Dipl.-Ing. Peter SCHEIBELHOFER

Robust Multivariate Process Control of Multi-Way Data With Applications in Semiconductor Manufacturing

DISSERTATION

zur Erlangung des akademischen Grades

Doktor/in der technischen Wissenschaften/Naturwissenschaften

eingereicht an der

Technischen Universität Graz

Betreuer/in:
Univ.-Prof. Dipl.-Ing. Dr.techn. Ernst STADLOBER

Institut für Statistik

Graz, Jänner 2015

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZ-online hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral dissertation.

Datum/Date	Unterschrift/Signature

Danksagung

Ich möchte mich ganz herzlich bei meinem Betreuer und Doktorvater Ernst Stadlober bedanken. Danke für die hervorragende und motivierende Betreuung und die immens lehrreiche Zusammenarbeit.

Meinen ebenso herzlichen Dank möchte ich der Firma ams AG aussprechen für die Möglichkeit diese Dissertation in einer sehr fruchtbaren und lehrreichen Zusammenarbeit zu erstellen. Spezieller Dank ergeht an meinen Betreuer und Mentor bei ams, Günter Hayderer, von dem ich weit über das Fachliche hinaus viel lernen konnte. Weiters bedanken möchte ich mich bei Peter Kailbauer, Günter Leditzky und Dietmar Gleispach für viel Motivation, Unterstützung und Freundschaft. Danke auch an Florian Fratte für seine Mithilfe bei der Implementierung der Spotfire-Oberfläche und seine umfassende Expertise.

Ganz besonderer Dank ergeht an meine Eltern, deren Unterstützung und Rückhalt unendlich wichtig waren. Ohne Euch wäre alles das nicht möglich gewesen.

Für meine Eltern.

Abstract

The monitoring and evaluation of the manufacturing process condition is a crucial challenge in semiconductor manufacturing. During production steps each production unit (wafer) records data information from multiple process variables and at multiple time points. This results in multi-way data arrays. In this work we present a generalized methodology for multivariate process control of such multi-way arrays by using multi-way principal component analysis. Kernel techniques allow the approach to also capture nonlinear relationships as frequently observed in semiconductor process data. Special attention is also paid on the robustness of the approach. In two case studies, observed changes in production processes can be detected and their root causes can be tracked down successfully. For the multivariate monitoring of post-production checks a software user interface has been created that makes the created models applicable for process engineers.

The thesis was written in cooperation with Austrian semiconductor manufacturer ams AG.

Zusammenfassung

Die Überwachung und Bewertung des Prozesszustandes ist eine zentrale Herausforderung in der Halbleiterherstellung. Während der Produktionsschritte werden für jede Produktionseinheit (Wafer) Daten von mehreren Prozessvariablen und zu mehreren Zeitpunkten aufgezeichnet. Das führt zu sogenannten Multi-way Datenanordnungen. In der vorliegenden Arbeit stellen wir eine verallgemeinerte Methodik für die mehrdimensionale Prozesskontrolle solcher Multi-way Datenanordnungen basierend auf der Multi-way Hauptkomponentenanalyse vor. Durch die Verwendung von Kernschätzern können auch nichtlineare Zusammenhänge, wie sie häufig bei Halbleiterprozessen beobachtet werden, berücksichtigt werden. Auch auf die statistische Robustheit des vorgestellten Verfahrens wird eingegangen. In zwei Fallstudien werden Änderungen im Produktionsprozess detektiert und ihre Ursache herausgefunden. Für die mehrdimensionale Überwachung von Qualitätskontrollen nach der Herstellung wurde eine Software-Oberfläche generiert, welche die erstellten Modelle für Prozessingenieure anwendbar macht.

Die Arbeit wurde in Kooperation mit dem österreichischen Halbleiterhersteller ams AG erstellt.

Contents

List of Figures

List of Tables

Symbols & Abbrevations

$\mathbf{x}$	column vector
$\mathbf{x}^T$	row vector as transposed column vector
x	scalar
X	scalar or random variable, as indicated
$\mathbf{X}$	matrix or array, as indicated
$\lVert\mathbf{x}\rVert$	Euclidean norm of $\mathbf{x}$
`x`	variable assigned in `R`
$\lVert\mathbf{P}\rVert$	$= \sqrt{\sum_{k=1}^{K}\sum_{j=1}^{J}\mathbf{P}[k,j]^2}$ for matrix $\mathbf{P}$
$\mathbf{E}^T[j,i,k]$	$= \mathbf{E}[i,j,k]$, three-way array $\mathbf{E}$ transposed in its first two dimensions
$\mathbf{t}\otimes\mathbf{P}$	$= \mathbf{t}[i]\mathbf{P}[k,j]$ for a vector $\mathbf{t}$ and a matrix $\mathbf{P}$
$\mathbf{E}^T\cdot\mathbf{t}$	$= \mathbf{E}^T[j,i,k]\cdot\mathbf{t}[i] = \sum_{i=1}^{I}\mathbf{E}^T[j,i,k]\mathbf{t}[i]$ for three-way array $\mathbf{E}$ and vector $\mathbf{t}$
$\mathbf{E}\odot\mathbf{P}$	$= \sum_{j=1}^{J}\sum_{k=1}^{K}\mathbf{E}[i,j,k]\mathbf{P}[k,j]$ for three-way array $\mathbf{E}$ and matrix $\mathbf{P}$
APC	Advanced Process Control
CPCA	Consensus Principal Component Analysis
FDC	Fault Detection and Classification
KPA	Kernel Parallel Analysis
KPCA	Kernel Principal Component Analysis
MCD	Minimum Covariance Determinant
MYT	Mason-Young-Tracy
NIPALS	Nonlinear Iterative Partial Least-Squares
PARAFAC	Parallel Factor Analysis
PC	Principal Component

PCA	Principal Component Analysis
RBF	Radial Basis Function
RMCD	Reweighted Minimum Covariance Determinant
RobMBMWKPCA	Robust Multi-Block Multi-Way Kernel Principal Component Analysis
SPC	Statistical Process Control
SPE	Squared Prediction Error
SVD	Singular Value Decomposition
TERR	TIBCO Enterprise Runtime for `R`
UCL	Upper Control Limit
WAT	Wafer Acceptance Test

Chapter 1

Introduction

The evaluation of the manufacturing process condition is a crucial challenge in modern semiconductor fabrication. The ever growing process complexity as well as novel developments like 180 nm manufacturing or 3D device manufacturing lead to a growing amount of recorded data. Large numbers of process parameters are recorded by sensors during each equipment operation in the production line as well as during various post-production quality and functionality tests. Strict quality and yield requirements as well as pressure of competition demand advanced monitoring strategies and techniques of advanced process control (APC) to evaluate the collected data. APC summarizes data-driven efforts for process monitoring, analysis and modelling such as fault detection and classification (FDC), intelligent maintenance planning (predictive maintenance) and data-based replacement of physical measurements (virtual metrology).

The numerous physical processes underlying semiconductor manufacturing are highly complex, and process parameters are often strongly interrelated. Thus, APC methods often employ multivariate approaches to capture these complex relationship structures. Multivariate statistical process control (MSPC) techniques are more and more common in order to monitor the performance of such processes based on the recorded data. In order to set up adequate MSPC models the assessment of the normal operating condition of a process is crucial as data of future production is compared to this reference situation using multivariate measures. Typically, the reference situation is modelled based on a historic data set of normal operating condition data, i.e., data without outliers and abnormalities. For standard modelling approaches outliers can have undesired effects on the estimation of the reference. This can result in less powerful models. Robust methods to estimate the true normal operating condition in an unaffected way are recommended and often superior to their classical, non-robust counterparts. Thus, they are well suited for the task of setting up MSPC models.

In this work we present advanced process control methods for robust multivariate monitoring of various aspects of a semiconductor production line. Figure 1.1 schematically outlines such a production line.

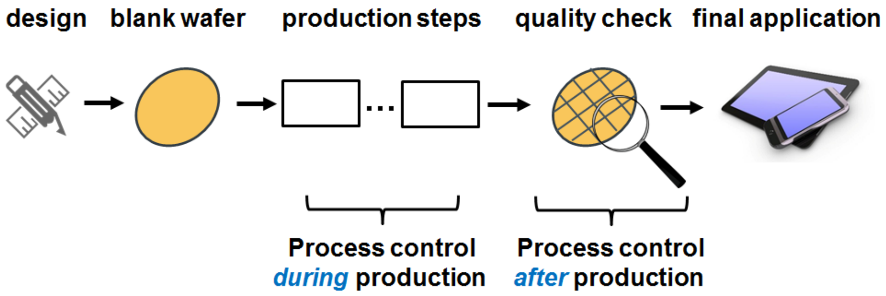


Figure 1.1: Schematic outline of a semiconductor manufacturing line.

In the initial design phase a chip's composition is defined based on its final application, and the operational ranges of the relevant electrical parameters are specified. Then the blank wafer goes through a large number of complex processing steps such as lithography, diffusion, ion implantation, etching or film deposition. Afterwards, wafer acceptance tests are used to examine the functioning of the resulting chips on a wafer and check if the relevant electrical parameters are in fact within their predefined operational ranges.

We present methodologies for robust multivariate process control of both the actual processing steps (process control during production) and wafer acceptance test results (process control after production).

Many processing steps like etching, chemical vapor deposition (CVD) or physical vapor deposition (PVD) process wafers sequentially over a certain processing time. This results in multiple multivariate measurement vectors over the course of the wafer processing. The recorded data is referred to as *trace data* or *multi-way data*. Furthermore, the underlying physical relationships can lead to nonlinear behaviour. The proposed method for monitoring such production steps is constructed to be able to handle both multi-way data and possible nonlinearities by using a kernel approach. Resulting monitoring models are able to detect and diagnose faults, even in real-time during the actual wafer processing.

We also propose a method to monitor wafer acceptance test (WAT) results in a multivariate way. The resulting models are able to take into account the correlations among the electrical WAT parameters. This gained information is considered highly valuable as usual WAT parameter monitoring is of univariate type. However, the electrical parameters are often interrelated and correlated. With the proposed method, abnormalities in the multivariate relationships among parameters can be detected and diagnosed using multivariate decomposition methods. In addition, a user interface was constructed that lets process engineers use the constructed models to monitor WAT data in a simple way.

The proposed methods offer the possibility of advanced and in-depth performance monitoring in a more and more complex and challenging production environment. Especially, the proposed kernel and multi-way approaches allow the evaluation and extraction of the most important process information. For the monitoring of production process steps these methods enable a novel evaluation of the health of the production as well as the health of the corresponding production equipment. Special attention is also paid on the robustness of the approaches in order to be able to estimate true process conditions unaffected by one-time effects, measurement errors or other abnormal behaviour.

This thesis is composed as follows. Chapter 2 summarizes multivariate process monitoring methods based on a robust version of Hotelling's T^2 statistic. Furthermore, it deals with distribution-free control limit estimation. Chapter 3 gives an overview on classical linear fault detection and diagnosis based on principal component analysis (PCA) and multi-way approaches. Chapter 4 discusses nonlinear process monitoring based on kernel PCA. Based on these fundamentals, a novel approach for advanced fault detection and diagnosis using robust multi-way multi-block kernel PCA is introduced in chapter 5. Chapter 6 and 7 then exemplariliy present applications of the proposed methods. One for advanced process control of a plasma etch equipment (chapter 6) and one for robust multivariate monitoring of WAT data of a specific process type (chapter 7).

All computations were performed in `R`, version 3.1.1 (see R Core Team (2014)). The package `knitr` (see Xie (2014)) was used to directly connect `R` code and results with the LaTeX output.

Chapter 2

Robust Multivariate Process Control

2.1 Hotelling's T^2

The majority of modern industrial processes rely on the behaviour of a set of parameters working together in order to create a desired production output. The parameters or variables of such multivariate processes are often interrelated and form a correlated set, i.e., the single variables do not behave independently of each another. In order to monitor such complex processes multivariate methods are needed that are capable of mapping the interrelated structure. They have to be preferred over univariate methods where process variables are monitored separately and where relationships between variables are neglected.

For the task of monitoring multivariate process data Hotelling's T^2 statistic (see Hotelling (1931)) is considered the standard method. It is named after Harold Hotelling who was among the first to apply it for multivariate quality control purposes (see Hotelling (1947)). The statistic maps the performance of a multivariate process to a univariate statistic, i.e., its values can be monitored in a univariate statistical process control (SPC) chart and a control limit can be given. Thus it is well suited for multivariate statistical process control and fault detection, especially with industrial applications (see Mason and Young (2002)).

Specifically, the T^2 statistic measures the distance of points to their center in a p dimensional space. It differs from the classical Euclidean distance measure in that it considers the variation of the involved variables via their covariance matrix.

For an individual p-variate normally distributed vector $\mathbf{x} = (x_1, \dots, x_p)^T$ with expectation vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ Hotelling's T^2 statistic is defined as

$$T^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}).$$

Under these assumptions it holds

$$T^2 \sim \chi^2_p$$

where χ^2_p represents a chi-square distribution with p degrees of freedom.

For a given sample of n observations in p variables the usually unknown parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be estimated from the given sample. For each observation $i = 1, \dots, n$ we have an observation vector $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ where the data are given as $n \times p$ matrix $\mathbf{X}$, i.e.,

$$\mathbf{X} = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}.$$

The corresponding classical sample mean vector is $\overline{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_p)^T$ where

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}, \quad j = 1, \dots, p.$$

The elements $\mathbf{S}_{jh}$ of the corresponding classical sample covariance matrix $\mathbf{S}$ are calculated as

$$\mathbf{S}_{jh} = \frac{1}{n-p} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ih} - \bar{x}_h), \quad j, h \in \{1, \dots, p\}.$$

This classical estimation approach leads to the sample version of Hotelling's T^2. For each multivariate observation $i = 1, \dots, n$ it is given by

$$T_i^2 = (\mathbf{x}_i - \overline{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i - \overline{\mathbf{x}}). \tag{2.1}$$

In the following we omit the subscript i and simply refer to T_i^2 as T^2 or T^2 value of observation $\mathbf{x}_i$.

Usually, a T^2 control chart for multivariate process control is implemented in two stages, a retrospective modeling stage usually referred to as *phase 1* and a monitoring stage referred to as *phase 2*. In phase 1 a historical data set (HDS) of in-control observations or "good" observations is needed to estimate the in-control mean vector and covariance matrix, i.e., process mean and covariance under normal operating conditions. In phase 2 new observations are compared to the in-control situation characterized in phase 1 in terms of their T^2 distance by using the HDS estimates.

Assume the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the underlying p-variate normal distribution are unknown and are estimated using the classical sample estimates as discussed above. In phase 1 these estimates are obtained using a HDS in the form of $\mathbf{X}$. In this case it holds

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

where $B_{(p/2,(n-p-1)/2)}$ represents a beta distribution with parameters $p/2$ and $(n-p-1)/2$.

In phase 2 the observation vector $\mathbf{x}$ is independent of the computation of $\overline{\mathbf{x}}$ and S. In this case it holds

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

where $F_{(p,n-p)}$ represents an F distribution with p and $(n-p)$ degrees of freedom (see Tracy et al. (1992)).

Based on these distributional properties an upper control limit (UCL) for the T^2 values can be determined and a control chart can be constructed to label phase 2 observations *in-control* or *out-of-control* (or *signaling*). Thereby the UCL simply represents the $(1-\alpha)$ quantile of the corresponding T^2 distribution where α is the false alarm rate.

2.2 MYT Decomposition

If an observation is labeled out-of-control by the T^2 procedure, i.e., its T^2 value is larger than the determined UCL, one is interested in the root cause. The Mason-Young-Tracy (MYT) decomposition of a T^2 signal offers the possibility to determine the involved variables (see Mason et al. (1995)). It does so by decomposing a T^2 value into independent parts that can be related to the p input variables or combinations of them. However, for a given T^2 value there are several possibilities to decompose it using the MYT-approach. For the general case with variables $x_1, \ldots, x_p$ one possible decomposition is given by

$$\begin{aligned}
T^2 &= T^2_{(x_1,\ldots,x_p)} = T^2_1 + T^2_{2.1} + T^2_{3.1,2} + \cdots + T^2_{p.1,2,\ldots,p-1} \\
&= T^2_{(x_1,\ldots,x_{p-1})} + T^2_{p.1,2,\ldots,p-1}
\end{aligned}$$

where the subscript $(x_1, \ldots, x_p)$ denotes that the T^2 signal is computed using the variables x_1 to x_p. T^2-terms with subscripts of the form $p.1, 2, \ldots, p-1$ are called conditional terms and denote relationships among variables. They can be computed iteratively via

$$\begin{aligned}
T^2_{p.1,2,\ldots,p-1} &= T^2 - T^2_{(x_1,\ldots,x_{p-1})}, \\
T^2_{p.1,2,\ldots,p-2} &= T^2_{(x_1,\ldots,x_{p-1})} - T^2_{(x_1,\ldots,x_{p-2})}, \\
&\vdots \\
T^2_{2.1} &= T^2_{(x_1,x_2)} - T^2_{(x_1)}
\end{aligned}$$

where the conditional term $T^2_{2.1}$ denotes the T^2 value of the relationship between x_1 and x_2, i.e., it indicates if there is a problem with the correlation. The term T^2_1 is the T^2 value determined using only the variable x_1. Such single variable terms are called unconditional.

The control limit for each term can be determined using the distributional properties of T^2 in the multivariate normal case and p accordingly. For unconditional terms the T^2 distributions are given by

$$T^2_j \sim \Big(\frac{n+1}{n}\Big) F_{(1,n-1)}, \quad j = 1, \ldots, p,$$

and for conditional terms it is given by

$$T^2_{j.j_1,\ldots,j_k} \sim \Big[\frac{(n+1)(n-1)}{n(n-k-1)}\Big] F_{(1,n-k-1)}, \quad j_1, \ldots, j_k \in \{1, \ldots, j-1\}$$

where k equals the number of conditioned variables. The respective UCLs are then obtained using $(1-\alpha)$ quantiles for a specified false alarm rate α.

There is a large number of possibilities for computing the MYT-decomposition if p is large. A sequential computational scheme that aims to reduce the number of computations is proposed in Mason et al. (1997).

2.2.1 Two-Dimensional Case

To illustrate the MYT decomposition we examine the case of two variables x_1 and x_2. In this case a T^2 value can be decomposed into

$$T^2 = T_1^2 + T_{2.1}^2 \quad \text{or} \quad T^2 = T_2^2 + T_{1.2}^2. \tag{2.2}$$

The unconditional terms in (2.2) are defined as

$$T_1^2 = \frac{(x_1 - \bar{x}_1)^2}{s_1^2} \quad \text{and} \quad T_2^2 = \frac{(x_2 - \bar{x}_2)^2}{s_2^2}$$

with $\bar{x}_i$ and s_i are the mean and standard deviation of variable $x_i, i \in \{1, 2\}$. These are simply 1-dimensional T^2 statistics. The conditional terms in (2.2) are given by

$$T_{2.1}^2 = \frac{(x_2 - \bar{x}_{2.1})^2}{s_{2.1}^2} \quad \text{and} \quad T_{1.2}^2 = \frac{(x_1 - \bar{x}_{1.2})^2}{s_{1.2}^2}.$$

They measure the statistical distance between the variable x_i and the conditional mean $\bar{x}_{i.j}$ for $i, j \in \{1, 2\}$, $i \neq j$. If $\bar{x}_{i.j}$ is out-of-control the observed value of x_i is not where it is expected to be for the given value of x_j, i.e., the relationship between x_i and x_j as it is described by the phase 1 covariance matrix has changed.

Figure 2.1 shows the acceptance regions associated with both conditional terms, i.e., the geometrical interpretations of $T_{2.1}^2$ and $T_{1.2}^2$, respectively. For $x_1 = a$ corresponding values of x_2 are accepted if they lie in the blue region, considering the relation to x_1 (left panel). For $x_2 = b$ values of x_1 are accepted if they lie in the red region (right panel). The line labeled as $\bar{x}_{2.1}$ denotes the regression line that results from modeling x_2 using only x_1 as explanatory variable, i.e., the estimated conditional mean of x_2 depending on x_1 (and vice versa for $\bar{x}_{1.2}$).

An example with four different out-of-control observations is shown in figure 2.2.

Point A in figure 2.2 signals unconditionally in T_2^2 and conditionally in $T_{2.1}^2$. Point B signals only in $T_{2.1}^2$ as the residual to the regression line $\bar{x}_{2.1}$ is too large compared to phase 1. This means that the observed value of x_2 differs from the value predicted by x_1 where the prediction is derived from phase 1. Point C signals in $T_{1.2}^2$ and point D signals in both conditional terms.

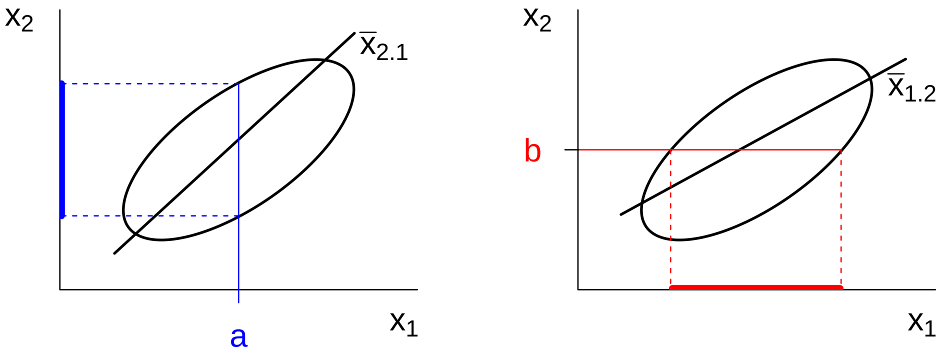


Figure 2.1: Acceptance regions associated with conditional T^2 terms.

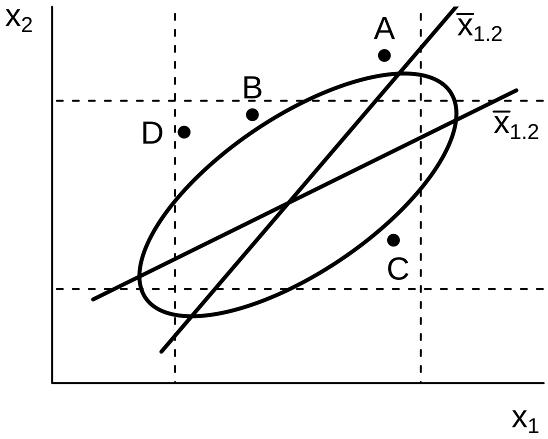


Figure 2.2: Univariate control limits (dashed) and four points with out-of-control T^2 values.

2.3 Robust T^2 Control Charts

The T^2 control chart given in equation (2.1) is based on the classical empirical estimates of mean and covariance. This makes it highly sensitive to outliers. This is due to the fact that these classical estimators are not robust. In addition, the assumption that phase 1 data comes from an in-control process is not always valid. Thus the underlying HDS on which the estimations are based has to be constructed carefully.

One way of constructing a proper HDS is to iteratively remove outliers and recompute the UCL. Then the final HDS is the data set where no outliers remain. This procedure can be cumbersome and may fail to detect moderate outliers. Also, it is well known that due to the *masking effect* multiple outliers may go undetected due to their effect on the estimators (see Vargas (2003)). Thus they do not necessarily have a large T^2 value. Furthermore, it can be shown that a T^2 chart based on the classical covariance estimator is not effective in detecting mean vector shifts or trends (see Sullivan and Woodall (1996), Sullivan and Woodall (1998)).

Alternatively, one can use robust estimates of mean and covariance to calculate the T^2 statistic, i.e.,

$$T^2 = (\mathbf{x} - \overline{\mathbf{x}}_{\text{rob}})^T \mathbf{S}_{\text{rob}}^{-1} (\mathbf{x} - \overline{\mathbf{x}}_{\text{rob}}). \tag{2.3}$$

Robust estimators are less affected by outlying observations as they reduce or remove their effects. Furthermore the probability of detecting outliers can be improved by the use of robust estimators (see Vargas (2003)).

Figure 2.3 shows the difference between classical and robust covariance estimation and the resulting effect on a T^2 ellipse with 95% confidence level. The classical estimation (red ellipse) is heavily influenced by the outlying points whereas the robustly estimated blue ellipse only focuses on the central data cloud.

A popular robust estimator of multivariate location and scatter is the minimum covariance determinant (MCD) estimator (see Rousseeuw (1984), Rousseeuw and Van Driessen (1999)). The MCD algorithm looks for a subset of size $n/2 \leq h < n$ of the data whose covariance matrix has the smallest determinant. The corresponding multivariate mean and covariance matrix (multiplied by a consistency factor) of the determined h-subset then serve as MCD estimators of multivariate location and scatter. These estimators are highly robust and have favourable asymptotics (see Butler et al. (1993)).

Another nice property of the MCD is its high breakdown point, i.e., the

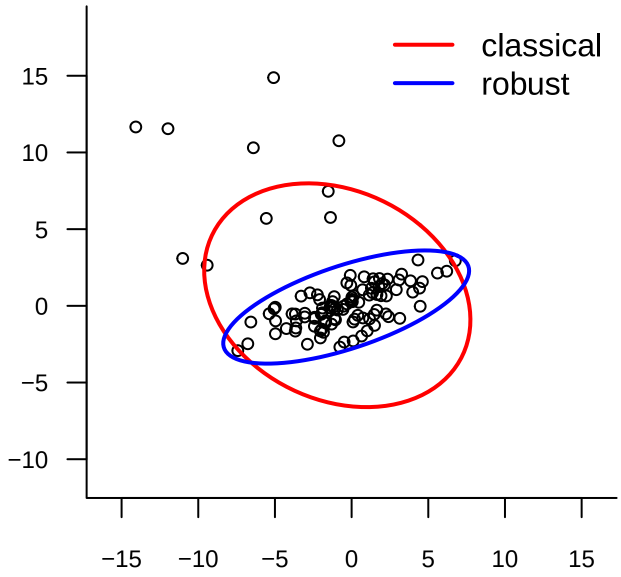


Figure 2.3: Comparison of classical and robust estimation of T^2 ellipses on 100 simulated data points from the bivariate normal distribution $BN(0, 0, \sigma_x = 10, \sigma_y = 2, \rho_{xy} = 3)$ and 10 uniformly distributed outliers from distribution $(U_x(-15, 0), U_y(0, 15))$.

smallest portion of outlier contamination in the data that can have an arbitrarily large influence on the estimators. The MCD reaches its highest possible breakdown value of $(n-p+2)/2$ when $(n+p)/2 \leq h \leq (n+p+1)/2$ (see Lopuhaä and Rousseeuw (1991), Hubert and Debruyne (2010)). Furthermore, the MCD estimators $(\hat{\boldsymbol{\mu}}_{\text{MCD}}(X), \hat{\boldsymbol{\Sigma}}_{\text{MCD}}(X))$ are called *affine equivariant*, i.e., for a dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$ they satisfy

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_{\text{MCD}}(\mathbf{XA}+\mathbf{b}) &= \mathbf{A}\hat{\boldsymbol{\mu}}_{\text{MCD}}(\mathbf{X})+\mathbf{b} \\
\hat{\boldsymbol{\Sigma}}_{\text{MCD}}(\mathbf{XA}+\mathbf{b}) &= \mathbf{A}^T\hat{\boldsymbol{\Sigma}}_{\text{MCD}}(\mathbf{X})\mathbf{A}
\end{aligned}$$

for any $\mathbf{b} \in \mathbb{R}^p$ and any nonsingular matrices $\mathbf{A} \in \mathbb{R}^{p \times p}$. This ensures that the estimators behave properly under affine transformations of the underlying data. It makes the analysis independent of measurement scales of the variables as well as translations or rotations of the data (see Hubert et al. (2008), section 2.2).

A statistically more efficient enhancement of the MCD estimator is the reweighted MCD (RMCD) estimator (see Willems et al. (2002), Croux and Haesbroeck (1999)). The RMCD procedure uses the weighted mean and weighted covariance estimators where the weights are based on robust distances (see Rousseeuw and Van Zomeren (1990)), i.e., Mahalanobis distances based on MCD estimators.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a random sample from a continuous distribution F in $\mathbb{R}^p$. The RMCD estimators of $\boldsymbol{\mu}$ and Σ are the weighted mean

$$\overline{\mathbf{x}}_{\text{RMCD}} = \frac{\sum_{i=1}^{n} w_i \mathbf{x}_i}{\sum_{i=1}^{n} w_i}$$

and the weighted covariance matrix

$$\mathbf{S}_{\text{RMCD}} = c_{\eta,p} d_{n,p}^{\eta} \frac{\sum_{i=1}^{n} w_i (\mathbf{x}_i - \overline{\mathbf{x}}_{\text{RMCD}})(\mathbf{x}_i - \overline{\mathbf{x}}_{\text{RMCD}})^T}{\sum_{i=1}^{n} w_i}. \tag{2.4}$$

The weights are based on robust distances determined by the MCD estimators $\overline{\mathbf{x}}_{\text{MCD}}$ and S_{MCD}, i.e.,

$$D(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \overline{\mathbf{x}}_{\text{MCD}})^T \mathbf{S}_{\text{MCD}}^{-1} (\mathbf{x}_i - \overline{\mathbf{x}}_{\text{MCD}})}.$$

The values of w_i, $i = 1, \ldots, n$ are then determined using a cutoff value based on the chi-square distribution for the robust distance measures $D(x_i)$, i.e.,

$$w_i = \begin{cases} 1 & \text{if } D(\mathbf{x}_i) \leq \chi^2_{p,\eta} \\ 0 & \text{otherwise,} \end{cases}$$

where $\chi^2_{p,\eta}$ is the ηth quantile of the chi-square distribution with p degrees of freedom, i.e., observations above the cutoff value $\chi^2_{p,\eta}$ are downweighted. As suggested in Rousseeuw and Van Zomeren (1990) and Chenouri et al. (2009) a value of $\eta = 0.975$ is typically used.

Furthermore, in expression (2.4) the factor $c_{\eta,p}$ makes $\mathbf{S}_{\text{RMCD}}$ consistent under the multivariate normal distribution and the factor $d_{n,p}^{\eta}$ is a finite-sample correction factor (see Chenouri et al. (2009); Pison et al. (2002) for further details).

T^2 control charts based on RMCD estimators are more efficient than classical T^2 charts when there are outliers in phase 1 (see Chenouri et al. (2009)).

2.4 Robust Model Construction

The T^2 statistic needs the covariance matrix to be nonsingular, i.e., without exact collinearities. However, severe but not exact collinearities in the covariance matrix may also have distorting effects (see Mason and Young (2002), p. 65). Collinearities can have their origin in theoretical relationships existing in the data and in the presence of outliers. For example, semiconductor manufacturing data often show complex dependency structures due to physical relationships among the process parameters. Collinearities in the covariance matrix correspond with collinearities in the correlation matrix. Thus, highly correlated variables should be removed from the data set.

The classical estimator of the correlation ρ, Pearson's sample correlation coefficient r, is not robust. Thus, it can give misleading results under the presence of outliers in the data. In order to construct a robust T^2 model robust correlation estimators and robust correlation matrices should be used. The goal is to remove parameters from the phase 1 data that are redundant due to high absolute values of the robust correlation.

A possible approach to robust correlation is to use the correlation matrix resulting from a robust RMCD covariance matrix estimation. The robust data subset determined by the RMCD estimator is used to estimate a robust correlation matrix. This leads to robust correlation coefficient r_{RMCD}. Alternative ways to estimate the correlation ρ in a robust way can be found in Shevlyakov and Smirnov (2011).

A guideline for identifying a near-singular covariance matrix is the condition number (see Mason and Young (2002), p. 67)

$$c = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the correlation matrix, respectively. A condition number greater than 30 implies a severe collinearity. However, when based on a non-robust correlation matrix, the condition number can give misleading results under outlier contamination. In order to compute a condition number not influenced by outliers a robust correlation matrix should be used to compute the eigenvalues.

For a robust T^2 modeling approach highly correlated variables should be removed from phase 1 data until its robust condition number is smaller than 30.

2.5 Nonparametric Control-Limit Estimation

In order to rely on the distributional properties as given in section 2.1 to determine the control limit for a T^2 control chart the underlying process data has to follow a multivariate normal distribution. However, often this is not the case. Moreover, by using robust estimators instead of classical ones one also cannot rely on the same distributional assumptions as for the classical T^2 statistic.

Thus, in situations where the observation vector $\mathbf{x} = (x_1, \ldots, x_p)^T$ does not follow a p-variate normal distribution or where the T^2 statistic is computed using robust estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, nonparametric techniques can be used to determine the $(1 - \alpha)$ quantile of the T^2 values in phase 1.

A simple but rough method for determining a UCL is to use Chebyshev's inequality (see Mason and Young (2002), p. 48). It states that regardless of the distribution of a random variable Y,

$$P(\mu - k\sigma < Y < \mu + k\sigma) \geq 1 - \frac{1}{k^2} \tag{2.5}$$

where μ and σ^2 are expectation and variance, respectively, of Y and $k > 1$ is a chosen constant. In order to use (2.5) for UCL estimation the mean $\overline{T}$ and standard deviation s_T of the T^2 values are estimated. Then,

$$\text{UCL} = \overline{T} + k s_T$$

where k is determined by selecting the false alarm rate α and solving $\alpha = 1/k^2$.

For a better approximation kernel density estimation (KDE) can be used to estimate the distribution of the T^2 statistic as well as of the UCL of the T^2 chart (see Chou et al. (2001)). The UCL can be estimated using the $(1 - \alpha)$ quantile of the fitted kernel distribution function of T^2. This approach may yield a good approximation provided the sample size is reasonably large. However, KDE requires the determination of several parameters including a smoothing parameter (bandwidth), the kernel function and the number of spaced points. Furthermore, for highly skewed distributions calculations of the area of the tail region can become less accurate.

As an alternative to a KDE-based approach a UCL estimation based on nonparametric bootstrapping (see Efron (1979), Davison and Hinkley (1997)) can be used. For nonnormal and skewed distributions such as the robust T^2 this is considered a simple and convenient approach (see Phaladiganon et al.

(2011)). Based on the algorithm given in Phaladiganon et al. (2011) the procedure for UCL estimation for robust T^2 control chart works as follows.

Bootstrap procedure for UCL determination

1. Compute the robust T^2 values of n observations from phase 1 data using (2.3). For example, use RMCD estimators.
2. Let $T_1^{2(i)}, T_2^{2(i)}, \ldots, T_n^{2(i)}$ be a set of n T^2 values from the ith bootstrap sample, $i = 1, \ldots, R$ (R large, e.g. $R > 1000$), randomly drawn with replacement from the initial robust T^2 values.
3. For each of the R bootstrap samples determine the $(1-\alpha)$ quantile for a user-specified false alarm rate α.
4. Determine the UCL by taking the mean or the median of the R $(1-\alpha)$ quantiles.

The determined UCL is then used to monitor new observations in phase 2, i.e., if the T^2 value of a new observation exceeds the UCL declare it as out-of-control.

2.5.1 Confidence Intervals

In order to assess the uncertainty of the UCLs estimated by bootstrapping nonparametric bootstrap confidence intervals are used (see Davison and Hinkley (1997), chapter 5). A confidence region with a specified coverage probability γ is a set $C_\gamma(y)$ of parameter values depending only on the data y which satifies

$$P\left(\theta \in C_\gamma(y)\right) = \gamma$$

for the true parameter value θ. A confidence interval is then defined by estimated limits $\hat{\theta}_\alpha$ and $\hat{\theta}_{1-\alpha}$ such that

$$P(\theta < \hat{\theta}_\alpha) = \alpha$$

for any value of α. The coverage of the resulting equi-tailed interval $[\hat{\theta}_\alpha, \hat{\theta}_{1-\alpha}]$ is then $\gamma = 1 - 2\alpha$.

Such intervals can be calculated in various ways. For a general setting, suppose that T is a continuous estimator for the scalar θ. Denote the pth quantile of $T - \theta$ by a_p, then

$$P(T - \theta \leq a_\alpha) = \alpha = P(T - \theta \geq a_{1-\alpha}).$$

With this, the $1-2\alpha$ equi-tailed interval has limits

$$\hat{\theta}_\alpha = t - a_{1-\alpha}, \quad \hat{\theta}_{1-\alpha} = t - a_\alpha. \tag{2.6}$$

As the distribution of $T-\theta$ is usually unknown, approximation methods have to be considered.

The simplest approach is a normal approximation, i.e.,

$$[\hat{\theta}_\alpha, \hat{\theta}_{1-\alpha}] = t \mp \sqrt{v} z_{1-\alpha} \tag{2.7}$$

where $z_{1-\alpha} = \Phi^{-1}(1-\alpha)$ is the $(1-\alpha)$ quantile of the standard normal distribution and v is the approximate variance of T.

If the normal approximation is inaccurate it can be replaced by resampling based methods. Based on equation (2.6), the quantiles a_α and $a_{1-\alpha}$ can be estimated by the corresponding quantiles of $T^* - t$. Approximating them by simulation leads to confidence limits

$$\hat{\theta}_\alpha = 2t - t^*_{((R+1)(1-\alpha))}, \quad \hat{\theta}_{1-\alpha} = 2t - t^*_{((R+1)\alpha)} \tag{2.8}$$

where $t^*_{(R+1)p}$ is the p quantile of t^* estimated by the $(R+1)p$th value of the R simulated and ordered values $t^*_{(1)}, \ldots, t^*_{(R)}$. These limits are generally referred to as *basic bootstrap confidence limits* for θ.

A simple modification of this is to use the normal approximation from equation (2.7), but to use bootstrapping instead of the $N(0,1)$ approximation for approximating $Z = (T-\theta)/\sqrt{V}$. The resulting bootstrap version $z^* = (t^*-t)/\sqrt{v^*}$ of Z is calculated using each simulated sample t^* and the variance estimate v^*. The p quantile of Z is then again estimated by the $(R+1)p$th value of the R simulated and ordered values $z^*_{(1)}, \ldots, z^*_{(R)}$. This leads to confidence limits

$$\hat{\theta}_\alpha = t - \sqrt{v} z^*_{((R+1)(1-\alpha))}, \quad \hat{\theta}_{1-\alpha} = t - \sqrt{v} z^*_{((R+1)\alpha)}. \tag{2.9}$$

These limits are generally referred to as *studentized bootstrap confidence limits* for θ. As outlined in Davison and Hinkley (1997), this method is, in principle, superior to the previous basic method.

In the nonparametric case where no model is assumed for the data distribution the resulting confidence intervals are similar to the parametric case. For the basic bootstrap method the only change is that the simulation model is the empirical distribution function $\hat{F}$. Otherwise, equation (2.8) applies. For the studentized bootstrap method equation (2.9) still applies. However, the standard variance estimate v can be replaced by some other variance estimates.

As further outlined in Davison and Hinkley (1997), simple confidence limit methods can be improved by applying transformations. Percentile methods use the existence of a good transformation but do not require this transformation to be known explicitely.

Suppose h is some unknown transformation of T and $U = h(T)$ with a symmetric distribution. Imagine that h is known and a $1 - 2\alpha$ confidence interval is calculated for the parameter $\phi = h(\theta)$ using the basic bootstrap method. However, the symmetry is used to write $a_\alpha = -a_{1-\alpha}$ in equation (2.6). This leads to taking $u - u^*_{((R+1)(1-\alpha))}$ instead of $u^*_{((R+1)\alpha)} - u$ and $u - u^*_{((R+1)\alpha)}$ instead of $u^*_{((R+1)(1-\alpha))}$ to estimate the α and $1 - \alpha$ quantiles of U. This changes equation (2.8) to

$$u^*_{((R+1)\alpha)}, \quad u^*_{((R+1)(1-\alpha))}.$$

The back transformation to the θ scale then is

$$t^*_{((R+1)\alpha)}, \quad t^*_{((R+1)(1-\alpha))}.$$

This can be computed without the actual knowledge of the transformation h. This method is referred to as *bootstrap percentile interval* for θ. As it turns out, the method does not work very well with nonparametric bootstrapping even for suitable transformations h. However, adjustments exist that provide successful results.

For the method to work well the estimator T should be unbiased on the transformed scale and invariant under distribution change from F to $\hat{F}$. This is usually not the case. Assume first that the data is described by a parametric model with an unknown parameter θ. The adjusted percentile method uses the assumption that for some unknown transformation h, unknown bias correction factor w and unknown skewness correction factor a the transformed estimator $U = h(T)$ satisfies

$$U \sim N(\phi - w\sigma(\phi), \sigma^2(\phi)) \quad \text{with } \sigma(\phi) = 1 + a\phi.$$

The method then uses the distribution of T^* denoted by $\hat{G}$ and

$$\begin{aligned}
\hat{G}(\hat{\theta}_\alpha) = P(T^* < \hat{\theta}_\alpha|t) = P(U^* < \hat{\phi}|u) &= \Phi\Big(\frac{\hat{\phi}_\alpha - u}{\sigma(u)} + w\Big) \\
&= \Phi\Big(w + \frac{w + z_\alpha}{1 - a(w + z_\alpha)}\Big).
\end{aligned}$$

to express the α confidence limit for θ in simulation terms as

$$\hat{\theta}_\alpha = t^*_{((R+1)\tilde{\alpha})} \quad \text{with } \tilde{\alpha} = \Phi\Big(w + \frac{w + z_\alpha}{1 - a(w + z_\alpha)}\Big).$$

These limits are generally referred to as *BC_a confidence limits*.

In the nonparametric case, the bias correction w remains the same but the skewness correction a slightly differs. For general details on the estimation of w and a see Davison and Hinkley (1997), section 5.3.

The calculation of a bootstrap UCL as well as the corresponding nonparametric equi-tailed two-sided confidence intervals can be done in `R` using the package `boot` (see Canty (2002)) and its functions `boot()` and `boot.ci()`.

Chapter 3

PCA-based Process Control

We now turn to methods of multivariate process monitoring that are based on latent structures in the involved process variables instead of the original inputs. The idea is to compute a new coordinate system formed by these latent variables where the dimension of the problem can be reduced to a most informative subset. The goal is to filter out the relevant multivariate information described by a few latent variables and separate it from noise. For high dimensional data sets this often reduces the problem complexity and allows a simplified multivariate monitoring of the essential process information.

Principal component analysis (PCA) is considered the standard method for such dimension reduction tasks (see Jackson (1991), Joliffe (2002)). PCA finds linear combinations of the input variables, i.e., directions that best describe the major variability trends in the data. It is adequate for data sets with a large number of variables and a complex correlation structure. This is often the case for data generated by physical or chemometric processes, e.g. semiconductor production processes. The PCA result is a transformation of possibly highly correlated original input variables into a set of uncorrelated latent variables or principal components (PCs). Ideally, the information needed to adequately describe the process operation is captured in a set of PCs that is smaller than the set of original inputs. Subsequently this projection to latent structures can be used as a starting point of multivariate statistical process control methods. In combination with variability measures like Hotelling's T^2 this PCA-based approach is widely used for the detection of out-of-control situations in industrial applications.

This chapter starts with a brief overview of the fundamentals of PCA-based process monitoring. Then various enhancements and adaptions such as robust PCA, multi-block PCA and multi-way PCA are reviewed and their applicability in multivariate process control, fault detection and fault diagnosis are discussed. These techniques then form the basis of nonlinear process monitoring as discussed in the next chapter.

3.1 Fundamentals

Consider a data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ composed of n observations in m variables. $\mathbf{X}$ is usually either mean centered for PCA based on the covariance matrix or autoscaled for PCA based on the correlation matrix. Robust centering and scaling can also be used. PCA decomposes $\mathbf{X}$ into a matrix $\mathbf{T} \in \mathbb{R}^{n \times R}$ of *scores*, a matrix $\mathbf{P} \in \mathbb{R}^{m \times R}$ of *loadings* and a residual (error) matrix $\mathbf{E} \in \mathbb{R}^{n \times m}$ such that

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

with $R \leq \min(m, n)$ being the number of principal components retained in the model. The columns of $\mathbf{T}$, i.e., the vectors $\mathbf{t}_r$, form an orthogonal set, thus $\mathbf{t}_i^T \mathbf{t}_j = 0$ for $i \neq j$. The columns of $\mathbf{P}$, the loadings $\mathbf{p}_r$, are eigenvectors of the covariance matrix (or correlation matrix if autoscaled), i.e.,

$$\text{cov}(\mathbf{X})\mathbf{p}_i = \lambda_i \mathbf{p}_i. \tag{3.1}$$

where λ_r is the eigenvalue associated with $\mathbf{p}_r$. The loadings are orthonormal, i.e., $\mathbf{p}_i^T \mathbf{p}_j = 0$ for $i \neq j$ and $\mathbf{p}_i^T \mathbf{p}_j = 1$ for $i = j$. The resulting space $S_P = \text{span}\{\mathbf{P}\}$ forms the so-called *principal component subspace*. Thus, the loadings contain information on how variables relate to each other. They are used to transform the original data into the principal component subspace. Thus, the scores are linear combinations of the original data $\mathbf{X}$ and the loadings, i.e.,

$$\mathbf{X}\mathbf{p}_r = \mathbf{t}_r.$$

for $r = 1, \ldots, R$ PCs. The scores contain information on how the observations are related to each other and can be seen as representation of the data in the principal component subspace. The eigenvalues λ_r associated with the pairs $(\mathbf{t}_r, \mathbf{p}_r)$ are arranged in descending order. They measure the amount of variance in the data matrix $\mathbf{X}$ that is described by the PCs.

These are direct consequences of formulating PCA as a mathematical maximization problem with constraints (see Varmuza and Filzmoser (2009), pp. 69-70). The first PC is by definition the latent variable with maximum variance of the scores, i.e., it explains as much variability of the data as possible. It is the linear combination of the variables

$$t_1 = x_1 b_{11} + \cdots + x_m b_{m1}$$

with unknown coefficients $\mathbf{b}_1 = (b_{11}, \ldots, b_{m1})^T$. Then we have $\mathrm{Var}(t_1) \to \max$ under the condition $\mathbf{b}_1^T \mathbf{b}_1 = 1$. The rth PC is determined via the maximization problem $\mathrm{Var}(t_r) \to \max$ under the constraints that the new loading vector has length one and is orthogonal to all previous loadings. Writing this as Lagrangian expression and solving it using Lagrange multipliers leads to the eigenvalue problem (3.1). The solution of the unknown $\mathbf{b}_i$ is found by taking the eigenvectors of $\mathrm{cov}(\mathbf{X})$, i.e., the loadings. The resulting eigenvalues are equal to the variances of the PCs.

For $R = \min(n, m)$ it holds $\mathbf{X} = \mathbf{T}\mathbf{P}^T$, i.e., $\mathbf{E} = 0$. Otherwise the PCs explain a portion of variability of $\mathbf{X}$. In order to determine the number of PCs to sufficiently describe the data several methods were proposed, e.g., cross-validation (see Wold (1978)) or parallel analysis (see Horn (1965), Glorfeld (1995)).

3.1.1 NIPALS

One possibility to determine the principal components is to compute all eigenvectors and eigenvalues of the estimated sample covariance matrix. Here, also robust estimates can be used. A standard procedure for this task is Jacobi rotation of the sample covariance matrix $\mathrm{cov}(\mathbf{X})$. If there are more variables than observations ($m > n$) singular value decomposition (SVD) of $\mathbf{X}$ can be used (see Varmuza and Filzmoser (2009)).

Another possibility is to compute the principal components not all at once but sequentially, i.e., one after the other. The nonlinear iterative partial least-squares (NIPALS) algorithm is based on this idea (see Geladi and Kowalski (1986), Wold et al. (1987a)). Originally developed by Herman Wold (see Wold (1966), Wold (1975)) the procedure is still popular in chemometrics, especially if the number of variables is large and only the first few principal components are of interest.

For a given data matrix $\mathbf{X}$ the algorithm is as follows.

NIPALS

1. Mean center or autoscale $\mathbf{X}$.

2. Start with an arbitrary column of $\mathbf{X}$ as initial score vector $\mathbf{t} = \mathbf{x}_j$.
   Set $\mathbf{E} = \mathbf{X}$.

3. Compute $\mathbf{p} = \mathbf{E}^T \cdot \mathbf{t}$.

4. Normalize $\mathbf{p}$ to length 1: $\mathbf{p} = \frac{\mathbf{p}}{\|\mathbf{p}\|}$.
5. Update $\mathbf{t} = \mathbf{E}\mathbf{p}$.
6. If $\mathbf{t}$ has converged with a predefined precision, a principal component has been computed; go to step 7. Otherwise iterate until convergence.
7. Update $\mathbf{E} = \mathbf{E} - \mathbf{t}\mathbf{p}^T$ and continue with step 3 for the next principal component.

The procedure of calculating a principal component by using the residual matrix of the previous principal component (step 7) is called *deflation*.

3.1.2 Process Control Using PCA

In order to apply PCA for multivariate process control tasks the first step is to establish a PCA model based on normal operating condition data. Then future process behaviour can be referenced against the in-control situation. A new m-dimensional test observation vector $\mathbf{x}^{\text{test}} \in \mathbb{R}^m$ is projected onto the principal component subspace by

$$t_r^{\text{test}} = (\mathbf{x}^{\text{test}})^T \mathbf{p}_r$$

where t_r^{test} is the corresponding score value of PC r and $r = 1, \ldots, R$. Note that $\mathbf{x}^{\text{new}}$ is mean-centered (or autoscaled) using the estimators from the in-control data set of phase 1. Here, also robust estimates can be used. The associated PCA residual $\mathbf{e}^{\text{test}} \in \mathbb{R}^m$ is given by

$$\mathbf{e}^{\text{test}} = \mathbf{x}^{\text{test}} - \hat{\mathbf{x}}^{\text{test}}$$

with $(\hat{\mathbf{x}}^{\text{test}})^T = (\mathbf{t}_R^{\text{test}})^T \mathbf{P}^T$ where $\mathbf{t}_R^{\text{test}} \in \mathbb{R}^R$ is the vector of the scores of $\mathbf{x}^{\text{test}}$ and $\mathbf{P} \in \mathbb{R}^{m \times R}$ is the loading matrix of the PCA model with R PCs. The scores can be monitored in a multivariate control chart based on Hotelling's T^2, i.e.,

$$T^2 = (\mathbf{t}_R^{\text{test}})^T \mathbf{\Lambda}_R^{-1} \mathbf{t}_R^{\text{test}} \tag{3.2}$$

where $\mathbf{\Lambda}_R \in \mathbb{R}^{R \times R}$ is the estimated covariance matrix of the scores computed from in-control data of phase 1. As the scores of different PCs are uncorrelated this is a diagonal matrix. Its main diagonal is composed by the variances of the scores, i.e., the eigenvalues of $\text{cov}(\mathbf{X})$. A control limit can be determined using the distributional assumptions of the T^2 statistic in case of multivariate normality of the scores or via nonparametric bootstrapping in nonnormal situations (see Phaladiganon et al. (2013)).

In summary, the T^2 statistic based on the first R PCs of $\mathbf{X}$ provides a test for deviations in the process variables that are of greatest importance to the variance of $\hat{X} = TP^T$ (see MacGregor and Kourti (1995), section 3.2).

However, a process monitoring procedure based solely on the first R PCs may not be sufficient. The T^2 statistic only measures deviations from the in-control situation within the principal component subspace. If a new type of abnormality occurs that deviates from the plane spanned by the PCs, the T^2 statistic may fail to detect it. In order to adequately detect and describe such events the squared prediction error (SPE, Q statistic) has to be computed (see Jackson and Mudholkar (1979)). It measures the squared perpendicular distance of an observation to the principal component subspace of the PCA model, i.e., the amount of variation not captured by the PCA model. For $\mathbf{x}^{\text{test}}$ it is given by

$$\text{SPE}_{x^{\text{test}}} = \|\mathbf{e}^{\text{test}}\|_2^2 = \sum_{i=1}^{m}(\mathbf{x}_i^{\text{test}} - \hat{\mathbf{x}}_i^{\text{test}})^2. \tag{3.3}$$

If the errors follow a normal distribution an upper control limit for the SPE statistic can be determined based on historical in-control data by using approximate distributions of quadratic forms. As shown in Box (1954) a weighted chi-square distribution is a good approximation, i.e.,

$$\text{SPE}_\alpha = g\chi^2_{h;\alpha} \tag{3.4}$$

where $g = \theta_2/\theta_1$ with $\theta_1 = \sum_{j=R+1}^{m} \lambda_j$ and $\theta_2 = \sum_{j=R+1}^{m} \lambda_j^2$, λ_j are the eigenvalues of cov($\mathbf{X}$), $\chi^2_{h,\alpha}$ is the α quantile of the chi-square distribution with h degrees of freedom and $h = \theta_1^2/\theta_2$ (see also Qin (2003), section 3.1). Another approach to estimate g and h is by using matching moments estimation technique (see Nomikos and MacGregor (1995)). Jackson and Mudholkar (1979) established an alternative upper control limit for the SPE statistic.

The assumption of normality of the errors may not always hold in practice. Thus, as with the T^2 statistic based on PCA scores, nonparametric bootstrapping can be used to determine the desired α quantile of the SPE statistic in a nonparametric way (see Phaladiganon et al. (2013)). As stated in Van Sprang et al. (2002), if the parameters of the χ^2 distribution used to determine the SPE limits are obtained directly from the moments of the sampling distribution of the historic in-control data, the approximating distribution is found to work well even in cases of nonnormal errors.

In summary, a T^2 chart to monitor the first R PCs in combination with an SPE chart to monitor the lack of PCA model fit forms an effective set for multivariate process fault detection. A fault detection index that combines both T^2 and SPE into a single measure is proposed in Yue and Qin (2001). They also propose an upper control limit for this combined measure.

The discussed PCA-based process control approach has been the subject of numerous studies. Overview and case studies can be found in Kresta et al. (1991), MacGregor and Kourti (1995), Wise and Gallagher (1996) or Qin (2003) or more recently in Kourti (2005) or Qin (2012).

3.2 Robust PCA

The classical PCA approach is based on the sample covariance matrix of the data and hence it is sensitive to outliers. In the presence of outliers the PCs, i.e., the directions of maximum variance in the data, can be distorted and attracted towards the abnormal observations. In such cases robust PCA methods instead of classical PCA should be used. Figure 3.1 shows the difference between classically and robustly computed maximum variance directions.

There are different ways to compute robust principal components (see e.g. Varmuza and Filzmoser (2009), section 3.5 or Filzmoser and Todorov (2011), section 5). The most straightforward and intuitive approach is to use robust estimation of the sample covariance or correlation matrix instead of classical estimation (see Croux and Haesbroeck (2000)). For example the MCD estimator can be used. It provides robust estimators of both the covariance matrix and the multivariate location. The location estimation can be used to robustly center the data. Then the eigenvalues can be derived from the MCD-estimated covariance matrix. The resulting PCs are highly robust against outliers. However, this approach is limited in the number of variables m. This is because the MCD estimator can only be computed if $h > m$ with $n/2 \leq h \leq n$ being the size of the robust MCD-subset. For $h \leq m$ the determinant of the resulting covariance matrix of the h-subset is zero. Thus this approach can not handle applications with $m > n$.

A second approach to robust PCA is based on the projection pursuit technique (see Li and Chen (1985), Croux and Ruiz-Gazen (2005)). The idea is to obtain the PCs by maximizing a robust measure of variance, e.g. the median absolute deviation (MAD), instead of the classical one and project the data points on the resulting directions. When maximizing a robust variance measure obtaining the desired PCs is not as simple as solving an eigenvalue problem. Thus, several approximative algorithms for robust projection-pursuit PCA have been proposed (see Croux and Ruiz-Gazen (1996), Croux and

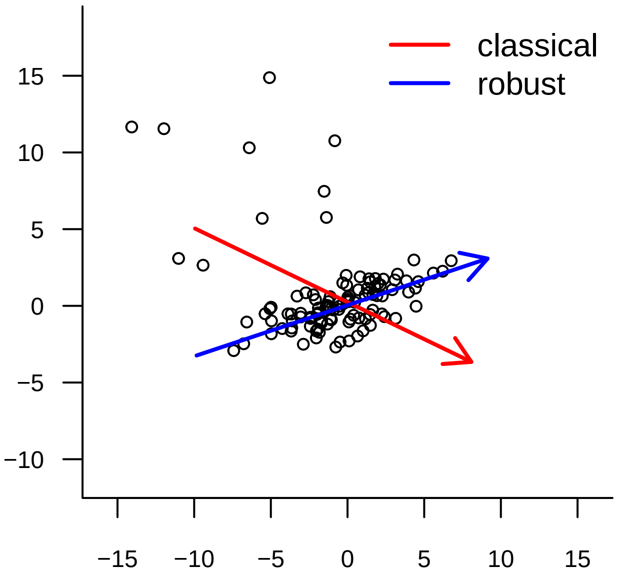


Figure 3.1: First principal component of 100 simulated multivariate normal data points and 10 outliers based on classical and robust (MCD) covariance matrix estimation.

Ruiz-Gazen (2005), Hubert et al. (2002), Croux et al. (2007)). The resulting PCA method is highly robust and works for high-dimensional applications with $m > n$.

Another idea that aims to combine the ideas of robust covariance estimation and projection-pursuit PCA is ROBPCA (see Hubert et al. (2005), Hubert et al. (2008)). At first, a singular value decomposition is applied on the robustly centered data to reduce the full data space to an affine subspace spanned by the n observations. Then the $h < n$ least outlying data points are used to compute a robust covariance matrix and to select the number of PCs to retain. In order to select these h data points the "outlyingness" of each observation is determined using the Stahel-Donoho outlyingness measure (see Stahel (1981), Donoho (1982)). It is defined as

$$r(\mathbf{x}_i) = \max_{\mathbf{v} \in B} \frac{|\mathbf{x}_i^T \mathbf{v} - \operatorname{med}(\mathbf{x}_j^T \mathbf{v})|}{\operatorname{mad}(\mathbf{x}_j^T \mathbf{v})} \tag{3.5}$$

where B is the set of all non-zero vectors, $\text{med}(\mathbf{x}_j^T\mathbf{v})$ is the median of $\{\mathbf{x}_j^T\mathbf{v}, j = 1, \ldots, n\}$ and $\text{mad}(\mathbf{x}_j^T\mathbf{v}) = \text{med}(|\mathbf{x}_j^T\mathbf{v} - \text{med}(\mathbf{x}_l^T\mathbf{v})|)$ with $l = 1, \ldots, n$ is the MAD. For the MAD to be a consistent estimator for the estimation of the standard deviation σ, it has to be standardized, i.e., $\hat{\sigma} = 1/(\Phi^{-1}(3/4)) \cdot \text{mad}$ for normally distributed data. However, for a more general formulation of (3.5) any univariate estimator $m(.)$ of location and $s(.)$ of scale can be used (e.g. see Debruyne (2009)). In the ROBPCA procedure, the median and MAD in (3.5) are replaced by the MCD location and scatter estimators, respectively. Furthermore, the set B is restricted to all directions through two data points or 250 random directions from B if $\binom{n}{2} > 250$. The covariance matrix of the h points with smallest outlyingness r is computed. Its first k eigenvectors span the subspace onto which the data points are then projected. Finally, center and covariance of the projected data is estimated using the RMCD procedure. The eigenvectors of this projected data covariance matrix determine the robust principal components.

In addition to robust computation of the scores and loadings an initial robust centering or autoscaling of the data is often recommended. A robust estimator of the multivariate center with good statistical properties is the L_1*-median* (*spatial median*, *geometric median*). It is a generalization of the univariate median to higher dimensions. For a data set $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of n observation vectors with each $\mathbf{x}_i \in \mathbb{R}^m$, the L_1-median $\hat{\mu}(\mathbf{X})$ is defined as the point for which the sum of the Euclidean distances to these n observations is minimal, i.e.

$$\hat{\boldsymbol{\mu}}(\mathbf{X}) = \arg\min_{\boldsymbol{\mu}} \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}\| \qquad (3.6)$$

where $\|\cdot\|$ denotes the Euclidean norm. As its univariate counterpart, the L_1-median has the best possible breakdown point of 0.5 (see Lopuhaä and Rousseeuw (1991)). There are several algorithms for its computation (see Fritz et al. (2012) for a comparison).

For autoscaling the data one also needs an estimator of scale. A usual choice for highly robust estimation of univariate scale is the median absolute deviation (MAD) with a breakdown point of 0.5. A less robust scale estimator is the interquantile range (IQR) with a breakdown point of 0.25. Alternative robust estimators of scale with high breakdown points can be found in Rousseeuw and Croux (1993).

3.3 Multi-Block PCA

The approach using T^2 and SPE is suitable for detecting faults, process shifts, drifts or other abnormal behaviour in a production process. However, if a fault is detected one is interested in fault diagnosis, i.e., a root cause analysis of the fault and the variables involved in the nature of the fault. As the underlying principal component scores are linear combinations of the original input variables they do not always allow a straightforward interpretation. In this case, a T^2 decomposition may not be suitable for fault diagnosis.

A possible tool for fault diagnosis in PCA-based process monitoring models is the concept of *contribution plots* (see Miller et al. (1998), Westerhuis et al. (2000), Qin (2003)). With contribution plots the contributions of each process variable to the scores of the PCA model can be determined. They can be given for T^2 as well as for the SPE statistic. Even if these plots may not explicitly diagnose the root cause of a fault they can provide additional insight and determine variable entries that are not consistent with the normal operating condition (see Kourti and MacGregor (1995)). Confidence bounds for contribution plots can also be defined (see Conlin et al. (2000), Westerhuis et al. (2000)). A method for determining variable contributions that has a higher rate of correct fault diagnosis for T^2 and SPE indices than traditional contribution plots is the reconstruction-based contribution approach (see Alcala and Qin (2009)).

Another fault diagnosis technique is *multi-block PCA*. It is applied if the contribution of groups or *blocks* of variables instead of single variables is of interest. The idea of multi-block PCA was first introduced by Wold et al. in 1987 (see Wold et al. (1987c)). It is suitable if the process variables can be grouped in meaningful blocks. This is often the case in semiconductor manufacturing processes as groups of variables are often conceptually related or can be assigned to certain parts of a manufacturing equipment. Multi-block PCA then allows monitoring of these variable groups individually in addition to an overall monitoring. This can make faults assignable to specific variable blocks or production equipment parts which is highly desirable for fault diagnosis in semiconductor production.

There are several versions of multi-block PCA (see Westerhuis et al. (1998)). We focus on *consensus* PCA (CPCA). Formally, the data set $\mathbf{X} \in \mathbb{R}^{n \times m}$ is divided into B blocks $\mathbf{X}_1, \ldots, \mathbf{X}_B$ with block b having m_b variables and $m = \sum_{b=1}^{B} m_b$. The procedure seeks a consensus direction among all blocks. In addition to global scores and loadings, called *super scores* and *super loadings*, the procedure also determines block statistics in terms of *block* *scores* and *block loadings*. The super statistics are derived using all variables whereas the block statistics are derived using only variables from the corresponding block. A multi-block adaption of the NIPALS algorithm can be given. However, as can be shown the results from multi-block PCA can be derived from standard PCA models (see Westerhuis et al. (1998), Chen and McAvoy (1998), Qin et al. (2001)). Furthermore, it can be shown that the super scores of CPCA are identical with the scores of regular PCA (see Qin et al. (2001)). Following Qin et al. (2001), a version of the CPCA algorithm based on the PCA scores is given as follows.

CPCA based on PCA scores
Let $\mathbf{X} = [\mathbf{X}_1 \cdots \mathbf{X}_B]$ be divided into B blocks.

1. Perform regular PCA on $\mathbf{X}$ to obtain the scores $\mathbf{t}_1, \ldots, \mathbf{t}_R$.
2. Set $\mathbf{E}_{b,1} := \mathbf{X}_b$ for $b = 1, \ldots, B$
3. For PC r compute

$$\begin{aligned} \text{block loadings:} \quad \mathbf{p}_{b,r} &= \frac{\mathbf{E}_{b,r}^T \mathbf{t}_r}{\|\mathbf{E}_{b,r}^T \mathbf{t}_r\|}, \\ \text{block scores:} \quad \mathbf{t}_{b,r} &= \mathbf{E}_{b,r}\mathbf{p}_{b,r} \quad \text{and} \quad \mathbf{T}_r = [\mathbf{t}_{1,r} \cdots \mathbf{t}_{B,r}], \\ \text{super loadings:} \quad \mathbf{p}_{T,r} &= \frac{\mathbf{T}_r^T \mathbf{t}_r}{\|\mathbf{T}_r^T \mathbf{t}_r\|}. \end{aligned}$$

4. Deflate residuals by

$$\mathbf{E}_{b,r+1} = \left(\mathbf{I} - \frac{\mathbf{t}_r \mathbf{t}_r^T}{\mathbf{t}_r^T \mathbf{t}_r}\right) \mathbf{E}_{b,r}.$$

   and go to step 3 to compute the next PC.

3.3.1 Block Statistics

In order to apply the multi-block PCA approach for process monitoring and fault diagnosis the super statistics as well as the block statistics have to be monitored. Faults are detected using super T^2 and super SPE statistics. Then fault diagnosis is achieved via block T^2 and block SPE charts.

As super scores of a multi-bock PCA model coincide with the scores from regular PCA, the super T^2 statistic is the same as the T^2 of regular PCA scores given in equation (3.2). The super SPE statistic also coincides with the regular SPE statistic from equation (3.3).

Followig Qin et al. (2001), the block T^2 statistics are given by

$$T_b^2 = \mathbf{t}_b^T \mathbf{\Lambda}_b^{-1} \mathbf{t}_b \tag{3.7}$$

where $\mathbf{\Lambda}_b$ is the covariance matrix of the block scores $\mathbf{t}_b$ for $b = 1, \ldots, B$ variable blocks and R PCs. The block scores $\mathbf{t}_b$ for a given block b can be correlated. Thus, $\mathbf{\Lambda}_b$ is not necessarily a diagonal matrix and can even become singular. In such cases, a pseudoinverse $\mathbf{\Lambda}_b^+$ of $\mathbf{\Lambda}_b$ should be used, i.e.,

$$T_b^2 = \mathbf{t}_b^T \mathbf{\Lambda}_b^+ \mathbf{t}_b.$$

The block T^2 statistic can be drilled down even further and variable contributions can be determined. This connects block statistics to the aforementioned contribution plots. Rewrite the super T^2 statistic as

$$T^2 = \mathbf{x}^T \mathbf{P} \mathbf{\Lambda}^{-1} \mathbf{P}^T \mathbf{x} = \|\mathbf{\Lambda}^{-1/2} \mathbf{P}^T \mathbf{x}\|^2 = \left\| \sum_{b=1}^{B} \mathbf{\Lambda}^{-1/2} \mathbf{P}_b^T \mathbf{x}_b \right\|^2$$

where $\mathbf{P}_b$ is the set of m_b rows of the loading matrix $\mathbf{P} \in \mathbb{R}^{m \times R}$ corresponding to block b. Then the block T^2 can be written as

$$T_b^2 = \|\mathbf{\Lambda}^{-1/2} \mathbf{P}_b^T \mathbf{x}_b\|^2 = \left\| \sum_{l=1}^{m_b} \mathbf{\Lambda}^{-1/2} \mathbf{P}_{b,l}^T \mathbf{x}_{b,l} \right\|^2$$

and the contribution of the lth variable in the bth block can be determined by

$$T_{b,l}^2 = \|\mathbf{\Lambda}^{-1/2} \mathbf{P}_{b,l}^T \mathbf{x}_{b,l}\|^2.$$

This leads to contribution plots as defined in Miller et al. (1998) (see Qin et al. (2001), section 4.2).

In an analogous manner the block SPE statistic for variable block b is given by

$$SPE_b = \|\mathbf{x}_b - \hat{\mathbf{x}}_b\|^2 \tag{3.8}$$

and the contribution of the lth variable to SPE_b is given by

$$\text{SPE}_{b,l} = (\mathbf{x}_{b,l} - \hat{\mathbf{x}}_{b,l})^2.$$

Furthermore,

$$\text{SPE}_b = \sum_{l=1}^{m_b} \text{SPE}_{b,l}.$$

These variable contribution statistics can be used to further localize process variables most affected by a fault.

Control limits for the block statistics in case of multivariate normality can be derived from the limits for regular statistics in a straightforward way and are given in Qin et al. (2001). In the nonnormal case the nonparametric bootstrapping approach can be used.

3.4 Multi-Way PCA

Up to now we have considered data matrices of the form $\mathbf{X} \in \mathbb{R}^{I \times J}$ for process monitoring tasks. Such data matrices consisting of observations in rows and variables in columns define two-dimensional or two-way arrays. However, in some applications it is necessary to consider *three-way* or, more generally, *multi-way* arrays that define higher dimensional data structures. A three-way array can occur if, for each observation, each variable is measured at multiple time points. This measuring scheme leads to a three dimensional array consisting of observations, variables and time points. Graphically this can be represented as data cube as illustrated in figure 3.2.

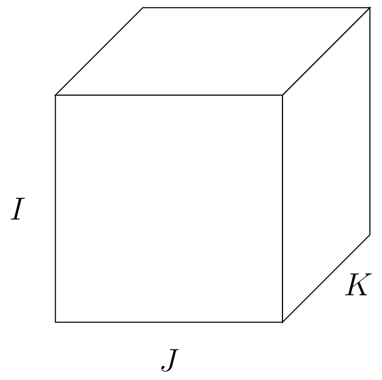


Figure 3.2: Illustration of a three-way data array as cube consisting of I observations (batches) in J variables measured at K measurement points, e.g. measurement sites or time points.

Among many applications in chemometrics and physical production processes, three-way arrays are the result of several crucial processing steps in semiconductor manufacturing. For example, for a wafer that takes $K = 200$ seconds to process $J = 10$ process variables are measured each second. By considering a data set of $I = 100$ wafers this leads to a three-way array with 100 samples in 10 variables and 200 time points. The third dimension can also be the result of measuring a wafer at multiple measurement sites. Processes like these are often referred to as *batch processes* and the production units, e.g. wafers, are referred to as *batches*.

Thus, in order to analyze all of the recorded data information methods are needed that are able to handle three-way arrays. Overviews of multi-way analysis techniques can be found in Geladi (1989), Smilde et al. (2004) or Acar and Yener (2009).

One possible multi-way analysis method is the *Tucker3* model proposed by Tucker in 1966 for psychometrical applications (see Tucker (1966)). Here, the multi-way array is decomposed into sets of scores and loadings. For a three-way array $\mathbf{X}$ with dimension $I \times J \times K$ consider the element $x_{ijk} \in \mathbf{X}$. The corresponding Tucker3 model is defined as

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} a_{ip} b_{jq} c_{kr} g_{pqr} + e_{ijk}$$

where a_{ip}, b_{jq} and c_{kr} are elements of the loading matrices (or *modes*) $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ respectively, g_{pqr} is an element of the core array $\mathbf{G} \in \mathbb{R}^{P \times Q \times R}$ and e_{ijk} is an element of the three-way error array $\mathbf{E} \in \mathbb{R}^{I \times J \times K}$. The arrays $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{G}$ can be computed by minimizing the sum of squared errors and the matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ can be determined via cross validation. Analogously, a *Tucker2* model can be defined where only two modes are treated and a *Tucker1* model treats only one mode. The Tucker1 method corresponds to simply rearranging the three-way array as a matrix and decomposing the unfolded data via classical PCA (see also Varmuza and Filzmoser (2009), section 3.8.5). For Tucker3, also a robust version is available (see Pravdova et al. (2001)).

Another approach for handling multi-way arrays is parallel factor analysis (PARAFAC) as introduced by Harshman in 1970 (see Harshman (1970)). For a three-way array it can be expressed as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + e_{ijk}$$

where R is the number of components to extract, a_{ir}, b_{jr} and c_{kr} are elements of the component matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ and e_{ijk} is again an element of the residual array $\mathbf{E} \in \mathbb{R}^{I \times J \times K}$. It can be shown that a PARAFAC model can be considered a constraint variant of a Tucker3 model (see Kiers (1991)). Details on PARAFAC and its use in chemometrics can also be found in a tutorial by Bro (1997). A robust version of PARAFAC has also been developed (see Engelen and Hubert (2011)).

A popular method for three-way analysis is to simply unfold the three-way array to a two-way matrix and analyse the unfolded matrix using PCA. This was already referred to as Tucker1 method. Although the approach does not consider the three-way nature of the data, as opposed to Tucker3 or PARAFAC, it is especially popular for process control applications. This is due to its simplicity and the fact that classical and well-known ideas based on PCA can be applied. The approach is also referred to as *multi-way PCA* as it provides multi-way solutions of PCA results (see Wold et al. (1987b)). Thus, multi-way PCA is algorithmically consistent with classical PCA.

Nomikos and MacGregor were among the first to apply multi-way PCA for the task of monitoring industrial batch processes (see Nomikos and MacGregor (1994), Nomikos and MacGregor (1995)). The simple but powerful approach led to many other discussions and industrial applications (see e.g. Kourti et al. (1995), Chen and McAvoy (1998), Dahl et al. (1999), Louwerse and Smilde (2000), Lennox et al. (2001)) and especially in semiconductor manufacturing where many forms of process data are naturally organized in three dimensions (see e.g. Wise et al. (1999), Yue et al. (2000), Qin et al. (2006), Cherry and Qin (2006)).

In the following subsections we discuss fault detection and diagnosis of three-way data processes in all data dimensions of the underlying three-way array. Batch level fault detection allows the identification of faulty batches (sample dimension). A multi-block approach makes fault diagnosis possible and allows one to identify variables or groups of variables that show the fault (variable dimension). Finally, on-line monitoring enables an identification of the time point of fault occurrence or when the process starts to drift off from its normal operation (time dimension).

3.4.1 Batch Level Fault Detection

At first the multi-way principle component subspace has to be determined. For a given three-way array $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ there are naturally three possibilities to unfold it to a two-way matrix. The most common technique is to unfold $\mathbf{X}$ by the time axis, i.e., put each matrix slice of dimension $I \times J$ side by side to the right. The result is a two-way matrix $\mathbf{X}^*$ of dimension $I \times JK$, thus each row represents one batch. This unfolding allows an analysis of $\mathbf{X}$ at batch level where the given data information is summarized with respect to variables and their time variation (see Nomikos and MacGregor (1994), p. 1364). An illustration of this unfolding technique is given in figure 3.3.

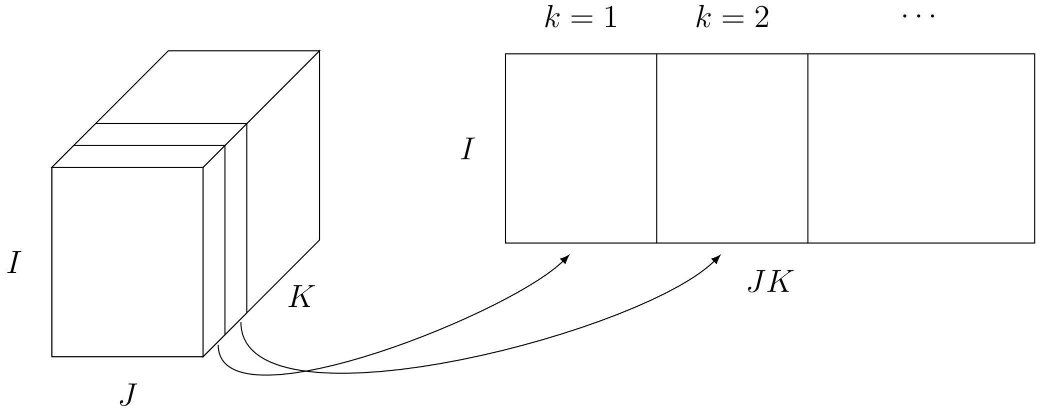


Figure 3.3: The unfolding of a three-way data array of dimension $I \times J \times K$ to a matrix of dimension $I \times JK$. For $k = 1, \ldots, K$ the $k+1$th matrix is positioned right of the kth matrix.

The goal of multi-way PCA is to decompose a three-way array $\mathbf{X} \in \mathbb{R}^{I\times J\times K}$ into a series of score vectors $\mathbf{t} \in \mathbb{R}^I$ and loading matrices $\mathbf{P} \in \mathbb{R}^{K\times J}$ and a three-way residual array $\mathbf{E}$ such that

$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{t}_r \otimes \mathbf{P}_r + \mathbf{E}$$

and R PCs. The vector-matrix multiplication is defined by

$$\hat{\mathbf{X}}[i,j,k] = \sum_{r=1}^{R} \mathbf{t}_r \otimes \mathbf{P}_r := \sum_{r=1}^{R} \mathbf{t}_r[i]\mathbf{P}_r[k,j].$$

A graphical depiction of this decomposition is given in figure 3.4.

A NIPALS version for three-way arrays can also be given (see Nomikos and MacGregor (1994)) and is formulated as follows.

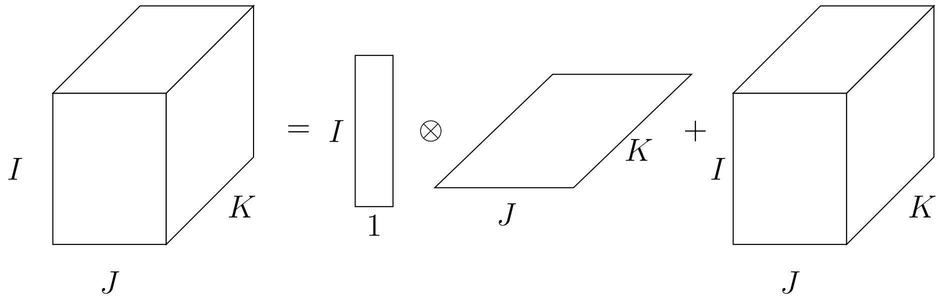


Figure 3.4: Decomposition of a three-way array by multi-way PCA.

NIPALS for three-way arrays
Let $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ be a three-way array.

1. Autoscale each column of $\mathbf{X}$.
2. Start with an arbitrary column of $\mathbf{X}$ as initial score vector $\mathbf{t}$.
   Set $\mathbf{E} = \mathbf{X}$.
3. $\mathbf{P} = \mathbf{E}^T \cdot \mathbf{t}$.
4. $\mathbf{P} = \mathbf{P}/\|\mathbf{P}\|$.
5. $\mathbf{t} = \mathbf{E} \odot \mathbf{P}$.
6. If $\mathbf{t}$ has converged with a predefined precision a principal component has been computed; go to step 7. Otherwise iterate steps 3 to 5 until convergence.
7. Update $\mathbf{E} = \mathbf{E} - \mathbf{t} \otimes \mathbf{P}$ and continue with step 3 for the next PC.

As in classical two-way PCA the result is a separation into a systematic part and a residual noise part. The systematic part consists of scores related to the batches (e.g. wafers) and loading matrices related to variables and time points. Thus, as in classical two-way PCA, the score vectors contain one value per observation (batch) per component and can be considered as coordinates in the multi-way principal component subspace.

Multivariate statistical process control techniques from the two-way case can be applied in a natural way to multi-way PCA results for three-way arrays. New batches can be monitored via their scores and residuals. Let

$\mathbf{x}^{\text{test}} \in \mathbb{R}^{i \times J \times K}$ be a new batch observation that is autoscaled using in-control estimators, then

$$\begin{aligned}\mathbf{t}_r^{\text{test}} &= \mathbf{x}^{\text{test}} \odot \mathbf{P}_r \\ \mathbf{e}^{\text{test}} &= \mathbf{x}^{\text{test}} - \sum_{r=1}^{R} \mathbf{t}_r^{\text{test}} \otimes \mathbf{P}_r.\end{aligned}$$

Due to batch level unfolding, faulty batches can be determined. T^2 monitoring of the scores is the same as in the two-way case. The SPE statistic is the sum of squares of the residuals for a batch. For $\mathbf{x}^{\text{test}}$ it is given by

$$\text{SPE}_i = \text{SPE}_{\mathbf{x}^{\text{test}}} = \sum_{j=1}^{J} \sum_{k=1}^{K} \mathbf{e}^{\text{test}}[i, j, k]^2.$$

The value SPE_i represents the squared perpendicular distance of $\mathbf{x}^{\text{test}}$ from the multi-way principal component subspace defined by the normal operating condition multi-way PCA model.

3.4.2 Fault Diagnosis

In order to identify the root cause of a fault, multi-block PCA can be applied to the unfolded three-way array to determine variable block contributions. For industrial processing, the grouping should be determined using process engineer knowledge in order to be conceptually meaningful and interpretable. Figure 3.5 depicts the unfolding procedure for multi-way multi-block PCA in a schematic way.

The blocking in figure 3.4.2 is depicted as solely by variable. This allows a drill-down root cause analysis to determine variable blocks or equipment parts that are connected with a fault. However, more sophisticated blocking schemes such as blocking by variable and measurement sites (see Qin et al. (2006)) or variables and time phases can also be applied. This way, time points could be summarized according to different processing steps. For example, for a wafer with a processing time of 200 seconds, the first and last 10 seconds could be warm-up and cool-down phase, respectively. Such phases often show special behaviour and, if they are of interest for the overall process result, could be treated separately. The remaining 180 seconds could be grouped according to different processing steps given by the process recipe.

Block contributions can then be monitored via the block T^2 statistic as given in equation (3.7) and the block SPE statistic given in equation (3.8).

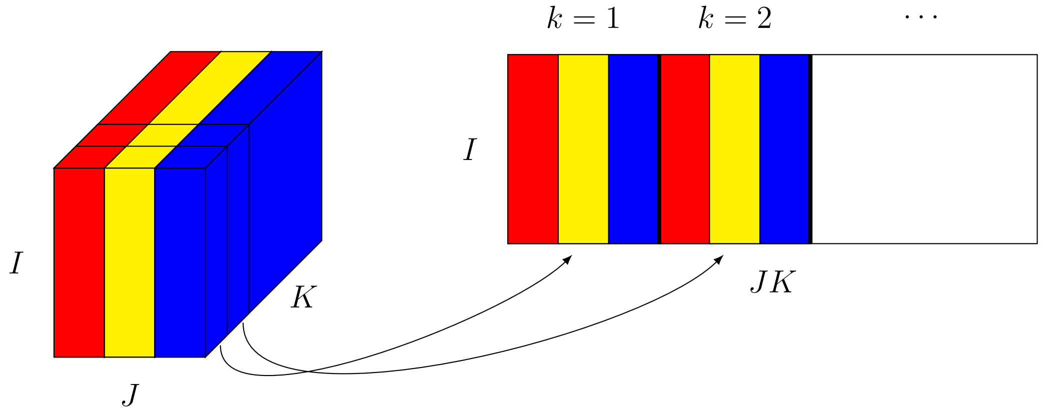


Figure 3.5: Schematic illustration of the unfolding of a $I \times J \times K$-three-way data array to a $I \times JK$-matrix if variables are summarized to blocks colored red, yellow and blue.

3.4.3 On-line Monitoring

The computation of scores and residuals for new batches $x^{\text{test}} \in \mathbb{R}^{1\times J\times K}$ is only possible if x^{test} is complete, i.e., if it has J-dimensional observations in all K time points. However, for batch processing one is interested in monitoring the new batch on-line during its processing, i.e., in time points $k < K$. If the process is in its kth time interval only $k < K$ rows of x^{test} are complete and all future observations from time points $k+1, \ldots, K$ are missing, i.e., yet to be measured. Figure 3.6 illustrates this problem for the data matrix of a single unfinished batch.

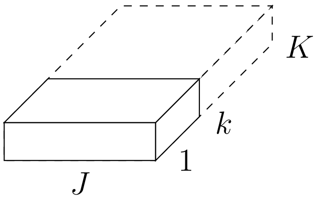


Figure 3.6: Illustration of the $1 \times J \times K$ dimensional data matrix generated by a single unfinished batch with J variables measured at K time points. From time point $k < K$ onwards the matrix is not complete as future observations are not yet measured.

To overcome this problem future observations have to be filled up in order to create a complete batch that is consistent with the multi-way PCA model structure. In Nomikos and MacGregor (1995) three approaches for fill-up strategies were proposed.

(a) Fill up the remaining slots with zeros.

(b) Fill up the remaining slots with the current value.

(c) Use the ability of PCA to handle missing values.

The approach to use depends on the characteristics of the process under consideration. However, the most common approach is (b), i.e., to assume that future deviations from the mean trajectory will remain constant at the value observed at time k for the remaining batch run (see Nomikos and MacGregor (1994)). For PARAFAC models an alternative approach can be given (see Meng et al. (2003)).

Applying one of the proposed methods results in complete batch measurements for each time point k with $k = 1, \ldots, K$ of in-production batches. This allows on-line monitoring of the production with T^2 and SPE monitoring statistics, both globally (super score level) and locally (block level), for each recorded data point during the processing of a batch. When implemented on-line in a production environment, the method can provide detailed real-time information on the process condition. Thus, faults can be detected and diagnosed as they occur during batch processing and, if possible, corrective measures can be applied immediately.

Chapter 4

Nonlinear Process Control

The data analysis and process monitoring approach based on PCA assumes that the underlying data set has linear characteristics as it linearly maps multivariate data onto lower dimensions. However, for many applications in industrial processes this assumption is not always valid. For example, plasma-based processes as used in semiconductor manufacturing often show nonlinear behaviour and nonlinear relationships among the process variables. Here, PCA often performs poorly and fails to sufficiently capture the data characteristics. It can even yield misleading results (e.g., see Palus and Dvorak (1992)). Due to this drawback, several nonlinear approaches to PCA have been proposed in the literature, e.g. in Kramer (1991), Dong and McAvoy (1996), Hiden et al. (1999) or Jia et al. (2001). However, the most popular technique for nonlinear PCA is *kernel PCA* (KPCA) as proposed in Schölkopf et al. (1998). The main advantage of kernel PCA is its simplicity as it merely involves linear algebra. As in the linear case, process monitoring based on KPCA relies on the T^2 and SPE statistics. However, the approach allows superior fault detection compared to classical PCA if nonlinearities are present. This makes it appropriate for monitoring complex industrial production processes with nonlinear characteristics.

This chapter reviews KPCA and basic nonlinear process monitoring. Recent approaches on robustifications as well as fault diagnosis for KPCA monitoring models are also discussed.

4.1 Kernel PCA

The basic idea of kernel PCA is to first map the original input data to a high-dimensional feature space using a nonlinear mapping function. In this feature space the variables are assumed to vary linearly and the PCs are extracted. Compared to other nonlinear PCA approaches KPCA is simple as it does not require a nonlinear optimization problem to be solved but simply an eigenvalue problem. Furthermore, the possibility to use several different nonlinear transformations of the inputs allows the modeling of a wide range of nonlinearities.

Formally, a given data set of n mean-centered observations in m variables, $\mathbf{x}_i \in \mathbb{R}^m$ with $i = 1, \ldots, n$ is first mapped into a feature space F via the nonlinear mapping function

$$\Phi : \mathbb{R}^m \rightarrow F.$$

In this space the mapped data is again assumed to be mean-centered, i.e., $\sum_{i=1}^n \Phi(\mathbf{x}_i) = \mathbf{0}$. Centering the data in the feature space will be discussed later. Using the covariance matrix $\mathbf{C}$ in F, i.e.,

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T,$$

the principal components are determined by solving the eigenvalue problem

$$\lambda \mathbf{v} = \mathbf{C}\mathbf{v}, \tag{4.1}$$

with eigenvalues $\lambda \geq 0$ and eigenvectors $\mathbf{v} \in F \backslash \{0\}$. All solutions of $\mathbf{v}$ with $\lambda \neq 0$ lie in $\text{span}(\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n))$, i.e., there exist coefficients $\alpha_1, \ldots, \alpha_n$ such that

$$\mathbf{v} = \sum_{j=1}^{n} \alpha_j \Phi(\mathbf{x}_j). \tag{4.2}$$

Furthermore, the eigenvalue problem (4.1) can be written as

$$\lambda \langle \Phi(\mathbf{x}_i), v \rangle = \langle \Phi(\mathbf{x}_i), \mathbf{C}\mathbf{v} \rangle, \tag{4.3}$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ is the dot product between vectors $\mathbf{x}$ and $\mathbf{y}$. Using

$$\mathbf{C}\mathbf{v} = \left(\frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T \right) v = \frac{1}{n} \sum_{i=1}^{n} \langle \Phi(\mathbf{x}_i), v \rangle \Phi(\mathbf{x}_i)$$

and combining (4.2) and (4.3) leads to

$$\lambda \sum_{j=1}^{n} \alpha_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \frac{1}{n} \sum_{j=1}^{n} \alpha_j \langle \Phi(\mathbf{x}_i), \sum_{k=1}^{n} \Phi(\mathbf{x}_k) \rangle \langle \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) \rangle \tag{4.4}$$

for all $i = 1, \dots, n$. Then, defining an $n \times n$ matrix $\mathbf{K}$ with

$$\mathbf{K}_{ij} := \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \tag{4.5}$$

simplifies equation (4.4) to

$$n\lambda \mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^2 \boldsymbol{\alpha} \tag{4.6}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$. In order to find solutions of equation (4.6) it is sufficient to find solutions for the eigenvalue problem

$$n\lambda \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{4.7}$$

for nonzero eigenvalues (as proven in Schölkopf et al. (1998), appendix A). Thus, performing PCA in the feature space F is equivalent to finding solutions to the eigenvalue problem (4.7). This gives eigenvectors $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n$ to the corresponding eigenvalues $\lambda_1 \geq \dots \geq \lambda_n$. Thus, in KPCA up to n PCs can be extracted. The problem of dimensionality can be reduced by retaining only the first R eigenvectors. The remaining eigenvectors $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_R$ are normalized. By requiring

$$\langle \mathbf{v}_r, \mathbf{v}_r \rangle = 1$$

for $r = 1, \dots, R$ and using (4.2) this leads to

$$\lambda_{F,r} \langle \boldsymbol{\alpha}_r, \boldsymbol{\alpha}_r \rangle = 1.$$

The principal component scores t_r for PC r of an observation vector $\mathbf{x}$ are extracted by computing its projections onto the vectors $\mathbf{v}_r$, i.e.,

$$t_r = \langle \mathbf{v}_r, \Phi(\mathbf{x}) \rangle = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^r \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle. \tag{4.8}$$

The mapping Φ can be an arbitrary nonlinear mapping into the possibly high-dimensional feature space F. The direct computation of this mapping can be computationally intense or is simply not possible. However, the mapping Φ is never needed explicitly except in dot products, see equations (4.5) and (4.8). With this, one can avoid the direct computation of Φ by introducing a *kernel function* of the form

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle. \tag{4.9}$$

This allows the computation of dot products of the nonlinear mapping without the explicit knowledge of Φ itself. This is computationally cheaper than performing the mapping on x and y, respectively. The idea is often referred to as *kernel trick*. In general, this provides simple nonlinear generalizations for algorithms that can be expressed solely in terms of dot products. Thus, based on the kernel trick several other kernel methods have been introduced for both supervised and unsupervised nonlinear analysis and modeling. For an extensive overview on these methods see, e.g., Schölkopf and Smola (2002).

A number of different kernel functions exist. The requirement on a kernel function is to satisfy Mercer's theorem (see Mercer (1909)). It states that if the kernel function k is a continuous kernel of a positive integral operator then there exists a mapping Φ into a space where equation (4.9) can be applied (see Christianini and Shawe-Taylor (2000)). Examples for valid kernel functions are the following:

- *Linear kernel*:
  $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$,
- *Polynomial kernel*:
  $k(\mathbf{x}, \mathbf{y}) = (a\langle \mathbf{x}, \mathbf{y} \rangle + b)^d$,
- *Radial basis functions (RBF, Gaussian kernel)*:
  $k(\mathbf{x}, \mathbf{y}) = \exp(-\sigma \|\mathbf{x} - \mathbf{y}\|^2)$,

where the parameters a, b, d and σ have to be specified in advance. Substituting all occurrences of $\langle \Phi(x), \Phi(y) \rangle$ with the chosen k allows the straightforward application of KPCA to a given data set. The choice of the function k then determines the mapping Φ and the feature space F. While the linear kernel does not induce any nonlinearity, the polynomial kernel as well as the radial basis functions are considered standard nonlinear kernels.

However, the choice of a kernel function that is most appropriate for a given data problem strongly depends on the system under study. Furthermore, there are several techniques on how to determine the number of principal components to extract in a kernel PCA model, e.g., cross-validation (e.g., see Alam and Fukumizu (2014)) or the average eigenvalue method (e.g., see Lee et al. (2004a)). For the Gaussian kernel, kernel parallel analysis (KPA) has been proposed to determine both the value of σ and the number of principal components as a function of σ (see Jorgensen and Hansen (2012)). It does so by comparing the eigenvalues for each KPCA component to the distribution of eigenvalues of data sets obtained by permuting the original input data several times. Such data sets are often referred to as null data sets.

However, in case of the Gaussian kernel, for a given system under study, testing the monitoring performance of a model for various values of σ can lead to satisfying results (see Lee et al. (2004a)). Then, KPA based on the determined σ can be used as a guideline for the number of components to extract due to its simplicity.

4.1.1 Mean Centering in the Feature Space

We have assumed that the observations $\mathbf{x}_i \in \mathbb{R}^m$ with $i = 1, \ldots, n$ are mean centered in the input space, i.e., $\sum_{i=1}^n \mathbf{x}_i = 0$, as well as in the feature space F, i.e., $\sum_{i=1}^n \Phi(\mathbf{x}_i) = 0$. In the feature space this is not as simple as we can not explicitly compute the mean of $\Phi(\mathbf{x}_i)$ for $i = 1, \ldots, n$. However, mean centering in the feature space can be expressed via the noncentered kernel matrix K (see Schölkopf et al. (1998), appendix B). Formally, for a given nonlinear mapping Φ centering is performed via

$$\tilde{\Phi}(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \frac{1}{n}\sum_{i=1}^n \Phi(\mathbf{x}_i).$$

Computing a centered version of K leads to

$$\begin{aligned}
\tilde{\mathbf{K}}_{ij} &= \left\langle \Phi(\mathbf{x}_i) - \frac{1}{n}\sum_{k=1}^n \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) - \frac{1}{n}\sum_{l=1}^n \Phi(\mathbf{x}_l) \right\rangle \\
&= \mathbf{K}_{ij} - \frac{1}{n}\sum_{k=1}^n \mathbf{1}_{ik}\mathbf{K}_{kj} - \frac{1}{n}\sum_{l=1}^n \mathbf{K}_{il}\mathbf{1}_{lj} + \frac{1}{n^2}\sum_{k=1}^n\sum_{l=1}^n \mathbf{1}_{ik}\mathbf{K}_{kl}\mathbf{1}_{lj} \\
&= (\mathbf{K} - \mathbf{1}_n\mathbf{K} - \mathbf{K}\mathbf{1}_n + \mathbf{1}_n\mathbf{K}\mathbf{1}_n)_{ij}
\end{aligned}$$

where $\mathbf{1}_{ij} := 1$ and $(\mathbf{1}_n)_{ij} := \frac{1}{n}$ for all $i, j = 1, \ldots, n$. Thus, the centered kernel matrix $\tilde{\mathbf{K}}$ can be determined directly from the noncentered original kernel matrix $\mathbf{K}$. The eigenvalue problem (4.7) can then be solved using the centered version $\tilde{\mathbf{K}}$.

New test data points can be centered using training data estimates. Consider n_{test} test observation vectors $\mathbf{y}_1, \ldots, \mathbf{y}_{n_{\text{test}}}$. The corresponding kernel matrix is given by

$$\mathbf{K}^{\text{test}}_{lj} = \langle \Phi(\mathbf{y}_l), \Phi(\mathbf{x}_j) \rangle$$

for $l = 1, \ldots, n_{\text{test}}$ and $j = 1, \ldots, n$, thus $\mathbf{K}^{\text{test}} \in \mathbb{R}^{n_{\text{test}} \times n}$. The corresponding mean centered version $\tilde{\mathbf{K}}^{\text{test}}$ can again be determined in terms of $\mathbf{K}^{\text{test}}$ in an analogous manner. This gives

$$\tilde{\mathbf{K}}^{\text{test}} = \mathbf{K}^{\text{test}} - \mathbf{1}_n^{'}\mathbf{K} - \mathbf{K}^{\text{test}}\mathbf{1}_n + \mathbf{1}_n^{'}\mathbf{K}\mathbf{1}_n$$

where $\mathbf{1}_n'$ is a $n_{\text{test}} \times n$ matrix with all entries equal to $1/n$.

4.1.2 Process Control Using KPCA

The ability to capture the nonlinear structure in data makes kernel PCA suitable for monitoring of complex industrial processes. The process control method based on KPCA is similar to the classical PCA-based approach and utilizes the T^2 statistic and the SPE in the feature space (see Lee et al. (2004a)).

A new data sample $\mathbf{x}^{\text{new}} \in \mathbb{R}^m$ is first mean centered or autoscaled in the original input space using training data estimates. Then the kernel vector $\mathbf{k}^{\text{test}} \in \mathbb{R}^n$ is computed via

$$\mathbf{k}_i^{\text{test}} = \langle \Phi(\mathbf{x}^{\text{test}}, \Phi(\mathbf{x}_i)) \rangle = k(\mathbf{x}^{\text{test}}, \mathbf{x}_i)$$

for $\mathbf{x}_i \in \mathbb{R}^m$ for $i = 1, \dots, n$ and a kernel function k. Then the mean centered test kernel vector $\tilde{\mathbf{k}}^{\text{test}}$ is computed as described above. The corresponding test score vector $\mathbf{t}^{\text{test}} \in \mathbb{R}^R$ for R PCs is determined via

$$\mathbf{t}_r^{\text{test}} = \langle \mathbf{v}_r, \tilde{\Phi}(\mathbf{x}^{\text{test}}) \rangle = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^r \langle \tilde{\Phi}(\mathbf{x}_i), \tilde{\Phi}(\mathbf{x}^{\text{test}}) \rangle = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^r \tilde{\mathbf{k}}_i^{\text{test}} \qquad (4.10)$$

for $r = 1, \dots, R$ and where $\tilde{\Phi}$ denotes the mean centered version of the mapping Φ. The T^2 statistic of the test scores is determined analogously to the linear case in equation (3.2), i.e.,

$$T^2 = (\mathbf{t}^{\text{test}})^T \boldsymbol{\Lambda}^{-1} \mathbf{t}^{\text{test}}$$

where $\boldsymbol{\Lambda}_R \in \mathbb{R}^{R \times R}$ denotes the estimated covariance matrix of scores from a KPCA model with R PCs of in-control data. If the scores follow a multivariate normal distribution the limit can be determined accordingly, otherwise nonparametric bootstrapping using the training data T^2 statistic can be used to determine the desired α quantile.

The squared prediction error can be applied to measure the goodness of fit of a sample to the KPCA model. It can be computed in the feature space F in a straightforward way (see Lee et al. (2004a)). Let $\hat{\Phi}_R(\mathbf{x}) = \sum_{r=1}^{R} \mathbf{t}_r \mathbf{v}_r$ be the reconstructed value of the nonlinear mapping $\Phi(\mathbf{x})$ using KPCA scores $\mathbf{t}_r$ and loadings $\mathbf{v}_r$ for $r = 1, \dots, R$ principal components. Clearly, for $R = n$ we have $\hat{\Phi}_R(\mathbf{x}) = \Phi(\mathbf{x})$. Then the SPE in the feature space is defined as

$$\text{SPE}_R = \|\Phi(\mathbf{x}) - \hat{\Phi}_R(\mathbf{x})\|^2. \qquad (4.11)$$

Using $\mathbf{v}_i^T\mathbf{v}_j = 1$ for $i = j$ and $\mathbf{v}_i^T\mathbf{v}_j = 0$ for $i \neq j$, the statistic can be computed via

$$\begin{aligned}
\mathrm{SPE}_R &= \|\Phi(\mathbf{x}) - \hat{\Phi}_R(\mathbf{x})\|^2 = \|\hat{\Phi}_n(\mathbf{x}) - \hat{\Phi}_R(\mathbf{x})\|^2 \\
&= \hat{\Phi}_n(\mathbf{x})^T\hat{\Phi}_n(\mathbf{x}) - 2\hat{\Phi}_n(\mathbf{x})^T\hat{\Phi}_R(\mathbf{x}) + \hat{\Phi}_R(\mathbf{x})^T\hat{\Phi}_R(\mathbf{x}) \\
&= \sum_{i=1}^{n} t_i\mathbf{v}_i^T \sum_{j=1}^{n} t_j\mathbf{v}_j - 2\sum_{i=1}^{n} t_i\mathbf{v}_i^T \sum_{j=1}^{R} t_j\mathbf{v}_j + \sum_{i=1}^{R} t_i\mathbf{v}_i^T \sum_{j=1}^{R} t_j\mathbf{v}_j \\
&= \sum_{i=1}^{n} t_i^2 - 2\sum_{i=1}^{R} t_i^2 + \sum_{i=1}^{R} t_i^2 \\
&= \sum_{i=1}^{n} t_i^2 - \sum_{i=1}^{R} t_i^2.
\end{aligned}$$

The control limit for the SPE statistic can be obtained similar to the classical PCA case, i.e., by using for example equation (3.4) if the errors follow a normal distribution or nonparametric bootstrapping otherwise.

4.2 Multi-Block Kernel PCA

If kernel PCA is applied to nonlinear process control tasks fault diagnosis is desirable. However, in a kernel context fault diagnosis is even more difficult than in the linear PCA case as the nonlinear mapping function from the original input space to the feature space is generally unknown.

As for linear PCA, fault diagnosis can be enabled by a multi-block approach. This leads to multi-block KPCA as recently proposed in Zhang et al. (2010). Although limited to a certain type of kernels the approach provides a kernel generalization of the CPCA procedure based on the NIPALS algorithm.

We assume that for a given data set $\mathbf{X}$ of m-dimensional observation vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ the m variables can be divided into B variable blocks where block b has m_b variables and $m = \sum_{b=1}^{B} m_b$. This yields the grouping scheme $\mathbf{X}_b = \mathbf{x}_{1,b}, \ldots, \mathbf{x}_{n,b}$ for $b = 1, \ldots, B$. In multi-block KPCA the kernel matrix is computed separately for each block. This leads to block kernel matrices $\mathbf{K}_{i,j}^b$. This is only possible when the overall kernel matrix can be decomposed such that

$$\mathbf{K}_{i,j} = \prod_{b=1}^{B} \mathbf{K}_{i,j}^b. \tag{4.12}$$

The Gaussian kernel fulfills the property as for the 2-norm it holds

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{b=1}^{B} \|\mathbf{x}_{i,b} - \mathbf{x}_{j,b}\|^2$$

which leads to

$$\mathbf{K}_{i,j} = \prod_{b=1}^{B} \exp\left\{ \frac{-\|\mathbf{x}_{i,b} - \mathbf{x}_{j,b}\|^2}{c} \right\} = \prod_{b=1}^{B} \mathbf{K}_{i,j}^{b}.$$

However, property (4.12) is not fulfilled for all kernels. For example, for the polynomial kernel with $d > 1$ it generally holds

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle^d \neq \sum_{b=1}^{B} \langle \mathbf{x}_{i,b}, \mathbf{x}_{j,b} \rangle^d.$$

Thus, the polynomial kernel with $d > 1$ does not allow a multi-block KPCA analysis as discussed here.

For valid kernels, the multi-block KPCA algorithm to sequentially extract R PCs derives as follows (see Zhang et al. (2010)). Substituting the original data $\mathbf{X}$ with its mapped version $\Phi(\mathbf{X})$ in the original CPCA algorithm formally yields the kernelized CPCA version. However, in order to allow the actual computation of the block statistics the approach has to be modified to be based on the use of the kernel matrix $\mathbf{K} = \langle \Phi(\mathbf{X}), \Phi(\mathbf{X}) \rangle = \Phi(\mathbf{X})^T \Phi(\mathbf{X})$ instead of the mapped data $\Phi(\mathbf{X})$. This results in the following procedure.

Multi-block kernel PCA

1. Center each block kernel matrix K_b, i.e.,

   $$\overline{\mathbf{K}}_b = \mathbf{K}_b - \mathbf{1}_n \mathbf{K}_b - \mathbf{K}_b \mathbf{1}_n + \mathbf{1}_n \mathbf{K}_b \mathbf{1}_n$$

   where $\mathbf{1}_n$ is an $n \times n$ matrix with all entries equal to $1/n$.

2. Initialize the super score $\mathbf{t}_{T,r}$. For the first PC, i.e., $r = 1$, choose $\mathbf{t}_{T,1}$ as an arbitrary column from an arbitrary block kernel matrix and set $\mathbf{K}_{b,1} := \mathbf{K}_b$.

3. For each block compute the block scores

   $$\mathbf{t}_{b,r} = \frac{\mathbf{K}_{b,r} \mathbf{t}_{T,r}}{\sqrt{\mathbf{t}_{T,r}^T \mathbf{K}_{b,r} \mathbf{t}_{T,r}}}.$$

4. Arrange all block scores into a single matrix, i.e.,
$$\mathbf{T}_r = [\mathbf{t}_{1,r} \cdots \mathbf{t}_{B,r}].$$

5. Compute the super loadings
$$\mathbf{p}_{T,r} = \frac{\mathbf{T}_r^T \mathbf{t}_{T,r}}{\|\mathbf{T}_r^T \mathbf{t}_{T,r}\|}.$$

6. Update the super scores, i.e.,
$$\mathbf{t}_{T,r} = \mathbf{T}_r \mathbf{p}_{T,r}$$

7. If $\mathbf{t}_{T,r}$ has converged, i.e., the change in the elements of $\mathbf{t}_{T,r}$ from one iteration to the next is smaller than a predefined precision value ϵ (e.g., $\epsilon = 10^{-6}$), the rth principal component has been computed; go to step 8. Otherwise iterate steps 3 to 6 until convergence.

8. For each block compute the residual block kernel matrix
$$\mathbf{K}_{b,r+1} = \left(\mathbf{I} - \frac{\mathbf{t}_{T,r}\mathbf{t}_{T,r}^T}{\mathbf{t}_{T,r}^T\mathbf{t}_{T,r}}\right) \mathbf{K}_{b,r} \left(\mathbf{I} - \frac{\mathbf{t}_{T,r}\mathbf{t}_{T,r}^T}{\mathbf{t}_{T,r}^T\mathbf{t}_{T,r}}\right)$$

9. Go to step 2 to compute the next PC.

A new sample $\mathbf{x}^{\text{test}}$ can be monitored via T^2, SPE charts and their block equivalents. At first, for each block $b = 1, \ldots, B$ the block kernel vector of $\mathbf{x}_b^{\text{test}}$ is determined by
$$\mathbf{k}_b^{\text{test}} = \langle \Phi(\mathbf{x}_b^{\text{test}}), \Phi(\mathbf{X}_b) \rangle$$
and centered using training data estimates, i.e.,
$$\tilde{\mathbf{k}}_b^{\text{test}} = \mathbf{k}_b^{\text{test}} - \mathbf{1}_n^T \mathbf{K}_b - \mathbf{1}_n \mathbf{k}_b^{\text{test}} + \mathbf{1}_n^T \mathbf{K}_b \mathbf{1}_n$$
where $\mathbf{1}_n^T = (\frac{1}{n}, \ldots, \frac{1}{n}) \in \mathbb{R}^{1 \times n}$. In order to determine the block scores of the new sample, the score conversion given in step 3 of the algorithm is summarized to a block coefficient matrix $\mathbf{A}_{b,r}$ for block b of PC r. It is given by
$$\mathbf{A}_{b,r} = \frac{\mathbf{t}_{T,r}}{\sqrt{\mathbf{t}_{T,r}^T \mathbf{K}_b \mathbf{t}_{T,r}}}.$$

Then the block score of the rth PC of a new sample is computed by

$$\mathbf{t}_{b,r}^{\text{test}} = \mathbf{k}_b^{\text{test}} \mathbf{A}_{b,r}.$$

Order the resulting block scores in a matrix $\mathbf{T}^{\text{test}} = [\mathbf{t}_{1,r}^{\text{test}} \cdots \mathbf{t}_{B,r}^{\text{test}}]$ and use the super loadings $\mathbf{p}_{T,r}$ to compute the super score element of the new sample vector $\mathbf{x}^{\text{test}}$ by

$$t_{T,r}^{\text{test}} = \mathbf{T}^{\text{test}} \mathbf{p}_{T,r}.$$

Use R PCs to compute the super T^2 statistic to monitor the super scores, i.e.,

$$T^2 = (\mathbf{t}_T^{\text{test}}))^T \mathbf{\Lambda}^{-1} \mathbf{t}_T^{\text{test}}$$

where $\mathbf{\Lambda}$ is the covariance matrix estimate of the super scores determined in phase 1. The corresponding block T^2 statistic is given by

$$T_b^2 = (\mathbf{t}_b^{\text{test}}))^T \mathbf{\Lambda}_b^{-1} \mathbf{t}_b^{\text{test}}$$

where $\mathbf{\Lambda}_b$ is the covariance matrix estimate of the phase 1 block scores of variable block b.

The block SPE statistic can be determined by

$$\begin{aligned} \text{SPE}_b &= \overline{\Phi}(\mathbf{x}_b^{\text{test}})(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\overline{\Phi}^T(\mathbf{x}_b^{\text{test}}) \\ &= \overline{k}(\mathbf{x}_b^{\text{test}}, \mathbf{x}_b^{\text{test}}) - (\tilde{\mathbf{k}}_b^{\text{test}})^T \mathbf{A}_b \mathbf{A}_b^T \tilde{\mathbf{k}}_b^{\text{test}} \end{aligned}$$

where

$$\overline{k}(\mathbf{x}_b^{\text{test}}, \mathbf{x}_b^{\text{test}}) = k(\mathbf{x}_b^{\text{test}}, \mathbf{x}_b^{\text{test}}) - \frac{2}{n} \sum_{i=1}^{n} k(\mathbf{x}_{b,i}, \mathbf{x}_b^{\text{test}}) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{x}_{b,i} \mathbf{x}_{b,j})$$

and $k(\mathbf{x}, \mathbf{y})$ is a kernel function valid for multi-block analysis, e.g. the Gaussian kernel. Then the overall super SPE statistic is simply the sum of the block SPE statistics, i.e.,

$$\text{SPE} = \sum_{b=1}^{B} \text{SPE}_b.$$

In summary, these statistics enable fault diagnosis and a root cause analysis drilled down to variable block level for process monitoring models based on KPCA when a multi-block-feasible kernel is used.

4.3 Robust Kernel PCA

As with classical PCA, outliers can also have an effect on the resulting scores and loadings of kernel PCA with any unbounded kernel (see Debruyne et al. (2010)). Several approaches to a more robust version of KPCA have been proposed (see e.g. Lu et al. (2004), Nguyen and De La Torre (2009)). However, these approaches are mostly restricted to the Gaussian kernel and not based on high-breakdown methods in the linear case (see Debruyne and Verdonck (2010), p. 156). As proposed in Debruyne and Verdonck (2010), three kernel versions of robust PCA approaches can be formulated. Spherical KPCA is based on spherical PCA (see Locantore et al. (1999)), where the data is first projected on a sphere around the spatial median and then PCA is performed on the sphered data. Kernel projection pursuit is based on robust projection pursuit PCA and kernel ROBPCA is a kernel version of ROBPCA. Out of the three proposed approaches, kernel ROBPCA essentially applies classical KPCA to a more robust h-subset of the initial data set. This feature makes kernel ROBPCA simple and flexible for further extensions as all techniques and modifications that apply to classical KPCA seamlessly apply to kernel ROBPCA. This is especially useful when it is considered in a process monitoring context where fault diagnosis techniques are needed in addition to a robust consideration. Based on the robust subset as determined by kernel ROBPCA we propose a technique for robust centering of the kernel matrix.

As in linear ROBPCA, kernel ROBPCA relies on the Stahel-Donoho outlyingness measure as given in equation (3.5). This is possible as the Stahel-Donoho outlyingness measure can be computed in any kernel induced feature space (see Debruyne (2009)). The kernel ROBPCA algorithm as proposed in Debruyne and Verdonck (2010) is given as follows.

Kernel ROBPCA

1. For $d = 1$ to 500 directions do the following.
   (a) Take a vector $\boldsymbol{\lambda}$ with all zeros except for 1 at position i and -1 at position j with (i, j) chosen at random. This vector represents a random direction between two points in the feature space F.
   (b) Compute the vector containing all univariate projections of the feature vectors on the corresponding direction, i.e.,

$$\mathbf{a} = \frac{\mathbf{K}\boldsymbol{\lambda}}{\sqrt{\boldsymbol{\lambda}^T \mathbf{K}\boldsymbol{\lambda}}}$$

   where $\mathbf{K}$ denotes the kernel matrix.

(c) Compute the outlyingness measure for each observation on this direction, i.e.,

$$\mathbf{r}_d = \frac{\mathbf{a} - m(\mathbf{a})}{S(\mathbf{a})}$$

where m and S denote the MCD estimators of location and scale respectively.

(d) Store for each observation its maximum outlyingness.

2. Form a subset using the h observations with the lowest maximum outlyingness.

3. Apply classical KPCA on this h-subset.

Using the robust loadings as determined by kernel ROBPCA the scores for the full set of n observations can be computed in a robust way.

Chapter 5

Robust Multi-block Multi-way KPCA

In the following we propose a novel method for monitoring complex production processes with nonlinear characteristics. Based on the methods discussed in the previous chapters we introduce *robust multi-block multi-way kernel principal component analysis* (RobMBMWKPCA) for robust fault detection, fault diagnosis and on-line monitoring of production processes that produce three-way data arrays.

In general, semiconductor production equipments such as chemical vapor deposition (CVD), physical vapor deposition (PVD) or etch tools process single wafers sequentially over a certain processing time. During the wafer processing data of several process variables is recorded at a fixed frequency, e.g., one measurement vector per second. The resulting data arrays are of three-way nature. Furthermore, process variables can often be grouped into variable blocks based on their functional relationships or their assignment to functional equipment units, e.g., variables measured on pressure or temperatur units. The resulting trajectories of such process variables over time as well as their relationships can often be of nonlinear nature due to the underlying physics.

The proposed approach offers the possibility to monitor such production processes in a nonlinear way. It ensures both robust fault detection by relying on robust statistics and fault diagnosis drilled down to variable block level. Furthermore, by using a multi-way approach fault detection and diagnosis is extended to the third data dimension and on-line monitoring is enabled. By getting information on the variables involved in a fault during batch processing, corrective measures can be applied before a batch is completed or scrapped. This allows a detailed and nonlinear process monitoring of batches during their processing on both overall and variable group level.

The chapter is organized as follows. At first, a variant of robust kernel autoscaling, i.e., mean centering and variance scaling in the kernel feature space, is proposed. Then, the complete RobMBMWKPCA procedure consisting of phase 1 model construction, phase 2 monitoring and on-line monitoring is introduced. Finally, its implementation in `R` is discussed.

5.1 Robust Autoscaling in the Feature Space

5.1.1 Robust Mean Centering

Based on the kernel ROBPCA algorithm a robust centering of the kernel matrix can be derived. The h least outlying points as determined by the kernel ROBPCA procedure form a robust subset in the feature space. Thus, the center of the kernel matrix $\mathbf{K}_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ can be robustly estimated using only the points from this subset.

Let H denote the set of the h least outlying points as determined by the Stahel-Donoho outlyingness measure. Then a robustly centered version $\tilde{\mathbf{K}}$ of the kernel matrix $\mathbf{K}$ can be computed by

$$\begin{aligned}
\tilde{\mathbf{K}}_{ij} &= \langle \Phi(\mathbf{x}_i) - \frac{1}{h} \sum_{\mathbf{x}_k \in H} \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) - \frac{1}{h} \sum_{\mathbf{x}_l \in H} \Phi(\mathbf{x}_l) \rangle \\
&= \mathbf{K}_{ij} - \frac{1}{h} \sum_{\mathbf{x}_k \in H} \langle \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) \rangle - \frac{1}{h} \sum_{\mathbf{x}_l \in H} \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_l) \rangle \\
&\quad + \frac{1}{h^2} \sum_{\mathbf{x}_k \in H} \sum_{\mathbf{x}_l \in H} \langle \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_l) \rangle \\
&= \mathbf{K}_{ij} - \frac{1}{h} \sum_{\mathbf{x}_k \in H} k(\mathbf{x}_k, \mathbf{x}_j) - \frac{1}{h} \sum_{\mathbf{x}_l \in H} k(\mathbf{x}_i, \mathbf{x}_l) + \frac{1}{h^2} \sum_{\mathbf{x}_k \in H} \sum_{\mathbf{x}_l \in H} k(\mathbf{x}_k, \mathbf{x}_l)
\end{aligned}$$

or written in terms of matrix operations using the kernel matrix $\mathbf{K}$:

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_h^* \mathbf{K} - \mathbf{K} (\mathbf{1}_h^*)^T + \mathbf{1}_h^* \mathbf{K} (1_h^*)^T \tag{5.1}$$

with $\mathbf{1}_h^* \in \mathbb{R}^{n \times n}$ such that

$$[\mathbf{1}_h^*]_{ij} = \begin{cases} \frac{1}{h} & \text{if } \mathbf{x}_j \in H \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

for $i, j = 1, \dots, n$. Analogously, a test data kernel matrix $\mathbf{K}_{\text{test}} \in \mathbb{R}^{n_{\text{test}} \times n}$ can be robustly centered with the robust training data center estimation by

$$\tilde{\mathbf{K}}_{\text{test}} = \mathbf{K}_{\text{test}} - \mathbf{1}_h^{*\prime}\mathbf{K} - \mathbf{K}_{\text{test}}(\mathbf{1}_h^*)^T + \mathbf{1}_h^{*\prime}\mathbf{K}(\mathbf{1}_h^*)^T$$

where $\mathbf{1}_h^{*\prime} \in \mathbb{R}^{n_{\text{test}} \times n}$ is defined as $\mathbf{1}_h^*$ but with n_{test} rows.

An alternative method for robust kernel matrix centering is centering around the spatial median instead of the mean as proposed in Debruyne et al. (2010). The procedure involves the use of an iterative algorithm in order to find the vector of coefficients that determine the spatial median in the feature space.

5.1.2 Multi-Way KPCA and Robust Variance Scaling

Batch processes usually have more complex nonlinear characteristics as standard continous processes as batches are often processed in different stages. Thus, efficient and detailed nonlinear process monitoring is necessary. For this purpose the standard KPCA method has been extended for three-way batch processes leading to multi-way kernel PCA (see Lee et al. (2004b)). The approach is basically the same as in the linear case. The three-way array with I observations in J variables measured K times is unfolded to a two-way matrix of dimension $I \times JK$ and standard KPCA is performed. For a robust version kernel ROBPCA can be used. However, in Lee et al. (2004b) it is suggested to perform variance scaling in the feature space in addition to kernel mean centering in the multi-way KPCA context.

Let $\mathbf{X} \in \mathbb{R}^{I \times JK}$ be an unfolded three-way array. Let $\mathbf{K}$ denote the corresponding kernel matrix, i.e., $\mathbf{K}_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{1 \times JK}$ are unfolded sample vectors for $i, j = 1, \dots, I$. Variance scaling of the centered kernel matrix $\tilde{\mathbf{K}}$ is performed by

$$\begin{aligned}
\tilde{\mathbf{K}}_{\text{scl},ij} &= \frac{\left\langle \Phi(\mathbf{x}_i) - \frac{1}{I}\sum_{k=1}^{I}\Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) - \frac{1}{I}\sum_{k=1}^{I}\Phi(\mathbf{x}_k) \right\rangle}{\frac{1}{I-1}\sum_{l=1}^{I}\left(\Phi(\mathbf{x}_l) - \frac{1}{I}\sum_{m=1}^{I}\Phi(\mathbf{x}_m)\right)^2} \\
&= \frac{\tilde{\mathbf{K}}_{ij}}{\frac{1}{I-1}\sum_{l=1}^{I}\tilde{\mathbf{K}}_{ll}}
\end{aligned}$$

or expressed in matrix operations,

$$\tilde{\mathbf{K}}_{\text{scl}} = \frac{\tilde{\mathbf{K}}}{\operatorname{trace}(\tilde{\mathbf{K}})/(I-1)}.$$

A robust kernel variance scaling can again be achieved by utilizing the robust h-subset as determined by kernel ROBPCA. Let H be this robust subset of the h least outlying observations in the feature space and let K^H denote the kernel matrix of the observations in H. Then a robustly autoscaled kernel matrix is given by

$$\begin{aligned}\tilde{\mathbf{K}}^{\text{rob}}_{\text{scl},ij} &= \frac{\left\langle \Phi(\mathbf{x}_i) - \frac{1}{h}\sum_{\mathbf{x}_k \in H}\Phi(\mathbf{x}_k), \Phi(\mathbf{x}_j) - \frac{1}{h}\sum_{\mathbf{x}_k \in H}\Phi(\mathbf{x}_k)\right\rangle}{\frac{1}{h-1}\sum_{\mathbf{x}_l \in H}\left(\Phi(\mathbf{x}_l) - \frac{1}{h}\sum_{\mathbf{x}_m \in H}\Phi(\mathbf{x}_m)\right)^2} \\ &= \frac{[\tilde{\mathbf{K}}^{\text{rob}}]_{ij}}{\frac{1}{h-1}\sum_{l=1}^{h}[\tilde{\mathbf{K}}^H]_{ll}}\end{aligned}$$

or written in matrix operations,

$$\tilde{\mathbf{K}}^{\text{rob}}_{\text{scl}} = \frac{\tilde{\mathbf{K}}^{\text{rob}}}{\text{trace}(\tilde{\mathbf{K}}^H)/(h-1)}.$$

Using robust kernel autoscaling as proposed above in combination with kernel ROBPCA allows a robust multi-way kernel PCA modeling.

5.2 The RobMBMWKPCA Procedure

Kernel ROBPCA simply applies classical KPCA on a robust subset of the data. This makes it very flexible and easy to extend. Thus, multi-block KPCA can be applied on the robust h-subset. Furthermore, the underlying block kernel matrices can be determined from an unfolded multi-way array where the unfolding is performed as schematically depicted in figure 3.4.2. This leads to robust multi-block multi-way KPCA. The complete procedure is presented in the following.

5.2.1 Phase 1: Construction of Normal Operating Condition Model

We begin with phase 1, i.e., the model construction stage as discussed in chapter 2. Let $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ be a multi-way array of data representing the normal operating condition of a production process. Let the J variables be ordered in B conceptually meaningful blocks.

1. Unfold $\mathbf{X}$ to a matrix $\mathbf{X}_{\mathrm{u}}$ of dimension $I \times JK$.

2. Autoscale $\mathbf{X}_{\mathrm{u}}$ using robust estimators to obtain $\tilde{\mathbf{X}}_{\mathrm{u}}$. For example, use the L_1-median as robust estimator of multivariate location and the MAD as robust scale estimator for each column.

3. Compute the kernel matrix $\mathbf{K}$ of $\tilde{\mathbf{X}}_{\mathrm{u}}$ using a multi-block valid kernel function k, e.g. the Gaussian kernel.

4. Apply steps 1 and 2 of the kernel ROBPCA procedure on $\mathbf{K}$ to determine the subset of h observations least outlying in terms of the Stahel-Donoho outlyingness. The value h determines the robustness and has to be chosen in advance. Typical values are $h = 0.75I$ or $h = 0.8I$.

5. For each variable block use k to compute the block kernel matrices $\mathbf{K}_b^h \in \mathbb{R}^{h\times h}$ using only observations from the h-subset.

6. Autoscale each block kernel matrix using kernel mean centering and kernel variance scaling to obtain $\tilde{\mathbf{K}}_b^h$, i.e.,

$$\begin{aligned}\mathbf{K}_b^{h,\mathrm{mc}} &= \mathbf{K}_b^h - \mathbf{1}_h\mathbf{K}_b^h - \mathbf{K}_b^h\mathbf{1}_h + 1_h\mathbf{K}_b^h\mathbf{1}_h \\ \tilde{\mathbf{K}}_b^h &= \frac{\mathbf{K}_b^{h,\mathrm{mc}}}{\mathrm{trace}(\mathbf{K}_b^{h,\mathrm{mc}})/(h-1)}\end{aligned}$$

   where $\mathbf{1}_h$ is a $h \times h$ matrix with all entries equal to $1/h$.

7. Apply steps 2 to 9 of the multi-block KPCA algorithm on the block kernel matrices $\mathbf{K}_b^h$ and extract R principal components. This results in matrices of robust super scores $\mathbf{t}_T^h \in \mathbb{R}^{h\times R}$, robust block scores $\mathbf{t}_{\cdot,r}^h \in \mathbb{R}^{h\times B}$ for $r = 1, \ldots, R$ and robust super loadings $\mathbf{p}_T \in \mathbb{R}^{B\times R}$.

8. For each block compute the robust block coefficient matrix $\mathbf{A}_b^h \in \mathbb{R}^{h\times R}$ by

$$\mathbf{A}_{b,r}^h = \frac{\mathbf{t}_{T,r}^h}{\sqrt{(\mathbf{t}_{T,r}^h)^T\mathbf{K}_b^h\mathbf{t}_{T,r}^h}}$$

9. In order to determine full block scores $\mathbf{t}_{\cdot,r} \in \mathbb{R}^{I\times B}$ and full super scores $\mathbf{t}_T \in \mathbb{R}^{I\times R}$ in a robust way, kernel block matrices $\mathbf{K}_b^* \in \mathbb{R}^{I\times h}$ are required for each block. For $b = 1, \ldots, B$ they are given by

$$[\mathbf{K}_b^*]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j^h)$$

   where $\mathbf{x}_i$ for $i = 1, \ldots, I$ are rows of $\mathbf{X}_{\mathrm{u}}$ and $\mathbf{x}_j^h$ for $j = 1, \ldots, h$ are observations from the robust h-subset.

10. Apply robust kernel autoscaling on $\mathbf{K}_b^*$ to obtain $\tilde{\mathbf{K}}_b^* \in \mathbb{R}^{I \times h}$. Here, the robust kernel centering can be written in terms of the unscaled $\mathbf{K}_b^*$ and the h-subset kernel matrix $\mathbf{K}_b^h$, i.e.,

$$\begin{aligned} \mathbf{K}_b^{*,\mathrm{mc}} &= \mathbf{K}_b^* - \mathbf{1}_h^{'}\mathbf{K}_b^h - \mathbf{K}_b^*\mathbf{1}_h + \mathbf{1}_h^{'}\mathbf{K}_b^h\mathbf{1}_h \\ \tilde{\mathbf{K}}_b^* &= \frac{\mathbf{K}_b^{*,\mathrm{mc}}}{\mathrm{trace}(\mathbf{K}_b^{h,\mathrm{mc}})/(h-1)} \end{aligned}$$

where $\mathbf{1}_h^{'}$ is a $I \times h$ matrix with all entries equal to $1/h$. Note that robust kernel centering of the non-square $\mathbf{K}_b^*$ can also be written only in terms of $\mathbf{K}_b^*$, as proposed in (5.1) for square kernel matrices. This gives the same result, i.e.,

$$\mathbf{K}_b^{*,\mathrm{mc}} = \mathbf{K}_b^* - \mathbf{1}_h^*\mathbf{K}_b^* - \mathbf{K}_b^*\mathbf{1}_h + \mathbf{1}_h^*\mathbf{K}_b^*\mathbf{1}_h$$

with $\mathbf{1}_h^* \in \mathbb{R}^{I \times I}$ defined analogously to (5.2).

11. Compute the full block scores for each block as well as the full super scores in a robust way, i.e.,

$$\begin{aligned} \mathbf{t}_{b,r} &= \tilde{\mathbf{K}}_b^*\mathbf{A}_{b,r} \\ \mathbf{t}_{T,r} &= \mathbf{T}_r\mathbf{p}_{T,r} \end{aligned}$$

where each $\mathbf{T}_r \in \mathbb{R}^{I \times B}$ is the matrix with block scores in columns for PCs $r = 1, \ldots, R$.

12. Compute the robust block T^2 statistics based on robust estimation of the block scores covariance and the robust super T^2 statistic based on a robust estimation of the super scores covariance matrix. For example, use the robust RMCD covariance matrix estimator. Furthermore, compute the block SPE and super SPE statistics as discussed above.

13. Determine the respective control limits for all monitoring statistics using the nonparametric bootstrapping procedure.

5.2.2 Phase 2: Test Data Monitoring

We continue with phase 2, i.e., the monitoring stage. Let $\mathbf{X}^{\mathrm{test}} \in \mathbb{R}^{I_{\mathrm{test}} \times J \times K}$ be a three-way array of data from I_{test} new batches with unknown operating condition and the same variable block ordering.

1. Unfold $\mathbf{X}^{\mathrm{test}}$ to a $I_{\mathrm{test}} \times JK$ matrix and autoscale it using the robust location and scale estimators from the normal operating condition model.

2. For each block use k to compute the test block kernel matrices $\mathbf{K}_b^{\text{test}} \in \mathbb{R}^{I_{\text{test}} \times h}$, i.e.,

$$[\mathbf{K}_b^{\text{test}}]_{ij} = k(\mathbf{x}_i^{\text{test}}, \mathbf{x}_j^h)$$

where $\mathbf{x}_i^{\text{test}} \in \mathbb{R}^{1 \times JK}$ are new observations for $i = 1, \ldots, I_{\text{test}}$.

3. Apply robust kernel autoscaling on $\mathbf{K}_b^{\text{test}}$ to obtain $\tilde{\mathbf{K}}_b^{\text{test}}$, i.e.,

$$\begin{aligned} \mathbf{K}_b^{\text{test,mc}} &= \mathbf{K}_b^{\text{test}} - \mathbf{1}_h^{\text{test}}\mathbf{K}_b^h - \mathbf{K}_b^{\text{test}}\mathbf{1}_h + \mathbf{1}_h^{\text{test}}\mathbf{K}_b^h\mathbf{1}_h \\ \tilde{\mathbf{K}}_b^{\text{test}} &= \frac{\mathbf{K}_b^{\text{test,mc}}}{\text{trace}(\mathbf{K}_b^{h,\text{mc}})/(h-1)} \end{aligned}$$

where $\mathbf{1}_h^{\text{test}}$ is a $I_{\text{test}} \times h$ matrix with all entries equal to $1/h$.

4. Compute the test block scores and test super scores using the robust loadings, i.e.,

$$\begin{aligned} \mathbf{t}_{b,r}^{\text{test}} &= \tilde{\mathbf{K}}_b^{\text{test}}\mathbf{A}_{b,r} \\ \mathbf{t}_{T,r}^{\text{test}} &= \mathbf{T}_r^{\text{test}}\mathbf{p}_{T,r} \end{aligned}$$

where each $\mathbf{T}_r^{\text{test}}$ is the matrix with test block scores in columns for PCs $r = 1, \ldots, R$.

5. Compute the T^2 and SPE monitoring statistics at block and super level using the robust covariance matrix estimation from the normal operating condition model.

6. For each of the I_{test} batches monitor whether T^2 or SPE exceeds its respective control limit to get information on the performance of the new batch in comparison to the normal operating condition.

5.2.3 On-line Monitoring

Let $x^t \in \mathbb{R}^{1 \times J \times t}$ be data of a single new batch that is in the middle of production, i.e., measurements are available only until the current time point $t < K$.

1. Unfold $\mathbf{x}^t$ to a vector of size $1 \times Jt$ and autoscale $\mathbf{x}^t$ with robust estimators from the normal operating condition.

2. Anticipate future measurements from t onwards by filling them with the deviation observed at time t. This results in a complete batch observation $\mathbf{x}^{t*}$.

3. Apply steps 2-5 from the test data monitoring procedure on $\mathbf{x}^{t*}$.

4. Repeat these steps every time a new measurement is conducted for this batch.

By applying this procedure monitoring statistics at block and super level are made available every time a new J-dimensional measurement vector is added to the batch, e.g. every second. As the procedure is fast and can be applied practically in real-time, this allows on-line fault detection and diagnosis during the batch production.

5.3 Implementation in R

The steps of the complete RobMBMWKPCA procedure are now discussed in terms of their implementation in `R`.

5.3.1 Phase 1 in R

1. We begin by assuming that the three-way array $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ of phase 1 data, i.e., data representing the normal operating condition, is given as an unfolded data matrix $\mathbf{X}_u \in \mathbb{R}^{I \times JK}$. The data is read in and stored in variable `X.unfold` with `I` rows and `J*K` columns.

2. Robust mean centering of the `X.unfold` is performed using the function `l1median_NLM()` from the `pcaPP` package (see Filzmoser et al. (2014)). Robust variance scaling is performed using the function `mad()`. This results in the robustly autoscaled data matrix `X.unfold.s`.

3. The unscaled `I` × `I` kernel matrix `kernel.matrix.unscld` of the robustly autoscaled data matrix `X.unfold.s` is computed using the function `kernelMatrix()` from the `kernlab` package (see Karatzoglou et al. (2004)). The `kernel` argument determines the underlying kernel function. As a multi-block valid kernel the Gaussian kernel can be used. It is set using the `kernlab` function `rbfdot()` with an adequate value of `sigma` and stored in the variable `krnl`.

4. In order to find the subset of h least outlying points we first set a corresponding value `alpha.h` that controls the subset size, e.g., `alpha.h <- 0.8`. A function `krobpca.subset()` has been written that determines the clean subset of size `h <- alpha.h*I` based on the kernel

ROBPCA procedure. It takes a kernel matrix and a value of `alpha.h` as arguments. It is called by

```
hs <- krobpca.subset(kernel.matrix=kernel.matrix.unscld,
                     alpha=alpha.h)
```

and returns a list of indices of the corresponding `kernel.matrix` that are least outlying in terms of the Stahel-Donoho outlyingness, i.e., `h.subset <- hs$indices`.

5. Let `numberofblocks` be the number of variable blocks. Robust unscaled kernel matrices can be constructed for each variable block using only clean observations from the h subset. These kernel matrices are stored in an array `kernel.matrices.rob.unscld` of dimension $\texttt{h} \times \texttt{h} \times$ `numberofblocks`. For a variable block `b` ($\texttt{b} = 1, \ldots, \texttt{numberofblocks}$) the matrices are computed using the function `kernelMatrix()` only on `X.unfold.s[h.subset, block.b.columns]` where `block.b.columns` are the columns of `X.unfold.s` that correspond to variable block `b`.

6. We can now perform autoscaling in the feature space on the resulting block kernel matrices. Then mean centering and variance scaling can be performed as proposed in section 5.1. The result is the array `kernel.matrices.rob` of dimension $\texttt{h} \times \texttt{h} \times$ `numberofblocks` of `numberofblocks` robustly autoscaled block kernel matrices of clean observations.

7. We are now ready to apply multi-block KPCA on the array of robustly autoscaled block kernel matrices `kernel.matrices.rob`. A function `mbkpca()` has been written that determines super scores and super loadings as well as block scores and block coefficient matrices from a given array of block kernel matrices. It takes the block kernel matrix array and the number `numberofcomponents` of principal components to extract as arguments. It is called by

```
model <- mbkpca(kernel.matrices=kernel.matrices.rob,
                components=numberofcomponents)
```

and returns robust super and block statistics for a given number of components, i.e.,

```
super.scores.h <- model$super.scores
super.loadings <- model$super.loadings
block.scores.h <- model$block.scores
```

where `super.scores.h` is a `h` × `numberofcomponents` matrix of super scores of the clean h-subset and `block.scores.h` results in a `h` × `numberofblocks` × `numberofcomponents` array of block scores of the clean h-subset. The `numberofblocks` × `numberofcomponents` matrix `super.loadings` is a matrix of robust multi-block KPCA super loadings.

8. The function `mbkpca()` also returns robust block coefficient matrices for each variable block, i.e.

   ```
   A.h <- model$block.coeff
   ```

   where `A.h` is a `h` × `numberofblocks` × `numberofcomponents` array.

9. Based on the super and block statistics of the clean subset of `h` we can now derive robust super and block statistics for all `I` observations. For this purpose we first compute full and unscaled `I` × `h` block kernel matrices for each block. These are stored in the `I`×`h`×`numberofblocks` array `kernel.matrices.full.unscld`. For block `b` we use the function `kernelMatrix(krnl,x,y)` with the chosen kernel function `krnl`. Its argument `x` is `X.unfold.s[,block.b.columns]` and its argument `y` is `X.unfold.s[h.subset,block.b.columns]`.

10. These kernel matrices can now again be autoscaled in the feature space. This results in a `I` × `h` × `numberofblocks` array `kernel.matrices` of robust and robustly autoscaled kernel matrices.

11. For observation `i` (`i` $= 1, \ldots,$ `I`), functional variable block `b` (`b` $= 1, \ldots,$ `numberofblocks`) and extracted principal component `c` (`c` $= 1, \ldots,$ `numberofcomponents`) we can now compute full robust block scores via

    ```
    block.scores[i,b,c] <- kernel.matrices[i,,b]%*%A.h[,b,c]
    ```

    where `block.scores` is an array of dimension `I` × `numberofblocks` × `numberofcomponents`. With this, full robust super scores can be computed via

```
super.scores[i,c] <- block.scores[i,,c]%*%super.loadings[,c]
```

where `super.scores` is a matrix with `I` rows and `numberofcomponents` columns.

12. We can now compute the phase 1 T^2 and SPE statistics. For the T^2 statistics we use the RMCD estimator for robust estimation of the covariance matrix of both block scores and super scores. It can be computed using the function `covRob()` along with its control function `covRob.control()` and the option `estim="weighted"` from the `robust` package (see Wang et al. (2014)). The T^2 values can then be computed using the function `mahalanobis()`.

    The phase 1 SPE statistics can be computed analogously to the SPE statistics for test data given in section 4.2. Here, x_b^{test} has to be replaced with the training data equivalent x_b.

13. For the computation of the upper control limits for the T^2 statistics we use the nonparametric bootstrapping approach. For this purpose, a general function `t2limit()` has been written that determines T^2 control limits using different methods. The call

```
limit <- t2limit(hds=super.scores, type="bootstrap",
                 alpha=0.05, ci.type="basic",
                 center=FALSE, cov=score.est)
```

    determines a bootstrapping based UCL as $1 -$ `alpha` quantile for the T^2 statistic using the matrix `super.scores` as historic data set (`hds`).

    By using the option `type="bootstrap"` the function uses the functionalities of the package `boot` (see Canty and Ripley (2014)) to compute a UCL based on 1000 bootstrap replicates. Other options for `type` are `"mvn"` for a UCL determination based on the assumption of multivariate normality of the underlying `hds`, i.e., the $1 -$ `alpha` of the corresponding F distribution (see section 2.1) and `"chebyshev"` for a UCL estimation based on Chebyshev's inequality (see section 2.5). For `type="bootstrap"` the option `ci.type` determines the type of the corresponding bootstrap confidence interval. Its options are the same as for the argument `type` of the `boot` function `boot.ci()`, i.e., `"norm"` for normal approximation, `"basic"` for basic bootstrap confidence limits, `"stud"` for studentized limits, `"perc"` for a bootstrap percentile interval and `"bca"` for BC_a confidence limits.

With the options `center` and `cov` estimates for multivariate location (`center`) and the covariance matrix (`cov`) can be specified. Here, `score.est` denotes the RMCD estimate of the covariance matrix of the multi-block KPCA super scores stored in `super.scores`.

Analogously, an upper control limit for the block T^2 statistic of variable block `b` can be determined via

```
limit.b <- t2limit(hds=block.scores[,b,], type="bootstrap",
                   alpha=0.05, ci.type="basic",
                   center=FALSE, cov=block.b.est)
```

where `block.b.est` denotes the covariance estimate of the block scores of variable block `b`.

Finally, upper control limits for the super SPE and block SPE statistics can be determined by directly using the functions `boot()` and `boot.ci()` from the `boot` package.

5.3.2 Phase 2 in R

1. We assume that a matrix of unfolded test data of dimension `I.test` $\times$ `J` $*$ `K` is stored in `Xtest.unfold`. It is again autoscaled using the robust estimates of phase 1. This leads to the matrix `Xtest.unfold.s`.

2. Based on the kernel function `krnl` we construct test data block kernel matrices using the function `kernelMatrix(krnl,x,y)`. The argument `x` is the matrix `Xtest.unfold.s[,block.b.columns]` where `block.b.columns` again denotes the columns of `Xtest.unfold.s` corresponding to variable block `b`. The argument `y` is set to the h-subset of the training data, i.e., `X.unfold.s[h.subset,block.b.columns]`. This results in the array `kernel.matrices.test.unscld` of dimension `I.test` $\times$ `h` $\times$ `numberofblocks` of robust unscaled test data kernel matrices.

3. The proposed robust autoscaling in the feature space is applied to these block kernel matrices of test data using the corresponding kernel matrices. This results in an array of autoscaled block kernel matrices `kernel.matrices.test` of dimension `I.test` $\times$ `h` $\times$ `numberofblocks`.

4. For test observation `i` (`i` $= 1, \dots,$ `I.test`), functional variable block `b` (`b` $= 1, \dots,$ `numberofblocks`) and extracted principal component `c`

(c = $1, \ldots,$ numberofcomponents) we can now compute robust test block scores via

```
block.scores.test[i,b,c] <-
  kernel.matrices.test[i,,b]%*%A.h[,b,c]
```

where block.scores.test is the array of test block scores of dimension I.test × numberofblocks × numberofcomponents. With this, full robust test super scores can be computed via

```
super.scores.test[i,c] <-
  block.scores.test[i,,c]%*%super.loadings[,c]
```

where super.scores.test is a I.test × numberofcomponents matrix.

5. T^2 and SPE statistics of the test block scores and test super scores can now be computed straightforwardly using training data estimates.

6. Each of the I.test batches can now be monitored using the determined control limits.

5.3.3 On-line Monitoring in R

Given the steps from phase 2 the on-line monitoring procedure can be applied straightforwardly. For an unfolded and robustly autoscaled single batch testbatch.unfold.s that is in the middle of production, i.e., unfinished, the fill-up strategy allows a straightforward computation of super T^2 values and super SPE values as well as block T^2 values and block SPE values every time a new measurement is conducted, e.g., each second.

For each time point the fill-up of future measurements, the computation of the kernel matrix via kernelMatrix(), the robust autoscaling of the kernel matrices and the subsequent computation of super and block scores can be performed instantly. Thus, with an appropriate automated framework the monitoring statistics are available within one second. This makes the procedure suitable for the actual application in on-line monitoring.

Chapter 6

Monitoring of a Plasma Etch Process Via RobMBMWKPCA

6.1 Motivation

The ever growing process complexity on the one hand and the installation of fault detection and classification (FDC) systems for data recording, management and monitoring on the other hand lead to growing amounts of data for each of several hundred production stages of a semiconductor manufacturing line. In order to monitor the performance of these single manufacturing stages, suitable analysis methods are needed to adequately handle the generated data. However, most FDC applications only rely on univariate control charts for monitoring process variables individually. Thus, complex and inherently multivariate problems or processing failures are easily missed. Multivariate monitoring techniques such as PCA and Hotelling's T^2 are capable of advanced process monitoring. However, such advanced FDC applications require data pretreatments. For instance, summarized statistics such as means or maxima over specific temporal windows of the wafer processing time are needed. This can lead to a large number of different indicators. With such indicators information of the variation over time of the process variables can often be lost and anomalies in the process variable trajectories can possibly be overlooked. Due to the complex production environment, such anomalies can accumulate in subsequent processing steps and can lead to yield problems and even scrapped wafers. Furthermore, anomalies in the trajectories of certain process variables can point at problems with the wafer processing or the manufacturing equipment itself. Thus, in addition to overall process performance measures, information on the performance during production can serve as an indicator for the equipment health. For such applications, advanced process control procedures based on multi-way considerations are suitable.

In this chapter we exemplarily apply the presented robust nonlinear multi-way multi-block approach on a single manufacturing stage. With the approach all of the recorded temporal data information can be considered. Possible nonlinear process behaviour and process faults with nonlinear characteristics can also be taken into account. By grouping the involved process variables in blocks created based on engineering knowledge, an interpretable root cause analysis is enabled. The presented on-line monitoring technique further allows process engineers to get in-depth information on the process condition during the actual processing, both overall and on variable block level. This results in performance monitoring statistics from the beginning to the end of the processing of each wafer. Furthermore, special attention is paid on robustness. This allows an adequate assessment of the production reference and an estimation of the true normal operating condition of the production process unaffected by outlying measurements, first wafer effects or other anomalies.

6.2 Process Data

We apply the presented methodology on a plasma etch process. Plasma etching is a crucial, highly complex and repeatedly applied operation during semiconductor manufacturing. In general, the goal is to selectively remove unwanted solid material from the substrate by means of a chemical reaction and plasma, i.e., ionized gas. The process allows the creation of very small profiles in the sub-micron range on the wafer surface with high precision and selectivity.

The case study is related to a failure of the magnetic field on a plasma etch tool that occured in June 2011 at ams AG (see Hayderer (2012)). Process engineers observed a sudden decrease in the etch rate of the tool as the electromagnetic field, which is created by inductors and should exist permanently around the wafer during the process, failed to work properly. Consequently, the impedance and the electric resistance of the etch chamber changed. As a result, the radio frequency (RF) power had to be adapted in order to deal with the changed conditions. However, a univariate analysis did not show any severe out-of-control alarms.

During the processing of each wafer, data information for every process variable is recorded by sensors with a fixed frequency of one data point per second. Thus, as is the case with most single manufacturing steps, the total recorded data information can be arranged in a three-way data array of dimension $I \times J \times K$ for I wafers, J variables and K time points.

In total, $I = 390$ wafers were selected as reference data set. The wafers were processed during several days before the occurence of the magnetic field failure. Pressure anomalies, not related to the magnetic field failure, were known to occur from time to time on the etch tool during that production period. Thus, the use of robust estimation is crucial to model the normal operating condition.

A number of process variables are measured during the course of the etch process. After discussion with the process engineers, the $J = 10$ most informative variables are chosen and grouped in $B = 3$ functional variable blocks, namely *pressure*, *helium* and *RF power*. Other process variables were either constant or near constant during the processing (e.g., gas flow variables) or not informative for the immediate condition of the process or the tool (e.g., etch endpoint variables). Table 6.1 gives an overview of the used variables and their block membership.

Variable	Description	Block
`Foreline_Pressure`	pressure in the foreline between turbo pump and foreline pump	pressure
`Pressure`	pressure in the etch chamber	pressure
`Throttle_Valve_Step`	adjustable valve for chamber pressure control	pressure
`He_Flow`	helium gas flow for wafer cooling	helium
`He_Pressure`	pressure of helium	helium
`Outer_He_Leak_Rate`	leak rate of helium	helium
`RF_Forward`	power for plasma stimulation	RF power
`RF_Load_Position`	setting of matching coil for RF power	RF power
`RF_Reflected`	reflected RF power	RF power
`RF_Tune_Position`	setting of matching capacity for RF power	RF power

Table 6.1: Overview of the variables used for monitoring a plasma etch process.

For each wafer, the process variables are measured each second during its processing. Typically, the first few and last few seconds are stabilization phase and shut down phase, respectively. After removing these phases, $K = 178$ relevant time points remain. Figure 6.1 shows a typical trajectory of the variable *Pressure* during the processing of one wafer.

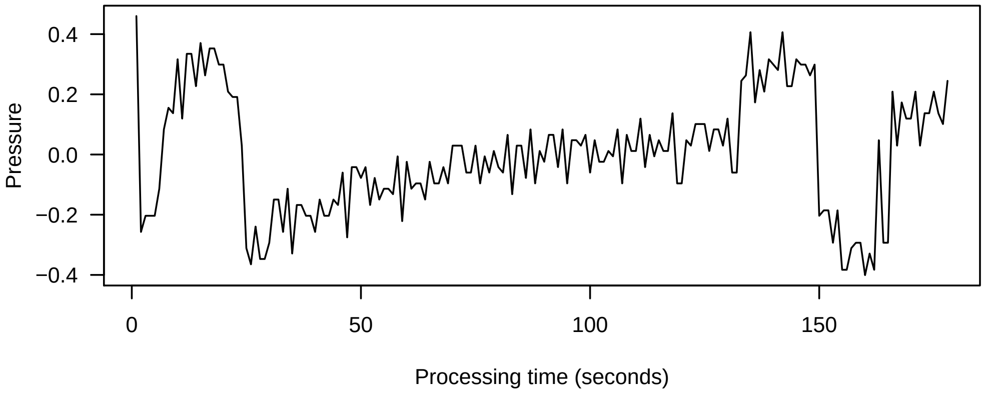


Figure 6.1: Typical trajectory of the variable *Pressure* (autoscaled) over the course of the processing of one wafer on a plasma etch tool (minus stabilization and shut down phase).

Furthermore, we only use data from a single production recipe that was affected by the fault. For a global monitoring approach, a normalization of the data separately for each recipe is recommended. Alternatively, process monitoring approaches that are independent of the underlying production recipe can also be proposed, e.g., see Chen and Blue (2009) and Blue et al. (2013).

In summary, the final reference data set to model the normal operating condition of the process consists of

- $I = 390$ wafers,
- $J = 10$ continuous variables grouped in $B = 3$ blocks,
- $K = 178$ time points.

6.3 Phase 1: Modeling

We begin with phase 1, i.e., the modeling stage. The first step is to unfold the $390 \times 10 \times 178$ array to a two-dimensional matrix of dimension 390×1780. Then this matrix is autoscaled. We use the L_1-median as defined in equation (3.6) for robust estimation of the multivariate center. It is computed using the nonlinear minimization routine as implemented in the function `l1median_NLM()` from the `R` package `pcaPP` (see Filzmoser et al. (2014)).

For robust estimation of the standard deviation of each column the median absolute deviation (MAD) can be used. However, several columns have a MAD of 0 as some variables have very small variation at certain time points. For example, for more than 70% of the wafers in the reference set the variable `RF_Forward` has a starting value of 1002, i.e., the value in the first second of the observed processing period. The result is a very low variation among the values in the respective column of the unfolded matrix. Whereas the classical and non-robust empirical standard deviation results in a value of 0.46, the MAD becomes 0. Also the interquartile range (IQR) and several alternatives to the MAD given in Rousseeuw and Croux (1993) have been checked, but all result in a robust scale value of 0 for several columns. Thus, in order to be able to perform robust autoscaling a trade-off in the degree of robustness has to be made. We chose a quantile range analogously to the IQR as estimator of univariate scale. For each column of the unfolded data matrix, the range of the inner 95%, the estimator QR_{95}, was found to work in this setting. It is given by

$$\hat{\sigma} = QR_{95} = \frac{x_{0.975} - x_{0.025}}{2q_{0.975}} = \frac{x_{0.975} - x_{0.025}}{3.92}. \tag{6.1}$$

where $x_{0.975}$ and $x_{0.025}$ denote the 97.5% and 2.5% quantiles of the data, respectively, and $q_{0.975}$ denotes the 97.5% quantile of the normal distribution. However, it has a very low degree of robustness with a breakdown point of only 0.05.

The behaviour of the variable `Pressure` as shown in figure 6.1 suggests the presence of nonlinearities and possible nonlinear relationships in the data. Thus, in order to capture possible nonlinear structures in the data we use the kernel approach. As a kernel function valid for multiblock analysis we chose the Gaussian kernel of the form

$$k(\mathbf{x}, \mathbf{y}) = \exp\Big(- \sigma \|\mathbf{x} - \mathbf{y}\|^2 \Big). \tag{6.2}$$

After testing the monitoring performance for various values of the parameter σ, we found $\sigma = 0.00001$ to be appropriate in this setting. In order to choose the number of principal components to extract we use kernel parallel analysis (kPA) for the chosen value of σ as a guideline due to its simplicity. Thus, for our training set of $I = 390$ wafers the resulting number of components q is 16.

We applied the proposed robust multi-way multi-block KPCA procedure to determine robust scores and loadings of this model. We compare two different robust approaches, i.e., with two different sizes of the h-subset of clean observations for both kernel ROBPCA (determined by α_{KPCA}) and the RMCD estimator (determined by α_{RMCD}), with a classical non-robust approach. The classical approach is based on autoscaling the input data with the sample mean and the sample standard deviation, classical kernel PCA of the multi-way and multi-block data and T^2 values based on the sample covariance matrix. The resulting T^2 charts of the $I = 390$ training data scores is shown in figure 6.2.

Overall, the robust chart with $\alpha_{KPCA} = \alpha_{RMCD} = 0.8$ describes best the behaviour of the process at the observed time. Strong single peaks indicate exactly the pressure anomalies known to occur during that production period. Due to robust estimation of both the KPCA results and the T^2 values, these excursions do not affect our estimation of the normal operating condition. This is considered crucial for adequate process monitoring as future wafers are compared to this reference. However, the abnormal wafers have an undesired effect on the estimation of the reference in the non-robust case. Thus, the overall T^2 level is much lower and the discrimination between normal and abnormal process behaviour is worse than in the robust cases. In summary, the classical non-robust approach does not depict the process behaviour as accurate and differentiated as the robust models.

Figure 6.3 shows the distributions of these three sets of T^2 values compared to the beta distribution that results from the multivariate normal case.

The deviations from the beta distribution for the robust models highlight the need of distribution-free procedures to determine the UCL. We use the discussed bootstrapping approach. For computation we use the `R` package `boot`. For a false alarm rate of $\alpha = 0.05$ this results in UCLs of 97.83 for $\alpha_{KPCA} = \alpha_{RMCD} = 0.9$ and 91.31 for $\alpha_{KPCA} = \alpha_{RMCD} = 0.8$. As expected, the distribution of the classical T^2 values is closer to the theoretical beta distribution than in the robust case. For comparison, the UCL as 0.95 quantile of the beta distribution is 25.95, its bootstrapped alternative is 29.77.

In summary, we use the robust model with $\alpha_{KPCA} = \alpha_{RMCD} = 0.8$ as reference.

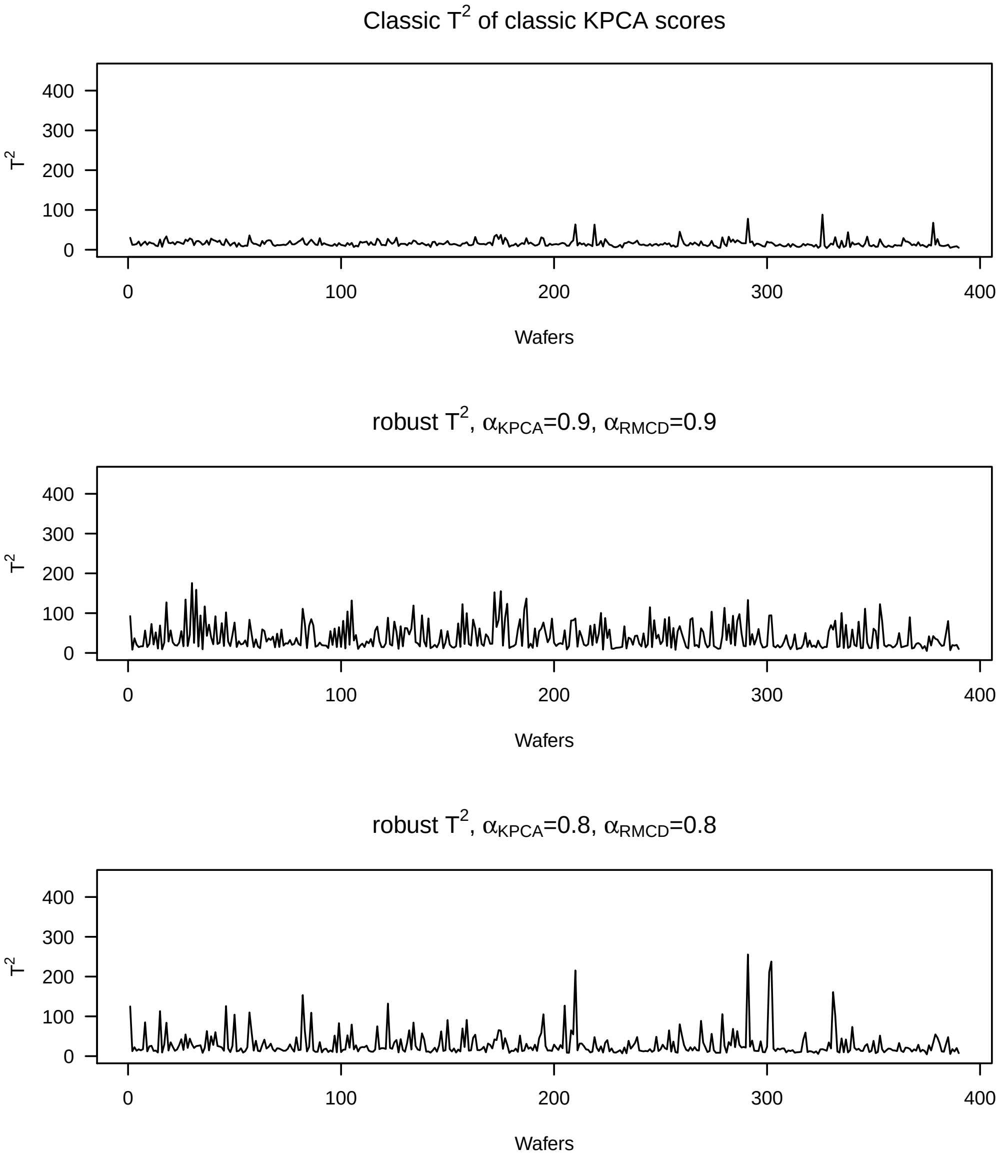


Figure 6.2: Classic (non-robust) and robust T^2 values of classic and robust KPCA scores.

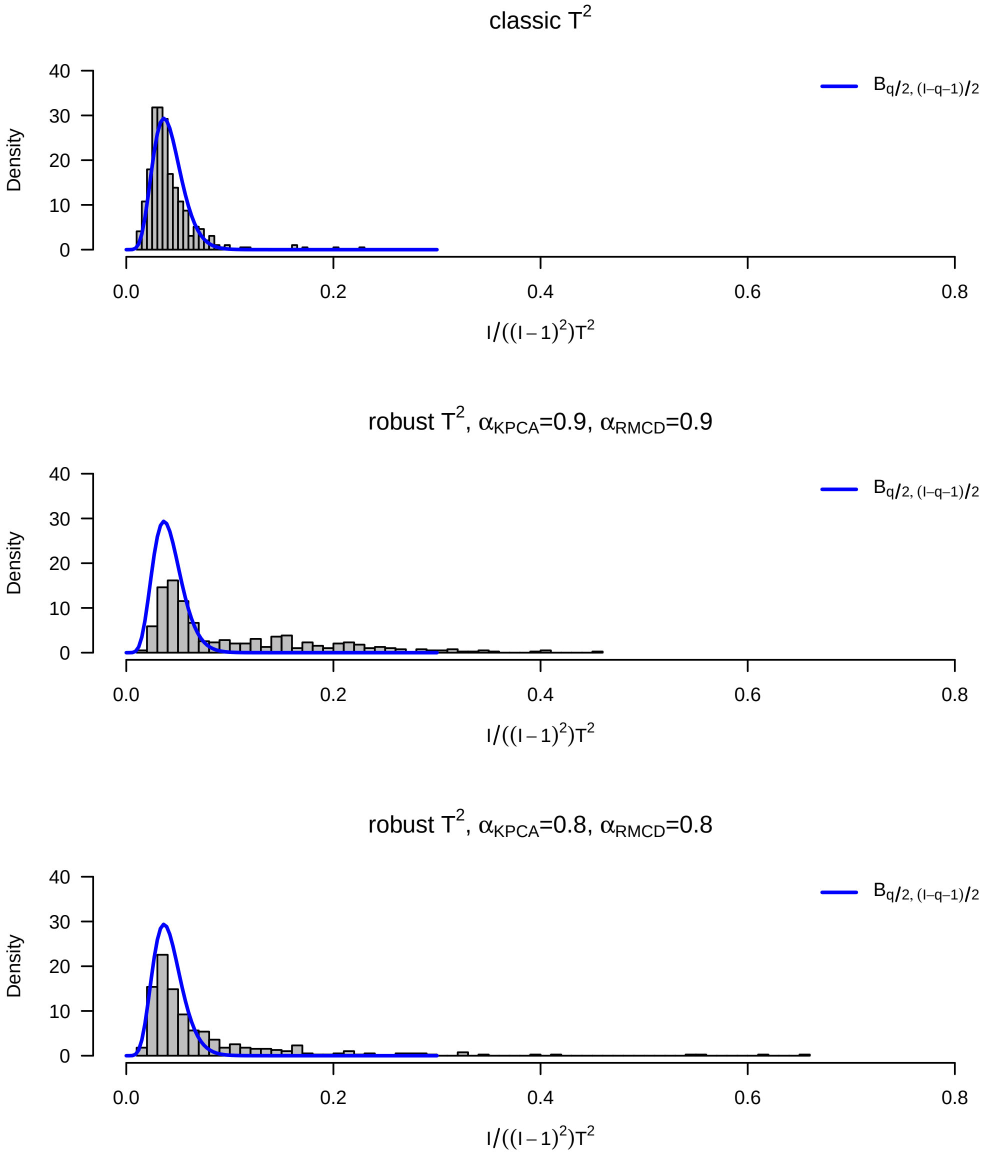


Figure 6.3: Histograms of $I = 390$ scaled T^2 values of KPCA scores ($q = 16$ PCs) along with the respective beta distribution (blue).

6.4 Phase 2: Test Data Results

For phase 2, we apply the proposed approach to an independent test data set of $I_{test} = 454$ wafers from the production period of interest in June 2011. The test scores where determined as proposed, based on the reference model with $q = 16$ extracted principal components. The resulting robust T^2 chart of robust test super scores is shown in figure 6.4 along with the respective bootstrap UCL ($\alpha = 0.05$) and its 95% BC_a confidence limits.

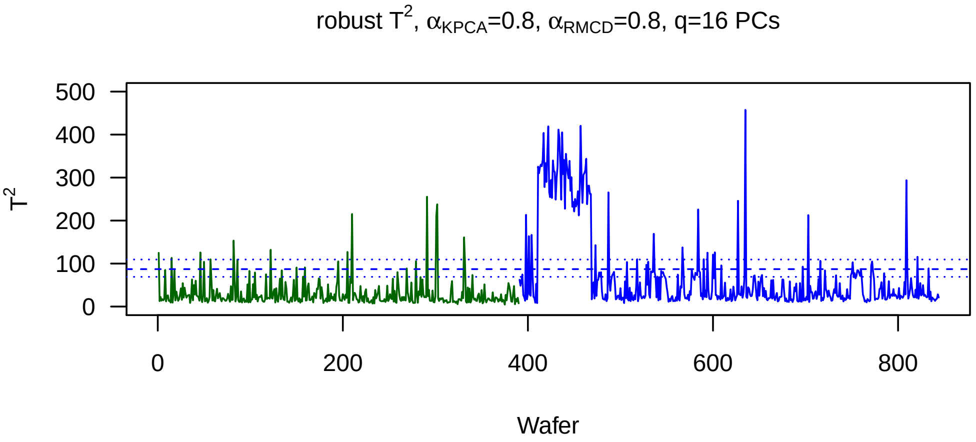


Figure 6.4: Robust T^2 chart of robust KPCA super scores of $I = 390$ reference wafers (green) and $I_{test} = 454$ test wafers (blue) along with the bootstrapped UCL (dashed) and its BC_a confidence interval (dotted).

Along with the UCL of 87.05 as determined by bootstrapping (1000 bootstrap resamples) the corresponding BC_a confidence interval $[69.15, 109.46]$ is also shown. In comparison, the corresponding basic bootstrap interval is given as $[66.64, 105.50]$.

Exactly the wafers affected by the magnetic field failure (wafers 412 to 469 of the 845 analyzed wafers) are correctly identified as highly abnormal. Furthermore, several single abnormalities are also identified. The corresponding squared prediction error (super SPE) chart that measures the deviations of the observations from the kernel PCA model space is shown in figure 6.5. In addition, the control limit for a false alarm rate $\alpha = 0.05$ and its 95% BC_a confidence interval is shown.

Again, the bootstrap control limit value 6.75 along with its BC_a confidence interval $[6.08, 8.27]$ is shown. In comparison, the corresponding basic

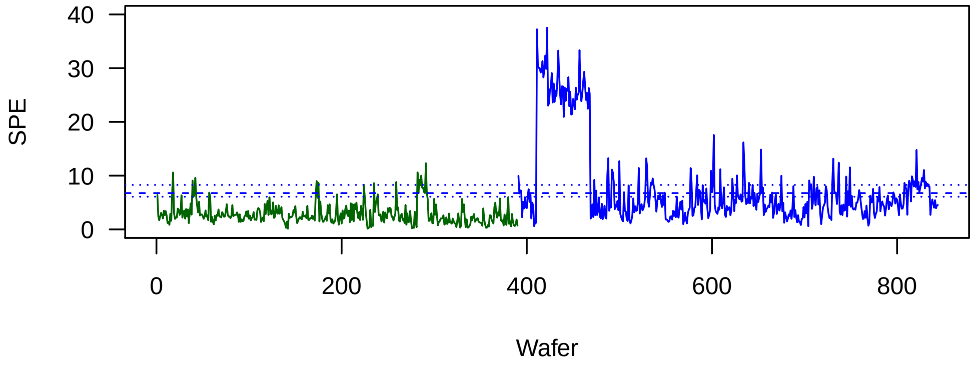


Figure 6.5: SPE chart of $I = 390$ reference wafers (green) and $I_{test} = 454$ test wafers (blue) along with the bootstrapped control limit (dashed) and its BC_a confidence interval (dotted).

bootstrap interval is $[5.47, 7.46]$.

The detected faults can now be diagnosed to find out the groups of variables that contribute to the abnormal behaviour. This is done by inspecting the block statistics. Figure 6.6 exemplarily shows the robust block T^2 charts for all three variable blocks along with the respective bootstrap UCLs ($\alpha = 0.05$).

Clearly, the magnetic field failure is mainly observable in the RF power block. This coincides with the nature of the fault as the RF power unit had to be adapted due to the breakdown of the magnetic field. Helium and pressure unit are not or only slightly affected. However, the pressure faults that also occured in the considered time frame are identified correctly as the related wafers are mainly observable as single excursions in the pressure group.

The faults can be further analysed by inspecting the on-line monitoring charts. These can give information on the exact time point of the occurence or start of a fault during the processing of an affected wafer. Figure 6.7 exemplarily shows robust on-line monitoring T^2 charts for three different test wafers out of the considered time frame.

The first wafer was a normal production run with no observed fault and an overall T^2 of the super scores of 8.10. Thus, from the beginning to the end of its processing time it shows no sign of abnormal behaviour. Only two insignificant exceedings of the UCL at seconds 10 and 20 can be observed. The second wafer with an out-of-control super T^2 value of 352.72 was affected by the magnetic field failure. Right from the start this wafer is out-of-control. Thus, with the proposed model implemented, the failure could have been detected at a very early stage of wafer processing. Information like this can lead to fewer scrap wafers or reworkings. The third wafer is related to the pressure failures. Again, it is out-of-control already from the beginning of its processing. Here, the actual fault seems to have occured during the first 30 seconds of the observed processing time. Afterwards, the T^2 value settles at a steady but still out-of-control level.

A simplification of the on-line monitoring can be achieved by summarizing the produced information, e.g., by computing means or medians over 5 or 10 seconds. This would result in values of the monitoring statistics every 5 or 10 seconds which is still considered sufficient for the task of monitoring the actual wafer processing. Furthermore, this would also smooth the resulting monitoring statistics.

In summary, the occured faults can be detected and diagnosed in every data dimension, i.e., our model detects which wafers are affected, which variable groups contribute and at what processing time the abnormality started or occured.

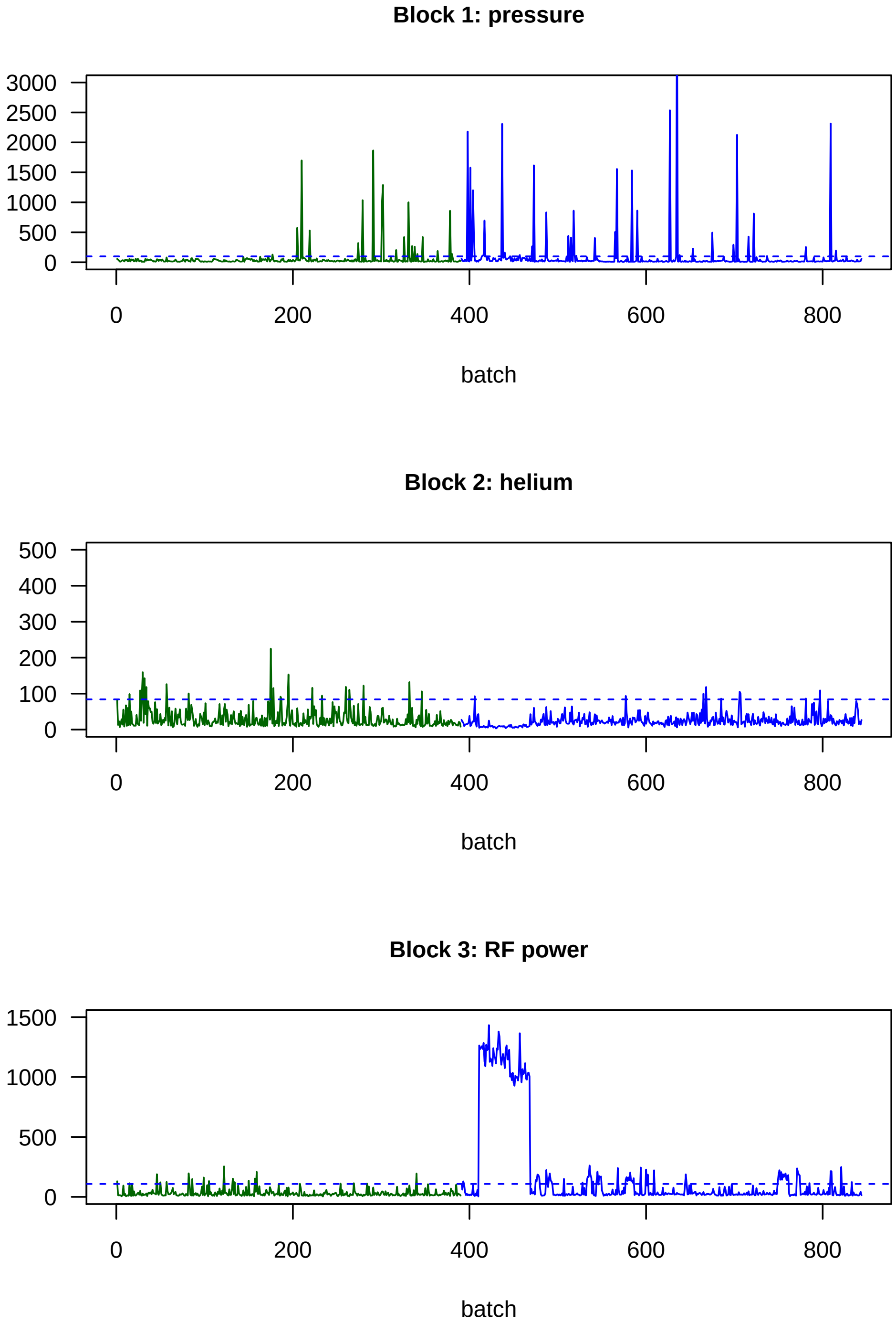
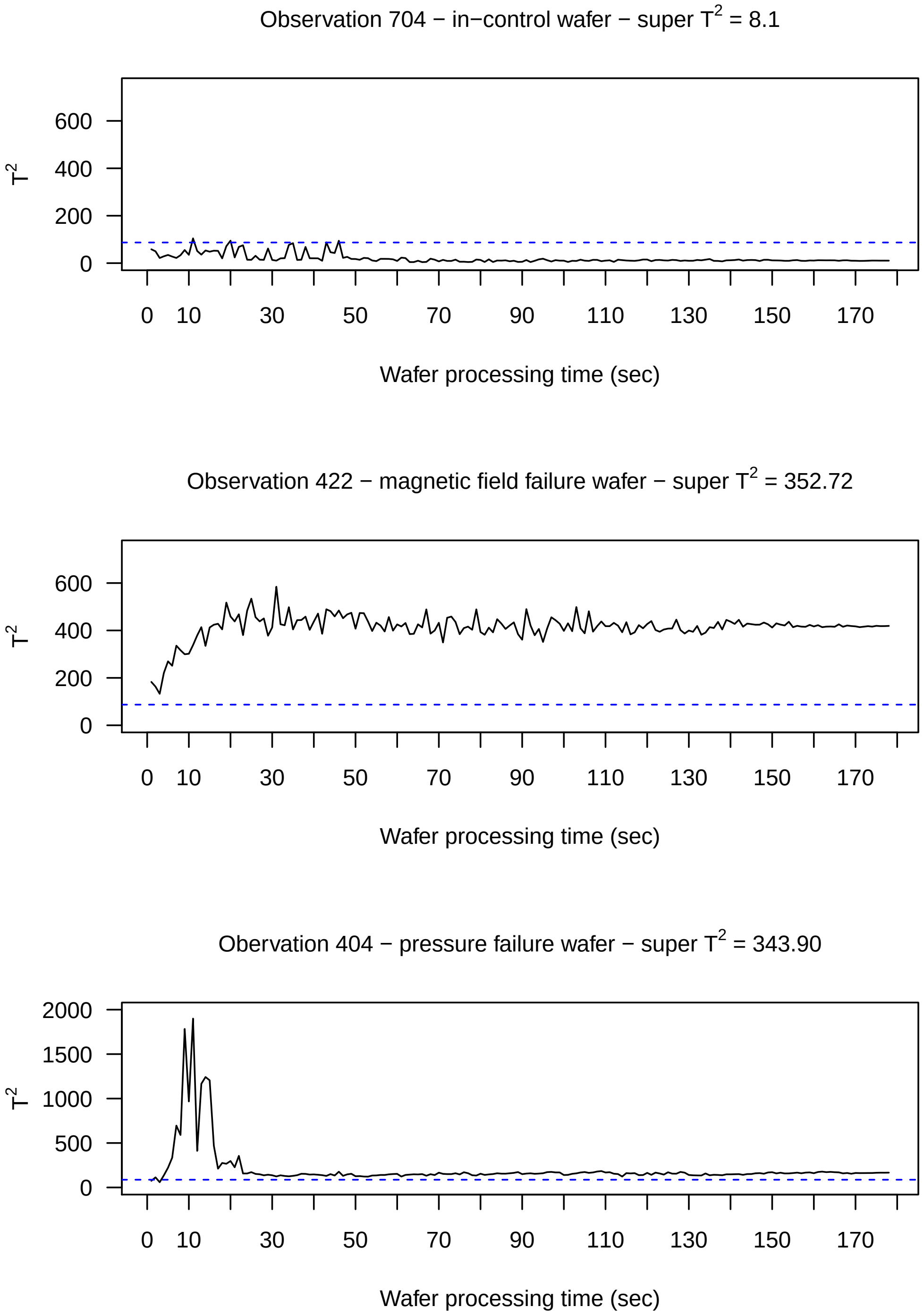


Figure 6.6: Robust T^2 charts of robust KPCA block scores along with the respective bootstrapped control limits.



Observation 704 − in−control wafer − super T^2 = 8.1

Observation 422 − magnetic field failure wafer − super T^2 = 352.72

Obervation 404 − pressure failure wafer − super T^2 = 343.90

Figure 6.7: On-line monitoring T^2 charts for three wafers over their processing time along with the upper control limit (dashed).

6.4.1 Comparisons

The number $q = 16$ of extracted PCs of our KPCA model is chosen based on kernel parallel analysis. This is only a guideline. Figure 6.8 shows the resulting robust T^2 charts ($\alpha_{RMCD} = 0.8$) of training and test data for $q = 16$, $2q = 32$ and $3q = 48$ extracted robust principal components ($\alpha_{KPCA} = 0.8$).

Clearly, the robust KPCA models based on $2q = 32$ and $3q = 48$ principal components display the process behaviour in more detail. We also compared the kernel PCA based models to linear PCA based models for both classic and robust approaches. For the linear PCA based models of the same training data, the number of principal components is again determined by parallel analysis. This results in $q_{linear} = 22$ extracted PCs. In general, kernel PCA can potentially utilize more principal components to map structure rather than noise (see Schölkopf et al. (1998), Lee et al. (2004a)). KPCA based on the Gaussian kernel extracts information from an infinitely high-dimensional feature space whereas linear PCA extracts information from the finite dimensional input space. Thus, the number of principal components to extract for KPCA models is larger than that for linear PCA models.

For comparison we use KPCA models with $2q = 32$ PCs and linear PCA models with $q_{linear} = 22$ PCs. Figure 6.9 shows the resulting T^2 charts for classic and robust approaches along with their respective UCLs.

The magnetic field error is captured correctly by all models. However, the proposed robust kernel approach captures the process behaviour the best as it identifies best all the occured faults. When compared to the robust linear PCA based model, the robust kernel PCA approach is able to detect more faults (e.g., around wafer 500). This suggests that the corresponding pressure fault is of nonlinear nature that the linear PCA model fails to capture. Abnormalities that are identified correctly by both robust approaches are better discriminated in the kernel case. This suggests that the robust kernel approach is more likely to identify faulty behaviour.

Overall, the proposed robust kernel PCA approach seems the most suitable for monitoring, fault detection and diagnosis of this plasma etch process.

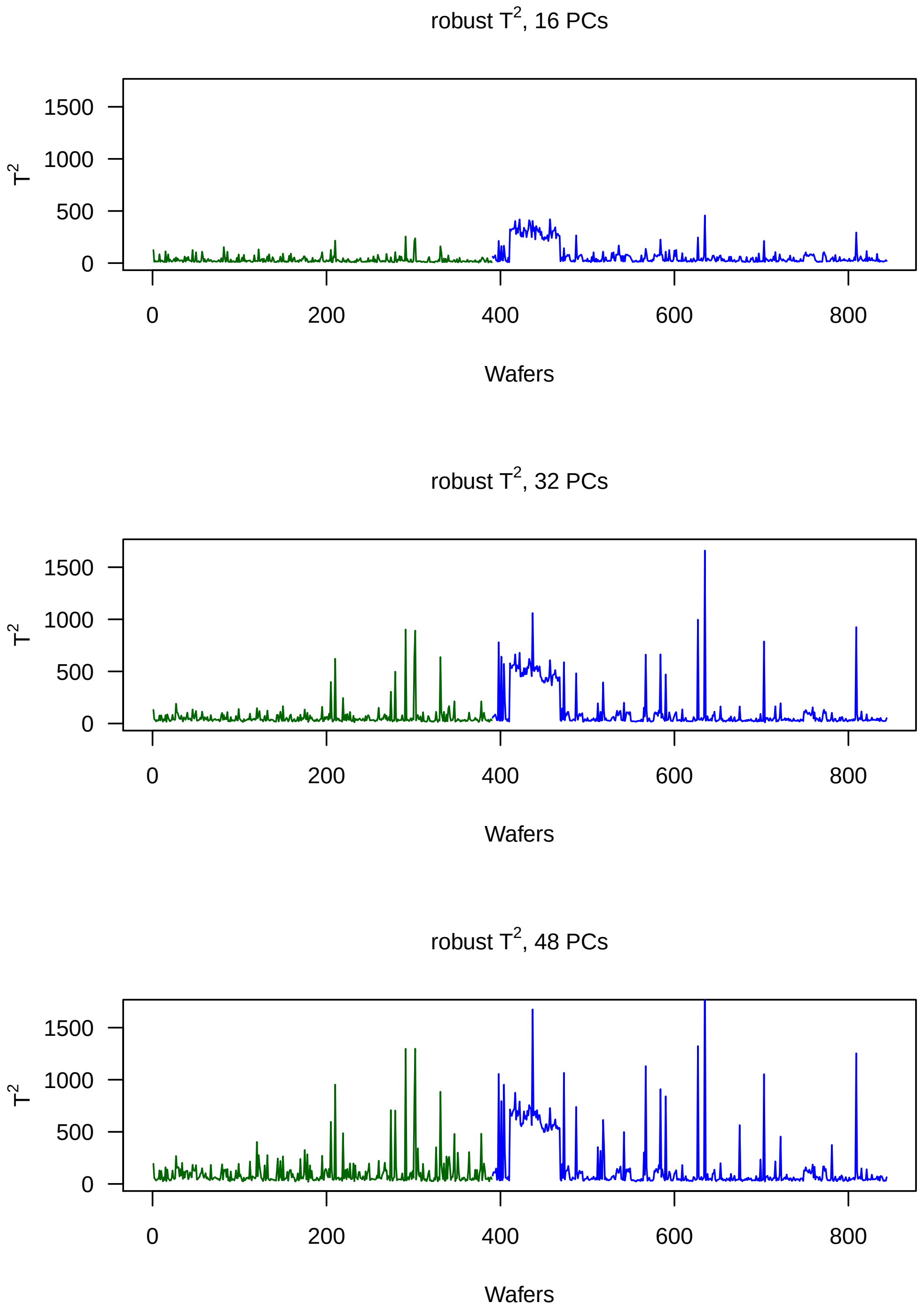


Figure 6.8: Comparison of robust kernel PCA models for different numbers of extracted principal components.

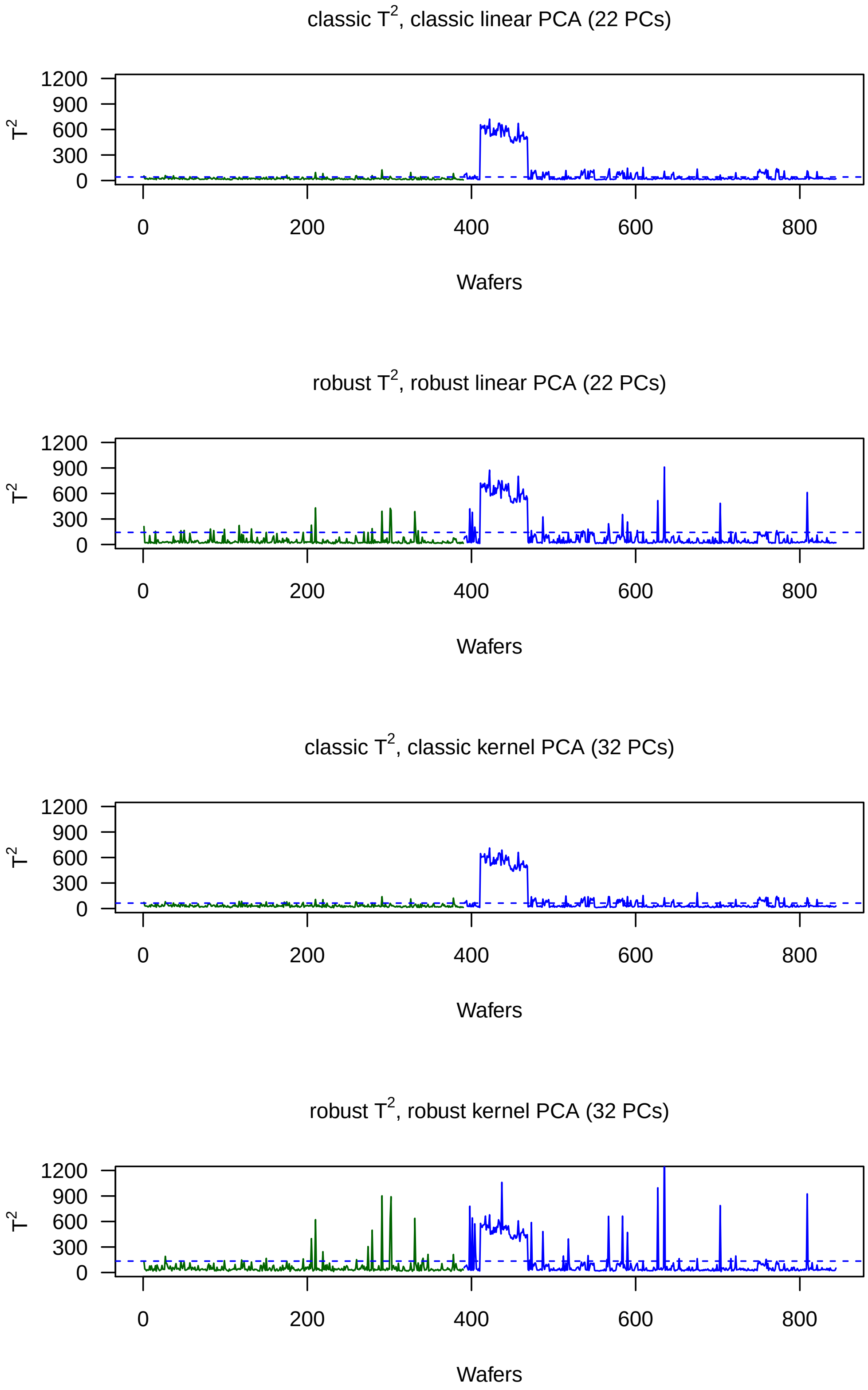


Figure 6.9: Comparison of linear and kernel PCA approaches.

6.5 Summary

We exemplarily applied the proposed robust multi-way multi-block kernel PCA approach to monitor the wafer processing of a plasma etch equipment. The case study is related to a failure of the magnetic field that occured on the equipment in June 2011 at ams AG. During the observed time frame several faults related to the pressure unit of the equipment also occured. The proposed model is able to detect all occured faults correctly. The multi-block approach then allows a diagnosis of the detected faults. The magnetic field failure can be correctly assigned to the RF power unit of the plasma etch equipment whereas the pressure faults can be correctly traced down to the pressure unit. This allows a fault interpretation and an identification of the involved process variable groups. Furthermore, on-line monitoring charts allow the identification of the exact time point of the occurence of each fault during the processing of the affected wafer. This enables an in-depth monitoring of the plasma etch process.

In comparison with non-robust and linear PCA based alternatives, the proposed model performs superior. The robust kernel approach allows the detection of faults of nonlinear nature that would have been overlooked by more traditional approaches. Thus it is most suitable for the sensitive task of monitoring such complex production processes.

The proposed approach can be applied to monitor the health and performance of any semiconductor production equipment that is of batch type, i.e., that produces multi-way wafer data.

In future research, more efficient ways to determine the adequate number of principal components as well as the most appropriate and problem-specific kernel function and its parameters can be examined.

Chapter 7

Multivariate Monitoring of Wafer Acceptance Tests Via Robust T^2

7.1 Motivation

In order to evaluate the quality and functionality of fabricated chips on a wafer the *wafer acceptance test* (WAT) serves as a crucial post-production check in semiconductor manufacturing. For each wafer, a large number of test parameters (WAT parameters) are measured electrically to check specifications and basic chip functionalities. These tests reflect the results of single process steps as well as of the whole manufacturing process and give important information on its long-term stability. Product performance and product yield are usually strongly related to the WAT data. In addition, the tests give information if the chips on a wafer fulfill the specifications as defined by the chip designers. The WAT data also provide a data base for the models used by chip designers to design chip functionality. Such design models are often multivariate, i.e., they consider the correlation structure among the parameters. Chip designers use this information to construct the functioning of a chip and specify operational ranges of parameters for a process variant prior to fabrication. However, standard univariate statistical tools like statistical process control (SPC) charts are used to monitor the univariate performance of WAT parameters after fabrication. These SPC charts do not reflect the fact that WAT parameters are interrelated and correlated, i.e., do not behave independently of each other. This makes the overall WAT data inherently multivariate. Classical univariate monitoring approaches fail to capture the multivariate structure. For example, deviations due to changes in the correlation structure can be missed. However, changes in the correlation can have serious effects on the functionality of a chip. Thus, even if all parameters are within their respective univariate SPC limits, non-observed abnormal multivariate relationships of the WAT parameters can cause significant yield losses of a product or even a complete malfunction.

WAT data monitoring or wafer test data analysis and modeling has been considered in the past (see, e.g., Fan et al. (2000), Skinner et al. (2002)). However, the monitoring of changes in the multivariate relationship structure was never considered explicitely. However, in order to better detect potential yield problems, gain more detailed insight into wafer fabrication results and ensure higher production quality information on the relationship structure of WAT data is crucial.

In order to take this structure into account and to identify outliers due to flawed relationships multivariate control charts have to be implemented. We propose the use of a robust Hotelling's T^2 statistics to monitor multivariate deviations from a reference situation of historic WAT data. Robust estimation of the T^2 statistic is crucial due to various sources of variation coming from the actual manufacturing process, measurement site-to-site variation, product-to-product variation, different measurement equipments or the presence of measurement errors. Fault diagnosis is achieved by applying an MYT decomposition of the robust T^2 values.

Furthermore, the T^2 statistic can be used not only as a tool to identify single abnormalities of WAT data but also to monitor the multivariate development of the wafer acceptance tests over time. An observed process drift or shift does not necessarily have to mean a change to the worse. It can rather be a confirmation that implemented process changes have an actual effect on the data or that the overall process behaviour developed or even improved over time. Such information is considered highly valuable for process engineers.

Multivariate relationships can be of bivariate order, i.e., correlations between two variables, and also of higher order. However, in our practical experience with WAT parameter relationships and the related T^2 analysis along with the MYT decomposition, trivariate or higher order relationships among variables were never observed. Abnormal behaviour that could be traced back to a flawed multivariate relationship was always observed to be of a bivariate nature. Thus, for monitoring WAT data it suffices to narrow our consideration of multivariate behaviour down to the case of bivariate relationships.

For multivariate monitoring models to be useful and applicable in a semiconductor production environment we constructed a novel software solution based on R and *TIBCO Spotfire*. The proposed implementation is able to connect advanced statistical modeling in the background with an easy-to-use and familiar user interface for process engineers.

7.2 WAT Data

The common approach to performance monitoring of a semiconductor manufacturing process is to fabricate a set of test structures on a wafer appropriate to give feedback on the process behaviour. These so called scribe line monitors (SLMs) cover the main process parameters and result in a few hundred parameters to be monitored. After the manufacturing process these parameters are electrically measured on pre-defined patterns on each wafer to generate the data set of WAT parameters.

WAT parameters can be grouped according to their physical relations and functional similarities. These groups are called *devices*. Devices describe basic functionalities of a chip. Chip designers use the devices as building blocks when constructing a process variant for a new product. Thus, production process variants differ in terms of the devices they are composed of, i.e., functionalities that are required for the application of a chip. For multivariate monitoring via Hotelling's T^2 several process variants with different sets of available historical samples and variables were considered. Models were constructed for process variants with the same devices for 0.35μ CMOS (complementary metal oxide semiconductor) and high-voltage CMOS process technologies . Mostly, a process variant is applicable for more than one product type, i.e., chips with different final applications can share the same process variant when manufactured.

Out of the various constructed models we exemplarily discuss a model for one process variant of 0.35μ CMOS manufacturing technology composed of 12 devices. The process variant is applicable to 10 product types. However, only the four most frequently manufactured products are considered for modelling. The resulting initial data set consists of

- $n = 1430$ measurements from 286 wafers
- $p_0 = 40$ WAT parameters.

The measurements are given in chronological order. Only in-control wafers with all parameters being within their respective univariate control limits are considered. The parameters are measured on three different test equipments at five sites on each wafer. Only full and regular five-site-measurements were considered. One line in the data set represents the multivariate WAT measurement conducted at one site of one wafer. The data was measured between January and August 2012.

Typical WAT parameters are electrical resistances, threshold voltages or film thicknesses. The variables are denoted as $x_1, \ldots, x_{40}$. Due to confidentiality reason, individual measurement vectors, descriptions of parameter meanings or their original operational ranges can not be provided.

7.3 Phase 1: Construction of the Reference Set

7.3.1 Overall Autoscaling

We exemplarily discuss a model for monitoring 40 WAT parameters denoted $x_1, \ldots, x_{40}$ for a single process variant. The variables are autoscaled.

A theoretical nonlinear relationship between x_1 and x_{23} is known. The distribution of x_1 is right-skewed. After a log-transformation of x_1 its distribution becomes symmetric and the relationship with x_{23} becomes linear. Thus, we continue with $\ln(x_1)$ in our model.

Analysis of the correlation structure of the initial data set shows evidence of several colinearities among the WAT parameters, i.e., redundant information. In cooperation with process engineers redundant variables were removed in a stepwise fashion until the corresponding maximum condition number of the correlation matrix was smaller than 30. Table 7.1 shows the order of variable removal and the corresponding condition numbers based on classicaly and robustly autoscaled data, and classical and RMCD correlation estimates. The robust autoscaling is based on the L_1-median for robust multivariate location estimation and the MAD for robust estimation of scale. Figure 7.1 depicts the results graphically.

Analysis of the robust condition number suggests to remove variables x_{32}, x_{22}, x_6 and x_{23} from the initial data set due to their high correlations with other variables. By using the robust RMCD method to estimate the correlation matrix more variables have to be removed. Hence, the robust model becomes simpler. Figure 7.2 shows the corresponding scatterplots.

	initially	Pass 1	Pass 2	Pass 3	Pass 4
removed variable (classic)		x_{32}	x_{11}	x_6	
max. classical condition	35.32	34.41	33.58	29.46	
removed variable (robust)		x_{32}	x_{22}	x_6	x_{23}
max. robust condition	41.06	40.08	32.37	31.78	21.69

Table 7.1: Condition number of the classically and robustly (RMCD) estimated correlation matrix under step-by-step removal of variables with highest absolute correlation. The variable pair (x_{28}, x_{32}) has the highest absolute value of both sample correlation (0.9274) and robust RMCD-based correlation (0.9302).

However, further analysis shows evidence that product type and measurement equipment are influential for some WAT variables. After robust autoscaling using the L_1-median and the MAD the influence becomes obvious. Figure 7.3 depicts the evidence exemplarily for robustly autoscaled variables x_{10} and x_{24}.

Differences in different product types are common. The boxplots of x_{24} on the right panel of figure 7.3 shows its overall bimodal distribution. While measurement equipments M_1 and M_2 are similar, equipment M_3 behaves differently. This is also confirmed by process engineers and is due to M_3 being of newer technology than M_1 and M_2.

After removing variables x_{32}, x_{22}, x_6 and x_{23} as suggested by the robust condition number analysis, T^2 statistics using classical as well as RMCD estimates for multivariate location and scatter were computed. Both statistics are shown in figure 7.4.

Only the robust T^2 chart shows a shift of the process between measurement 750 and 850. Further analysis of the data affected by the shift shows that it is only visible for a certain combination of product type and measurement equipment.

7.3.2 Combination-Based Autoscaling

There are 7 different combinations of product type and measurement equipment in the data. Table 7.2 gives an overview of the number of observations for each combination.

Figure 7.5 shows the robust T^2 chart colored by these combinations.

Observations of product P_3 on measurement equipment M_1 are clearly

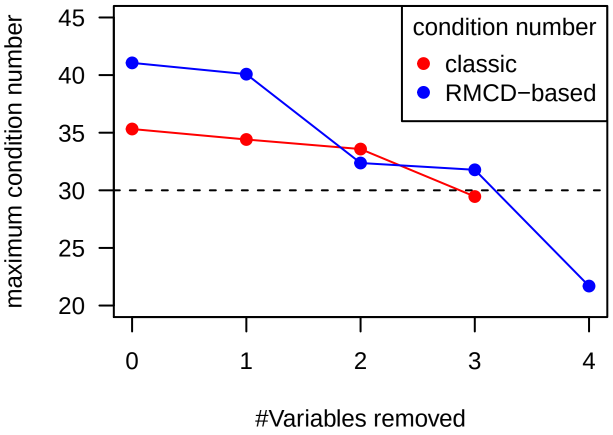


Figure 7.1: Graphical depiction of the change in the condition number of the classically and robustly (RMCD) estimated correlation matrix under removal of variables with highest absolute correlation. The threshold for a acceptable condition number of 30 is proposed in the literature.

different from the remaining observations in terms of T^2. By performing robust autoscaling of the initial data separately for each product-equipment combination, this effect can be filtered out. Furthermore, the combination-based autoscaling results in a covariance matrix based on all 40 variables with a smaller maximum robust condition number of 18.47. Thus, the T^2 statistic can be computed based on all 40 variables. A histogram of the resulting T^2 statistic is shown in figure 7.6.

As depicted in figure 7.6, the assumption of the beta distribution does not hold for the T^2 statistic based on robust estimates. Thus, in order to determine the upper control limit the parametric assumption has to be dismissed in favour of nonparametric methods. We use the nonparametric bootstrapping approach as discussed in chapter 2. The resulting T^2 control chart is shown in figure 7.7.

The resulting UCL based on 5000 bootstrap resamples is 175.70 for a false alarm rate $\alpha = 0.01$. The associated BC_a bootstrap confidence interval is [157.52, 192.49]. In comparison, the associated basic bootstrap interval is [160.77, 194.65].

Product type	Measurement equipment	measurements
P_1	M_1	125
P_2	M_1	470
P_2	M_2	245
P_2	M_3	125
P_3	M_1	120
P_3	M_3	125
P_4	M_3	220

Table 7.2: Number of observations for all 7 combinations of product type and measurement equipment in the data set.

In summary, only the T^2 statistic based on robust estimates and autoscaling based on each product-equipment combination reflects the normal operating condition of the WAT data without the undesired influence of different products or measurement equipments. The underlying data set of $n = 1430$ observations and $p = 40$ variables is now used as reference data set for further analysis.

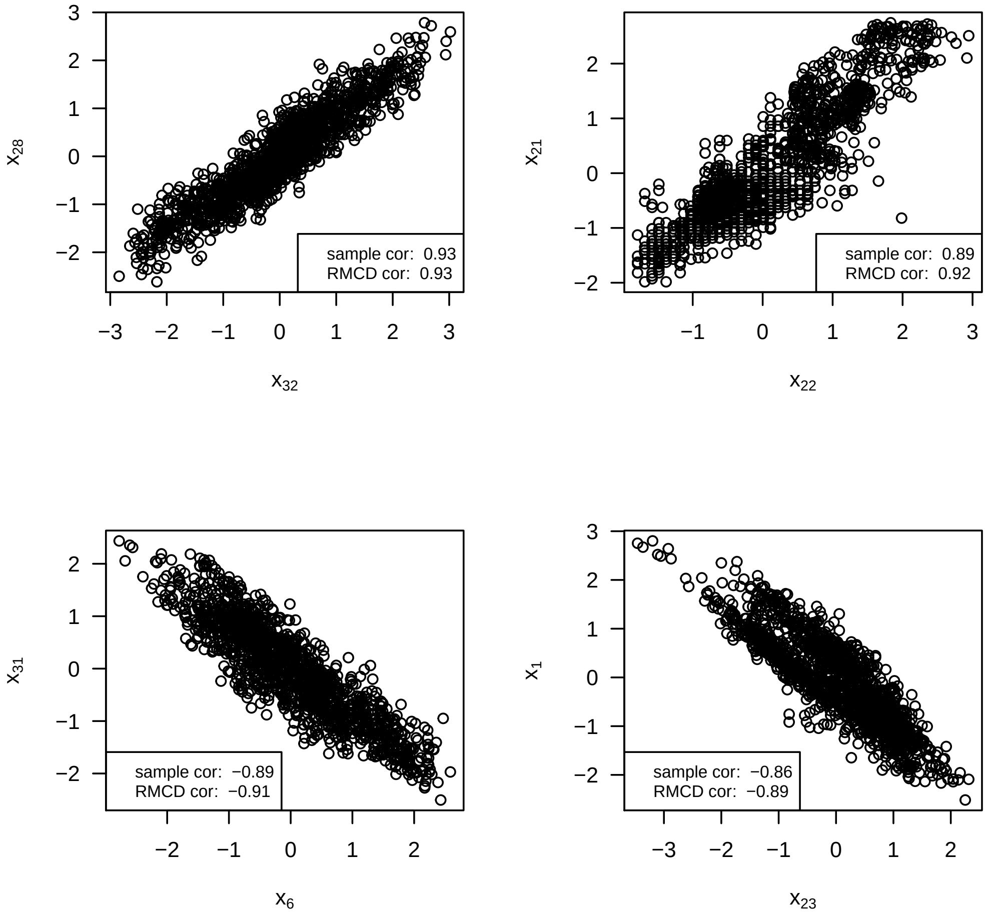


Figure 7.2: Scatterplots of variables with the highest correlations in the data set. After discussions with process engineers, variables x_{32}, x_{22}, x_6 and x_{23} are removed.

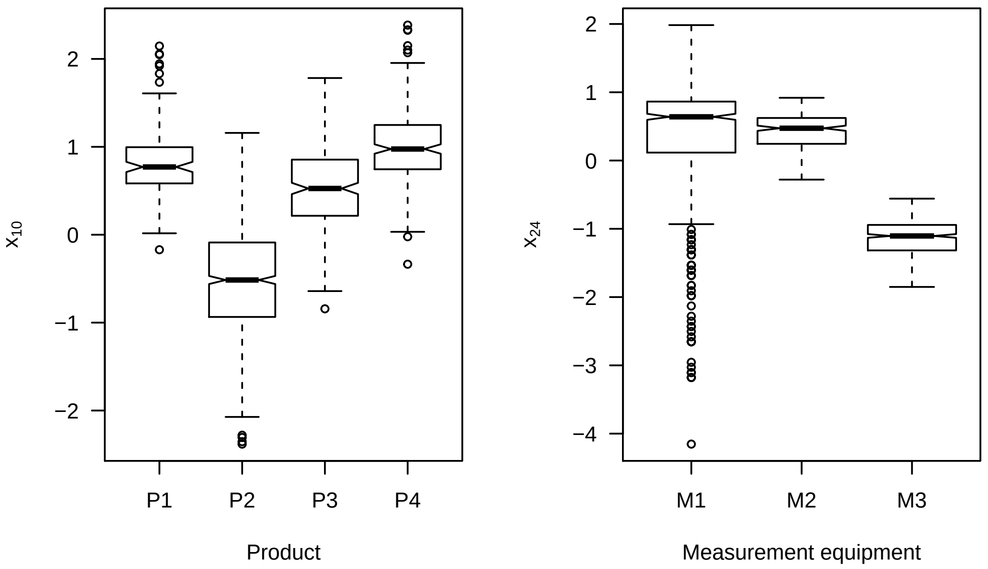


Figure 7.3: Graphical depiction of the influence of product type and measurement equipment on WAT variables. Boxplots of robustly autoscaled x_{24} for each measurement equipment (left panel) and boxplots of robustly autoscaled x_{10} for each product type (right panel).

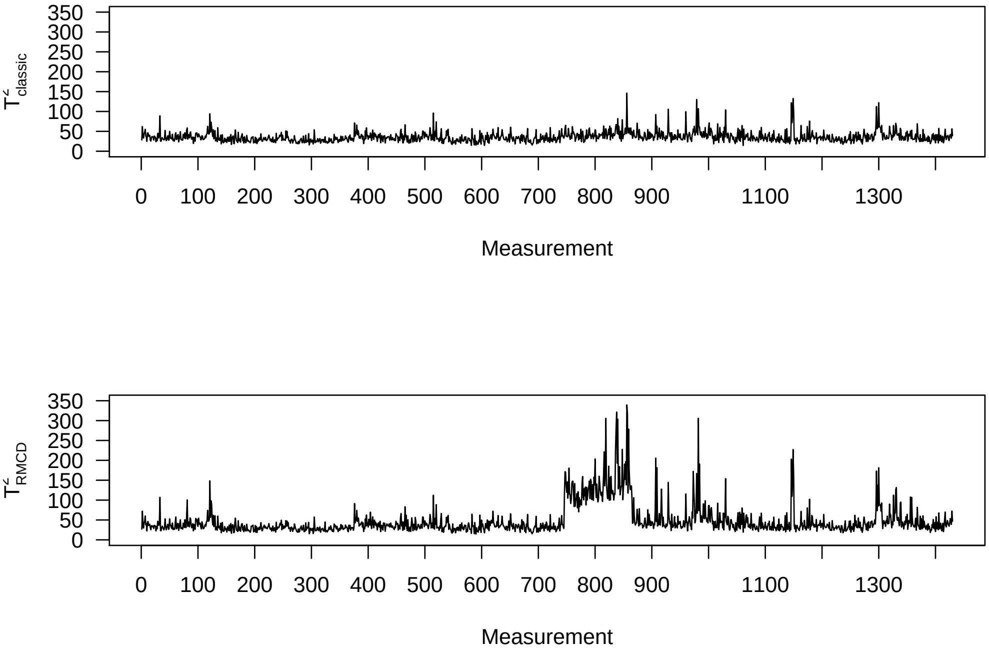


Figure 7.4: T^2 charts based on classical sample estimation (above) and robust RMCD-based estimation of the covariance matrix.

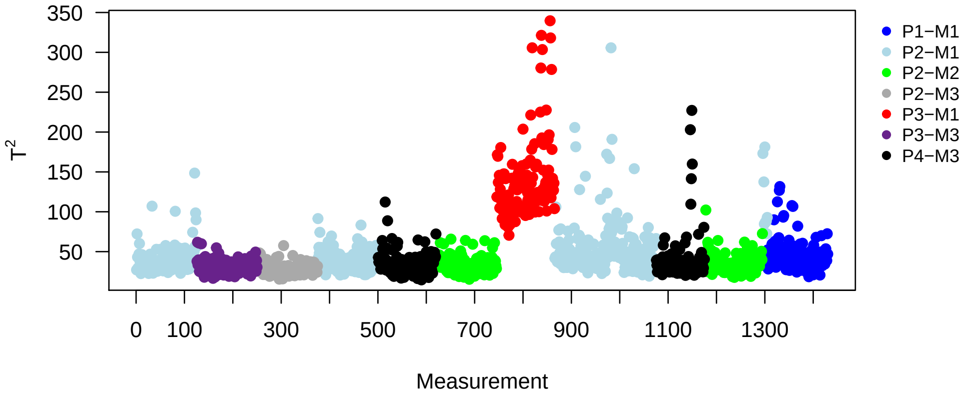


Figure 7.5: Robust T^2 values colored by product-measurement equipment combinations.

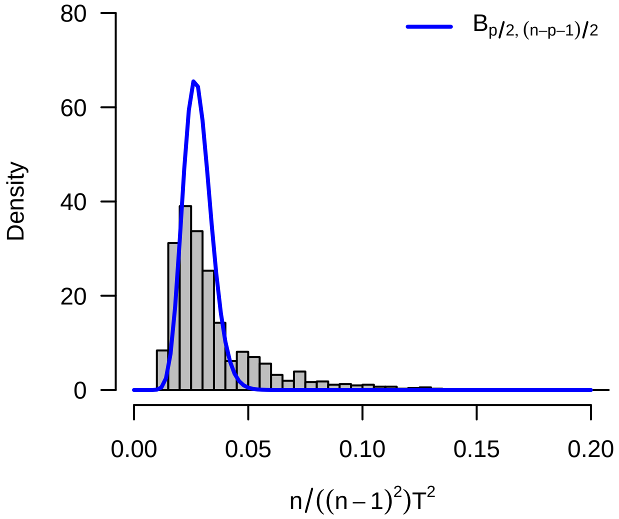


Figure 7.6: Histogram of the robust T^2 statistic based on data autoscaled separately for each product-equipment combination and RMCD estimation of the covariance matrix. The statistic is scaled with $n/(n-1)^2$ to compare it to the beta distribution $B_{(p/2,(n-p-1)/2)}$ (blue line).

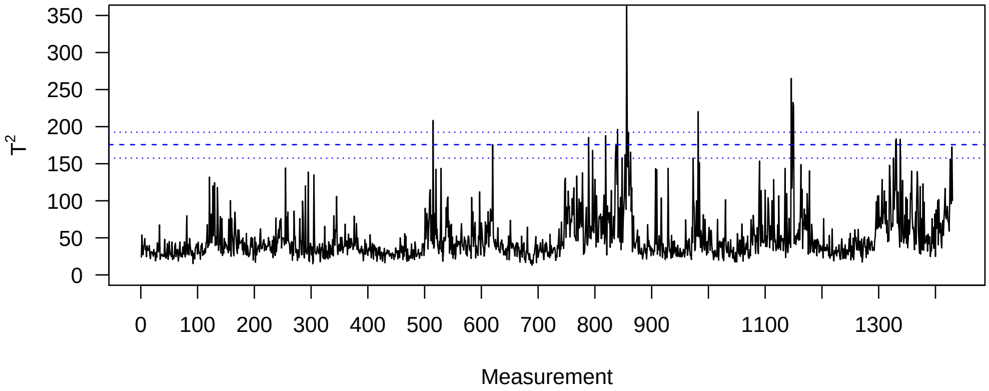


Figure 7.7: T^2 control chart based on data autoscaled separately for each product-equipment combination and RMCD estimation of the covariance matrix. The UCL as determined via bootstrapping is shown as dashed blue line, the corresponding BC_a confidence limits are shown as dotted blue lines.

7.4 Phase 2: Test Data Results

The constructed reference model is tested using an independent test data set (phase 2). The test data set consists of $n_{test} = 360$ measurements conducted on wafers produced between January and May 2014 based on the same 0.35μ CMOS process technology variant and the same $p = 40$ variables as the reference model. In the considered time frame products P_2 and P_3 were manufactured and measured on equipments M_1, M_2 and M_3.

The data is autoscaled separately for each observed product-equipment combination using the robust estimates determined from the reference data set. A T^2 control chart of the test data set based on the robust reference covariance matrix is shown in figure 7.8.

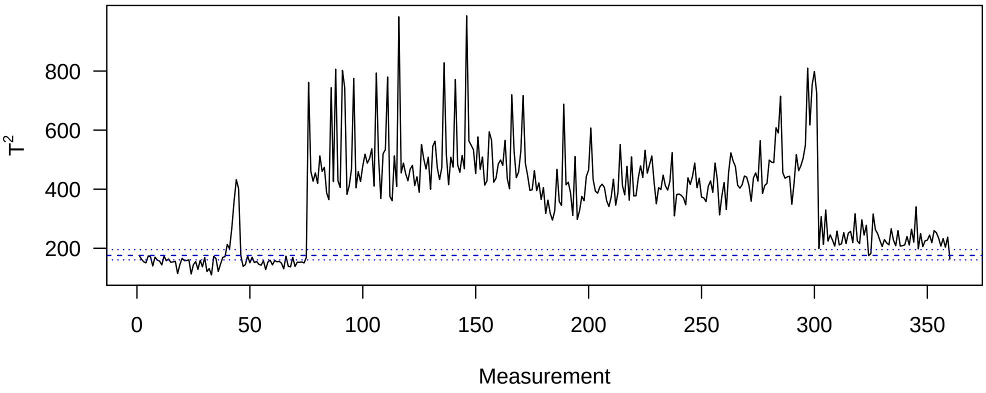


Figure 7.8: Robust T^2 values for the test data set.

The T^2 chart of the test data shows a first peak between measurement 40 and 45 and a process shift between observation 76 and 301. Then, the remaining observations remain on a shifted level.

The first peak is formed by 6 measurements. Figure 7.9 shows the robust RMCD-based MYT-decompositions of observations 42 ($T^2_{RMCD} = 268.08$) and 44 ($T^2_{RMCD} = 432.18$) as barplots. The values are relative to the UCL.

Clearly, variables x_{27} and x_{30} are both strongly related to the changed behaviour in both decompositions. For observation 44, x_{24}, x_{12} and the bivariate relationship between x_1 and x_{23} also contribute. Due to the complex correlation structure of the data, several other variables also show more or less contribution. The corresponding scatterplots shown in figure 7.10 confirm the conclusions drawn from the MYT decompositions. In summary, the peak is mainly univariate.

A longer lasting process shift between observation 76 and 301 follows. Figure 7.11 exemplarily shows the MYT decomposition of observation 147.

The decomposition clearly identifies variables x_{30}, x_5 and x_{19} as the main contributors to the abnormality of this observation. Further analysis of the observations affected by the shift shows that x_2 also contributes, again, among others that contribute more or less due to the underlying relationship structure. Figure 7.12 shows the associated scatterplots. In all variables identified by the MYT decomposition univariate shifts in the test data are clearly observable.

The remaining data from observation 301 onwards still show changed behaviour. While minor abnormalities in x_{30} and x_5 can still be identified, the main drivers are often abnormalities in correlations among variables. Figure 7.13 shows the MYT decomposition of observation 321.

Variables x_{21} and x_4 are identified as having a correlation abnormality. While their values are acceptable in univariate terms, their bivariate relationship is out of tolerance as both conditional terms signal in terms of T^2. Distances to the respective regression lines are too large given the reference situation, i.e., observations on either variable are not where they are expected to be relative to the position of the other variable. Figure 7.14 depicts the abnormality.

Further analysis identifies more observations where correlation problems are the main contributors to an out-of-control T^2 value. Thus, a robust T^2 statistic along with the corresponding robust MYT decomposition seems to be a reliable tool to identify these measurements as abnormal.

Furthermore, the T^2 analysis gives valuable information on the change of the multivariate WAT parameter behaviour between the reference set measured in 2012 and the test data set measured in 2014.

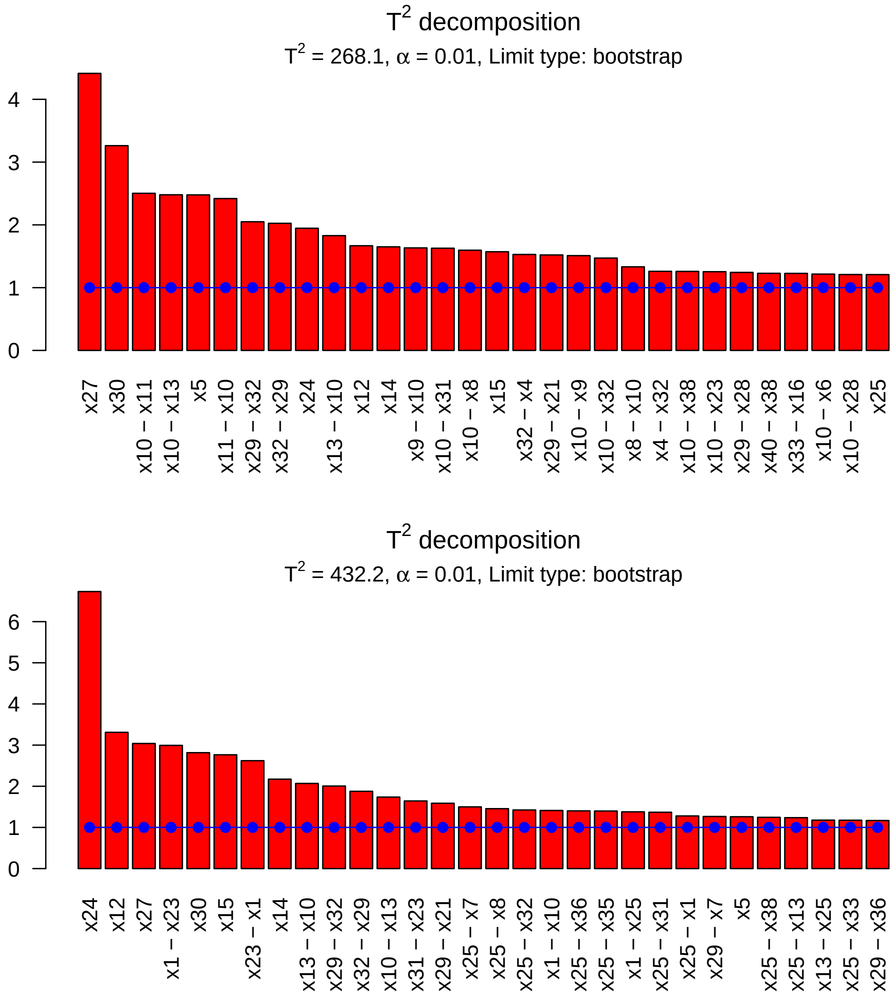


Figure 7.9: Barplots of the MYT decomposition of observations 42 (above) and 44 (below). For each term, the values are divided by the corresponding control limit for better comparability (false alarm rate $\alpha = 0.01$).

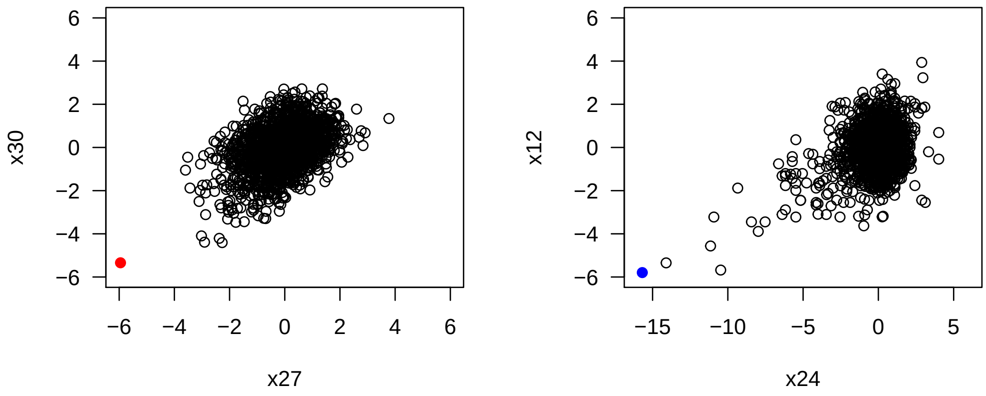


Figure 7.10: x_{27} versus x_{30} from the reference data set along with test observation 42 in red (left panel) and reference data variables x_{24} versus x_{12} and test observation 44 in blue (right panel).

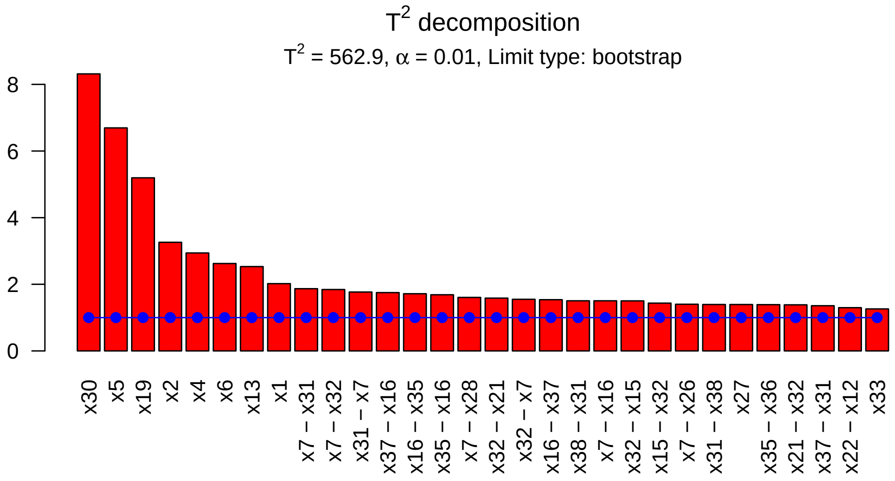


Figure 7.11: Barplot of the MYT decomposition of observation 147. For each term, the values are divided by the corresponding control limit for better comparability (false alarm rate $\alpha = 0.01$).

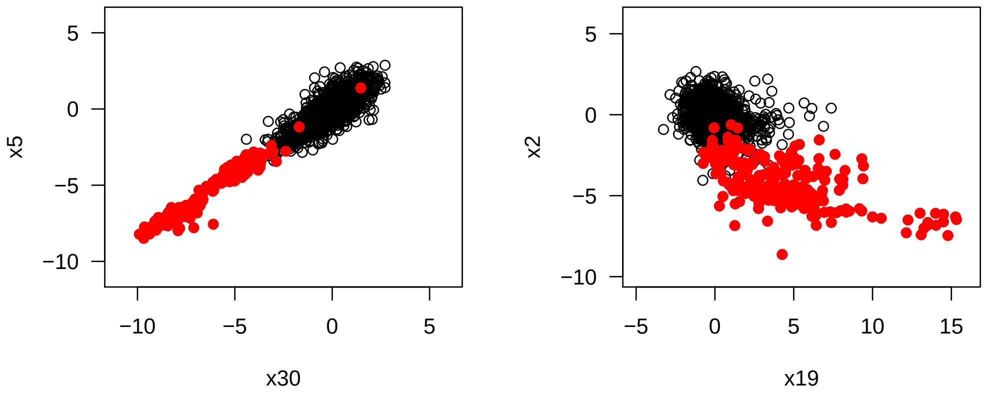


Figure 7.12: x_{30} versus x_5 (left panel) and x_{19} versus x_2 (right panel) from the reference data set along with test observation 76 to 301 (red).

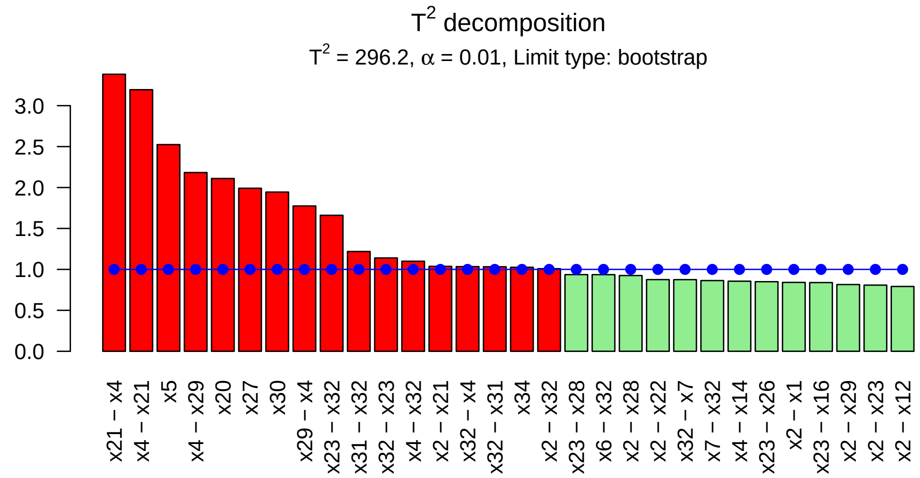


Figure 7.13: Barplot of the MYT decomposition of observation 321. For each term, the values are divided by the corresponding control limit for better comparability (false alarm rate $\alpha = 0.01$).

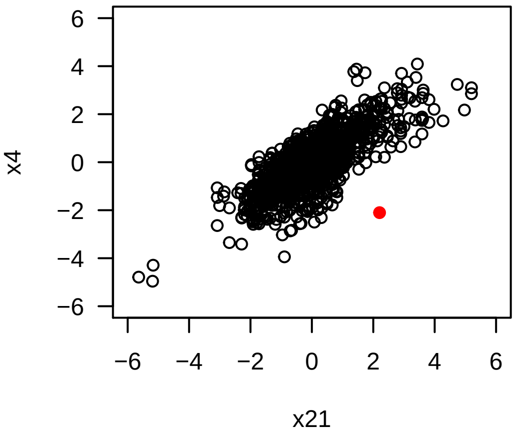
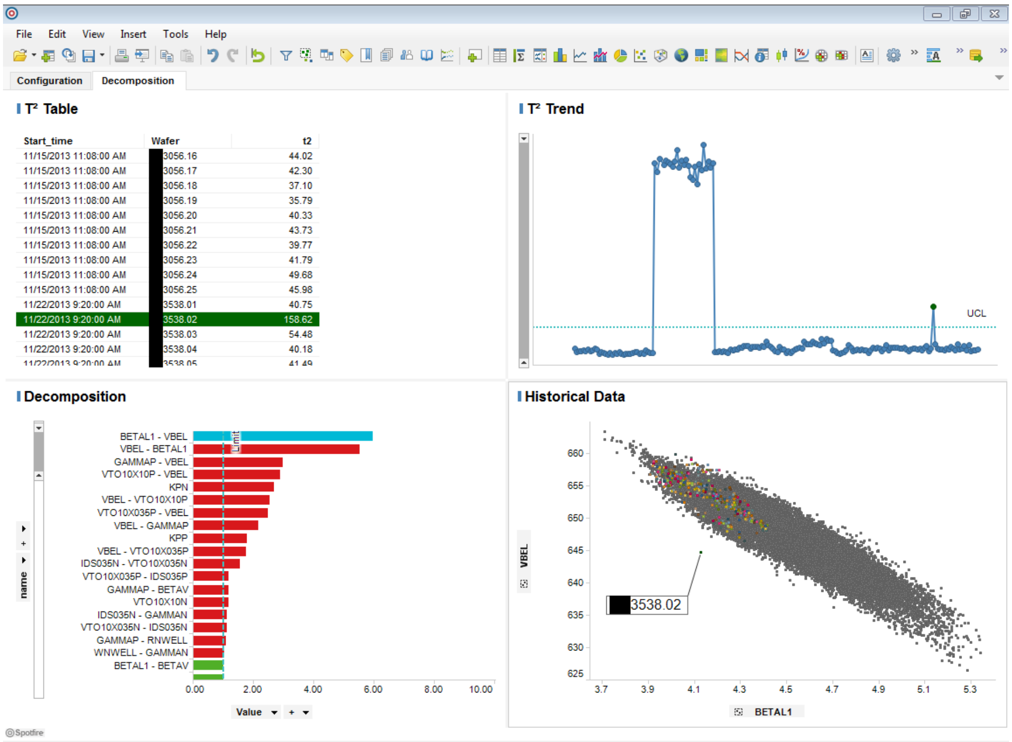


Figure 7.14: Scatterplot of reference data variables x_{21} versus x_4 along with test observation 321 (red).

7.5 Implementation

For the presented multivariate monitoring approach to be useful and applicable in a fast and simple way an appropriate software environment was implemented. The implementation is able to connect the statistical modelling in the background with an easy-to-use graphical user interface (GUI). This allows process engineers to apply advanced statistical models to a chosen set of new test data in a familiar software environment. The results offer new and advanced insights into wafer acceptance tests and allow an improved assessment of the quality of each wafer.

Figure 7.15 shows a screenshot of the created Spotfire GUI.



Start_time	Wafer	t2
11/15/2013 11:08:00 AM	[redacted]3056.16	44.02
11/15/2013 11:08:00 AM	[redacted]3056.17	42.30
11/15/2013 11:08:00 AM	[redacted]3056.18	37.10
11/15/2013 11:08:00 AM	[redacted]3056.19	35.79
11/15/2013 11:08:00 AM	[redacted]3056.20	40.33
11/15/2013 11:08:00 AM	[redacted]3056.21	43.73
11/15/2013 11:08:00 AM	[redacted]3056.22	39.77
11/15/2013 11:08:00 AM	[redacted]3056.23	41.79
11/15/2013 11:08:00 AM	[redacted]3056.24	49.68
11/15/2013 11:08:00 AM	[redacted]3056.25	45.98
11/22/2013 9:20:00 AM	[redacted]3538.01	40.75
11/22/2013 9:20:00 AM	[redacted]3538.02	158.62
11/22/2013 9:20:00 AM	[redacted]3538.03	54.48
11/22/2013 9:20:00 AM	[redacted]3538.04	40.18
11/22/2013 9:20:00 AM	[redacted]3538.05	41.49

Figure 7.15: Screenshot of the TIBCO Spotfire user interface for presenting results of multivariate monitoring models of wafer acceptance tests.

The implementation runs all model computations in the background using R. Then, the results are presented in a novel graphical and interpretable user interface created in the commercially available software *TIBCO Spotfire*. This is enabled through the possibility to connect Spotfire with R and Spotfire's ability to interpret R code.

In a first step a data selection mask is shown where the user can load WAT data from the data base for a specified process variant and time frame. Then the multivariate monitoring model corresponding to the data selection is automatically applied in the background. When the application is finished the results are presented in the user interface shown in figure 7.15.

The upper left panel of this interface shows logistic details of the loaded wafer samples and measurements along with their computed robust T^2 values. In the upper right panel the T^2 control chart is shown. The respective UCL is determined via the nonparametric bootstrapping approach as discussed in chapter 2. The false alarm rate in the example is $\alpha = 0.01$. Single outlying wafers as well as process shifts and drifts can be detected in a fast and simple way. By clicking on a measurement (upper left panel) or a T^2 value (upper right panel) of interest its MYT-decomposition is shown as bar chart along with respective control limits in the bottom left panel. It works in real-time and immediately informs which single WAT parameters or correlations among WAT parameters are responsible for an out-of-control T^2 value. Finally, by clicking on a bar of interest in the MYT decomposition chart, the corresponding scatterplot of the involved WAT parameters is shown in the bottom right panel where the chosen measurement is automatically highlighted. Data from the underlying reference data set used to construct the T^2 is shown as dark gray points whereas the new test data points are highlighted in different colors. This allows a more detailed graphical inspection of single out-of-control measurements or process shifts of any kind.

The presented information is considered novel and highly valuable for process engineers. The multivariate approach allows a more detailed evaluation of process shifts and possibly earlier detection of drifts that can lead to future parameter shifts. Faults in the relationships among WAT parameters can be evaluated and important correlations can be monitored. Violations in the correlations that can lead to significant yield loss can be catched for the first time. This also allows an improved assessment of the agreement of actual production with the multivariate specifications formulated by the chip designers.

As Spotfire is a popular tool for data analysis in semiconductor fabrication and especially at ams AG, the implementation can also serve as a guideline for future applications of statistical analysis and modelling for various types of wafer fabrication data.

7.5.1 Implementation in R

For the implementation the models are first constructed in R. Then an R workspace is created that contains the robust estimation of the mean vector, i.e., m, the robust estimation of the covariance matrix, i.e., S, the determined UCL (ucl) as well as the reference data set as data frame xtrain. Moreover, the respective false alarm rate is stored as alpha and the model name is stored as model.

Furthermore, the MYT decomposition was implemented as R function myt.dcmp(). For the ith test observation vector out of a $I_{\text{test}} \times p$ data frame xtest of independent test data it is called by

```
myt.dcmp(hds=xtrain, center=m, cov=S, test.vector=xtest[i,],
         limits=ucl.list, plot.it=TRUE)
```

for a reference data set (hds) xtrain with mean vector estimate m and covariance estimate S. The logical argument plot.it then determines if the result of the MYT decomposition for test observation xtest[i,] is plotted as barplot. For plot.it=FALSE, only the values are computed and no plot is generated.

The argument limits takes a list containing control limits for all unconditional as well as conditional terms. In order to generate these limits, a function t2limit.gen() has been written and called beforehand via

```
ucl.list <- t2limit.gen(hds=xtrain, alpha=alpha, center=m,
                        cov=S, type="bootstrap")
```

for a fixed false alarm rate alpha. The argument type specifies the type of UCL determination. Here, "bootstrap" leads to UCL computation based on nonparametric bootstrapping for the corresponding quantile of the T^2 values based on m and S. Other options are "mvn" for UCL determination based on the assumption of multivariate normality of xtrain and "chebyshev" for a distribution-free UCL determination based on Chebyshev's inequality.

The generation of ucl.list can take several minutes of computation time depending on the chosen type option. However, this object is generated once and also saved in the R workspace. The computation of the actual MYT decomposition values based on the generated limits can then performed within seconds.

Then, TIBCO Spotfire uses its integrated analytic engine *TIBCO Enterprise Runtime for R (TERR)* to interpret the `R` workspace. This way, the constructed models can be deployed to TERR in the usual `R` format for model objects. Within TERR, a wrapper has been created that calls the functions `mahalanobis()` and `myt.dcmp()` with the option `plot.it=FALSE` on new test observations. The resulting T^2 value as well as the results of the MYT decomposition are passed to the created Spotfire user interface shown in figure 7.15 to create the graphical output.

7.6 Summary

We applied a robust Hotelling's T^2 statistic to assess the outcome of wafer acceptance tests in a multivariate way. In contrast to traditional univariate checks, the multivariate approach allows the identification of problems with correlations among WAT parameters. This is considered crucial as specification limits and operational scope of WAT parameters are initially designed based on their relationship structure. Due to a variety of measurement equipments and product types as well as in-line production variation, the use of robust statistics is essential to adequately estimate the reference situation to which the WAT measurements are compared via T^2. By using the MYT decomposition based on robust estimates of center and covariance complete information on the variables or variable relationships involved in an out-of-control T^2 value can be given and abnormality profiles can be constructed.

A T^2 model was constructed based on a reference data set measured in 2012. In order to test the approach we analysed WAT data measured between January and May 2014. We detected peaks and shifts due to univariate abnormalities as well as out-of-control signals mainly driven by correlation abnormalities. The results also highlight the potential of Hotelling's T^2 as a tool to analyse changes and developments in the WAT parameters over time.

The constructed T^2 models for WAT data monitoring were implemented in `R`. The outcome is presented in a novel graphical and interpretable user interface created in *TIBCO Spotfire*. The constructed interface offers process engineers a multivariate assessment of WAT data in a fast and simple way.

Bibliography

Acar, E. and Yener, B. Unsupervised multiway data analysis: a literature survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1): 6–20, 2009.

Alam, M.A. and Fukumizu, K. Hyperparameter selection in kernel principal component analysis. *Journal of Computer Science*, 10(7):1139–1150, 2014.

Alcala, C.F. and Qin, S.J. Reconstruction-based contribution for process monitoring. *Automatica*, 45(7):1593–1600, 2009.

Blue, J.; Gleispach, D.; Roussy, A., and Scheibelhofer, P. Tool condition diagnosis with a recipe-independent hierarchical monitoring scheme. *IEEE Transactions on Semiconductor Manufacturing*, 26(1):82–91, 2013.

Box, G.E.P. Some theorems on quadratic forms applied in the study of analysis of variance problems: effect of inequality of variance in one-way classification. *The Annals of Mathematical Statistics*, 25:290–302, 1954.

Bro, R. PARAFAC. tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.

Butler, R.W.; Davies, P.L., and Jhun, M. Asymptotics for the minimum covariance determinant estimator. *The Annals of Statistics*, 21(3):1385–1400, 1993.

Canty, A. and Ripley, B. *boot: Bootstrap R (S-Plus) Functions*, 2014. R package version 1.3-11.

Canty, A.J. Resampling methods in R: the boot packages. *R News*, 2(3):2–7, 2002.

Chen, A. and Blue, J. Recipe-independent health indicator for tool predictive maintenance and fault diagnosis. *IEEE Transactions on Semiconductor Manufacturing*, 22(4):522–535, 2009.

Chen, G. and McAvoy, T.J. Predictive on-line monitoring of continuous processes. *Journal of Process Control*, 8(5-6):409–420, 1998.

Chenouri, S.; Steiner, S.H., and Variyath, A.M. A multivariate robust control chart for individual observations. *Journal of Quality Technology*, 41(3): 259–271, 2009.

Cherry, G. and Qin, S.J. Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *IEEE Transactions on Semiconductor Manufacturing*, 19(2):159–172, 2006.

Chou, Y.-M.; Mason, R.L., and Young, J.C. The control chart for individual observations from a multivariate non-normal distribution. *Communications in Statistics - Theory and Methods*, 30(8&9):1937–1949, 2001.

Christianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University press, UK, 2000.

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

Croux, C. and Haesbroeck, G. Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71:161–190, 1999.

Croux, C. and Haesbroeck, G. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87:603–618, 2000.

Croux, C. and Ruiz-Gazen, A. A fast algorithm for robust principal components based on projection pursuit. In *Compstat*, pages 211–216. Physica-Verlag HD, 1996.

Croux, C. and Ruiz-Gazen, A. High breakdown estimators for principal components: the projection pursuit approach revisited. *Journal of Multivariate Analysis*, 95:206–226, 2005.

Croux, C.; Filzmoser, P., and Oliveira, M.R. Algorithms for projection-pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225, 2007.

Dahl, K.S.; Piovoso, M.J., and Kosanovich, K.A. Translating third-order data analysis methods to chemical batch processes. *Chemometrics and Intelligent Laboratory Systems*, 46:161–180, 1999.

Davison, A.C. and Hinkley, D.V. *Bootstrap Methods and their Application.* Cambridge University Press, Cambridge, 1997.

Debruyne, M. An outlier map for support vector machine classification. *The Annals of Applied Statistics*, 3(4):1566–1580, 2009.

Debruyne, M. and Verdonck, T. Robust kernel principal component analysis and classification. *Advances in Data Analysis and Classification*, 4(2-3): 151–167, 2010.

Debruyne, M.; Hubert, M., and Van Horebeek, J. Detecting influential observations in kernel PCA. *Computational Statistics and Data Analysis*, 54 (12):3007–3019, 2010.

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

Donoho, D. Breakdown properties of multivariate location estimators. Ph.D. Qualifying paper, Harvard University, 1982.

Efron, B. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7(1):1–26, 1979.

Engelen, S. and Hubert, M. Detecting outlying samples in a PARAFAC model. *Analytica Chimica Acta*, 705(1-2):155–165, 2011.

Fan, C.-M.; Guo, R.-S.; Chang, S.-C., and Wei, C.-S. SHEWMA: an end-of-line SPC scheme using wafer acceptance test data. *IEEE Transactions on Semiconductor Manufacturing*, 13(3):344–358, 2000.

Filzmoser, P. and Todorov, V. Review of robust multivariate statistical methods in high dimension. *Analytica Chimica Acta*, 705(1):2–14, 2011.

Filzmoser, P.; Fritz, H., and Kalcher, K. *pcaPP: Robust PCA by projection pursuit*, 2014. URL `http://CRAN.R-project.org/package=pcaPP`. R package version 1.9-50.

Fritz, H.; Filzmoser, P., and Croux, C. A comparison of algorithms for the multivariate L_1-median. *Computational Statistics*, 27(3):393–410, 2012.

Geladi, P. Analysis of multi-way (multi-mode) data. *Chemometrics and Intelligent Laboratory Systems*, 7:11–30, 1989.

Geladi, P. and Kowalski, B.R. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

Glorfeld, L.W. An improvement on Horn's parallel analysis methodology for selecting the correct number of factors to retain. *Educational and Psychological Measurement*, 55(3):377–393, 1995.

Harshman, R.A. Foundations of the PARAFAC procedure: models and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

Hayderer, G. Multivariate fault detection and classification of magnetic field breakdown on a plasma etch tool. In *Proceedings of the European Advanced Process Control and Manufacturing (APCM) Conference, April 16-19, 2012, Grenoble, France*, 2012.

Hiden, H.G.; Willis, M.J.; Tham, M.T., and Montague, G.A. Nonlinear principal component analysis using genetic programming. *Computers & Chemical Engineering*, 23(3):413–425, 1999.

Horn, J.L. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

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

Hotelling, H. *Multivariate quality control.* McGraw-Hill, New York, 1947.

Hubert, M. and Debruyne, M. Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):36–43, 2010.

Hubert, M.; Rousseeuw, P.J., and Verboven, S. A fast method for robust principal components with applications in chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60:101–111, 2002.

Hubert, M.; Rousseeuw, P.J., and Vanden Branden, K. ROBPCA: a new approach to robust principal component analysis. *Technometrices*, 47(1): 64–79, 2005.

Hubert, M.; Rousseeuw, P.J., and Van Aelst, S. High-breakdown robust multivariate methods. *Statistical Science*, 23(1):92–119, 2008.

Jackson, J.E. *A User's Guide To Principal Components.* Wiley, New York, 1991.

Jackson, J.E. and Mudholkar, G.S. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

Jia, F.; Martin, E.B., and Morris, A.J. Nonlinear principal component analysis with application to process fault detection. *International Journal of Systems Science*, 31:1473–1487, 2001.

Joliffe, I.T. *Principal Component Analysis.* Springer, New York, 2002.

Jorgensen, K.W. and Hansen, L.K. Model selection for Gaussian kernel PCA denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 23(1):163–168, 2012.

Karatzoglou, A.; Smola, A.; Hornik, K., and Zeileis, A. kernlab - An S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9): 1–20, 2004. URL `http://www.jstatsoft.org/v11/i09`.

Kiers, H.A.L. Hierarchical relations among three-way methods. *Psychometrika*, 56(3):449–470, 1991.

Kourti, T. Application of latent variable methods to process control and multivariate statistical process control in industry. *International Journal of Adaptive Control and Signal Processing*, 19:213–246, 2005.

Kourti, T. and MacGregor, J.F. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.

Kourti, T.; Nomikos, P., and MacGregor, J.F. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *Journal of Process Control*, 5(4):277–284, 1995.

Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *American Institute of Chemical Engineers Journal*, 37 (2):233–243, 1991.

Kresta, J.V.; MacGregor, J.F., and Marlin, T.E. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, 69(1):35–47, 1991.

Lee, J.-M.; Yoo, C.K.; Choi, S.W.; Vanrolleghem, P.A., and Lee, I.B. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1):223–234, 2004a.

Lee, J.-M.; Yoo, C.K., and Lee, I.B. Fault detection of batch processes using multiway kernel principal component analysis. *Computers and Chemical Engineering*, 28(9):1837–1847, 2004b.

Lennox, B.; Montague, G.A.; H.G., Hiden; Kornfeld, G., and Goulding, P. Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, 74(2):125–135, 2001.

Li, G. and Chen, Z. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. *Journal of the American Statistical Association*, 80:759–766, 1985.

Locantore, N.; Marron, J.S.; Simpson, D.G.; Tripoli, N.; Zhang, J.T., and Cohen, K.L. Robust principal component analysis for functional data. *Test*, 8(1):1–73, 1999.

Lopuhaä, H.P. and Rousseeuw, P.J. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19:229–248, 1991.

Louwerse, D.J. and Smilde, A.K. Multivariate statistical process control of batch processes based on three-way models. *Chemical Engineering Science*, 55(7):1225–1235, 2000.

Lu, C.-D.; Zhang, T.-Y.; Du, X.-Z., and Li, C.-P. A robust kernel PCA algorithm. In *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, pages 3084–3087. IEEE, 2004.

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

Mason, R.L. and Young, J.C. *Multivariate Statistical Process Control with Industrial Applications*. ASA-SIAM, Philadelphia, PA, 2002.

Mason, R.L.; Tracy, N.D., and Young, J.C. Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, 27: 99–108, 1995.

Mason, R.L.; Tracy, N.D., and Young, J.C. A practical approach for interpreting multivariate T^2 control chart signals. *Journal of Quality Technology*, 29(4):396–406, 1997.

Meng, X.; Morris, A.J., and Martin, E.B. On-line monitoring of batch processes using a PARAFAC representation. *Journal of Chemometrics*, 17(1): 65–81, 2003.

Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, pages 415–446, 1909.

Miller, P.; Swanson, R., and Heckler, C. Contribution plots: a missing link in multivariate quality control. *Applied Mathematics and Computer Science*, 8:775–792, 1998.

Nguyen, M.H. and De La Torre, F. Robust kernel principal component analysis. In *Advances in Neural Information Processing Systems 21*, pages 1185–1192. Curran Associates, Inc., 2009.

Nomikos, P. and MacGregor, J.F. Monitoring batch processes using multiway principal component analysis. *American Institute of Chemical Engineers Journal*, 40(8):1361–1375, 1994.

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

Palus, M. and Dvorak, I. Singular-value decomposition in attractor reconstruction: pitfalls and precautions. *Physica D*, 55(1-2):221–234, 1992.

Phaladiganon, P.; Kim, S.B.; Chen, V.C.P., and Baek, J.-G. Bootstrap-based T^2 multivariate control charts. *Communications in Statistics - Simulation and Computation*, 40(5):645–662, 2011.

Phaladiganon, P.; Kim, S.B.; Chen, V.C.P., and Jiang, W. Principal component analysis-based control charts for multivariate nonnormal distributions. *Expert Systems with Applications*, 40:3044–3054, 2013.

Pison, G.; Van Aelst, S., and Willems, G. Small sample corrections for lts and mcd. *Metrika*, 55(1-2):111–123, 2002.

Pravdova, V.; Estienne, F.; Walczak, B., and Massart, D.L. A robust version of the Tucker3 model. *Chemometrics and Intelligent Laboratory Systems*, 59(1):75–88, 2001.

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

Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2):220–234, 2012.

Qin, S.J.; Valle, S., and Piovoso, M.J. On unifying multiblock analysis with application to decentralized process monitoring. *Journal of Chemometrics*, 15:715–742, 2001.

Qin, S.J.; Cherry, G.; Good, R.; Wan, J., and Harrison, C.A. Semiconductor manufacturing process control and monitoring: a fab-wide framework. *Journal of Process Control*, 16:179–191, 2006.

R Core Team, . *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

Rousseeuw, P. and Croux, C. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

Rousseeuw, P.J. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

Rousseeuw, P.J. and Van Driessen, K. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Rousseeuw, P.J. and Van Zomeren, B.C. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85 (411):633–639, 1990.

Schölkopf, B. and Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT press, Cambridge, MA, 2002.

Schölkopf, B.; Smola, A.J., and Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Shevlyakov, G. and Smirnov, P. Robust estimation of the correlation coefficient: an attempt of survey. *Austrian Journal of Statistics*, 40(1&2): 147–156, 2011.

Skinner, K.R.; Montgomery, D.C.; Runger, G.C.; Fowler, J.W.; McCarville, D.R.; Rhoads, T.R., and Stanley, J.D. Multivariate statistical methods for modeling and analysis of wafer probe test data. *IEEE Transactions on Semiconductor Manufacturing*, 15(4):523–530, 2002.

Smilde, A.; Bro, R., and Geladi, P. *Multi-Way Analysis with Applications in the Chemical Sciences.* Wiley, Chichester, UK, 2004.

Stahel, W.A. *Robuste Schätzungen: Infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen.* PhD thesis, Eidgenössische Technische Hochschule, Zürich, 1981.

Sullivan, J.H. and Woodall, W.H. A comparison of multivariate control charts for individual observations. *Journal of Quality Technology*, 28:398–408, 1996.

Sullivan, J.H. and Woodall, W.H. Adapting control charts for the preliminary analysis of multivariate observations. *Communications in Statistics - Simulation and Computation*, 27:953–979, 1998.

Tracy, N.; Young, J.C., and Mason, R.L. Multivariate control charts for individual observations. *Journal of Quality Technology*, 24(2):88–95, 1992.

Tucker, L. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Van Sprang, E.N.M.; H.-J., Ramaker; Westerhuis, J.A.; Gurden, S.P., and Smilde, A.K. Critical evaluation of approaches for on-line batch process monitoring. *Chemical Engineering Science*, 57(18):3979–3991, 2002.

Vargas, J.A. Robust estimation in multivariate control charts for individual observations. *Journal of Quality Technology*, 35(4):367–376, 2003.

Varmuza, K. and Filzmoser, P. *Multivariate Statistical Analysis in Chemometrics.* Taylor & Francis - CRC Press, Boca Raton, FL, 2009.

Wang, J.; Zamar, R.; Marazzi, A.; Yohai, V.; Salibian-Barrera, M.; Maronna, R.; Zivot, E.; Rocke, D.; Martin, D.; Maechler, M., and Konis, K. *robust: Robust Library*, 2014. URL `http://CRAN.R-project.org/package=robust`. R package version 0.4-16.

Westerhuis, J.A.; Kourti, T., and MacGregor, J.F. Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics*, 12: 301–321, 1998.

Westerhuis, J.A.; Gurden, S.P., and Smilde, A.K. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51:95–114, 2000.

Willems, G.; Pison, G.; Rousseeuw, P.J., and Van Aelst, S. A robust Hotelling test. *Metrika*, 55:125–138, 2002.

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

Wise, B.M.; Gallagher, N.B.; Butler, S.W.; White, D.D., and Barna, G.G. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13 (3-4):379–396, 1999.

Wold, H. *Nonlinear estimation by iterative least squares procedures*, pages 37–52. Research papers in statistics. Wiley, New York, 1966.

Wold, H. *Path models with latent variables: The NIPALS approach*, pages 307–357. Quantitative Sociology: International perspectives on mathematical and statistical model building. Academic Press, 1975.

Wold, S. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, 1978.

Wold, S.; Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and Intelligent Systems*, 2:37–52, 1987a.

Wold, S.; Geladi, P., and Esbensen, K. Multi-way principal component analysis and PLS-analysis. *Journal of Chemometrics*, 1:41–56, 1987b.

Wold, S.; Hellberg, S.; Lundstedt, T.; Sjostrom, M., and H., Wold. Proc. Symp. on PLS model building: theory and application. *Frankfurt am Main*, 1987c.

Xie, Y. *knitr: A general-purpose package for dynamic report generation in R*, 2014. URL `http://yihui.name/knitr/`. R package version 1.7.

Yue, H.H. and Qin, S.J. Reconstruction-based fault identification using a combined index. *Industrial & Engineering Chemistry Research*, 40:4403–4414, 2001.

Yue, H.H.; Qin, S.J.; Marke, R.J.; Nauert, C., and Gatto, M. Fault detection of plasma etchers using optical emission spectra. *IEEE Transactions on Semiconductor Manufacturing*, 13(3):374–385, 2000.

Zhang, Y.-W.; H., Zhou, and Qin, S.J. Decentralized fault diagnosis of large-scale processes using multiblock kernel principal component analysis. *Acta Automatica Sinica*, 36(4):593–597, 2010.