's proposal. Another co-worker suggested to modify the model complexity term in the information criterion (5.12) to a/n. Evaluate the relative merits of the second proposal. Based on Figure 10.23, propose a modification of the model complexity term which will give the best results for the simulated chemical plant. How well does your modified model complexity term perform? [Note that designing the best information criterion for one specific process application does not necessarily give the best possible information criterion for other process applications.]

6. Formulate dynamic discriminant PLS for diagnosing faults. Apply this approach to the data collected from the Tennessee Eastman plant simulator. Compare the results with the discriminant PLS results as shown in this chapter. Does dynamic discriminant PLS perform better?
7. Discuss the effect of lag order h and state order k selection on the fault detection performance using all the CVA statistics. Apply the Q, T_s^2, and T_r^2 statistics for fault detection to the data collected from the Tennessee Eastman plant simulator. Now, perturb h and k from their optimal values. Report on your results. Which statistic deviates the most? Why?
8. Describe in detail how to formulate CVA for fault diagnosis. Apply these techniques to the data collected from the Tennessee Eastman plant simulator. How do these fault diagnosis results compared with the results reported in this chapter?
9. Write a report describing in detail how to implement PCA and PLS with EWMA and CUSUM charts to detect faults. Apply this technique to the data collected from the Tennessee Eastman plant simulator. Compare the results with the DPCA results as shown in this chapter. Which technique seems to better capture the serial correlations of the data? Justify your findings. List an advantage and disadvantage of using each technique.
10. A co-worker proposed to average each measurement over a period of time before applying the data to the process monitoring algorithms. Evaluate the merits of this "moving window" proposal and apply the approach to PCA, DPCA, and CVA for fault detection using the data collected from the Tennessee Eastman plant simulator. Investigate the effect of the number of data points used in the averaging on the process monitoring performance. Was it possible to improve on DPCA and CVA using this approach? Justify your answers.
11. Evaluate the effects of the size of training set and the sampling interval on the reduction order and process monitoring performance. Construct training and testing data sets for the TEP using (i) 150 points with a sampling interval of 10 minutes, (ii) 1500 points with a sampling interval of 1 minute, and (iii) 1500 points with a sampling interval of 10 minutes. Implement all process monitoring statistics described in this book. How is the relative performance of each process monitoring statistic affected? Why? How is the reduction order affected? Compare the techniques in terms of the sensitivity of their performance to changes in the size of the training set and the sampling interval.
12. While the threshold for the Q statistic (Equation 4.22) is widely used in practice, its derivation relies on certain assumptions that are not always true (as mentioned in Section 10.6). Write a report on the exact distribution for Q and how to compute the exact threshold for the Q statistic. Under what conditions is Equation 4.22 a valid approximation? Would these conditions be expected to hold for most applications to process data collected from large scale chemical plants? (Hint: Several papers that de-

scribe the exact distribution for Q are cited at the end of the paper by Jackson and Mudholkar [101].)

Part V

OTHER APPROACHES

CHAPTER 11
OVERVIEW OF ANALYTICAL AND KNOWLEDGE-BASED APPROACHES

As discussed in Section 1.2, process monitoring measures are derived based on the data-driven, analytical, or knowledge-based approaches. This book focuses mostly on the data-driven methods, which include control charts (Shewhart, CUSUM, and EWMA charts) and dimensionality reduction techniques (PCA, PLS, FDA, and CVA). A well-trained engineer should also have some familiarity with the analytical and knowledge-based approaches since they have advantages for some process monitoring problems. The analytical approach can provide improved process monitoring when an accurate first-principles model is available. Also, both analytical and knowledge-based approaches can incorporate process flowsheet information in a straightforward way.

Given that several detailed reviews of analytical and knowledge-based approaches are available [53, 92, 95, 93, 117], only a high level overview with pointers to some representative works is provided here. This should provide enough background to determine which approach is likely to be most promising in a particular application, with enough references for the reader to know where to go to learn about implementation. Analytical approaches based on parameter estimation, state estimation, and analytical redundancy are discussed in Sections 11.1 and 11.2. Knowledge-based approaches based on causal analysis and expert systems are discussed in Section 11.3. The use of pattern recognition approaches for process monitoring is covered in Section 11.4. The chapter concludes in Section 11.5 by discussing combinations of various techniques.

11.1 Parameter and State Estimation

Parameter and state estimation are two of the quantitative approaches for fault detection and diagnosis based on detailed mathematical models. Assuming the system is observable and appropriate mathematical models are available, these approaches are suitable for detecting and diagnosing faults associated with parameter or state changes.

The state estimation approach is appropriate if the process faults are associated with changes in unmeasurable state variables. The states are reconstructed from the measurable input and output variables of the process

E. L. Russell et al., *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*

using a state observer, also known as a Kalman filter [19, 23, 104]. Thresholds on some or all of the changes in the estimated state can be defined similarly as done for Canonical Variate Analysis (CVA) in Chapter 7. The main difference is that CVA constructs the states directly from the process data, rather than through the use of state space equations and a state observer. Numerous variations of this approach which can detect abrupt changes in the state variables and the output variables have been developed [227]. Many of these approaches focus on detecting changes in the process noise, actuator behavior, and sensor behavior.

The parameter estimation approach is appropriate if the process faults are associated with changes in parameters in the process model. The model parameters can be estimated using the standard least-squares techniques of parameter estimation [12], which can be implemented recursively to reduce computational expense. Constructing the models from first-principles facilitates relating the model parameters directly to parameters that have physical meaning in the process. Thresholds can be placed on the individual differences between the nominal model parameters and the parameter estimates, or on some combination of these differences.

As the number of faults represented by the undetermined parameters in the model grows large, observability may be violated and structural parameters cannot typically be included in the estimation models. To solve both problems, parallel estimators can be used to reduce the number of adjustable parameters per model and/or replace structural parameterizations with explicitly enumerated structural alternatives [117]. Because of the strict modeling requirement, the parallel estimator approach is not extensively applied in most large scale chemical plants.

Several reviews describing process monitoring methods based on parameter and state estimation are available [91, 53, 95, 93]. Several recent papers have been published using these approaches [244, 153, 92, 93, 96, 109, 124, 160].

11.2 Analytical Redundancy

The state and parameter estimation approaches are subsets of a broader approach known as analytical redundancy, which is the underlying principle behind the analytical approach to process monitoring. Approaches that use analytical redundancy incorporate an explicit process model to generate and evaluate residuals [54, 154]. In some cases this residual is generated between the model prediction and the observed behavior. The observer-based approach can be used to reconstruct the output of the system from the measurements or a subset of the measurements with the aid of observers, in which case the output estimation error is used as the residual [53, 35]. In the case of parameter estimation, the residuals can be taken as the difference between the nominal model parameters and the estimated model parameters. In the

case of state estimation, the residuals can be taken as the difference between the nominal state variables and the estimated state variables. The residuals can also be defined in terms of some combination of the states or parameters, as in the case of parity equations [82, 64, 196, 118]. While first-principles models are preferred, empirical models such as artificial neural networks can be used [110, 30].

The residuals can be caused by unknown process disturbances, measurement noise, faults, and model uncertainty. Quantifying the contribution of the unknown process disturbances and measurement noise on the residuals is rather straightforward provided that the disturbances and noise are modeled stochastically. Characterizing the model uncertainties and quantifying their effect on the residuals are more difficult. The larger the model uncertainty, the more difficult it is to detect and diagnose faults using residuals. Much attention has been focused on improving the robustness of analytical redundancy approaches to model uncertainty. Two of the more dominant methods are to use robust residual generators [221, 69, 55], or to use structured residuals with an unknown input observer [196, 56, 174].

The second step of analytical redundancy approaches is the **residual evaluation** step, in which the resulting residual is used as feature inputs to fault detection and diagnosis through logical, causal, or pattern recognition techniques. Gomez *et al.* [67] suggested using operating point computation, the Hotelling's statistic, and Scheffee's statistic to detect the normality of the residuals. The results are then formulated as a fuzzy logic rule for detecting and diagnosing faults. Frank and Kiupel [57] evaluated the residual based on fuzzy logic incorporated with either adaptive thresholds or fuzzy inference with the assistance of a human operator. Garcia and Frank [58] proposed a method to integrate the observer-based approach with the parameter estimation approach. The observer-based residual is used for fault detection; when the signals are sufficiently rich, the parameter identification residual is then used for fault diagnosis. Ding and Guo [35] suggested that integrating the residual generation and residual evaluation stages of fault detection and diagnosis design may improve the performance. They proposed a frequency domain approach to design an integrated fault detection system. Many other recent papers on analytical redundancy based approaches are available [36, 136, 22, 168, 176, 177, 14].

11.3 Causal Analysis and Expert Systems

Approaches based on causal analysis use the concept of **causal modeling** of **fault-symptom** relationships.

The **signed directed graph** (SDG) is a qualitative model-based approach for fault diagnosis that incorporates causal analysis [90, 197, 210]. The SDG represents pathways of causality in normal operating conditions. In a SDG, each node represents a process variable, which can be classified as

normal, high, or low. Each arc represents the causal relationship between the nodes. The direction of deviation of the nodes is represented by signs on the arcs. Assuming that a single fault affects only a single node (root node) and that the fault does not change other causal pathways in the SDG, the causal linkages will connect the fault origin to the observed symptoms of the fault. The advantages of SDG are that all the possible root nodes can be located. However, the SDG is tedious to develop for large scale chemical plants.

The processing time required for using the SDG can be reduced by compiling the SDG into rules [115]. The SDG has been extended to handle variables with compensatory response and inverse response [49, 173]. A digraph-based diagnosis reasoning approach known as the **possible cause-effect graph** can reduce the search space [226]. The SDG has also been extended to multiple fault diagnosis by assuming that the probability of occurrence of a multiple fault decreases with an increasing number of faults [219]. Several recent papers based on the SDG are available [87, 158, 206].

Many recent applications of SDG have made use of expert systems. **Expert systems** are knowledge-based techniques which are closer in style to human problem solving. Analysis proceeds deductively in a sequence of logical steps. Expert systems based on heuristics and expert testimony are called **experiential knowledge** expert systems, while those based on models are called **model-based** knowledge expert systems [116]. A combination of SDG, expert systems, and fuzzy logic was applied to a simulated propane evaporator in which 38 clusters representing 50 faults were considered [205].

A traditional approach to build an experiential knowledge expert system is to develop IF-THEN rules through expert knowledge; then the rules encode the domain knowledge [116]. Experiential knowledge expert systems are flexible and their conclusion can be easily verified and explained. A model-based expert system based on engineering fundamentals can supplement an experiential expert system for fault diagnosis.

Another approach to build an experiential knowledge expert system for fault diagnosis is through machine learning techniques. One approach is to integrate the symbolic information into an artificial neural network learning algorithm [202]. Such a learning system allows for knowledge extraction and background knowledge encoding in the form of rules; fuzzy logic is used to deal with uncertainty in the learning domain. Several recent papers on fault diagnosis using expert systems are available [21, 20].

A closely related representation to the SDG that can be used in causal analysis is the **symptom tree model** (STM). The STM is a real time version of the fault tree model that relates the faults and symptoms [237, 240, 241]. In STM, the root cause of a fault is determined by taking the intersection of causes attached to observed symptoms. It is highly likely that this procedure will result in more than one candidate fault, and it is impossible to determine the most probable cause among the suggested candidates. The **weighted symptom tree model** (WSTM) resolves the problem by at-

taching a weight to each symptom-fault pair, with the weight obtained by training the WSTM. With the WSTM, the proposed candidate faults are ranked according to their probability. In the next step, a pattern matching algorithm is used which matches the observed fault propagation trends with standard fault propagation trends based on training set [172]. The fault that best matches the observed process variable changes is selected as the most probable candidate among the proposed ones.

11.4 Pattern Recognition

Many data-based, analytical, and knowledge based approaches incorporate pattern recognition techniques to some extent. For example, Fisher discriminant analysis is a data-driven process monitoring method based on pattern classification theory. Numerous fault diagnosis approaches described in Part III combined dimensionality reduction (via PCA, PLS, FDA, CVA, or a combination of FDA and PCA) with discriminant analysis, which is a general concept from the pattern recognition literature. Other uses of pattern recognition in process monitoring were discussed earlier in Section 11.3.

Some pattern recognition approaches to process monitoring use the relationship between the data patterns and fault classes without modeling the internal process states or structure explicitly. These approaches include **rule-based systems**, **artificial neural networks** (ANN), and **decision trees**. Since pattern recognition approaches are based on inductive reasoning through generalization from a set of stored or learned examples of process behaviors, these techniques are useful when data are abundant, but when expert knowledge is lacking. A recent review of pattern recognition approaches is available [134].

An artificial neural network (ANN) is a nonlinear mapping between input and output which consists of interconnected "neurons" arranged in layers. The layers are connected such that the signals at the input of the neural net are propagated through the network. The choice of the neuron nonlinearity and the weights of connections between neurons specifies the nonlinear overall behavior of the neural network. Numerous papers are available which apply ANNs to fault detection and diagnosis; many of these techniques were derived from the pattern recognition perspective [80, 79, 213, 13, 175, 239, 68, 11].

Neural network models can also be used for unsupervised learning using a **self-organizing map (SOM)**. An SOM maps the nonlinear statistical dependencies between high-dimensional data into simple geometric relationships, which preserve the most important topological and metric relationships of the original data. This allows data to be clustered without knowing the class memberships of the input data. An SOM has been successfully applied in fault diagnosis [198, 199]. For fault detection, an SOM is trained to form a mapping of the input space during normal operating conditions; a fault can be detected by monitoring the quantization error [4].

11.5 Combinations of Various Techniques

Each fault detection and diagnosis technique has its advantages and disadvantages. Incorporating several techniques for fault detection and diagnosis seems attractive. Garcia and Vilim [59] combined physical modeling, neural processing, and likelihood testing for fault detection. Zhao *et al.* [243] proposed a hybrid ANN integrated with an expert system for dynamic fault diagnosis. Zhang *et al.* [242] combined a feedforward neural network (FNN) and a multiple model adaptive estimator (MMAE) for fault detection and diagnosis. Engelmore and Morgan [45] proposed a fault diagnosis system consisting of diagnostic experts and a scheduler to integrate different diagnostic methods. Mylaraswamy and Venkatasubramanian [162] developed a hybrid, distributed, multiple-expert based framework called Dkit, which integrates SDG, observer-based methods, qualitative trend analysis, and statistical classifiers to perform collective fault diagnosis.

REFERENCES

1. H. Akaike. Stochastic theory of minimal realization. *IEEE Trans. on Auto. Control*, 19:667–674, 1974.
2. H. Akaike. Markovian representation of stochastic processes by canonical variables. *SIAM J. of Control*, 13:162–173, 1975.
3. H. Akaike. Canonical correlation analysis of time series and the use of an information criterion. In R. K. Mehra and D. G. Jainiotis, editors, *System Identification: Advances and Case Studies*, pages 27–96. Academic Press, New York, 1976.
4. E. Alhoniemi, J. Hollmen, O. Simula, and J. Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer-aided Engineering*, 6(1):3–14, 1999.
5. D. J. Allen. Digraphs and fault trees. *Ind. Eng. Chem. Fund.*, 23:175–180, 1984.
6. B. K. Alsberg, R. Goodacre, J. J. Rowland, and D. B. Kell. Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods. *Analytica Chimica Acta*, 348:389–407, 1997.
7. F. B. Alt. Multivariate quality control. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistical Sciences*. John Wiley & Sons, New York, 1985.
8. B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
9. T. W. Anderson. *Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, New York, 1958.
10. H. B. Aradhye, B. R. Bakshi, and R. Strauss. Process monitoring by PCA, dynamic PCA, and multiscale PCA – Theoretical analysis and disturbance detection in the Tennessee Eastman process. In *AIChE Annual Meeting*, 1999. Paper 224g.
11. T. Asakura, T. Kobayashi, and S. Hayashi. A study of fault diagnosis system using neural networks. In *Proc. of the 29th ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, pages 19–24, Tokyo, Japan, 1998.
12. J. V. Beck and K. J. Arnold. *Parameter Estimation in Engineering and Science*. Wiley, New York, 1977.
13. H. Benkhedda and R. J. Patton. B-spline network integrated qualitative and quantitative fault detection. In *Proc. of the 13th IFAC World Congress*, volume N, pages 163–168, Piscataway, NJ, 1996. IEEE Press.
14. G. Betta and A. Pietrosanto. Instrument fault detection and isolation: State of the art and new research trends. In *Proc. of the IEEE Instrumentation and Measurement Technology*, volume 1, pages 483–489, Piscataway, NJ, 1998.
15. C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, New York, 1995.

16. G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters - An Introduction to Design, Data Analysis, and Model Building.* Wiley, New York, 1978.
17. R. D. Braatz and E. L. Russell. Robustness margin computation for large scale systems. *Comp. & Chem. Eng.*, 23:1021–1030, 1999.
18. R. D. Braatz, P. M. Young, J. C. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Trans. on Auto. Control*, 39:1000–1002, 1994.
19. W. L. Brogan. *Modern Control Theory.* Prentice-Hall, New Jersey, 1991.
20. P. Burrell and D. Inman. Expert system for the analysis of faults in an electricity supply network: Problems and achievements. *Computers in Industry*, 37(2):113–123, 1997.
21. C. S. Chang, J. M. Chen, A. C. Liew, D. Srinivasan, and F. S. Wen. Fuzzy expert system for fault diagnosis in power systems. *International Journal of Engineering Intelligent Systems for Electrical Engineering & Communications*, 5(2):75–81, 1997.
22. I. Chang, C. Yu, and C. Liou. Model-based approach for fault diagnosis. 1. principles of deep model algorithm. *Ind. Eng. Chem. Res.*, 33:1542–1555, 1994.
23. C.-T. Chen. *Linear System Theory and Design.* Harcourt Brace College Publishers, Orlando, Florida, 1984.
24. G. Chen and T. J. McAvoy. Predictive on-line monitoring of continuous processes. *J. of Process Control*, 8:409–420, 1997.
25. Y. Q. Cheng, Y. M. Zhuang, and J. Y. Yang. Optimal Fisher discriminant analysis using the rank decomposition. *Pattern Recognition*, 25:101–111, 1992.
26. L. H. Chiang, E. L. Russell, and R. D. Braatz. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Intelligent Systems*, 2000. in press.
27. E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. on Auto. Control*, 29:603–614, 1984.
28. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 30:21–27, 1967.
29. R. B. Crosier. Multivariate generalizations of cumulative sum quality-control schemes. *Technometrics*, 30:291–303, 1988.
30. K. Danai and V. B. Jammu. Robust residual generation for fault diagnosis thru pattern classification. In *Proc. of the 13th IFAC World Congress*, volume N, pages 193–198, Piscataway, NJ, 1996. IEEE Press.
31. S. de Jong. An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.
32. M. Defernez and E. K. Kemsley. The use and misuse of chemometrics for treating classification problems. *Trends in Analytical Chemistry*, 16:216–221, 1997.
33. L. Desborough and T. Harris. Performance assessment measures for univariate feedback control. *Can. J. of Chem. Eng.*, 70:262–268, 1992.
34. W. R. Dillon and M. Goldstein. *Multivariate Analysis, Methods and Applications.* John Wiley & Sons, New York, 1984.
35. X. Ding and L. Guo. Observer-based fault detection optimized in the frequency domain. In *Proc. of the 13th IFAC World Congress*, volume N, pages 157–162, Piscataway, NJ, 1996. IEEE Press.
36. X. Ding and L. Guo. Observer-based optimal fault detector. In *Proc. of the 13th IFAC World Congress*, volume N, pages 187–192, Piscataway, NJ, 1996. IEEE Press.

37. N. Doganaksoy, F. W. Faltin, and W. T. Tucker. Identification of out of control quality characteristics in a multivariate manufacturing environment. *Commun. Stat.- Theory Methods*, 20:2775, 1991.
38. D. Dong and T. J. McAvoy. Nonlinear principal component analysis: based on principal curves and neural networks. *Comp. & Chem. Eng.*, 20:65–78, 1996.
39. J. J. Downs and E. F. Vogel. A plant-wide industrial-process control problem. *Comp. & Chem. Eng.*, 17:245–255, 1993.
40. R. J. Doyle, L. Charest, N. Rouquette, J. Wyatt, and C. Robertson. Causal modeling and event-driven simulation for monitoring of continuous systems. *Computers in Aerospace*, 9:395–405, 1993.
41. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.
42. R. Dunia, S. J. Qin, and T. F. Edgar. Multivariable process monitoring using nonlinear approaches. In *Proc. of the American Control Conf.*, pages 756–760, Piscataway, NJ, 1995. IEEE Press.
43. R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE J.*, 42:2797–2812, 1996.
44. D. L. Dvorak and B. J. Kuipers. Model-based monitoring of dynamic systems. In *11th Int. Conf. on Artificial Intelligence*, pages 1238–1243, Detroit, Michigan, Aug. 1988.
45. R. Engelmore and T. Morgan. *Blackboard Systems.* Addison-Wesley, Menlo Park, CA, 1988.
46. D. F. Enns. Model reduction with balanced realizations: an error bound and a frequency weighted generalization. In *Proc. of the IEEE Conf. on Decision and Control*, pages 127–132, Piscataway, NJ, 1984. IEEE Press.
47. Y. E. Faitakis and J. C. Kantor. Residual generation and fault detection for discrete-time systems using an l_∞ approach. *Int. J. of Control*, 64:155–174, 1996.
48. A. P. Featherstone and R. D. Braatz. Integrated robust identification and control of large scale processes. *Ind. Eng. Chem. Res.*, 37:97–106, 1998.
49. F. E. Finch, O. O. Oyeleye, and M. A. Kramer. A robust event-oriented methodology for diagnosis of dynamic process systems. *Comp. & Chem. Eng.*, 14(12):1379–1396, 1990.
50. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.
51. I. E. Frank. A nonlinear PLS model. *Chemometrics and Intelligent Laboratory Systems*, 8:109–119, 1990.
52. I. E. Frank, J. Feikema, N. Constantine, and B. R. Kowalski. Prediction of product quality from spectral data using the partial least-squares method. *J. of Chemical Information and Computer Sciences*, 24:20–24, 1983.
53. P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results. *Automatica*, 26:459–474, 1990.
54. P. M. Frank. Robust model-based fault detection in dynamic systems. In P. S. Dhurjati and G. Stephanopoulos, editors, *On-line Fault Detection and Supervision in the Chemical Process Industries*, pages 1–13. Pergamon Press, Oxford, 1993. IFAC Symposia Series, Number 1.
55. P. M. Frank. Enhancement of robustness in observer-based fault detection. *Int. J. of Control*, 59:955–981, 1994.
56. P. M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30:789–804, 1994.

57. P. M. Frank and N. Kiupel. Residual evaluation for fault diagnosis using adaptive fuzzy thresholds and fuzzy inference. In *Proc. of the 13th IFAC World Congress*, volume N, pages 115–120, Piscataway, NJ, 1996. IEEE Press.
58. E. A. Garcia and P. M. Frank. On the relationship between observer and parameter identification based approaches to fault detection. In *Proc. of the 13th IFAC World Congress*, volume N, pages 25–29, Piscataway, NJ, 1996. IEEE Press.
59. H. E. Garcia and R. B. Vilim. Combining physical modeling, neural processing, and likelihood testing for online process monitoring. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 806–810, Piscataway, NJ, 1998. IEEE Press.
60. S. Geisser. Discrimination, allocatory and separatory, linear aspects. In *Classification and Clustering*. Academic Press, New York, 1977.
61. P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.
62. C. Georgakis, B. Steadman, and V. Liotta. Decentralized PCA charts for performance assessment of plant-wide control structures. In *Proc. of the 13th IFAC World Congress*, pages 97–101, Piscataway, NJ, 1996. IEEE Press.
63. J. Gertler, W. Li, Y. Huang, and T. McAvoy. Isolation enhanced principal component analysis. *AIChE J.*, 45(2):323–334, 1999.
64. J. J. Gertler. Analytical redundancy methods in fault detection and isolation. In *Proc. of IFAC Safeprocess Symp.*, pages 9–21, Oxford, U.K., 1991. Pergamon Press.
65. M. Gevers and V. Wertz. On the problem of structure selection for the identification of stationary stochastic processes. In *Sixth IFAC Symposium on Identification and System Parameter Estimation*, pages 387–392, Piscataway, NJ, 1982. IEEE Press.
66. G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.
67. E. Gomez, H. Unbehauen, P. Kortmann, and S. Peters. Fault detection and diagnosis with the help of fuzzy-logic and with application to a laboratory turbogenerator. In *Proc. of the 13th IFAC World Congress*, volume N, pages 235–240, Piscataway, NJ, 1996. IEEE Press.
68. J. B. Gomm. On-line learning for fault classification using an adaptive neuro-fuzzy network. In *Proc. of the 13th IFAC World Congress*, volume N, pages 175–180, Piscataway, NJ, 1996. IEEE Press.
69. F. Hamelin and D. Sauter. Robust residual generation for F.D.I. in uncertain dynamic systems. In *Proc. of the 13th IFAC World Congress*, volume N, pages 181–186, Piscataway, NJ, 1996. IEEE Press.
70. D. Hanselman and B. Littlefield. *The Student Edition of MATLAB: Version 5, User's Guide*. Prentice Hall, New Jersey, 1997.
71. D. Hanselman and B. Littlefield. *Mastering MATLAB 5, A Comprehensive Tutorial and Reference*. Prentice Hall, New Jersey, 1998.
72. T. J. Harris. Assessment of control loop performance. *Can. J. of Chem. Eng.*, 67:856–861, 1989.
73. T. J. Harris and W. H. Ross. Statistical process control procedures for correlated observations. *Can. J. of Chem. Eng.*, 69:48–57, 1991.
74. I. Hashimoto, M. Kano, and K. Nagao. A new method for process monitoring using principal component analysis. In *AIChE Annual Meeting*, 1999. Paper 224a.
75. D. M. Hawkins. Multivariate quality control based on regression-adjusted variables. *Technometrics*, 33:61–67, 1991.

76. J. D. Healy. A note of multivariate CUSUM procedures. *Technometrics*, 29:409–412, 1987.
77. D. M. Himes, R. H. Storer, and C. Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *Proc. of the American Control Conf.*, pages 1279–1283, Piscataway, NJ, 1994. IEEE Press.
78. D. M. Himmelblau. *Fault Detection and Diagnosis in Chemical and Petrochemical Processes.* Elsevier Scientific Publishing Co., New York, 1978.
79. D. M. Himmelblau. Use of artificial neural networks to monitor faults and for troubleshooting in the process industries. In *IFAC Symposium On-line Fault Detection and Supervision in the Chemical Process Industry*, Oxford, U.K., 1992. Pergamon Press.
80. D. M. Himmelblau, R. W. Braker, and W. Suewatanakul. Fault classification with the aid of artificial neural networks. In *IFAC/IMAC Symposium Safeprocess*, pages 369–373, Oxford, U.K., 1991. Pergamon Press.
81. W. W. Hines and D. C. Montgomery. *Probability and Statistics in Engineering and Management Science.* John Wiley & Sons, New York, 3rd edition, 1990.
82. T. Hofling and R. Isermann. Adaptive parity equations and advanced parameter estimation for fault detection and diagnosis. In *Proc. of the 13th IFAC World Congress*, pages 55–60, Piscataway, NJ, 1996. IEEE Press.
83. T. Holcomb and M. Morari. PLS/neural networks. *Comp. & Chem. Eng.*, 16(4):393–411, 1992.
84. Z. Q. Hong and J. Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24:317–324, 1991.
85. H. Hotelling. Relations between two sets of variables. *Biometrika*, 26(4):321–377, 1936.
86. H. Hotelling. Multivariate quality control. In Eisenhart, Hastay, and Wallis, editors, *Techniques of Statistical Analysis.* McGraw Hill, New York, 1947.
87. B. N. Huallpa, E. Nobrega, and F. J. V. Zuben. Fault detection in dynamic systems based on fuzzy diagnosis. In *Proc. of the IEEE International Conference on Fuzzy Systems*, volume 2, pages 1482–1487, Piscataway, NJ, 1998.
88. R. Hudlet and R. Johnson. Linear discrimination and some further results on best lower dimensional representations. In J. V. Ryzin, editor, *Classification and Clustering*, pages 371–394. Academic Press, New York, 1977.
89. IMSL. Visual Numerics, Inc., 1997. computer software.
90. M. Iri, K. Aoki, E. O'Shima, and H. Matsuyama. An algorithm for diagnosis of system failures in the chemical process. *Comp. & Chem. Eng.*, 3:489–493, 1979.
91. R. Isermann. Process fault detection based on modeling and estimation methods: a survey. *Automatica*, 20:387–404, 1984.
92. R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing - tutorial paper. *Automatica*, 29:815–835, 1993.
93. R. Isermann. Integration of fault detection and diagnosis methods. In *Fault Detection, Supervision, and Safety for Technical Processes: IFAC Symposium*, Oxford, U.K., 1994. Pergamon Press.
94. R. Isermann. Model based fault detection and diagnosis methods. In *Proc. of the American Control Conf.*, pages 1605–1609, Piscataway, NJ, 1995. IEEE Press.
95. R. Isermann and P. Ball. Trends in the application of model based fault detection and diagnosis of technical processes. In *Proc. of the 13th IFAC World Congress*, volume N, pages 1–12, Piscataway, NJ, 1996. IEEE Press.

96. R. Isermann and B. Freyermuth. Process fault diagnosis based on process model knowledge — part I: principles for fault diagnosis with parameter estimation. *ASME J. of Dynamics, Measurement, and Control*, 113:620–626, 1991.
97. R. Isermann and B. Freyermuth. Process fault diagnosis based on process model knowledge — part II: case study experiments. *ASME J. of Dynamics, Measurement, and Control*, 113:627–633, 1991.
98. J. E. Jackson. Quality control methods for two related variables. *Industrial Quality Control*, 7:2–6, 1956.
99. J. E. Jackson. Quality control methods for several related variables. *Technometrics*, 1:359–377, 1959.
100. J. E. Jackson. *A User's Guide to Principal Components*. John Wiley & Sons, New York, 1991.
101. J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341–349, 1979.
102. E. W. Jacobsen. *Studies on Dynamics and Control of Distillation Columns*. PhD thesis, University of Trondheim, Trondheim, Norway, 1991.
103. R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, New Jersey, 3rd edition, 1992.
104. T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
105. M. H. Kaspar and W. H. Ray. Chemometric methods for process monitoring and high-performance controller design. *AIChE J.*, 38:1593–1608, 1992.
106. M. H. Kaspar and W. H. Ray. Dynamic PLS modeling for process control. *Chem. Eng. Sci.*, 48:3447–3461, 1993.
107. A. H. Kemna, W. E. Larimore, D. E. Seborg, and D. A. Mellichamp. On-line multivariable identification and control chemical processes using canonical variate analysis. In *Proc. of the American Control Conf.*, pages 1650–1654, Piscataway, New Jersey, 1994. IEEE Press.
108. E. K. Kemsley. Discriminant analysis of high-dimensional data: a comparison of principal components analysis and partial least squares data reduction methods. *Chemometrics and Intelligent Laboratory Systems*, 33:47–61, 1996.
109. P. Kesavan and J. H. Lee. Diagnostic tools for multivariable model-based control systems. *Ind. Eng. Chem. Res.*, 36:2725–2738, 1997.
110. B. Koppen-Seliger and P. M. Frank. Neural networks in model-based fault diagnosis. In *Proc. of the 13th IFAC World Congress*, pages 67–72, Piscataway, NJ, 1996. IEEE Press.
111. K. A. Kosanovich, M. J. Piovoso, K. S. Dahl, J. F. MacGregor, and P. Nomikos. Multi-way PCA applied to an industrial batch process. In *Proc. of the American Control Conf.*, pages 1294–1298, Piscataway, NJ, 1994. IEEE Press.
112. T. Kourti and J. F. MacGregor. Process analysis, monitoring and diagnosis using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.
113. T. Kourti and J. F. MacGregor. Multivariate SPC methods for process and product monitoring. *J. of Quality Technology*, 28:409–428, 1996.
114. M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.*, 37:233–243, 1991.
115. M. A. Kramer and J. B. L. Palowitch. A rule-based approach to fault diagnosis using the signed directed graph. *AIChE J.*, 33:1067–1078, 1987.
116. M. A. Kramer and F. E. Finch. Development and classification of expert systems for chemical process fault diagnosis. *Robotics and Computer-integrated Manufacturing*, 4(3/4):4376–446, 1988.

117. M. A. Kramer and R. Fjellheim. Fault diagnosis and computer-aided diagnostic advisors. In *International Conference on Intelligent Systems in Process Engineering, AIChE Symposium Series*, volume 92, pages 12–24, 1996.
118. F. Kratz, W. Nuninger, and S. Ploix. Fault detection for time-delay systems: a parity space approach. In *Proc. of the American Control Conf.*, pages 2009–2011, Piscataway, NJ, 1998. IEEE Press.
119. J. Kresta, J. F. MacGregor, and T. Marlin. Multivariate statistical monitoring of process operating performance. *Can. J. Chem. Eng.*, 69(35):35–47, 1991.
120. J. V. Kresta, T. E. Marlin, and J. F. MacGregor. Multivariable statistical monitoring of process operating performance. *Can. J. of Chem. Eng.*, 69:35–47, 1991.
121. W. J. Krzanowski. Between-group comparison of principal components. *J. Amer. Stat. Assn.*, 74:703–706, 1979.
122. A. M. Kshirsagar. *Multivariate Analysis.* Marcel Dekker, New York, 1972.
123. W. Ku, R. H. Storer, and C. Georgakis. Isolation of disturbances in statistical process control by use of approximate models. In *AIChE Annual Meeting*, 1993. Paper 149g.
124. W. Ku, R. H. Storer, and C. Georgakis. Uses of state estimation for statistical process control. *Comp. & Chem. Eng.*, 18:S571–S575, 1994.
125. W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30:179–196, 1995.
126. W. E. Larimore. System identification, reduced-order filtering and modeling via canonical variate analysis. In *Proc. of the American Control Conf.*, pages 445–451, Piscataway, New Jersey, 1983. IEEE Press.
127. W. E. Larimore. Canonical variate analysis for system identification, filtering, and adaptive control. In *Proc. of the IEEE Conf. on Decision and Control*, pages 635–639, Piscataway, NJ, 1990. IEEE Press.
128. W. E. Larimore. Identification and filtering of nonlinear systems using canonical variate analysis. In *Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity.* Addison-Wesley, Reading, MA, 1992.
129. W. E. Larimore. *ADAPTx Automated System Identification Software Users Manual.* Adaptics, Inc., McLean, VA, 1996.
130. W. E. Larimore. Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52:131–144, 1996.
131. W. E. Larimore. Canonical variate analysis in control and signal processing. In *Statistical Methods in Control and Signal Processing.* Marcel Dekker, New York, 1997.
132. W. E. Larimore. Optimal reduced rank modeling, prediction, monitoring, and control using canonical variate analysis. In *IFAC ADCHEM*, Alberta, Canada, June 1997.
133. W. E. Larimore and D. E. Seborg. *Short Course: Process Monitoring and Identification of Dynamic Systems Using Statistical Techniques.* Los Angeles, CA, 1997.
134. B. K. Lavine. Chemometrics. *Anal. Chem.*, 70:209R–228R, 1998.
135. L. Lebart, A. Morineau, and K. M. Warwick. *Multivariate Descriptive Statistical Analysis.* John Wiley & Sons, New York, 1984.
136. S. C. Lee. Sensor value validation based on systematic exploration of the sensor redundancy for fault diagnosis. *IEEE Trans. Sys. Man Cyber.*, 24:594–605, 1994.
137. L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

138. X. Lou, A. S. Willsky, and G. C. Verghese. Optimally robust redundancy relations for failure detection in uncertain systems. *Automatica*, 22:333–344, 1986.
139. C. A. Lowry and W. H. Woodall. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34:46–53, 1992.
140. C. A. Lowry, W. H. Woodall, C. W. Champ, and S. E. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34:46–53, 1992.
141. P. R. Lyman. *Plant-wide Control Structures for the Tennessee Eastman process*, M.S. thesis, Lehigh University, 1992.
142. P. R. Lyman and C. Georgakis. Plant-wide control of the Tennessee Eastman problem. *Comp. & Chem. Eng.*, 19:321–331, 1995.
143. D. L. Ma, S. H. Chung, and R. D. Braatz. Worst-case performance analysis of optimal batch control trajectories. *AIChE J.*, 45:1469–1476, 1999.
144. J. F. MacGregor. Statistical process control of multivariate processes. In *Proc. of the IFAC Conference on Advanced Control of Chemical Processes*, pages 427–435, New York, 1994. Pergamon Press.
145. J. F. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi. Process monitoring and diagnosis by multiblock PLS methods. *AIChE J.*, 40:826–838, 1994.
146. J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995.
147. E. Malinowski. Statistical F-test for abstract factor analysis and target testing. *J. Chemometrics*, 3:46, 1989.
148. E. C. Malthouse, A. C. Tamhane, and R. S. H. Mah. Nonlinear partial least squares. *Comp. & Chem. Eng.*, 21:875–890, 1997.
149. R. Manne. Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2:187–197, 1987.
150. H. Martens and T. Naes. *Multivariate Calibration.* John Wiley & Sons, 1989.
151. T. J. McAvoy. A methodology for screening level control structures in plantwide control systems. *Comp. & Chem. Eng.*, 22:1543–1552, 1998.
152. T. J. McAvoy, Y. Nan, and G. Chan. An improved base control for the Tennessee Eastman problem. In *Proc. of the American Control Conf.*, pages 240–244, Piscataway, NJ, 1995. IEEE Press.
153. A. Medvedev. State estimation and fault detection by a bank of continuous finite-memory filters. In *Proc. of the 13th IFAC World Congress*, volume N, pages 223–228, Piscataway, NJ, 1996. IEEE Press.
154. R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–640, 1971.
155. T. Mejdell and S. Skogestad. Estimation of distillation compositions from multiple temperature measurements using partial least squares regression. *Ind. Eng. Chem. Res.*, 30:2543–2555, 1991.
156. T. Mejdell and S. Skogestad. Output estimation using multiple secondary measurements: high-purity distillation. *AIChE J.*, 39:1641–1653, 1993.
157. P. Miller, R. E. Swanson, and C. E. Heckler. Contribution plots: a missing link in multivariate quality control. *Applied Mathematics & Computer Science*, 8(4):775–792, 1998.
158. K. J. Mo, Y. S. Oh, and E. S. Yoon. Development of operation-aided system for chemical processes. *Expert Systems with Applications*, 12(4):455–464, 1997.
159. D. C. Montgomery. *Introduction to Statistical Quality Control.* John Wiley and Sons, New York, 1985.

160. O. Moseler and R. Isermann. Model-based fault detection for a brushless dc motor using parameter estimation. In *Proc. of the 24th Annual Conference of the IEEE Industrial Electronics Society, IECON.*, volume 4, pages 956–1960, Piscataway, NJ, 1998.
161. R. J. Muirhead. *Aspects of Multivariate Statistical Theory.* John Wiley & Sons, New York, NY, 1982.
162. D. Mylaraswamy and V. Venkatasubramanian. A hybrid framework for large scale process fault diagnosis. In *Joint 6th International Symposium on Process System Engineering and 30th European Symposium on Computer Aided Process Engineering*, pages S935–S940, Oxford, U.K., 1997. Elsevier Science Ltd.
163. Y. Nan, T. J. McAvoy, K. A. Kosanovich, and M. J. Piovoso. Plant-wide control using an inferential approach. In *Proc. of the American Control Conf.*, pages 1900–1904, Piscataway, NJ, 1993. IEEE Press.
164. A. Negiz and A. Cinar. On the detection of multiple sensor abnormalities in multivariate processes. In *Proc. of the American Control Conf.*, pages 2364–2368, Piscataway, New Jersey, 1992. IEEE Press.
165. A. Negiz and A. Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE J.*, 43:2002–2020, 1997.
166. A. Negiz and A. Cinar. Monitoring of multivariable dynamic processes and sensor auditing. *J. of Process Control*, 8:375–380, 1998.
167. C. N. Nett, C. A. Jacobson, and A. T. Miller. An integrated approach to controls and diagnostics: the 4-parameter controller. In *Proc. of the American Control Conf.*, pages 824–835, Piscataway, New Jersey, 1988. IEEE Press.
168. I. Nikiforov, M. Staroswiecki, and B. Vozel. Duality of analytical redundancy and statistical approach in fault diagnosis. In *Proc. of the 13th IFAC World Congress*, volume N, pages 19–24, Piscataway, NJ, 1996. IEEE Press.
169. P. Nomikos and J. F. MacGregor. Monitoring batch processes using multiway principal component analysis. *AIChE J.*, 40:1361–1375, 1994.
170. J. Nouwen, F. Lindgren, W. K. B. Hansen, H. J. M. Verharr, and J. L. M. Hermens. Classification of environmentally occurring chemicals using structural fragments and PLS discriminant analysis. *Environ. Sci. Technol.*, 31:2313–2318, 1997.
171. B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control.* Oxford University Press, New York, 1994.
172. Y. S. Oh, J. H. Yoon, D. Nam, C. Han, and E. S. Yoon. Intelligent fault diagnosis based on weighted symptom tree model and fault propagation trends. In *Joint 6th International Symposium on Process System Engineering and 30th European Symposium on Computer Aided Process Engineering*, pages S941–S946, Oxford, U.K., 1997. Elsevier Science Ltd.
173. O. O. Oyeleye, F. E. Finch, and M. A. Kramer. Qualitative modeling and fault diagnosis of dynamic processes by MIDAS. *Chem. Eng. Comm.*, 96:205–228, 1990.
174. R. J. Patton and J. Chen. Optimal unknown input disturbance matrix selection in robust fault diagnosis. *Automatica*, 29:837–841, 1993.
175. R. J. Patton, J. Chen, and T. M. Siew. Fault diagnosis in non-linear dynamic systems via neural-networks. In *Proc. of the International Conference on CONTROL*, volume 2, pages 1346–1351, Stevenage, U.K., 1994. IEE Press.
176. R. J. Patton and M. Hou. A matrix pencil approach to fault detection and isolation observers. In *Proc. of the 13th IFAC World Congress*, volume N, pages 229–234, Piscataway, NJ, 1996. IEEE Press.
177. R. J. Patton and M. Hou. Design of fault detection and isolation observers: A matrix pencil approach. *Automatica*, 43(9):1135–1140, 1998.

178. R. K. Pearson. Data cleaning for dynamic modeling and control. In *Proc. of the European Control Conf.*, Karlsruhe, Germany, 1999. IFAC. Paper F853.
179. D. W. Peterson and R. L. Mattson. A method of finding linear discriminant functions for a class of performance criteria. *IEEE Trans. Info. Theory*, 12:380–387, 1966.
180. J. J. Pignatiello, Jr. and G. C. Runger. Comparisons of multivariate CUSUM charts. *J. of Quality Technology*, 22:173–186, 1990.
181. M. J. Piovoso and K. A. Kosanovich. Applications of multivariate statistical methods to process monitoring and controller design. *Int. J. of Control*, 59:743–765, 1994.
182. M. J. Piovoso, K. A. Kosanovich, and R. K. Pearson. Monitoring process performance in real time. In *Proc. of the American Control Conf.*, pages 2359–2363, Piscataway, New Jersey, 1992. IEEE Press.
183. B. L. S. Prakasa Rao. *Identifiability in Stochastic Models Characterization of Probability Distributoins.* Academic Press, New York, 1992.
184. S. J. Qin. Recursive PLS algorithms for adaptive data modeling. *Comp. & Chem. Eng.*, 22:503–514, 1998.
185. S. J. Qin and T. J. McAvoy. Nonlinear PLS modeling using neural networks. *Comp. & Chem. Eng.*, 16:379–391, 1992.
186. A. C. Raich and A. Cinar. Statistical process monitoring and disturbance isolation in multivariate continuous processes. In *Proc. of the IFAC Conf. on Advanced Control of Chemical Processes*, pages 427–435, New York, 1994. Pergamon Press.
187. A. C. Raich and A. Cinar. Multivariate statistical methods for monitoring continuous processes: assessment of discriminatory power disturbance models and diagnosis of multiple disturbances. *Chemometrics and Intelligent Laboratory Systems*, 30:37–48, 1995.
188. A. C. Raich and A. Cinar. Process disturbance diagnosis by statistical distance and angle measures. In *Proc. of the 13th IFAC World Congress*, pages 283–288, Piscataway, NJ, 1996. IEEE Press.
189. A. C. Raich and A. Cinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE J.*, 42:995–1009, 1996.
190. R. R. Rhinehart. A watchdog for controller performance monitoring. In *Proc. of the American Control Conf.*, pages 2239–2240, Piscataway, New Jersey, 1995. IEEE Press.
191. E. L. Russell and R. D. Braatz. The average-case identifiability and controllability of large scale systems. *J. of Process Control.* in press.
192. E. L. Russell and R. D. Braatz. Fault isolation in industrial processes using Fisher discriminant analysis. In J. F. Pekny and G. E. Blau, editors, *Foundations of Computer-Aided Process Operations*, pages 380–385. AIChE, New York, 1998.
193. E. L. Russell and R. D. Braatz. Model reduction for the robustness margin computation of large scale uncertain systems. *Comp. & Chem. Eng.*, 22:913–926, 1998.
194. E. L. Russell, C. P. H. Power, and R. D. Braatz. Multidimensional realizations of large scale uncertain systems for multivariable stability margin computation. *Int. J. of Robust and Nonlinear Control*, 7:113–125, 1997.
195. C. D. Schaper, W. E. Larimore, D. E. Seborg, and D. A. Mellichamp. Identification of chemical processes using canonical variate analysis. *Comp. & Chem. Eng.*, 18:55–69, 1994.
196. D. Shields. Quantitative approaches for fault diagnosis based on bilinear systems. In *Proc. of the 13th IFAC World Congress*, volume N, pages 151–155, Piscataway, NJ, 1996. IEEE Press.

197. J. Shiozaki, H. Matsuyama, E. O'shima, and M. Iri. An improved algorithm for diagnosis of system failures in the chemical process. *Comp. & Chem. Eng.*, 9(3):285–293, 1985.
198. O. Simula, E. Alhoniemi, J. Hollmen, and J. Vesanto. Monitoring and modeling of complex processes using hierarchical self-organizing maps. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 73–76, Piscataway, NJ, 1996. IEEE Press.
199. O. Simula and J. Kangas. Process monitoring and visualization using self-organizing maps. In *Neural Networks for Chemical Engineers, Computer-Aided Chemical Engineering*, chapter 14, pages 371–384. Elsevier, Amsterdam, 1995.
200. R. S. Spraks. Quality control with multivariate data. *Australian Journal of Statistics*, 34:375–390, 1992.
201. N. Stanfelj, T. E. Marlin, and J. F. MacGregor. Monitoring and diagnosing process control performance: the single loop case. *Ind. Eng. Chem. Res.*, 32:301–314, 1993.
202. A. K. Sunol, B. Ozyurt, P. K. Mogili, and L. Hall. A machine learning approach to design and fault diagnosis. In *International Conference on Intelligent Systems in Process Engineering, AIChE Symposium Series*, volume 92, pages 331–334, 1996.
203. E. Sutanto and K. Warwick. Cluster analysis for multivariate process control. In *Proc. of the American Control Conf.*, pages 749–751, Piscataway, NJ, 1995. IEEE Press.
204. B. G. Tabachnick and L. S. Fidell. *Using Multivariate Analysis.* Harper & Row, Cambridge, 1989.
205. E. E. Tarifa and N. J. Scenna. Fault diagnosis, direct graphs, and fuzzy logic. In *Joint 6th International Symposium on Process System Engineering and 30th European Symposium on Computer Aided Process Engineering*, pages S649–S654, Oxford, U.K., 1997. Elsevier Science Ltd.
206. E. E. Tarifa and N. J. Scenna. Methodology for fault diagnosis in large chemical processes and an application to a multistage flash desalination process: Part ii. *Reliability Engineering & System Safety*, 60(1):41–51, 1998.
207. Q. Tian. Comparison of statistical pattern-recognition algorithms for hybrid processing, II: eigenvector-based algorithm. *J. Opt. Soc. Am. A*, 5:1670–1672, 1988.
208. H. Tong and C. M. Crowe. Detection of gross errors in data reconciliation by principal component analysis. *AIChE Journal*, 41(7):1712–1722, 1995.
209. N. D. Tracy, J. C. Young, and R. L. Mason. Multivariate control charts for individual observations. *J. of Quality Control*, 24:88–95, 1992.
210. Y. Tsuge, J. Shiozaki, H. Matsuyama, and E. O'Shima. Fault diagnosis algorithms based on the signed directed graph and its modifications. *Ind. Chem. Eng. Symp. Ser.*, 92:133–144, 1985.
211. M. L. Tyler and M. Morari. Optimal and robust design of integrated control and diagnostic modules. In *Proc. of the American Control Conf.*, pages 2060–2064, Piscataway, NJ, 1994. IEEE Press.
212. M. L. Tyler and M. Morari. Performance monitoring of control systems using likelihood methods. *Automatica*, 32:1145–1162, 1996.
213. S. G. Tzafestas and P. J. Dalianis. Fault diagnosis in complex systems using artificial neural networks. In *Proc. of The Third IEEE Conference on Control Applications*, pages 877–882, Piscataway, NJ, 1994. IEEE Press.
214. C. V. Van Loan. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.*, 13:76–83, 1976.

215. P. Van Overschee and B. De Moor. N4SID*: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.
216. P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory - Implementation - Applications.* Kluwer Academic Publishers, Norwell, MA, 1996.
217. J. G. VanAntwerp and R. D. Braatz. A tutorial on linear and bilinear matrix inequalities. *J. of Process Control.* in press.
218. J. G. VanAntwerp, R. D. Braatz, and N. V. Sahinidis. Globally optimal robust process control. *J. of Process Control*, 9:375–383, 1999.
219. H. Vedam and V. Venkatasubramanian. Signed digraph based multiple fault diagnosis. In S. Skogestad, editor, *Joint 6th International Symposium on Process System Engineering and 30th European Symposium on Computer Aided Process Engineering*, pages S655–S660, Oxford, U.K., 1997. Elsevier Science Ltd.
220. D. R. Vinson, C. Georgakis, and J. Fossy. Studies in plant-wide controllability using the tennessee eastman challenge problem, the case for multivariable control. In *Proc. of the American Control Conf.*, pages 250–254, Piscataway, NJ, 1995. IEEE Press.
221. N. Viswanadham and K. D. Minto. Robust observer design with application to fault detection. In *Proc. of the American Control Conf.*, pages 1393–1399, Piscataway, NJ, 1988. IEEE Press.
222. E. Walter. *Identifiability of Parametric Models.* Pergamon Press, New York, 1987.
223. Y. Wang, D. E. Seborg, and W. E. Larimore. Process monitoring using canonical variate analysis and principal component analysis. In *IFAC ADCHEM*, Alberta, Canada, June 1997.
224. W. W. S. Wei. *Time Series Analysis.* Addison-Wesley, Reading, Massachusetts, 1994.
225. S. J. Wierda. Multivariate statistical process control, recent results and directions for future research. *Statistica Neerlandica*, 48:147–168, 1994.
226. N. A. Wilcox and D. M. Himmelblau. The possible cause-effect graph model for process fault diagnosis -i. methodology. *Comp. & Chem. Eng.*, 18(2):103–116, 1994.
227. A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.
228. B. M. Wise and N. B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *J. of Process Control*, 6:329–348, 1996.
229. B. M. Wise and N. B. Gallagher. PLS_Toolbox 2.0 for use with Matlab. Software manual, Eigenvector Research, Manson, WA, 1998.
230. B. M. Wise, N. L. Ricker, D. J. Velkamp, and B. R. Kowalski. A theoretical basis for the use of principal component models for monitoring multivariate processes. *Process Control and Quality*, 1:41–51, 1990.
231. B. M. Wise, N. L. Ricker, and D. F. Veltkamp. Upset and sensor failure detection in multivariate processes. Technical report, Eigenvector Research, Manson, WA, 1989.
232. S. Wold. Cross-validatory estimation of components in factor and principal components models. *Technometrics*, 20:397–405, 1978.
233. S. Wold, K. Esbensen, and P. Geladi. Principal components analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37, 1987.
234. S. Wold, N. Kettaneh-Wold, and B. Skagerberg. Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 7:53–65, 1989.
235. S. Wold, H. Martens, and H. Russwurm. *Food Research and Data Analysis.* Applied Science Publishers, London, 1983.

236. W. H. Woodall and M. M. Ncube. Multivariate CUSUM quality-control procedures. *Technometrics*, 27:285–292, 1985.
237. E. S. Yoon and J. H. Han. Process failure detection and diagnosis using the tree model. In *Proc. of the IFAC World Congress*, pages 126–129, Oxford, U.K., 1987. Pergamon Press.
238. J. Zhang, E. Martin, and A. J. Morris. Fault detection and classification through multivariate statistical techniques. In *Proc. of the American Control Conf.*, pages 751–755, Piscataway, NJ, 1995. IEEE Press.
239. J. Zhang, A. J. Morris, and E. B. Martin. Robust process fault detection and diagnosis using neuro-fuzzy networks. In *Proc. of the 13th IFAC World Congress*, volume N, pages 169–174, Piscataway, NJ, 1996. IEEE Press.
240. Q. Zhang. A frequency and knowledge tree/causality diagram based expert system approach for fault diagnosis. *Reliability Engineering & System Safety*, 43:17–28, 1994.
241. Q. Zhang, X. An, J. Gu, B. Zhao, D. Xu, and S. Xi. Application of FBOLES - a prototype expert system for fault diagnosis in nuclear power plants. *Reliability Engineering & System Safety*, 44:225–235, 1994.
242. Y. Zhang, X. Li, G. Dai, H. Zhang, and H. Chen. Fault detection and identification of dynamic systems using multiple feedforward neural networks. In *Proc. of the 13th IFAC World Congress*, volume N, pages 241–246, Piscataway, NJ, 1996. IEEE Press.
243. J. Zhao, B. Chen, and J. Shen. A hybrid ANN-ES system for dynamic fault diagnosis of hydrocracking process. In *Joint 6th International Symposium on Process System Engineering and 30th European Symposium on Computer Aided Process Engineering*, pages S929–S933, Oxford, U.K., 1997. Elsevier Science Ltd.
244. A. Zolghadri. Model based fault detection in a multivariable hydraulic process. In *Proc. of the 13th IFAC World Congress*, volume N, pages 253–258, Piscataway, NJ, 1996. IEEE Press.

INDEX

Adjusted PLS1, *see* $PLS1_{adj}$
Adjusted PLS2, *see* $PLS2_{adj}$
Analytical-based approaches, 6
- analytical redundancy, 170
- parameter and state estimation, 169
ARMA, 81
Artificial neural networks, 173
ARX, 50
- comparison with DPCA, 50
Autoregressive model, *see* ARX
Autoregressive Moving Average model, *see* ARMA
Autoscaling, 14

Between class-scatter-matrix, 54

Canonical Correlation Analysis, 84
Canonical correlations, 83
Canonical variables, 84
Canonical Variate Analysis, *see* CVA
Combined discriminant, 47
Common cause, 13
Contribution plots
- CVA, 94
- PCA, 43
- PLS, 74
Cumulative sum chart, *see* CUSUM
CUSUM, 17, 51, 114
CVA
- Akaike's information criterion, 90
- algorithm, 85
- canonical correlations, 83
- canonical variables, 84
- comparison with discriminant PLS, 85
- comparison with DPCA, 81
- comparison with FDA, 85
- comparison with PCA, 84
- fault detection, 130
- fault diagnosis, 94
- fault identification, 94, 139
- identifiability, 88
- information criterion, 90
- Q statistic, 94
- SVD, 84
- T^2 statistic, 93
- Theorem, 83

Data-driven approaches, 6
DFDA, 64
- fault diagnosis, 146
Dimensionality reduction, 29
Discrepancy detection, 7
Discriminant analysis, 25, 26
- discriminant PLS, 74
Discriminant function, 26, 28, 46, 59
Discriminant Partial Least Squares, *see* Discriminant PLS
Discriminant PLS, 9
- comparison with CVA, 85
- comparison with DPCA, 75
- dummy variables, 68
- fault diagnosis, 148
- prediction, 73
- reduction order, 73
Discriminant Projection to Latent Structures, *see* Discriminant PLS
DPCA
- comparison with ARX, 50
- comparison with CVA, 81
- fault detection, 136
- fault diagnosis, 153
- fault identification, 50, 139
Dynamic Fisher Discriminant Analysis, *see* DFDA
Dynamic Principal Component Analysis, *see* DPCA

Eigenvalue decomposition
- FDA, 55
- PCA, 34
- T^2 statistic, 19
EWMA, 17, 51, 114
Expert systems, 172

Exponentially-weighted moving average, *see* EWMA

False alarm, 15
Fault detection, 4
- CVA, 130
- DPCA, 50
- FDA, 58
- PCA, 39
- PLS, 74
Fault diagnosis, 4
- CVA, 94
- DFDA, 146
- DPCA, 153
- FDA, 58, 143
- PCA, 45, 153
- $PLS1_{adj}$, 74, 148
- $PLS2_{adj}$, 74, 148
- PLS1, 74, 148
- PLS2, 74, 148
Fault identification, 4
- CVA, 94, 139
- DPCA, 50, 139
- PCA, 42, 139
- PLS, 74
- univariate statistic, 42
FDA, 8, 53
- Akaike's information criterion, 56
- between class-scatter-matrix, 54
- comparison with CVA, 85
- comparison with PCA, 59
- eigenvalue decomposition, 55
- fault diagnosis, 58, 143
- FDA/PCA1, 59
- FDA/PCA2, 59
- optimization, 55
- reduction order, 56
- total-scatter matrix, 54
- within-class-scatter matrix, 54
Feature extraction, 25, 28
Fisher Discriminant Analysis, *see* FDA

Generalized singular value decomposition, *see* GSVD
GSVD, 84

Identifiability, 88
Information criterion
- CVA, 90
- DFDA, 147
- discriminant PLS, 149
- FDA, 56, 147

KLIB, 90

Knowledge-based approaches, 6
- causal analysis, 171
- pattern recognition, 173
Kullback-Leibler information distance, *see* KLIB

Limit sensing, 7, 15
Limit value checking, 15
Loading vectors, 34, 69

Markov process, 86
Maximum selection, 25
Mean overlap, 49
Missed detection, 15
MOESP, 91
Multivariate Statistics, *see* MS

N4SID, 91
NIPALS
- PLS1, 72
- PLS2, 70
Non-Iterative Partial Least Squares, *see* NIPALS
Non-supervised classification, 45

Observability, 134
Ordinary Least Squares, 69

Parallel analysis, 38
Partial Least Squares, *see* PLS
Pattern classification
- discriminant analysis, 25, 26
- feature extraction, 25, 28
- maximum selection, 25
PCA, 8
- application, 33
- combined discriminant, 47
- comparison with FDA, 59
- comparison with CVA, 84
- comparison with discriminant PLS, 75
- fault detection, 39
- fault diagnosis, 45, 153
- fault identification, 42, 139
- multiway, 51
- nonlinear, 51
- optimization problem, 34
- parallel analysis, 38
- percent variance method, 37
- PRESS statistic, 39
- properties, 35
- Q statistic, 41
- reduction order, 37
- residual discriminant, 47

- residual matrix, 35
- score discriminant, 46
- scree test, 38
- SPE, 41
- SVD representation, 34
- T^2 statistic, 39
Percent variance method, 37
PLS
- loading vectors, 69
- multiblock, 79
- multiway, 79
- NIPALS algorithm, 72
- nonlinear, 78
- $PLS1_{adj}$, 74, 148
- $PLS2_{adj}$, 74, 148
- PLS1, 70
- PLS2, 68
- prediction, 73
- score matrix, 69
- score vectors, 69
Prediction Sum of Squares statistic, *see* PRESS statistic
PRESS statistic, 39, 73
Principal Component Analysis, *see* PCA
Process monitoring
- analytical, 6, 169, 170
- data-driven, 6
- discrepancy detection, 7
- knowledge-based, 6, 171, 173
- limit sensing, 7
- methods, 5, 7
- multivariate statistic, 19
- objective, 6
- procedure, 4
- univariate statistic, 15
Process recovery, 4
Promptness of statistics, 129

Q statistic
- CVA, 94
- PCA, 41
- PLS, 74

Reduction order
- discriminant PLS, 73
- FDA, 56
- PCA, 37
Removing outliers, 14
Removing variables, 14
Residual discriminant, 47
Residual vector
- CVA, 94
- PCA, 35
Robustness of statistics, 129

Score discriminant, 46
Score matrix, 69
Score vectors, 69
Scree test, 38
Sensitivity of statistics, 129
Serial correlation, 7, 49, 64, 77, 111, 129
Shewhart chart, 15
Signed directed graph, 171
Similarity index, 48
Singular Value Decomposition, *see* SVD
Spacial correlation, 43
SPE, 41
Special cause, 13
Squared prediction error, *see* SPE
State equation, 85
Statistical process control, *see* Process monitoring
Subspace algorithm, 82
Supervised classification, 45, 142
SVD
- CVA, 84
- PCA, 34
Symptom tree model, 172
System identification theory, 29

T^2 statistic, 19
- CVA, 93
- eigenvalue decomposition, 19
- MS, 19
- PCA, 39
- threshold, 20
Tennessee Eastman Process, *see* TEP
TEP
- controller parameters, 105
- faults, 100
- manipulated variables, 100
- process variables, 100
Threshold
- Q statistic, 41
- T^2 statistic, 20, 40, 93
- univariate statistic, 15
Total-scatter matrix, 54
Type I error, 16
Type II error, 16

Univariate statistic, 15
- CUSUM, 17
- EWMA, 17
- fault identification, 42

– Shewhart chart, 15
– threshold, 15
Within-class-scatter matrix, 54