Multivariate Statistical Process Control for Fault Detection and Diagnosis

George Stefatos

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Quality Systems) at
Concordia University
Montreal, Quebec, Canada

June 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-34461-3
Our file Notre référence
ISBN: 978-0-494-34461-3

Abstract

Multivariate Statistical Process Control for Fault Detection and Diagnosis

George Stefatos

The great challenge in quality control and process management is to devise computationally efficient algorithms to detect and diagnose faults. Currently, univariate statistical process control is an integral part of basic quality management and quality assurance practices used in the industry. Unfortunately, most data and process variables are inherently multivariate and need to be modelled accordingly. Major barriers such as higher complexity and harder interpretation have limited their application by both engineers and operators. Motivated by the lack of techniques dedicated in monitoring highly correlated data, we introduce in this thesis new multivariate statistical process control charts using robust statistics, machine learning, and pattern recognition techniques to propose our algorithms. The core idea behind our proposed techniques is to fully explore the advantages/limitations under a wide array of environments, and to also take advantage of the latter to develop a theoretically rigorous and computationally feasible methodology for multivariate statistical process control. Illustrating experimental results demonstrate a much improved performance of the proposed approaches in comparison with existing methods currently used in the analysis and monitoring of multivariate data.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. A. Ben Hamza, for his continuous support and encouragement throughout my graduate studies. His great help is essential to the completion of this thesis, more importantly, the challenging research that lies behind it.

I am also grateful to my colleagues Yan Luo and Mohammadreza Ghaderpanah for their friendship and valuable comments relating to this research.

At last but not least, I would also like to thank my family and friends for their unconditional support and help.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

With stronger competition and stricter safety and environmental regulations, there has been an increasing demand for better quality products. To better meet this new reality, many manufacturing industries have reviewed their processes and raised their specifications and acceptable standards [1]. Modern industrial processes contain a large number of variables that are regularly monitored and inspected for any type of malfunction [2].

This type of process monitoring is known as statistical process control (SPC). The concept is based on the assumption that high variation leads to inferior quality. Therefore when several process parameters are controlled within specific targets, the end product trends to be in control and within specification. To achieve this, samples are frequently collected for each variable and displayed with the visual help of a control chart [3].

A control chart is a useful statistical tool that can be used to distinguish and detect between common causes of variation (random noise) and special causes (signal). The samples are usually plotted over time between two thresholds defined as control limits. Any on-line data violating these limits, would indicate a fault (see Fig. 1.1). This is a signal that some process investigation is needed in order to detect and remove these unusual sources of variation [1].

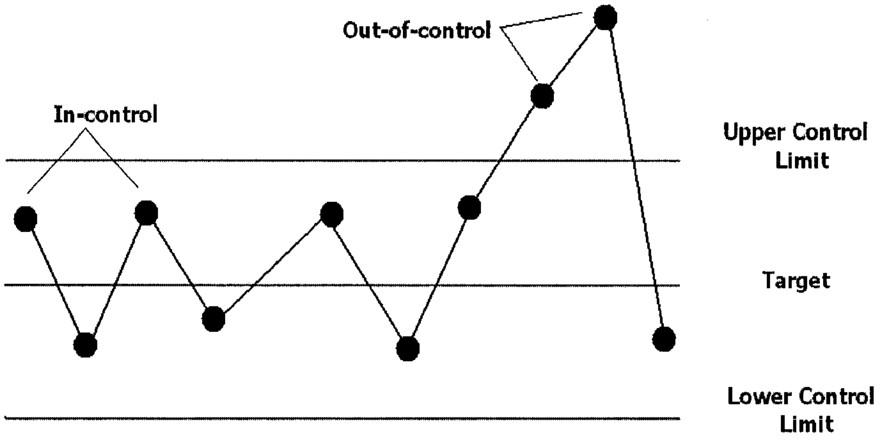


Figure 1.1: Control chart illustration.

1.1 Framework and motivation

1.1.1 Process management

In any process, there are five major steps in continuously controlling and improving the product. The first step is to monitor the performance of the process. This entails selecting and determining the key variables that need to be monitored, choosing an adequate sampling interval, and establishing key targets and thresholds for each variable [1].

The second step consists of determining if a fault has occurred. A fault or an outlier is usually defined as a deviation from the normal operating conditions [25]. An early fault detection can limit the damage and avoid serious process upsets [1].

The third step is to identify the root cause of the fault. This includes determining the variable or combination of variables that created this excess variation.

The fourth step consists of diagnosing the faults by determining the type, magnitude and time of the fault. In the end, the solution needs to overcome and eliminate the fault from reoccurring.

Finally the last step consists of applying the solution and confirming the removal of the fault.

Once the process returns back to normal operating conditions, the operator can return to monitoring the process [1].

This five step problem-solving approach will help eliminate and reduce variability in the process and result to a higher quality product (see Fig. 1.2) . This framework is also part of the basic concept of six -sigma [4].

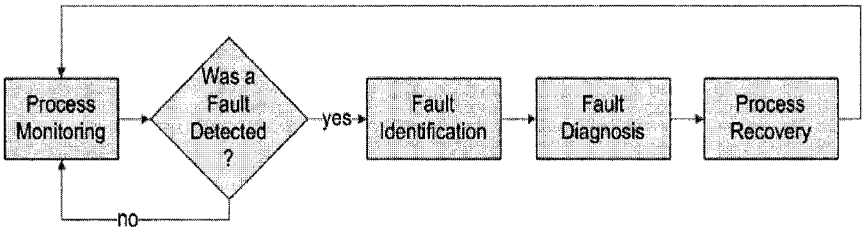


Figure 1.2: Process management framework.

1.1.2 Multivariate statistical process control

Univariate statistical process control is widely used to monitor and diagnose faults and outliers. For attribute variables, common charts used are the count chart (c chart) and the fraction defective chart (p chart). For continuous variables, operators and engineers use the the X-bar chart, R chart and S chart. These tools are very well documented and understood [4].

Unfortunately, in most industrial environment, the variables are highly correlated by nature. This is particularly true for manufacturing and chemical processes. Currently, most industries model their processes by monitoring each variable independently of the other. This has a result to overwhelm the operator and create misleading results [4].

In Fig. 1.3, we can observe the result of modelling two variables who are highly correlated as independent. The ellipse region defines where the process is operating under normal operating conditions. Any sample falling outside the ellipse is considered an outlier. However, if the variables were modelled as independent, the control region would be defined between the rectangle. As observed, some out-of-control observations would be misidentified. Therefore the importance of representing the correlation structure between variables in order to accurately characterize the behavior of most industrial environments is demonstrated [4].

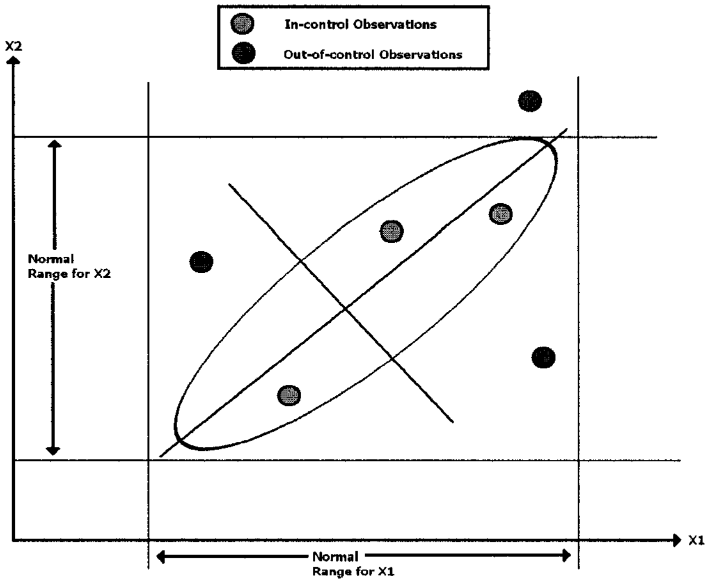


Figure 1.3: Comparison of univariate and multivariate control charts.

1.2 Background

This thesis addresses the application of multivariate statistical process control for fault detection and diagnosis. The following background material is presented to provide context for this work.

1.2.1 Multivariate T^2 statistic

The most familiar and widely used process-monitoring and control procedure is the Hotelling T^2 control chart. Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m]^T$ be an $m \times p$ data matrix of m vectors $\boldsymbol{x}_i \in \mathbb{R}^p$, that is each observation $\boldsymbol{x}_i$ is a row vector p variables. Hotelling's T^2 statistic, also referred to as Mahalanobis distance, is defined as

$$T_i^2 = (\boldsymbol{x}_i - \bar{\boldsymbol{x}})S^{-1}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \quad \text{for} \quad i = 1 \ldots m$$

where

$$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \quad \text{and} \quad S = \frac{1}{m-1}\sum_{i=1}^{m}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})$$

are the sample mean and sample covariance matrix respectively.

The T^2 statistic is derived from the assumption that the random observations follow a multivariate normal distribution. Therefore if the mean and covariance matrix are known, then the T^2 statistic follows a χ^2 distribution with p degrees of freedom evaluated at the confidence level $1-\alpha$. The control limits can be derived as [4]

$$\begin{aligned} UCL &= \chi^2_{\frac{\alpha}{2},p} \\ LCL &= \chi^2_{1-\frac{\alpha}{2},p} \end{aligned}$$

When the actual covariance matrix and mean are not known, then the process monitoring can be separated in two major phases. The first phase consists of preparing an outlier-free reference sample that will be used as a benchmark for all the new observations [1,4]. In this case, the upper and lower control limits are defined as

$$\begin{aligned} UCL &= \frac{(m-1)^2}{m}\frac{[\frac{p}{m-p-1}]F_{\frac{\alpha}{2},p,m-p-1}}{1+[\frac{p}{m-p-1}]F_{\frac{\alpha}{2},p,m-p-1}} \\ LCL &= \frac{(m-1)^2}{m}\frac{[\frac{p}{m-p-1}]F_{1-\frac{\alpha}{2},p,m-p-1}}{1+[\frac{p}{m-p-1}]F_{1-\frac{\alpha}{2},p,m-p-1}} \end{aligned}$$

where F_{α,ν_1,ν_2} is the $(1-\alpha)$ percentile of the inverse of the F cumulative distribution with ν_1 and ν_2 degrees of freedom.

The second phase consists of monitoring new observations based on the reference sample found in phase one. In this case, the upper and lower control limit will be different. This is due to the fact that each observation is now dependent of the sample mean and sample covariance extracted from phase one. The control limits are as follows:

$$\begin{aligned} UCL &= \frac{p(m+1)(m-1)}{m(m-p)} F_{\frac{\alpha}{2},p,m-p} \\ LCL &= \frac{p(m+1)(m-1)}{m(m-p)} F_{1-\frac{\alpha}{2},p,m-p} \end{aligned}$$

Note that most interest is generated from an observation exceeding the UCL than the LCL. Therefore it is a common practice to ignore the LCL and evaluate the UCL at α instead of $\alpha/2$ [3]. In some SPC researchers suggest that an observation that exceeds the LCL should still be investigated since this out-of-control situation might express a fault in the data recording [4].

One of the main difficulties of multivariate control charts is to diagnose the root cause for an out-of-control situation. One approach would be to plot each individual variable separately and then determine the combination of variables that caused this excess variation. However, this method might not give the correct solution [4]. Another approach that may be useful to diagnose outliers is to decompose the T^2 statistic into individual components representing each variable. The indicator for each variable can be defined as

$$d(j) = T_i^2 - T_i^2(j) \quad \text{for} \quad j = 1, \dots p$$

where T_i^2 statistic is the current outlying value and $T_i^2(j)$ is the value of the statistic for all variables except the j^{th} one. This would require to diminish the data matrix by one dimension represented by the j^{th} variable and reduce both mean and sample covariance matrix. The variables associated with large $d(j)$ indicator can be associated as the root cause for this out-of-control variation [3].

Also, in the presence of large number of variables, direct use of the T^2 statistic is not efficient. This is particularly true for data sets containing redundant information which may lead to ill-conditioning or collinearity problems. For this reason, data reduction techniques such as principal component analysis (PCA) are needed to separate and extract key signals from the process [41]

1.2.2 Principal component analysis

Principal component analysis is a dimensionality reduction technique that transforms the original variables into a set of linear combinations. This is achieved by determining a set of eigenvectors, called principal components, that capture most of the variance (i.e eigenvalues) present in the data [4].

To obtain the eigenvectors and the eigenvalues needed for PCA, we need to perform the singular value decomposition on the covariance matrix

$$S = A\Lambda A^T,$$

where Λ is a $p \times p$ diagonal matrix of real non-negative eigenvalues sorted in decreasing magnitude and A is $p \times p$ matrix containing the corresponding column eigenvectors.

To minimize the negative impact that random noise has on capturing the true variation of the data, k largest eigenvalues $\Lambda_k = (\lambda_1, \ldots, \lambda_k)$ are selected with its corresponding eigenvectors $A_k = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$. Therefore, PCA reduces the dimensionality of the original data matrix X by projecting it into a new coordinate system where the axes maximize the variability

$$Y = XA_k$$

where Y is referred as the principal component score matrix.

The projection of the score matrix with the reduced eigenvectors will retrieve back the p dimensional data matrix

$$\widetilde{X} = YA_k^T$$

The residual matrix can be obtained by subtracting the retrieved data matrix from the original data matrix

$$E = X - \widetilde{X}$$

This residual matrix captures the variance associated with the $p - k$ smallest eigenvalues [1].

1.2.3 Dimension reduction techniques

As mentioned earlier, selecting the correct number of eigenvectors k will separate the key signals of the process from the random noise. Several techniques exist for determining the value of the reduction order, but there is no apparent dominant technique [1]. In the sequel we will briefly present an overview of some of these methods.

The percent variance test determines the number of k components by calculating the number of eigenvectors needed to represent a certain percentage of the total variance. Given that each eigenvalue (λ_i) represents a certain percentage of the variance, the percent variance test equation can be given by

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i} = \alpha\%$$

Since the minimum percentage ($\alpha\%$) is chosen arbitrary, the number k components needed for each application may be too low or too high [1,3].

The scree method shown in Fig. 1.4, plots linearly the eigenvalues in decreasing order. The number of components k is chosen at the location where the profile is starting to curve. This is based on the assumption that random noise should form a linear profile. In Fig. 1.4 we can observe a typical scree curve for a data matrix containing 10 variables. In this case, we would choose between 2 and 3 components. The identification of the number of components may be ambiguous to identify and therefore hard to automate [1].

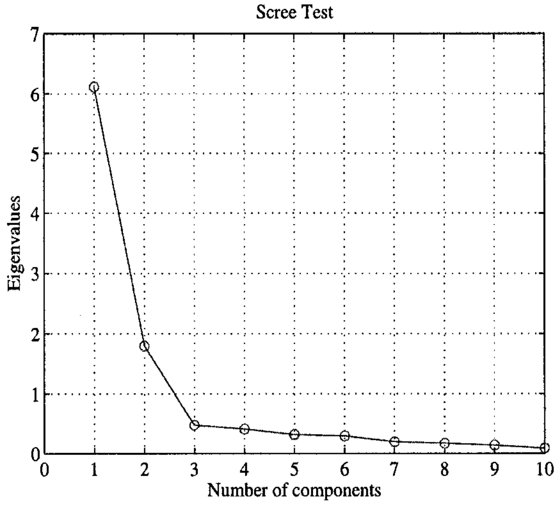


Figure 1.4: Scree plot.

The pareto chart is a combination of the two previous methods discussed. It linearly plots the accumulative percentage of each eigenvalue and display the individual percent variance contribution of each component in decreasing order in a bar graph. We can select the k components by observing the break of the linear curve. If the break is between two components, we may look at the bar graph and decide on the total variance each component will additional contribute. By looking at the pareto chart in Fig. 1.5 we can observe that $k = 2$ seems to be the appropriate choice.

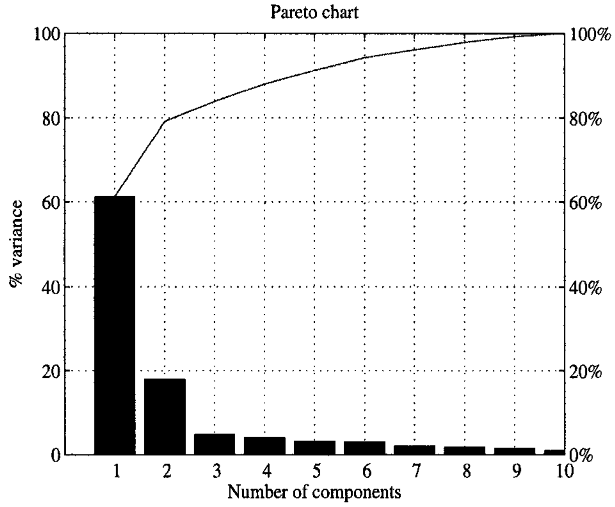


Figure 1.5: Pareto chart.

Finally, the number of components needed to represent the key signals of the process can also be determined using the residual matrix for cross-validation. This test is known as the $PRESS$ statistic and is expressed as

$$PRESS(k) = \frac{1}{mp}\|X - \widetilde{X}\|_F^2,$$

where k is the number of components chosen and $\|\cdot\|_F$ is the Frobenius norm. To implement this technique, the data matrix X can be subdivided into smaller subgroups where the $PRESS$ statistic is calculated for a preselected number of k components. This step is repeated for all subgroups using different k components each time. We choose the k components that average the smaller $PRESS$ statistic for all the different subgroups [1].

1.2.4 PCA monitoring statistics

As we mentioned earlier, PCA is a method for transforming the observations in a dataset into new observations which are uncorrelated with each other and account for decreasing proportions of the total variance of the original variables.

Standardizing the data is often preferable when the variables are in different units or when the variance of the different columns of the data is substantial. The standardized data matrix is given by

$$Z = (X - \mathbf{1}\,\bar{\boldsymbol{x}})D^{-1/2},$$

where $\mathbf{1} = (1, \ldots, 1)^T$ is a $n \times 1$ vector of all 1's, and $D = (\mathrm{diag}(S))^{1/2}$ is the diagonal standard deviation matrix. It is worth pointing out the covariance matrix S of the standardized data Z is exactly the correlation matrix of the original data, and it is given by $R = D^{-1/2}SD^{-1/2}$. PCA is then performed by applying eigen-decomposition to the matrix R, that is $R = A\Lambda A^T$.

The Mahalanobis distance (T^2) based on the first k principal components can then be defined as

$$T_i^2 = \boldsymbol{y}_i {\Lambda_k}^{-1} \boldsymbol{y}_i^T \quad \text{for} \quad i = 1, \ldots, m$$

where $Y = ZA_k = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]^T$ is $m \times k$ principal component score data matrix.

The derived control limits are very similar to the conventional T^2 statistic. The first phase control limits are defined as

$$\begin{aligned} UCL &= \frac{(m-1)^2}{m} \frac{[\frac{k}{m-k-1}]F_{\frac{\alpha}{2},k,m-k-1}}{1 + [\frac{k}{m-k-1}]F_{\frac{\alpha}{2},p,m-p-1}} \\ LCL &= \frac{(m-1)^2}{m} \frac{[\frac{k}{m-k-1}]F_{1-\frac{\alpha}{2},k,m-k-1}}{1 + [\frac{k}{m-k-1}]F_{1-\frac{\alpha}{2},k,m-k-1}} \end{aligned}$$

and for new observations, the second phase control limits are defined as

$$\begin{aligned} UCL &= \frac{k(m+1)(k-1)}{m(m-k)} F_{\frac{\alpha}{2},k,m-k} \\ LCL &= \frac{k(m+1)(k-1)}{m(m-k)} F_{1-\frac{\alpha}{2},k,m-k} \end{aligned}$$

Moreover, we have an additional statistic based on the deviation of the observation to the PCA representation. The second metric is referred to as the Q^2 statistic or as the squared prediction error (SPE). It is expressed as the the squared difference between the observed values and predicted values:

$$Q_i^2 = (\boldsymbol{x}_i - \boldsymbol{y}_i A_k^T)(\boldsymbol{x}_i - \boldsymbol{y}_i A_k^T)^T = \|\boldsymbol{x}_i - \boldsymbol{y}_i A_k^T\|^2$$

The distribution for the Q^2 statistic, which can also be used as a threshold, is defined as

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

where

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \quad \text{and} \quad \theta_i = \sum_{j=k+1}^{p} \lambda_j^i \quad \text{for} \quad i = 1, 2, 3$$

and c_α is the value of the inverse Gaussian cumulative distribution evaluated at the confidence level $(1 - \alpha)$. Therefore an out of control observation in the T^2 chart would identify a major variation of the data in which the correlation structure is preserved where as an increase in the Q^2 would identify a breakdown of the correlation structure between the principal components and the subspace [46]. It is important to note that the mean and covariance matrix are extracted under normal operating conditions (phase 1) once all outliers are identified and deleted [1,3,37,45].

1.2.5 Robust statistics-based approaches

In this section, we will briefly review some robust statistics based methods that are used for detecting multivariate outliers.

Multivariate Trimming (MVT)

Trimming was first successfully applied to univariate control charts to detect outliers and then was later modified to fit the multivariate settings. The trimming approach consists of calculating the Mahalanobis distance for the m observations, followed by the deletion of the observation with the largest Mahalanobis distance. Then, the remaining $(m - 1)$ observations are used to recalculate the T^2 statistic and its control limits. These steps are repeated until a fixed percentage of the observations have been excluded [33]. However, the trimming technique is shown to be efficient when only a small amount of outliers are present in the data [7,27].

Sullivan & Woodall First Approach (SW1)

This technique was proposed by Sullivan & Woodall [17] who showed that the T^2 chart using the sample covariance matrix is not effective in detecting shifts in the mean vector, and they recommended that the covariance matrix should be estimated using the vector difference between two successive observations $\boldsymbol{v}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i$, $i = 1, \ldots, m-1$. This method is, however, not effective in detecting large number of outliers, and it excels in detecting a sudden consecutive shift in the mean [7].

Sullivan & Woodall Second Approach (SW2)

This approach is based on the stalactite plot for outliers detection [15]. The idea is to calculate the mean and covariance matrix from $(p+1)$ randomly picked observations. Then, the Mahalanobis distance using all m observations is calculated. Next, $(p+2)$ observations that have the smallest Mahalanobis distance are picked, and the process continues adding one observation at a time until a fixed amount out of m observations is selected. Again, this technique remains vulnerable to data that contains a large number of outliers and also depends on the robustness of its initial random sample [7].

Minimum Volume Ellipsoid (MVE)

The first step of the minimum volume ellipsoid (MVE) method is to find the smallest ellipsoid containing at least half of the observations. Then, a robust mean and covariance matrix based on these observations are estimated and the distances based on these estimates are proved to be very effective in detecting several outliers in multivariate environments [7].

The MVE algorithm consists of drawing a random sample (W) containing $p+1$ observations to calculate the mean and the sample covariance matrix. Then, the Mahalanobis distance for all m observations is calculated. Next, we calculate the parameter $V_w = \det(m_w^2 S_W)^{\frac{1}{2}}$ where m_w^2 is the h^{th} order statistic of the m Mahalanobis distances and $h = \lfloor m + p + 0.5 \rfloor$. Then, we store the parameter V_w and the indices of the observations and we repeat the above steps n times. The observations associated with the smallest V_w are used to calculate the robust mean and robust sample covariance matrix given by

$$S_{MVE} = \left(1 + \frac{15}{m-p}\right)^2 (\chi^2_{0.5,p})^{-1} m_w^2 S_W$$

where $\chi^2_{\beta,v}$ is the is the value of the inverse of the chi-square cumulative distribution with v degrees of freedom, evaluated at the value β.

Minimum Covariance Determinant (MCD)

Minimum covariance determinant is considered by many as the current best-performing technique for low dimensional data [27]. It randomly selects a subset of the data containing $p+1$ observations to calculate the mean and covariance matrix. The Mahalanobis distance is then calculated for all m observations in order to select $\geq m/2$ observations with the smallest distance. The determinant of the covariance matrix of the selected observation are then calculated. This process is repeated n times until the smallest determinant is found. The mean and covariance matrix of the $\geq m/2$ observations containing the smallest determinant are considered to be robust [7, 24, 27, 33]. A detailed description of the algorithm can be found in chapter two.

1.3 Contributions

The contributions of this thesis are as follows:

☞ **Multivariate robust quality control chart for outlier detection:** We present a new robust multivariate control chart using principal component analysis and robust statistics. The proposed approach consists of two main steps. In the first step, we calculate a robust covariance matrix using the minimum covariance determinant algorithm. In the second step, we apply eigen-decomposition to the robust correlation matrix in order to extract the eigenvalues that will be used to define the proposed control chart. Our experimental results illustrate the much better performance of the proposed algorithm in comparison with existing statistical monitoring and controlling charts.

- ☞ **Statistical process control using kernel PCA:** We propose a new multivariate statistical process control chart using kernel principal component analysis. The core idea behind our proposed technique is to project our data into higher dimension space in order to extract the eigenvalues and eigenvectors of the kernel matrix. The proposed control chart is robust to outliers, and its control limits are derived from the eigen-analysis of the Gaussian kernel matrix in the Hilbert feature space. Our experimental results demonstrate a much improved performance in comparison with the current multivariate control charts.

- ☞ **Cluster principal component analysis for outlier detection:** We introduce a new method to detect multiple outliers in both low and high dimensional data. We combine the advantages behind hierarchical clustering and principal component analysis to improve the performance while keeping the complexity and computation time relatively low. Our main idea consist of identifying an optimal subset of observations that will be used as a comparison model to identify all outliers present. The experimental and simulated results clearly show a much improved performance of the proposed approach in comparison with existing robust algorithms.

- ☞ **Dynamic fault detection and diagnosis using independent component analysis:** We present a new process monitoring techniques which we refer to as dynamic independent component analysis (DICA). We extend the advantages behind ICA to detect outliers in a time correlated environment and introduce an innovative way on how to diagnose faults. Our experimental results demonstrate that modelling a time dependent process dynamically and applying DICA on the data outperforms existing monitoring methods that are currently in use.

1.4 Thesis overview

The organization of this thesis is as follows:

- ❑ The first Chapter contains a brief review of essential concepts and definitions which we will refer to throughout the thesis, and presents a short summary of material relevant to multivariate statistical process control.

- ❑ In Chapter 2, we introduce a new robust multivariate control chart using robust statistics and principal component analysis. The effectiveness of our method will be demonstrated through extensive experimental results.
- ❑ In Chapter 3, we introduce a different multivariate statistical process control using kernel principal component analysis. A detailed description of the algorithm and a comparison study with similar methodologies will be presented.
- ❑ In Chapter 4, we present a new distance based approach to detect multivariate outliers given no previous model history. A description behind the foundation of our algorithm will be explained. Simulations and experimental results will be presented to demonstrate the much improved performance of the proposed approach.
- ❑ In Chapter 5, we introduce a new extension of independent component analysis to monitor time correlated multivariate data. A description of the methodology as well a fault diagnosis will be explained. Experimental results on the Tennessee Eastman process will demonstrate the higher sensitivity in detecting outliers in comparison with existing process monitoring techniques.
- ❑ In the **Conclusions** Chapter, we summarize the contributions of this thesis, and we propose several future research directions that are directly or indirectly related to the work performed in this thesis.

1.5 Publications

- ✍ G. Stefatos and A. Ben Hamza, "Multivariate robust quality control chart for outlier detection," revised & resubmitted to *Quality Engineering Journal*, 2007.
- ✍ G. Stefatos, Yan Luo, and A. Ben Hamza, "Kernel principal component chart for defect detection," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, Vancouver,

Canada, 2007.

✍ G. Stefatos and A. Ben Hamza, “Statistical process control using kernel PCA,” *Proc. 15th IEEE Mediterranean Conference on Control and Automation,* Athens, Greece, 2007.

✍ G. Stefatos and A. Ben Hamza, “Cluster PCA for outliers detection in multivariate data,” *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Montreal, 2007.

✍ G. Stefatos and A. Ben Hamza, “Dynamic fault detection and diagnosis using independent component analysis,” to be submitted, 2007.

CHAPTER 2

Multivariate Robust Quality Control Chart for Outlier Detection

We present a new multivariate statistical process control chart using robust statistics and principal component analysis [52]. The proposed approach consists of two main steps. In the first step, a robust covariance matrix is calculated using the minimum covariance determinant algorithm. In the second step, an eigen-analysis of the robust correlation matrix is performed. Our experimental results illustrate the much better performance of the proposed algorithm in comparison with existing statistical monitoring and controlling charts.

2.1 Introduction

Many manufacturing and service businesses use statistical methods to monitor the performance of their processes [3]. However, in many cases, there will be more than one measurement process to monitor [4]. Currently, the industries use independent univariate control charts to study each variable. This approach not only leads to frequent adjustments of the process but also it does not account for correlation between the measurement processes [6]. Most processes are, however, multivariate and highly correlated. [4,6].

In recent years, several techniques have been proposed to analyze and monitor multivariate data [7,8,17]. With multivariate quality control charts, it is possible to have well-defined control limits, while taking in consideration the cross correlation between the variables. In addition, these charts may be used to analyze the process for its stability without the complication of monitoring several univariate control charts [4].

In this chapter, we present a new robust multivariate control chart using principal component analysis [30] and robust statistics [11–13]. The proposed approach consists of two main steps. In the first step, we calculate a robust covariance matrix using the minimum covariance determinant algorithm. In the second step, we apply eigen-decomposition to the robust correlation matrix in order to extract the eigenvalues that will be used to define the proposed control chart.

The remainder of the paper is organized as follows. In the next section, we describe the problem formulation. In Section 2.3 we introduce the proposed robust multivariate control chart. In Section 2.4, experimental results are presented to demonstrate the performance of the proposed approach in comparison with existing monitoring techniques. Section 2.5 concludes the chapter.

2.2 Problem formulation

Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^T$ be an $m \times p$ data matrix of m vectors $\boldsymbol{x}_i \in \mathbb{R}^p$, where each observation $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ is a row vector with p variables.

Mapping a multivariate situation as a univariate may lead to results where processes might seem to be in control when in fact they are not or vise versa. There are typically two phases for multivariate control charts. In Phase I, the collected data is used to establish the control limits. In phase II, a distinction is made between "historical" data and "future" data. Historical data, collected in phase I, is used to generate the control limits for the future data, collected in phase II.

A common method used by both engineers and manufactures is the Hotelling's T^2 statistic chart [4] which allows several characteristics of a manufactured component to be monitored simultaneously for specified control limits. However, the T^2 statistic chart has its limitations: in order to obtain significant results, both the mean and covariance matrix must be robust to outliers [7]. Moreover, unlike the univariate charts, the T^2 statistic does not represent the original variables. Hence, when out-of-control situations occur, the cause can not be determined, whether it be due to an excess variation of a particular variable or to a change in the covariance/correlation matrix [4].

2.3 Proposed method

Principal component analysis (PCA) is a linear transformation of the original data set into smaller number of components while keeping most of the variance [30]. PCA has two main advantages over other multivariate statistical process control techniques. First, the principal components are uncorrelated and second, only a few components are needed in order to capture most of the variance [4].

On the other hand, robust statistics are very effective for outliers detection and removal [11–13]. Motivated by the outperformance of PCA and robust statistics, we propose a new multivariate quality control chart algorithm which consists of two main phases as shown in Table 2.1. In Phase I, we calculate the robust mean and the robust covariance of our data matrix using the minimum covariance determinant (MCD) method. The MCD technique finds a subset containing half of the data such that its covariance matrix has the lowest determinant, and based on half of these observations a robust mean and a robust covariance matrix are calculated [25].

Standardizing the data is often preferable when the variables are in different units or when the variance of the different columns of the data is substantial. In Phase II of our algorithm, we perform PCA using a standardized data with the robust mean and robust standard deviation in order to further separate the outliers from the data set and to also unitize the magnitude of the variables.

The robust standardized data matrix $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m]^T$ is given by $Z = (X - \mathbf{1}\,\bar{\boldsymbol{x}})D^{-1/2}$, where $\mathbf{1} = (1, \ldots, 1)^T$ is a $n \times 1$ vector of all 1's, and $D = (\text{diag}(S_H))^{1/2}$ is the diagonal of the robust standard deviation matrix.

PCA is then performed by applying eigen-decomposition to the robust correlation matrix R_H, that is $R_H = A\Lambda A^T$ where $A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)$ is a $p \times p$ matrix of eigenvectors (also called principal components) and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$ is a diagonal matrix of eigenvalues. These eigenvalues are arranged in decreasing order where each value expresses a certain percentage of the total variance. The principal component score matrix is a linear transformation given by $Y = ZA$. Moreover, the covariance matrix is

$$\text{cov}(Y) = \frac{1}{m-1} Y^T Y = \frac{1}{m-1} A^T Z^T Z A = \Lambda.$$

Table 2.1: Algorithmic steps robust PCA.

Algorithm: Robust Principal Component Control Chart

Phase I:

1. Draw a random sample $K = [\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_{p+1}]^T$ from the data $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m]^T$ containing $(p+1)$ observations.
2. Compute the mean $\bar{\boldsymbol{x}}_K$ and the sample covariance matrix S_K.
3. Compute the Mahalanobis distance:

$$T_i^2 = (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_K) S_K^{-1} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_K)^T, \quad i = 1 \dots, m$$

4. Select $h = \lfloor m + p + 0.5 \rfloor$ observations with the smallest Mahalanobis distance T_i^2 to obtain a subsample $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_h]^T$.
5. Compute the sample mean $\bar{\boldsymbol{x}}_B$ and sample covariance matrix S_B of the sample B.
6. Recompute the Mahalanobis distance using $\bar{\boldsymbol{x}}_B$ and S_B .
7. Reselect h observations with the smallest Mahalanobis distance and repeat step 4) to step 6) until the selected h observations remain consistent.
8. Store the h observations and determinant of the sample covariance matrix.
9. Return to step 1) and repeat n times.
10. Compute the robust mean $\bar{\boldsymbol{x}}_H$, the robust covariance matrix S_H, and the robust correlation matrix R_H from the $h^\star$ observations chosen with the smallest determinant:

$$\begin{aligned}
\bar{\boldsymbol{x}}_H &= \frac{1}{h^\star} \sum_{i=1}^{h^\star} \boldsymbol{b}_i \\
S_H &= \frac{1}{h^\star - 1} \sum_{i=1}^{h^\star} (\boldsymbol{b}_i^\star - \bar{\boldsymbol{x}}_H)^T (\boldsymbol{b}_i^\star - \bar{\boldsymbol{x}}_H) \\
R_H &= D^{-1/2} S_H D^{-1/2}
\end{aligned}$$

where $D = (\mathrm{diag}(S_H))^{1/2}$ is the diagonal of the robust standard deviation matrix.

Phase II:

1. Standardize the data with the robust mean and robust standard deviation.
2. Perform the eigen-analysis of the robust correlation matrix R_H.
3. Compute the principal component scores

Therefore, each principal component score $\boldsymbol{y}_k$ has a variance equal to $var(\boldsymbol{y}_k) = \lambda_k$ for $k = 1, \ldots, p$.

Assuming we want $\pm 3\sigma$ confidence intervals, the upper control limit (UCL), the center line (CL), and the lower control limit (LCL) are given by

$$\begin{aligned} UCL &= +3\sqrt{\lambda_k} \\ CL &= 0 \\ LCL &= -3\sqrt{\lambda_k} \end{aligned}$$

For the purpose of analysis, we may keep r principal component scores out of p depending on the precision (total variance) that is required. As mentioned earlier, each principal component score is a linear combination of the standardized data Z and the eigenvector coefficients:

$$\boldsymbol{y}_j = \sum_{k=1}^{p} a_{kj} \boldsymbol{z}_k, \quad j = 1, \ldots, r$$

2.4 Experimental results

In this section we will compare our proposed method with several robust control charts. Results for the conventional T^2 chart, MVT, SW1, SW2, as well as MVE will be included. Note that for MVT and SW2, we removed 15% of the observations. The following three sets of experiments were performed:

2.4.1 Experiment #1: Woodmod dataset

We tested the performance of our proposed technique on a data set $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{20}]^T$ (called woodmod data [16]) which contains 20 observations as shown in Table 2.2. Each observation $\boldsymbol{x}_i$ has 5 variables which correspond respectively to:

- number of fibers per square millimeter in Springwood
- number of fibers per square millimeter in Summerwood
- fraction of Springwood
- fraction of light absorption by Springwood
- fraction of light absorption by Summerwood

Table 2.2: Woodmod dataset.

x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}
0.5730	0.1059	0.4650	0.5380	0.8410
0.6510	0.1356	0.5270	0.5450	0.8870
0.6060	0.1273	0.4940	0.5210	0.9200
0.4370	0.1591	0.4460	0.4230	0.9920
0.5470	0.1135	0.5310	0.5190	0.9150
0.4440	0.1628	0.4290	0.4110	0.9840
0.4890	0.1231	0.5620	0.4550	0.8240
0.4130	0.1673	0.4180	0.4300	0.9780
0.5360	0.1182	0.5920	0.4640	0.8540
0.6850	0.1564	0.6310	0.5640	0.9140
0.6640	0.1588	0.5060	0.4810	0.8670
0.7030	0.1335	0.5190	0.4840	0.8120
0.6530	0.1395	0.6250	0.5190	0.8920
0.5860	0.1114	0.5050	0.5650	0.8890
0.5340	0.1143	0.5210	0.5700	0.8890
0.5230	0.1320	0.5050	0.6120	0.9190
0.5800	0.1249	0.5460	0.6080	0.9540
0.4480	0.1028	0.5220	0.5340	0.9180
0.4170	0.1687	0.4050	0.4150	0.9810
0.5280	0.1057	0.4240	0.5660	0.9090

The woodmod data variables are highly correlated as shown in Fig. 2.1, and hence multidimensional quality control charts should be applied.

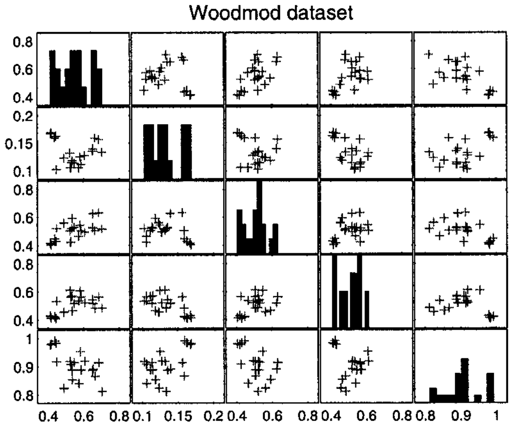


Figure 2.1: Scatter plot of the woodmod dataset.

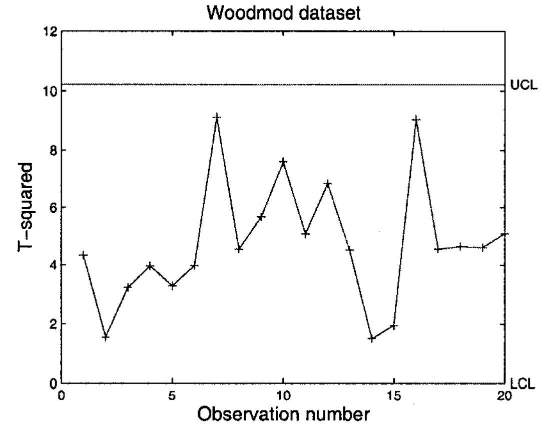


Figure 2.2: Non-robust T^2 control chart.

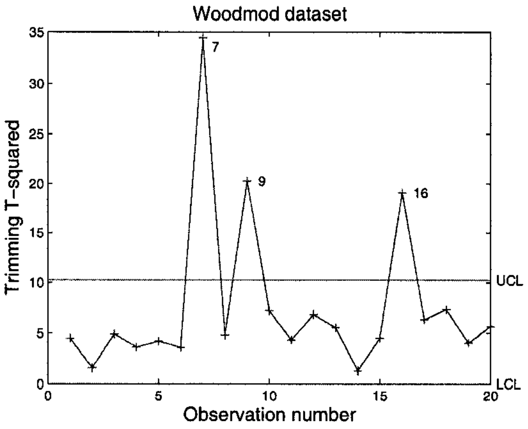


Figure 2.3: MVT T^2 control chart.

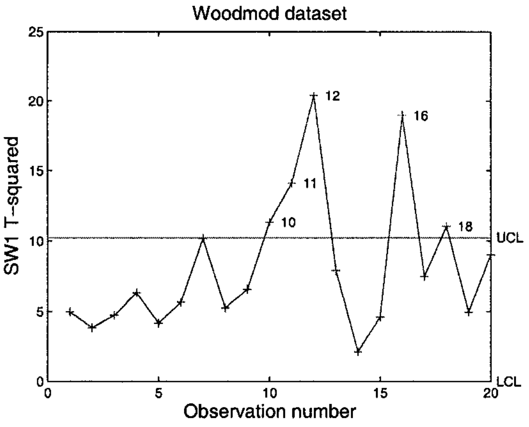


Figure 2.4: SW1 T^2 control chart.

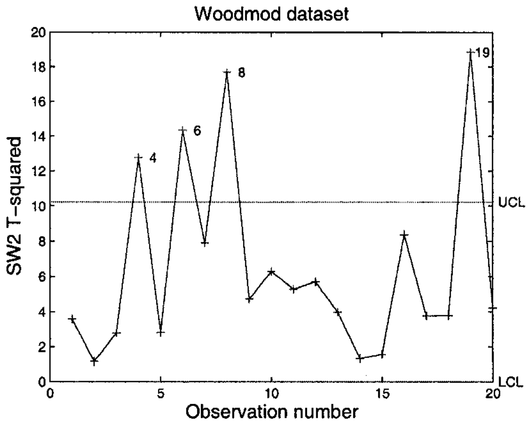


Figure 2.5: SW2 T^2 control chart.

Fig. 2.2 through Fig. 2.6 show the T^2 control charts using the various robust methods discussed in the introduction. It can be seen that SW2 and MVE are able to detect the observations 4, 6, 8 and 19 as outliers, whereas MVT and SW1 are unable to detect the correct outliers. Moreover, the T^2 control chart is unable to find any outliers. It is worth mentioning that for all the T^2 charts, we used a probability of type I error equal to $\alpha = 5\%$.

Fig. 2.7 depicts the principal component charts which clearly indicate the inability of PCA to detect outliers in any of the principal component scores.

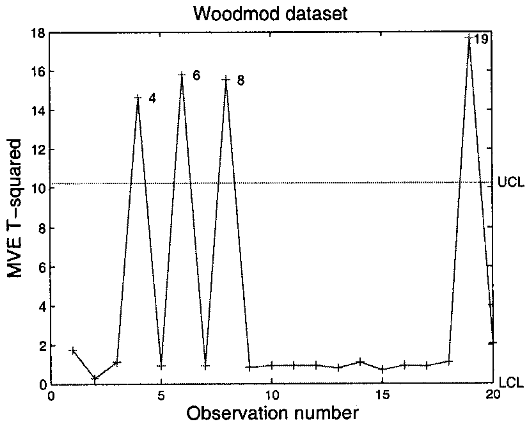
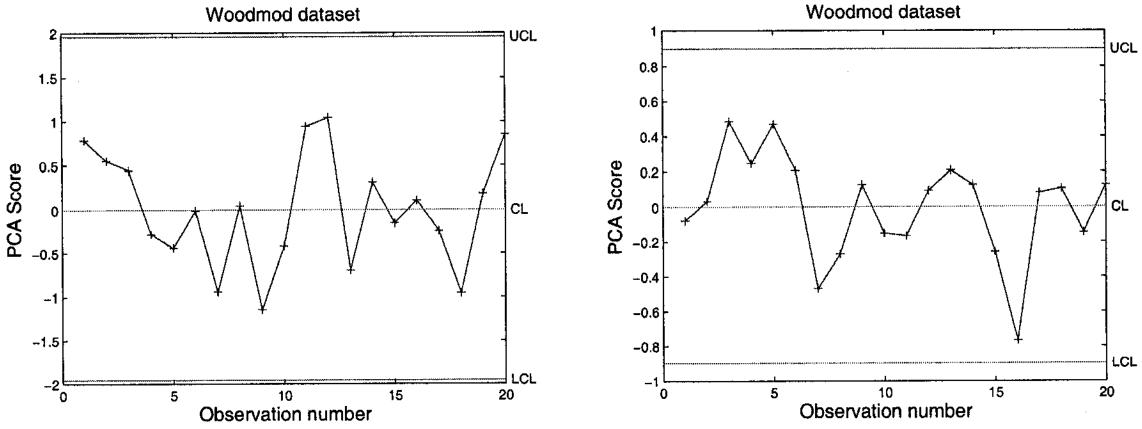


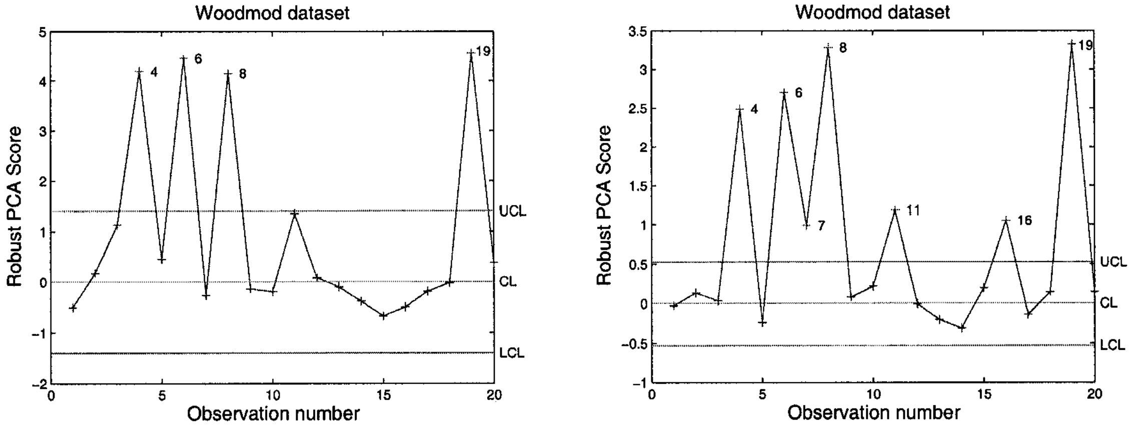
Figure 2.6: MVE T^2 control chart.



(a) (b)

Figure 2.7: (a)-(b) PCA chart with 8.5% variance and 1.8% variance respectively.

The robust principal component charts shown in Fig. 2.8 not only are able to identify all the outliers detected by SW2 and MVE T^2 charts, but also robust PCA was able to find some other smaller outliers such as 7, 11 and 16. This better performance of the proposed multivariate control chart is consistent with a variety of datasets used for experimentation. The contribution plot demonstrating the causation for this excess variation is shown in Fig. 2.9. For example, the excess variation caused on the the 4^{th} principal component is due to the 3^{rd} and 4^{th} variable moving opposite direction with the 5^{th} variable. Also, variable 1 and 2 have minimal impact in this principal component.



(a) (b)

Figure 2.8: (a)-(b) Robust PCA chart with 4.3% variance and 0.6% variance respectively.

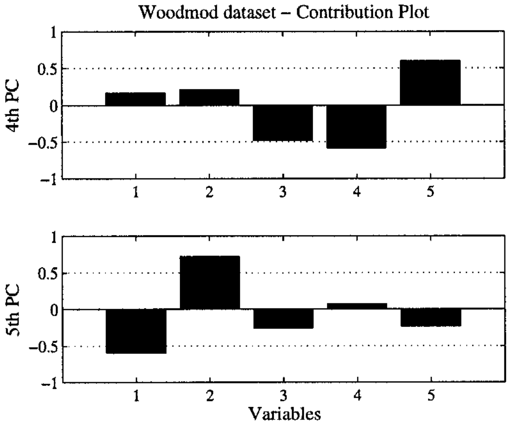


Figure 2.9: Contribution plot for the 4^{th} and 5^{th} principal component.

2.4.2 Experiment #2: Stackloss dataset

Our second analysis was performed on a dataset called Stackloss shown in Table 2.3. This dataset describe the plant oxidation of ammonia to nitric acid, and contains 21 observations, where each observation has 4 variables: rate, temperature, acid concentration, and stackloss.

Table 2.3: Stackloss dataset.

x_{i1}	x_{i2}	x_{i3}	x_{i4}
80.0	27.0	89.0	42.0
80.0	27.0	88.0	37.0
75.0	25.0	90.0	37.0
62.0	24.0	87.0	28.0
62.0	22.0	87.0	18.0
62.0	23.0	87.0	18.0
62.0	24.0	93.0	19.0
62.0	24.0	93.0	20.0
58.0	23.0	87.0	15.0
58.0	18.0	80.0	14.0
58.0	18.0	89.0	14.0
58.0	17.0	88.0	13.0
58.0	18.0	82.0	11.0
58.0	19.0	93.0	12.0
50.0	18.0	89.0	8.0
50.0	18.0	86.0	7.0
50.0	19.0	72.0	8.0
50.0	19.0	79.0	8.0
50.0	20.0	80.0	9.0
56.0	20.0	82.0	15.0
70.0	20.0	91.0	15.0

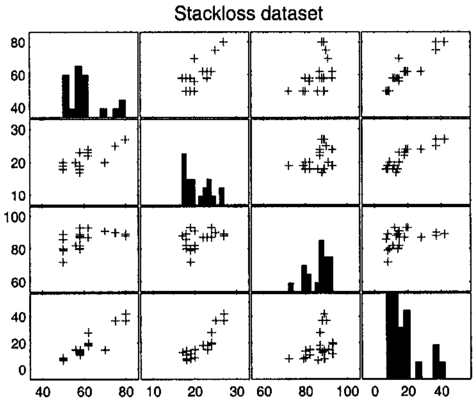


Figure 2.10: Scatter plot of Stackloss dataset.

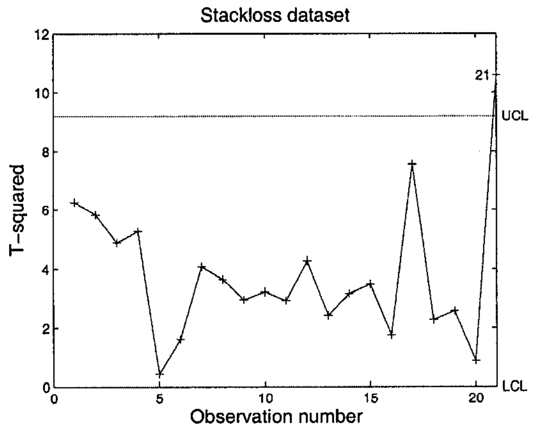


Figure 2.11: Non-robust T^2 control chart.

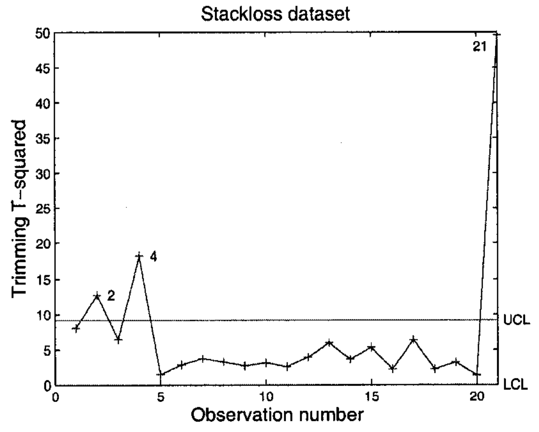


Figure 2.12: MVT T^2 control chart.

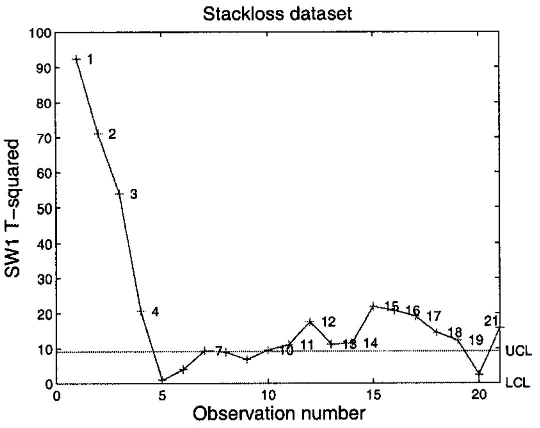


Figure 2.13: SW1 T^2 control chart.

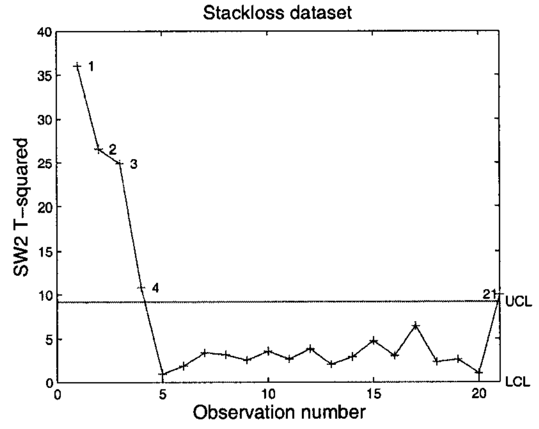


Figure 2.14: SW2 T^2 control chart.

The scatter plot shown in Fig. 2.10 confirms the existence of a high correlation between the variables. The various T^2 control charts are displayed in Fig. 3.6 through Fig. 2.15. It is apparent that there is a common trend between all charts. There is a higher variation in the initial observations ($m = 1, 2, 3, 4$) and on the final observation ($m = 21$). Depending of the algorithm chosen different outliers will be detected.

The principal component control charts are depicted in Fig. 2.16, where no outliers were detected. Robust PCA, however, was able to detect all the outliers present in the different components as shown in Fig. 2.17. The largest outliers are present in the first principal component scores,

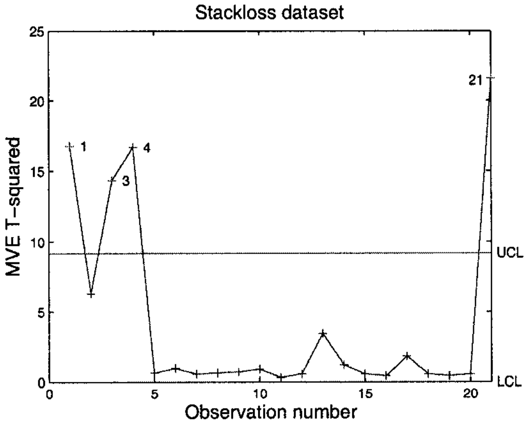


Figure 2.15: MVE T^2 control chart.

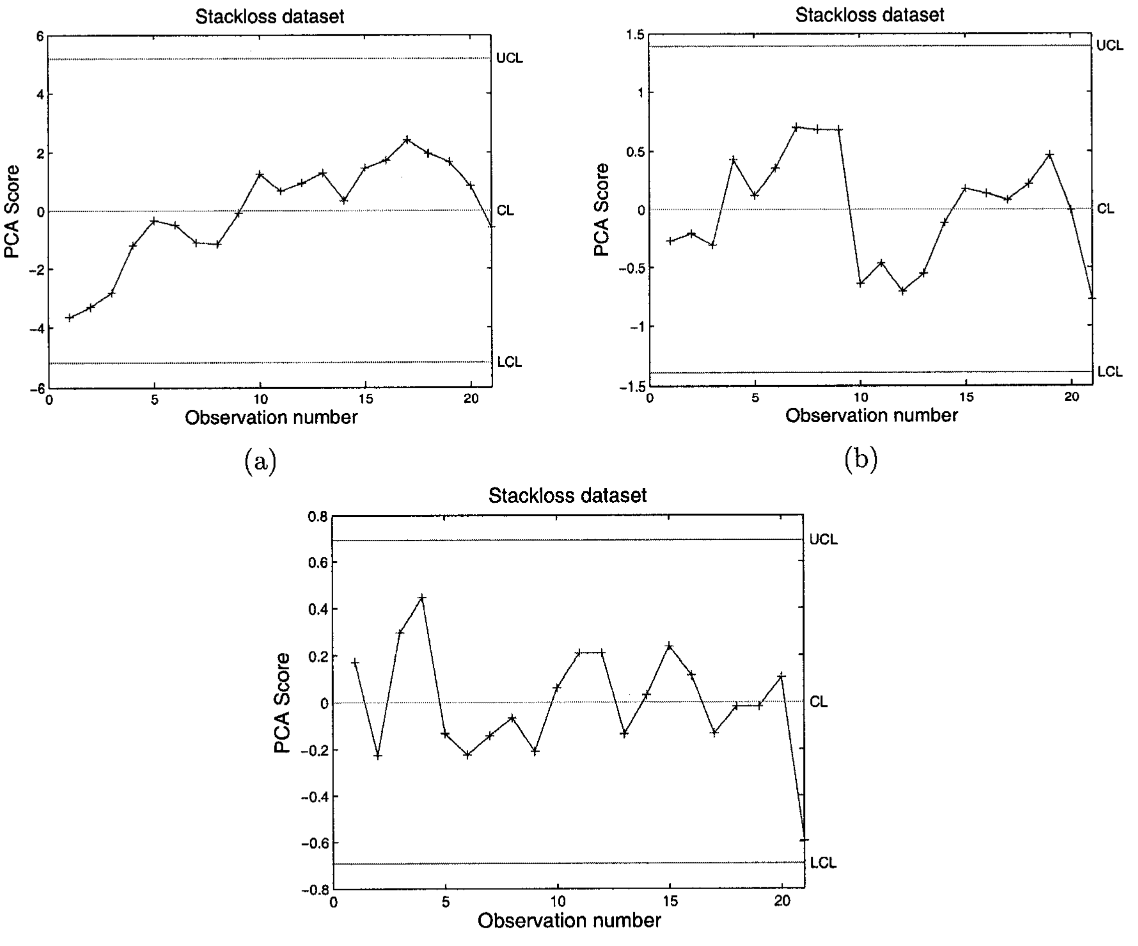
whereas the smallest ones are present in the lower principal component scores. Contribution plot is shown in Fig. 2.18 demonstrating the influence of the variables for each principal component.

2.4.3 Experiment #3: Phosphorus content dataset

The phosphorus content data shown in Table 2.4 contains 18 observations, where each observations has 3 variables: inorganic phosphorus, organic phosphorus, and plant phosphorus. This dataset studies the effect of organic and inorganic phosphorus in the soil in comparison with the phosphorus content of the corn grown. The scatter plot of the data set is shown in Fig. 2.19.

As observed in Fig. 2.20 through Fig. 2.24, the T^2 control charts display a higher variation for the observations 1, 6, 10, and 17. All the control charts detect observation 17 as an outlier. PCA was unable to detect any outlier as shown in Fig. 2.25.

On the other hand, robust PCA was able to detect all the outliers in the different component scores as shown in Fig 2.26. The observation 17 was detected as an outlier in the first principal component, whereas the observations 6 and 10 were detected in the second principal component score. Smaller outliers were also detected in the last principal component containing only 1.8% of the total variance. The contribution plot for all the principal components is shown in Fig. 2.27.



(a)

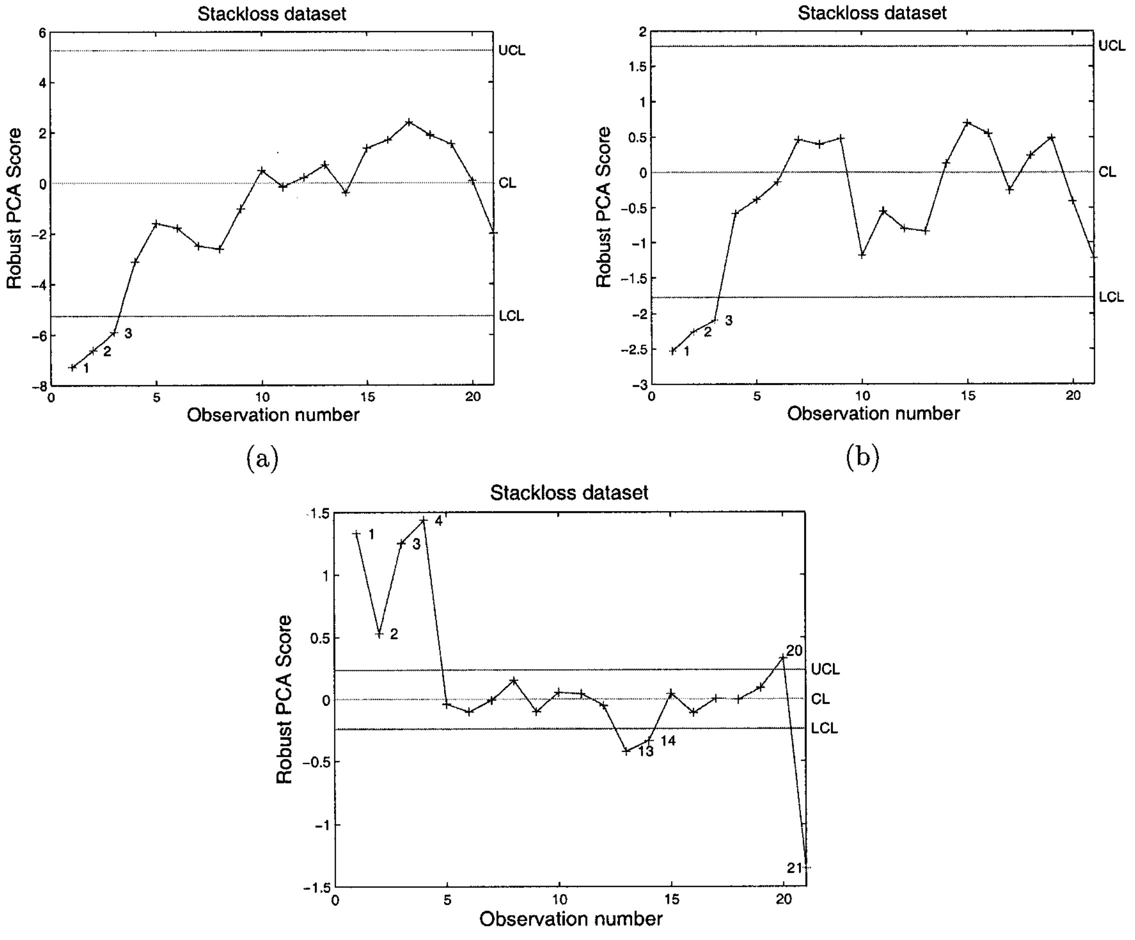
(b)

(c)

Figure 2.16: (a) PCA chart with 74.9% variance, (b) PCA chart with 5.38% variance, (c) PCA chart with 1.33% of variance.

2.5 Conclusions

In this chapter, we introduced a new multivariate quality control chart by combining robust statistics and principal component analysis. The core idea behind our proposed technique is to use robust statistics to estimate the correlation matrix that will be used to extract the robust eigenvalues and eigenvectors. The eigenvalues can then be used to define the control limits of our proposed control chart. The experimental results clearly show a much improved performance of the proposed



(a)

(b)

(c)

Figure 2.17: (a) RPCA chart with 76.54% variance, (b) RPCA chart with 8.84% variance, (c) RPCA chart with 0.16% of variance.

approach compared with the existing quality control methods currently used in the analysis and the monitoring of multivariate data.

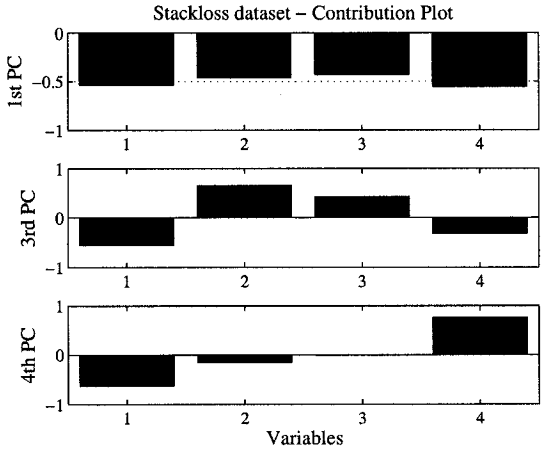


Figure 2.18: Contribution plot for the 1st, 3rd and 4th principal component.

Table 2.4: Phosphorus content dataset.

x_{i1}	x_{i2}	x_{i3}
0.40	53.00	64.00
0.40	23.00	60.00
3.10	19.00	71.00
0.60	34.00	61.00
4.70	24.00	54.00
1.70	65.00	77.00
9.40	44.00	81.00
10.10	31.00	93.00
11.60	29.00	93.00
12.60	58.00	51.00
10.90	37.00	76.00
23.10	46.00	96.00
23.10	50.00	77.00
21.60	44.00	93.00
23.10	56.00	95.00
1.90	36.00	54.00
26.80	58.00	168.00
29.90	51.00	99.00

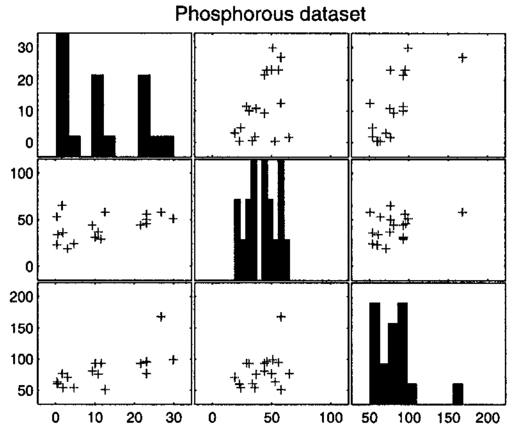


Figure 2.19: Scatter plot of Phosphorus content dataset.

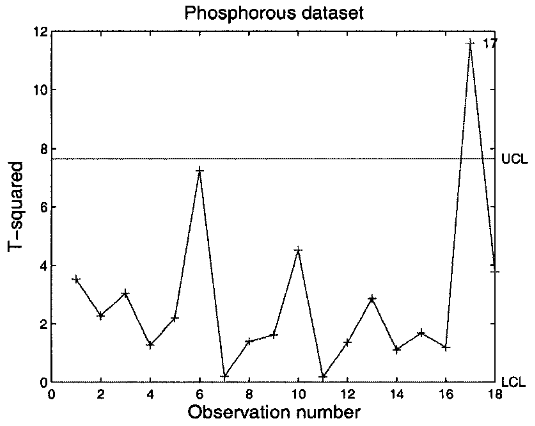


Figure 2.20: Non-robust T^2 control chart.

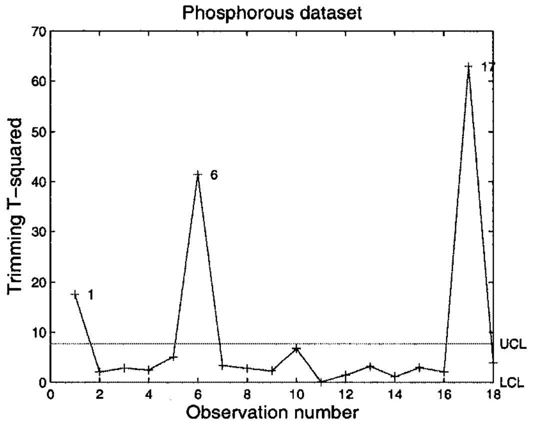


Figure 2.21: MVT T^2 control chart

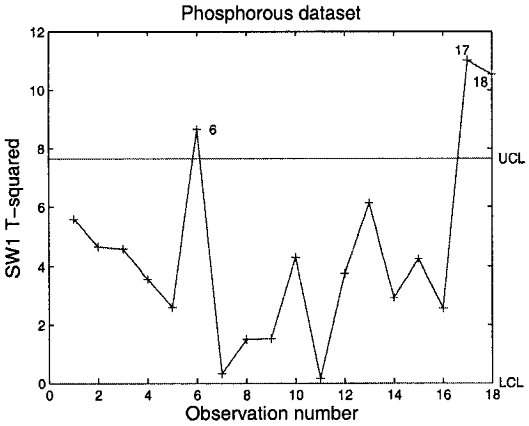


Figure 2.22: SW1 T^2 control chart.

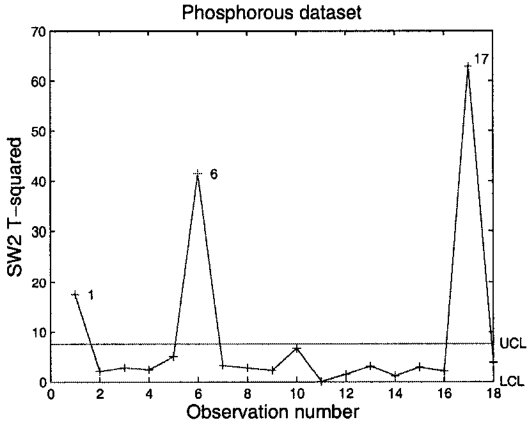


Figure 2.23: SW2 T^2 control chart

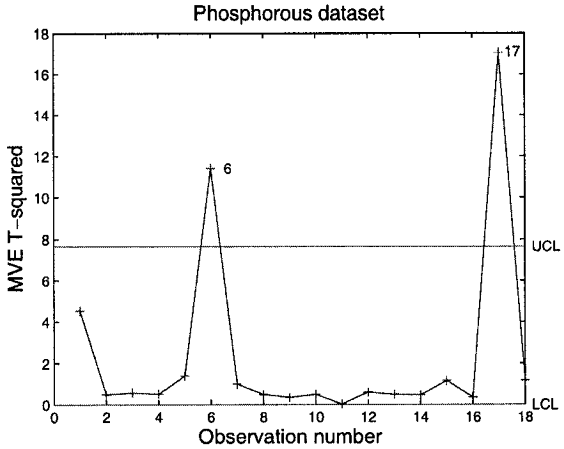
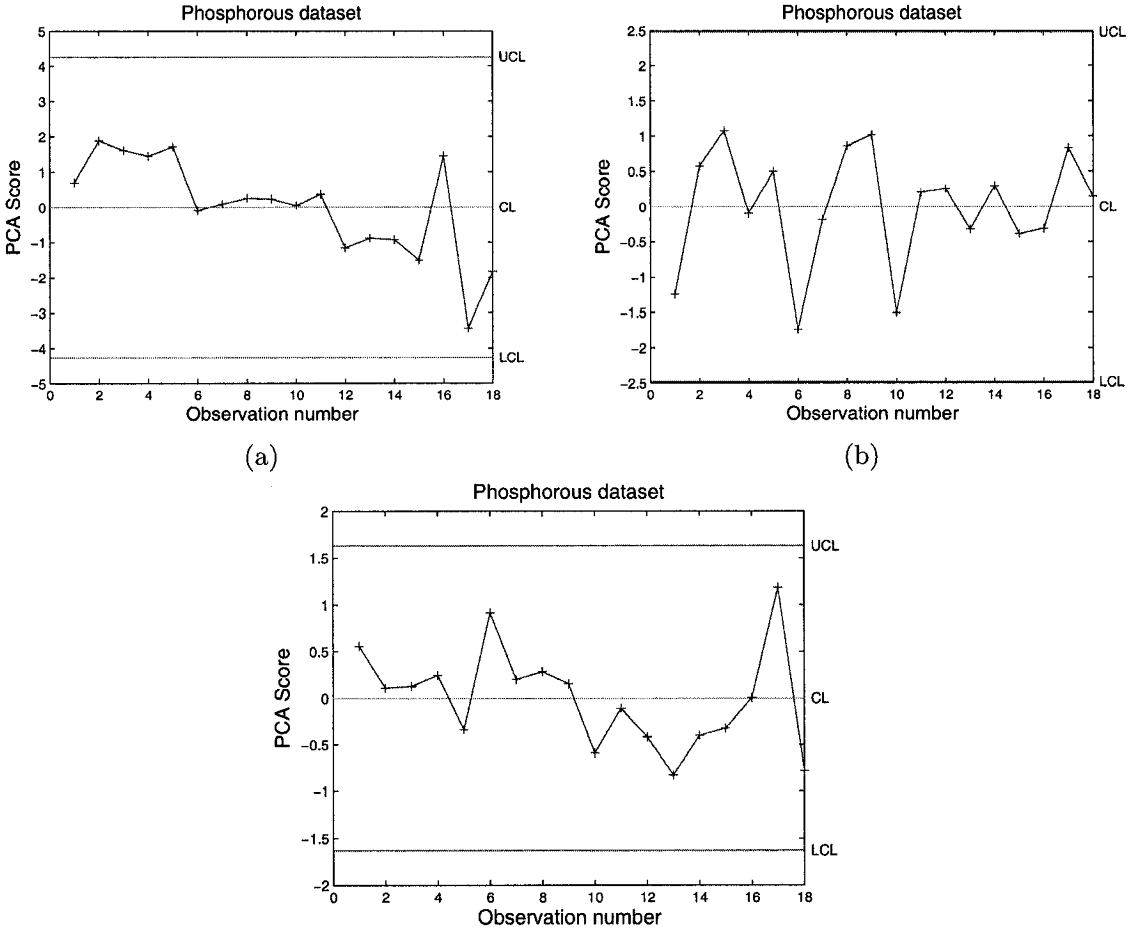


Figure 2.24: MVE T^2 control chart.

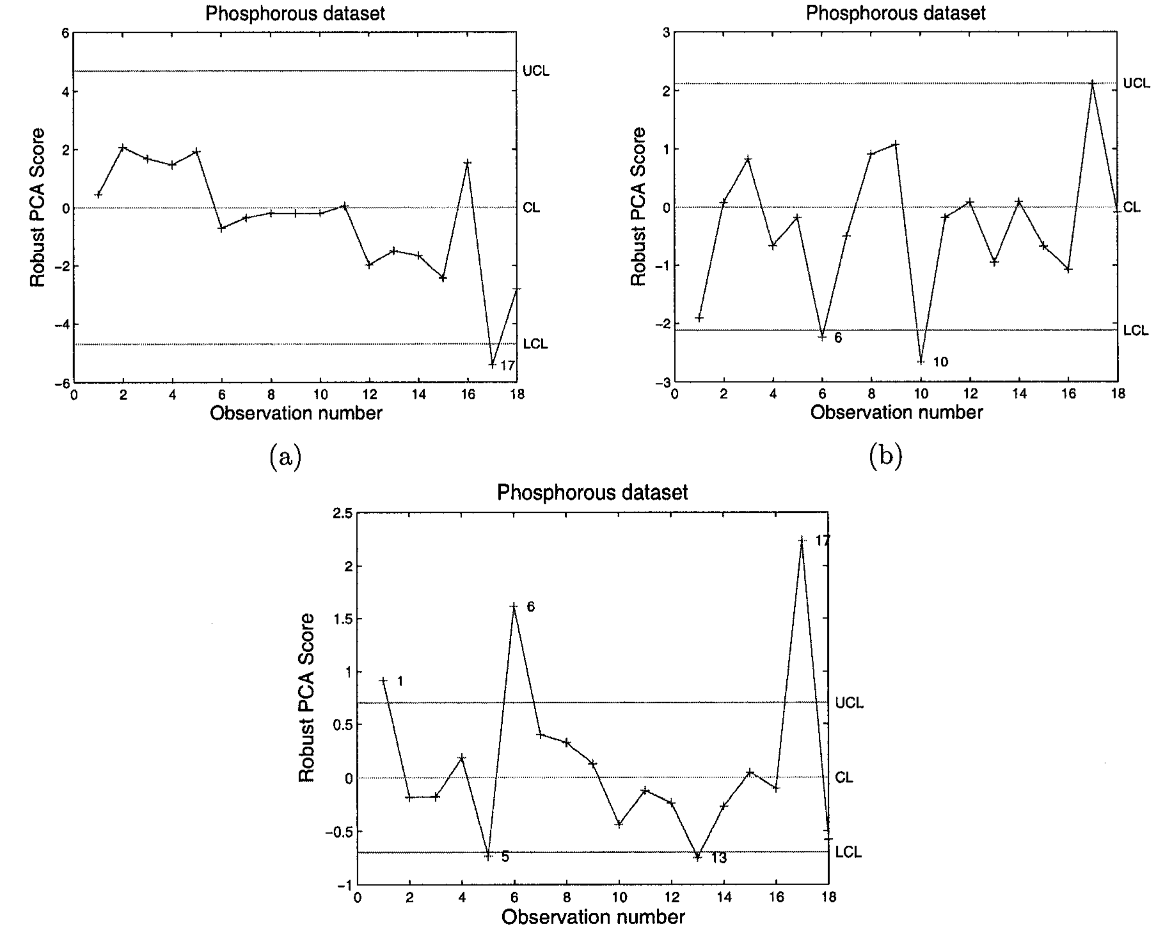


(a)

(b)

(c)

Figure 2.25: (a) PCA chart with 67.37% variance, (b) PCA chart with 22.83% variance, (c) PCA chart with 9.81% of variance.



(a)

(b)

(c)

Figure 2.26: (a) RPCA chart with 81.48% variance, (b) RPCA chart with 16.72% variance, (c) RPCA chart with 1.80% of variance.

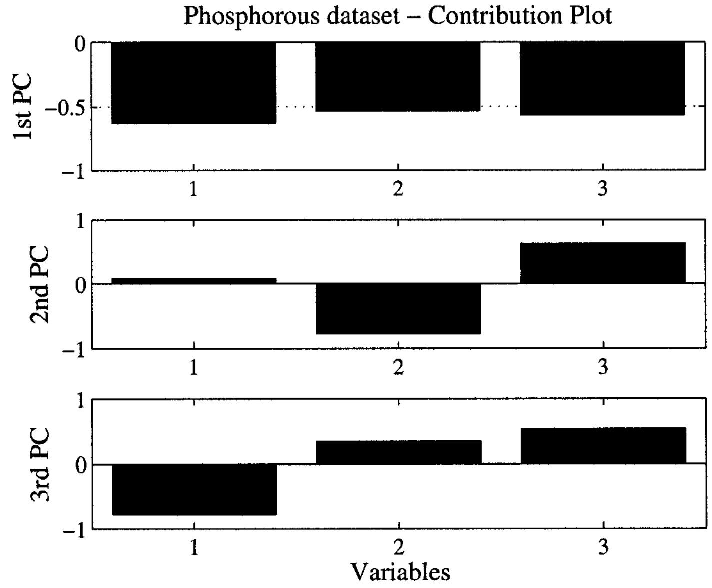


Figure 2.27: Contribution plot for all 3 principal components.

CHAPTER 3

Statistical Process Control using Kernel Principal Component Analysis

We present a robust multivariate statistical process control chart using kernel principal component analysis. The proposed control chart is effective in the detection of outliers, and its control limits are derived from the eigenanalysis of the Gaussian kernel matrix in the Hilbert feature space [53,54]. Our experimental results show the much improved performance of the proposed control chart in comparison with existing multivariate monitoring and controlling charts.

3.1 Introduction

Typically process monitoring applies to systems or processes in which only one variable is measured and tested. One of the disadvantages of a univariate monitoring scheme is that for a single process, many variables may be monitored and even controlled [3]. Multivariate quality control methods overcome this disadvantage by monitoring several variables simultaneously [4]. Using multivariate quality control methods, engineers and manufacturers who monitor complex processes may monitor the stability of their process.

With multivariate situations, the probability that a process is completely in control is less than in the univariate case [3]. Similarly, the likelihood that a multivariate process is completely out of control is less than that of the univariate case. Using multivariate control charts, it is possible to maintain a specific error rate, while taking advantage of cross correlation between the variables, and the process can be analyzed for its stability without the complication of maintaining many control charts at once. Multivariate quality control provides a way for engineers and manufacturers to test their products in an environment that provides many advantages over univariate models. It is inherently more complex than univariate statistical process control, but it may be a more realistic representation of the data since in the real world processes do not usually have only one variable that is measured independent of all other variables in a system. Mapping a multivariate situation as a univariate may lead to results where processes might seem to be in control when in fact they are not or vise versa.

In this chapter, we present a new multivariate statistical process control chart using kernel principal component analysis [19]. The proposed control chart is robust to outliers detection, and its control limits are derived from the eigen-analysis of the Gaussian kernel matrix in the Hilbert feature space.

The remainder of the chapter is organized as follows. In the next Section, we described the problem formulation. In section 3.3, we propose a kernel principal component control chart. In Section 3.4, we perform experimental results to demonstrate that the performance of the proposed multivariate control chart chart has greatly been improved in comparison with existing monitoring and controlling charts. Finally, some conclusions are included in Section 3.5.

3.2 Problem formulation

In recent years, a variety of statistical quality control methods have been proposed to monitor multivariate data including the Hotelling's T^2-statistic chart [3], and the principal component analysis control chart based on principal component analysis [30]. These control charts are widely used in the industry particularly in assembly operations and chemical process control [4]. The T^2 statistic is, however, vulnerable to outliers and in order to obtain significant good results both the mean and the covariance matrix must be robustly estimated [7,8,16,17]. Also, principal component analysis is very sensitive to outliers [4,18]. Therefore, to obtain significant results, extra processing is required using robust statistics.

3.3 Proposed method

Kernel principal component analysis is a nonlinear generalization of PCA, and consists in mapping the data to a higher (possibly infinite) dimensional feature space via a nonlinear map, and then computing the dot products in the feature space. Suppose we have an input data set $X = \{\boldsymbol{x}_i : i = 1, \dots, m\}$ where each observation $\boldsymbol{x}_i$ is an p-dimensional vector and the distribution of the data is nonlinear. Kernel PCA algorithm consists of two main steps: the first step is to linearize the distribution of the input data by using a nonlinear mapping $\Phi : \mathbb{R}^p \to \mathcal{F}$ from the input space $\mathbb{R}^p$ to a higher-dimensional (possibly infinite-dimensional) feature space $\mathcal{F}$. The mapping Φ is defined implicitly, by specifying the form of the dot product in the feature space. In other words, given any pair of mapped data points, the dot product is defined in terms of a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$.

The most commonly used kernels are the Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/(2\sigma^2))$ with parameter σ. In the second step, PCA in applied to the mapped data set $\Phi = \{\Phi_i : i = 1, \dots, m\}$ in the feature space, where $\Phi_i = \Phi(\boldsymbol{x}_i)$. The second step of kernel PCA is to apply PCA in the feature space by performing an eigendecomposition on the covariance matrix of the mapped data which is given by

$$C = \frac{1}{m-1} \sum_{i=1}^{m} \widetilde{\Phi}(\boldsymbol{x}_i)^T \widetilde{\Phi}(\boldsymbol{x}_i)$$

where $\widetilde{\Phi}(\boldsymbol{x}_i) = \Phi(\boldsymbol{x}_i) - (1/m) \sum_{i=1}^{m} \Phi(\boldsymbol{x}_i)$ is the centered mapped data.

The eigenvectors of C are given by

$$\boldsymbol{v} = \frac{1}{\mu} C \boldsymbol{v} = \sum_{i=1}^{m} \widetilde{\Phi}(\boldsymbol{x}_i) \left(\frac{1}{\mu(m-1)} \widetilde{\Phi}(\boldsymbol{x}_i)^T \boldsymbol{v} \right) = \sum_{i=1}^{m} \alpha_i \widetilde{\Phi}(\boldsymbol{x}_i),$$

where $\alpha_i = (\widetilde{\Phi}(\boldsymbol{x}_i)^T \boldsymbol{v})/(\mu(m-1))$. In other words, an eigenvector of C is a linear combination of $\{\widetilde{\Phi}(\boldsymbol{x}_i)\}$. Taking the dot product of $\widetilde{\Phi}(\boldsymbol{x}_j)$ with $\boldsymbol{v}$ yields

$$\widetilde{\Phi}(\boldsymbol{x}_j) \cdot \boldsymbol{v} = \sum_{i=1}^{m} \alpha_i \widetilde{\Phi}(\boldsymbol{x}_i) \cdot \widetilde{\Phi}(\boldsymbol{x}_j) = \sum_{i=1}^{m} \alpha_i \widetilde{K}_{ij},$$

which implies that $\mu(m-1)\alpha_j = \sum_{i=1}^{m} \alpha_i \widetilde{K}_{ij}$. Hence

$$\widetilde{K} \boldsymbol{\alpha} = \tilde{\mu} \boldsymbol{\alpha},$$

Algorithm: Kernel Principal Component Control Chart

1. Choose the appropriate σ for the Gaussian kernel matrix
2. Construct the kernel matrix $K = (K_{ij})$ of the mapped data: $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$.
3. Construct the kernel matrix $\widetilde{K} = HKH$ of the centered mapped data, where $H = I - J/n$ the centering matrix is defined in terms of the identity matrix I and the matrix of all ones J.
4. Find the largest p eigenvectors $\boldsymbol{\alpha}_r$ $(r = 1, \ldots, p)$ of $\widetilde{K}$ and their corresponding eigenvalues $\tilde{\mu}_r$.
5. Given a test point $\boldsymbol{x}$ with image $\Phi(\boldsymbol{x})$, compute the projections onto the eigenvectors $\boldsymbol{v}_r$ given by the equation
$$\boldsymbol{v}_r \cdot \widetilde{\Phi}(\boldsymbol{x}) = \frac{1}{\sqrt{(n-1)}} \sum_{i=1}^{m} \alpha_i \widetilde{\Phi}(\boldsymbol{x}_i) \cdot \widetilde{\Phi}(\boldsymbol{x})$$

Table 3.1: Algorithmic steps for Kernel PCA.

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ and $\tilde{\mu} = \mu(m-1)$. That is, $\boldsymbol{\alpha}$ is an eigenvector of $\widetilde{K}$. If the eigenvectors of C are orthonormal (i.e. $\boldsymbol{v}^T\boldsymbol{v} = 1$) then

$$\begin{aligned} 1 = \boldsymbol{v}^T\boldsymbol{v} &= \sum_{i,j=1}^{m} \alpha_i\alpha_j \widetilde{\Phi}(\boldsymbol{x}_i) \cdot \widetilde{\Phi}(\boldsymbol{x}_j) = \sum_{i,j=1}^{m} \alpha_i\alpha_j \widetilde{K}_{ij} \\ &= \boldsymbol{\alpha}^T \widetilde{K} \boldsymbol{\alpha} = \mu(m-1)\boldsymbol{\alpha}^T\boldsymbol{\alpha} \end{aligned}$$

and hence $\|\boldsymbol{\alpha}\| = 1/\sqrt{\mu(m-1)}$.

The main algorithmic step of the proposed kernel principal component chart as shown in Table 3.1.

Assuming we want $\pm 3\sigma$ confidence intervals, the upper control limit (UCL), the center line (CL), and the lower control limit (LCL) of the kernel principal component chart are

$$UCL = +3\sqrt{\mu_r}$$

$$CL = 0$$

$$LCL = -3\sqrt{\mu_r}$$

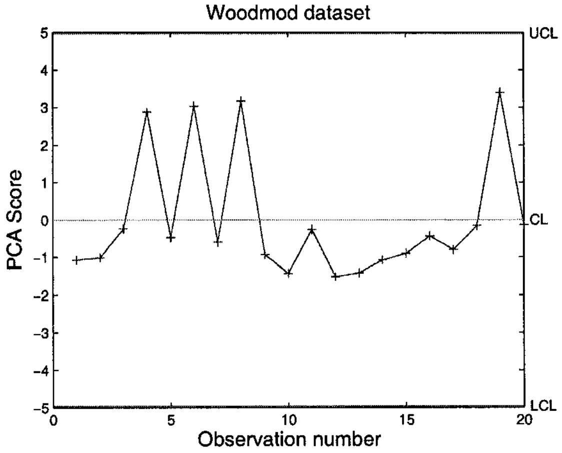


Figure 3.1: Principal component chart with 54.8% of variance.

3.4 Experimental results

We conducted experiments on three different data sets. In all the experiments, the width of the Gaussian kernel is estimated as follows

$$\sigma = \frac{2}{m(m-1)} \sum_{i<j}^{m} \|z_i - z_j\|,$$

where $[z_1, \ldots, z_m]^T$ is the standardized data. Also, we compare the results between linear PCA and kernel PCA using the same experiments used in chapter two.

3.4.1 Experiment #1: Woodmod dataset

The linear principal component control chart is unable to detect outliers as depicted in Fig. 3.1. We can clearly see that the observations 4, 6, 8 and 19 have higher variations than the rest of the observations although they still lie within the upper and lower control limits.

The kernel principal component chart is able to detect the observations 4, 6, 8 and 19 as outliers as shown in Fig. 3.2 containing 44.2% of the variance.

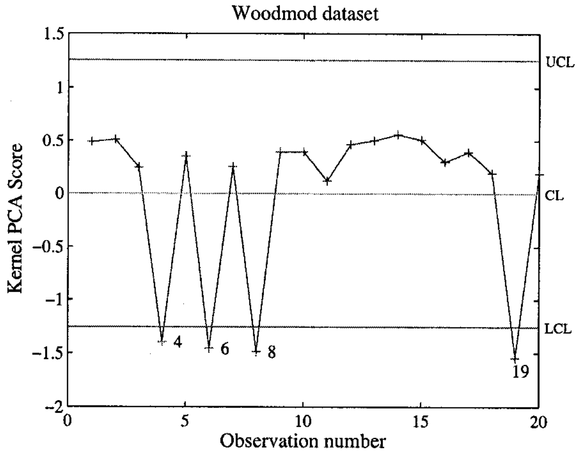


Figure 3.2: Kernel principal component chart with 44.2% of variance

3.4.2 Experiment #2: Stackloss dataset

The principal component chart, did not detect any outliers as shown in Fig. 3.3. On the other hand, kernel principal component chart (see Fig. 3.4) was able to identify the first 3 observations as outliers.

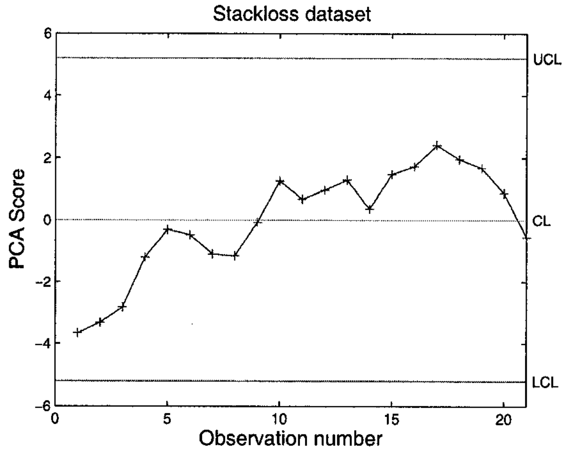


Figure 3.3: Principal component chart with 74.94% of variance.

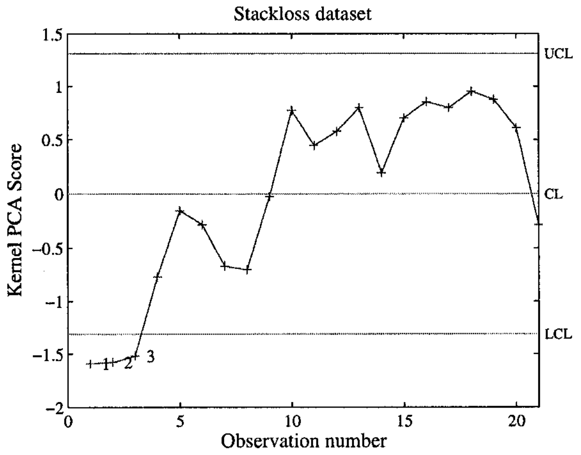


Figure 3.4: Kernel principal component chart with 49.76% of variance.

3.4.3 Experiment #3: Phosphorus content dataset

For the phosphorus content data, the linear principal component chart did not identify any outliers as illustrated in Fig. 3.5. Kernel principal component chart was, however, able to detect the observation 17 as an outlier as shown in Fig. 3.6.

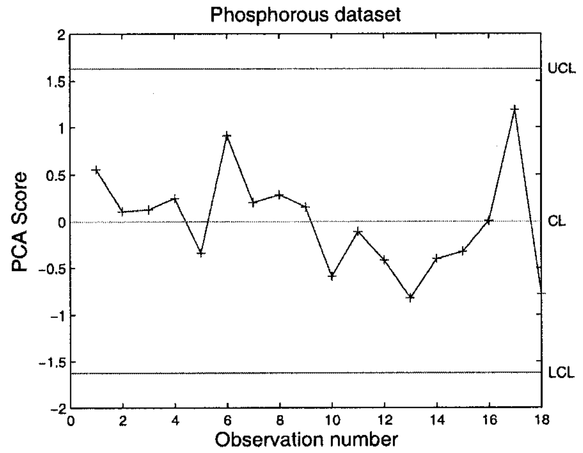


Figure 3.5: Principal component chart with 9.81% of variance.

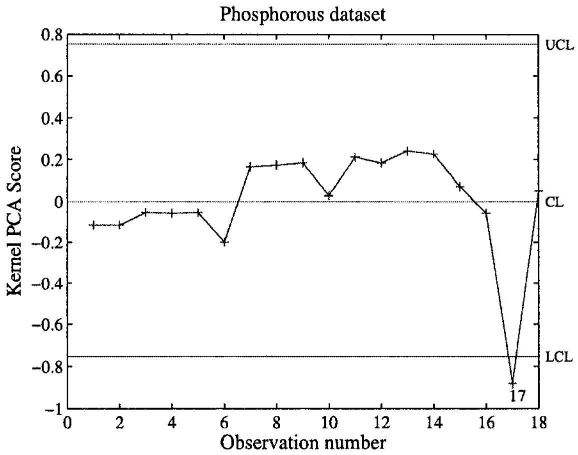


Figure 3.6: Kernel principal component chart with 16.18% of variance.

3.5 Conclusions

In this chapter, we introduced a new multivariate control chart by using the concept of kernel principal component analysis. The core idea behind our proposed technique is to project the data into a higher dimension Hilbert space in order to extract the eigenvalues and eigenvectors of a Gaussian kernel matrix. The experimental results clearly show a much improved performance of the proposed approach in comparison with the current multivariate control charts.

CHAPTER 4

Cluster Principal Component Analysis for Outlier Detection

We introduce a new method to detect multiple outliers in high-dimensional datasets using the concepts of hierarchical clustering and principal component analysis. The proposed cluster PCA algorithm is computationally fast and robust to outliers detection [55]. A comparative study with existing techniques is performed on both synthetic and real-world datasets. Our experimental results demonstrate an improved performance of our algorithm in comparison with existing multivariate outlier detection schemes.

4.1 Introduction

With fast automated data collection tools and larger databases, there is a tremendous amount of information that is stored for future analysis [22]. Consequently, the need to develop tools and techniques to extract relevant information has made research areas such as data mining increasingly important [23]. Outlier detection is among these research areas that has attracted considerable attention in recent years [24]. Outliers are defined as abnormal data points which deviate from the normal variability found in a dataset. These outliers are often of primary interest in both chemical and engineering related processes [25]. For example, in geochemical exploration, outliers can often identify important mineral deposits [26].

In recent years, various techniques have been proposed for outlier detection in both univariate and multivariate settings [7,26,27]. These methods typically fall under two categories: supervised and unsupervised approaches. The supervised approaches compare new observations under an existing well defined model, whereas the unsupervised approaches classify each observation under normal and extreme variation based on a certain distance [28].

In this chapter, we present a new distance-based approach which we refer to as cluster principal component analysis (cluster PCA). The goal of the proposed method is to identify outliers in both low and high dimensional datasets by combining the concepts of hierarchical clustering [29] and PCA [30] in order to improve the performance while keeping the complexity and computation time relatively low.

The layout of this chapter is as follows. In the next section, we describe the problem formulation. In section 4.3 introduces the proposed robust multivariate algorithm. In Section 4.4, and section 4.4 experimental and simulated results are presented to demonstrate the performance of the proposed approach in comparison with existing robust techniques. Section 4.6 concludes the chapter.

4.2 Problem formulation

Mining information from a dataset containing multiple dimensions is becoming very common [31]. Many industries use univariate techniques to study each dimension. The univariate approaches may lead faulty results since it takes little or no account of the covariance that exist between the observations [3]. To overcome this problem, multivariate control charts are used. This is not trivial when a dataset contains multiple outliers. Even a small percentage of outliers can distort the results and render the outcome misleading or useless. To overcome this problem, statisticians have recently proposed robust methods to estimate key parameters such as the mean and covariance/correlation matrix without the negative effect of outliers [32]. Techniques such as the minimum volume ellipsoid (MVE) and minimum covariance determinant (MCD) have proven their robustness but are limited to small moderate dimensions [25,27].

For multivariate data, the process of finding meaningful outliers becomes inherently more complex [31]. For example, in the field of chemometrics, datasets containing thousands of dimensions are not uncommon. For these types of applications, projection pursuit (PP) and PCA may be used to process and analyze such large information [25].

4.3 Proposed method

Most statistical-based techniques use the covariance matrix as the basis for detecting outliers in datasets [27]. For example, the MCD and MVE methods use the volume (determinant) of the covariance matrix to identify the subset of observations that are considered robust to calculate both the mean and the covariance matrix. This process might result very lengthy because the optimum subset might require $n = m!/(m-h)!$ permutations, where h is cardinality of the optimum subset of observations. Also, the determinant of the covariance matrix can only be computed if $p < h$, otherwise the determinant will be equal to zero. This is the main reason why some of the more robust algorithms are limited to only a small value of p (i.e. small number of dimensions) [25].

Our proposed algorithm, however, does not depend on a the number of permutations and can be applied to both high (large p) and low dimensional (small p) datasets. Our approach is based on hierarchical clustering in combination with PCA to obtain a robust subset of observations that are used to identify outliers. PCA is a method for transforming the observations in a dataset into new observations which are uncorrelated with each other and account for decreasing proportions of the total variance of the original variables. Each new observation is a linear combination of the original observations. Standardizing the data is often preferable when the variables are in different units or when the variance of the different columns of the data is substantial. The standardized data matrix is given by

$$Z = (X - \mathbf{1}\,\bar{\boldsymbol{x}})D^{-1/2} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m]^T,$$

where $\mathbf{1} = (1, \ldots, 1)^T$ is a $m \times 1$ vector of all 1's, and $D = (\operatorname{diag}(S))^{1/2}$ is the diagonal standard deviation matrix.

The Euclidean distance matrix between all the standardized observations is given by

$$d_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|, \quad 1 \leq i, j \leq m.$$

Initially, each observation can be considered a cluster of its own until all the h observations are eventually integrated in one big cluster. If $m > p$, the optimal subset contains $h = \lfloor (m+p+1)/2 \rfloor$

observations, and if $m < p$ then the optimal subset contains $h = \lfloor \alpha m \rfloor$, where $\alpha \in (1/2, 1)$ is a parameter and $\lfloor x \rfloor$ denotes the floor function that returns the largest integer less than or equal to x. A smaller value of α tends to increases the robustness of the algorithm whereas a higher value of α tends to give better estimates of the uncontaminated data [25,33].

Given a set of h observations to be clustered and an $h \times h$ distance matrix $D = (d_{ij})_{1 \leq i,j \leq h}$, the hierarchical clustering is performed as follows [29]:

1. Assign each observation to a cluster so that we have h clusters, each containing one cluster. Let the distances between the clusters be the same as the distances between the observations they contain.

2. Find the closest pair of clusters and merge them into a single cluster, so that we have one cluster less.

3. Compute distance between the new cluster and each of the old clusters. The distance between one cluster and another cluster is equal to the shortest distance from any member of one cluster to any member of the other cluster.

4. Repeat steps 3) and 4) until all observations are clustered into a single cluster of size h.

Once the sample $H = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_h]^T$ of the h observations are selected, we compute its robust sample mean $\overline{\boldsymbol{z}}_{\boldsymbol{H}}$ and its robust sample covariance matrix S_H. Then, we apply the eigendecomposition on S_H, that is $S_H = A \Lambda A^T$, where A is a matrix of eigenvectors (principal components) and Λ is a diagonal matrix of eigenvalues. We may select the most significant k principal components according to the following criteria

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \geq 90\%$$

which can be used as reasonable cut-off value. The robust distance in the PCA subspace is then defined as

$$T_i^2 = (\boldsymbol{z_i} - \overline{\boldsymbol{z}}_{\boldsymbol{H}}) A_k {\Lambda_k}^{-1} A_k^T (\boldsymbol{z_i} - \overline{\boldsymbol{z}}_{\boldsymbol{H}})^T$$

where $A_k = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$ and $\Lambda_k = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$.

4.4 Experimental results

In this section, we test the performance of the proposed cluster PCA method with simulated and real-world data sets. We also compare the results with the previous methods discussed in the robust statistics section as well as the conventional T^2 and projection pursuit(T^2 PCA).

4.4.1 Hawkins-Bradu-Kass (HBK) dataset

We tested the performance of our proposed technique on the HBK data set [33]. This data set $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{75}]^T$ contains 75 observations where each observation $\boldsymbol{x}_i$ is 4-dimensional (i.e. has 4 variables). It is known that the first 14 observations are outliers [33].

To determine which observations are outliers, we used a cut-off line as the value of $\chi^2_{0.975,p}$ where $\chi^2_{\beta,v}$ denotes the value of the inverse chi-square cumulative distribution with v degrees of freedom evaluated at the value β. In the case for Cluster PCA and T^2 PCA we used $\chi^2_{0.975,k}$ where k represents the number of principal components selected [25].

The conventional Mahalanobis distance is shown in Fig. 4.1, where is can be clearely seen that the masking effect has limited its performance to only identifying the 4 largest outliers. The projection pursuit using PCA was also unable to identify the first 10 outliers as shown in Fig. 4.2. The more robust algorithm (Fig. 4.3 - Fig. 4.7) were all able to detect the correct outliers. It is also important to notice that the SW1 and the MCD techniques have also identified some smaller outliers. For the HBK data set, all observations other than 1 to 14 that are found outside the cut-off limit should be considered as a false alarms.

4.5 Simulation results

In this section, we test the performance of the proposed cluster PCA method on two datasets with different dimensions, and we also compare the results with the previous methods discussed in the Introduction.

The datasets are generated using the following Gaussian mixture model

$$(1-\varepsilon)N_p(0,\Sigma) + \varepsilon N_p(\widetilde{\mu},\Sigma), \tag{1}$$

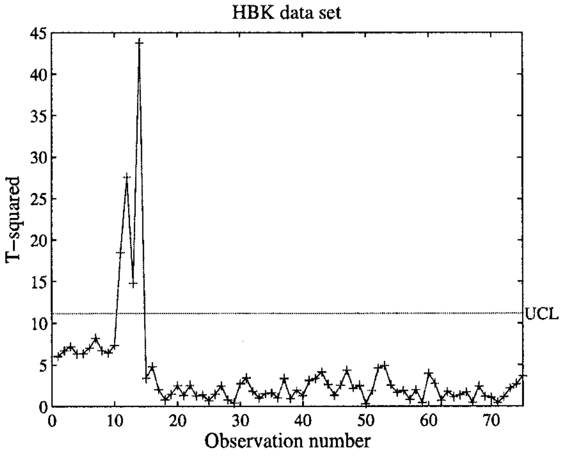


Figure 4.1: Conventional T^2 chart.

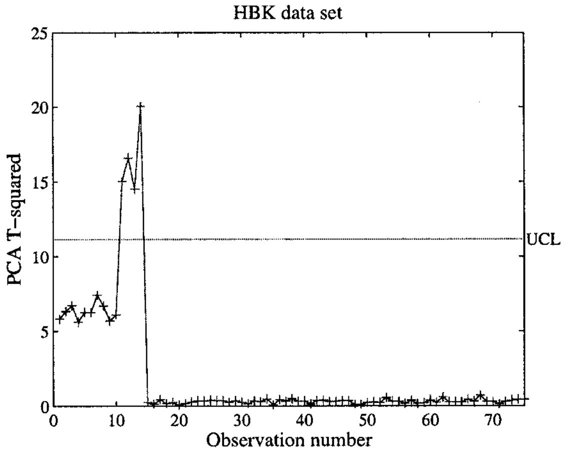


Figure 4.2: PCA T^2 chart with 90% of total variance.

where ε is the percentage of outliers, N_p denotes a p-variate Gaussian distribution, and $\widetilde{\mu}$ denotes the mean shift [25,34]. Therefore each observation is normally distributed at a certain mean and varied randomly at certain standard deviation. Moreover, the simulated data is restricted between $\pm 2\sigma$ in order to have better control of the uncontaminated data and to also avoid any extreme case scenario.

We varied the values of $m, p, \varepsilon, \widetilde{\mu}$ for different settings and repeated the experiment 50 times in order to achieve the best estimates. A detailed description of the experiment can be summarized

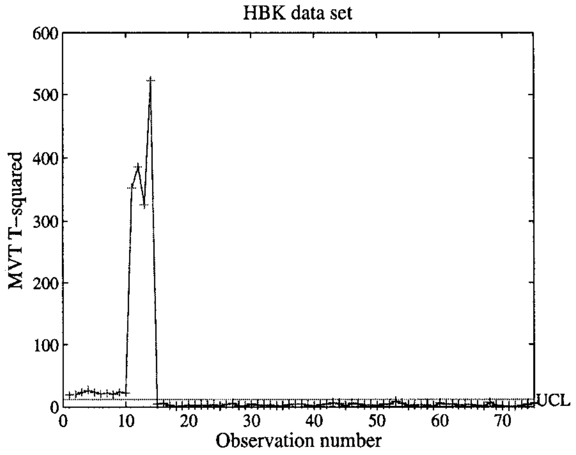


Figure 4.3: Multivariate trimming chart (removing 15% of observations.)

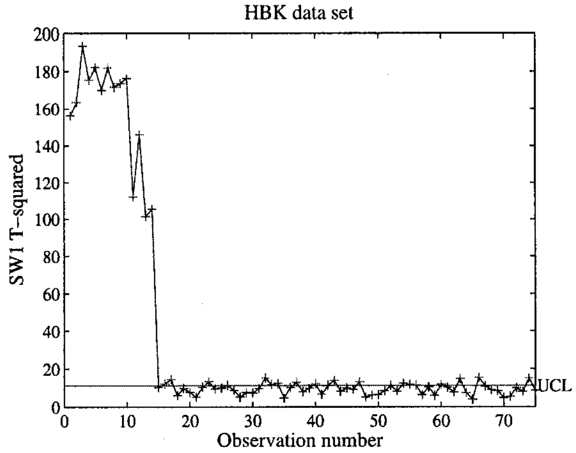


Figure 4.4: Sullivan and Woodall first approach chart.

as follows:

- Two datasets $X_{m \times p}$ with $(m, p) = (100, 4)$ and $(m, p) = (50, 100)$ are generated using the Gaussian mixture model defined in Eq. (1).
- The percentage of outliers ε was set to 0%, 10%, 20%, 30%, and 40%.
- The mean shift $\widetilde{\mu}$ was set to 0.1, 0.15, 0.2, 0.25, and 0.30. The standard deviation was consistently set to $\sigma = 0.1$ in order to get different signal to noise ratios.
- We tested the performance of the algorithms under two criteria: (i) the percentage of

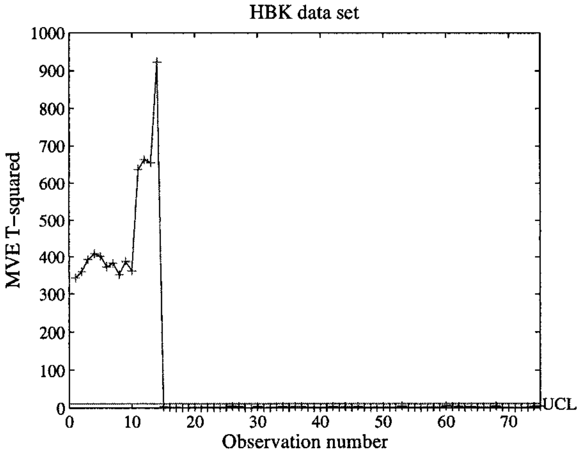


Figure 4.5: Minimum volume ellipsoid chart.

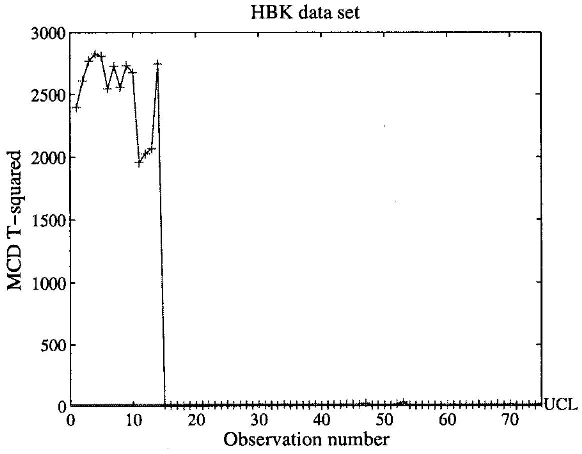


Figure 4.6: Minimum covariance determinant char.t

correct outliers detection (# of correct outliers found / total # of outliers) and (ii) the false alarm ratio (# of observations found as outliers but are not / total # of good observations). We also tested the performance for both consecutive mean shift and scattered mean shift across the observations.

- The cardinality h of the optimum subset of observations was set to $\lfloor h = [(m+p+1)/2\rfloor$ when $m \geq p$, and to $h = \lfloor m/2 \rfloor$ when $m < p$.

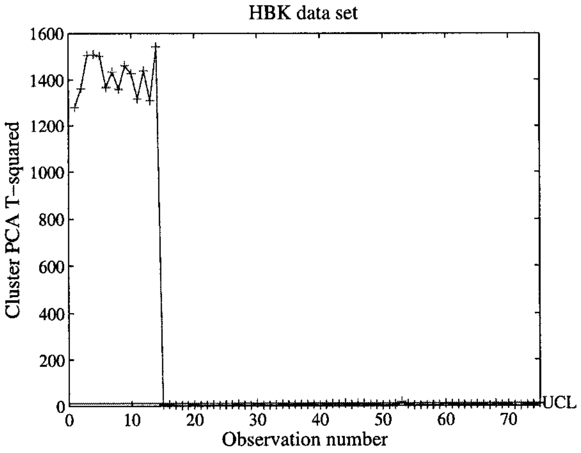


Figure 4.7: Cluster PCA chart using 90% of total variance.

4.5.1 First dataset

In this subsection, we perform a simulation study on the first Gaussian-mixture generated dataset $X_{m \times p}$ with $(m, p) = (100, 4)$, and we compare the performance of the proposed approach with several robust methods discussed in the Introduction.

Performance of conventional T^2 algorithm

As observed in Fig. 4.8-(a), the conventional Mahalanobis distance is limited to very low percentage of outliers. In this case, a consecutive mean shift or a scattered mean shift has no effect on the results. Also, the false alarm performance (Fig. 5.5-(b)) is rated at less than 1%.

Performance of the MVT algorithm

For the multivariate trimming algorithm, we tested its performance by removing 15% and 25% of the observations. As shown in Fig. 4.9 and Fig. 4.10, this method is limited to relatively low percentage of outliers (10% to 20%). Also, the more the observations we remove the better outlier detection performance we achieve to the cost of a higher (increased by a factor of two) false alarm ratio. It is also important to note that there is no performance difference between a consecutive mean shift and a scatter mean shift as can be observed in Fig. 5.8.

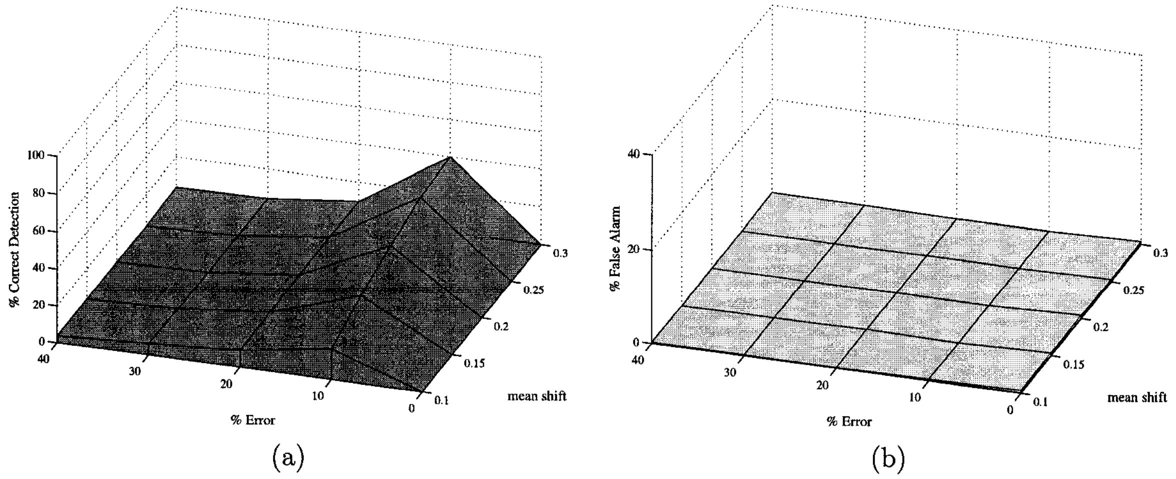


Figure 4.8: (a) Conventional T^2 outlier detection performance in a consecutive mean environment, (b) Conventional T^2 false alarm performance in a consecutive mean environment.

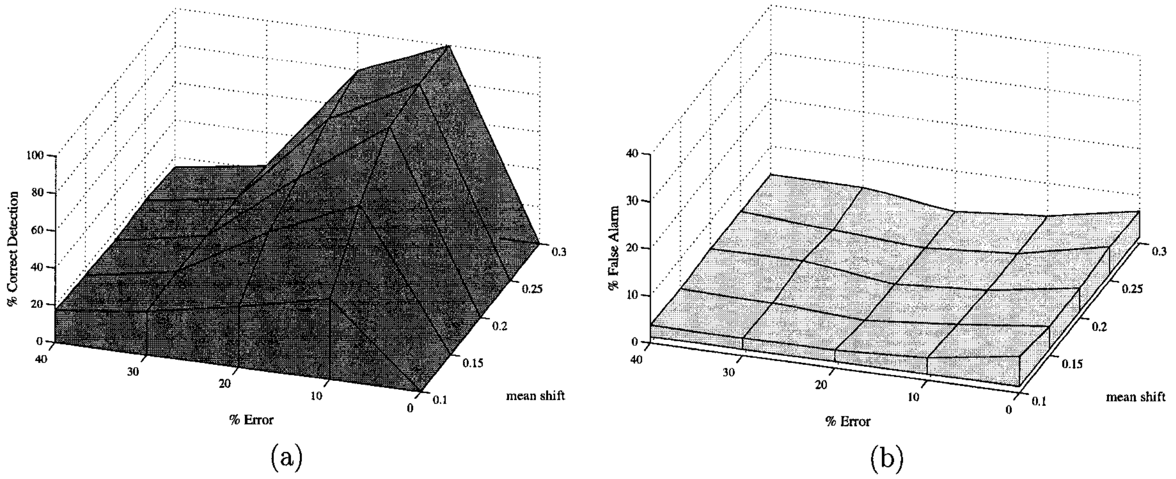
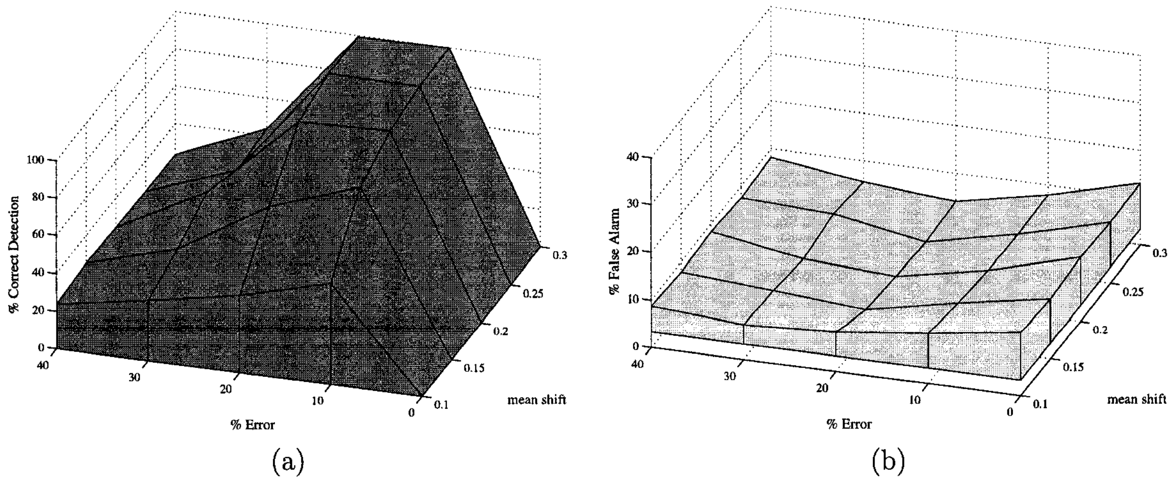
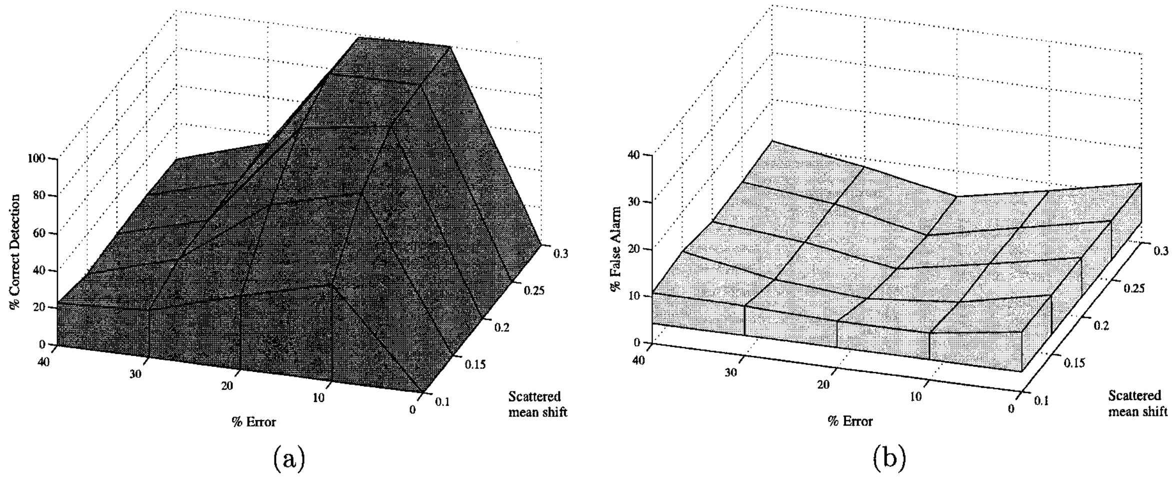


Figure 4.9: (a) MVT outlier detection performance when removing 15% of the observations in a consecutive mean environment, (b) MVT false alarm performance when removing 15% of the observations in a consecutive mean environment.



(a) (b)

Figure 4.10: (a) MVT outlier detection performance when removing 25% of the observations in a consecutive mean environment, (b) MVT outlier detection performance when removing 25% of the observations in a scattered mean environment.

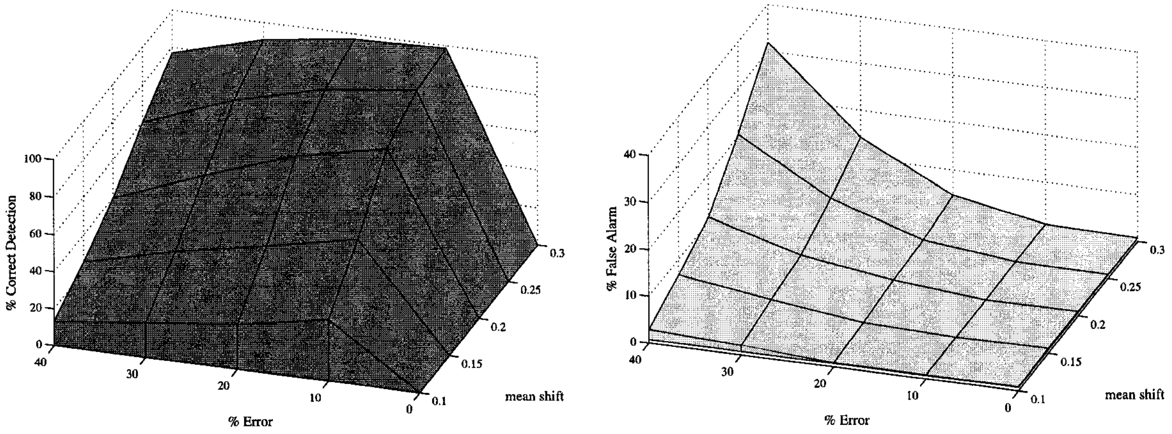


(a) (b)

Figure 4.11: (a) MVT outlier detection performance when removing 25% of the observations in a scattered mean environment, (b) MVT false alarm performance when removing 25% of the observations in a scattered mean environment.

Performance of the SW1 algorithm

The performance of Sullivan and Woodall approach is depicted in Fig. 4.12. As observed, this method works relatively well only in the presence of a consecutive mean shift. If the environment was a scattered mean shift this methods would fail to detect the correct outliers as depicted in Fig. 5.10. Also, we can observe that the false alarm performance increases exponentially as the percentage of errors and mean shift increases.

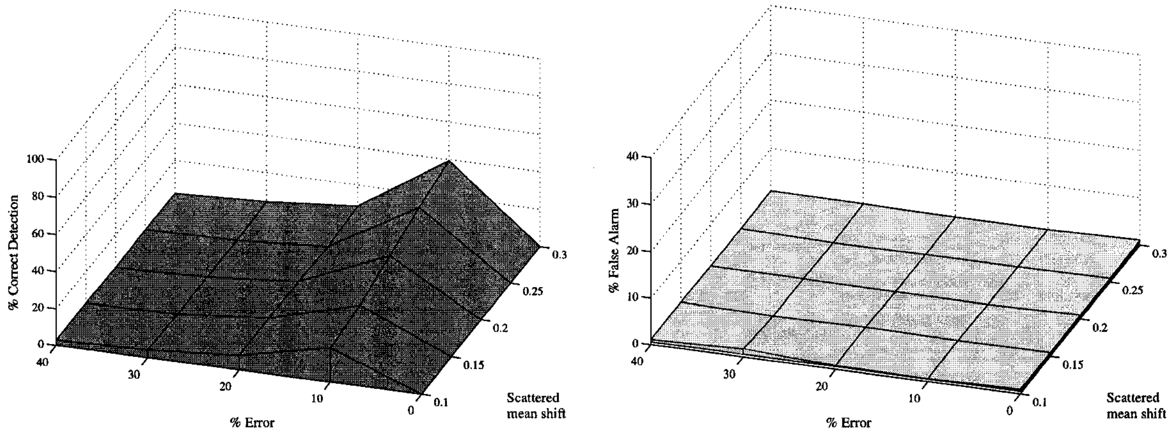


(a) (b)

Figure 4.12: (a) SW1 outlier detection performance in a consecutive mean environment, (b) SW1 false alarm performance in a consecutive mean environment.

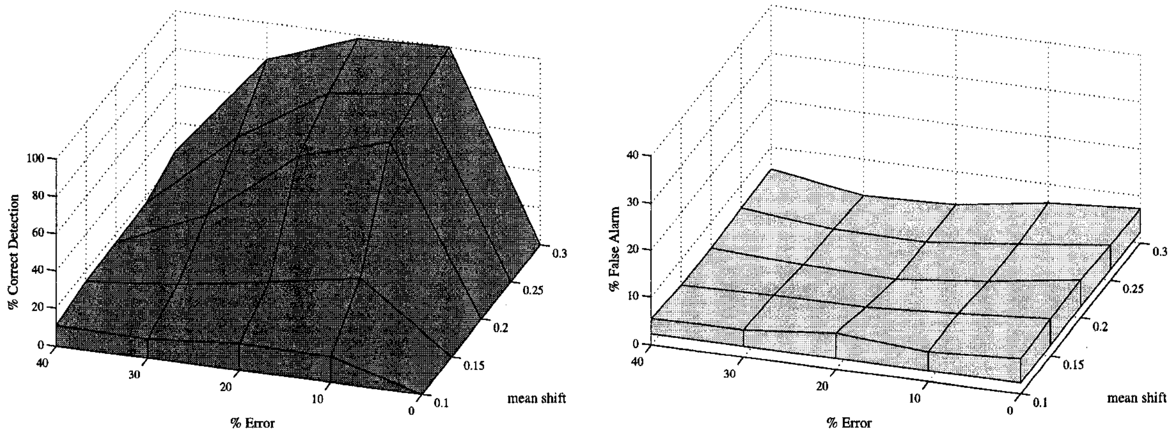
Performance of the MVE algorithm

The minimum volume ellipsoid performance is shown in Fig. 4.14 and Fig. 4.15. It can be noted that this procedure works well under large mean shifts and small percentage of outliers. Also, it slightly performs better in the scattered mean environment particularly when $\widetilde{\mu} \approxeq 0.1$. Finally, the false alarm ratio is contained under 10% in both environments.



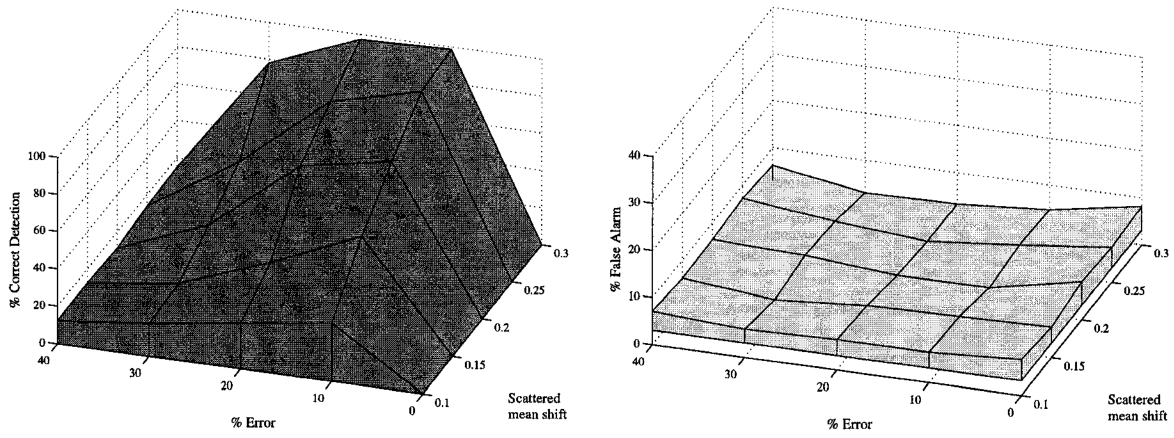
(a) (b)

Figure 4.13: (a) SW1 outlier detection performance in a scattered mean environment, (b) SW1 false alarm performance in a scattered mean environment.



(a) (b)

Figure 4.14: (a) MVE outlier detection performance in a consecutive mean environmen, (b) MVE false alarm performance in a consecutive mean environment.



(a) (b)

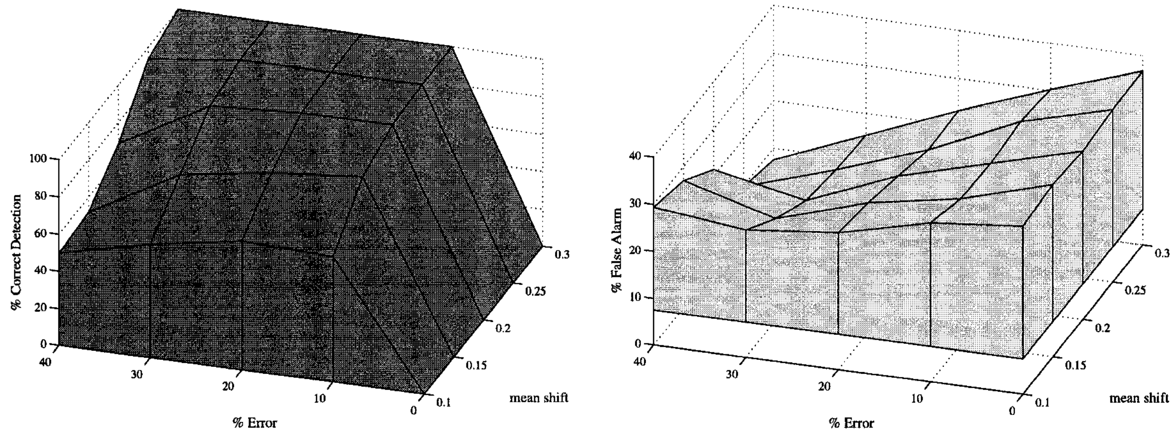
Figure 4.15: (a) MVE outlier detection performance in a scattered mean environment, (b) MVE false alarm performance in a scattered mean environment.

Performance of the MCD algorithm

The minimum covariance determinant is the most robust algorithm for low dimensional data as shown in Fig. 4.16 and Fig. 5.13. It works very well across all percentage of errors and mean shifts. We also obtain similar results for both scattered and consecutive mean environments. Where it lacks the most is the false alarm ratio. As observed, we obtain a performance of anywhere between 8% to 37%.

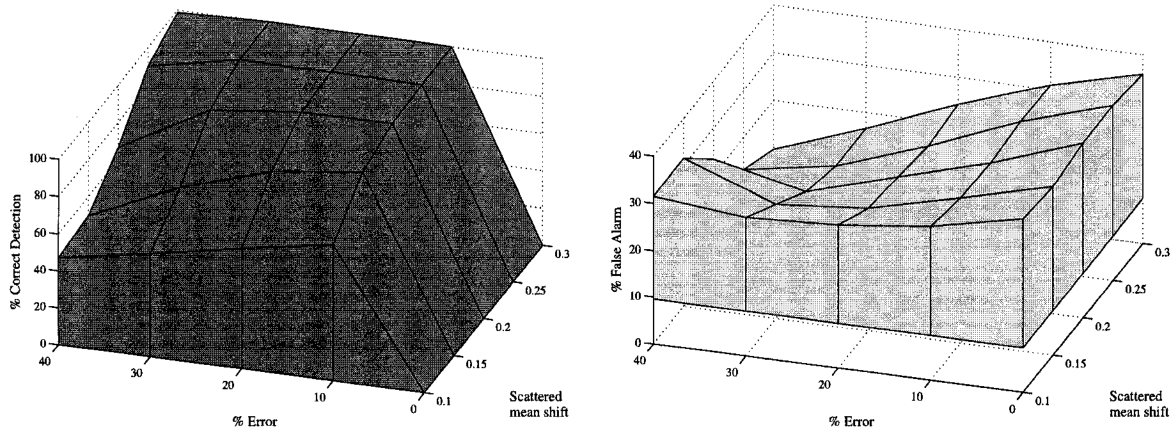
Performance of the T^2 PCA

For the projection pursuit method, we have the number of components selected as an extra parameter. In this case we varied k in such a way that 80% and 100% of the total variance would be selected. First, let us analyze the performance of a consecutive mean shift environment. As observed in Fig. 5.14 and Fig. 4.19 the higher the total variance is represented in projection pursuit, the better the outlier detection performance. In the case of a scattered mean shift environment there is no difference in the outlier detection performance as shown in Fig. 4.20 and Fig. 5.15. Also the false alarm ratio is below 1% in all case scenarios and environments as depicted in all Figures.



(a) (b)

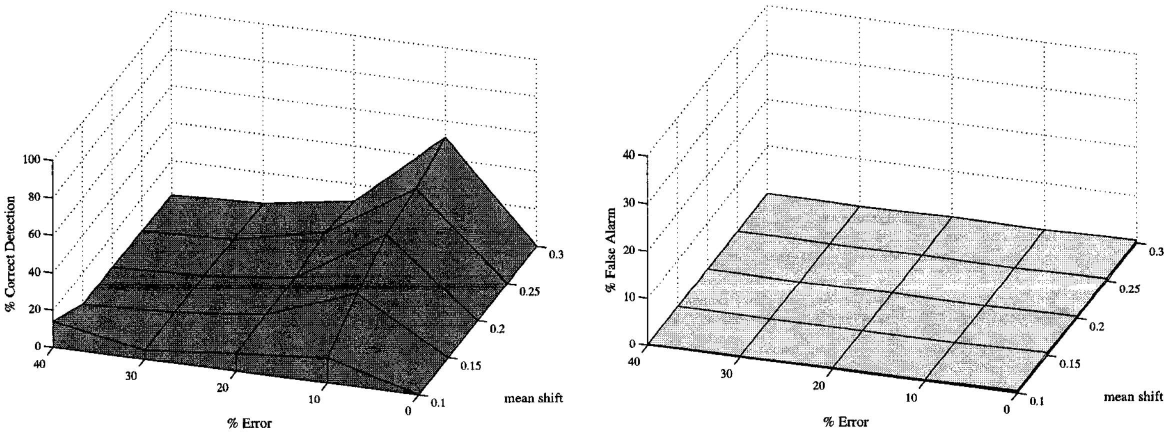
Figure 4.16: (a) MCD outlier detection performance in a consecutive mean environment, (b) MCD false alarm performance in a consecutive mean environment.



(a) (b)

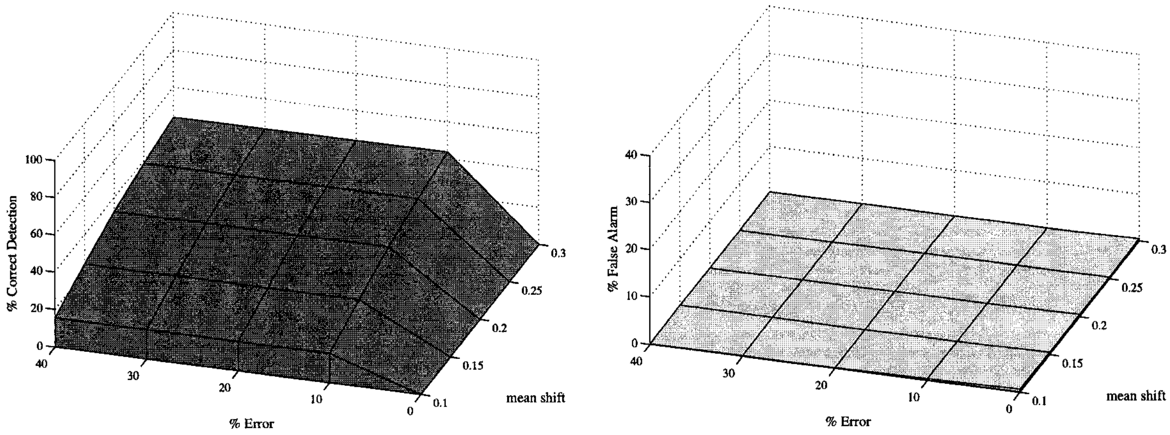
Figure 4.17: (a) MCD outlier detection performance in a scattered mean environment, (b) MCD false alarm performance in a scattered mean environment.

We can conclude that this method is restricted for small percentages of errors as the conventional T^2.



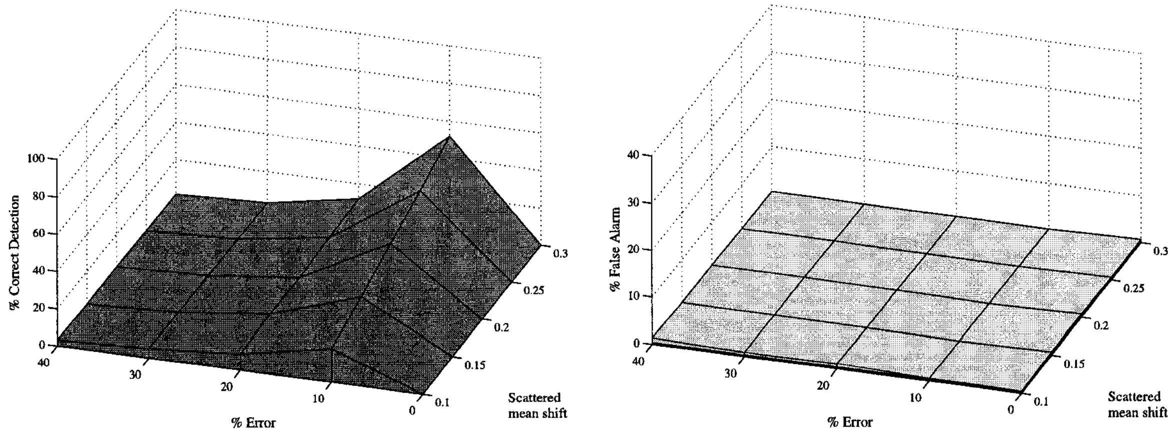
(a) (b)

Figure 4.18: (a) PCA T^2 outlier detection performance in a consecutive mean environment (80% of variance used), (b) PCA T^2 false alarm performance in a scattered mean environment (80% of variance used).



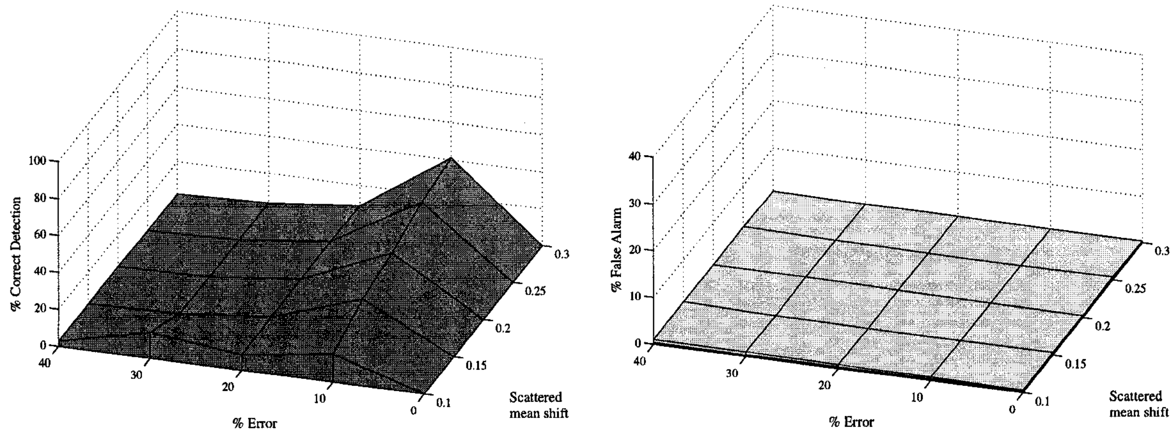
(a) (b)

Figure 4.19: (a) PCA T^2 outlier detection performance in a consecutive mean environment (100% of variance used), (b) PCA T^2 false alarm performance in a consecutive mean environment (100% of variance used).



(a) (b)

Figure 4.20: (a) PCA T^2 outlier detection performance in a scattered mean environment (80% of variance used), (b) PCA T^2 false alarm performance in a scattered mean environment (80% of variance used).



(a) (b)

Figure 4.21: (a) PCA T^2 outlier detection performance in a scattered mean environment (100% of variance used), (b) PCA T^2 false alarm performance in a scattered mean environment (100% of variance used).

Performance of the cluster PCA algorithm

Cluster PCA also has the number of components selected as a parameter. In this case, we also varied k in such a way that 80% and 100% of the total variance would be selected. Our results are similar to those of MCD in terms of outlier detection performance (as depicted in Fig. 4.22 to Fig. 4.24) in both consecutive and scattered mean shift environments. Where our algorithm outperforms MCD is in the false alarm category. The false alarm ratio lies at worst a the 20% range depending on parameters selected. Also, it is important to note that the more k components are selected the better the outlier detection performance is at the cost of higher false alarm ratio. This is due that as more information (variance) is added, there is also more noise added.

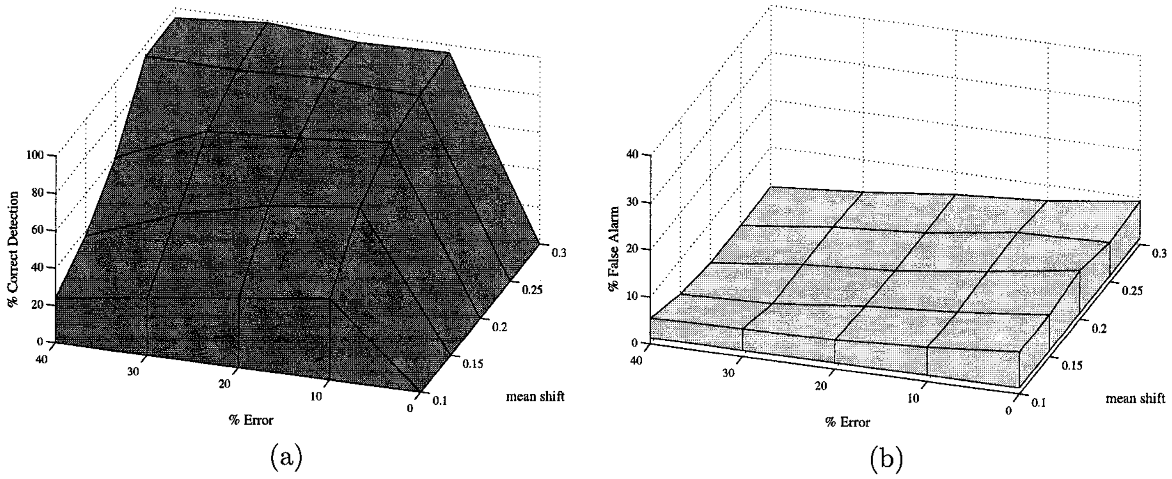
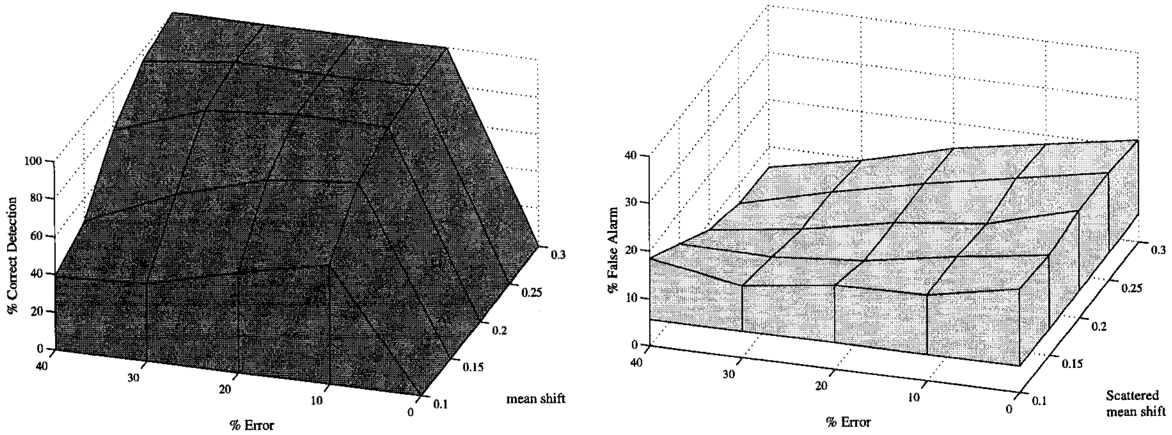


Figure 4.22: (a) Cluster PCA outlier detection performance in a consecutive mean environment (80% of variance used), (b) Cluster PCA false alarm performance in a consecutive mean environment (80% of variance used).

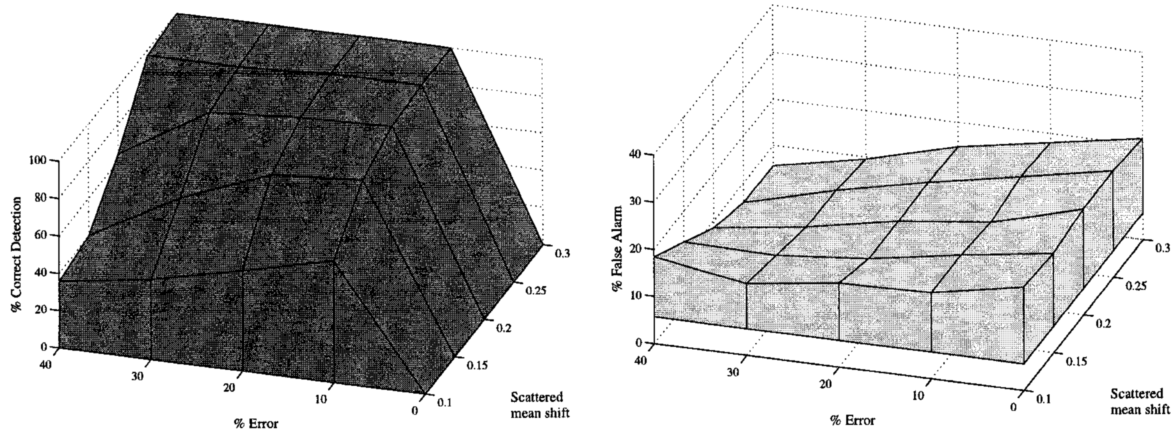
4.5.2 Second dataset

The main limitation of the MVE and MCD algorithms is their inapplicability to datasets having more variables than observations, that is when $p > m$. Other techniques such SW1 and multivariate trimming can be modified using the concepts behind principal component analysis and projection



(a) (b)

Figure 4.23: (a) Cluster PCA outlier detection performance in a consecutive mean environment (100% of variance used), (b) Cluster PCA false alarm performance performance in a consecutive mean environment (100% of variance used).



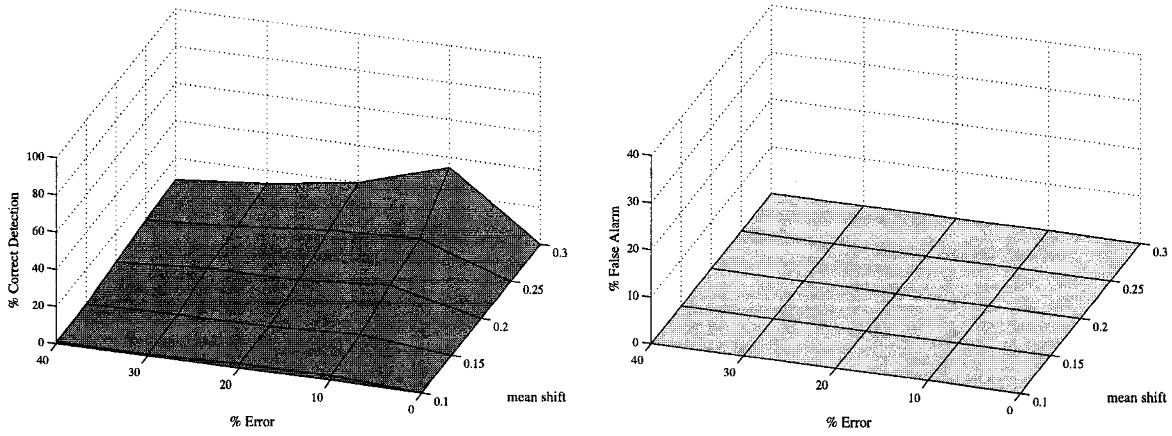
(a) (b)

Figure 4.24: (a) Cluster PCA outlier detection performance in a scattered mean environment (100% of variance used), (b) Cluster PCA false alarm performance performance in a scattered mean environment (100% of variance used).

pursuit to fit the model of higher-dimension. But, given their poor performance for low dimensional data, their results are expected to degrade even more. Our proposed cluster PCA is, however, applicable to such datasets as will be shown in the sequel. To this end, we generated a Gaussian-mixture dataset $X_{m \times p}$ with $(m, p) = (50, 100)$.

Performance of the T^2 PCA algorithm

Projection pursuit is observed in Fig. 4.25 and Fig. 4.26 in both consecutive and scattered mean shift environments. As shown, it is unable to detect the outliers present. Also, as we increase the number of component selected k, the performance for detecting outliers deteriorates as seen in Fig. 4.27. The False alarm ratio is less than 0.8% across all parameters.

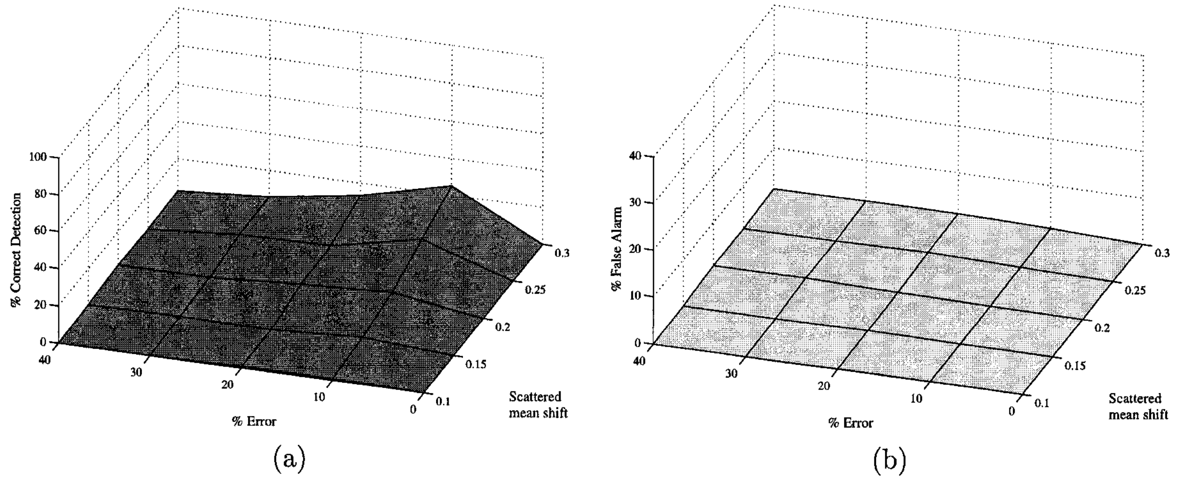


(a) (b)

Figure 4.25: (a) PCA T^2 outlier detection performance in a consecutive mean environment (80% of variance used), (b) PCA T^2 false alarm performance in a consecutive mean environment (80% of variance used).

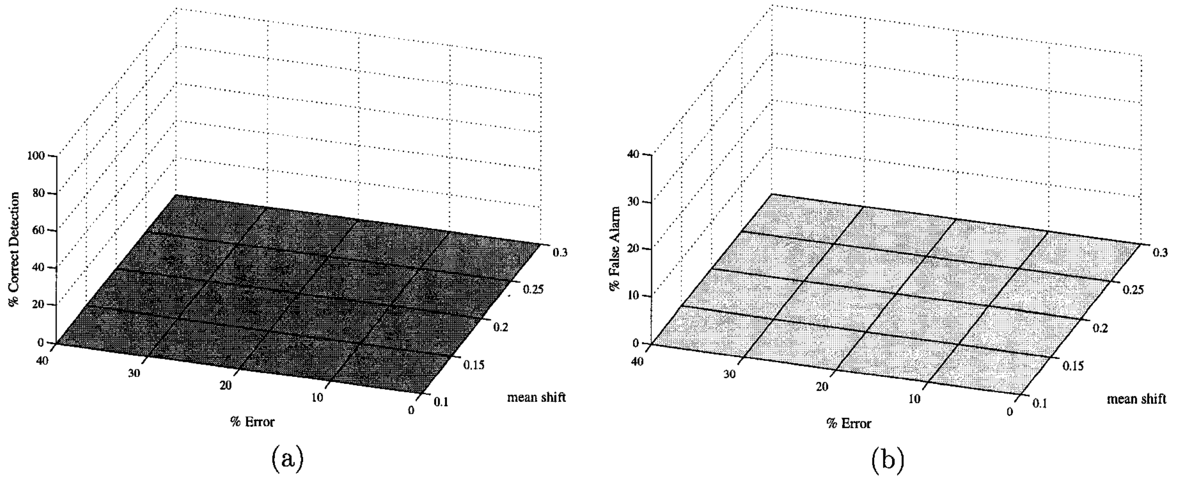
Performance of the cluster PCA algorithm

The performance for cluster PCA comprising different percentage of variance can be observed in Fig. 4.28 and Fig. 4.29. As it can be observed, in high dimensional data the more components k are selected the more accurate the outlier detection will perform. Also, the performance is similar



(a) (b)

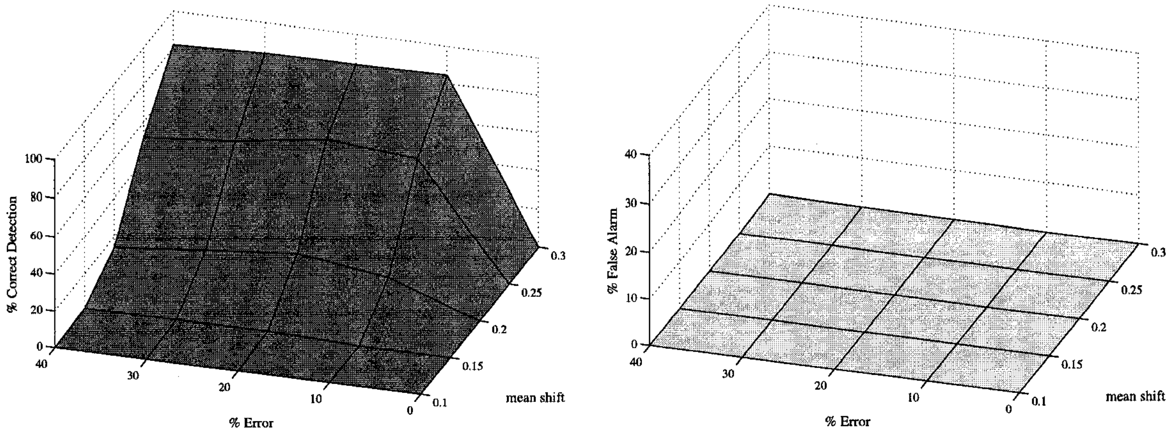
Figure 4.26: (a) PCA T^2 outlier detection performance in a scattered mean environment (80% of variance used), (b) PCA T^2 false alarm performance in a scattered mean environment (80% of variance used).



(a) (b)

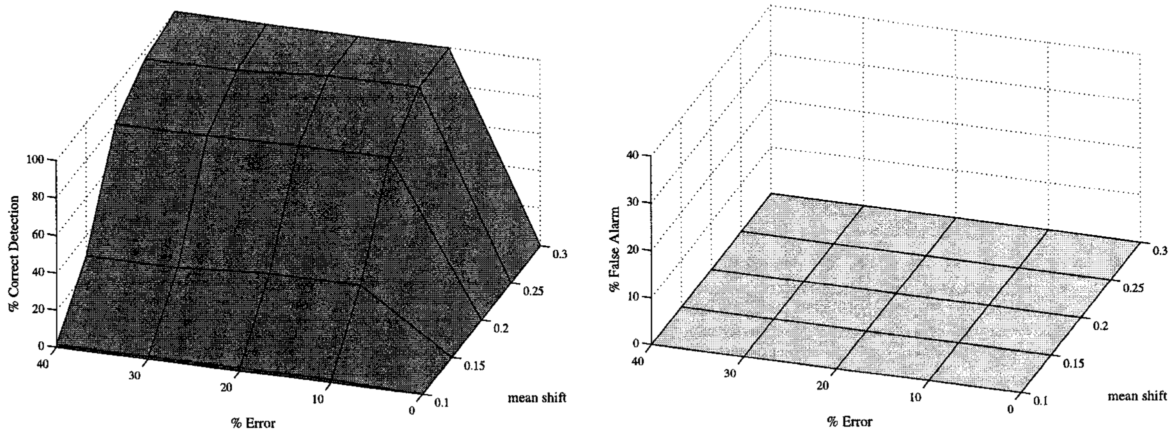
Figure 4.27: (a) PCA T^2 outlier detection performance in a consecutive mean environment (99% of variance used), (b) PCA T^2 false alarm performance in a scattered mean environment (99% of variance used).

in both consecutive and scattered mean shift as seen in Fig. 4.30. Finally, the false alarm ratio is virtually zero across all parameters and environments.



(a) (b)

Figure 4.28: (a) Cluster PCA outlier detection performance in a consecutive mean environment (80% of variance used), (b) Cluster PCA false alarm performance in a consecutive mean environment (80% of variance used).



(a) (b)

Figure 4.29: (a) Cluster PCA outlier detection performance in a consecutive mean environment (99% of variance used), (b) Cluster PCA false alarm performance in a consecutive mean environment (99% of variance used).

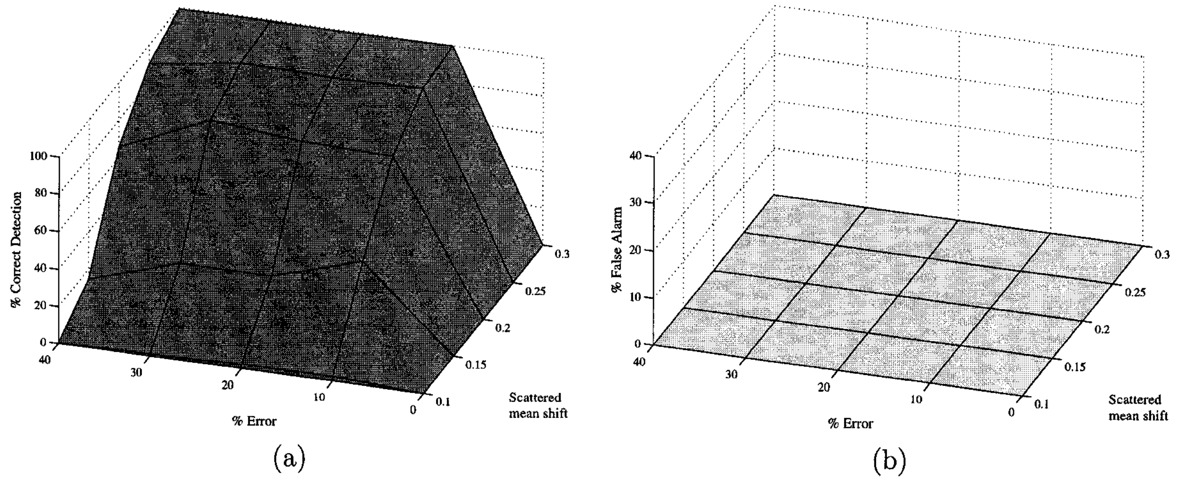


Figure 4.30: (a) Cluster PCA outlier detection performance in a scattered mean environment (99% of variance used), (b) Cluster PCA false alarm performance in a scattered mean environment (99% of variance used).

4.6 Conclusions

In this chapter, we introduced a new multivariate robust algorithm by combining hierarchical clustering and principal component analysis. The core idea behind our proposed technique is to use clustering analysis to determine the optimal subset of observations that will be used to calculate key parameters needed for our analysis. We then used PCA on the original data set in order to determine the outliers present. The experimental and simulation results clearly showed a much improved performance of the proposed approach in comparison with existing methods.

Chapter 5

Dynamic Fault Detection and Diagnosis using Independent Component Analysis

We introduce a new extension of the conventional independent component analysis to deal with multivariate dynamic data [56]. The proposed technique is more robust to outliers in comparison with existing multivariate outlier detection schemes. We also suggest an innovative way to detect and diagnose faults. A comparative study on the Tennessee Eastman process will illustrate the improved performance of our proposed algorithm in comparison with existing statistical monitoring and controlling charts.

5.1 Introduction

With recent advances in computer and instrumentation technology, we have seen an uprise of data collection. This has helped operation of processes greatly in detecting defects, equipment malfunctions or any type of signal that might deviate the process from its normal operating conditions [35]. Conventional Shewhart charts and CUSUM charts have been widely used in the industry to monitor univariate processes, but are inefficient for multivariate processes where variables are highly correlated [3]. This is particularly true for chemical and engineering related environments [4, 6].

Therefore multivariate statistical process control (MSPC) has been developed and used to monitor complex models. Data reduction techniques such as principal component analysis (PCA) and partial least square (PLS) are widely used for signal to noise extraction and process monitoring. [1,4,6].

Recently, a new MSPC method based on independent component analysis (ICA) was suggested. It was shown that ICA reveals more useful information from observed data than PCA, and therefore more robust to outliers [36,37,39,40].

The above mentioned techniques are all based on the assumption that the variables monitored are independent in time. For conventional industrial processes, the statement is only true for long sampling intervals (2 to 12 hours). Therefore, for process monitoring with fast sampling intervals, serial correlation needs to be considered [1,41]. To accommodate dynamic multivariate settings, Dynamic PCA (DPCA) was introduced using the same concepts behind static PCA [41]. In this paper, we present a new process monitoring techniques which we refer to as dynamic independent component analysis (DICA). We extend the advantages behind ICA to detect outliers in a time correlated environment and introduce an innovative way on how to diagnose faults.

The layout of this chapter is as follows. In the next section, we briefly review some related work. Section 5.3 introduces the proposed robust multivariate algorithm. In section 5.4, experimental results are presented to demonstrate the performance of the proposed approach using the Tennessee Eastman Process (TEP). Section 5.5 concludes the chapter.

5.2 Related work

In this section, we briefly review some multivariate control charts that will be used for comparison with our proposed approach.

5.2.1 Dynamic principal component analysis

Dynamic PCA is a method used to take into account the serial correlation that exist between variables. This process is similar to static PCA to the exception that the data matrix X needs to transformed to an augmented data matrix by the previous β observations (also described as a Hankel matrix). When PCA is applied on the augmented data matrix, we obtain a multivaiate autoregressive model ARX(β) [1,41–44].

The same statistics (T^2 and Q^2) described for PCA can also be applied for DPCA. Also, including lags in the data matrix can result to a better model representation and higher correlation of information. It has been proven for serially correlated data, DPCA is expected to outperform PCA [1,41].

5.2.2 Independent component analysis

Independent component analysis is a method used to find new combination of factors from multivariate statistical data by distinguishing components that are both independent and nongaussian [38]. Given any standardized data matrix M of size $s_1 \times s_2$ where the mean is subtracted, variance is unitized and the data is whitened, the relationship between the independent component and the measured variables may be given as

$$M = SA$$

where S is a $s_1 \times s_2$ matrix of independent component vectors and A is a $s_2 \times s_2$ unknown mixing matrix. Out of s_2 dimensions we may keep $\kappa \leq s_2$ independent components to reconstruct the data matrix.

The primary objective of ICA is to determine the demixing matrix W used to retrieve the independent component scores. It can be expressed as

$$S = MA^{-1} = MW$$

The independent component scores can then be used to extract the proper monitoring statistics and their associated control limits [39,40].

5.3 Proposed method

Our method is presented in four main phases. The first phase describes how to set-up the process using DICA. The second phase will demonstrate how to derive the control limits. The third phase will explain how to monitor new observations. Finally, the last phase will explain how to isolate and diagnose faults.

5.3.1 Process set-up

1) Acquire a time series data where a process is operated under normal conditions, that is the data is outlier free.

2) Compute the mean $\bar{\boldsymbol{x}}$ and covariance matrix R in order to subtract and unitize the variance of our data matrix

$$Z = (X - \mathbf{1}\bar{\boldsymbol{x}})D^{-1/2}$$

where $\mathbf{1} = (1, \dots, 1)^T$ is a $m \times 1$ vector of all 1's and $D = (diag(R))^{1/2}$ is the diagonal standard deviation.

3) Stack the data by augmenting each observation with the previous β observations to obtain a new data set

$$H = [\boldsymbol{h}_1, \dots, \boldsymbol{h}_m]^T = [Z(t), Z(t-1), \dots Z(t-\beta)]$$

This will result to an $m \times \beta p$ matrix. Experience indicates that $\beta \epsilon \{1, 2, 3\}$ lags is sufficient for process monitoring [1]. For more details on how to automatically select β refer to [41].

4) Apply the eigendecomposition to the covariance matrix $R_H = V \Delta V^T$ of the augmented data matrix where V represent the column of eigenvectors and Δ represents a diagonal matrix of the eigenvalues ordered by magnitude.

5) Eliminate the cross-correlation between random variables by whitening the data matrix H

$$\tilde{H} = HB \quad \text{where} \quad B = V\Delta^{-1/2}$$

6) Apply ICA to the whitened data $\tilde{H}$ and determine the independent component $\tilde{S}$ and demixing matrix $\tilde{W}$

$$\tilde{H} = \tilde{S}\tilde{A} = SAB$$

where

$$\tilde{S} = HV\Delta^{-1/2}\tilde{A}^{-1} = H\tilde{W}$$

and

$$\tilde{W} = V\Delta^{-1/2}\tilde{A}^{-1}$$

7) Sort the columns of the demixing matrix $\tilde{W} = (\tilde{\boldsymbol{w}}_1, \ldots \tilde{\boldsymbol{w}}_{\beta p})$ in decreasing order based on the Euclidean norm of each $\tilde{\boldsymbol{w}}_j$ (and in consequence the rows of the mixing matrix $\tilde{A}$) [39,47].

8) Select d dominant columns from the demixing matrix $\tilde{W}$. The SCREE test obtained from the Eulcidean norm can be used to facilitate the selection criteria [47]. We will obtain two reduced matrix denoted as $\tilde{W}_d$ for dominant demixing matrix and $\tilde{W}_e$ for the excluded columns of the demixing matrix. Split the rows the mixing matrix $\tilde{A}$ in two district matrices ($\tilde{A}_d$ and $\tilde{A}_e$) in the same fashion as $\tilde{W}$.

9) Calculate the dominant I_d^2 statistic, the excluded I_e^2 statistic and the squared prediction error Q^2 statistic for $i = 1, \ldots, m$ observations

$$I_d^2(i) = ||\boldsymbol{h}_i \tilde{W}_d||^2$$

$$I_e^2(i) = ||\boldsymbol{h}_i \tilde{W}_e||^2$$

$$Q^2(i) = ||\boldsymbol{h}_i - \boldsymbol{h}_i \tilde{W}_d \tilde{A}_d^{-1}||^2$$

5.3.2 Process control limits

The control limits may be established for each ICA statistic (I_d^2, I_e^2 and Q^2) using the univariate kernel estimator

$$\hat{f}(x) = \frac{1}{m\eta} \sum_{i=1}^{m} K\left(\frac{x - x_i}{\eta}\right)$$

where $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$ is the Gaussian kernel, $\eta = \frac{1.06}{m^{1/5}}\sigma$ is the bandwidth (also called the smoothing parameter), m is the number of observations and σ is the standard deviation for the statistic under study [48]. We can then select the control limit as the point δ occupying 99% of the area of the density function

$$\int_{-\infty}^{\delta} \hat{f}(x) dx = 0.99$$

Given the fact that the ICA statistics do not follow a Gaussian distribution or any other particular distribution, the kernel density estimator is good alternative. It also has the advantage of following the data distribution more closely than the Hotelling's T^2 statistic [39].

5.3.3 Process monitoring for new observations

For any new data matrix X^{new}, we need to standardize the data using the mean and variance found from the normal operating conditions

$$Z^{new} = (X^{new} - \mathbf{1}\bar{\boldsymbol{x}})D^{-1/2}$$

where $\bar{x}$ and D are the same parameters retrived from step 2) of the process set-up.

Next, the data Z^{new} needs to get stacked as described in step 3) of the process set-up (H^{new}). Then, we can determine the three ICA statistic and plot them against the threshold determined for each statistic.

$$I_d^2(i) = ||\boldsymbol{h}_i^{new}\tilde{W}_d||^2$$

$$I_e^2(i) = ||\boldsymbol{h}_i^{new}\tilde{W}_e||^2$$

$$Q^2(i) = ||\boldsymbol{h}_i^{new} - \boldsymbol{h}_i^{new}\tilde{W}_d\tilde{A}_d^{-1}||^2$$

5.3.4 Process fault diagnosis

We propose a new contribution plot for DICA to diagnose faults by taking into account the spacial correlation between variables and the serial correlation between observations (time). We also assume that all observations that are found outlying are due to the same fault. In the case that each outlying observation is a fault of its own, the diagnosis need to be done with respect to these observations. The process is explained as follows:

1) Assuming we determined $r \leq m$ observations exceeding the threshold determined for the I_d^2 statistic, calculate the contribution for each variable to the out of control observation based on the number of d selected IC's

$$cont_{k,j} = \sum_{i=1}^{r} \tilde{w}_{j,k}\phi_{i,k}^T h_{i,j}$$

where

$$\Phi = (\phi_1, \ldots, \phi_{\beta p}) = H\tilde{W}_d\tilde{A}_d$$

and $\tilde{w}_{j,k}$ is the $(j,k)^{th}$ element of the demixing matrix $\tilde{W}_d$.

2) If $cont_{k,j}$ is negative, we can set it to zero.

3) Calculate the contribution for each j process variable

$$Cont_j = \sum_{k=1}^{d} cont_{k,j} \quad \text{for} \quad j = 1, \ldots, \beta p$$

4) Unfold $\boldsymbol{CONT} = (Cont_1, \ldots, Cont_{\beta p})$ from $1 \times \beta p$ vector to a $1 \times p$ vector by adding the lagged variables with the original p variables.

5) Plot the contribution for all p variables on a single graph. The variables that have an excess contribution are the root causes for the fault.

The same steps described can also be applied for the I_e^2 statistic except that Φ is now defined as

$$\Phi = H\tilde{W}_e\tilde{A}_e$$

We can also define a contribution plot for squared prediction error (Q^2). Assuming there are r observations that exceed the control limit, the contribution can then be defined as

$$\boldsymbol{CONT} = \sum_{i=1}^{r} |\boldsymbol{h}_i - \boldsymbol{h}_i\tilde{W}_d\tilde{A}_d|$$

We need to unfold the $\boldsymbol{CONT}$ vector as described in step 4) and then plot all p variables on a single graph.

In some instances, the contribution for each variable using the I_d^2 statistic and the I_e^2 will not perform as expected. This is due to the fact that the multiplication $\tilde{W}\tilde{A}$ allows certain variables to dominate without any specific contribution and therefore falsifying the results [1]. This can be demonstrated by plotting the contribution plot of the normal operating data.

In order to overcome this problem, we suggest the following approach:

1) Find the contribution of each variable when the process is operating under normal conditions.

2) Find the percentage of the total contribution of each variable

$$\%Cont_j = \frac{Cont_j}{\sum_{j=1}^{p} Cont_j} \quad \text{for} \quad j = 1, \ldots, p$$

3) Find the median value from the p percent contributions. This value will be denoted as M_c.

4) Determine the scalable coefficients that will distribute the variables to have equal probability

$$SC_j = \frac{M_c}{\%Cont_j} \quad \text{for} \quad j = 1, \ldots, p$$

5) Multiply the contribution of each variable with its scaled conjugate

$$Cont_j^{(new)} = Cont_j \cdot SC_j \quad \text{for} \quad j = 1, \ldots, p$$

This process fault diagnosis can be applied to any process monitoring technique including PCA, DPCA and ICA.

5.4 Simulation and results

In this section, we will compare the performance our proposed technique with conventional methods using the Tennessee Eastman process (TEP).

5.4.1 Tennessee Eastman process

The Tennessee Eastman process is considered as a benchmark simulation for various process monitoring techniques [36]. The process consists of five major transformation units: a reactor, a condensor, a compressor a separator, and a striper [49]. The structure is shown in Fig 5.1. There is a total of 41 variables measured and 12 manipulated variables [1]. For this experiment, we have selected 22 continuous process measurements and 11 manipulated variables. The agitation speed has been excluded due to the lack of control and 19 composite measurement were left out due to the difficulty of measurement [36]. The 33 variables chosen are described in Table 5.1. We have created 22 data sets consisting of 960 observations. The first data set was created under normal operating conditions and the rest were simulated based on a fault introduced at sample 160. A detailed description of each fault can be found in Table 5.2. The sampling interval for each observation is 3 minutes and therefore dynamic monitoring is suited for this experiment [1, 36].

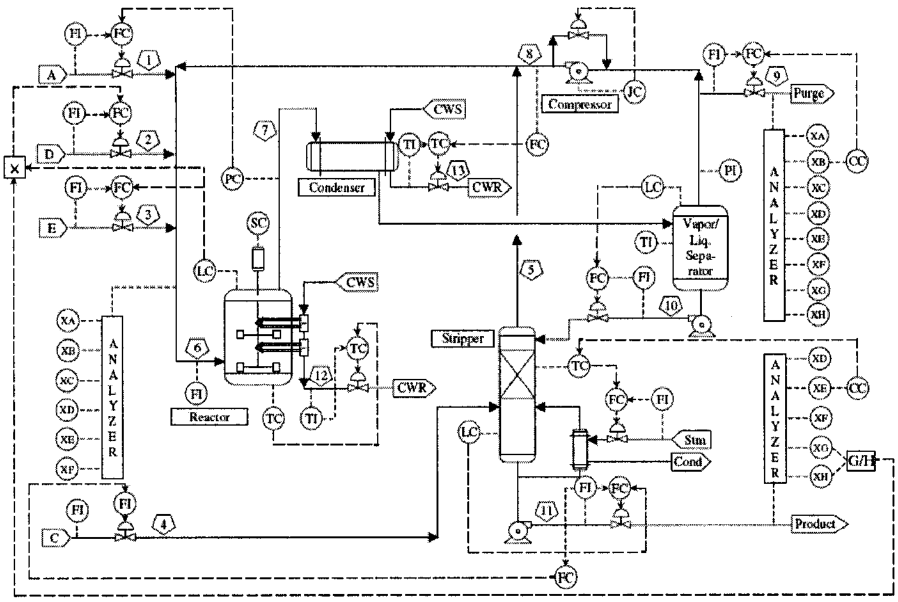


Figure 5.1: Control System of the Tennessee Eastamn Process [35].

Table 5.1: Monitoring variables in the TEP.

No.	Measured Variables	No.	Measured Variables	No.	Manipulated Variables
1	A feed	12	Product separator level	23	D feed flow valve
2	D feed	13	Product separator pressure	24	E feed flow valve
3	E feed	14	Product separator underflow	25	A feed flow valve
4	Total feed	15	Stripper level	26	Total feed flow valve
5	Recycle Flow	16	Stripper pressure	27	Compressor recycle valve
6	Reactor feed rate	17	Stripper underflow	28	Purge valve
7	Reactor pressure	18	Stripper temperature	29	Separator pot liquid flow valve
8	Reactor level	19	Stripper steam flow	30	Stripper pot liquid flow valve
9	Reactor temperature	20	Compressor work	31	Stripper steam valve
10	Purge rate	21	Reactor cooling water outlet temperature	32	Reactor cooling water flow
11	Product separator temperature	22	Separator cooling water outlet temperature	33	Condensor cooling water flow

Table 5.2: Process faults.

Fault No.	Description	Type
1	A/C feed ratio, B composition constant (stream 4)	Step
2	B composition, A/C ratio constant (stream 4)	Step
3	D feed temperature (stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (stream 1)	Step
7	C header pressure loss - reduced availability (stream 4)	Step
8	A,B,C feed composition (stream 4)	Random variation
9	D feed temperature (stream 2)	Random variation
10	C feed temperature (stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reactor kinetics	Slow Drift
14	Reactor cooling water	Sticking
15	Condenser cooling water	Sticking
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown
21	Valve position constant (stream 4)	Constant position

5.4.2 Fault detection and diagnosis

We have ran simulations on all 21 faults based on the four techniques (PCA, DPCA, ICA, DICA) discussed in the previous sections. We selected 10 dominant IC's for both DICA and ICA. For PCA and DPCA, the number of component chosen were based on the total variation contained in the principal components. In this case we selected k components containing 90% of the total variance.

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \geq 90\%$$

Finally, the number of lags used for this dynamic process was set to $h = 3$. For each statistic, the detection rate and false alarm rate were tabulated for all 21 process fault. The results can be observed in Table 3 and Table 4. To facilitate the comparison between the various techniques, we have selected the statistic with the highest detection rate for each method along with its associated false alarm rate. The results are shown in Fig 5.2.

Table 5.3: Detection rate on the TEP.

Faults	DICA			ICA			DPCA		PCA	
	I_d^2	I_e^2	Q^2	I_d^2	I_e^2	Q^2	T_d^2	Q^2	T^2	Q^2
1	99.75	99.88	99.25	99.50	99.88	99.50	99.75	99.88	99.25	100.00
2	96.38	99.13	98.00	97.25	98.38	98.00	98.63	98.00	98.63	96.75
3	0.00	5.63	0.38	0.38	2.63	0.75	5.38	11.63	7.50	8.88
4	90.88	100.00	95.38	94.88	99.88	77.25	26.88	100.00	75.75	100.00
5	100.00	100.00	100.00	99.88	100.00	100.00	29.50	69.13	30.25	40.13
6	100.00	100.00	100.00	100.00	100.00	100.00	99.00	100.00	99.38	100.00
7	98.25	100.00	100.00	99.88	100.00	100.00	100.00	100.00	100.00	100.00
8	97.50	93.88	97.75	96.38	92.50	97.88	97.38	82.13	98.38	97.63
9	0.38	5.25	0.50	0.38	1.88	0.63	4.88	9.88	6.88	8.75
10	80.50	93.88	73.13	72.75	88.13	67.13	40.38	59.50	40.88	52.38
11	68.13	93.88	65.88	56.25	75.25	59.38	47.88	96.38	64.13	71.38
12	99.25	99.88	99.88	99.25	99.88	99.75	99.13	96.75	98.75	94.00
13	95.25	96.38	94.50	94.38	95.38	94.38	95.00	95.63	94.13	95.63
14	99.88	100.00	99.88	100.00	99.88	99.88	99.88	100.00	100.00	99.63
15	1.88	34.38	2.88	0.38	4.75	1.50	5.00	9.50	6.88	10.88
16	67.88	97.00	50.00	77.50	90.50	67.25	23.88	58.63	25.12	53.00
17	96.88	97.88	94.75	85.13	96.75	88.38	85.38	98.00	84.13	97.25
18	90.25	90.88	89.88	89.88	90.00	89.63	89.50	91.75	90.00	91.13
19	72.88	99.75	7.25	74.13	88.88	44.75	49.25	75.50	29.00	38.38
20	62.50	91.63	54.13	80.75	90.13	67.88	57.38	65.63	43.50	64.13
21	15.13	56.38	37.75	33.75	53.37	36.88	54.25	45.63	47.25	59.25

As it can be observed, DICA outperforms ICA, DPCA and PCA for almost all faults. It is particularly successful with smaller outlying conditions that regular PCA is unable to detect (Fault 5, 10, 16, 19, and 20). We can also conclude that dynamic processing is a more suitable model for time correlated data. Finally, the kernel density estimator has provided us with a smaller false alarm ratio than PCA or DPCA. This is due to the fact that the control limits follow more closely the data and are less likely to incorporate unknown operating regions [39].

Finally, some faults are easier to identify than others. Faults such as 1, 2, 4, 6, and 14 are easier to detect since they largely deviate from the normal operating conditions. Other faults such as 3, 9, and 15 have a smaller influence on the process and are not easy to identify.

In the following sections, we will focus on giving a complete analysis on Faults 5 and 11:

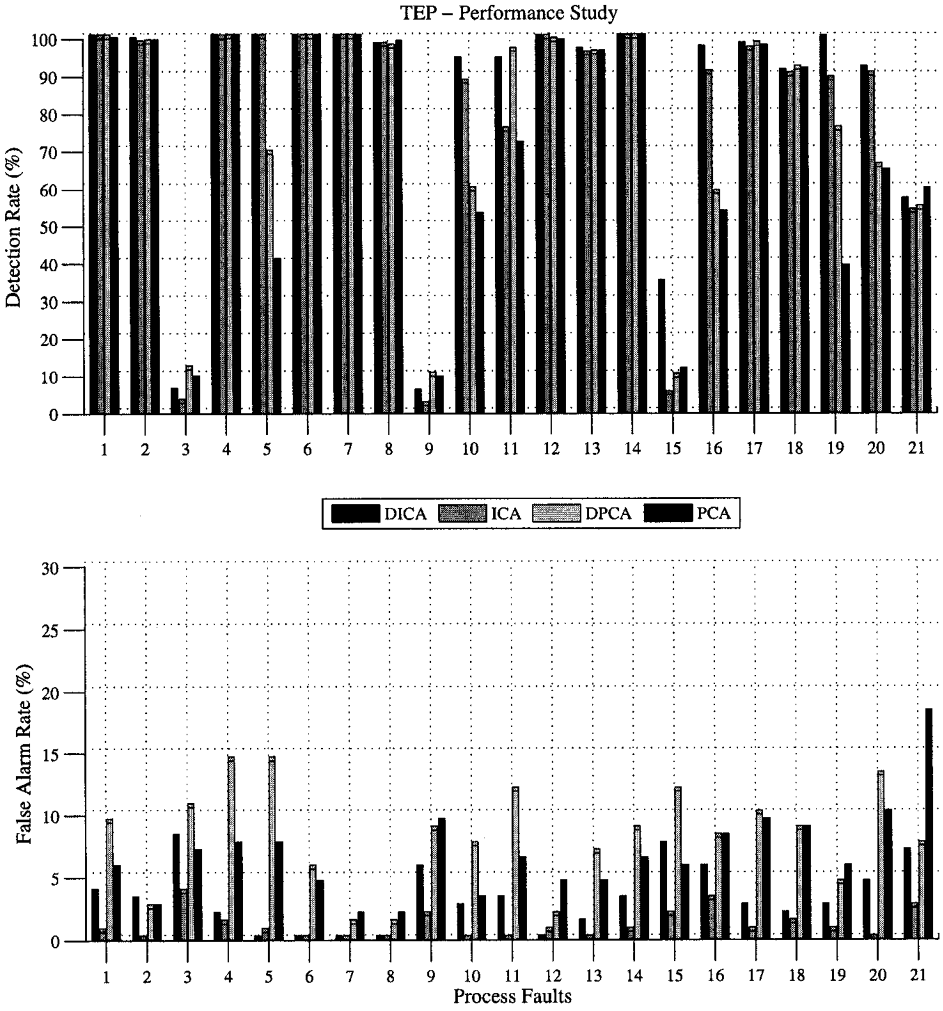


Figure 5.2: Performance study: detection rate and false alarm rate.

Table 5.4: False Alarm Rate on the TEP.

	I_d^2	I_e^2	Q^2	I_d^2	I_e^2	Q^2	T_d^2	Q^2	T^2	Q^2
1	0.63	3.75	0.00	0.63	0.63	0.00	3.13	9.38	0.63	5.63
2	0.00	3.13	0.00	0.00	0.00	0.00	2.50	8.13	2.50	4.38
3	0.00	8.13	0.00	0.00	3.75	0.00	1.25	10.63	1.88	6.88
4	0.00	1.88	0.00	0.00	1.25	0.63	1.25	14.37	2.50	7.50
5	0.00	1.88	0.00	0.00	1.25	0.63	1.25	14.37	2.50	7.50
6	0.00	2.50	0.00	0.00	0.00	0.00	1.25	5.63	2.50	4.38
7	0.00	2.50	0.00	0.00	0.63	0.00	1.25	8.75	1.88	5.63
8	0.00	5.63	0.00	0.00	0.00	0.00	1.25	5.63	1.88	6.88
9	0.00	5.63	0.00	1.25	1.88	4.38	6.25	8.75	9.38	9.38
10	0.00	2.50	0.00	0.00	0.00	0.00	2.50	7.50	2.50	3.13
11	0.00	3.13	0.00	0.00	0.00	0.63	3.75	11.88	3.75	6.25
12	0.00	4.38	0.00	0.00	0.63	0.00	1.88	8.75	4.38	10.00
13	0.00	1.25	0.00	0.00	0.00	0.00	1.88	6.88	1.25	4.38
14	0.00	3.13	0.00	0.63	0.00	0.00	0.63	8.75	1.88	6.25
15	0.00	7.50	0.00	0.63	1.88	0.63	3.75	11.88	1.88	5.63
16	0.00	5.63	3.13	0.63	3.13	3.75	11.25	8.13	11.88	8.13
17	0.00	2.50	0.00	0.63	0.63	0.00	1.88	10.00	2.50	9.38
18	0.00	1.88	0.00	0.63	1.25	0.00	3.13	8.75	5.00	8.75
19	0.00	2.50	0.00	0.63	0.63	0.00	1.25	4.38	1.88	5.63
20	0.00	4.38	0.00	0.00	0.00	0.00	0.63	13.13	1.25	10.00
21	0.00	6.88	0.00	0.00	2.50	0.00	7.50	10.00	2.50	18.13

Fault 5

Fault 5 is due to a step change in the condenser cooling water inlet temperature which causes a mean shift on the condenser cooling flow (variable 33). This step change creates a chain reaction on most of the variables which causes them to initially go out of control. The control loop detects this excess variation and is able to compensate for the change. 200 samples later, the variables return within their normal operating conditions except for the original variable who caused this fault. As observed from our results, DICA and ICA (Fig 5.3 and Fig 5.5) outperform DPCA and PCA (Fig 5.4 and Fig 5.6). If an operator would be using (D)PCA model to monitor the process, she/he would assume that the process has returned back in control where in reality the process is still behaving abnormally.

Using our method to diagnose the faults, we will first demonstrate the negative influence that some variable naturally exert. In Fig 5.7, we can observe the contribution plot for the variables

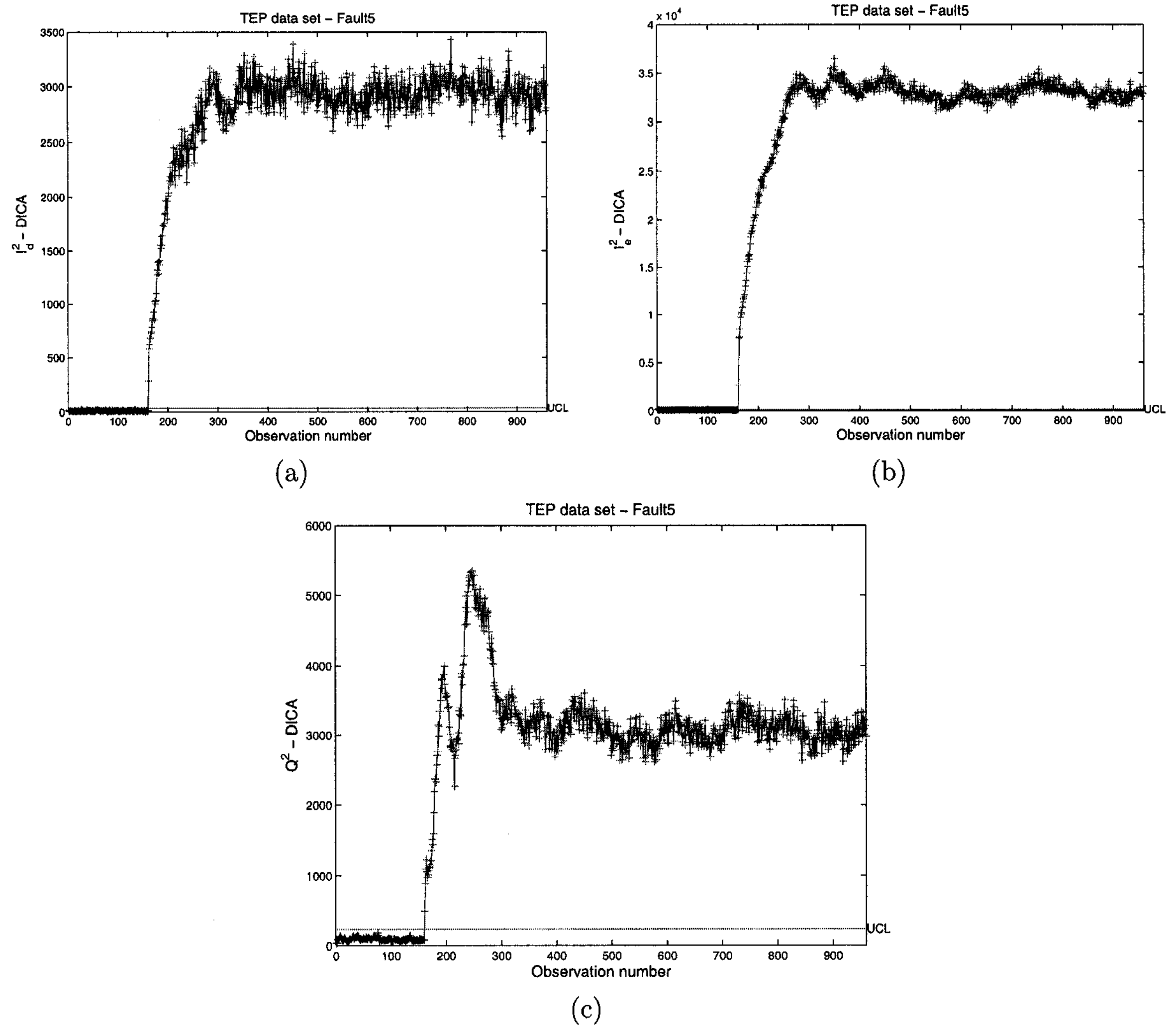
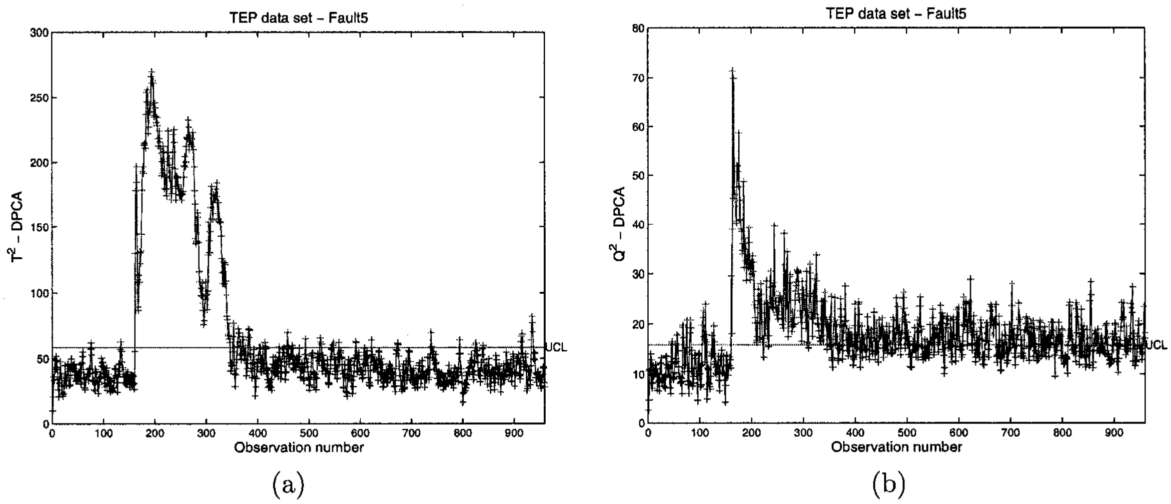


Figure 5.3: (a) I^2_d, (b) I^2_e and (c) Q^2 DICA multivariate statistic for Fault 5.

under normal operating conditions. As observed, variables 12, 15, 17, 29, 30, and 33 have a stronger influence and therefore scaling is necessary. Without it, the diagnostic would be useless. In Fig 5.8, we can observe the contribution of the three DICA statistics. As we can see, variables 19, 31, and 33 have a stronger influence for this particular fault. Also, in this case, the Q^2 statistic was unable to detect the root cause for this fault. In Fig 5.9 we can observe the contribution of each targeted variable. It confirms our theory that variable 33 is the root cause for this fault. It also confirms



(a) (b)

Figure 5.4: (a) T^2 and (b) Q^2 DPCA multivariate statistic for Fault 5.

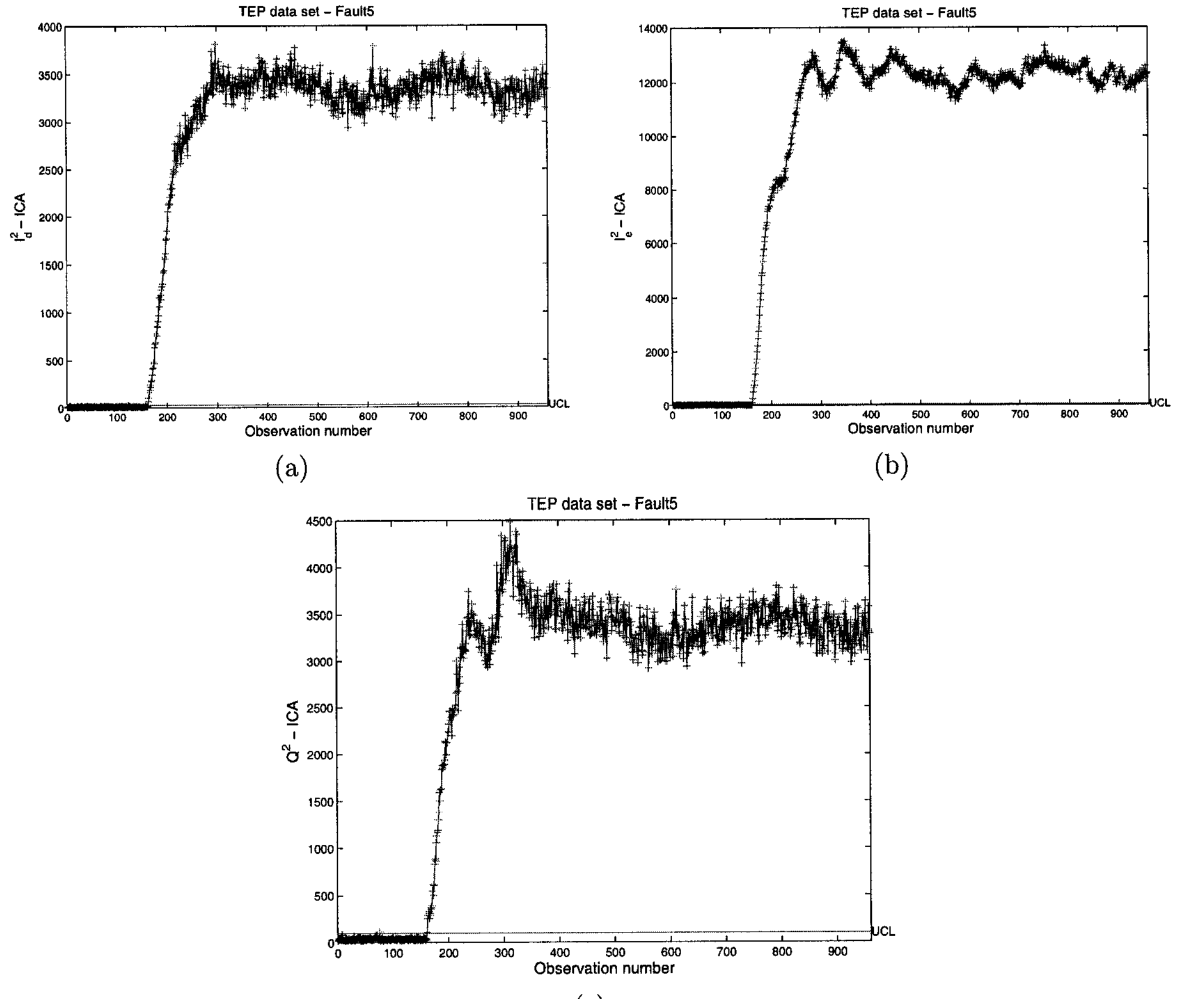
(by observing variable 19 and 31) that there was an initial excess variation which later returned under normal operating conditions.

Fault 11

Fault 11 is a random variation of the reactor cooling inlet temperature. This fault induces an oscillation in the reactor cooling water flow (variable 32) which results in a rise of temperature in the reactor (variable 9). The results obtained using the methods described can be observed in Fig 5.10 through Fig 5.13. For this fault, dynamic processing is more efficient in continuously detecting the variation. The I_e^2-DICA multivariate statistic (Fig 5.10-(b)) and Q^2-DPCA (Fig 5.11-(c)) multivariate statistic are the two indicators that observe the best performance. Finally the contribution plots (Fig 5.14) and the individual variable contribution (Fig 5.15) clearly demonstrate that variable 32 and variable 9 are the root causes for this excess variation.

5.5 Conclusions

In this chapter, we introduced a dynamic fault detection and diagnosis using ICA to monitor serial correlated data. We also introduced a novel strategy to detect and diagnose faults. The proposed



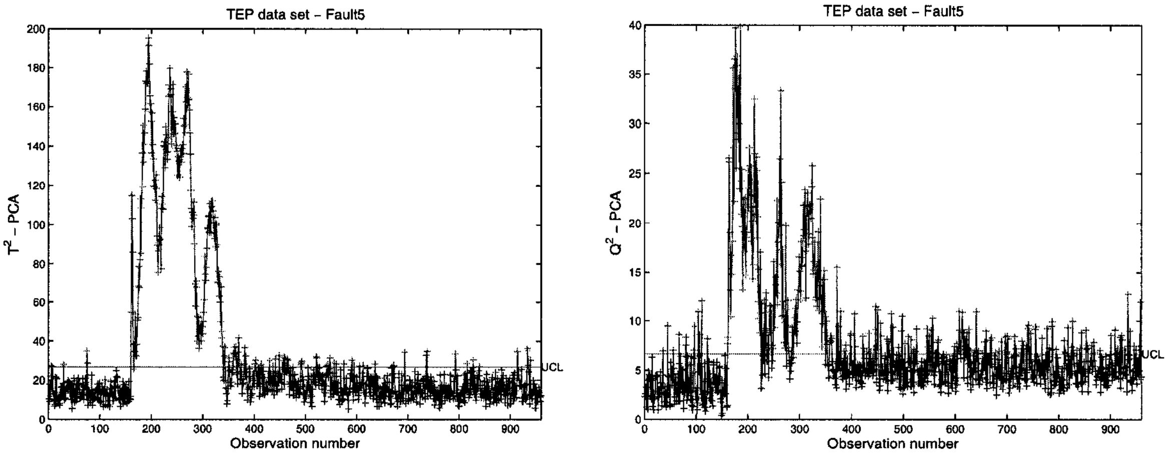
(a)

(b)

(c)

Figure 5.5: (a) I^2_d, (b) I^2_e and (c) Q^2 ICA multivariate statistic for Fault 5.

method was evaluated on the Tennessee Eastman process on 21 different faults. As demonstrated, modelling the process dynamically and applying DICA on the data outperforms existing detection methods that are currently in use. Also, our diagnostic algorithm was able to accurately detect and isolate the root causes for each individual fault.



(a) (b)

Figure 5.6: (a) T^2 and (b) Q^2 PCA multivariate statistic for Fault 5.

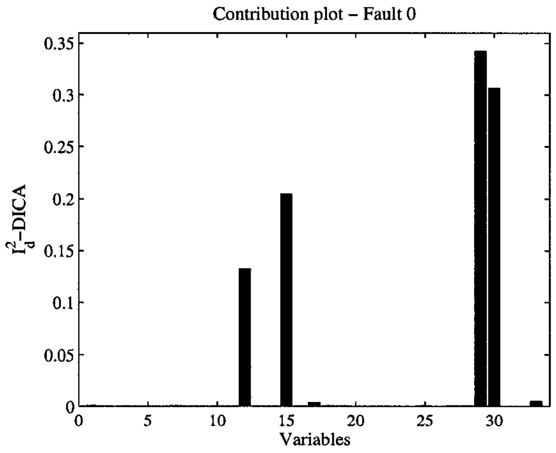
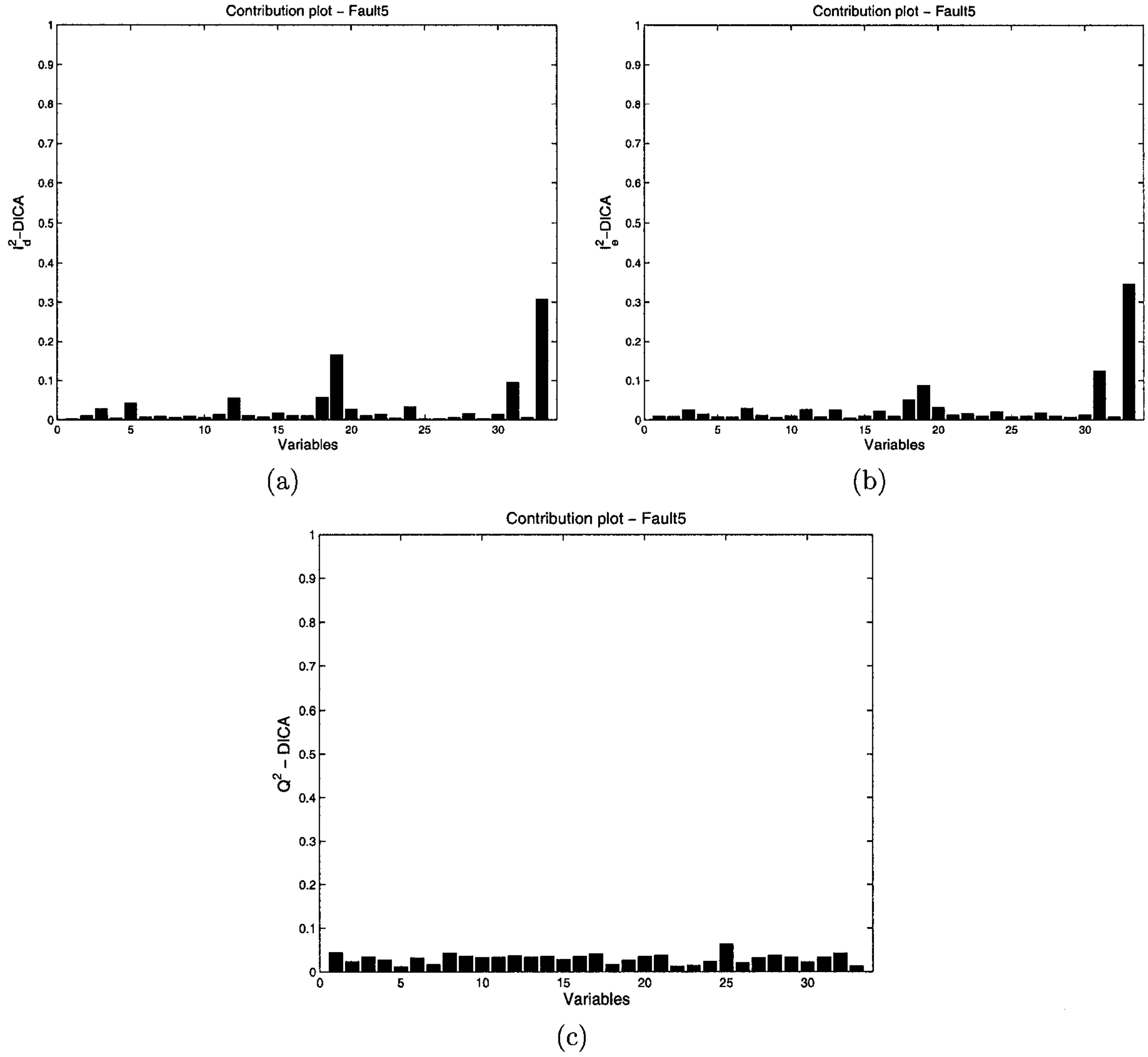


Figure 5.7: Contribution plot for normal operating data.



(a)

(b)

(c)

Figure 5.8: (a) I_d^2, (b) I_e^2 and (c) Q^2 DICA contribution plot for Fault 5.

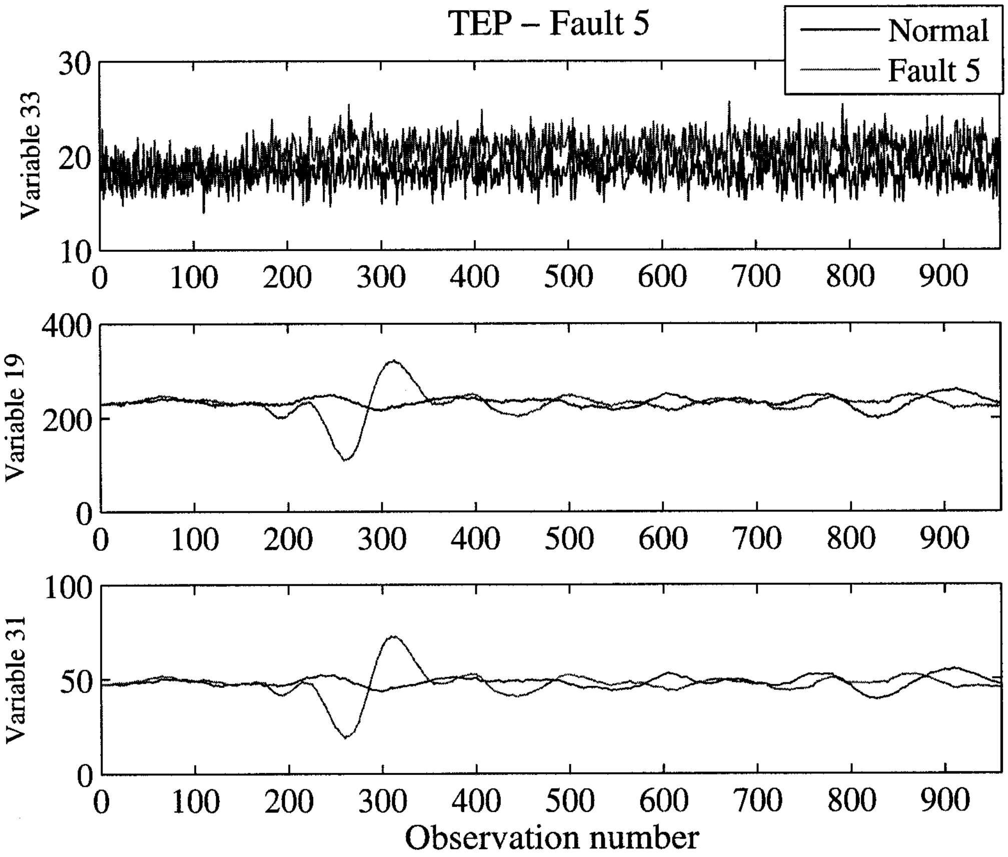
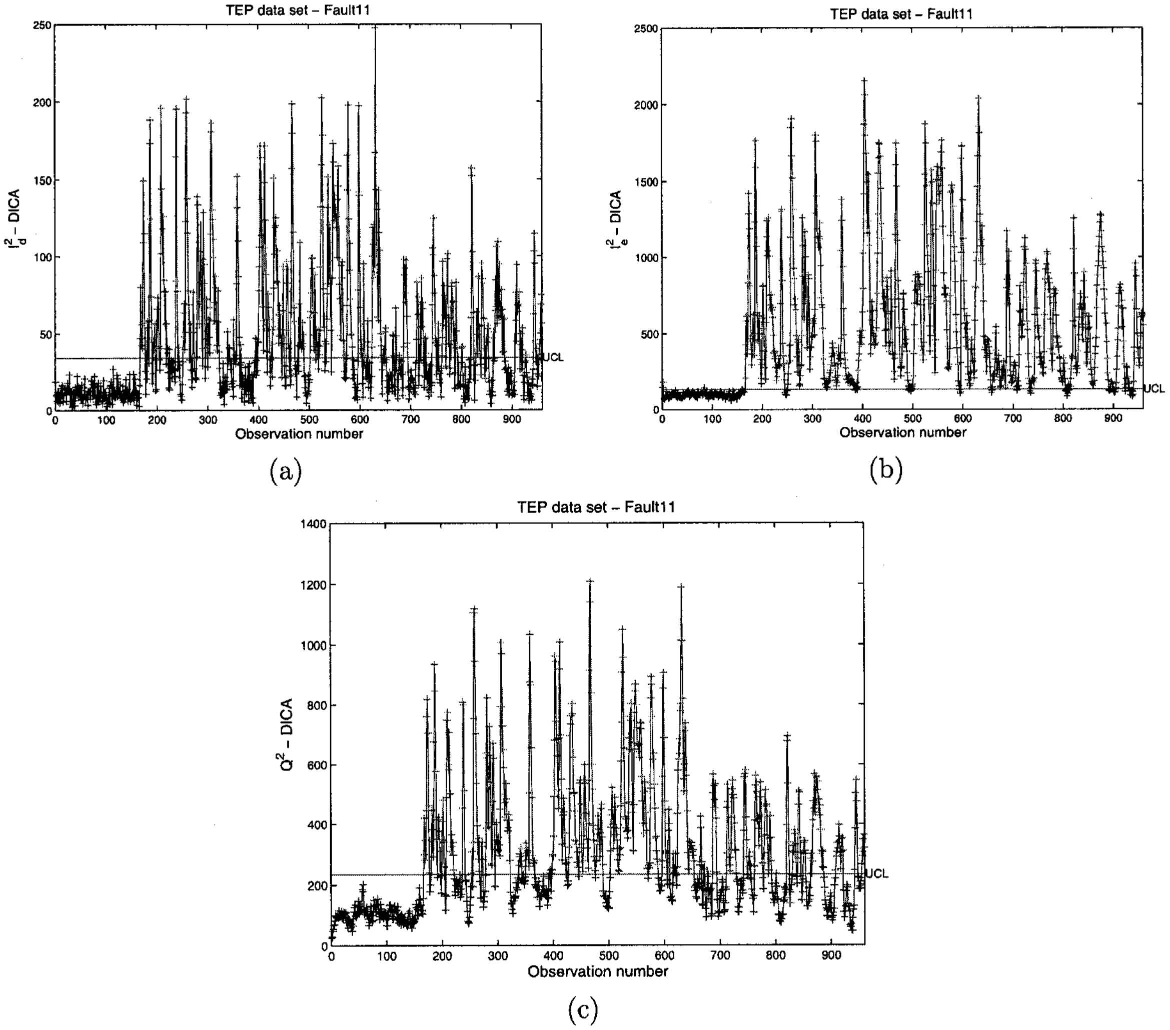


Figure 5.9: Individual variable contribution for Fault 5.

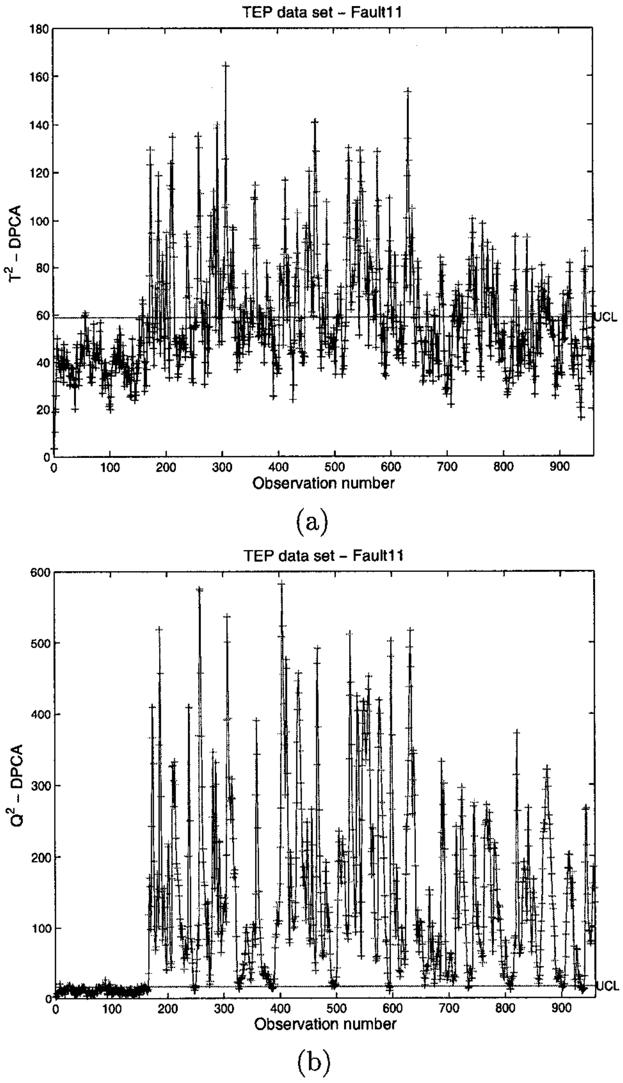


(a)

(b)

(c)

Figure 5.10: (a) I^2_d, (b) I^2_e and (c) Q^2 DICA multivariate statistic for Fault 11.



(a)

(b)

Figure 5.11: (a) T^2 and (b) Q^2 DPCA multivariate statistic for Fault 11.

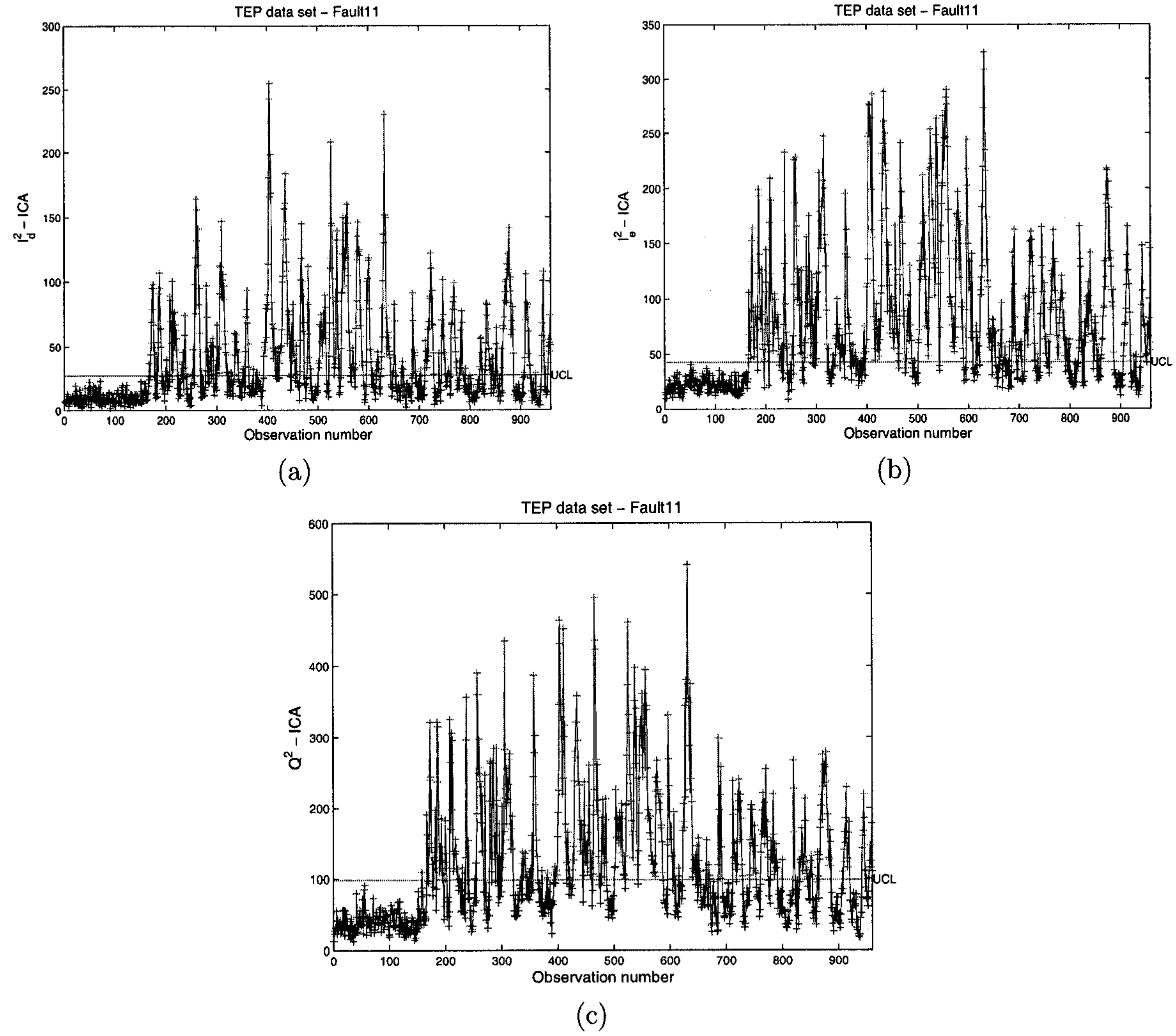
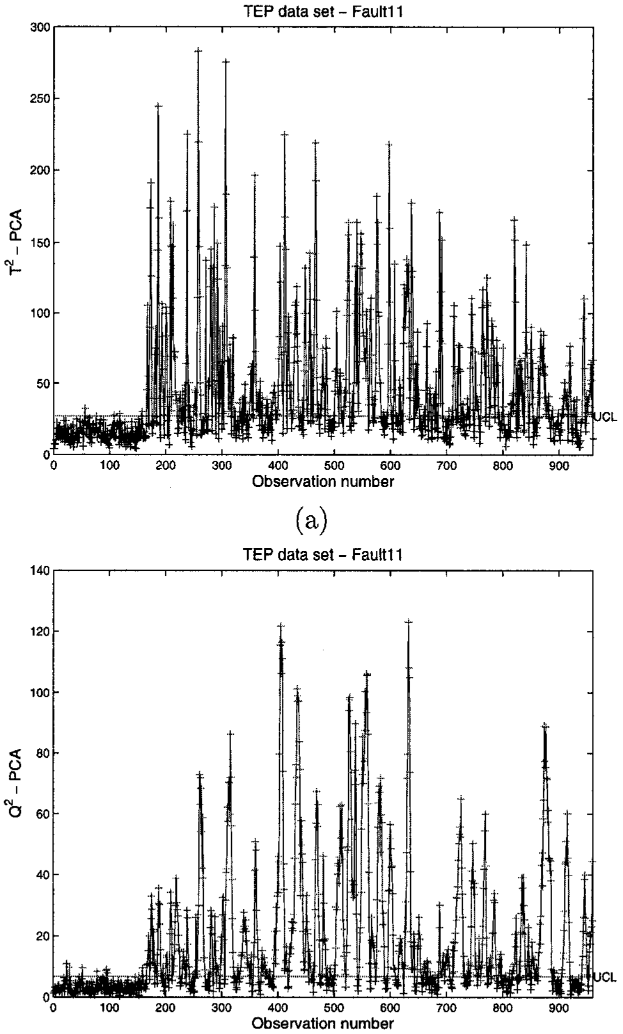


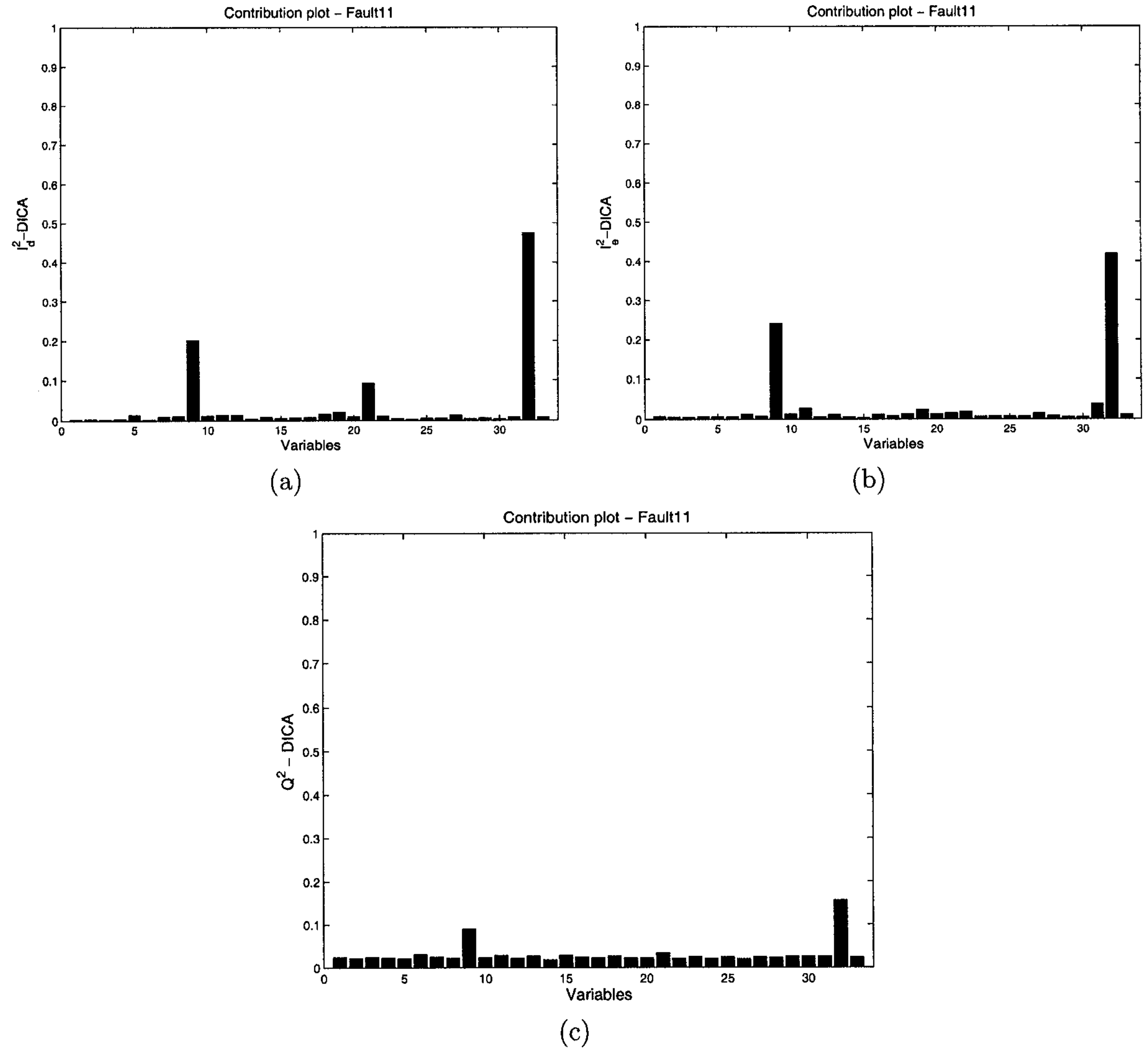
Figure 5.12: (a) I_d^2, (b) I_e^2 and (c) Q^2 ICA multivariate statistic for Fault 11.



(a)

(b)

Figure 5.13: (a) T^2 and (b) Q^2 PCA multivariate statistic for Fault 11.



(a)

(b)

(c)

Figure 5.14: (a) I_d^2, (b) I_e^2 and (c) Q^2 DICA contribution plot for Fault 11.

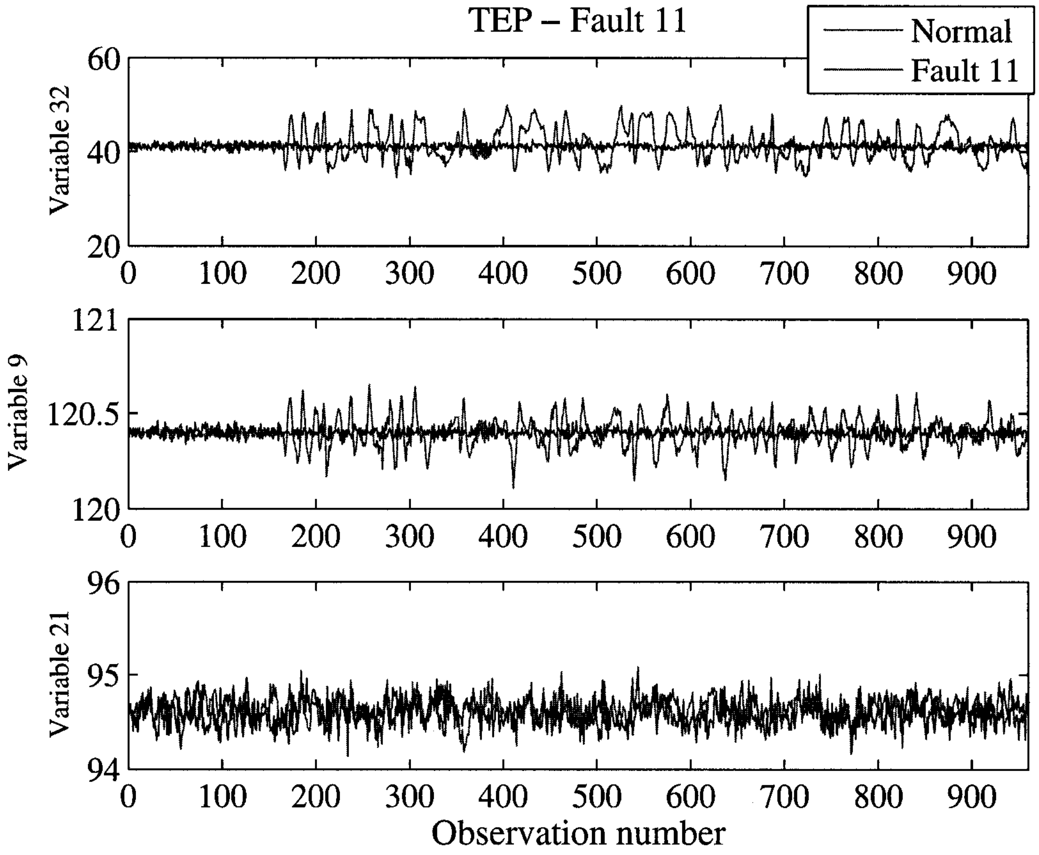


Figure 5.15: Individual variable contribution for Fault 11.

Chapter 6

Conclusions and Future Work

This thesis has presented computationally efficient SPC algorithms to detect and diagnose faults in multivariate data. Our algorithms are built on the foundations of robust statistics and data reduction techniques. We have demonstrated the effectiveness of the proposed methods through extensive experiments with synthetic and real data.

In the next Section, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are provided in Section 6.2.

6.1 Contributions of the thesis

6.1.1 Multivariate robust quality control chart for outlier detection

We presented a new robust multivariate control chart using principal component analysis and robust statistics. The proposed approach consists of two main steps. In the first step, we calculate a robust covariance matrix using the minimum covariance determinant algorithm. In the second step, we apply eigen-decomposition to the robust correlation matrix in order to extract the eigenvalues that will be used to define the proposed control chart. Our experimental results illustrate the much better performance of the proposed algorithm in comparison with existing statistical monitoring and controlling charts.

6.1.2 Statistical process control using kernel PCA

We proposed a new multivariate statistical process control chart using kernel principal component analysis. The core idea behind our proposed technique is to project our data into higher dimension space in order to extract the eigenvalues and eigenvectors of the kernel matrix. The proposed control chart is robust to outliers, and its control limits are derived from the eigen-analysis of the Gaussian kernel matrix in the Hilbert feature space. Our experimental results demonstrate a much improved performance in comparison with the current multivariate control charts.

6.1.3 Cluster principal component analysis for outlier detection

We introduced a new method to detect multiple outliers in both low and high dimensional data. We combined the advantages of hierarchical clustering and principal component analysis to improve the performance while keeping the complexity and computation time relatively low. Our main idea consists of identifying an optimal subset of observations that is used as a comparison model to identify all outliers present in the data. The results clearly show a much improved performance of the proposed approach in comparison with existing robust algorithms.

6.1.4 Dynamic fault detection and diagnosis using independent component analysis

We presented a new process monitoring technique called dynamic independent component analysis (DICA). We extended the advantages behind ICA to detect outliers in a time correlated environment and introduced an innovative way on how to diagnose faults. Our experimental results demonstrated that modelling a time dependent process dynamically and applying DICA on the data outperforms existing monitoring methods that are currently in use.

6.2 Future research directions

Several interesting research directions motivated by this thesis are discussed next. In addition to designing new methodologies for detecting and diagnosing faults, we intend to accomplish the following projects in the near future:

6.2.1 Locally linearly embedding

Recently we have been working on the construction of a new multivariate control chart using the LLE algorithm. It builds a data-dependent kernel that preserves the geometry of the k closest neighbors of each data point. It is similar to kernel PCA with a different kernel matrix. Eigen-analysis can then be performed on the kernel matrix to extract the relevant eigenvalues and eigenvectors. Future work is needed to determine the number of k neighbors and to extract the relevant d eigenvectors and eigenvalues needed to construct the necessary LLE control chart.

6.2.2 Artificial neural networks

Artificial neural networks (ANN) were originally motivated from the study of neural interconnection in order to mimic the computation structure of the human brain. This technique maps input and output non-linearly in separate layers. Therefore these layers are connected in such a way to form a very complex network where each signal will propagate according to the input. Many studies have shown the potential of combining pattern recognition with ANN to detect and diagnose faults. Future studies might include process mapping of non-linear processes and adding elements of robustness in the algorithm.

List of References

[1] L.H. Chiang, E.L. Russell, and R.D. Braatz,*Fault Detection and Diagnosis in Industrial Systems*, Springer, 2001.

[2] K. Yang and J. Trewn, *Multivariate Statistical Process Control with Industrial Application*, ASA-SIAM, 2002.

[3] D. C. Montgomery, *Introduction to Statistical Quality Control*, John Wiley & Sons, 2005.

[4] K. Yang and J. Trewn, *Multivariate Statistical Methods in Quality Management*, Mc Graw Hill Professional, 2004.

[5] M. Hubert, P.J. Rousseeuw, and K.V. Branden, "ROBPCA: a new approach to robust principal component analysis," *Technometrics*, vol. 47, pp. 64-79, 2005.

[6] K.H. Chen, D.S. Boning, and R.E. Welch, "Multivariate statistical process control and signature analysis using eigenfactor detection methods," *Proc. Symposium on the Interface of Computer Science and Statistics*, Costa Mesa, CA, June 2001

[7] J.A. Vargas, "Robust estimation in multivariate control charts for individual observations," *Journal of Quality Technology*, vol. 35, no. 4, pp. 367-376, 2003.

[8] N.D. Tracy, J.C. Young, and R.L. Mason, "Multivariate quality control charts for individual observations," *Journal of Quality Technology*, vol. 24, no. 22, pp. 88-95, 1992.

[9] J. H Sullivan and W.H. Woodall, "A comparison of multivariate control charts for individual observations," *Journal of Quality Technology*, vol. 28, no. 24, pp. 398-408, 1996.

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

[11] P. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.

[12] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, Wiley, 1986.

[13] A. Ben Hamza and H. Krim, "Image denoising: a nonlinear robust statistical approach," *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp. 3045-3054, December 2001.

[14] P.J. Rousseeuw and A.M Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, NY, 1987.

[15] A.C. Atkinson and H.M. Mulira, "The stalactite plot for the detection of multivariate outliers," *Statistics and Consulting*, vol. 3, pp. 27-35, 1993.

[16] F.A. Alqallaf, K.P. Konis, and R.D. Martin, and R.H. Zamar, "Scalable robust covariance and correlation estimates for data mining," *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, pp. 14-23, 2002.

[17] J. H Sullivan and W.H. Woodal, "A comparison of multivariate control charts for individual ibservations," *Journal of Quality Technology*, vol. 28, no. 24, pp. 398-408, 1996.

[18] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2nd edition, 1998.

[19] B. Scholkopf, A. Smola, and K-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

[20] E.B. Martin, A.J. Morris, and J. Zhang,"Process performance monitoring using multivariate statistical process control," *IEEE Process Control Theory Applied*, vol 143, pp. 132-144, 1996.

[21] H. Zhang, A.K. Tangirala, and S.L Shah,"Dynamic process monitoring using multiscale PCA," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, vol 3, pp. 1579-1584, 1999.

[22] M.S. Chen, J. Han, and P.S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering* vol. 8, pp. 866-883, 1996.

[23] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996.

[24] P.J. Rousseeuw, and K.V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212-223,1999.

[25] M. Hubert, P.J. Rousseeuw, and K.V. Branden, "ROBPCA: a new approach to robust principal component analysis," *Technometrics*, vol. 47, pp. 64-79, 2005.

[26] P. Filzmoser, "A multivariate outlier detection method" *Proc. International Conference on Computer Data Analysis and Modeling*, vol. 1, pp. 18-22, 2004.

[27] W.J. Egan and S.L. Morgan, "Outlier detection in multivariate analytical chemical data," *Analytical Chemistry*, vol. 70, pp. 2372-3279, 1998.

[28] F. Angiulli, S. Basta, C. Pizzuti. "Distance-based detection and prediction of outliers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 145-160, 2006.

[29] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification*, 2nd Edition, Wiley Interscience, 2000.

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

[31] C.C. Aggarwal, and P.S. Yu, "Oulier detection for high dimensional data," *Proc. ACM SIGMOD*, 2001.

[32] F.A. Alqallaf, K.P. Konis, and R.D. Martin, "Scalable robust covariance and correlation estimates for Data Mining," *Proc. ACM SIGKDD*, 2002.

[33] P.J. Rousseeuw and A.M Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, NY, 1987.

[34] S. Engelen, M. Hubert, and K. Vanden Branden, "A comparison of three procedures for robust PCA in high dimensions," *Austrian Journal of Statistics*, vol. 34, pp. 117-126, 2005.

[35] L.H. Chiang, R.J Pell, and M.B.Seasholtz,"Exploring process data with the use of robust outlier detection algorithms," *Journal of Process Control*, vol. 13, pp. 437-449, 2003.

[36] Z. Ge and Z. Song,"Process monitoring based on independent component analysis-principal component analysis (ICA-PCA) and similarity factors," *Industrial & Engineering Chemistry Research* , vol. 46, pp. 2054-2063, 2007.

[37] M. Kano, S. Tanaka, S. Hasebe, I. Hashimoto, and H. Ohno, "Combination of independent component analysis and principal component analysis for multivariate statistical process control," *Proc. International Symposium on Design, Operation and Control of Chemical Plants* , pp. 319-324, 2002.

[38] A. Hyvarinen and E. Oja,"Independent component anlysis: algorithms and application," *Neural Networks*, vol. 13, pp. 411-430, 2000.

[39] J.M. Lee, C. Yoo, and I.B Lee,"Statisical process monitoring with independent component anlysis," *Journal of Process Control*, vol. 14, pp. 467-485, 2004.

[40] H. Al-Bazzaz and X.Z. Wang,"New statistical process control chart for Batch operations based on independent component analysis," *Industrial & Engineering Chemistry Research* , vol. 43, pp. 6731-6741, 2004.

[41] W. Ku, R.H. Storer, and C. Georgakis,"Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and Intellignt Labaratory Systems* , vol. 30, pp. 179-196, 1995.

[42] J. Mina and C. Verde, "Fault detection using dynamic principal component analysis by average

estimation," *Electrical and Electronics Engineering, 2005 2nd International Conference on Electrical Engineering,*pp. 374-377 2005.

[43] W. Li and S. J. Qin,"Consistent dynamic PCA based on errors-in-variables subspace identification," *Journal of Process Control*, vol. 11, pp. 661-678, 2001.

[44] J. Chen and C. M. Liao,"Dynamic process fault monitoring based on neural network and PCA," *Journal of Process Control*, vol. 12, pp. 277-289, 2002.

[45] H. Zhang, A.K. Tangirala, and S.L Shah,"Dynamic process monitoring using multiscale PCA," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 3, pp. 1579-1584, 1999.

[46] J. Liang and N. Wang,"Fault diagnosis in industrial reheating furnace using principal component analysis," *IEEE Conference Neural Networks & Signal Processing*, vol. 2, pp. 1615-1618, 2003.

[47] J.F. Cardoso and A. Soulomica, "Blind beamforming for non-Gaussian signals,"*IEEE Proc. F, Radar & Signal Processing*, vol. 140, pp. 362-370,1993.

[48] W. Hardle,*Smoothing Techniques*, Springer, 1991.

[49] J.J. Downs and E.F. Vogel,"A plant-wide Industrial process control problem," *Computers & Chemical Engineering*, vol. 17, pp. 245-255, 1993.

[50] M. Kano, K. Nagao, S. Hasebe, I. Hashimoto, H. Ohno, R. Strauss, and B. Bakshi"Comparison of statistical process monitoring methods: application to the Estman challenge problem," *Computers & Chemical Engineering*, vol. 24, pp. 175-181, 2000.

[51] C. Lee, S.W. Choi, I.B. Lee"Variable reconstruction and sensor fault identification using canocical variate analysis," *Journal of Process Control*, vol. 16, pp. 747-761, 2006.

[52] G. Stefatos and A. Ben Hamza, "Multivariate robust quality control chart for outlier detection," revised & resubmitted to *Quality Engineering Journal*, 2007.

[53] G. Stefatos, Yan Luo, and A. Ben Hamza, "Kernel principal component chart for defect detection," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, Vancouver, Canada, 2007.

[54] G. Stefatos and A. Ben Hamza, "Statistical process control using kernel PCA," *Proc. 15th IEEE Mediterranean Conference on Control and Automation,* Athens, Greece, 2007.

[55] G. Stefatos and A. Ben Hamza, "Cluster PCA for outliers detection in multivariate data," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Montreal, 2007.

[56] G. Stefatos and A. Ben Hamza, "Dynamic fault detection and diagnosis using independent component analysis," to be submitted, 2007.