technische universität
dortmund

Kernel methods for advanced Statistical Process Control

Issam Ben Khedhiri

Doctoral Thesis
in partial fulfilment of the
requirements for the Degree of
“Doktor der Naturwissenschaften”
in the Department of Statistics
University of Dortmund

Dortmund, November 2011

Prüfungskommission:

Prof. Dr. Jörg Rahnenführer (Vorsitzender)	Universität Dortmund
Prof. Dr. Claus Weihs (Gutachter)	Universität Dortmund
Prof. Dr. Joachim Kunert (Gutachter)	Universität Dortmund
Prof. Dr. Mohamed Limam (Gutachter)	University of Tunis
Dr. Dr. Marco Grzegorczyk (Beisitzer)	Universität Dortmund

Tag der mündlichen Prüfung: 23. November 2011

Acknowledgment

I would like to express my sincere gratitude to my supervisors Professor Claus Weihs and Professor Mohamed Limam. This thesis would not have been possible without their encouragement, guidance and support. I am deeply indebted to the Deutscher Akademischer Austausch Dienst (DAAD) for financing my study and for giving me the chance to conduct my research in such high quality environment. Many thanks go to my DAAD coordinator Mrs Cornelia Hanzlik-Rudolph for her valuable help and advice. I would also like to acknowledge my colleagues Björn Bornkamp and Adrian Wilk for helping me with valuable advice during the start of my study at the University of Dortmund. I am heartily thankful to Mrs Sabine Bell for her great efforts throughout my stay at the university. Lastly, I would like to thank my family for their support. I am greatly and forever indebted to my parents for their love and generosity throughout my entire life.

List of Figures

List of Tables

Liste of Abbreviations

ARL	Average Run Length
ANN	Artificial Neural Networks
AR	Autoregressive model
ARIMA	Autoregressive Integrated Moving Average
APCA	Adaptive Principal Components Analysis
AKPCA	Adaptive window Kernel Principal Components
CUSUM	Cumulative Sum
ExpAR	Exponential Autoregressive model
EWMA	Exponentially Weighted Moving Average
KPCA	Kernel Principal Components Analysis
KKT	Karush Kuhn Tucker
LCL	Lower Control Limit
MA	Moving Average
MCUSUM	Multivariate Cumulative Sum
MEWMA	Multivariate Exponentially Weighted Moving Average
MWKPCA	Moving Window Kernel Principal Components
MSE	Mean Squared Error
PCA	Principal Components Analysis
PCs	Principal Components
RL	Run Length
SVD	Singular Value Decomposition
SV	Support Vectors
SPC	Statistical Process Control
SVR	Support Vector Regression
SVDD	Support Vector Domain Description
SPE	Squared Prediction Error
TE	Tennessee Eastman
UCL	Upper Control Limit

Contents

1 Introduction

In order to assure a high level of process monitoring performance, researchers and practitioners frequently use many statistical methods called Statistical Process Control (SPC). For many decades, the primary use of SPC methods was focused on industrial quality control applications. However, with recent increase of information technology systems, new areas of SPC applications have emerged. Examples of new applications range from video surveillance systems (Elbasi et al. (2005)), network intrusion detection (Park (2005)) to health and safety management (Benneyan et al. (2003)). The introduction of SPC methods to these new areas, the increase of processes complexities and the increase of control improvement requirements, has led to the necessity of enhancing classical SPC procedures. Indeed, in order to apply classical SPC methods, very restrictive assumptions that are usually difficult to be satisfied in several process control applications are imposed. To overcome these limits, introduction of advanced computational intelligence tools into SPC has lately witnessed lot of interest. The use of these techniques can allow managing and monitoring of complex processes with high accuracy. Moreover, computational intelligence methods can overcome several restrictions and provide better process control results. Some of the most recent promising computational intelligence methods are Kernel methods. These methods have many advantages since they allow learning the particular structure of a model from data, can handle non linear relationships and could be applied in a number of off-line and on-line process assessments. Moreover, Kernel methods are very flexible and have various forms to handle different problems such as one-class classification, data reduction and regression problems. Therefore, this study investigates development of several kernel methods in order to provide advanced SPC procedures that can be applied to several processes where assumptions of classical methods are not satisfied. Main developments of kernel methods for SPC will be carried through three methods namely Kernel Principal Component Analysis (KPCA), Support Vector Regression (SVR) and Support Vector Domain Description (SVDD). In this section first an insight into SPC will is given, then an overview of the organization of this dissertation is exposed.

1.1 Statistical Process Control

As stated by Montgomery (2005), SPC is a collection of tools that help achieving process stability and improving capability through reduction of the variance. This objective is mainly obtained by quick detection and elimination of unusual disruptions and faults. Formally speaking, a fault is defined as a departure of a calculated statistic from an acceptable range. Therefore, fault detection is considered as a hypothesis testing, where one should decide whether the process is in-control or out-of-control. To distinguish between these two states, one should distinguish between common causes of variations, that are a natural change of the process and that can not be eliminated, and unusual shocks and disturbances that should be removed. Usually, this goal is achieved through the use of control charts. As defined by Stapenhurst (2005), a control chart is a plot of process characteristics usually through time with statistically determined limits. The control chart summarizes all provided process information into a single index that gives an idea about the quality of the operating system. Then, control limits of this index are used to decide if faults are present in the process. Figure 1.1 illustrates one of the most frequently used control charts which is called Hotelling's T^2 chart.

The construction of any control chart includes two distinct phases (Woodall (2000)). In phase I, when the first observations are available, one should decide whether the process is in control or not. In this phase, the role of the engineers and practitioners is very important. The process should be studied in order to define and give an idea about the degree of the process quality. The data collected during this phase are then analyzed in an attempt to define what is meant by a process being statistically in control. This fact is achieved by determining or estimating process parameters and constructing control chart limits. In phase II, the control chart is used to monitor the process. This is performed by testing whether the process remains in control when future observations are available. In certain cases some changes

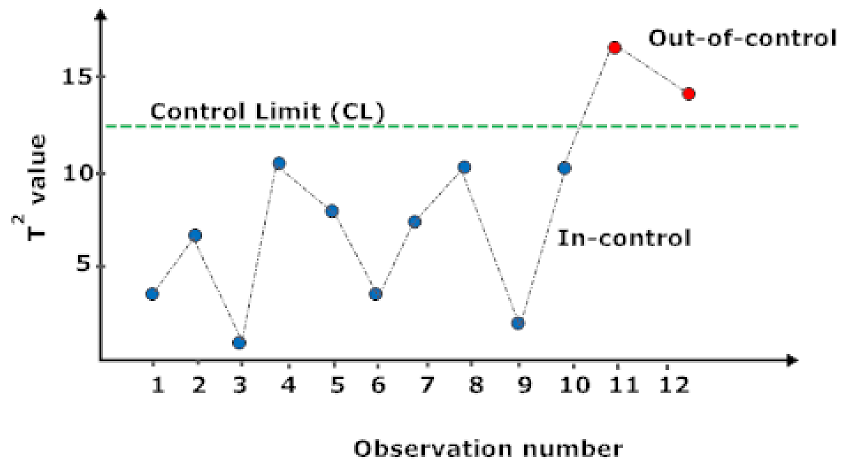

Figure 1.1: Illustration of Hotelling's T^2 control chart

in the process characteristics can be considered as normal and therefore the in-control state conditions are re-estimated.

1.2 Outline of the Dissertation

Despite the wide use of classical control charts there are some problems related to their application in real situations. In fact, usually assumptions to different models such as linearity are imposed. Also, conventional charts are not adequate for analyzing large data sets and dealing with real time processes since they are basically non-adaptive procedures. This control strategy does not suit non-stationary continuous systems that tend to drift due to various phenomena as the process may undergo changes. This drift may change the relationship between variables and could cause control chart to become invalid. Therefore, the first part of this dissertation is dedicated to the development of a monitoring procedure applied to different non-stationary processes. The proposed chart is based on an adaptive KPCA method that allows modelling of non-linear process behaviour without the need to impose a predefined structure.

Moreover, conventional charts stipulate that process observations should be independently distributed. However, due to advances made in process automation, violations of this assumption are frequent and process variables are usually autocorrelated. Indeed, high sampling collection often produces series of observations that are close enough to be dependent. As mentioned by Psarakis and Papaleonida (2007), autocorrelation is present in most continuous systems. This violation could affect the performance of control charts and one of its typical effect is the increase of false alarms rate. An approach to overcome this issue is the use of residuals control charts. These charts model the autocorrelation structure and then use residuals, which would approximately be independent, for process monitoring. However, time series modelling could be difficult in many applications because of the specification problem especially for nonlinear processes. Thus, the second part of this dissertation is devoted to the development of residuals control charts based on SVR method that allow monitoring of nonlinear systems without requiring lot of knowledge about the process structure.

In addition to the above mentioned assumptions, most conventional charts impose certain probability distribution and in most cases a normal distribution is used. However, in practice this assumption rarely holds. Moreover, for complex systems it is usually difficult to determine which distribution is appropriate to a given process. Another issue that is not enough investigated in the literature is the monitoring of processes that run under multiple operating modes. Indeed, classical control charts assume that the underlying process is operating only under one nominal operating region. However, in many cases systems can run under several modes because of product changes, set-point changes and manufacturing strategies (Zhiqiang and Zhihuan (2008)). Therefore, the third part of this dissertation investigates the development of local SVDD based control chart to monitor such systems.

Part I

Adaptive Kernel Principal Components Analysis for non-stationary process control

Many studies have shown that adaptive control charts, in which one or more design parameters vary in real time during the production process, have superior statistical and economic performances, when compared to traditional control charts (Magalhaes et al. (2006)). The statistical design of adaptive classical charts has only focused on adapting few parameters of the chart such as the upper percentage factor that is used for determining the limit. Adaptive classical control charts allow changing only few parameters of the chart and not those related to process structure. For this reason several researchers proposed adaptive techniques to better suit on-line monitoring and construct online control charts. Development of adaptive data reduction techniques for process monitoring has mostly concerned linear Principal Components Analysis (PCA). In fact, Li et al. (2000) proposed an Adaptive Principal Components Analysis (APCA) algorithm for an adaptive process monitoring by introducing an efficient approach to update the correlation matrix, the number of principal components and the confidence limits recursively. To deal with changing process conditions, Lee et al. (2005) developed adaptive multiway PCA models to update the covariance structure at each scale. The proposed adaptive multiscale method is successfully applied to a sequencing batch reactor. Choi et al. (2006) introduced recursive updated PCA along with two monitoring metrics, Hotelling's T^2 and the Q statistic, for monitoring time-varying processes. However, PCA assumes that the relationship between variables is linear and therefore its application to nonlinear processes can provide poor results. Thus, many researchers have proposed the use of Kernel Principal Components Analysis (KPCA) method in order to monitor such processes.

Lee et al. (2004) applied KPCA technique as a new nonlinear process monitoring technique for fault detection in two multivariate processes. Authors showed that the proposed approach is effective in capturing the nonlinear relationship in the process variables and that it has superior process monitoring performance compared to linear PCA. Hoffmann (2007) investigates the use of KPCA for novelty detection and demonstrated that it has a competitive performance on two-dimensional synthetic distributions and on two real-world data sets. Ruixiang et al. (2007) present a novel approach, called Evolving Kernel Principal Component Analysis, to fault classification based on the integration of KPCA with an evolutionary optimization algorithm. The application in fault diagnosis to a large-scale rotating machine shows that the proposed method is efficient in discovering the optimal nonlinear features corresponding to real-world operational data. Cui et al. (2007) improved KPCA for fault detection in two aspects. Firstly, a feature vector selection scheme based on a geometrical consideration is given to reduce the computational complexity of KPCA. Secondly, a KPCA plus Fisher Discriminant Analysis scheme are adopted to improve the fault detection performance of KPCA. Their simulation results show the effectiveness of these improvements for fault detection performance in terms of low computational cost and high fault detection rate.

Recently Liu et al. (2009) proposed application of adaptive KPCA for online process monitoring called Moving Window Kernel Principal Components Analysis (MWKPCA). Results showed that applying such models provides good detection capabilty. However, in order to train KPCA continually, the approach adopted by Hoegaerts et al. (2007) is used. This adopted method allows introducing and to eliminate only one observation. Also because of this fact the window size of KPCA is assumed to be constant. However, in many practical situations not only one new observation but a block of new data is present. Moreover, sometimes it is of interest to freeze the model for certain time or to eliminate a number of observations that do not characterize the process states. Therefore, in this part, to have a flexible control strategy, an adaptive block KPCA technique is applied in a monitoring procedure with a variable window size model. Also, a method to recursively determine both window size and chart control limits is proposed. Moreover, we present an efficient adaptive KPCA method that allows introduction or elimination of a data block at the same time. When a block of data need to be introduced in the model, this adaptive KPCA could overcome the computational cost of storing and manipulating the kernel matrix and it is faster than batch computation of KPCA and adaptive KPCA of Hoegaerts et al. (2007). Then, a comparison between developed adaptive KPCA and different PCA control charts is performed.

This Part of the thesis is outlined as follows: The first chapter presents an overview of linear PCA and KPCA methods and discusses principle of PCA based control charts along with the proposed monitoring chart. Chapter 2 proposes a fast adaptive block KPCA method and evaluates its computational cost and its accuracy. Chapter 3 is devoted to the application and analysis of the proposed chart to simulated data and to a real case study.

2

Process Monitoring using PCA models

This chapter is organized as follows: First section gives an overview of PCA method. Then, the nonlinear Kernel PCA method is presented in Section 2. Section 3 exposes the way to construct Kernel functions and highlights most used functions. In Section 4, PCA and KPCA based control charts are discussed. Section 5 outlines a proposition for development of a variable window size adaptive KPCA chart for non-stationary process monitoring. Eventually, Section 6 resumes this chapter.

2.1 Principal Components Analysis

As stated by Montgomery (2005), conventional multivariate control charts are effective as long as the number of process variables is not very large. When the number of controlled variables increases, classical control charts lose their effectiveness. Therefore, methods which resume most important process information in few variables should be used. One of the most widely used methods to reduce a complex data set to a lower dimension is PCA. The objective of PCA method is to compute the principal components $y_1, ..., y_m$ of the original variables $x_{1,...,}x_m$ by finding a set of linear combinations such that

$$y_i = w_{i1}x_1 + ... + w_{im}x_m \qquad \forall\ i = 1, .., m. \tag{2.1}$$

However, the hope is that only a reduced set of p principal components approximates the original space. Following Hotelling (1933), the p principal axes are those orthonormal axes onto which the variance retained under projection is maximal. Let X be an $(m \times n)$ matrix of the centred original variables, n being the number of observations, y_1 being the first principal component to find and $w = [w_1 ... w_m]$, a column vector with size m, then the empirical variance of y_1 is calculated as follows,

$$\begin{aligned} V(y_1) &= V(w^T X), \\ &= \frac{1}{n} w^T X X^T w, \\ &= w^T S w, \end{aligned} \tag{2.2}$$

where S is the empirical covariance matrix. Then, we choose w to maximize $w^T S w$ while constraining w to have unit length such that,

$$\begin{aligned} &Max \;\; w^T S w \\ &w^T w = 1. \end{aligned} \tag{2.3}$$

Using the method of Lagrange multipliers, we obtain

$$L(w, \alpha_1) = w^T S w - \alpha_1 (w^T w - 1), \tag{2.4}$$

where α_1 is the lagrange multiplier.

Solving this optimization problem, we can easily show that

$$S w = \alpha_1 w. \tag{2.5}$$

This fact clearly shows that α_1 and w are respectively an eigenvalue and an eigenvector of the covariance matrix S.

2.2 Kernel Principal Components Analysis

Since basic PCA performs well only on linear processes, a nonlinear PCA technique for estimating nonlinear problems, called kernel PCA, was introduced by Schölkopf et al. (1998). The basic idea of KPCA is to first map the input space into a feature space via nonlinear mapping and then to compute the principal components in that feature space. This stipulates that the data can always be mapped into a higher-dimensional space in which they vary linearly. As a result, KPCA performs a nonlinear PCA in the input space. Figure 2.1 illustrates the principle of this method.

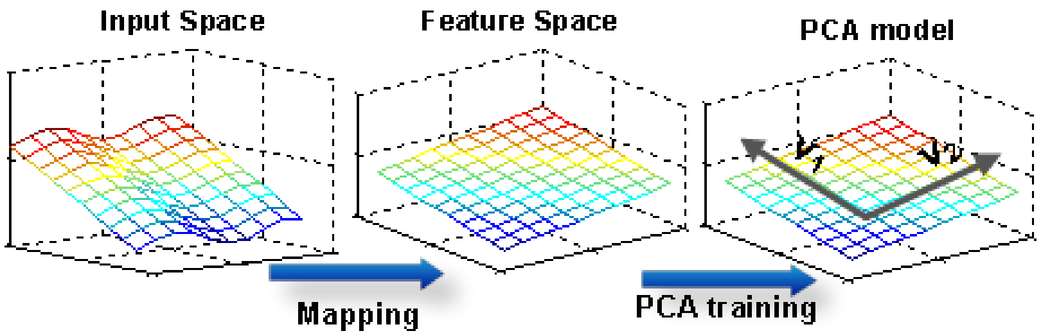

Figure 2.1: Basic idea of Kernel PCA

First, we note that KPCA is a basis transformation of the empirical covariance matrix of the transformed data $x_i \in \mathbb{R}^m$, $i = 1, .., n$, defined as

$$C = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T, \tag{2.6}$$

where it is assumed that $\sum_{i=1}^{n}\phi(x_i) = 0$ and $\phi(.)$ is a nonlinear mapping. To find the principal components, we solve the eigenvalue problem in the feature space such that

$$\lambda\mu = C\mu, \tag{2.7}$$

where eigenvalues $\lambda \geq 0$, μ is a vector of eigenloadings and there must exist coefficients γ_i , $i = 1, .., n$, such that

$$\mu = \sum_{i=1}^{n}\gamma_i\phi(x_i). \tag{2.8}$$

Equation (2.7) is equivalent to

$$\lambda < \phi(x_k), \mu >=< \phi(x_k), C\mu > \quad \forall\ k = 1, .., n. \tag{2.9}$$

Combining Equations (2.6), (2.8) and (2.9), we obtain for $k = 1, ..., n$,

$$\lambda\sum_{i=1}^{n}\gamma_i < \phi(x_i), \phi(x_k) >= \frac{1}{n}\sum_{i=1}^{n}\gamma_i \left\langle \phi(x_k), \sum_{j=1}^{n}\phi(x_j) < \phi(x_j), \phi(x_i) > \right\rangle . \tag{2.10}$$

Calculating the feature vectors $\phi(x_i)$ can be computationally expensive or even impossible if the dimension of the feature space is high or infinite. Clearly this approach is not feasible and there is a need to find a computationally cheaper way. Fortunately, all the calculation involving the $\phi(x_i)$'s appear as inner products. Instead of explicitly mapping the vectors and then computing the inner product, there exist under certain condition a function $K(x_i, x_j)$ whose value gives the inner product $< \phi(x_i), \phi(x_j) >$. To show this let us consider the map $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$,

$$\begin{aligned} \phi(x) &= \phi(X_1, X_2), \\ &= (X_1^2, \sqrt{2}X_1X_2, X_2^2). \end{aligned} \tag{2.11}$$

Thus, we have

$$\begin{aligned} < \phi(x), \phi(x') > &= < \phi(X_1, X_2), \phi(X_1', X_2') >, \\ &= \ < (X_1^2, \sqrt{2}X_1X_2, X_2^2), (X_1'^2, \sqrt{2}X_1'X_2', X_2'^2) >, \\ &= \ (X_1X_1' + X_2X_2')^2, \\ &= \ < x, x' >^2 . \end{aligned} \tag{2.12}$$

As a consequence, when using a function $K =< x, x' >^2$, the inner product is computed efficiently by saving lot of calculation. In the literature the function $K(.,.)$ is usually called a *Kernel* and can be considered as a measure of similarity. Then, the inner product $< \phi(x_i), \phi(x_j) >$ of Equation (2.10) is changed by the kernel function $K(x_i, x_j)$ and abbreviated by K_{ij} as follows,

$$\lambda\sum_{i=1}^{n}\gamma_i K_{ik} = \ \frac{1}{n}\sum_{i=1}^{n}\gamma_i \left(\sum_{j=1}^{n}K_{kj}K_{ji}\right) \quad \forall\ k = 1, .., n. \tag{2.13}$$

Equation (2.13) can be written as follows,

$$\lambda K V = \frac{1}{n} K^2 V , \tag{2.14}$$

$$\text{If } K \text{ is invertible, then} \quad \lambda V = \frac{1}{n} K V , \tag{2.15}$$

where $V = [\gamma_1, ..., \gamma_n]^T$ and K is an $(n \times n)$ matrix defined by K_{ij}.

Now, applying PCA can be realized by solving the eigen-problem of Equation (2.15). This yields eigenvectors $V^1, ..., V^n$ with eigenvalues $\lambda_1 \geq ... \geq \lambda_n$. In order to insure the normality of $\mu^1, ..., \mu^n$, (Equation (2.7)), the corresponding vectors $V^1, ..., V^n$ should be scaled such that

$$< \mu^l, \mu^l >= 1, \quad \forall\, l = 1, .., n \tag{2.16}$$

Using Equation (2.8) this translates to

$$\begin{aligned} \sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_i^l \gamma_j^l \ \langle \phi(x_i), \phi(x_j) \rangle &= 1, \quad l = 1, .., n & (2.17)\\ \sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_i^l \gamma_j^l K_{ij} &= 1, \quad l = 1, .., n & (2.18)\\ \left\langle V^l, K V^l \right\rangle &= 1, \quad l = 1, .., n & (2.19) \end{aligned}$$

Using Equation (2.15), $V^1, ..., V^n$ should be normalized such that

$$\begin{aligned} n\lambda_l \left\langle V^l, V^l \right\rangle &= 1, \quad l = 1, .., n & (2.20)\\ \left\langle V^l, V^l \right\rangle &= \frac{1}{n\lambda_l}, \quad l = 1, ..., n & (2.21) \end{aligned}$$

The first p principal components (t_z) of a test vector x are then extracted by projecting x onto eigenvectors $V^1, ..., V^p$, where,

$$t_z = \sum_{i=1}^{n} V_i^z K(x_i, x), \quad z = 1, ..., p. \tag{2.22}$$

Because the kernel function is known as a measure of similarity, KPCA can work very well for process monitoring since the goal is to distinguish aberrant observations from others. Next section examines the necessary conditions that must be satisfied by a valid kernel function, along with providing some well known kernels.

2.3 Construction and examples of Kernel functions

2.3.1 Condition of kernel functions

A kernel function is a function that returns the inner product between images of two inputs in some feature space. Which function $K(.,.)$ corresponds to a dot product in some feature space is given by the following condition given by Mercer (1909).

Let G a matrix containing measures of similarity of a given training set,

$$G = \begin{pmatrix} k_{11} & k_{12} & ... & k_{1n} \\ k_{21} & k_{22} & . & . \\ . & . & . & . \\ k_{n1} & k_{n2} & . & k_{nn} \end{pmatrix} \tag{2.23}$$

This matrix is called *Gram matrix.*

Theorem: The function $K(x, x')$:$\chi \times \chi \longrightarrow \mathbb{R}$ is a kernel if and only if

$$G = (K(x_i, x_j))_{i,j=1:n}, \tag{2.24}$$

is non-negative definite for all finite sequence of points $x_1, ..., x_n$ of the interval χ. That is, for any non-zero vector $\lambda \in \mathbb{R}^n$, $\lambda^T G \lambda \geq 0$.

2.3.2 Standard kernel functions

In practice it is difficult to check if some particular kernel satisfies Mercer's conditions, since these conditions should hold for every non-zero vector $\lambda \in \mathbb{R}^n$. For this reason, usually some well known kernels are used. In this section we present the most widely used functions.

Dot-product type Kernels

A function of dot-product type is given by $k(x, x\prime) = k(< x, x\prime >)$. Table 2.1 gives some well known dot-product kernel functions.

Table 2.1: Standard dot-product type kernel functions

Name	Formula	Parameters
Homogeneous polynomial	$< x, x\prime >^p$	$p \in \mathbb{N}$
Inhomogeneous polynomial	$(< x, x\prime > + \lambda)^p$	$p \in \mathbb{N}$, $\lambda \in \mathbb{R}^+$
Sigmoid	$tanh(\lambda + \beta < x, x' >)$	$\lambda \in \mathbb{R}^+$, $\beta \in \mathbb{R}^+$

Translation Invariant type kernel

A translation invariant kernel has the following form: $k(x, x\prime) = k(x - x\prime)$. Table 2.2 provides some well known translation invariant kernel functions.

Table 2.2: Standard translation type kernel functions

Name	Formula	Parameters
Exponential Radial	$\exp(\frac{-\|x - x\prime\|}{\sigma^2})$	$\sigma \in \mathbb{R}^*$
Gaussian Radial	$\exp(\frac{-\|x - x\prime\|^2}{\sigma^2})$	$\sigma \in \mathbb{R}^*$
Anisotropic Radial	$\exp(\frac{-(x - x\prime) \sum^{-1} (x - x\prime)^t}{\sigma^2})$	$\sigma \in \mathbb{R}^*$

$\sum^{-1}$ is the inverse of the covariance matrix.

This study will concentrate on the use of the Gaussian kernel function. In fact, Evangelista et al. (2007) stated that this function is a powerful kernel used in pattern recognition. Moreover, it is employed with several heuristics. Bu et al. (2009) mentioned that due to its superior property and versatility, the Gaussian kernel has been widely used in practice.

2.4 PCA and KPCA based control charts

We now present two monitoring statistics based on KPCA, named the Squared Prediction Error (SPE), known as the Q statistic, and Hotelling's T^2 statistic.

2.4.1 Error based PCA control chart

For a new observation x_{new}, using linear PCA, the Q statistic is defined as follows,

$$Q = x_{new}x_{new}^T - (x_{new}P)(x_{new}P)^T, \tag{2.25}$$

where x_{new} a row vector and P is the matrix of PCA eigenloadings. The KPCA based monitoring method is similar to that using PCA in that the Q statistic in the feature space can be interpreted in the same way. The Q statistic is defined by Choi et al. (2005) as follows

$$\begin{aligned} Q &= |\bar{k}(x_{new}, x_{new}) - \hat{k}(x_{new}, x_{new})|, && (2.26)\\ &= |\bar{k}(x_{new}, x_{new}) - tt^T|, && (2.27) \end{aligned}$$

where $\bar{k}$ is the scaled kernel product, $\hat{k}$ the projection of $\bar{k}$ into KPCA model obtained from Equation (2.15) where

$$t = [\bar{k}(x_{new}, x_1), .., \bar{k}(x_{new}, x_n)]V_{np}, \tag{2.28}$$

where $V_{np} = [V^1, ..., V^p]$.

The scaling is based on mean centering and variance scaling. Mean centering is performed as follows

$$\tilde{k}_{ij} = k_{ij} - \frac{1}{n}\sum_{j=1}^{n} k_{ij} - \frac{1}{n}\sum_{i=1}^{n} k_{ij} + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} k_{ij}, \tag{2.29}$$

whereas the variance scaling is as next,

$$\bar{k}_{ij} = \frac{\tilde{k}_{ij}}{trace(\tilde{k}_{..})/(n-1)}. \tag{2.30}$$

The Q statistic indicates the extent to which each sample conforms to the PCA model. It is a measure of the amount of variation not captured by the principal component model. The upper limit for the Q statistic is given by

$$Q_{\text{limit}} = \frac{\theta_2}{2\theta_1}\chi_\alpha\left(\frac{2\theta_1^2}{\theta_2}\right), \tag{2.31}$$

where $\theta_1$and θ_2 are the sample mean and variance of Q values, and χ_α is the chi-squared distribution with risk level α (Nomikos and MacGregor (1995)).

For slowly time-varying processes, the confidence limit for detection indices changes with time, making adjustment of this limit necessary for on-line monitoring. For the Q statistic, parameter values of θ_1, θ_2 are recursively updated using the p largest eigenvalues after each collection of new data block making Q_{limit} time-varying. Also, a forgotten factor for Q is introduced. This fact is performed by excluding the influence of the oldest Q value from the mean and variance values, θ_1 and θ_2.

2.4.2 Hotelling's T^2 based PCA control chart

Hotelling's T^2 statistic allows measuring the distance of new observations to the KPCA model. This statistic is calculated as follows,

$$T^2 = t\Omega_p^{-1}t^T. \tag{2.32}$$

where t is obtained by Equation (2.22) and Ω_p is a diagonal matrix of eigenvalues $\lambda_1, ..., \lambda_p$.

Since the principal components of non-stationary nonlinear processes may not follow a multivariate normal distribution (Liu and Wu (2006)), the control limit of T^2 can be calculated by the kernel density function estimator

$$f(x) = \frac{1}{ns}\sum_{i=1}^{n}\Phi(\frac{x-T_i^2}{s}), \tag{2.33}$$

where s is a smoothing parameter and Φ is a gaussian function. The detailed selection of s can be found in Wang (1995). Using the cumulative distribution function $F(x)$, the control limit of T^2 control chart are then calculated by the $(1-\alpha)^{th}$ quantile of T^2, such that

$$T_{\text{limit}} = F^{-1}(1-\alpha). \tag{2.34}$$

Since for non-stationary processes the T^2 statistic can vary with time, this method allows adjustment of the confidence limit of T^2 by reestimating the cumulative density function of Equation (2.34) and therefore adapting the T_{limit}.

2.5 Adaptive variable window KPCA for non-stationary processes

In order to have an efficient adaptive control chart, the definition of the window size which gives information about the training data is also an important factor. A use of constant window size is very restrictive and sometimes provides poor performance because it may imply a use of corrupted training samples especially for processes that undergo several changes. For certain applications it is of interest to use a variable window size for training of KPCA model. The idea here is to train the model in regions which better characterize the actual state of the process. By this way the window size can grow or decrease depending on the states of the process. An approach to determine this window size H is to use the information contained in Q and T^2 statistics. If certain old observations imply an increase of the standard deviation of the Q or T^2 statistics then this would mean that these late observations differ from the actual process state and therefore better to eliminate them from the model. We also note that other indexes than the standard deviation of Q or T^2 can be used. Let us suppose that the Q statistic is used, then the estimated varying window size can be implemented by the following algorithm. Let H denote the actual window size, H_{max} and H_{min} the maximal and the minimal window sizes that can be used, h, k and w (typically $w = 5$) are the number, the maximal number and the size of samples to be eliminated. Table 2.3 shows how to determine KPCA model size. Notice that Table 2.3 includes updating and downdating.

Table 2.3: Algorithm to determine KPCA window size

```
$H \leftarrow H +$ size(new data)
for $h = 1 : k$
    $SD(h) \leftarrow std(Q((w(h-1)+1) : H))$
end
$h \leftarrow Index(\min(SD))$
$H \leftarrow H - (h-1) \times w$
if $H > H_{max}$
    $H \leftarrow H_{max}$
elseif $H < H_{\min}$
    $H \leftarrow H_{\min}$
```

In addition to the window size, to have an adaptive chart, not affected by integration of out-of-control variables, a condition for updating KPCA model, Q_{limit} and T_{limit} values is introduced. This fact lets the model avoid contamination by observations which could make the model insensitive to faults. Moreover, because the model can produce false alarms, specially in the case of non-stationary process, where sometimes an out-of control signal can characterize a change in the relationship between variables and not a proper fault, a margin of acceptability of observations is applied. By this way the adjustment condition is activated only if Q or T^2 values of the observation do not overstep the value of ηQ_{limit} or ηT_{limit}, where $\eta > 1$. In this study a value of η is taken to be equal to 1.1. The main algorithmic step of the proposed adaptive window kernel principal component chart is shown in Table 2.4.

Table 2.4: Algorithm of Adaptive Window KPCA control chart

```
Step 1: Given an initial standardized block of data do:
    Step 1.1: Set PC numbers and kernel parameter.
    Step 1.2: Construct the kernel matrix $K$ and scale it.
    Step 1.3: Estimate the initial KPCA model.
    Step 1.4: Calculate the initial control limit of the monitoring statistic.
Step 2: For a new block of data of size N, do:
    Step 2.1: Compute $\text{k}_{new} = (k_1, ..., k_N)$,
        $\text{k}_i = [\text{k}_{i,1}, .., \text{k}_{i,n+i-1}]$, (n is the actual number of observations of KPCA model).
    Step 2.2: Obtain the scaled kernel vector $\bar{k}_{new}$ .
    Step 2.3: Project $\bar{k}_{new}$ into KPCA and obtain $\hat{k}_{new}$.
    Step 2.4: Calculate the monitoring statistic for every observation $i = 1, ..., N$.
    Step 2.5: Test if each observation is out-of-control as in section 2.4.
Step 3: If updating condition is satisfied, do:
    Step 3.1: Adapt the limit of the monitoring statistic using (2.31) and (2.34).
    Step 3.2: Determine the new window size as in Table 2.3.
    Step 3.3: Adapt KPCA model as in chapter 3.
Step 4: Return to Step 2.
```

2.6 Conclusion

This chapter presented an overview of PCA and KPCA methods and introduced linear and kernel PCA based control charts. To control non-stationary processes, an adaptive control procedure that allows adjustment of KPCA model, the confidence limits and the window size is proposed. The proposed strategy requires that a continuous training of KPCA model should be performed. Next Chapter proposes a fast updating procedure which eases introduction of new block of data into previous KPCA model.

3

Fast adaptive Kernel Principal Components Analysis

Analysis of complex high dimensional systems, as in image processing, computer vision recognition or fault detection issues, usually requires the use of large datasets. However, in order to obtain a KPCA model from a database that contains n observations and using a standard Singular Value Decomposition (SVD) method, a computation cost of order $O(n^3)$ is needed. This fact can limit the use of KPCA for online processes where continuous observations are provided which need to be fastly integrated in the model. Therefore, to allow a better fast updating procedure, Hoegaerts et al. (2007) proposed a method to track the dominant KPCA components with reduced computation costs. Indeed the proposed algorithm requires only $O(n^2)$ operations to train KPCA model. But to update KPCA model, this algorithm has the disadvantage of allowing only the introduction of one observation at a time. However, in many online processes data are provided by blocks of new observations. Suppose there are N observations that require to be integrated in KPCA model, then the algorithm of Hoegaerts et al. (2007) needs to repeat N times the training of KPCA which requires $O(n^2), .., O((n+N)^2)$ operations. In order improve the updating strategy, this chapter proposes an extension of Hoegaerts et al (2007) algorithm to quickly update KPCA model as soon as blocks of data are present using an algorithm that requires reduced computation cost. This chapter provides an overview of the proposed fast adaptive KPCA of dominant principal components.

This chapter is organized as follows: The first section presents an overview of the way to update, downdate and downsize the KPCA model. Section 2 provides complexity analysis of the proposed algorithm. In Section 3 application and comparison of different adaptive methods is investigated.

3.1 Adapting KPCA model

3.1.1 Group updating of KPCA

Suppose that we have, for an initial kernel matrix K_n, the eigenvalue matrix Ω_P and the corresponding eigenvector matrix V_{np} of the first p principal components such that

$$K_n \approx V_{np}\Omega_P V_{np}^T \text{ and } K_n = \begin{bmatrix} K_{11} & ...... & K_{1n} \\ . & & . \\ K_{n1} & ...... & K_{nn} \end{bmatrix}. \tag{3.1}$$

The key idea is to update Ω_P and $V_{(n+N)p}$ for each new sample group of size N using Model (3.1) by adding the fact that the new kernel matrix gets N additional rows and columns as shown below

$$K_{n+N} = \begin{bmatrix} \begin{matrix} K_{11} & ...... & K_{1n} \\ . & . & . \\ K_{n1} & ...... & K_{nn} \end{matrix} & \begin{bmatrix} K_{1(n+1)} & .. & K_{1(n+N)} \\ . & .. & . \\ K_{n(n+1)} & .. & K_{1(n+N)} \end{bmatrix} \\ \begin{bmatrix} K_{(n+1)1} & . & K_{(n+1)n} \\ . & . & . \\ K_{(n+N)1} & . & K_{(n+N)n} \end{bmatrix} & \begin{bmatrix} K_{(n+1)(n+1)} & . & K_{(n+1)(n+N)} \\ . & . & . \\ K_{(n+N)(n+1)} & . & K_{(n+N)(n+N)} \end{bmatrix} \end{bmatrix}, \tag{3.2}$$

Let $\alpha^T = \begin{bmatrix} K_{(n+1)1} & .. & K_{(n+1)n} \\ . & .. & . \\ K_{(n+N)1} & .. & K_{(n+N)n} \end{bmatrix}$ and $\beta = \begin{bmatrix} K_{(n+1)(n+1)} & .. & K_{(n+1)(n+N)} \\ . & .. & . \\ K_{(n+N)(n+1)} & .. & K_{(n+N)(n+N)} \end{bmatrix}$,

then

$$K_{n+N} = \begin{bmatrix} K_n & \alpha \\ \alpha^T & \beta \end{bmatrix}, \tag{3.3}$$

$$\approx \begin{bmatrix} V_{np}\Omega_P V_{np}^T & \mathbf{0} \\ \mathbf{0}^T & \mathit{0} \end{bmatrix} + \begin{bmatrix} O & \alpha \\ \alpha^T & \beta \end{bmatrix}, \tag{3.4}$$

where $\mathbf{0}$, $\mathit{0}$ and O are respectively an $n \times N$, $N \times N$ and $n \times n$ matrices of zeros. First matrix of Equation (3.4) is derived as follows

$$\begin{bmatrix} V_{np}\Omega_P V_{np}^T & \mathbf{0} \\ \mathbf{0}^T & \mathit{0} \end{bmatrix} = \begin{bmatrix} V_{np} \\ \mathbf{0}^T \end{bmatrix} \Omega_P \begin{bmatrix} V_{np}^T & \mathbf{0} \end{bmatrix} = V_0 \Omega_P V_0^T. \tag{3.5}$$

Moreover, let

$$A^T = \left[\begin{array}{cccccc} \alpha_1 & \dfrac{\beta_1}{2} + 1 & & 0 & . & 0 \\ \alpha_1 & \dfrac{\beta_1}{2} - 1 & & 0 & . & 0 \\ & \alpha_2 & & \dfrac{\beta_2}{2} + 1 & . & 0 \\ & \alpha_2 & & \dfrac{\beta_2}{2} - 1 & . & 0 \\ & & . & & . & 0 \\ & & . & & . & 0 \\ & & \alpha_N & & & \dfrac{\beta_N}{2} + 1 \\ & & \alpha_N & & & \dfrac{\beta_N}{2} - 1 \end{array} \right] \text{ and } \Lambda = I_N \otimes \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}, \tag{3.6}$$

where $\alpha_{i=1:N} = \begin{bmatrix} K_{(n+i)1} & ... & K_{(n+i)(n+i-1)} \end{bmatrix}$ and $\beta_{i=1:N} = K_{(n+i)(n+i)}$. Then, we can show that

$$\begin{bmatrix} O & \alpha \\ \alpha^T & \beta \end{bmatrix} = A \Lambda A^T. \tag{3.7}$$

Now, let denote $\mathbf{O}$ a $p \times (2N)$ matrix of zeros, then we can approximate the new kernel matrix by

$$K_{n+N} \approx \begin{bmatrix} V_0 & A \end{bmatrix} \begin{bmatrix} \Omega_P & \mathbf{O} \\ \mathbf{O}^T & \Lambda \end{bmatrix} \begin{bmatrix} V_0^T \\ A^T \end{bmatrix}, \tag{3.8}$$

$$\approx V^*_{(n+N)(p+2N)} \Omega^*_{(p+2N)} V^{*T}_{(n+N)(p+2N)}. \tag{3.9}$$

Because of A, the new eigenvector matrix $V^*_{(n+N)(p+2N)}$ is no more orthogonal and therefore the following extraction of the orthogonal part of A to V_0

is essential such that

$$A = (I_{n+N} - V_0 V_0^T)A + V_0 V_0^T A. \tag{3.10}$$

Also the part of A orthogonal to V_0 needs to be mutually orthogonal. Then, we perform the QR-decomposition to obtain

$$(Q_A, R_A) \leftarrow qr((I_{n+N} - V_0 V_0^T)A). \tag{3.11}$$

Integrating (3.10) and (3.11) into (3.8) we obtain

$$K_{n+N} \ \approx Q^* R^* \Omega^*_{(p+2N)} R^{*T} Q^{*T}, \tag{3.12}$$

where $Q^* = \begin{bmatrix} V_0 & Q_A \end{bmatrix}$ and $R^* = \begin{bmatrix} I_P & V_0^T A \\ \mathbf{O}^T & R_A \end{bmatrix}$.

Finally, in order to obtain the updated eigenspace, an SVD on the three middle matrices is needed to get the p first principal components as follows

$$K_{n+N} \ \approx V_{(n+N)p} \Omega_p V^T_{(n+N)p}, \tag{3.13}$$

where

$$(V^{'}, \Omega_p) \leftarrow svd(R^* \Omega^*_{(p+2N)} R^{*T}, p) \tag{3.14}$$

and

$$V_{(n+N)p} = Q^* V^{'}. \tag{3.15}$$

Assume we have N observations to be introduced into a KPCA model of size n. While a batch updating of KPCA requires $O((n+N)^3)$ computation steps and the updating using Hoegaerts et al. (2007) needs to train N times the KPCA model, that is $O(pn^2), .., O(p(n+N)^2)$ operations, the proposed method uses only $O(p(n+N)^2)$ operations. Therefore, it makes a gain in computation, especially when $p \ll n$, which is usually the case for SPC problems. This fact makes the proposed control strategy useful for many complex fault detection problems and fast real time processes.

3.1.2 Group downdating of KPCA

In order to use the KPCA model for process control, many strategies of updating are possible. Depending on the objective, if for example the model needs to be only sensitive to large shifts, we may leave the training data grow for a certain time then an updating is performed. For a faster detection of abnormal process behaviour, we should integrate a forgotten factor by excluding the influence of the oldest observations. To downdate Ω_P and $V_{n,p}$ by excluding the influence of oldest sample of size h using Model (3.1), we need that the first h rows and columns of the projected new kernel K_n^* equal to zeros as shown below

$$K_n^* = \left[\begin{array}{cc} 0 & \mathbf{0} \\ \mathbf{0}^T & K_{n-h} \end{array} \right]. \tag{3.16}$$

Let $\alpha^T = \begin{bmatrix} K_{1(h+1)} & .. & K_{1n} \\ . & .. & . \\ K_{h(h+1)} & .. & K_{hn} \end{bmatrix}$ and $\beta = \begin{bmatrix} K_{11} & .. & K_{1h} \\ . & .. & . \\ K_{h1} & .. & K_{hh} \end{bmatrix}$, then we have

$$\begin{aligned} K_n^* &= K_n - \begin{bmatrix} \beta & \alpha^T \\ \alpha & O \end{bmatrix}, && (3.17) \\ &\approx V_{np}\Omega_P V_{np}^T + D\Delta D^T, && (3.18) \end{aligned}$$

$$D^T = \begin{bmatrix} \frac{\beta_1}{2}+1 & & \alpha_1 & & \\ \frac{\beta_1}{2}-1 & & \alpha_1 & & \\ 0 & \frac{\beta_2}{2}+1 & & \alpha_2 & \\ 0 & \frac{\beta_2}{2}-1 & & \alpha_2 & \\ 0 & . & & . & \\ 0 & . & & . & \\ 0 & 0 & \frac{\beta_h}{2}+1 & \alpha_h & \\ 0 & 0 & \frac{\beta_h}{2}-1 & \alpha_h & \end{bmatrix} \text{ and } \Delta = I_N \otimes \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{3.19}$$

where $\alpha_{i=1:h} = \begin{bmatrix} K_{i(i+1)} & \dots & K_{in} \end{bmatrix}$ and $\beta_{i=1:h} = K_{ii}$. Now, let denote $\mathbf{O}$ a $p \times (2h)$ matrix of zeros, then we approximate the new kernel matrix by

$$K_n^* \approx \begin{bmatrix} V_{np} & D \end{bmatrix} \begin{bmatrix} \Omega_P & \mathbf{O} \\ \mathbf{O}^T & \Delta \end{bmatrix} \begin{bmatrix} V_{np}^T \\ D^T \end{bmatrix}, \tag{3.20}$$

$$\approx V^*_{(n)(p+2h)} \Omega^*_{(p+2h)} V^{*T}_{(n)(p+2h)}. \tag{3.21}$$

Then, to perform downdating, the same procedure followed in KPCA updating is used to obtain V_{np} and Ω_p.

3.1.3 Downsizing KPCA model

In order to downsize the KPCA model by eliminating the first h rows of the obtained V_{np}, that are approximately equal to zero, and to preserve the orthogonality of the eigenvector, a QR decomposition of $V_{(h+1:n)p}$ that costs a negligible O(2np^2) operation is performed, such that

$$[V'_{(n-h)p}, R] = qr(V_{(h+1:n)p}). \tag{3.22}$$

To absorb the effect of the QR transformation, the eigenvalues matrix is adapted by $\Omega'_p = R\Omega_p R^T$.

3.2 Complexity analysis of the block updating algorithm

To highlight the gain of using the proposed algorithm, suppose we need to integrate N observations into an initial model of a size $(n \times p)$. While a batch updating of KPCA using a standard SVD would require $22(n + N)^3$ operations (Golub & Loan (1996)), the updating using Hoegaerts et al. (2007) needs to repeat N times a recursive computation cost of order $O(n^2), O((n+1)^2)...O((N+n)^2)$. The difference with the proposed algorithm is that instead of repeating each step N times with a recursive updating of the model size n, the algorithm integrates N observations at one time. For example, using algorithm of Hoegaerts et al. (2007), the calculation of the part of A that is orthogonal to V_0 of Equation (3.10), needs $(8p+2)n+(8p+$

Table 3.1: The Computation cost to update a model with N observations using block updating

Equation	Computation Cost
$(I_{n+N} - V_0 V_0^T)A$	$8npN + 2Nn$
$qr((I_{n+N} - V_0 V_0^T)A)$	$16(n+N)^2N + \dfrac{16N^3}{3} - 8(n+N)N^2$
$R^*\Omega^*_{(p+2N)}R^{*T}$	$4(p+2N)^3$
$SVD(R^*\Omega^*_{(p+2N)}R^{*T})$	$22(p+2N)^3$
$Q^*V^{'}$	$2(2N+p)(n+N)p$

$2)(n+1) + ... + (8p+2)(n+N-1)$ operations, while performing this step using the proposed block updating uses only $(8p+2)Nn$ operations.

The steps to calculate the total computation cost of the proposed algorithm are as follows: Performing efficient matrix multiplication, the calculation of the part of A that is orthogonal to V_0 of Equation (3.10) needs $(8p+2)nN$ operations. Then, performing the QR-decomposition of Equation (3.11) requires $6(n+N)^2N + \frac{16N^3}{3} - 8(n+N)N^2$ operations (Golub & Loan (1996)). Thereafter, calculation of the product of the three middle matrix of Equation (3.12) would need $4(p+2N)^3$ operations. Then, an application of a small SVD in the obtained matrix of Equation (3.14) would require only $22(p+2N)^3$ operations. Eventually a rotation of the eigenvector using Equation (3.15) requires $2(2N+p)(n+N)p$ operations. Adding these numbers of operations, we obtain the total computational cost present in Table 3.1.

This algorithm allows gain of computation only when a small number of the dominant principal components are updated. Because the amount of computation gain, of using the block updating method instead of using batch updating with SVD or single updating of Hoegaerts et al. (2007), is not clear, Figure 3.1 provides comparisons of the computation cost for different algorithms using different numbers of principal components p.

It is clear from Figure 3.1 that for a reduced number of principal components (p less than half of n) imply an important reduction in computation cost compared to the SVD method and single updating. Moreover, Figure 3.2 shows that block updating of KPCA reduces computation cost as compared to single updating. However, this gain in computation cost is at the expense of an approximated solution of the eigenvalues and eigenvectors. To analyze

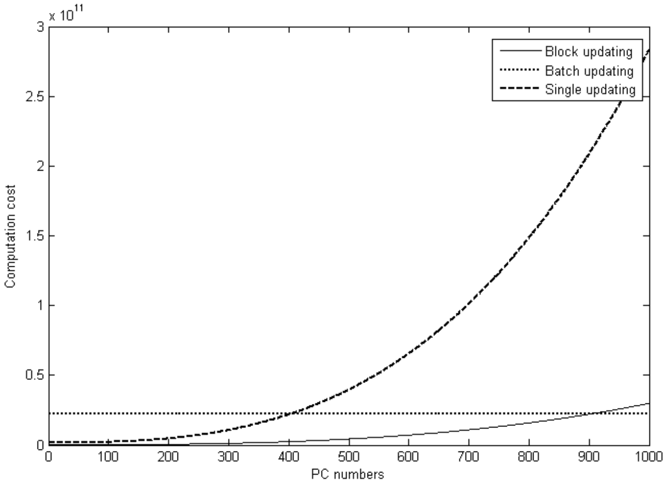

Figure 3.1: Comparison of computation cost for different numbers of principal components p with n equal to 1000 and N equal to 10

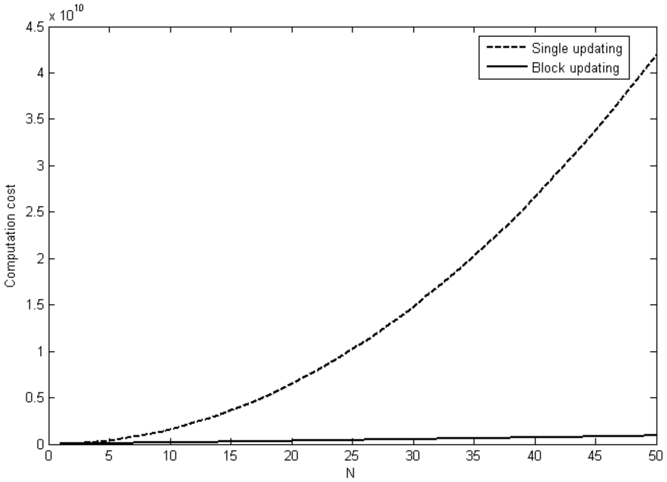

Figure 3.2: Computation cost in relation to the size of data block N with n equal to 1000 and p equal to 10

the performance of the proposed algorithm in terms of accuracy next section outlines results obtained using a benchmark data.

3.3 Comparison of block updating based on benchmark study

In this section, based on a benchmark dataset, we present an analysis of the proposed algorithm in terms of accuracy and speed. The dataset is called Abalone (Blake and Merz, 1998) and it allows predicting the age of abalone from physical measurements. The data set has 4177 instances and 7 exploratory variables. The radial kernel is used with a parameter sigma equal to 5. Because handling the eigenproblem with batch SVD for a 4177×4177 kernel matrix is difficult, only a set of 3000 observations is used to analyze the different methods. Even though 5 principal components cover more than 90 % of the total variance, as a caution, we select a number of principal components equal to 20, which explain more than 99 % of the total energy. The eigenvector problem starts with an initial model based on training of the first 500 instances and tracking is made by introducing 10 observations into each updating step. In order to compare the speed of different methods to update the KPCA model, Figures 3.3 and 3.4 provide the time needed to update the model when new data are available. The experiment is conducted in a Matlab environment with a computer that has 1.75 GB RAM memory and 2.71 GHz processor. As comparison to batch calculation using standard SVD, the gain of computation time increases as the size of the training data becomes bigger. Also the time needed by block KPCA to update the eigenproblem is more stable than when using batch SVD. As comparison to single updating, we notice that Block SVD is approximately 5 times faster than using single updating. This fact is useful in many applications where the updating procedure needs to be processed with other steps in a short time.

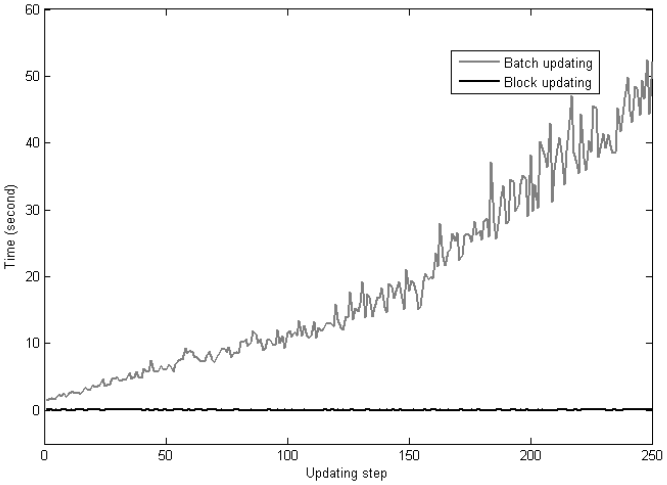

Figure 3.3: Time needed to update the KPCA model using batch SVD and block updating

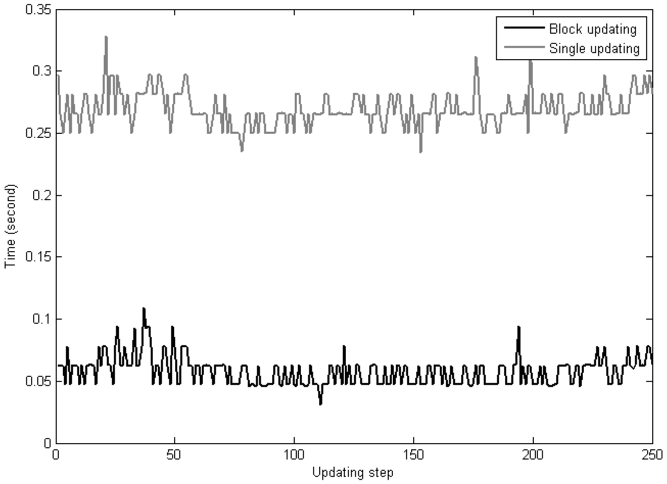

Figure 3.4: Time needed to update the KPCA model using single updating and block updating

3.3.1 Relative error analysis

In order to assess the performance of the proposed method in terms of approximation accuracy, first, we use an overall measure of accuracy which is the relative error calculated by the distance between the constructed matrix obtained by KPCA model and the original data. This measure is calculated by the matrix Frobenius norm, as follows

$$RE = \frac{\left\| K - \tilde{K} \right\|_F}{\|K\|_F}, \tag{3.23}$$

where K is the original matrix and $\tilde{K} = V\Omega_p V^T$ is a rank$-p$ approximation using KPCA model. This measure is computed for different algorithms and comparison of the relative error of the proposed algorithm with that of single updating and batch using SVD is given in Figure 3.5. We notice that the relative errors are approximately the same with a maximal relative error equal to 1.5 10^{-4}. This means that different methods represent relatively well the original data with a good tracking for single updating and block updating.

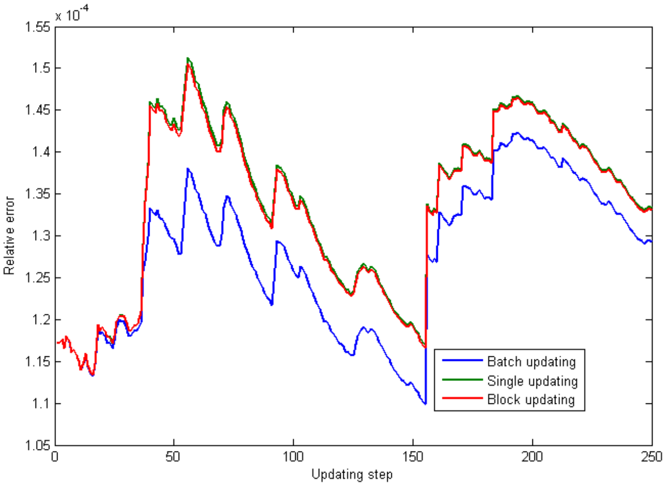

Figure 3.5: Comparison of reconstruction errors of different updating methods

Another measure of accuracy is the Relative Error between the eigenvalues computed by batch SVD and those computed using block updating. This indicator is calculated as follows

$$RE_\lambda = \frac{\lambda - \tilde{\lambda}}{\lambda}. \tag{3.24}$$

where λ and $\tilde{\lambda}$ are the eigenvalues computed respectively by SVD and block updating.

This measure is controlled over 250 iteration steps, for the 10 first dominant components. From Figure 3.6 we notice that the dominant principal components are well approximated, with maximal relative errors rate equal to 3.5 10^{-6} . Also, we note that the accuracy increases for the first principal components.

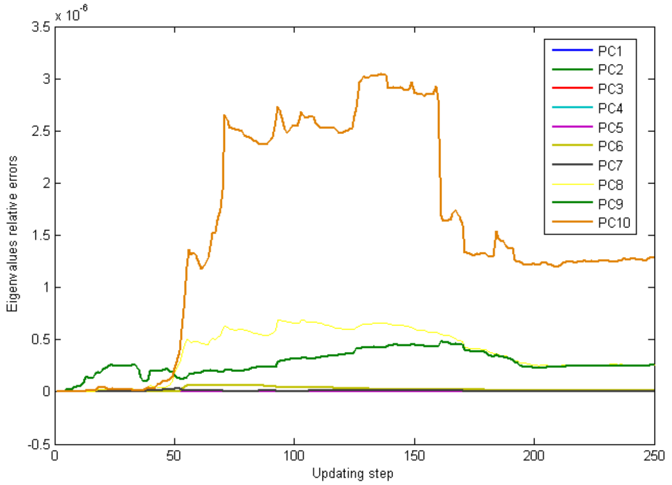

Figure 3.6: Relative errors of the first 10 principal components calculated by the difference between eigenvalues obtained by batch SVD and block updating

3.3.2 Orthogonality analysis

In order to test the orthogonality of the obtained principal components using block updating for several iterations, a criterion of the orthogonality of the eigenspace can be calculated by the scalar orthogonality measure, obtained as follows

$$E_{\perp} = \left\| I - V^T V \right\|_2 .$$

A zero value indicates a perfect orthogonality. During all the updating process, the orthogonality of the KPCA model using block updating is very well-preserved with values of scalar orthogonality measure between $10^{-8}$and 10^{-12}. This fact means that the obtained components using block updating of KPCA are, over several iterations, approximately orthogonal.

3.4 Conclusion

This chapter discusses a proposed block updating algorithm for training of the dominant components of KPCA model. The updating method can provide a reduced computation cost in comparison to batch SVD and single updating of Hoegaerts et al (2007). This gain facilitates extension of the use of KPCA method to many machine learning based applications, where fast updating of a large-scale model is required. Analysis of the accuracy and stability of the proposed updating method shows that it provides a good tracking of the original matrix with a small reconstruction error. Also in comparison to batch KPCA, the proposed algorithm has a good approximation of the eigenvalues with a good preservation of eigenvectors orthogonality.

4

Assessment of the proposed chart using benchmark studies

An issue concerning the proposed adaptive monitoring technique is to evaluate it with respect to other PCA process control strategies. We propose to compare performance of Adaptive window Kernel Principal (AKPCA) with Moving Window KPCA (MWKPCA) and Adaptive PCA (APCA). APCA is based on linear PCA and MWKPCA is based on KPCA model. In contrast to the proposed AKPCA control chart, APCA and MWKPCA have fixed model sizes and allow handling of a single observation at a time. Section 1 provides the way to tune control chart parameters. Section 2 investigates performances of different charts using a simulated multivariate nonlinear process. Section 3 applies the developed charts to monitor a complex industrial chemical process called Tennessee Eastman (TE) process. Section 4 provides the conclusion of this chapter.

4.1 Parameters selection

In order to implement the different control charts, several parameters need to be tuned. These parameters are the sigma of the radial kernel function, the number of Principal Components (PCs), the risk level of the control charts and the size of the initial PCA models.

First of all, for KPCA based charts, the value σ of the radial kernel function is tuned based on the method of Park and Park (2005), which proposes to select $\sigma = C \cdot Averd$, where $Averd$ is the mean distance between all observations in feature space and C is a predetermined value. In this study, the C value is set to be equal to the square root of the number of process variables. To select the number of PCs, this study uses two of the most used procedures named cross-validation and Cumulative percent variance. The first technique divides the dataset into a training dataset that allows obtaining the eigenproblem solution and a test dataset that allows calculating the reconstruction error of the PCA model. The number of PCs which provides most contribution to the minimization of the reconstruction error is selected. The cross-validation procedure is more adequate to select the number of PCs for Q control chart because this statistic allows monitoring of the reconstruction error. However, the Cumulative percent variance procedure selects the p first eigenvalues that capture 95 % of the total variance. This procedure is used to select the number of PCs for T^2 control chart since this statistic measures the distance obtained by projecting observations to the most important PCs. The values of α levels of T^2 and Q based control charts are set in a way to provide approximately equally small false alarm rates. Concerning the size of the initial PCA models and the size of the sample used to calculate the initial control chart limits, while Li et al. (2000) used 22 % of the data, Liu et al. (2009) used 30 % of the available data to train and calculate initial PCA models and control charts limits. Also in this study, for each process, 30% of the data is used for construction of the different control charts, such that 20 % of the available sample is used to train initial PCA models and 10 % to calculate initial control limits. For AKPCA chart, the size of the window can vary between 80 % and 120 % of the initial KPCA model size.

Evaluation of different control strategies is given by reporting the degree of accuracy of each method in detecting the true out-of-control situations and avoiding false alarms. Performance evaluation of the accuracy can be reported by using the false alarm rate (Type I error rate) and the detection rate criterion. The first statistic gives information about the robustness of the adopted method against normal system changes whilst the second statistic gives information about the sensitivity and efficiency of detecting faults.

In order to assess different control strategies, we use two simulated benchmark case studies. The first simulated process is a nonlinear multivariate non-stationary system and the second process is named Tennesee Eastman process which contains 41 state variables that exhibit several nonlinear dynamics and multistage phases. The description of these processes as well as performances of different control charts are presented in next sections.

4.2 Multivariate nonlinear simulated process

The simulated multivariate nonlinear process is described as follows (Zhang and Yang (2000)):

$$\begin{cases} y_1 = 0.5t^2 - 2t + 0.5 + \zeta \\ y_2 = t^2 + t + \sin(\pi t) + \zeta \\ y_3 = 2t^2 - t - 2\cos(\pi t) + \zeta \end{cases} \tag{4.1}$$

where ζ is a random noise and $t \in [-1, 1]$.

An illustration of the simulated process is shown in Figure 4.1. In this study, 20 normal samples, each containing 500 observations, are generated. To test the detection performance of the different procedures, this study simulates 7 faults introduced after observation 200. These faults are presented in Table 4.1.

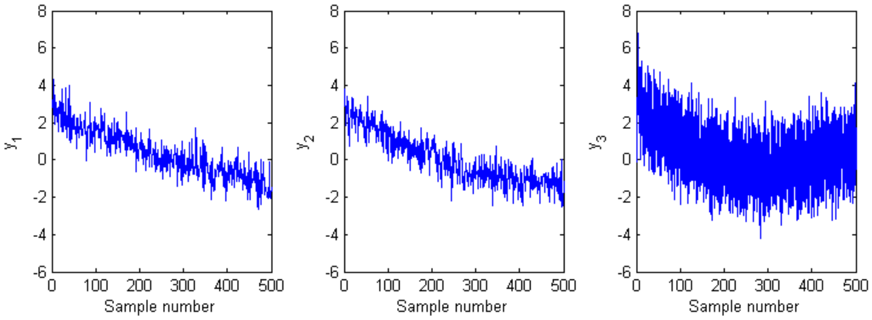

Figure 4.1: Illustration of the simulated multivariate nonlinear process

For Q and T^2 based APCA control charts, 2 PCs are selected based on the cross-validation method and the Cumulative percent variance. This value contributes to more than 95 % in the minimization of the reconstruction error and explains more than 95 % of the total variance. For KPCA based models, the value of sigma is set to 5 and the number of PCs is set to

Table 4.1: List of the simulated faults for the multivariate nonlinear process

Fault	Description
1	Increase of y_1 by $0.02k$. (k is the sample number)
2	Increase of y_2 by $0.02k$.
3	Add to y_3 a random effect $N(0,1)$.
4	$y_1 = y_1 + 2t$; $y_2 = y_2 + 0.02k$. (t is the time index)
5	$y_1 = y_1 + N(0,1)$; $y_3 = y_3 + 2t^2$.
6	$y_2 = y_2 + 0.02k.$; $y_3 = y_3 + 2t$.
7	$y_1 = y_1 + 2t$; $y_2 = y_2 + 0.02k$; $y_3 = y_3 + 2t$.

10 for Q control chart and 4 for T^2 control chart. Introduction of observations into the AKPCA model is made by blocks of 5. First, in order to show that Batch PCA and KPCA models are not appropriate to monitor of non-stationary processes, Figure 4.2 shows monitoring performances of both methods when the process is operating under normal condition. The false alarm rate provided by Batch PCA and Batch KPCA are respectively 48 % and 90 %. Therefore analysis of the detection performance of these methods is not performed, since these control charts are not adequate to monitor non-stationary processes. However, as shown in Figure 4.3 and in contrast to the fixed models, applying adaptive PCA based control charts to the same data set, allows better capabilities of adaptation to nonlinear non-stationary behaviour of the process. It should be noted that the control limits Q_{limit} and T_{limit} of the different adaptive charts are slightly variable over time due to the fact that they are calculated recursively as described in Section 4. This fact allows a better adaptation to the condition in which the process is operating. To analyze detection capabilities for the different adaptive procedures, monitoring performances of APCA, MWKPCA and AKPCA are summarized in Table 4.2.

First, we notice that Q based adaptive charts have overall better fault detection capabilities than T^2 based control charts, especially for Kernel based procedures. This result confirms the analysis of Lee et al. (2004) which states that T^2 of KPCA is less reliable than Q chart. Moreover, detection rates of AKPCA are higher than that of APCA and MWKPCA control charts. As an example, when Fault 4 is introduced APCA (Q), MWKPCA (Q) and AKPCA (Q) detects respectively on average 59 %, 45 % and 96 %, of the total faults. Therefore, this result confirms that AKPCA is more

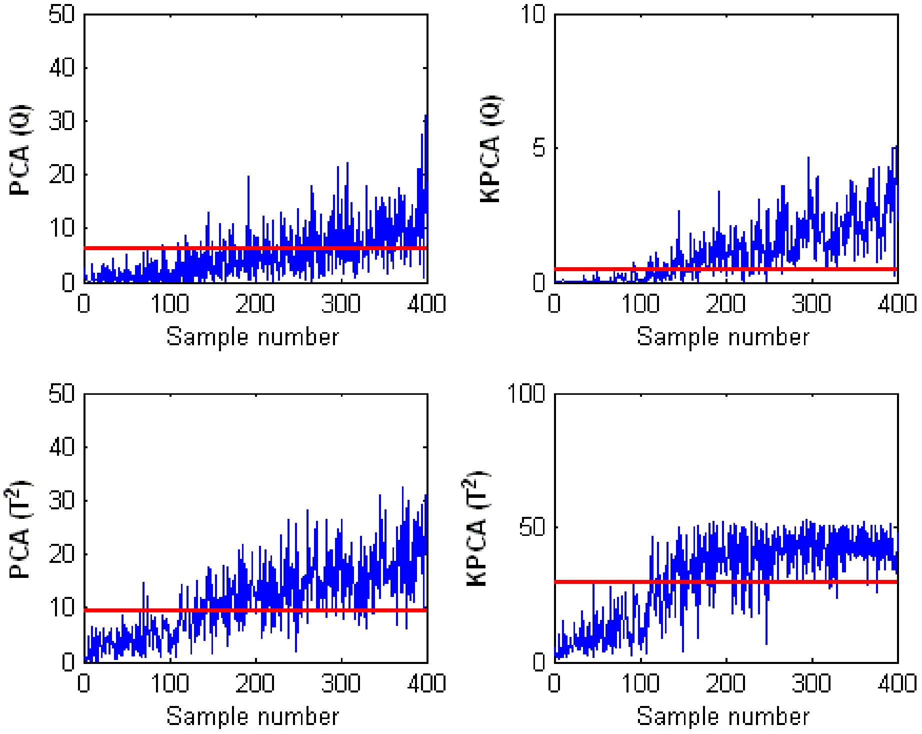

Figure 4.2: Batch PCA and KPCA monitoring charts in the case of normal operating condition. The blue and red lines represent respectively the value of the control statistic and its limit

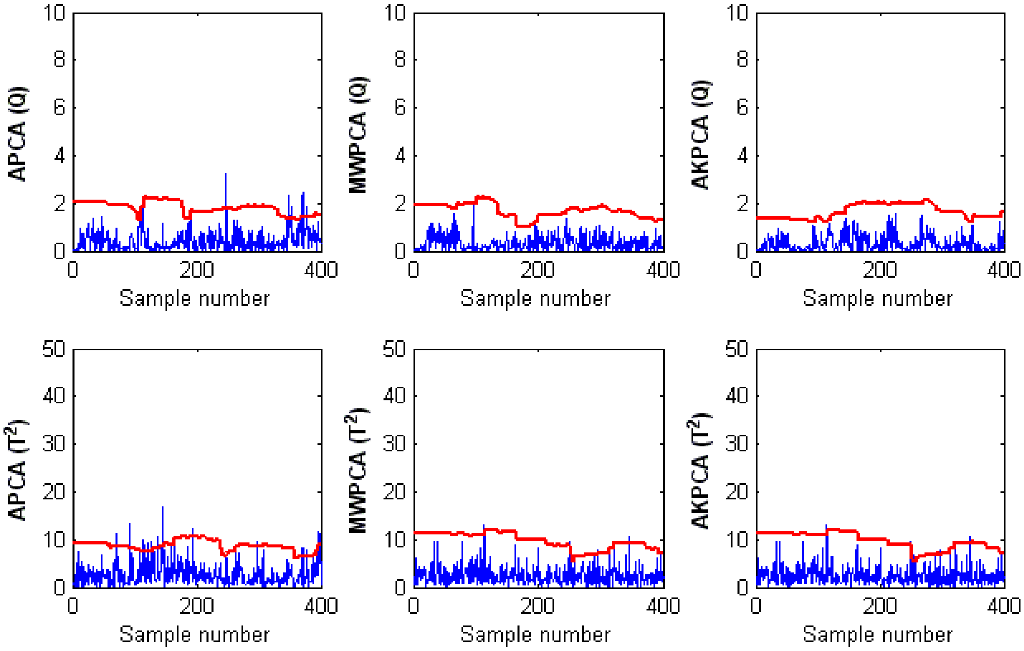

Figure 4.3: Adaptive PCA and KPCA based monitoring charts in the case of normal operating condition

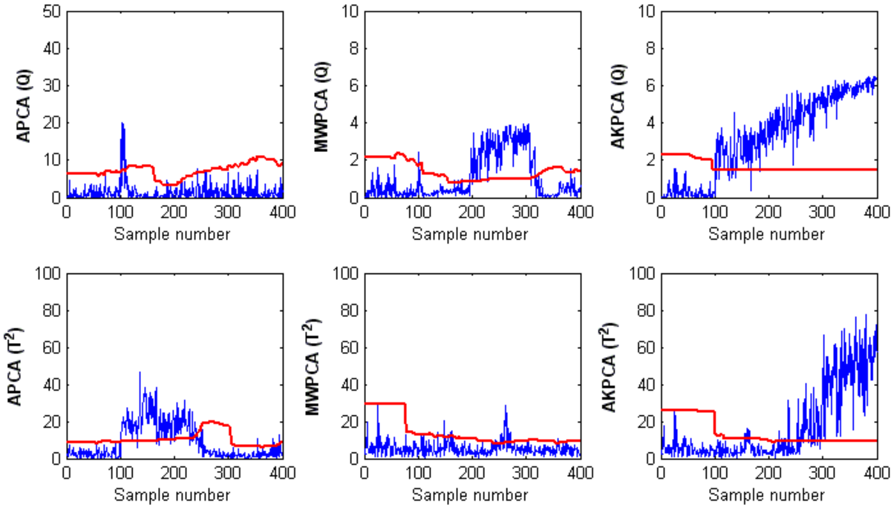

Figure 4.4: Adaptive PCA and KPCA based monitoring charts in the case of Fault 4, where the fault starts after observation 100

robust in detecting disturbances. This can be explained by the fact that AKPCA model does not update the model after one observation but after a block of data is tested. Thus, the model has a better ability to overcome contamination by out-of-control observations. This fact can be observed in Figure 4.4, where APCA (Q) and MWKPCA (Q) charts first detect the fault, but after testing and introducing certain observations into the models, detection of this fault is not more possible since it is assimilated as a normal operating change. Moreover, the extreme frequency of updating APCA and MWKPCA models, which is made after each new observation, can imply a decrease in the sensitivity to smaller shifts. In contrast, block AKPAC allows a small delay for updating which can improve detection of this kind of shifts.

Table 4.2: Monitoring results of APCA, MWKPCA and AKPCA charts for several faults

	APCA		MWKPCA		AKPCA	
	T^2	Q	T^2	Q	T^2	Q
False alarm rate	0.02	0.02	0.03	0.02	0.03	0.02
Detection rate						
Fault 1	0.62	0.76	0.08	0.41	0.54	**0.99**
Fault 2	0.31	0.56	0.06	0.43	0.44	**0.99**
Fault 3	0.02	0.01	0.07	0.01	0.05	**0.09**
Fault 4	0.53	0.59	0.10	0.45	0.40	**0.96**
Fault 5	0.02	0.02	**0.10**	0.03	**0.10**	0.01
Fault 6	0.37	0.46	0.05	0.30	0.42	**0.94**
Fault 7	0.52	0.73	0.10	0.42	0.41	**0.98**
Mean Detection	0.34	0.45	0.08	0.29	0.34	**0.71**

4.3 Tennessee Eastman process

The Tennessee Eastman (TE) process is a benchmark simulation model of a complex industrial chemical process proposed by Downs and Vogel (1993). The system contains five major units: a reactor, a condenser, a recycle compressor, a separator, and a stripper. The process has 41 state variables that are used for monitoring. About half of the measured variables are component compositions, available at discrete sampling intervals. The remaining 22 measured variables are available at significantly higher sampling frequency (0.01 h). The original process is open-loop unstable and therefore a control strategy must be used to avoid process shutdown. This study uses the simulation program that is available at Ricker's home page (Ricker, 2008). In this study, a set of 1000 observations are simulated under normal operating condition. In addition, more process data are generated with 20 fault cases, introduced after observation 500. Table 4.3 presents these faults.

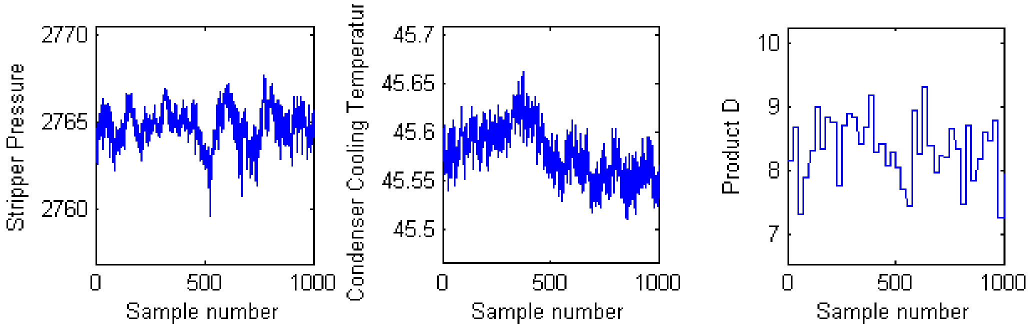

Figure 4.5: Illustration of the non-stationary behaviour for some variables of Tennessee Eastman process

As shown in Figure 4.6 and based on a cross-validation and Cumulative percent variance methods, 7 and 30 PCs are selected for respectively Q and T^2 based APCA control charts. For KPCA based procedures, for a sigma value equal to 80, 30 PCs are selected for both Q and T^2 based charts. As mentioned previously, 30 % of the data are used to construct the different control charts. Introduction of observations into the AKPCA is made by blocks of size 10.

Table 4.4 summarizes results of different methods for the Tennessee Eastman process. As Table shows, overall, AKPCA (Q) control chart provides better results in comparison to APCA and MWKPCA control charts. Moreover, there is a clear improvement of kernel based PCA procedures. Indeed, the detection rate increased from MWKPCA(Q) to AKPCA(Q) from 31 % to 50 %. Moreover, when the T^2 statistic is used for Kernel based PCA charts, monitoring reliability of detecting abnormalities decreases in comparison to

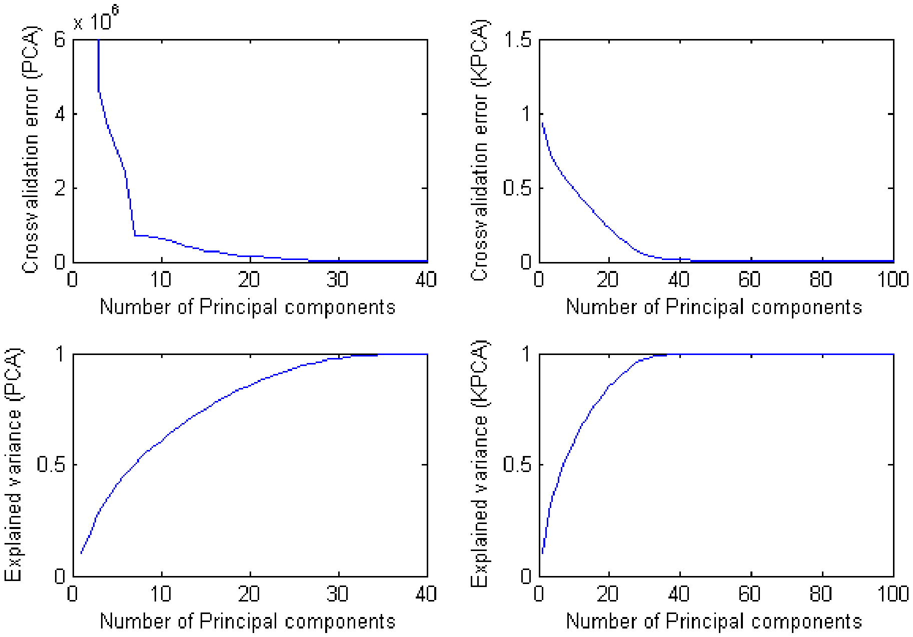

Figure 4.6: Cross-validation errors and Percent of explained variance for different numbers of Principal Components

Table 4.3: Process faults for Tennessee Eastman process

Fault	Description	Type
1	A/C feed ratio, B constant	Step
2	B composition, A/C ratio constant	Step
3	D feed temperature (Stream 2)	Step
4	Reactor water's temperature	Step
5	Condenser 's water temperature	Step
6	A feed loss	Step
7	C header pressure loss	Step
8	A, B, C feed composition	Random
9	D Feed Temperature	Random
10	C feed temperature	Random
11	Reactor's water temperature	Random
12	Condenser's water temperature	Random
13	Reaction kinetics	Slow drift
14	Reactor's water valve	Sticking
15	Condenser's water valve	Sticking
16	Unknown	-
17	Unknown	-
18	Unknown	-
19	Unknown	-
20	Unknown	-

the Q statistic. However, an interesting aspect of MWKPCA and AKPCA is that when the Q statistic provides poor detection results, the T^2 exhibits a better performance. This fact can be observed for Fault 2, 4 and 15, where T^2 based MWKPCA and AKPCA charts provide slightly better results. Figure 4.7 illustrates the performance of Q and T^2 based PCA charts for, respectively, Fault 9 and Fault 4. Though the fact that most charts detect the fault introduced after sample 300, APCA and MWKPCA charts assimilate the fault as a normal behaviour and after certain alarms the fault is no more detectable because some faulty observations are introduced in PCA models.

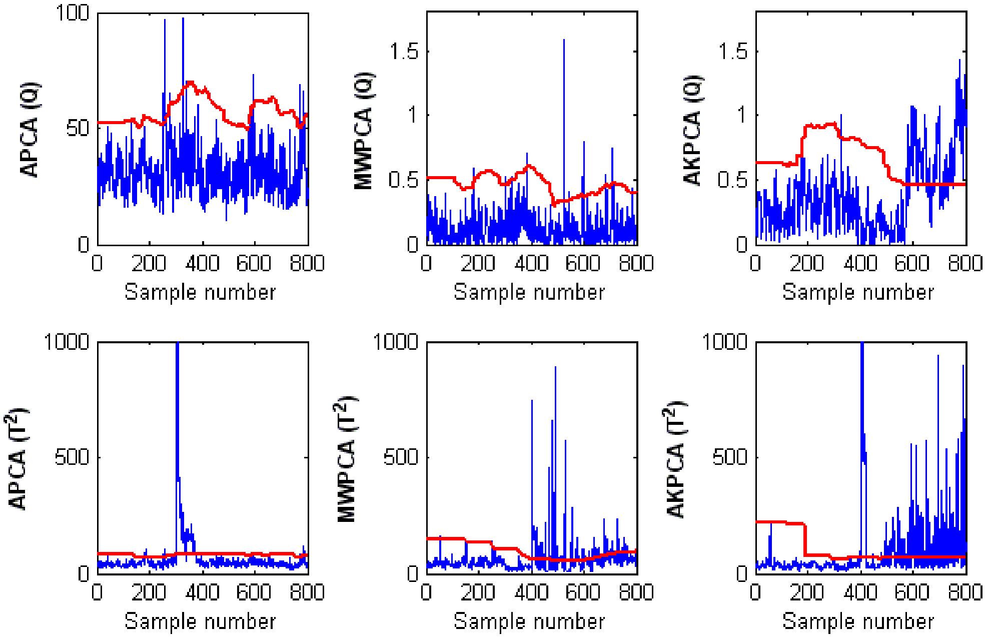

Figure 4.7: Monitoring chart in case of Fault 9 (Q statistic) and Fault 4 (T^2 statistic)

Table 4.4: Analysis of APCA, MWKPCA and AKPCA charts for several faults

	APCA		MWKPCA		AKPCA	
	T^2	Q	T^2	Q	T^2	Q
False alarm rate	0.03	0.03	0.04	0.03	0.03	0.03
Detection rate						
Fault 1	0.97	**0.98**	0.11	**0.98**	0.08	0.97
Fault 2	0.03	0.04	0.15	0.04	**0.22**	0.06
Fault 3	0.03	0.03	0.06	0.02	**0.18**	0.13
Fault 4	0.16	0.15	0.26	0.17	**0.42**	0.13
Fault 5	**1**	**1**	0.52	**1**	0.46	**1**
Fault 6	**1**	**1**	0.22	**1**	0.09	**1**
Fault 7	**0.75**	0.47	0.26	0.47	0.08	0.63
Fault 8	**0.84**	0.04	0.21	0.05	0.29	**0.84**
Fault 9	0.06	0.03	0.10	0.06	0.05	**0.35**
Fault 10	0.85	0.02	0.10	0.05	0.05	**0.86**
Fault 11	0.50	0.50	0.24	**0.58**	0.06	0.45
Fault 12	**0.92**	**0.92**	0.42	**0.92**	0.23	**0.92**
Fault 13	**0.45**	0.44	0.07	0.04	0.01	0.22
Fault 14	0.29	0.14	0.06	0.11	0.09	**0.53**
Fault 15	0.03	0.02	0.06	0.03	**0.30**	0.04
Fault 16	0.06	0.02	**0.08**	0.03	0.04	0.07
Fault 17	0.27	**0.52**	0.06	0.50	0.13	**0.52**
Fault 18	**0.26**	0.23	0.16	0.04	0.05	0.03
Fault 19	**0.75**	0.14	0.23	0.09	0.05	**0.75**
Fault 20	0.20	0.09	0.07	0.06	0.02	**0.50**
Mean Detection	0.47	0.34	0.17	0.31	0.15	**0.50**

4.4 Conclusion

Analysis and comparison with batch models, as well as with adaptive PCA procedures, shows that first, Batch PCA and KPCA are unable to adequately control non-stationary processes since they are based on the use of fixed models for monitoring dynamic processes. Moreover, results show that the proposed AKPCA chart provides overall better detection results in comparison to AKPCA and MWKPCA. However, as stated by Lee et al. (2004), the enhancement of the detection capability is made essentially by using Q statistic. Also, this study shows that for certain cases, when the Q statistic provides poor detection results, T^2 exhibits better performance. Therefore, a combination of these statistics into one AKPCA chart could be an interesting topic for future research.

Part II

Support Vector Regression based control charts for non-linear autocorrelated processes

Traditional control charts require several assumptions and a fundamental one is that the process data should be independently distributed. However, with the advances made in process automation, violation of this assumption is frequent. In fact, high sampling collection often produces series of observations that are close enough to be dependent. As mentioned by Psarakis and Papaleonida (2007), autocorrelation is present in most continuous and batch process operations. Many studies reported the effect of the violation of the independence assumption on control charts and showed that, when this assumption is not satisfied, traditional models exhibit poor performances. Bagshaw and Johnson (1975) stated that, for Autoregressive AR(1) or a Moving Average MA(1) processes, incorrect conclusions can be drawn by using conventional CUSUM schemes. Harris and Ross (1991) discussed the impact of autocorrelation on the performance of CUSUM and EWMA charts, and showed that it affects the average and median run length. Alwan and Roberts (1995) studied several chart applications and showed that they displayed incorrect control limits due to violation of serial correlation which affects the effectiveness of control charts. Noorossana and Vaghe (2006) showed that breaking the independency assumption affects the average run length (ARL) of the control charts and makes them unreliable. Psarakis and Papaleonida (2007) asserted that even small levels of autocorrelation can have big effects on the statistical properties of conventional control charts and may cause substantial increase in the average false alarm rate and a decrease in the ability of detecting changes on the process.

To deal with the problem of autocorrelation, one approach is to filter out autocorrelation by a time series model and use residuals for control. In fact, if the time series model is accurate, residuals would be statistically uncorrelated. Then, assumptions of traditional quality control charts are satisfied and conventional charts will be appropriate. The most widely used time series methods are autoregressive integrated moving average (ARIMA) models. Using AR(1) based control charts to monitor a sequential injection analysis, Callao and Rius (2003) show that residuals control charts provide better understanding of the system behaviour over time and efficient detection abilities. Loredo et al. (2002) presented a regression adjustment scheme to monitor autocorrelated processes and showed that the residuals based control is better than observation control charts in detecting a mean shift in data.

However, time series modelling may be often awkward in actual applications because of the specification problem (Zhang, (1998), Jiang et al. (2000)). Stone and Taylor (1995) reported some industrial processes that exhibit autocorrelation and which are not adequately handled by standard time series models. Many researchers seem to agree that residuals charts do not have the same properties as traditional charts and that their ability to detect a shift depends on the model that is assumed to describe the data appropriately. Longnecker and Ryan (1992) investigated AR(1), AR(2) and ARMA(1,1) models for a residuals X-chart and pointed out that the latter chart may have poor capability to detect the process mean shift. Apley and Tsung (2002) state that, if the model is inaccurate, residuals will not be uncorrelated, and the in-control ARL of a residual-based CUSUM may be substantially shorter than what is intended. Moreover, basic time series methods require a predefined structure of the process. Therefore, application of such models for process control needs a lot of expertise of the process on hand. Also, basic methods allow only handling of linear processes and are not capable of modelling nonlinear processes. In fact, to handle nonlinear systems different differential equations are provided. However, Shi et al. (2001) stated that there is a problem in finding analytical solutions to describe phenomena of interest.

To overcome this problem, researchers have presented great interest in Artificial Neural Networks (ANN) for quality monitoring because of the capability of ANN to determine the process structure from the data and to model nonlinear systems. Dooley and Guo (1992) applied ANN models to detect positive changes in the process mean and variance. While Hwarng and Hubele (1993) integrated several ANN models for construction of Shewhart X control charts for identification of out control situations. Recently, Jamal et al. (2007) have introduced an ANN based model to construct residuals Multivariate CUSUM chart for multivariate AR processes and showed that it performs better than MCUSUM. Pacella and Semeraro (2007) proposed a simple NN model to control autocorrelated processes and showed that it performs well for several mean shifts. Despite their advantages, most ANN methods have some drawbacks that can affect their performance such as the problem of optimal parameters selection. Indeed, ANN have several parameters to tune such as transfer functions, number of nodes, number of layers, number of epoches, momentum rate, learning rate, etc. This implies a certain difficulty to construct an ANN model. Also, ANN is prone to the problem of overfitting, which means that the model can provide bad generalization performance on data not used in the training phase. Moreover, ANN does not guarantee global optima in the optimization phase. In fact Ryan (2000) claims that when ANN is repeatedly run on the same dataset different results are produced.

As an alternative method to deal with this shortcoming, several researchers suggest the use of Support Vector Regression (SVR) (Thissen et al. (2003) and Sato et al. (2008)). This method allows learning the process structure directly from the data and no predefined process structure is required. The model can handle systems that present linear or nonlinear time series. In comparison to ANN, the optimal solution provided by SVR model is global and therefore the model is stable when it is reapplied to the same dataset. Also, the model includes a generalization term that allows overcoming the problem of overfitting. Eventually, few parameters need to be tuned and it is easy to construct an SVR model that performs well. To confirm their attractive properties, Müller et al. (1997) used SVR to time series forecasting and demonstrated its superiority to ANN. Tay and Cao (2001) and Ping-Feng and Chih-Sheng (2005) showed that SVR has a good prediction performance when applied to financial assets. Thissena (2003) applied SVR to complex time series and showed that it outperforms ARMA and ANN models. Sato et al. (2008) tested SVR for nonlinear processes and concluded that the approach is adequate even for small samples. Therefore, this chapter proposes the use of SVR method to construct residuals control charts. Also, the majority of process control articles focus on monitoring the mean of linear time series, further characteristics of a time series have not been enough considered. In this study we propose to extend application of SVR time series control charts to nonlinear processes. In order to detect not only mean shifts but also other parameters shifts, several control charts are investigated in the next chapter.

5
SVR residuals control charts

This chapter is organized as follows: Section 1 focuses on presenting the principal of using SVR method for time series estimation. In this section, the way to estimate Autoregressive AR models is presented. Then, Section 2 presents most used univariate and multivariate residuals control charts along with their properties. Section 3 proposes residuals Support Vector Control charts and the way they are evaluated. Section 4 resumes this chapter.

5.1 Time series regression using SVR

Time Series models forecast future values of time series variables by extrapolating trends and patterns in the past values of the series. There are many models used for time series and the most important one is AR model. This section presents the way these models are estimated using SVR method. The problem of SVR for AR(p) time series model estimation is, given a set of observations $Y = (y_1, ..., y_n) \in \mathbb{R}^n$, find under a specific loss function a function $f : \mathbb{R}^p \longrightarrow \mathbb{R}$. For linear functions, f is defined as follows,

$$y_t = f(y_{(t-1)}, .., y_{(t-p)}) = b + < w, x_t >, \tag{5.1}$$

where $x_t = [y_{(t-1)}, .., y_{(t-p)}]^T$, w is a vector of dimension p and b a bias term.

The measurable function needs to minimize the following *risk functional* $R(f)$ which measures the average amount of error probability associated with the estimator f.

$$R(f) = \int L(y_t, f(x_t))dP(x_t, y_t).$$

where L is the loss function.

For SVR case, the particularity of the loss function is such that errors of points that do not exceed a given distance ε are not considered as errors and therefore they do not contribute to any loss to the objective function. For this reason, the loss function is insensitive to these points. Vapnik (1998) used the following insensitive loss

$$L(y_t, f(x_t)) = max\{0, \mid y_t - f(x_t) \mid - \ \varepsilon\}, \tag{5.2}$$

where ε is called the *insensitivity margin,* errors below positive ε are not penalized.

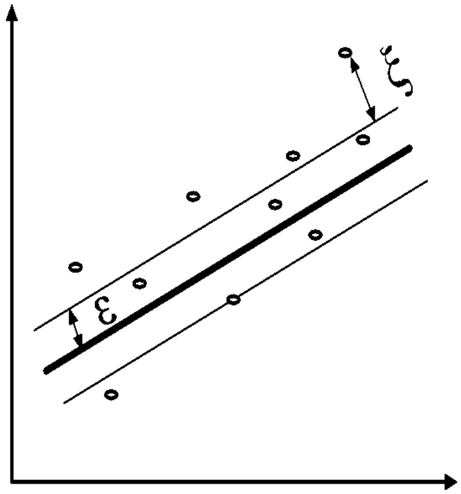

Figure 5.1: Error lying outside the ϵ-insensitive band around the regression function

But, the problem is that there could be infinitely many linear regressors that estimate the training set with reduced errors. Consequently, the task is to find the best one taking into account its capacity of generalization. That is, the function has to perform well on unseen data not used in training. Therefore, the regression algorithm is developed to reach a small regularized risk $\hat{R}$ using $\|w\|^2$ as a regularization parameter of the function complexity. This last term restricts the set of admissible functions, in order to have a good capacity of generalization. Then, the algorithm needs to minimize the following objective

$$\frac{1}{2}\|w\|^2 + C \cdot (\nu \cdot \varepsilon + \hat{R}(f)), \tag{5.3}$$

where the parameter ε is the ε-insensitivity margin, C and v are constants determining the trade-off between the error risk $\hat{R}$ and the complexity regulator $\|w\|^2$. The error risk is defined by

$$\hat{R}(f) = \frac{1}{n}\sum_{t=1}^{n} \max(|y_t - f(x_t)| - \varepsilon, 0). \tag{5.4}$$

Then, we have to solve the following constrained optimization problem

$$\begin{cases} \underset{(w,b)}{Min} \ \dfrac{\| w \|^2}{2} \ + \ C \cdot (\upsilon \cdot \varepsilon + \dfrac{1}{n}\sum_{t=1}^{n}(\xi_t \ + \xi_t^*)) \\ s.t. \\ \qquad (< w, x_t > +b) - y_t \leq \varepsilon + \xi_t \qquad \forall t = 1, ..., n \\ \qquad y_t - (< w, x_t > +b) \leq \varepsilon + \xi_t^* \qquad \forall t = 1, ..., n \\ \qquad \varepsilon, \ \xi_t, \ \xi_t^* \geq 0 \qquad \forall t = 1, ..., n \end{cases} \tag{5.5}$$

Solving the problem is finding parameters w and b. It is useful to transform formulation (5.5), called Primal, to the Dual form by Lagrangian transformation. There are two reasons for this transformation. First, the constraints are replaced by constraints on Lagrange multipliers themselves, which is much easier to handle. Second, the training data will only appear in the form of dot products between vectors. This is a crucial property which will allow us to generalize the procedure to the non linear case. We define Lagrangian of Equation (5.5) as follows

$$
\begin{aligned}
L(w,b,\varepsilon,\xi,\beta,\mu,\gamma) = \frac{\parallel w \parallel^2}{2} \; + \; C(\upsilon \cdot \varepsilon + \frac{1}{n}\sum_{t=1}^{n}(\xi_t \; + \xi_t^*)) \\
- \sum_{t=1}^{n}\beta_t\,(-(< w, x_t > +b) + y_t + \varepsilon + \xi_t \;) \\
- \sum_{t=1}^{n}\beta_t^*\,((< w, x_t > +b) - y_t + \varepsilon + \xi_t^* \;) \\
- \sum_{t=1}^{n}(\mu_t\xi_t + \mu_t^*\xi_t^*) - \gamma\varepsilon,
\end{aligned}
\tag{5.6}
$$

$$
\beta_t,\; \beta_t^*,\; \mu_t \text{ and } \; \mu_t^* \geq 0 \quad \forall\, t = 1,..,n. \tag{5.7}
$$

Then, the solution is given by the saddle point of the Lagrangian. The Lagrangian has to be minimized with respect to w, b, ε, μ_t and maximized with respect to the lagrange multipliers β_t, β_t^*, μ_t, μ_t^* and γ. Following up, at this saddle point, the following derivatives of L with respect to the primal variables must vanish,

$$
\frac{\partial}{\partial b}L \;=\; 0 \Longrightarrow \sum_{t=1}^{n}(\beta_t - \beta_t^*) = 0. \tag{5.8}
$$

$$
\frac{\partial}{\partial w}L \;=\; 0 \Longrightarrow \sum_{t=1}^{n}(\beta_t^* - \beta_t)x_t = w. \tag{5.9}
$$

$$
\frac{\partial}{\partial \varepsilon}L \;=\; 0 \Longrightarrow \sum_{t=1}^{n}(\beta_t + \beta_t^*) = C \cdot \upsilon - \gamma. \tag{5.10}
$$

$$
\frac{\partial}{\partial \xi_t}L \;=\; 0 \Longrightarrow C/n - \beta_t - \mu_t = 0. \tag{5.11}
$$

$$
\frac{\partial}{\partial \xi_t^*}L \;=\; 0 \Longrightarrow C/n - \beta_t^* - \mu_t^* = 0. \tag{5.12}
$$

Introducing results of the Lagrangian derivatives into Equation (5.6), we get the following problem that needs to be solved numerically,

$$\begin{cases} \underset{\beta_i,\beta_i^*}{Max} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\beta_i^* - \beta_i)(\beta_j^* - \beta_j) < x_i, x_j > + \sum_{i=1}^{n} y_i(\beta_i^* - \beta_i) \\ s.t. \\ \qquad \sum_{i=1}^{n}(\beta_i - \beta_i^*) \ = 0 \\ \qquad \sum_{i=1}^{n}(\beta_i + \beta_i^*) \ \leq C \cdot \upsilon \\ \qquad 0 \leq \beta_i \leq C/n \qquad\qquad \forall\ i = 1,..,n \\ \qquad 0 \leq \beta_i^* \leq C/n \qquad\qquad \forall\ i = 1,..,n \end{cases} \tag{5.13}$$

Then, the function f of Equation (5.1) takes the following form

$$y_t = \sum_{i=1}^{n}(\beta_i - \beta_i^*) < x_i, x_t > +b \tag{5.14}$$

In order to compute b, the Karush Kuhn Tucker (KKT) theorem is used. According to KKT, only the Lagrange multipliers $\beta_t^\prime s$ and $\mu_t^\prime s$ that are non zero at the saddle point correspond to constraints which are exactly saturated. These conditions state that at the point of the solution we have

$$\mu_t \xi_t \quad = \quad 0 \qquad\qquad \forall\ t = 1,..,n \tag{5.15}$$

$$\mu_t^* \xi_t^* \quad = \quad 0 \qquad\qquad \forall\ t = 1,..,n \tag{5.16}$$

$$((< w, x_t > +b) - y_t - \varepsilon - \xi_t\ )\,\beta_t \quad = \quad 0 \qquad\qquad \forall\ t = 1,..,n \tag{5.17}$$

$$((< w, x_t > +b) - y_t + \varepsilon + \xi_t^*\ )\,\beta_t^* \quad = \quad 0 \qquad\qquad \forall\ t = 1,..,n \tag{5.18}$$

Using Equations (5.11) and (5.15), we show that $\xi_t > 0$ only when $\beta_t = C/n$. Thus, for $\beta_t \in\ ]0, C/n[$, we have $\xi_t = 0$. Hence, b can be computed by using the fact that

$$b = y_t - < w, x_t > +\varepsilon \qquad \text{for } \beta_t \in\ ]0, C/n[\tag{5.19}$$

$$b = y_t - < w, x_t > -\varepsilon \qquad \text{for } \beta_t^* \in\ ]0, C/n[\tag{5.20}$$

This section provides a discussion of linear SVR formulation. Generalization to non-linear functions is performed by using the kernel mapping such that the dot product $< x_i, x >$ is replaced by the kernel function $k(x_i, x)$.

5.2 Residuals control charts

Residuals control charts are based on charting residuals that should be independently distributed over $N(0, \sigma_r)$, if process parameters are well estimated. This section presents control charts that only use the current observation or sample to monitor the process. Moreover, we first discuss univariate control charts that are used when only one process variable is monitored. These charts are Shewhart, Cumulative Sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) control charts. Then, we discuss the case of monitoring a battery of variables. In this case we use multivariate control charts named Hotelling's, Multivariate Cumulative Sum (MCUSUM) and Multivariate Exponentially Weighted Moving Average (MEWMA).

5.2.1 Residuals univariate control charts

X-Shewhart Control chart

A simple univariate statistic used to monitor the stability of the process against large shifts is the well known Shewhart X-Chart. Statistically speaking, a large magnitude is defined in terms of standard deviation. Since residuals are assumed to have a zero mean and a σ_r standard deviation, an observation is considered in-control if the target residual value r_i lies between the Upper Control Limit (UCL) and the Lower Control Limit (LCL) as follows

$$\begin{aligned} UCL &= \lambda\sigma_r. & (5.21) \\ LCL &= -\lambda\sigma_r. & (5.22) \end{aligned}$$

where σ_r is the estimated residuals standard deviation and λ determines a given in-control Run Length (RL) property.

CUSUM Control chart

CUSUM chart was developed to overcome the problem of Shewhart which is relatively insensitive to moderate shifts, since they use only information from the most recent observation. Let $S_0^+ = S_0^- = 0$ and $k_{cusum} > 0$ be pre-specified constants. The CUSUM statistic is calculated iteratively using the following equations

$$S_i^+ = max\{0, S_{i-1}^+ + r_i - k_{cusum}\}, \tag{5.23}$$
$$S_i^- = max\{0, S_{i-1}^- - r_i - k_{cusum}\}. \tag{5.24}$$

The CUSUM scheme signals when the statistic $S^- > H_{cusum}$ or $S^+ > H_{cusum}$, where k_{cusum} and H_{cusum} need to be determined previously by a given in-control desired RL property.

EWMA Control chart

While CUSUM charts take into account all previous measurements, EWMA charts weight the latest observation based on its importance in characterizing the process. The larger the value of λ is, the greater is the influence of the latest observation and vice versa. The EWMA statistic is calculated as follows

$$Z_i = (1 - \lambda)Z_{i-1} + \lambda r_i. \tag{5.25}$$

EWMA scheme signals if the statistic $Z > UCL$ or $Z < LCL$, where limits are calculated as follows

$$UCL = L\sigma_r\sqrt{\frac{\lambda}{2-\lambda}}, \tag{5.26}$$
$$LCL = -L\sigma_r\sqrt{\frac{\lambda}{2-\lambda}}. \tag{5.27}$$

where L is pre-determined for a given in-control RL property.

5.2.2 Residuals multivariate control charts

In order to monitor more than one quality autocorrelated characteristic, it is necessary to control residuals variation as shown in the next sections.

T^2 Chart

The multivariate extension of Shewhart-type chart is the Hotelling's T^2 chart. This chart reduces residuals to a scalar using the Mahalanobis distance. An observation is assumed to be in-control if the following condition is satisfied

$$T_i^2 = R_i^T \Sigma_R^{-1} R_i \sim \frac{m(n-1)}{n-m} F_{m,n-m}(\alpha). \tag{5.28}$$

where m is the number of variables, R_i is the residuals vector and $F_{m,n-p}(\alpha)$ is a Fisher distribution. The term α is the risk level and it ensures a desired in-control ARL or in-control rate of false alarms. Σ_R^{-1} is the inverse of the estimated residuals covariance matrix.

MCUSUM Chart

MCUSUM chart was developed to overcome the problem of Hotelling control charts which are relatively insensitive to moderate shifts. To detect small process shifts, MCUSUM charts accumulate deviations of the residuals of previous observations from a given target. The most widely used MCUSUM control procedure is Croisier's chart. For $S_0 = \mathbf{0}$, where $\mathbf{0}$ is a $1 \times m$ matrix of zeros, the statistic is as follows,

$$S_i = \begin{cases} \mathbf{0} & \text{if } C_i \leq k \\ (S_{i-1} + R_i)(1 - k_{cro}/C_i) & \text{otherwise} \end{cases} \tag{5.29}$$

where

$$C_i = (S_{i-1} + R_i)\Sigma_R^{-1}((S_{i-1} + R_i)^T. \tag{5.30}$$

Croisier's chart signals a shift when $T_S^2 = S_i^T \Sigma_R^{-1} S_i$ exceeds a pre-determined limit H_{cro}.

Residuals MCUSUM charts for nonlinear processes exhibit some differences as compared to linear systems. Indeed, any shift in a parameter of a nonlinear process can not only affect the mean of residuals but also the residuals distribution. Therefore, an alternative simple and efficient control procedure is the use of the cumulative sum of T^2. This chart is named the COT scheme and is the most direct extension of the multivariate T^2 chart. Let $S_0 \geq 0$ and $k_{\text{cot}} > 0$ be a pre-specified constants. The MCUSUM statistic is calculated iteratively using the following equation,

$$S_i = max\{0, S_{i-1} + T_i^2 - k_{\text{cot}}\} \text{ for } i = 1, 2, .., \tag{5.31}$$

The MCUSUM scheme signals when the S statistic exceeds a certain level H. That is, the chart signals a process change if $S_i > H_{\text{cot}}$, where k_{cot} and H_{cot} need to be determined previously for a given in-control desired RL characteristic.

MEWMA Chart

While MCUSUM charts take into account all previous measurements, MEWMA charts weight the last observation based on its importance in characterizing the process. The larger the value of λ is, the greater is the influence of the last observation and vice versa. The MEWMA statistic is defined iteratively as follows

$$Z_i = (1 - \lambda)Z_{i-1} + \lambda R_i \text{ for } i = 1, 2, .., \tag{5.32}$$

where λ is a diagonal matrix of values λ_j, $j = 1, ..., m$. The MEWMA scheme signals if the $T_Z^2 = Z_i^T \Sigma_Z^{-1} Z_i$ exceeds a predetermined value H.

In practice, usually the weight values λ_j are taken to be equal, such that $\lambda_1 = .. = \lambda_m$. In this case, we proposes to use a simple extension chart of T^2 for MEWMA, called EWMAT, based on the following statistic

$$Z_i = (1 - \lambda)Z_{i-1} + \lambda T_i^2 \text{ for } i = 1, 2, .., \tag{5.33}$$

EWMAT scheme signals if the Z statistic exceeds a value L, pre-determined for a given in-control RL property.

5.3 Design of Support Vector Control charts

In designing SVR based control charts, there are two phases. Phase I defines SVR parameters and control chart limits based on in-control set of observations. Because SVR has two parameters to tune besides the parameter of the kernel function, the use of grid search technique with a cross-validation method allows selecting the optimal parameters. These parameters should guarantee not only the smallest Mean Squared Error (MSE) but the residuals should also be independent. A way to examine the autocorrelation of the residuals is the use of Ljung-Box Q test given by

$$Q = n(n+2)\sum_{k=1}^{h} \frac{\hat{\rho}_k}{n-k}, \tag{5.34}$$

where n is the sample size, $\hat{\rho}_k$ is the sample autocorrelation at lag k, and h is the number of lags being tested. For a significance level α, the critical region for rejection of the hypothesis of residuals independency is

$$Q > \chi^2_{(1-\alpha,h)}, \tag{5.35}$$

where $\chi^2_{(1-\alpha,h)}$ is the α-quantile of the chi-square distribution with h degrees of freedom.

After parameters tuning, we perform estimation of parameters of the monitored process. To perform time series estimation for a multivariate process by SVR, the input variable Y_t is modelled by the previous variables of the series $(Y_{(t-1)}, .., Y_{(t-p)})$, where p is the lagged time. Suppose we have an AR process with order p represented by the following function that need to be estimated,

$$Y_t = f(Y_{(t-1)}, .., Y_{(t-p)}). \tag{5.36}$$

Estimation of the process using SVR provides $\hat{\text{f}}$ which allows predicting Y_t as follows,

$$\hat{Y}_t = \hat{f}(Y_{(t-1)}, .., Y_{(t-p)}). \tag{5.37}$$

If the function f is well estimated, then the error term $\hat{e}_t = (Y_t - \hat{Y}_t)$, which would be time independent and normally distributed with zero means, is used to construct the control chart. Indeed if a shift is present, then the process is no more represented by the function f and therefore the residual term $\hat{e}_t$ would be also affected and shifted. Specifying control limits is also one of the critical decisions that should be made in designing a control chart. Moving control limits further leads to a trade off between the risk of a type I error and that of type II error. That is, the first one is the risk of indicating an out-of-control condition when no assignable cause is present and the second is indicating normal system condition while the process is really out of control. To evaluate the control chart parameters, the most used statistic is the ARL which represents the average number of points that must be plotted before a point indicates an out-of-control condition.

5.4 Conclusion

This chapter presents an overview of estimating time series models using Support Vector Regression method. Moreover, several residuals univariate and multivariate control charts are discussed. To control nonlinear autocorrelated processes, a procedure that allow estimating the process structure in order to filter out autocorrelation and to use residuals for control is proposed. The evaluation of the effectiveness of the proposed procedure is conducted in the next chapter.

6 Assessment of the proposed chart using simulated processes

This chapter investigates application of the proposed residuals based SVR control charts with different simulated nonlinear processes. Section 1 discusses monitoring of univariate autocorrelated nonlinear processes. Then, in Section 2, the extension to multivariate processes is performed. Eventually, Section 3 resumes provides a conclusion.

6.1 Univariate process control analysis

In order to analyze the proposed procedures, a univariate nonlinear time series is used. The process follows an Exponential autoregressive model ExpAR of order 2 which represents a random oscillation process with nonlinear dynamics. The equation of the system is as follows

$$x_t = (\alpha_1 + \alpha_2 e^{-\theta x_{t-1}^2})x_{t-1} - (\alpha_3 + \alpha_4 e^{-\theta x_{t-1}^2})x_{t-2} + \varepsilon_t \tag{6.1}$$

where $\varepsilon_t \sim N(0, \delta)$.

In this study the used simulation parameters are as follows $\alpha_1 = 1.95$, $\alpha_2 = 0.23$,$\alpha_3 = 0.96$, $\alpha_4 = 0.24$, $\delta = 0.05$, $\theta = 1$ and initial values $x_1 = 0.2$ and $x_2 = 1$. Figure 6.1 illustrates the nonlinear dynamics of the process and Figure 6.2 presents its sample autocorrelation.

Example of such system can be found in Messaoud et al. (2008), which applied the ExpAR function for Deep-Hole Drilling Process in order to construct residuals EWMA control chart. Shi et al. (2001) used ExpAR model to

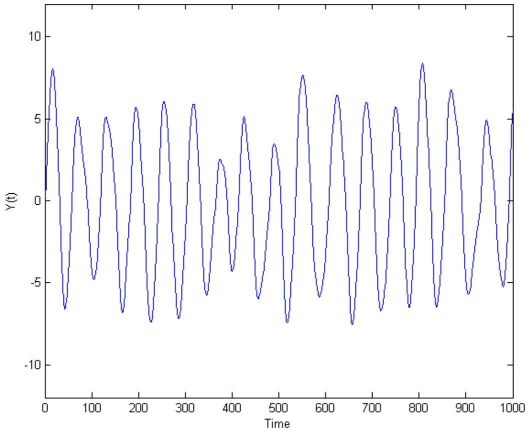

Figure 6.1: Exponential autoregressive nonlinear process

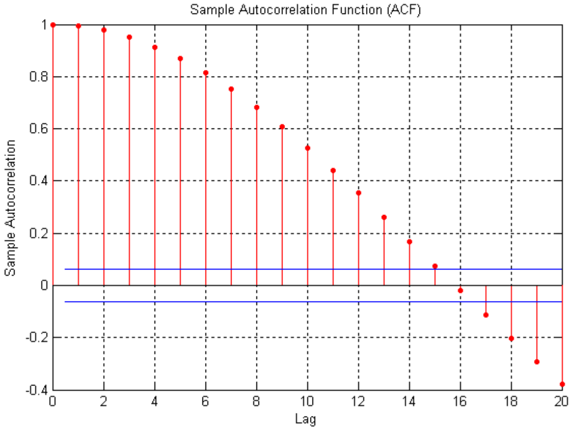

Figure 6.2: Process variables autocorrelation

monitor the dynamics evolution of Boiling water reactor oscillation to control nuclear systems. First of all, this section provides results obtained from the exact specification model by using an ExpAR (2) model. This fact allows comparing results of SVR model which does not require any pre-specified model structure. Estimation of this model using the maximum likelihood requires the use of nonlinear optimization procedure that can provide multiple local optima, since the derivative-based method does not guarantee convergence to a global optimum (Shi et al. (2001)). Therefore, we used the method of Haggan and Ozaki (1981) which is based on the approximation of a pre-specified interval for the value of the term θ. This optimal value is selected using a grid of candidate values. Then, the parameters are estimated by linear least squares method. Usually, the order p of the fitted model is selected based on the Akaike Information Criterion. However, we suppose that the exact order is known beforehand. In this study 10 validation samples with 2000 in-control observations each are used. For each iteration one sample is used for training and the other samples are used for testing. Using the described re-sampling method for training and validation, the exact estimated ExpAR model structure provides an average p-value of 0.34 for Ljung-Box Q-test and an MSE of 0.003.

To estimate the process structure through SVR, ν, σ^2 and C parameters need to be estimated. Using grid search method, Table 6.1, provides the obtained results taking into account several criteria. These criteria are the MSE criteria and the mean of the p-values of Ljung-Box Q-test. Analyzing results present in Table 6.1, one can notice overall, by minimizing the MSE criteria, the p-value is maximized. Also several combinations of parameters provide better results for some criteria and worse for others. Indeed, the optimal parameters based on the MSE criterion are $\nu = 0.5$, $C = 2^8$ and $\sigma^2 = 2^7$. Whilst using the p-value as a criterion, we obtain $\nu = 0.5$, $C = 2^8$ and $\sigma^2 = 2^5$. One can notice here that only the value of σ^2 is different. Therefore, we use the mean of both optimal values, i.e. $\sigma^2 = 2^6$. These optimal parameters provide an MSE equal to 0.006 and a mean p-value of Ljung-Box Q-test equal to 0.35. This fact shows that SVR method provides comparable results to the estimated ExpAR model without a need for any model specification structure. Also, we tested the normality of process residuals using Jarque-Bera test (Jarque and Bera (1987)). We obtained a p-value greater than 0.5, which means that residuals are normally distributed. Figure 6.3 and Figure 6.4 illustrate residuals independency.

Table 6.1: Grid search for parameters optimization

	C	$\mathbf{2^3}$	$\mathbf{2^4}$	$\mathbf{2^5}$	$\mathbf{2^6}$	$\mathbf{2^7}$	$\mathbf{2^8}$
v	σ^2	**MSE**					
0.3	$\mathbf{2^2}$	0.209	0.192	0.181	0.174	0.166	0.158
	$\mathbf{2^3}$	0.096	0.082	0.075	0.067	0.060	0.053
	$\mathbf{2^4}$	0.037	0.029	0.026	0.023	0.021	0.018
	$\mathbf{2^5}$	0.014	0.011	0.009	0.008	0.007	0.006
	$\mathbf{2^6}$	0.08	0.007	0.005	0.005	0.004	0.004
0.4	$\mathbf{2^2}$	0.207	0.190	0.178	0.171	0.159	0.154
	$\mathbf{2^3}$	0.095	0.080	0.073	0.064	0.058	0.052
	$\mathbf{2^4}$	0.036	0.028	0.025	0.022	0.019	0.018
	$\mathbf{2^5}$	0.014	0.011	0.009	0.008	0.007	0.006
	$\mathbf{2^6}$	0.008	0.006	0.005	0.005	0.004	0.004
0.5	$\mathbf{2^2}$	0.206	0.190	0.175	0.166	0.152	0.147
	$\mathbf{2^3}$	0.093	0.078	0.070	0.062	0.055	0.051
	$\mathbf{2^4}$	0.035	0.028	0.024	0.021	0.019	**0.016**
	$\mathbf{2^5}$	0.013	0.010	0.008	0.007	0.007	**0.006**
	$\mathbf{2^6}$	0.008	0.006	0.005	0.005	0.004	**0.003**
		Mean p-value					
0.3	$\mathbf{2^2}$	0.130	0.229	0.288	0.317	0.326	0.330
	$\mathbf{2^3}$	0.197	0.247	0.290	0.327	0.340	0.357
	$\mathbf{2^4}$	0.267	0.298	0.320	0.335	0.346	0.362
	$\mathbf{2^5}$	0.259	0.295	0.307	0.324	0.342	0.355
	$\mathbf{2^6}$	0.171	0.238	0.269	0.289	0.297	0.316
0.4	$\mathbf{2^2}$	0.157	0.253	0.295	0.321	0.327	0.330
	$\mathbf{2^3}$	0.215	0.265	0.306	0.326	0.340	0.350
	$\mathbf{2^4}$	0.283	0.306	0.325	0.336	0.351	0.362
	$\mathbf{2^5}$	0.274	0.300	0.312	0.323	0.331	0.339
	$\mathbf{2^6}$	0.196	0.255	0.272	0.295	0.309	0.330
0.5	$\mathbf{2^2}$	0.184	0.268	0.305	0.321	0.332	0.332
	$\mathbf{2^3}$	0.228	0.277	0.317	0.337	0.343	0.352
	$\mathbf{2^4}$	0.290	0.317	0.328	0.340	0.358	**0.373**
	$\mathbf{2^5}$	0.283	0.305	0.311	0.322	0.339	**0.357**
	$\mathbf{2^6}$	0.209	0.259	0.277	0.295	0.311	**0.337**

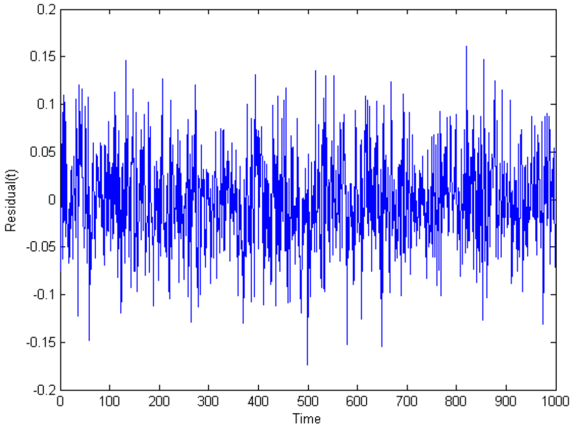

Figure 6.3: Process residuals using SVR model

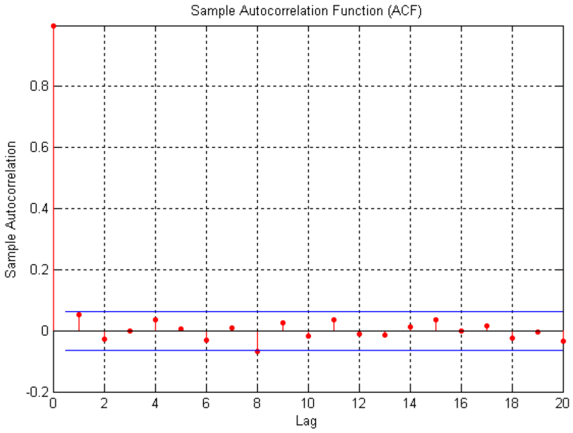

Figure 6.4: Process residuals autocorrelation

Using the estimated model, parameters of the control charts are defined in order to have an in-control ARL being approximately equal to 302. Selected chart parameters are as follows: for X-Shewhart, $\lambda = 2.97$, for CUSUM, $H_{cusum} = 0.75$ and $k_{cusum} = 2.8$ and for EWMA, $\lambda = 0.1$ and $L = 2.78$. To evaluate the performance of residuals control charts, 1000 samples are simulated for each shift. Shifts affect all process parameters. Using the defined SVR and control chart parameters, Table 6.2 provides performances of different control charts by providing detection delays measured by ARL.

Table 6.2: Fault detection results of the simulated univariate nonlinear process

	Parameters shift	**X**	**CUSUM**	**EWMA**
		ARL	ARL	ARL
In-control		302	302	302
Fault 1	$\triangle\delta = 0.01$	89	101	142
Fault 2	$\triangle\delta = 0.02$	36	49	81
Fault 3	$\triangle\delta = 0.05$	9	17	24
Fault 4	$\triangle\delta = 0.15$	2.3	6.2	6
Fault 5	$\triangle\alpha_1 = 0.03$	17	303	13
Fault 6	$\triangle\alpha_1 = 0.05$	9.8	28	7.3
Fault 7	$\triangle\alpha_1 = 0.15$	1.9	1.7	2.4
Fault 8	$\triangle\alpha_2 = 0.1$	210	138	172
Fault 9	$\triangle\alpha_2 = 0.25$	57	28	37
Fault 10	$\triangle\alpha_2 = 0.5$	6.5	14	27
Fault 11	$\triangle\alpha_3 = -0.03$	19	286	14
Fault 12	$\triangle\alpha_3 = -0.05$	11	33	8
Fault 13	$\triangle\alpha_3 = -0.15$	2.1	2	2.8
Fault 14	$\triangle\alpha_4 = -0.1$	255	102	91
Fault 15	$\triangle\alpha_4 = -0.2$	75	31	35
Fault 16	$\triangle\alpha_4 = -0.3$	39	22	29

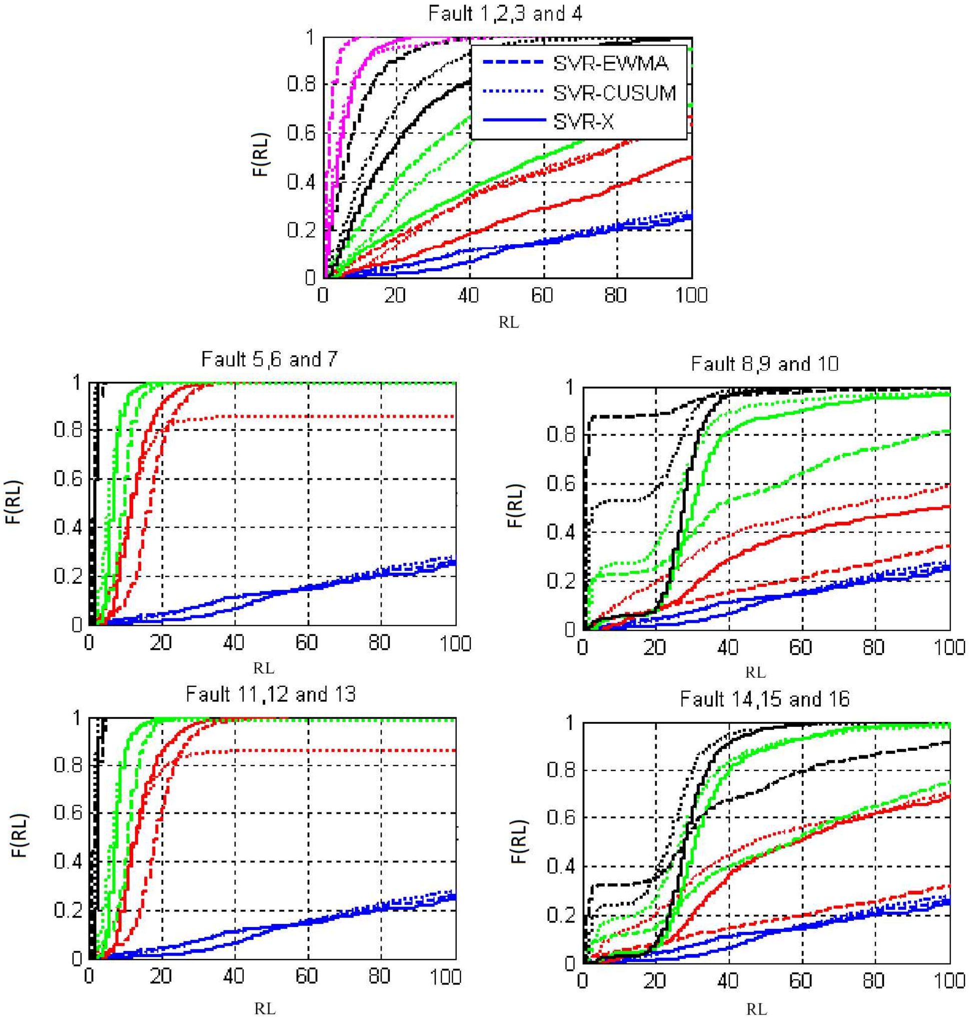

Figure 6.5: Run Length cumulative distributions of SVR-X, SVR-CUSUM and SVR-EWMA charts for the different applied shifts. Blue line represents the in-control situation. Red, green, black and violet lines represent the shift magnitude

First of all it is obvious that overall the applied control charts are able to capture all drift disturbances. For example a shift of the standard deviation δ equal to 0.02 decreased the ARL from 302 in in-control situation to $36, 49$ and 81. Whereas a shift of 0.15 implied an ARL of 2.3, 6.2 and 6 for respectively SVR-X, SVR-CUSUM and SVR-EWMA. On the other hand, the monitoring results in Table 6.2 show that SVR-X control chart provides better results for parameter shifts in δ, α_1, and α_3. However, for shifts in α_2 and α_4, SVR-CUSUM and SVR-EWMA provide overall better detection abilities. We notice that SVR-EWMA is always better for small and medium sized faults than the other charts except for δ. Moreover, SVR-CUSUM is unable to detect small shifts in α_1 and α_3. This is explained by the fact that this chart stipulates that shifts are linear and thus some nonlinear shifts can not be detected. These conclusions are illustrated in Figure 6.5 which presents RL cumulative distributions of the different control charts when the process is in-control(Blue line) and cumulative distributions when shifts are introduced.

6.2 Multivariate process control analysis

In order to study performances of different multivariate control charts an analysis of a multivariate nonlinear autocorrelated system is performed. The used nonlinear benchmark multivariate process is proposed by Chen and Liao (2002) and used by Choi and Lee (2004) and Yoo and Lee (2006). The process dynamics are as follows,

$$\begin{cases} Y(t) = Z(t) + v(t) \\ Z(t) = A \times Z(t-1) + B \times u^2(t-1) \\ u(t) = C \times u(t-1) + D \times w(t), \end{cases} \tag{6.2}$$

where $w(t) \sim N(0, \sigma_1)$ and $v(t) \sim N(0, \sigma_2)$. Parameters used in this study are as follows: $\sigma_1 = 0.5$, $\sigma_2 = 0.1$,

$$A = \begin{bmatrix} 0.8 & -0.2 & -0.1 \\ 0.1 & 0.9 & 0.2 \\ 0.1 & -0.1 & 0.9 \end{bmatrix}, \; B = \begin{bmatrix} 0.05 & 0.01 \\ 0.05 & 0.05 \\ 0.01 & 0.05 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.8 & -0.2 \\ 0.5 & 0.4 \end{bmatrix} \text{ and } D = \begin{bmatrix} 0.2 & 0.7 \\ -0.3 & -0.75 \end{bmatrix}.$$

Figure 6.6 illustrates process nonlinear multivariate dynamics and variables autocorrelation. To perform time series estimation for a multivariate process by SVR, each input variable $Y_{(i=1..m)t}$ is defined by previous variables of the series $(Y_{1(t-1)}, .., Y_{1(t-p)}, .., Y_{m(t-1)}, .., Y_{m(t-p)})$, where m is the number of variables and p the lagged time. This study used 10 validation samples that contains 1000 in-control observations each. For each iteration one sample is used for training and the other samples are used for testing. Applying an AR model of order $p = 3$, the selected optimal parameters of SVR are $\sigma = 2$, $C = 2$, $\nu = 0.1$, for variables Y_1 and Y_3 and $\nu = 0.2$, for variable Y_2. These parameters provide an MSE equal to 0.014, 0.018 and 0.014 and a p-value of the Ljung-Box Q test equal to 0.13, 0.22 and 0.12 for respectively variables Y_1, Y_2 and Y_3. Moreover, we tested the normality of process residuals using Henze-Zirkler test (Henze and Zirkler (1990)). We obtained a p-value equal to 0.48, which means that residuals have a multivariate normal distribution. Figure 6.7 presents process residuals and their autocorrelation for the different variables.

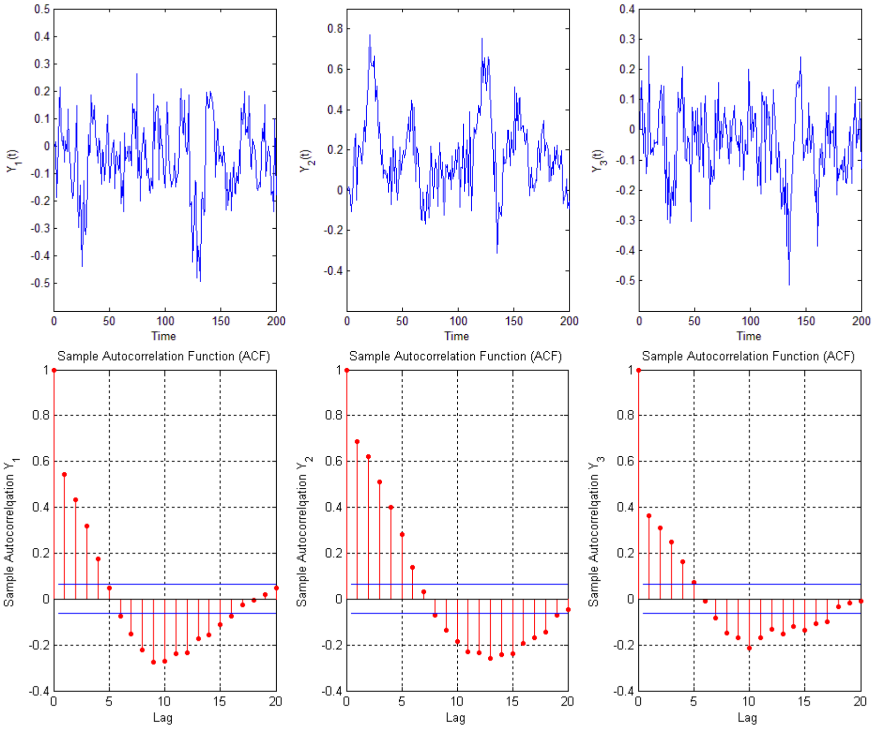

Figure 6.6: Illustration of the multivariate nonlinear process and variables autocorrelation

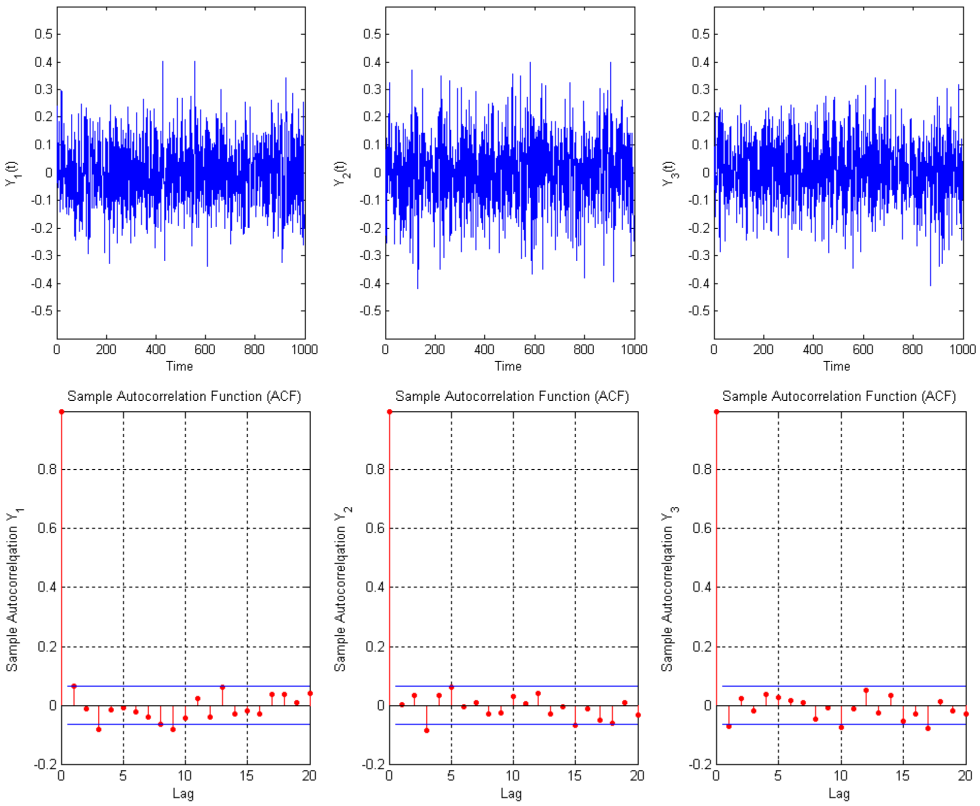

Figure 6.7: Illustration of multivariate process residuals and residuals autocorrelation

Control chart limits are defined such that the in-control ARL is approximately equal to 202. Applying 1000 samples, the defined SVR control charts parameters have the following values: $\alpha = 0.5$ %, $H_{\text{cot}} = 12.2$, $k_{cot} = 4.5$, $\lambda = 0.1$ and $L = 4.36$. To evaluate the performance of multivariate residuals control charts, 1000 samples are simulated for each shift. As made for univariate SVR residuals control chart analysis, different shifts in the process parameters are introduced. Analysis of the detection capability of the charts is presented in Table 6.3 which provides the detection delays measured by the ARL of SVR based T^2, COT and EWMAT charts for several shifts. Figure 6.8 compares the cumulative distribution of RL for in-control process with shifted processes.

Table 6.3: Fault detection results of the simulated multivariate nonlinear process

Shift	**SVR-T^2**	**SVR-COT**	**SVR-EWMAT**
	ARL	ARL	ARL
$A(1,1) = 0.9$	143	130	122
$A(1,1) = 1$	49	41	43
$A(1,1) = 1.2$	16.5	15.4	16.5
$A(2,2) = 0.95$	122	114	109
$A(2,2) = 1.05$	30	28	28.7
$A(2,2) = 1.2$	9.8	9.8	10.8
$A(3,1) = 0.15$	174	165	159
$A(3,1) = 0.2$	130	119	114
$A(3,1) = 0.5$	35	32.5	33
$B(1,2) = -0.05$	161	154	153
$B(1,2) = -0.1$	86	83	81
$B(1,2) = 0.5$	6.4	6.1	8
$B(3,1) = 0.02$	163	167	162
$B(3,1) = 0.05$	63	50.5	57
$B(3,1) = 0.1$	21.3	17.5	20.5
$D(1,1) = 0.5$	111	102	104
$D(1,1) = 0.7$	67	63	63
$D(1,1) = 1$	33	29.5	33
$C(2,1) = 0.7$	129	122	123
$C(2,1) = 1$	49	42	46
$C(2,1) = 1.2$	26	23	27
$\sigma_1 = 0.6$	75	70	70
$\sigma_1 = 0.8$	22	19.5	22
$\sigma_1 = 1.5$	3.3	3.3	4.3
$\sigma_2 = 0.11$	97	90	81
$\sigma_2 = 0.13$	30	23.6	26
$\sigma_2 = 0.2$	3.9	3.7	6.5

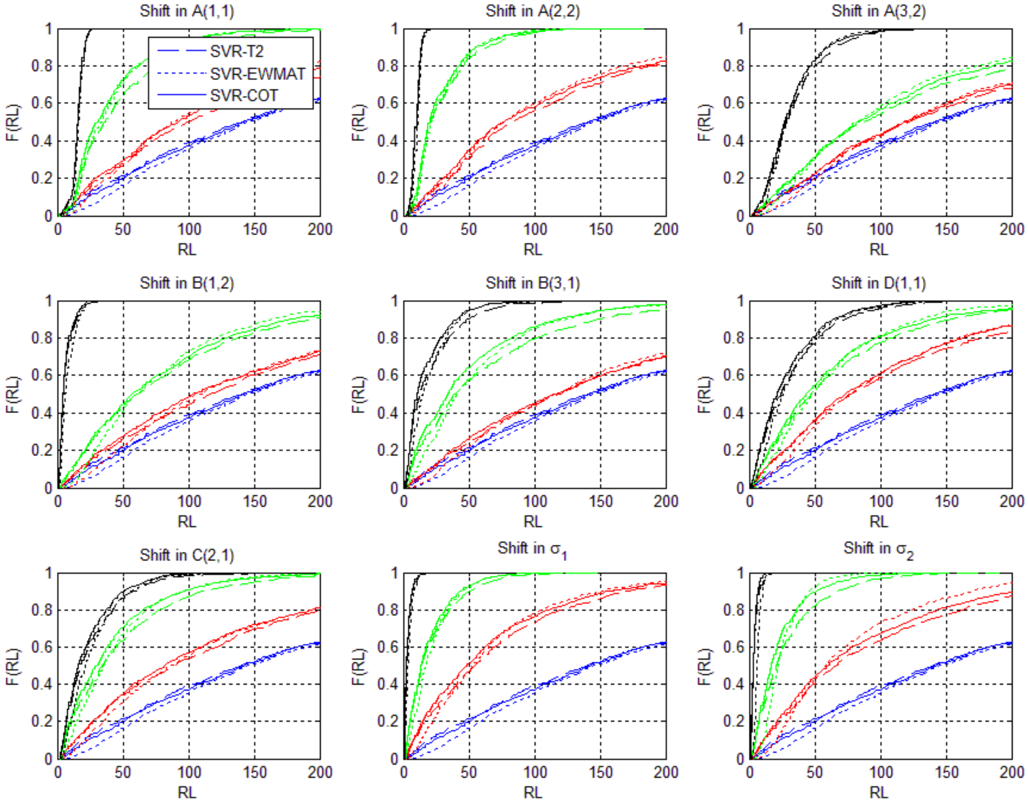

Figure 6.8: Run Length cumulative distributions of SVR-T^2, SVR-COT and SVR- EWMAT charts for different applied shifts.Blue line represents the in-control situation. Red, green and black lines represent the shift magnitude

First of all, it is obvious that overall SVR control charts are able to capture all drift disturbances. For example a small shift in $A(1,1)$ of 0.1 decreased the ARL from 202 in in-control situation to 143, 130.5 and 122, whereas a shift of 0.4 implied an ARL of 16.5, 15.4 and 16.5 for respectively SVR-T^2, SVR-COT and SVR-EWMAT. For $B(3,1)$, a shift of 0.05 decrease the ARL to 174, 165 and 159, whereas a shift of 0.4 implied an ARL equal to 35, 32.5 and 33 for respectively SVR based T^2, COT and EWMAT. Also an increase of the standard deviation σ_1, by 0.1 decreased the ARL to 75, 70 and 70 for respectively SVR-T^2, SVR-COT and SVR-EWMAT. These results are summarized in Figure 6.8 which present the cumulative distribution of run length and which makes clear the good sensitivity of the charts to the different applied shifts.

On the other hand, the monitoring results of Table 6.3 show that SVR-COT and SVR-EWMAT control charts tend to provide better results for smaller parameter shifts. Meanwhile, the detection results can not confirm which control chart between SVR-COT and SVR-EWMAT has better detection ability for small shifts. Indeed as shown in Table 6.3, each chart provides better performance for a certain specific shift. However, for larger shifts in the same parameters, different control charts seem to provide overall equivalent detection abilities.

6.3 Conclusion

This chapter investigates monitoring of multivariate nonlinear autocorrelated processes by means of several SVR based control charts. The proposed schemes allow handling of complex systems without requiring a predefined process structure and can be of great interest in manufacturing of high technology where nonlinear continuous processes are frequently assessed, such as in chemical industry. Results show that the used control charts can effectively monitor the process behaviour guaranteeing an acceptable robustness against in-control false alarms. Indeed, introduction of shifts to different parameters is efficiently detected by the different charts with a slightly better sensitivity performance for some charts over the others. An interesting future research concerns the use of adaptive SVR method for online estimation of the parameters when the process exhibits a non-stationary behaviour.

Part III

Multimodal process monitoring using Local Support Vector Domain Description

As technologies become more complex, modelling and detection of potential faults become more and more difficult. Indeed, usually modern processes do not satisfy classical methods assumptions, such as normality or linearity. To overcome these difficulties many non-parametric and non-linear procedures have been recently proposed. One of these methods is the Support Vector Domain Description (SVDD), proposed by Tax and Duin (1999). This method belongs to the unsupervised learning category and it detects novelties and outliers whose application involves machine faults diagnosis. Sun and Tsung (2003) demonstrated that SVDD based control procedure can perform better than conventional charts when the underlying process distribution is not multivariate normal. Shin et al. (2005) applied this method for detection of faults in an electromechanical machinery and concluded that SVDD method and kernel-based learning algorithms are highly competitive on a variety of problems and are efficient for machine fault detection. Bu et al. (2009) used SVDD method to detect gradual change fabric defects and proved that a low missing rate can be achieved simultaneously under a low false alarm rate.

However, despite the success of applying SVDD method to fault detection of several processes, there was not enough research concerning applications of such methods to processes that run under multiple operating modes, because of product changes, set-point changes and manufacturing strategies (Zhiqiang and Zhihuan (2008), Hwang and Han (1999)). Unfortunately, direct application of current methods to such processes tends to produce unsatisfactory performance due to the adopted assumption of only one nominal operating region for the underlying process.

To address the above problem, this part of the dissertation proposes a process monitoring scheme based on separate local models. On the one hand, this fact would simplify and reduce the complexity of the problem which can help selecting SVDD parameters without a need for optimization. On the other hand this permits not only detecting faults but also distinguishing in which mode the underlying faults are happening. But because different modes are usually not well-defined, this study applies a clustering method in order to separate different process modes. Finding these local regions can be successfully solved through the k-means algorithm which is well known in pattern classification. The aim of k-means clustering is to group observations on the basis of similarities using the sum-of-squares criterion. However, if the separation boundaries between clusters are nonlinear then basic k-means may fail (Girolami (2002)). In order to overcome this problem, we use Kernel k-means to cluster datasets. This fact is performed by mapping the data into another space where k-means can be applied efficiently. Then, for each cluster group an SVDD model will be trained.

This part of the dissertation is outlined as follows: Chapter 1 introduces the principal of SVDD method, presents the problem of parameter estimation and provides an overview of kernel k-means clustering method. Then, a control chart based on local SVDD models for monitoring of multimodal systems is proposed. Chapter 2 studies the performance of the proposed chart based on simulated and benchmark processes.

7

Multimodal Support Vector Domain Description control chart

This Chapter is organized as follows. In Section 1, an overview of the SVDD method is presented. Section 2 discusses the problem of parameters selection for SVDD method. Section 3 exposes Kernel k-means clustering method. Then, in Section 4, the proposed control chart is discussed.

7.1 Support Vector Domain Description

SVDD is a method presented by Tax and Duin (1999) that allows detecting novel or abnormal observations by modelling the support of the target class from the outlier class. Therefore, SVDD defines a function that is positive where most observations lie and negative elsewhere. This goal is achieved by finding a spherical boundary with a minimal radius R and center a that contains most of the data. To take into account outliers that have a distance larger than the sphere radius, slack variables ε_i are introduced. This principal is illustrated in Figure 7.1. Thus, the task is to minimize the volume R^2 and slack variables ε_i as follows,

$$\left\{\begin{array}{rl} min_{(R,a)}\ R^2 + \frac{1}{nv}\sum\limits_{i=1}^{n}\varepsilon_i & \\ (x_i - a)(x_i - a)^T \leq R^2 + \varepsilon_i & \forall i = 1,..,n \\ \varepsilon_i \geq 0 & \forall i = 1,..,n \end{array}\right. \tag{7.1}$$

where $v \in [0,1]$ allows a trade-off between minimizing R^2 and $\sum\limits_{i=1}^{n}\varepsilon_i$.

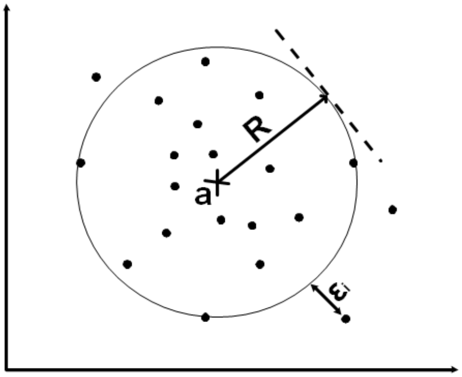

Figure 7.1: Principle of Support Vector Domain Description

The solution of the problem (7.1) can be solved through introduction of Lagrange multipliers that are constructed as follows:

$$L(R, a, \varepsilon_i, \alpha, \beta) = R^2 + \frac{1}{nv}\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n}\alpha_i\left(-(x_i-a)(x_i-a)^T + R^2 + \varepsilon_i\ \right) - \sum_{i=1}^{n}\beta_i\varepsilon_i. \tag{7.2}$$

where α_i and β_i are Lagrange multipliers.

Then, the following conditions at the solution point can be derived,

$$\frac{\partial}{\partial R}L = 0 \Longrightarrow \sum_{i=1}^{n}\alpha_i = 1. \tag{7.3}$$

$$\frac{\partial}{\partial \varepsilon_i}L = 0 \ \Longrightarrow \frac{1}{nv} - \alpha_i - \beta_i = 0. \tag{7.4}$$

Because of Equation (7.3),

$$\frac{\partial}{\partial a}L \ = \ 0 \Longrightarrow a = \sum_{i=1}^{n}\alpha_i x_i. \tag{7.5}$$

Since $\alpha_i \geq 0$ and $\beta_i \geq 0$, we can remove variables β_i from Equation (7.4) and use the constraint $0 \leq \alpha_i \leq \frac{1}{nv}$.

Developing Equation (7.2) and integrating Equations (7.3), (7.4) and (7.5) into the Lagrange function, we obtain the next optimization problem to solve numerically

$$\begin{cases} min_\alpha \ \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j < x_i, x_j > - \sum_{i=1}^{n} \alpha_i < x_i, x_i > \\ \\ 0 \ \leq \alpha_i \leq \frac{1}{nv} \qquad\qquad \forall i = 1, .., n \\ \sum_{i=1}^{n} \alpha_i = 1 \end{cases} \tag{7.6}$$

Only few observations have an $\alpha_i \neq 0$ and they are called Support Vectors (SV). According to KKT, only the Lagrange multipliers $\alpha_i' s$ and $\beta_i' s$ that are non zero at the saddle point correspond to constraints which are exactly saturated. These conditions state that at the point of the solution we have

$$\beta_i \varepsilon_i = 0 \qquad\qquad \forall \ i = 1, .., n \tag{7.7}$$

$$\alpha_i \left((x_i - a)(x_i - a)^T - R^2 - \varepsilon_i \ \right) = 0 \qquad \forall \ i = 1, .., n \tag{7.8}$$

Using Equation (7.7), $\varepsilon_i = 0$ when $\beta_i > 0$. Thus, using Equation (7.4), we get $\alpha_i \in \ ]0, \frac{1}{nv}[$. These SV are called Boundary SV. From equation (7.8), we notice that these points are located on the sphere boundary. The other SV are then called Non-boundary SV. This is illustrated in Figure 7.2.

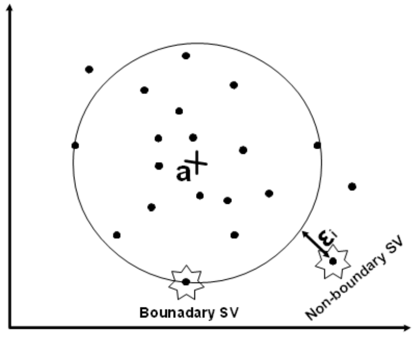

Figure 7.2: Support Vectors illustration for SVDD

The euclidean distance of any observation z to the center a is calculated as follows

$$\begin{aligned} Distance\ (z,a) &= \sqrt{(z-a)(z-a)^T}, & (7.9)\\ &= \sqrt{<z,z> -2<z,a> + <a,a>}. & (7.10)\end{aligned}$$

Using Equation (7.5), we obtain

$$Distance\ (z,a) = \sqrt{<z,z> -2\sum_{i=1}^{n}\alpha_i <z,x_i> + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j <x_i,x_j>}. \tag{7.11}$$

Since the distance of a boundary SV x_{sv} to the center a is equal to the radius R, we can calculate R as follows

$$R = \sqrt{<x_{sv},x_{sv}> -2\sum_{i=1}^{n}\alpha_i <x_{sv},x_i> + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j <x_i,x_j>}. \tag{7.12}$$

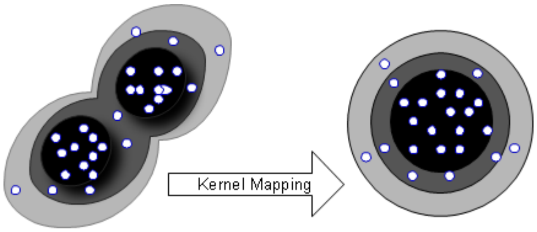

Figure 7.3: Kernel mapping into another space where data can be spherically distributed

As illustrated in Figure 7.3, in certain cases applying a spherical boundary in the original space does not provide a good characterization of the target class. Therefore, to overcome this problem, one can use the kernel trick by mapping the dataset into another space where they can be spherically distributed. Then, by using the kernel function the quadratic optimization problem (7.6) becomes as follows

$$\begin{cases} min_\alpha \ \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j k(x_i,x_j) - \sum_{i=1}^{n}\alpha_i k(x_i,x_i) \\ \\ \quad 0 \ \le \alpha_i \le \frac{1}{nv} \qquad\qquad \forall i=1,..,n \\ \quad \sum_{i=1}^{n}\alpha_i = 1 \end{cases} \tag{7.13}$$

Using a boundary support vector x_{sv}, R is calculated as follows,

$$R=\sqrt{k(x_{sv},x_{sv}) - 2\sum_{i=1}^{n}\alpha_i k(x_i,x_{sv}) + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j).} \tag{7.14}$$

In order to test if a new observation x_{new} is in-control, R is used as a limit. An observation x_{new} is considered to be in-control if the distance

$$\sqrt{k(x_{new},x_{new}) - 2\sum_{i=1}^{n}\alpha_i k(x_i,x_{new}) + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j).} \tag{7.15}$$

is below R.

7.2 Optimal SVDD parameters estimation

7.2.1 The Problem of SVDD parameters estimation

To apply SVDD method, several parameters need to be tuned. These parameters are the type of the kernel function and its parameter value. However, as mentioned previously, this study uses the Gaussian kernel function. Added to that, one needs to optimize the parameter v of the problem defined in

Equation (7.1). Schölkopf et al. (2001) proved that v is an upper bound on the fraction of outliers and also a lower bound on the fraction of SV of the training dataset. Tax and Duin (2004) show that the expectation of the fraction of SV is the upper bound of false alarm rate. Then, knowledge about the false alarm rate can help specifying the corresponding value of v.

Concerning the parameter σ of Gaussian kernel, an appropriate value is important for designing a detector with a good performance. When σ is chosen to be too small then an underfitting problem can occur. This means a detector with a simple decision boundary that almost looses its discriminating capability and hence poor results can be obtained when testing is performed. In contrast, large values of σ may imply an overfitting problem, which gives an excessively complicated decision boundary with poor generalization during testing. Tax et Duin (1999) proposes to use a value of σ that provides a priori approximately the maximal allowed rate of outliers, i.e. the error on the target set. Applying leave-one-out estimation on the training set, Tax and Duin (1999) estimate the error on the target set by the fraction of SV as follows

$$E(P(error)) = \frac{N_{SV}}{N}, \tag{7.16}$$

where N$_{SV}$ is the number of SV and N is the number of observations.

Tax and Duin (1999) stated that Equation (7.16) is a good estimate of the error on the target class and used the fraction of SV to determine the value of σ. Using the same idea, Schölkopf et al. (2001) proposes a simple heuristic that starts with a large σ value and decreases it until the number of SV does not decrease any further. Banerjee et al. (2006) used cross-validation method to search for the value of σ that minimizes the expectation of the fraction of SV and hence the upper bound of false alarm rate. However, using this principal a problem arises when the used dataset does not contain any outlier and therefore the rate of SV can not be set to zero. Moreover, Bu et al. (2009) showed that, for another experiment, the average fraction of SV does not guarantee the expected alarm rate. Therefore, using cross-validation, Bu et al. (2009) stipulate that an optimal value should minimize the difference between the average of fractions of Support Vectors obtained from training datasets and the average of alarm rates from testing datasets. Next, we summarize the selection criteria.

$Best_{Tax \text{ and Duin (1999)}}$: Choose v and σ such that

$$\frac{N_{SV}}{N} \simeq \text{A priori outliers rate} \tag{7.17}$$

$Best_{Bu \text{ et al. (2009)}}$: Choose v and σ such that

$$v \simeq \text{A priori outliers rate} \tag{7.18}$$

$$\sigma \leftarrow \min_{\sigma} \left(Mean \left(\frac{N_{SV}}{N} \right) - Mean \left(\text{Error rate} \right) \right) \tag{7.19}$$

This short literature overview shows that researchers provided different methods for estimation of SVDD parameters and there is no consensus about the best method. Therefore, next section is dedicated to the analyses of optimal selection methods.

7.2.2 Analysis of optimal SVDD parameters estimation

In order to analyze the effectiveness of SVDD optimal parameters selection methods, we use several simulated nonlinear processes as well as some benchmark datasets. The simulated datasets are a cross, a ring and a simple multimodal form and they are illustrated in Figure 7.4.

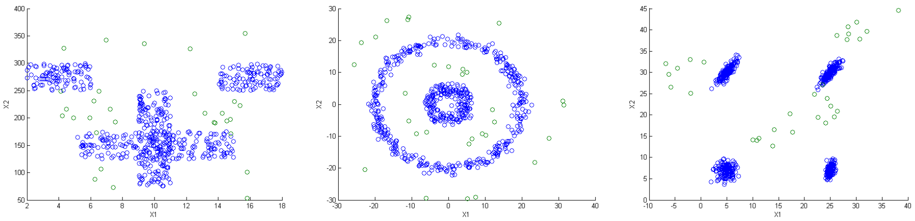

Figure 7.4: Simulated datasets. The blue points are in-control observations and the green points are out-of control

Concerning the benchmark datasets, this study uses Iris, Balance, Ionosphere and Breast datasets. These datasets are frequently used in literature to assess classifications methods and are available in UCI Repository. In

Table 7.1: Benchmark datasets used to evaluate SVDD parameters selection methods

Dataset	Number of observations	Number of variables	Number of labels
Iris	150	4	3
Balance	625	4	3
Ionosphere	351	34	2
Breast	682	9	2

order to use them for one class method evaluation, each time one label is used as a outlier class and the other labels are used as target class. Table 7.1 describes the used datasets

Moreover, in order to test the influence of the outliers when they are introduced into the training datasets, this study affects the target class of the simulated datasets with 0 %, 2% and 5% of outliers. Table 7.2 presents performances of the optimal parameters based on Tax and Duin (1999) and Bu et al (2009) criteria. In this study, we suppose that we can allow 5 % of rejection from the target class plus the supposed rate of outliers introduced in the training dataset. That means that we select the smallest σ and v that allow an approximate 5 %, 7 % and 10 % of Support Vectors when respectively a rate of 0 %, 2% and 5% of outliers are introduced. Concerning Bu et al (2009) criterion, v is set to 0.05 when 0 % and 2 % of outliers are introduced in the simulated datasets and 0.1 when 5 % of outliers are introduced in the simulated datasets.

In Table 7.2, performances of SVDD parameters optimization methods for all simulated and benchmark datasets are presented. To estimate results, 5-fold cross validation is used. Results contain the false alarm rates and the detection rates. First, we notice that the bound provided by Tax and Duin (1999) does not guarantee the False alarm rate for the test sets. This problem is due to the fact that the used bound concerns only the training set. Concerning the detection results, Table 7.2 shows that both methods of Tax and Duin (1999) and Bu et al (2009) present overall approximately equivalent results with little better performance using the criterion of Tax and Duin (1999) for only some datasets. Concerning the ring and multimodal datasets, results show that as outliers rate in the training dataset increases, the detection performance increases which is not the case for the cross dataset. This

Table 7.2: Fault detection results of SVDD with different parameters selection methods

	Tax and Duin (1999)			Bu et al (2009)		
	$(v,\ \sigma)$	False alarm rate	Detection rate	$(v,\ \sigma)$	False alarm rate	Detection rate
Ring dataset						
0 % outliers	$(0.05,\ 2^{4.5})$	0.06	0.5	$(0.05,\ 2^{4.5})$	0.07	0.5
2 % outliers	$(0.05,\ 2^{3.5})$	0.06	0.69	$(0.05,\ 2^{5})$	0.04	0.5
5 % outliers	$(0.05,\ 2^{3})$	0.06	0.86	$(0.1,\ 2^{3})$	0.06	0.86
Cross						
0 % outliers	$(0.05,\ 2^{5})$	0.06	0.83	$(0.05,\ 2^{5})$	0.06	0.83
2 % outliers	$(0.05,\ 2^{4})$	0.05	0.81	$(0.05,\ 2^{5})$	0.06	0.76
5 % outliers	$(0.05,\ 2^{3.5})$	0.06	0.64	$(0.1,\ 2^{3.5})$	0.06	0.64
Multimodal						
0 % outliers	$(0.05,\ 2^{4})$	0.06	0.63	$(0.05,\ 2^{4.5})$	0.07	0.53
2 % outliers	$(0.05,\ 2^{2})$	0.06	0.99	$(0.05,\ 2^{5})$	0.05	0.5
5 % outliers	$(0.05,\ 2^{2})$	0.04	0.84	$(0.1,\ 2^{1.5})$	0.04	0.84
Iris						
Class 1	$(0.05,\ 2^{3})$	0	0	$(0.05,\ 2^{3})$	0	0
Class 2	$(0.05,\ 2^{2.5})$	0.2	0	$(0.05,\ 2^{3})$	0.15	0
Class 3	$(0.05,\ 2^{1.5})$	0.2	1	$(0.05,\ 2^{3})$	0	0
Balance						
Class 1	$(0.05,\ 2^{3})$	0.04	0.29	$(0.05,\ 2^{5})$	0.07	0.25
Class 2	$(0.05,\ 2^{5})$	0.07	0.12	$(0.05,\ 2^{4.5})$	0.06	0.12
Class 3	$(0.05,\ 2^{1.5})$	0.18	0.8	$(0.05,\ 2^{3.5})$	0.06	0.31
Ionosphere						
Class 1	$(0.05,\ 2^{5})$	0.2	0.01	$(0.05,\ 2^{5})$	0.2	0.01
Class 2	$(0.05,\ 2^{5})$	0.13	0.63	$(0.05,\ 2^{2.5})$	0.13	0.74
Breast						
Class 1	$(0.05,\ 2^{3})$	0.06	0.97	$(0.05,\ 2^{4.5})$	0.08	0.97
Class 2	$(0.05,\ 2^{5})$	0.06	0.97	$(0.05,\ 2^{5})$	0.06	0.97

result is explained by the fact that these datasets have different complexities. However, the proposed parameters selection criteria take into account only the rate of outliers contained in the training dataset. In fact, as the complexity increases the rate of SV should increase. Therefore, one should integrate also the problem complexity as a criterion in order to select the optimal parameters. Concerning real world datasets, both Tax and Duin (1999) and Bu et al. (2009) criteria provide good and equivalent detection performance for Breast dataset. However, Tax and Duin (1999) gives acceptable results only when the testing Class is 3, 3 and 2 for respectively Iris, Balance and Ionosphere datasets. For Bu et al (2009) criterion, results are only acceptable for Class 2 of Ionosphere dataset. The poor performance for other datasets can be explained by the distribution of the classes. In fact Tax and Duin (1999) show that when the outlier class lies between the other classes the detection performance becomes very poor. In order to overcome these problems namely the definition of the problem complexity and the problem when the outlier class lies between the training classes, we propose to use clustering method in order to construct several local SVDD models. Clustering the dataset would reduce the problem complexity and therefore parameters selection can be easier when using several local models. Also, this fact can overcome the problem of SVDD method when used in multimodal processes where the oultier class lies between the other classes. Another important advantage of using the proposed method is the interpretation of the out-of-control condition. In fact, when using a global SVDD model one can not decide in which mode the problem is happening which is not the case when using several clusters.

7.3 Kernel k-means clustering

Clustering is an unsupervised method that aims to partition a dataset into smaller subsets of similar characteristics. One of the most used clustering approaches is k-means. To cluster a dataset, classical k-means uses distances that assume that the shape of the individual cluster is hyperspherical (Euclidean) or hyperelliptical (Mahalanobis) (Jain and Dubes (1998)). Given an unlabeled dataset $X_{i=1,..,n}$, euclidean k-means estimates the centroides $m_{i=1:c}$, where c is a predefined number of clusters, based on the following optimization,

$$Min_{(m_i)} \sum_{i=1}^{c} \sum_{j \in \pi_i} ||x_j - m_i||, \tag{7.20}$$

$$m_i = \frac{\sum\limits_{j \in \pi_i} x_j}{|\pi_i|}, \tag{7.21}$$

where π_i is the cluster i and $|\pi_i|$ is the number of its observations.

Unfortunately due to the shape assumption, the use of classical k-means in many applications fails when used to non-hyperelliptical data. To overcome this limitation, among solutions that have been provided is the Kernel k-means method. This approach stipulates that by mapping the data into a higher space, we can find a higher dimensional space where the hyperelliptical shape can be used.

Kernel k-means apply a nonlinear mapping function from the initial feature space to a high dimensional feature space by using a nonlinear map Φ. Then, the estimation of centroides is based on the optimization of the following problem, where x_j of Equation (7.20) is replaced by $\Phi(x_j)$.

$$Min_{(m_i)} \sum_{i=1}^{c} \sum_{j \in \pi_i} ||\Phi(x_j) - \Phi(m_i)||, \tag{7.22}$$

$$\Phi(m_i) = \frac{\sum\limits_{j \in \pi_i} \Phi(x_j)}{|\pi_i|}, \tag{7.23}$$

$$\Phi(m_i) = \frac{\sum\limits_{k=1}^{n} \delta_{ik} \Phi(x_k)}{\sum\limits_{k=1}^{n} \delta_{ik}}, \tag{7.24}$$

$$\Phi(m_i) = \sum_{k=1}^{n} \nu_{ik} \Phi(x_k), \tag{7.25}$$

where

$$\nu_{ik} = \frac{\delta_{ik}\Phi(x_k)}{\sum\limits_{k=1}^{n} \delta_{ik}} \tag{7.26}$$

and

$$\delta_{ik} = \begin{cases} 1 & \text{if } x_k \in \pi_i \\ 0 & \text{otherwise} \end{cases} \tag{7.27}$$

However because Φ is not explicitly known, we could use the kernel trick to compute distances between patterns and codevectors $\Phi(m_i)$. Indeed, this can be done using the kernel trick as follows:

$$\begin{aligned} ||\Phi(x_i) - \Phi(x_j)||^2 &= < \Phi(x_i), \Phi(x_i) > + < \Phi(x_j), \Phi(x_j) > \\ &\quad -2 < \Phi(x_i), \Phi(x_j) >, \\ &= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j). \end{aligned} \tag{7.28}$$

By writing each centroid in feature space as a combination of data vectors in feature space, we have

$$||\Phi(x_j) - \Phi(m_i)||^2 = ||\Phi(x_j) - \sum_{k=1}^{n} \nu_{ik}\Phi(x_k)||^2, \tag{7.29}$$

$$= k_{jj} - 2\sum_{k=1}^{n} \nu_{ik}k_{jk} + \sum_{r=1}^{n}\sum_{s=1}^{n} \nu_{ir}\nu_{is}k_{rs}. \tag{7.30}$$

For a new observation x$_{new}$, prediction of the cluster membership is based on the minimal distance to centroid calculated in the higher dimensional space as follows

$$\text{Membership } (x_{new}) = min_i(||\Phi(x_{new}) - \Phi(m_i)||^2). \tag{7.31}$$

Then, kernel k-means algorithm has different steps present in Table 7.3.

Table 7.3: Kernel K-means algorithm

Step 1.	Set the number of clusters p and the kernel parameters.
Step 2.	Project the dataset X using kernel functions.
Step 3.	Random initialization of clusters memberships.
Step 4.	Compute distances to each cluster centroid m_i^{Φ}.
Step 5.	Update the membership of each observation using Equation 7.31.
Step 6.	Go to step 4 until no membership change is performed.

7.4 Kernel k-means based SVDD fault detection

To address the problem of monitoring systems with multiple operating modes that have several local different characteristics, this study implements a control procedure based on separate local SVDD models. The proposed scheme is applied by combining the Kernel k-means clustering algorithm and SVDD algorithm presented in the preceding section. To design the monitoring chart, first the dataset is clustered using Kernel k-means method. This fact would allow separating different groups of datasets that have common characteristics. Then, for each group of data a local SVDD method is trained to obtain the underlying data distribution model. At this stage new observations are monitored by first determining their group membership and then by computing their distance within the underlying group using the corresponding SVDD model.

Another important issue in designing the control chart is the determination of different parameters. The first parameter to estimate is the number of clusters. Indeed although in certain cases this number is known beforehand, usually the number of modes or clusters is unknown. Therefore a method to estimate the number of clusters need to be investigated. As indicated by Girolami (2002), an estimation of the number of clusters within the data can be given by the eigenvalue decomposition of the kernel matrix. A principal components decomposition gives the approximation of the kernel matrix by Equation (7.32),

$$K = \mathbf{V}\Lambda\mathbf{V}^T, \tag{7.32}$$

where $\mathbf{V}$ and Λ represent respectively matrices of eigenvectors v_i and eigenvalues λ_i. Therefore, Equation (7.32) can be rewritten as follows:

$$\mathbf{I}_n^T K \mathbf{I}_n = \mathbf{I}_n^T \left\{ \sum_{i=1}^{n} \lambda_i v_i v_i^T \right\} \mathbf{I}_n = \sum_{i=1}^{n} \lambda_i \left\{ \mathbf{I}_n^T v_i \right\}^2 , \tag{7.33}$$

where $\mathbf{I}_n$ is an n dimensional vector of elements $1/n$.

Equation (7.33) indicates that the biggest contribution to $\mathbf{I}_n^T K \mathbf{I}_n$ is made by the most significant terms $\lambda_i \left\{ \mathbf{I}_n^T v_i \right\}$. So if there are c separate clustered regions within the sample then the kernel matrix is approximated by the c dominant terms in the summation. Thus, the eigenvalue decomposition provides an accurate estimate of the number of clusters to use in the kernel space. Therefore, we use as an estimate of the number of clusters, the minimal number of principal components which guarantee α percent of the total explained variance. In this study, we use $\alpha = 95$ %.

The other parameters to tune are the kernel function and the parameter v of the SVDD method. The application of local SVDD models based on clustering of the data allows a simple and automatic parameter selection without a need for optimization. A method to determine the kernel parameter δ^2 is to consider it as a scale for the kernel product, by taking it to be equal to the sum of data variance. Suppose that the problem has m variables, then we have

$$\delta^2 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ji} - \bar{x}_{j.})^2}{n}. \tag{7.34}$$

Concerning the parameter v of SVDD method and as mentioned in Section 2, it represents an upper bound for the outlier rate. Therefore an appropriate value can vary between 0.01 and 0.05, and here we opt for 0.05 which allows, in the training sample, a maximal outlier rate of 5 %. As discussed in Section 7.2.2, the upper bound concerns only the training set and does not guarantee the same outlier rate for the test sets. Therefore, in this study we smooth the radius R of SVDD by using as limit $(1 + \lambda)R$, where λ is tuned to have a given in-control alarm rate using cross-validation method. In order to conclude this section, we provide the different steps to construct the proposed control chart in Table 7.4.

Table 7.4: Algorithmic steps of the proposed control procedure

```
Step 1: For an initial block of data do
  Step 1.1: Determine the kernel parameter $\delta^2$.
  Step 1.2: Construct the kernel matrix.
  Step 1.3: Estimate the number of clusters to apply.
  Step 1.4: Cluster data using Kernel k-means.
  Step 1.5: For each cluster do
    Step 1.5.1: Determine $\delta^2$ of the underlying cluster.
    Step 1.5.2: Train a separate SVDD model.
Step 2: For each new observation $x_{new}$ do
  Step 2.1: Construct the kernel column
  Step 2.2: Calculate its distance to each cluster.
  Step 2.3: Determine cluster membership i.
  Step 2.4: Using SVDD model of cluster i,
                test the state of $x_{new}$.
```

7.5 Conclusion

This chapter investigates control of complex and multimodal processes using Kernel k-means clustering to construct local SVDD models. This monitoring strategy monitors processes without requiring a predefined data distribution. Moreover, construction of local models allows simplifying the parameters optimization process and can provide better detection results. Also we have a better interpretation of out-of-control situations. This control chart can be used in many domains such as biology, industrial multimodal systems, audio-visual systems... etc, where problems are usually complex and where multimodality is frequently present. Analysis of this control strategy is conducted in the next chapter.

8 Performance evaluation of Kernel k-means local SVDD

This chapter is devoted to the analysis of local SVDD based Kernel k-means control chart. Section 1 assesses the proposed chart through simulation of two processes. Section 2 analyses a real case study of a Semiconductor process and Section 3 concludes this chapter.

8.1 A multivariate simulated multimodal process study

In order to assess performances of the proposed procedure, two simulated datasets are used. The first dataset is simulated by using 4 normally distributed clusters that have the following parameters

$$\begin{pmatrix} u_1 = & [5;7] \\ u_2 = & [8\ ;12] \\ u_3 = & [2;\ 3] \\ u_4 = & [17\ ;25] \end{pmatrix},$$

$$\begin{pmatrix} \Sigma_1 = \begin{pmatrix} 2 & 0.2 \\ 0.2 & 3 \end{pmatrix} & \Sigma_3 = \begin{pmatrix} 0.2 & 0.2 \\ 0.2 & 1 \end{pmatrix} \\ & \\ \Sigma_2 = \begin{pmatrix} 1 & 1.2 \\ 1.2 & 2 \end{pmatrix} & \Sigma_4 = \begin{pmatrix} 3 & 1.2 \\ 1.2 & 5 \end{pmatrix} \end{pmatrix}.$$

Faults are introduced after 300 in-control observations and they are simulated following a normal distribution with parameters

$$u_5 = [13; 17] \text{ and } \Sigma_5 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

For the second dataset, simulation is based on modelling a ring with two clusters that have as radius $r_1 \sim N(5, 1)$ and $r_2 \sim N(15, 1)$. Faults are introduced after observation 400 and they concern 10 out-of-control observations with $r \sim N(10, 1)$ and another 10 faults with $r \sim N(20, 1)$. Figure 8.1 illustrates both simulated data.

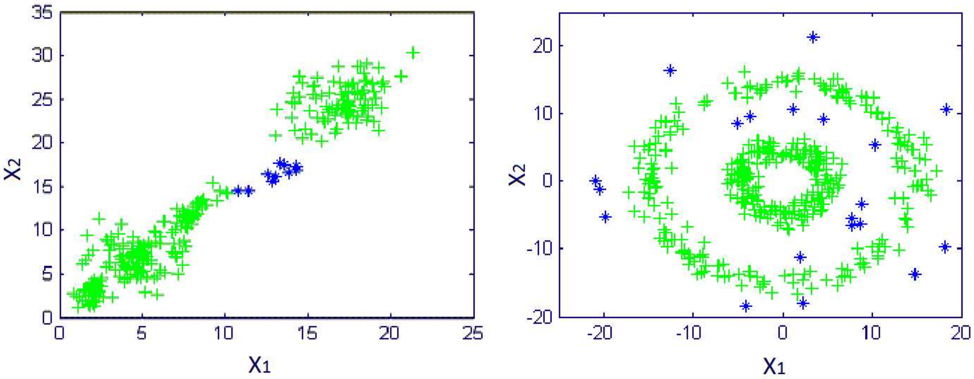

Figure 8.1: Illustration of the simulated datasets. The green points (+) are in-control observations and the blue points (*) are out-of control data that will be tested

Figure 8.2 and 8.3 illustrate several SVDD models with different numbers of clusters. As figures show, when using a global SVDD model with only one cluster, the decision boundary provides a poor decision capability for observations lying between the different modes. This fact is clear in Figure 8.3, where the decision boundary includes observations in one circle and implies a poor detection for outliers located between the two rings. However, the increase of the number of clusters improves the decision boundary and hence the detection capability. Nevertheless, as Figure 8.2 shows, when the number of cluster exceed a certain number, local SVDD models turn to provide complex decision boundaries with poor generalization capability.

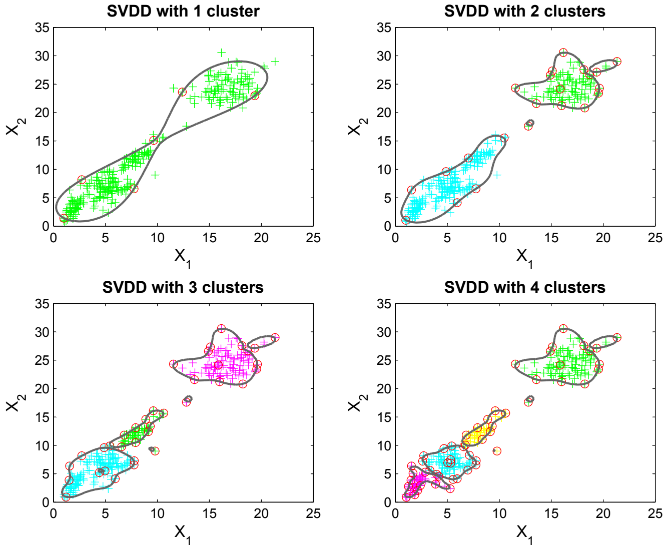

Figure 8.2: SVDD models with different numbers of clusters for the multimodal process. Red circles are Support Vectors

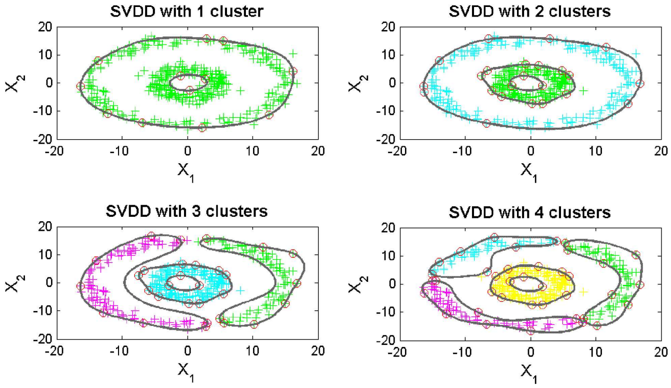

Figure 8.3: SVDD models with different numbers of clusters for the ring process. Red circles are Support Vectors

To define the optimal number of clusters to use, this study applies the method discussed in Section 4. Figure 8.4 illustrates the contribution of different number of Principal Components to the total variance. The number of principal components provides an idea about the number of clusters to use. For the first model, 2 clusters are selected and they explain more than 95 % of the total variance, whereas for the ring data, 5 clusters are selected.

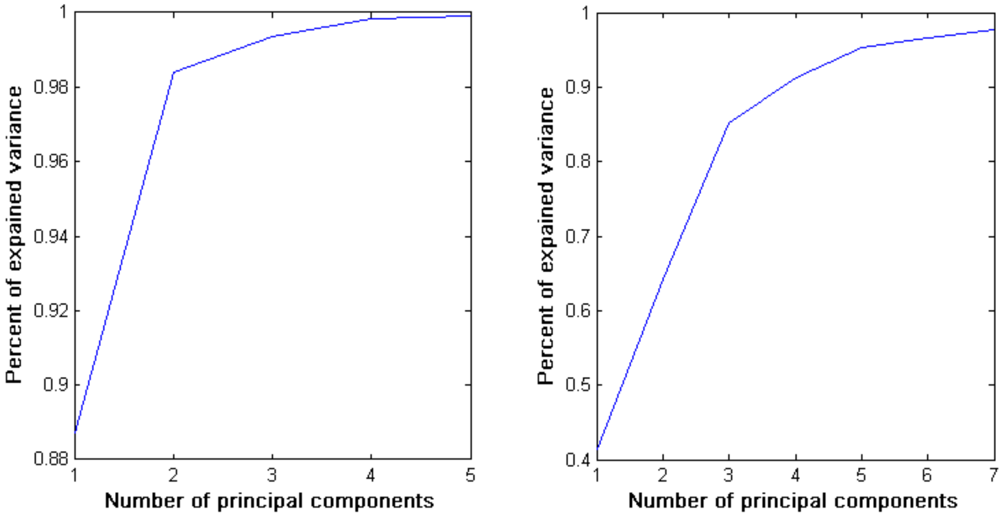

Figure 8.4: Contribution of different numbers of principal components to the total variance. The left figure concerns the first simulated data and the right one is for the ring data

Using 10-fold cross validation samples, where each time 90 % of the in-control data is used for training and 10 % for validation, we smooth the radius R of SVDD in order to have approximately the same alarm rate of 2 % for all models. The illustration of boundary smoothing is present in Figure 8.5. Table 8.1 resumes the obtained results based on several number of clusters. As Table 8.1 shows, despite that global SVDD model provides the same false alarm rate, the detection of faults measured by the detection rate is very poor with respectively detection rates of 22 % and 50 %. This fact is understandable because using only one global SVDD model does not allow modelling the data distribution within the class. Thus, when faults lie between the clusters a single SVDD model would be unable to detect them.

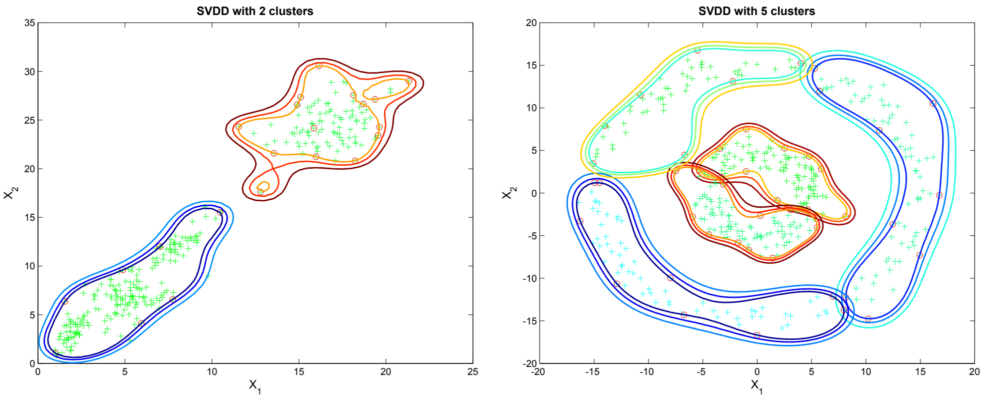

Figure 8.5: Boundary smoothing of SVDD models

However, clustering the data and using several local SVDD models improves the detection rate. In terms of faults detection, the improvement of detection rate is significant with a detection rate of 100 % for Process 1 and 91 % for Process 2. Moreover, eventhough the use of several number of clusters provides an acceptable improvement in detection performance, models with a number of clusters equal to 2, for Process 1 and 5 clusters for the ring data, provide the best results as compared to single SVDD. These models follow the discussed rule to select the best numbers of clusters in Section 7.4.

Table 8.1: Detection results for the multivariate simulated multimodal processes using different models

	SVDD	Kernel k-means SVDD			
Nbr of clusters	-	2	3	4	5
False alarm rate					
Process 1	0.02	0.02	0.02	0.02	0.02
Process 2	0.02	0.022	0.022	0.022	0.02
Detection rate					
Process 1	0.22	**1**	0.82	0.82	0.83
Process 2	0.5	0.75	0.78	0.89	**0.91**

8.2 Fault detection evaluation of Metal Etch process

This section evaluates the proposed process monitoring model using a real case study of a Semiconductor process. The dataset is publicly available and can be downloaded from Eigenvector Research, Inc. Metal Etcher Semiconductor is a chemical process that aims to etch the TiN/Al - 0.5% Cu/TiN/oxide stack with an inductively coupled BCl3/Cl2 plasma. The process consists of a series of six steps. The first two are for gas flow and pressure stabilization. Step 3 is a brief plasma ignition step. Step 4 is the main etch of the Al layer terminating at the Al endpoint, with Step 5 acting as the over-etch for the underlying TiN and oxide layers (Wise et al. (1999)).

To monitor this process, several sensor systems for machine state variables are used. The data corresponds to three experiments run several weeks apart. Data from different experiments have different means and slightly different covariance structure. Faults are intentionally introduced by changing specific manipulated variables (transformer coupled plasma power, radio frequency power, pressure, plasma flow rate and helium chunk pressure). The controlled 8 engineering variables are listed in Table 8.2.

Table 8.2: Variables used to monitor the Metal Etch process

Variable	Definition
1	BCI 3 Flow
2	End point A detector
3	Pressure
4	Radio frequency load
5	Radio frequency phase error
6	Radio frequency power
7	Transformed-coupled Plasma tuner
8	Transformed-coupled Plasma load

The data consists of 128 observations in which variable means over several runs throughout the batch are used, with 108 normal observations and

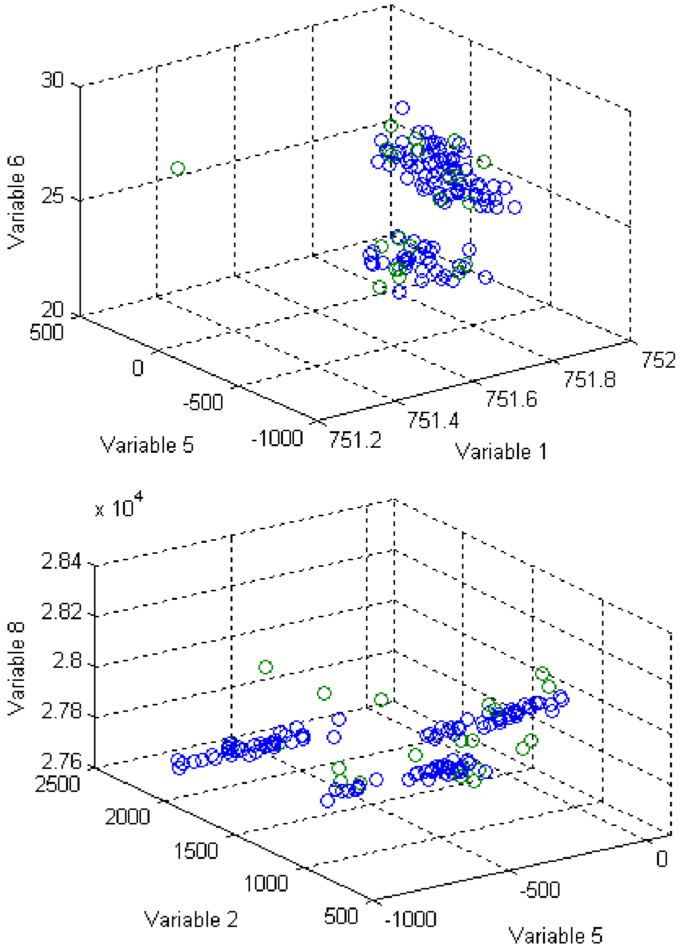

Figure 8.6: Illustration of the data distribution of Metal Etch process. Blue points are in-control data, whereas green points are faults

21 observations with intentionally induced faults. The 56th observation is excluded from the normal sample because it has very few runs compared with other normal observations. To select the number of clusters, Figure 8.7 illustrates the contribution of the number of clusters in the total variance. As it is shown, a model with 3 clusters provides more than 95 % of the total variance and can be considered as an optimal value.

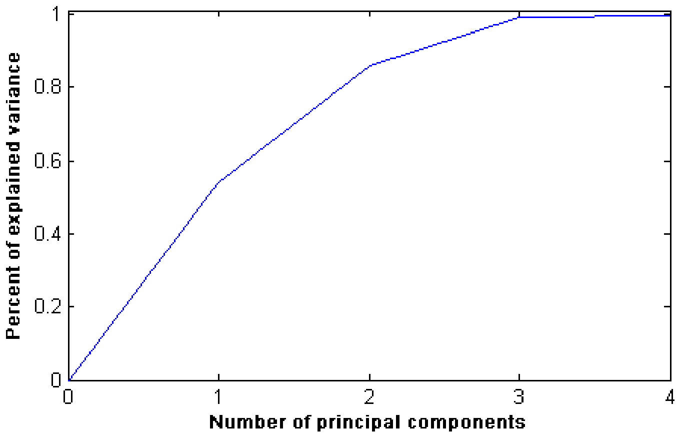

Figure 8.7: Contribution of different numbers of principal components to the total variance

To conduct analysis, 10-fold cross validation sets are constructed from the normal wafers. In this case, in each resampling step 90 % of the normal batches are used to build the SVDD model and the remaining 10 % of normal batches are selected for validation. We smooth the radius R of SVDD in order to have approximately the same alarm rate of 5 % for all models. Table 8.3 shows performances of the different charts based on different numbers of clusters.

As results indicate, although single SVDD provides the same false alarm rate on the validation set, the chart has poor detection abilities measured by a detection rate of 34 %. However, by clustering the data and using several local SVDD models, the detection rate increases with the used number of clusters. If we use the model based on 3 clusters, which follows the selection rule discussed in Section 7.4, then this model provides an acceptable false alarm rate of 4.8 % and a significant improvement of detection with a rate of 67 %.

Table 8.3: Detection results for Metal Etch process using different models

	SVDD	Kernel k-means SVDD			
Nbr of clusters	-	2	3	4	5
False alarm rate					
Mean	0.057	0.054	0.048	0.055	0.056
Detection rate					
Mean	0.34	0.48	**0.67**	0.55	0.55
Fault 1	1	1	1	1	1
Fault 2	0	0	0	0	0.1
Fault 3	0	0	0	0.8	0.2
Fault 4	1	1	1	1	1
Fault 5	0	0.9	1	1	1
Fault 6	0	0	0	0	0
Fault 7	1	1	1	1	1
Fault 8	0.1	0	0.1	0	0.1
Fault 9	0	0	0	0	0
Fault 10	1	1	1	1	1
Fault 11	0	0	0	0	0
Fault 12	1	0	1	1	1
Fault 13	0	1	0	0	0
Fault 14	0.1	1	1	1	1
Fault 15	0.1	1	1	1	1
Fault 16	0	1	1	1	1
Fault 17	0	0	1	0.1	0.4
Fault 18	0	0	1	0.1	0.4
Fault 19	0	0	1	0.1	0
Fault 20	0.9	0.1	1	0.7	0.4
Fault 21	1	1	1	1	1

8.3 Conclusion

This chapter presents a local SVDD fault detection algorithm based on Kernel k-means clustering, which increases sensitivity of the charts to faults. The algorithm detects faults by first deciding to which cluster it belongs then by testing their state with the underlying SVDD. Unlike existing global SVDD control procedures, using membership measurements, the algorithm uses different SVDD models for each cluster. Simulation results show that this procedure gives better detection rate than those with global SVDD model. Future research concerns application and evaluation of adaptive models to handle non-stationary processes. Moreover procedures to train online Kernel k-means and SVDD models can be investigated.

9
Summary and further research

In this thesis we have investigated development and application of Kernel methods to enhance Statistical Process Control procedures. The main results of the thesis are summarized in Section 1, then, further research that can be investigated to improve and extend this work is discussed in Section 2.

9.1 Summary

Chapter 1 proposed a control chart based on adaptive Kernel Principal Components Analysis to model non-stationary nonlinear process behavior. Chapter 2 presented an efficient adaptive KPCA method that allows introduction or elimination of a data block at the same time. This updating method provides a reduced computation cost and allows a fast updating of a large-scale KPCA model. Analysis of the accuracy and stability of the proposed updating method shows that it provides a good tracking of the original matrix with a small reconstruction error. Chapter 3 compare the proposed adaptive KPCA and different PCA control charts. Analysis and comparison with batch models, as well as with other adaptive PCA procedures, shows that first, Batch PCA based control charts are unable to adequately control non-stationary processes since they are based on the use of fixed models for monitoring dynamic processes. Moreover, results show that the proposed chart provides overall competitive detection results.

In order to monitor nonlinear autocorrelated processes, the second part of this thesis is dedicated to the development of residuals control charts. Chapter 4 is devoted to the development of control charts based on SVR method to monitor nonlinear systems without requiring much knowledge about the process structure. The advantage of using SVR method is that this method allows modelling of nonlinear processes without the need to find analytical solutions to describe phenomena of interest. Chapter 5 evaluates the effectiveness of the proposed control charts using univariate and multivariate simulated nonlinear processes. Results show that the used control charts can effectively monitor the process behaviour while guaranteeing an acceptable robustness.

The third part of this dissertation deals with development of local SVDD based control chart to monitor complex and multimodal processes without specifying a probability distribution. Chapter 6 discusses the problem of parameters selection for SVDD and shows that knowledge about the rate of outliers and the dataset complexity plays an important role to set optimal parameters. Then, a monitoring scheme based on separate local models is discussed. This procedure allows simplifying and reducing the complexity of the problem which can help selecting SVDD parameters. Chapter 7 is devoted to the analysis of local SVDD based control chart based on simulated and real case study. Results show that this procedure allows better detection rate while guaranteeing a reduced false alarm rate.

9.2 Further research

Methods used in this thesis could be further enhanced. A monitoring procedure based on a recursive updating of KPCA model accompanied by a recursive determination of kernel parameters could be investigated. Moreover, an interesting problem is the use of adaptive SVR method or adaptive local SVDD control charts for online estimation of the parameters when the process exhibit a non-stationary behavior.

Also, further investigation of kernel functions can be investigated. Indeed, performances of kernel methods depend on the used kernel function. However, current kernel selection methods are mainly dedicated to learning the kernel function for supervised problems. In contrast, for unsupervised problems, such as KPCA, SVDD or Kernel k-means, no general rule for kernel selection is used and existing methods choose the kernel function empirically from a given set of candidates. Moreover, only some standard kernel functions such as Gaussians or Polynomials are investigated. Therefore, further theoretical and empirical analysis of kernel functions is very important. For example one can investigate the adequacy of certain functions to the geometrical distribution of data. Also, one can further investigate criteria to use in order to optimize the kernel parameters in the case of data reduction, one class classification or Kernel clustering. Another interesting research that is linked to parameter selection is the development of complexity measures of datasets. This could have an impact on the selection of kernel methods parameters, by either reducing the optimization parameters space or by approximating the value of optimal parameters.

Bibliography

[1] Alwan, L.C. and Roberts, H.V., 1995, The problem with misplaced control limits, Applied Statistics, 44, 269-278.

[2] Apley, D.W. and Tsung, F., 2002, The Autoregressive T2 Chart for Monitoring Univariate Autocorrelated Processes, Journal of Quality Technology, 34, 80-96.

[3] Bu, H., Wang, J. and Huang, X., 2009, Fabric defect detection based on multiple fractal features and support vector data description, Engineering Applications of Artificial Intelligence, 22, 224-235.

[4] Bagshaw, M. and Johnson, R. A., 1975, The effect of serial correlation on the performance of CUSUM tests II, Technometrics, 17, 73-80.

[5] Banerjee, A, Burlina, P. and Diehl, C., 2006, A Support Vector Method for Anomaly Detection in Hyperspectral Imagery, IEEE Transactions on geoscience and remote sensing, 44, 2282-2291.

[6] Blake, C. and Merz, C., 1998, Uci repository of machine learning databases, Online: http://archive.ics.uci.edu/ml/.

[7] Benneyan, J.C., Lloyd, R.C. and Plsek, P.E. 2003, Statistical Process Control as a Tool for Research and Healthcare Improvement, Quality and Safety in Health Care, 12, 458-464.

[8] Choi, S. and Lee, I., 2004, Nonlinear dynamic process monitoring based on dynamic kernel PCA, Chemical Engineering Science, 59, 5897-5908

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

[10] Callao, M. and Rius, A., 2003, Time series: a complementary technique to control charts for monitoring analytical systems, Chemometrics and Intelligent Laboratory Systems, 66, 79-87.

[11] Choi, S.W., Martin, E.B., Morris, A.J. and Lee, I., 2006, Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes, Industrial Engineering and Chemical Research, 45, 3108-3118.

[12] Choi, S.W., Lee, C., Lee, J.MP., Park, J.H. and Lee, I.B., 2005, Fault detection and identification of nonlinear processes based on KPCA, Chemometrics and Intelligent Laboratory Systems, 75, 55-67.

[13] Cui, P., Li, J. and Wang, G., 2007, Improved kernel principal component analysis for fault detection, Expert Systems with Applications, 36, 1423-1432.

[14] Dooley, J. and Guo, K., 1992, Identification of change structure in statistical process control, International Journal of Production Research, 30, 1655-1669.

[15] Downs, J.J and Vogel, E.F, 1993, A plant-wide industrial process control problem. Computers and Chemical Engineering, 17, 245-255.

[16] Elbasi, E., Zuo, L., Mehrota, K., Mohan, C. and Varshney, P., 2005, Control Charts Approach for Scenario Recognition in Video Sequences, Turkish journal of electrical engineering and computer sciences, 13, 303-309.

[17] Evangelista, P., Embrechts, M. and Szymanski, B., 2007, Some properties of the Gaussian kernel for one class learning, Lecture Notes in Computer Science, 4668, 269-278.

[18] Girolami, M., 2002, Mercer Kernel-Based Clustering in Feature Space, IEEE Transactions on Neural Networks, 13, 780-784.

[19] Golub, G.H., and Loan C.F.V., 1989, Matrix computations, Baltimore, MD 21218: The John Hopkins University Press.

[20] Henze, N. and Zirkler, B., 1990, A class of invariant consistent tests for multivariate normality, Communications in Statistics - Theory and Methods, 19, 3595-3617.

[21] Hotelling, H., 1933, Analysis of a complex of statistical variables into components. Journal of Educational Psychology, 24, 417-441.

[22] Hwang, D. and Han, C., 1999, Real-time monitoring for a process with multiple operating modes, Control Engineering Practice, 7, 891-902.

[23] Hwarng, H.B. and Hubele, N.F., 1993, X-bar control chart pattern identification through efficient off-line neural network training, IEEE Transactions, 25, 27-40.

[24] Harris, T.J. and Ross, W.H., 1991, Statistical process control procedures for correlated observations, Canadian Journal of Chemical Engineering, 69, 48-57.

[25] Haggan, V. and Ozaki, T., 1981, Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model, Biometrika, 68, 189-196.

[26] Hoffmann, H., 2007, Kernel PCA for novelty detection, Pattern Recognition, 40, 863-874.

[27] Hoegaerts, L., Lathauwer, L., Goethalsa, I., Suykens, J.A.K, Vandewalle, J. and De Moor, B., 2007, Efficiently updating and tracking the dominant kernel principal components, Neural Networks, 20, 220-229.

[28] Jarque, C.M. and Bera, A.K., 1987, A test for normality of observations and regression residuals, International Statistical Review, 55, 163-172.

[29] Jain, A.K. and Dubes, R.C., 1998, Algorithms for clustering Data, Upper Saddle River, Prentice Hall.

[30] Jiang, W., Tsui, K.L. and Woodall, W.H., 2000, A new SPC monitoring method: The ARMA chart, Technometrics, 42, 399-410.

[31] Jamal, A., Seyed, T., Akhavan, N. and Babak, A., 2007, Artificial neural networks in applying MCUSUM residuals charts for AR(1) processes, Applied Mathematics and Computation, 189, 1889-1901.

[32] Longnecker, M. T. and Ryan, T. P., 1992, Charting Correlated Process Data, Technical Report No.166, Texas A&M University, Department of Statistics.

[33] Loredo, E.N., Jaerkpaporn, D. and Borror, C.M., 2002, Model-based control chart for autoregressive and correlated data, Quality and Reliability Engineering International 18, 489-496.

[34] Lee, J.M., Yoo, C., Choi, S.K., Vanrolleghemb, P.A. and Lee, I., 2004, Nonlinear process monitoring using kernel principal component analysis, Chemical Engineering Science, 59, 223-234.

[35] Lee, D.S., Park, J.M. and Vanrolleghem, P.A., 2005, Adaptive multiscale principal component analysis for on-line monitoring of a sequencing batch reactor, Journal of Biotechnology, 116, 195-210.

[36] Liu, X., Xie, L., Kruger, U., Littler, T., and Wang, S., 2009, Moving window kernel PCA for adaptive monitoring of nonlinear processes, Chemometrics and Intelligent Laboratory Systems, 96, 132-143.

[37] Liu, F. and Wu, C., 2006, Improvement on Multivariate Statistical Process Monitoring Using Multi-scale ICA, in Independent Components Analysis and Blind signal Separation, Springer.

[38] Li, W., Yue, H.H., Valle-Cervantes, S. and Qin, S.J., 2000, Recursive PCA for adaptive process monitoring, Journal of Process Control, 10, 471-486.

[39] Lee, J., Yoo, C., Choi, S., Vanrolleghem, P.A. and Lee, I., 2004, Nonlinear process monitoring using kernel principal component analysis Chemical Engineering Science, 59, 223-234

[40] Montgomery, D.C., 2005, Introduction to Statistical Quality Control, 5th Edition.

[41] Mercer, J., 1909, Functions of positive and negative type and their connection with the theory of integral equations, Philosophical Transactions of the Royal Society, London, A 209, 415-446.

[42] Müller, K.R., Smola, A., Ratsch, G., Schölkopf, B., Kohlmorgen, J. and Vapnik, V., 1997, Predicting time series with support vector machines, in: Gerstner, W., Germond, A., Hasler, M. and Nicoud J.D., Artificial Neural Networks ICANN'97, Lecture Notes in Computer Science 1327, 999-1006.

[43] Messaoud, A., Weihs, C. and Hering, F., 2008, Detection of chatter vibration in a drilling process using multivariate control charts. Computational Statistics & Data Analysis, 52, 3208-3219.

[44] Magalhaes, M.S., Antonio, D., Costa, E., Francisco, F.B. and Netoc, D.M., 2006, Adaptive control charts: A Markovian approach for processes subject to independent disturbances, International Journal of Production Economics, 99, 236-246.

[45] Nomikos, P. and MacGregor, J.F., 1995, Multivariate SPC charts for monitoring batch processes, Technometrics, 37, 41-59.

[46] Noorossana, R. and Vaghefi, S.J.M., 2006, Effect of autocorrelation on performance of the MCUSUM control chart, Quality and Reliability Engineering International, 22, 191-197.

[47] Park, C.H. and Park, H., 2005, Nonlinear Discriminant Analysis using kernel functions and the gerneralized Singular Value Decomposiotion, Journal of Matrix analysis and applications, 27, 87-102.

[48] Pacella, M. and Semeraro, Q., 2007, Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring, Computers and Industrial Engineering, 52, 502-520.

[49] Ping-Feng, P. and Chih-Sheng, L., 2005, Using support vector machines to forecast the production values of the machinery industry in Taiwan, International Journal of Advanced Manufactoring Technology, 27, 205-210.

[50] Psarakis, S. and Papaleonida, G. E. A., 2007, SPC Procedures for Monitoring Autocorrelated Processes, Quality Technology and Quantitative Management, 4, 501-540.

[51] Park, Y., 2005, A Statistical process control approach for network inrusion detection, Docoral dissertation, Georgia Institute of Technology.

[52] Ryan, T.P., 2000, Statistical Methods for Quality Improvement, 2nd edition. John Wiley & Sons, New York.

[53] Ricker, N. L., 2008, Tennessee Eastman Challenge. Available at http://depts.washington.edu/control/LARRY/TE/download.html.

[54] Ruixiang, S., Fugee, T. and Qu, L., 2007, Evolving kernel principal component analysis for fault diagnosis, Computers and Industrial Engineering, 53, 361-371.

[55] Sukchotrat, T., Kim, S.B. and Tsung, F., 2010, One-class classification-based control charts for multivariate process monitoring IIE Transactions, 42, 107120.

[56] Shin, H., Eom, D. and Ki, S., 2005, One-class support vector machines: an application in machine fault detection and classification, Computers and Industrial Engineering, 48, 395-408.

[57] Schölkopf, B. and Smola, A.J., 2002, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press.

[58] Sun R., and Tsung F., A., 2003, Kernel-distance-based multivariate control charts using support vector methods, International Journal of Production Research, 41, 2975-2989.

[59] Stapenhurst, T., 2005 ,Mastering Statistical Process Control: A Handbook for Performance Improvement Using SPC Cases, Elsevier.

[60] Stone, R. and Taylor, M., 1995, Time series models in statistical process control: considerations of applicability, The Statistician, 44, 227-234.

[61] Sato, J.R., Costafreda, S., Morettin P.A. and Brammer, M.J., 2008, Measuring Time Series Predictability Using Support Vector Regression. Communications in Statistics, Simulation and Computation, 37, 1183-1197.

[62] Shi, Z., Tamura, Y. and Ozaki, T., 2001, Monitoring the stability of BWR oscilation by nonlinear time series modeling, Annals of Nuclear Energy, 28, 953-966.

[63] Schölkopf, B., Smola, A. and Muller, K.B., 1998, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, 10, 1299-1319.

[64] Tax, D., and Duin, R., 1999, Support vector domain description, Pattern Recognition Letters, 20, 1191-1199.

[65] Tax, D. and Duin, R., 2004, Support Vector Data Description, Machine Learning, 54, 45-66.

[66] Thissena, U., Van Brakela, R., Weijerb, A.P., Melssena, W.J. and Buydens, L.M., 2003, Using support vector machines for time series prediction, Chemometrics and Intelligent Laboratory Systems, 69, 35-49.

[67] Tay, F.E.H. and Cao, L.J., 2001, Application of support vector machines in financial time series forecasting, Omega, 29, 309-317.

[68] Vapnik, V., 1998, Statistical Learning Theory, Wiley.

[69] Woodall, W.H., 2000, Controversies and contradictions in statistical process control, Journal of Quality Technology, 32, 341-350.

[70] Wise, B.M., Gallagher, N.B, Butler, S.W, White, D.D. and Barna G.G., 1999, 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, 379-396.

[71] Wang, M.P. and Jones, M.C, 1995, Kernel Smooting, Chapman and Hall, London, UK.

[72] Yoo, C.K. and Lee, I., 2006, Nonlinear multivariate filtering and bioprocess monitoring for supervising nonlinear biological processes, Process Biochemistry, 41, 1854-1863.

[73] Zhiqiang, G. and Zhihuan, S., 2008, Online monitoring of nonlinear multiple mode processes based on adaptive local model approach, Control Engineering Practice, 16, 1427-1437.

[74] Zhang, J. and Yang, X., 2000, Multivariate Statistical Process Control, Beijing, China: Chemical Industry.

[75] Zhang, N.F., 1998, A statistical control chart for stationary process data, Technometrics, 40, 24-39.