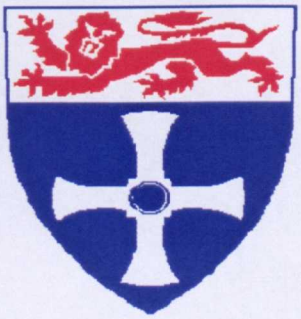
UNIVERSITY OF
NEWCASTLE UPON TYNE



Modelling and Abnormal Change Detection in Multivariate Signals and Systems using Subspace Projection Techniques

by
Sukhbinder Kumar

A Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

School of Chemical Engineering and Advanced Materials

The University of Newcastle upon Tyne

September 2004

In creating, the only hard thing's to begin

James Russel Lowell

Dedicated to my parents, Shri Jaipal Sharma and Shrimati Shanti Devi

Publications from the Thesis

Conference

1. Kumar, S., Martin, E.B, Morris, A.J. (2002). Detection of process model change in PCA based performance monitoring. *Proceedings American Control Conference, Anchorage, Alaska*, May 8-10, 2719-2724

2. Kumar, S., Martin, E.B, Morris, A.J. (2002). Detection of process model change in PLS based performance monitoring. *Proceedings of IFAC 15th Triennial World Congress, Barcelona, Spain*, July 21-26, (paper number, 2144)

3. Kumar, S., Thennadil, S.N., Martin, E.B., Morris, A.J (2003). Adaptive Partial Least Squares with Applications to Process Monitoring. *Proceedings of IFAC SAFEPROCESS, Washington D.C, (USA)*, June 9-11,185-190

4. Kumar, S., Martin, E.B., Morris, A.J (2003). Process modelling using dynamic partial least squares. *Proceedings of PLS'03 Conference, Lisbon (Portugal)*, September, 15-17, 485-496.

5. Kumar, S., Martin, E.B and Morris, A.J (2004). Analysis of non-linear partial least squares algorithms. *Proceedings of 7th International Symposium on Dynamics and Control of Process Systems, DYCOPS*, Boston, U.S.A, July 5-7, (paper number, 103)

6. Sharma, S.K, Kruger, U, Irwin, G, Kumar, S (2004). A Covariance based non-linear partial least squares algorithm. *Proceedings of UKACC, University of Bath*, September 6-9, (paper number, 241).

Journal

7. Kumar, S., Martin, E.B, Morris, A.J. Detection of change in variance-covariance structure in PCA based performance monitoring scheme. *Submitted to Journal of Process Control.*

8. Kumar, S., Martin, E.B, Morris, A.J. Detection of change in cross covariance structure in a PLS based performance monitoring scheme. *Submitted to Journal of Process Control.*

9. Kumar, S., Thennadil, S.N., Martin, E.B, Morris, A.J. Recursive partial least squares with application to process monitoring. *Submitted to Computers and Chemical Engineering*

10. Kumar, S., Martin, E.B, Morris, A.J. Integrated dynamic partial least squares. *Submitted to Computers and Chemical Engineering*

11. Kumar, S., Kruger, U., Martin, E.B, Morris, A.J. A covariance based non-linear partial least squares algorithm. *Submitted to Journal of Chemometrics.*

Abstract

The focus of the thesis is black-box modelling and the detection of abnormal events in multivariate systems. Subspace projection techniques have been widely applied for the modelling and monitoring of multivariate systems. The popularity of these techniques stems from the fact that these methods can address multicollinearity, a problem commonly encountered when modelling using ordinary least squares with strongly correlated input (process) variables. The subspace techniques of principal component analysis and partial least squares are the methodology of specific interest throughout the thesis.

Several non-linear PLS algorithms have been proposed over the last decade. In this thesis analysis of existing non-linear PLS algorithms is undertaken. In particular, following a mathematical analysis of the non-linear PLS algorithm proposed by Baffi et al., (1999(a)), it is proven that the algorithm is a non-linear extension of reduced rank regression. It is also argued that a 'true' non-linear generalization to linear PLS should be based on the maximization of a 'non-linear covariance' function if the spirit of linear PLS is to be preserved in its non-linear extension. A mathematical analysis of the algorithm of Wold et al., (1989) is undertaken and it is proven that this algorithm makes an attempt to maximize the non-linear covariance function but with certain limitations. The limitations of the algorithm of Wold et al., (1989) are addressed in two new non-linear PLS algorithms, NLPLS1 and NLPLS2. Also following a critical analysis, all existing non-linear PLS algorithms are divided into three categories namely, quick and dirty, covariance based and error based depending on the underlying objective functions optimized by the algorithms. An application of PLS as a parameter estimator is explored and it is shown that when a subspace of dimension A ($< K$, number of input variables) is correlated with the output variable and a PLS1 model is built using A latent variables then PLS1 gives an unbiased estimate of the parameters.

One approach to extending PLS to take into consideration the dynamics of the process is to replace the inner static relationship between the t- and u-scores of conventional PLS by a dynamic relationship. An algorithm that integrates the dynamics of the data within a PLS framework is proposed. The performance of the algorithm is evaluated against alternative methodologies presented in the literature using an artificial data set and two simulations of chemical processes.

The second aspect of the thesis is concerned with detecting abnormal changes in variance-covariance structure of variables. The conventional PCA based monitoring scheme is known to be insensitive to small changes in the variance-covariance structure of variables. A new monitoring scheme that derives a monitoring statistic from the PCA model identification procedure is proposed. The proposed scheme is compared with conventional PCA based monitoring scheme on two artificial data sets and a data set generated from a continuous stirred tank reactor system.

A new monitoring scheme for detecting changes in the cross-covariance structure (between input and output variables) in a PLS based monitoring scheme is proposed. The derivation of monitoring statistic requires that a recursive algorithm exists for identifying the PLS model parameters. A new recursive PLS algorithm is derived and the statistic derived from it is used to detect change in parameters of an artificial system before applying to detect fouling in the heat exchanger of a CSTR system. The performance of the proposed scheme is also compared with conventional PLS based monitoring scheme.

Acknowledgements

I take this opportunity to sincerely thank my supervisors, Prof Elaine Martin, Professor of Industrial Statistics, School of Chemical Engineering and Advanced Materials and Prof Julian Morris, Head of School of Chemical Engineering and Advanced Materials, for giving me an opportunity to work under their esteemed supervision and providing me with support, encouragement, and guidance throughout the period of my PhD. I am especially indebted to Prof Elaine Martin for doing an excellent job in reading the drafts of my thesis. I would also like to thank the Engineering and Physical Sciences Research Council (EPSRC) and the Centre for Process Analytics and Control Technology for funding this research.

I owe a great deal to Dr Uwe Kruger, working with whom was a great learning experience. I am especially thankful to him and his wife Xun Wang for their hospitality and making my stay comfortable during my visits to Belfast. Thanks are due to Dr. Suresh Thennadil for his feedback and expert advice on many topics.

I am thankful to my friends and office mates: Katarina Novakovic, for being a wonderful and concerned neighbour and for sharing her views on all topics ranging from 'problems in raising a child' to 'problems in statistics'; Fabrice Pabois, for helping me whenever there was any problem with my machine or software, Tom Muscika, Zhen Lu for making a friendly environment in the office.

I express my gratitude to Sri Dham and Clara for providing me with shelter when I first came to the UK, Bhakti Rasa Dasa and Kirtida Devi Dasi for providing me with spiritual strength whenever the goings were tough. My friends of all times Rameshwar Miniyar, Sridhar Kurse, and Prashant Joshi were as helpful as ever and I thankfully acknowledge their support.

I gratefully acknowledge the help of my teachers Mrs V.K Sehgal and Mrs Chaman Gupta without whose invaluable support at the beginning of my studies, it would not have been possible to come this far. I am grateful to my teachers, Dr Rameshwar Jha, Prof I.J Nagrath and Prof R.C.Jain for motivating and supporting me to follow the path of learning. The support of my friends and colleagues, Prof S.N.Sharan and Dr Harish Chandra, at BITS, Pilani is highly appreciated.

Last but by no means least I thank my family: my parents, who were always there with me in all the ups and downs of life and this thesis is dedicated to them; my wife, for her patience, constant support and encouragement; my sweet and beautiful daughter, for giving me the invaluable present of love; my sister, Usha, for taking all responsibilities back home in India and making the family proud by doing well in her studies; my sister, Pushpa and my younger brother, Sanjeev, for their support and encouragement.

Table of Contents

List of Tables

List of Figures

List of Symbols

$\Delta \mathbf{w}_i$	Incremental change in the weight vector $\mathbf{w}_i$
$\mathbf{d}_i$	$[\Delta \mathbf{w}_i \ \Delta \mathbf{c}_i]^T$
a_i	Coefficient of $u(n-i)$ in the inner ARX model
$\mathbf{X}_{dyn}$	Input data matrix augmented with lagged variables
$H_i(z)$	Input-output transfer function for i^{th} pair of latent variables
t_i	i^{th} input latent variable
u_i	i^{th} output latent variable
θ_i	Parameter vector of ARX model for i^{th} pair of latent variables
$\hat{u}_i$	Prediction of i^{th} output latent variable
$\mathbf{B}_{dyn}$	Regression matrix linking $\mathbf{X}_{dyn}$ to $\mathbf{Y}$
$\| \ \|$	2^{nd} norm of a vector
H_1	Alternative hypothesis
$\rightarrow$	Asymptotically approaches to
$\mathbf{k}$	Augmented estimation function (primary residuals)
$Cov(.)$	Covariance function
$\mathbf{\Sigma}_f$	Covariance matrix of faulty data
$\mathbf{\Sigma}$	Covariance matrix of input vector $\mathbf{x}$
$\mathbf{\Sigma}_0$	Covariance matrix of normal operating condition data
$\mathbf{\Sigma}_t$	Covariance matrix of $\mathbf{t}$
$\mathbf{\Sigma}$	Covariance matrix of $\mathbf{x}$
S_n	Cumulative sum of first n observations
$\mathbf{z}_n$	Efficient score at time n
$\hat{\boldsymbol{\beta}}_{OLS}$	Estimate of $\boldsymbol{\beta}$ using ordinary least squares
$\hat{\boldsymbol{\beta}}_{PLS}$	Estimate of $\boldsymbol{\beta}$ using partial least squares
$\hat{\boldsymbol{\beta}}_{PCR}$	Estimate of $\boldsymbol{\beta}$ using principal component regression
$\mathbf{k}_i$	Estimating function (primary residuals) corresponding to i^{th} loading vector
$\boldsymbol{\gamma}$	Fault (change) vector

$\mathbf{r}_n$	Improved residuals at time n
$\Delta\mathbf{c}_i$	Incremental change in the weight vector $\mathbf{c}_i$
$\mathbf{I}_n$	Information matrix calculated using observations from time 1 to n
$\mathbf{X}_i$	Input data matrix after (i-1) steps of deflation
$\mathbf{X}_{aug}$	Input data matrix $\mathbf{X}$ augmented with non-linear terms
J_n	Instantaneous estimate of objective function
$\mathbf{w}_i$	i^{th} input weight vector of PLS model
$\mathbf{v}_i$	i^{th} output weight vector of PLS model
$\mathbf{p}_i$	i^{th} input loading vector of PLS model
x_i	i^{th} input variable
$\mathbf{p}_i$	i^{th} loading vector
$\mathbf{q}_i$	i^{th} output loading vector of PLS model
λ	Lagrange multiplier (eigenvalue of covariance matrix)
η	Learning rate
n_0	Length of window from the current time n over which primary residuals are
S_n	Local approach based statistic
$\mathbf{X}_n$	Matrix containing all observations from time 1 to n
$\mathbf{Z}_i$	Matrix containing derivatives of $\mathbf{f}(.)$ with respect to $\mathbf{w}_i$ and function $\mathbf{c}_i$
$\mathbf{Z}$	Matrix relating $\mathbf{X}_j$ and $\mathbf{X}_i$ for (i< j) (Chapter, 2)
$\boldsymbol{\mu}$	Mean (population) of input vector
$\boldsymbol{\theta}$	Model parameter vector
$\boldsymbol{\theta}_0$	Model parameter vector under normal conditions
$\boldsymbol{\theta}_0$	Model parameter vector under normal conditions
$\omega(\mathbf{p}_i)$	Neighbourhood of $\mathbf{p}_i$ (excluding $\mathbf{p}_i$)
$\mathbf{f}(.)$	Non-linear function for the inner scores model (Chapter 3)
$\boldsymbol{\tau}$	Non-linearly transformed t-scores vector
H_0	Null hypothesis
J	Objective function
$\mathbf{Y}_i$	Output data matrix after (i-1) steps of deflation
$\mathbf{D}$	Parameter matrix of AR model

$\mathbf{c}_i$	Parameter vector of function $\mathbf{f}(.)$ for the i^{th} latent variable
$\hat{\mathbf{u}}_i$	Predicted u-score for the i^{th} latent variable
$\hat{\mathbf{u}}_i$	Predicted u-scores for the i^{th} latent variable
$\hat{\mathbf{x}}$	Predicted value of $\mathbf{x}$ from PCA model
$\hat{u}_i^{'}$	Prediction of $u_i^{'}$
p_θ	Probability density function with parameter θ
α	Probability of type I error
Θ	Process (System) parameter vector
Θ_0	Process parameter vector under normal conditions
$t_i^{'}$	Projection of deflated $\mathbf{x}$ on i^{th} input weight vector $\mathbf{w}_i$
$u_i^{'}$	Projection of deflated $\mathbf{y}$ on i^{th} input weight vector $\mathbf{v}_i$
u_i	Projection of $\mathbf{y}$ on i^{th} input weight vector $\mathbf{v}_i$
b_i	Regression coefficient of the inner linear model for the i^{th} latent variable
β	Regression vector
$\mathbf{e}_i$	Residual vector for the i^{th} inner scores model
σ_x	Standard deviation of variable x
t_0	Threshold for local approach based statistic
n_1	Time at which the local statistic starts monitoring
$\mathbf{t}_i$	t-scores vector for the i^{th} input latent variable
$\mathbf{u}_i$	u-scores vector for the i^{th} latent variable
$\mathbf{W}_A$	Weight matrix $[\mathbf{w}_1\ \mathbf{w}_2 \ldots \mathbf{w}_A]$
z	Weighted sum in EWMA chart
$G(\mu, \Sigma)$	Gaussian probability density function with mean vector μ and covariance matrix Σ
(p, q, d)	Triplet indicating ARX model order: p = number of lags in output, q = number of lags in input, d = number of delays
a	Auto-regressive model order
A	Number of latent variables
$\mathbf{B}$	Regression matrix
$\mathbf{e}$	Residual error vector of PCA model
$E\{\ \}$	Statistical expectation operator

$\mathbf{f}(.)$	Mapping between process and model parameters (chapter 6)
K	Number of input variables
$\mathbf{M}$	Jacobian matrix of primary residuals
M	Number of output variables
n	Instantaneous time (chapter 4, chapter 7)
n	current time point up to which observations have been collected (chapter 6)
N	Number of observations of input and output vectors
$\mathbf{P}$	Loading matrix
$\mathbf{p}$	Loading vector
r	Instance at which fault (change) occurs in the process
t	Instantaneous time (chapter 6)
$\mathbf{T}$	Scores (t-scores) matrix $[\mathbf{t}_1\ \mathbf{t}_2 \ldots \mathbf{t}_A]$
T	Target value
$\mathbf{t}$	Vector of latent variables
$\mathbf{X}$	Input data matrix
$\mathbf{x}$	Input vector
$\mathbf{Y}$	Output data matrix
$\mathbf{y}$	Output vector

CHAPTER 1

Introduction

This thesis is concerned with the modelling and detection of abnormal changes in multivariate systems (processes). The thesis is divided into two parts. The modelling of multivariate systems is first considered prior to looking at abnormal change detection. More specifically the tools considered for both modelling and abnormal change detection are constrained within the family of multivariate subspace projection techniques. The aim of this chapter is to provide a brief introduction to the problems of modelling and abnormal change detection and present the key contributions of the thesis.

1.1 Problem Formulation

This section presents the background from a mathematical perspective to the two areas that are the focus of the thesis, modelling and abnormal change detection in multivariate systems.

1.1.1 System Modelling

The objective of system modelling is to develop a mathematical representation of a physical system. The mathematical model of the system is required to explain the behaviour of the physical system under study and can be used for several applications including, for example, prediction, simulation and control. Broadly there are two approaches to developing a model of a system. The first is based on understanding the physics and chemistry of the system and then defining the mathematical equations governing the system. This approach, known as first principle based modelling, has been the most popular in science and Newton's famous equation of motion relating force, mass and acceleration, $F = ma$, is perhaps one of the earliest examples of this approach. While the model developed by adopting this approach has great physical significance and closely describes the 'truth' underlying the system, there are major challenges when it comes to developing models for engineering systems. Given the complexity and size of modern engineering systems, it is very difficult and time consuming to develop a comprehensive first principle model.

The second approach, known as black box modelling or empirical modelling, has become very popular with the engineering community in the last three decades. The idea behind this

approach is to use the measured data from the underlying system to develop a mathematical model. Black box modelling may, therefore be thought of as a mapping from the measured data to a model. The main advantages of this approach are simplicity of the model, less expensive in terms of time and effort and since the technique is not confined to any particular system, this methodology is generic in terms of its applicability. The main drawback of this approach is that the model developed cannot, in general, provide physical understanding of the system, and therefore may be far from the underlying 'truth' of the system. This however is acceptable to those practitioners who are not necessarily concerned with the 'truth' and who are willing to accept 'that the model works' and is hence 'fit for purpose'.

In the engineering literature, a distinction between the two approaches is made, the former is termed 'system modelling' and the latter 'system identification'. The theory of black box modelling is well developed and a number of text books (Ljung, 1999; Soderstrom and Stoica, 1988; Ljung and Soderstrom, 1983) have been specifically dedicated to this subject.

The problem of black box modelling can be formulated as follows. Consider the system shown in Figure 1.1. $\mathbf{x}(t) \in \mathrm{R}^K$ denotes the vector containing K input signals to the system and $\mathbf{y}(t) \in \mathrm{R}^M$ describes the vector of M measurable output signals of the system. The focus of this thesis is the case where K and M are greater than unity. The vector $\mathbf{h}(t) \in \mathrm{R}^M$ denotes the measurement noise in the output variables.

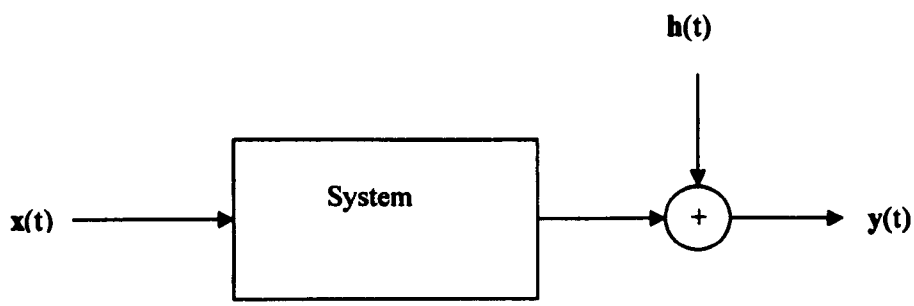


Figure 1.1: Illustration of black box modelling

Given N measurements of the input, $\mathbf{x}$ and output $\mathbf{y}$, the problem is to identify a suitable model of the system using these measurements. An important decision that needs to be taken to solve the above problem is the selection of a suitable structure for the model. To make this choice, the user has to make a hierarchy of decisions. First, the user has to decide between a

'static' and 'dynamic' model. Once this decision has been taken, the next issue to be addressed is to decide between a 'linear' or 'non-linear' model. Before a decision is taken on these issues, it is very important to understand what the model is to be used for. If, for example, the model is to be used for the control of a system, a dynamic model should be developed. If, on the other hand, the model is to be used for prediction, a static model may be appropriate. Another important decision that a user needs to make is to chose between an 'accurate' and a 'simple' model. It might be the case that the user can get a more 'accurate' model but at the cost of increased complexity. The issue is whether to opt for greater 'accuracy' and less 'simplicity' or for greater 'simplicity' and less 'accuracy'. There are no hard and fast rules and the user's experience, intuition and insights into the system plays a major role in making these choices. For this reason many experts prefer to call black box modelling an 'art'.

The multivariate modelling techniques considered in this thesis belong to the family of multivariate subspace projection techniques. These techniques are especially suitable for systems where a large number of variables are measured, that is, the values of K and M in the system shown in Figure 1.1 are large. The modelling task in this situation is more challenging because the measured variables are often highly correlated and corrupted by noise. To develop a model from this type of data, subspace projection techniques have been widely applied. The philosophy behind these techniques is that, behind the large number of variables that are accessible and measured, there lie a smaller number of independent variables which are latent (hidden), and that all the events in a system are manifestation of variations of these latent variables. The objective of subspace projection techniques is to extract the latent variables by analysing the measured variables. The underlying methodology of these techniques is that the original variables are projected onto a subspace spanned by the latent variables. Usually the number of latent variables required to explain a 'sufficient' amount of information contained in the measured variables is smaller than the original number of variables. Any variation orthogonal to the space spanned by the latent variables is considered to be noise and is therefore discarded. The subspace projection techniques therefore not only reduce the dimensionality of the problem but also act as a filter to remove the noise.

The subspace projection techniques can be divided into two families. The first consists of Principal Component Analysis (PCA), Partial Least Squares (PLS), and Canonical Correlation Analysis (CCA) with the second family incorporating a set of algorithms collectively known as Numerical Algorithms for Subspace State Space System Identification (N4SID). While the first class of algorithms are relatively old and have found a variety of

applications in different disciplines of science and engineering, the second class of algorithms were developed in the late 1980's for building state space models of a system. The aim of this thesis is restricted to the first class of subspace projection techniques with particular emphasis on modelling using PLS which is the most recent member of this family.

The conventional PLS algorithm assumes that a linear relationship exists between the input and output variables. This assumption may not be valid in modelling data collected from complex (chemical) processes where the relationship may be significantly non-linear. To integrate non-linear features within the PLS framework, several non-linear PLS algorithms have been proposed over the last decade. It is therefore, essential to analyze which algorithm represents a 'true' non-linear extension to the PLS algorithm. In particular, a mathematical analysis of the non-linear PLS algorithm proposed by Baffi et al., (1999(a)) is undertaken and it is proven that the algorithm is a non-linear extension of reduced rank regression. It is argued that a 'true' non-linear generalization of linear PLS should be based on the maximization of 'non-linear covariance' function if the spirit of linear PLS is to be preserved in its non-linear extension. A mathematical analysis of the algorithm of Wold et al., (1989) revealed that despite this algorithm being considered 'complicated', it makes an attempt to maximize the non-linear covariance function but with certain limitations. The limitations of the algorithm of Wold et al., (1989) are addressed in two new non-linear PLS algorithms, NLPLS1 and NLPLS2. Also following a critical analysis, all existing non-linear PLS algorithms are divided into three categories namely, quick and dirty, covariance based and error based depending on the underlying objective functions optimized by the algorithms.

In most of the applications of PLS, the objective is to predict the response variables as accurately as possible. Another application of PLS can be in parameter estimation where the objective is to estimate the parameters from the data in such a way that they are 'close' to the 'true' parameters. It is known that PLS gives biased estimate of parameters when the number of latent variables retained in the model is less than the number of input variables. However, it is shown that when a subspace of dimension A ($< K$, number of input variables) is correlated with the output variable and a PLS 1 model is built using latent variables then PLS1 gives unbiased estimate of the parameters.

Another important generalization of conventional PLS is to make it suitable for identifying a dynamic model of a system. One approach to incorporate dynamics into the PLS framework has been to change the static inner relationship of conventional PLS to a dynamic relationship (Lakshminarayan et al., 1997). In this approach, conventional PLS is first performed between

the input and output data matrices and a dynamic relationship is then fitted between each pair of corresponding t- and u-scores. The limitation of this methodology, however, is that the outer weights are not determined by the dynamics of the system and, therefore the constructed dynamic model may not be optimal in terms of its predictive capability. In the thesis, a scheme is proposed to optimally determine all the parameters (outer weights and inner scores model parameters) as per the dynamics of the system.

1.1.2 Abnormal Change Detection

With the increasing complexity of modern technological processes and the need for high quality and consistent product coupled with additional requirements of safety, ecological and economic concerns, reducing plant breakdowns, it is of paramount importance that a system (process) is monitored continuously. The technological challenge is to detect abnormal changes in the process as quickly as possible to ensure zero-defect products. This is all the more important in processes that are subject to fluctuating operating conditions. The problem of abnormal change detection, also known as fault detection or process monitoring, is closely related to quality control which is concerned with ensuring the quality of the final product.

The first step in developing a monitoring scheme for a system is to develop a mathematical model of the system when it is operating under normal operating conditions. The system is then monitored by determining the 'distance' between new observations measured from the system and the system model. If the 'distance' is below a threshold value, the system is declared to be operating under normal conditions, otherwise some abnormal event has occurred in the system. The problem of abnormal change detection can be placed in the following framework. Let $\mathbf{y}(t), \mathbf{x}(t)\big|_{1\le t\le N}$ be a sequence of observed random vectors from a system with conditional density function $p_{\theta}\left(\mathbf{y}(t), \mathbf{x}(t) \middle| \mathbf{y}(t-1), \mathbf{x}(t-1), ..., \mathbf{y}(1), \mathbf{x}(1)\right)$. Before the occurrence of an abnormal change, the conditional density parameter θ is constant and is equal to θ_0. After the change has occurred, the parameter changes to θ_1 ($\neq \theta_0$). The problem of abnormal change detection is to detect the occurrence of the abnormal change as soon as possible with the smallest possible false alarm rate.

A particular case of the above problem arises when it is assumed that the variables are multivariate Gaussian. Since the multivariate Gaussian distribution is completely characterized by the mean vector and the variance-covariance matrix, the abnormal changes in a system can be divided into two categories. The first category is related to the case where

the mean vector moves away from its normal value with the second being associated with a change in the variance-covariance structure of the process variables. While the first category has attracted a lot of attention from researchers and there exist optimal methods (in the sense of minimizing the delay for a given false alarm rate) e.g. Cumulative Sum (CUSUM) and Generalized Likelihood Ratio (GLR) test, very little work has been done to specifically address the second category of changes. The aim of this thesis is restricted to the second type of abnormality and a methodology is proposed to detect this change in an optimal way.

1.2 Contributions of the Thesis

The main aim of this thesis is the modelling and abnormal change detection in multivariate systems using subspace projection techniques. In particular, this thesis proposes extensions to the conventional PLS methodology so as to make it suitable for the modelling of, non-linear and dynamic systems. On the abnormal change detection front, a scheme is proposed to detect the change in the variance-covariance structure of a multivariate system in PCA and PLS based performance monitoring schemes. More specifically the contributions of the thesis are:

1. In most applications of PLS, its performance is evaluated based on its predictive capability. In this thesis, the performance of PLS as a parameter estimator is considered and evaluated.

2. Several non-linear PLS algorithms have been proposed in the literature. It is therefore, important to analyze the existing algorithms to identify which are 'true non-linear PLS' algorithms. In particular, one of the algorithms proposed by Baffi et al., (1999(a)) is analyzed. The reason for selecting this algorithm is that it is well known for its better predictive capability than other non-linear PLS algorithms. It is shown that this algorithm is a non-linear extension of Reduced Rank Regression (RRR), a classical regression technique, and therefore should not be considered as a 'true' non-linear extension of PLS.

3. It is argued that a 'true' non-linear PLS algorithm should be based on the 'non-linear covariance criterion'. After careful analysis of the algorithm by Wold et al., (1989) it is proven that this algorithm attempts to maximize the 'non-linear covariance' function.

4. The limitations of the algorithm of Wold et al., (1989) in the optimization of the non-linear covariance function are identified and two new non-linear PLS algorithms are proposed to overcome the limitations.

5. All the existing non-linear PLS algorithms are classified into three categories namely "quick and dirty", covariance based and error based depending on the objective functions used by the algorithms to determine the model parameters

6. One approach to taking into consideration the dynamics of the data in PLS is through the algorithm proposed by Lakshminarayan et al., (1997). The algorithm is divided into two steps. In the first step, conventional PLS is applied to the input and output data without augmenting the input matrix with lagged values of the input and/or output variables and in the second step, a dynamic model is fitted between each set of input and output scores. One limitation of the algorithm is that the outer weights (parameters in the first step) are not determined as per the dynamics of the data and therefore, the algorithm can be inefficient in situations where the dynamics are fast. An algorithm is proposed to overcome this limitation. In the proposed algorithm, all the parameters (outer weights as well as the parameters of the inner score model) are determined as dictated by the dynamics of the data

7. PCA based monitoring is based on the integration of two statistics, namely Hotelling T^2 and the Q-statistic. The poor sensitivity of this scheme to detect abnormal changes in the variance-covariance structure of the process is well known (Kano et al., 2001). An intuitive explanation of the poor sensitivity of Hotelling T^2 and the Q-statistic and the limitations of the scheme proposed by Kano et al., (2001) to detect changes in variance-covariance are given. A new scheme, that is especially suitable for detecting small changes in the covariance structure of a multivariate process, is then proposed. The proposed scheme has the advantage that it is 'nearly optimal' and can be analytically designed to detect changes.

8. A new monitoring scheme for detecting changes in the cross-covariance structure (between input and output variables) in a PLS based performance monitoring scheme is proposed. The monitoring scheme requires that a recursive PLS algorithm exists for identifying the parameters of the PLS model. A new recursive PLS algorithm that

converges to the parameters identified by NIPALS algorithm is derived. The monitoring statistic derived from the algorithm is 'nearly optimal' in its performance.

1. 3 Outline of the Thesis

Chapters 2 to 4 form the part of the thesis which considers the modelling aspects of multivariate systems with Chapters 6 and 7 concerned with abnormal change detection in multivariate systems.

Chapter 2 is a review chapter and starts with describing the theory of PCA. The properties of PCA and its application in regression, Principal Component Regression (PCR), are then reviewed. Limitations of PCR and ordinary least squares (OLS) are identified and these provide the motivation for the use of PLS. The PLS algorithm is then explained in detail and its properties proven. Modifications of Wold's NIPALS algorithm, namely the kernel algorithms, are then reviewed. A comparison of the predictive abilities of PCR and PLS is undertaken. Finally within the chapter, the performance of PLS as a parameter estimator is studied empirically.

Non-linear extension of PLS form the basis of Chapter 3. The chapter starts with an extensive literature survey of non-linear PLS. The algorithm proposed by Baffi et al., (1999(a)) is analyzed and it is shown that this algorithm is a non-linear extension of reduced rank regression. The algorithm of Wold et al., (1989) is then analyzed and it is proven that this algorithm attempts to maximize the non-linear covariance function between the scores. The limitations of this algorithm in terms of not determining all the parameters that influence the 'non-linear covariance' function so as to maximize the covariance function are highlighted. Two new non-linear PLS algorithms, NLPLS1 and NLPLS2, that address these limitations are then proposed. The performance of the new algorithms is evaluated and compared on two artificial data sets and a data set generated from a pH neutralization process with linear PLS and the non-linear PLS algorithm of Wold et al., (1989).

Chapter 4 is concerned with the extension of conventional PLS to model multivariate dynamic data. The chapter introduces the limitations of conventional PLS for identifying a dynamic model of the system. A comprehensive review of the approaches to incorporate dynamics in the PLS algorithm is carried out and a new method is derived. Finally a comparative study between the proposed method and the existing method is undertaken through simulations on both artificial data and chemical process data.

Chapter 5 is a review chapter on process performance monitoring. A brief overview of univariate monitoring schemes is first presented and the limitations of univariate monitoring schemes for a multivariate process are stated. Following an overview of multivariate statistical process control (MSPC, a brief literature survey of MSPC methodologies is undertaken.

Chapter 6 is specifically concerned with the detection of abnormal changes in the variance-covariance structure of multivariate Gaussian random vectors. The chapter first describes the limitations of existing PCA based monitoring schemes to detect these changes. A new monitoring scheme is then derived from the PCA model identification procedure to detect these changes in a 'near optimal' way by making use of the classical local approach to hypothesis testing. A brief introduction to the local approach of hypothesis testing is then given. The proposed technique is then applied to detect changes in two artificial data sets before using it to detect fouling in a heat exchanger in a continuous stirred tank reactor (CSTR) system.

In Chapter 7, a recursive version of PLS is derived and tested on an artificial data set for convergence. A monitoring statistic from this recursive algorithm is then derived to detect changes in the cross-covariance structure of the input and output variables in a PLS based monitoring scheme. The monitoring scheme is then applied to detect a change in a parameter of an artificial system before using it to detect fouling in a heat exchanger in a CSTR system.

Finally Chapter 8 gives conclusions and suggestions for future work.

1.4 Conclusions

In this chapter the formulation of the problems and the issues to be addressed in the subsequent chapters of this thesis are given. In particular, the problems of multivariate system modelling and the detection of abnormal changes are reviewed. A brief outline of each of the chapters and the contributions made are also summarized.

CHAPTER 2

Review of Multivariate Statistical Modelling Techniques

2.1 Introduction

In the chemical and process industries, a large number of variables are measured frequently resulting in large databases. The black box modelling of a process requires utilising this database to build a model of the process. An important feature of the process variables is that they are typically strongly correlated. One approach to handling this situation is through the application of statistical projection based techniques. This chapter reviews three multivariate subspace projection techniques that can be applied for the steady state modelling of a system: Principal Component Analysis (PCA), Principal Component Regression (PCR) and Partial Least Squares (PLS).

2.2 Principal Component Analysis

Principal component analysis is a classical statistical method that dates back to 1901 (Pearson, 1901). The method was further investigated by Hotelling (1933) who proposed an iterative least square method to implement the PCA algorithm. Since then many texts have been written on PCA (Jolliffe, 1986; Jackson, 1991) and it is included as a topic in most text books on multivariate statistics (Mardia et al., 1979). On the applications front, PCA was first applied in the social and behavioural sciences with subsequent applications being in industry in the area of quality control (Jackson, 1956, 1959; Jackson and Morris, 1957). In the last three decades PCA has been widely applied in the chemical and process industries for both the modelling and monitoring of, continuous processes (Kresta et al., 1991; Martin et al., 1996), batch processes (Nomikos and MacGregor, 1994; 1995), data compression and rectification (Kramer and Mah, 1994) and the detection of faulty sensors (Dunia et al., 1996).

Principal component analysis is known by alternative names in different disciplines, for example, in image processing (Jain, 1989) it is referred to as the Karhunen-Loeve transform or Hotelling transform and in the signal processing community, it is more commonly termed as the signal subspace or eigenstructure approach (Therrien, 1992). The aim of the subsequent section is to review the mathematical details of PCA.

2.2.1 Theory of Principal Component Analysis

Let $\mathbf{x} \in R^K$ be a K-dimensional random vector with (population) mean vector $\boldsymbol{\mu}$ and (population) covariance matrix $\boldsymbol{\Sigma}$. Without loss of generality, the mean vector $\boldsymbol{\mu}$ can be assumed to be zero, i.e. $E\{\mathbf{x}\} = \mathbf{0}$, where $E\{.\}$ denotes the statistical expectation operator. PCA seeks to find a vector $\mathbf{p}_1 \in R^K$ such that the projection of $\mathbf{x}$ on $\mathbf{p}_1$

$$t_1 = \mathbf{x}^T \mathbf{p}_1 = \mathbf{p}_1{}^T \mathbf{x} \tag{2.1}$$

has maximum variance. The variance of the projection t_1 is given by:

$$var(t_1) = E\{t_1^2\} - (E\{t_1\})^2 = \mathbf{p}_1^T E\{\mathbf{x}\mathbf{x}^T\}\mathbf{p}_1 = \mathbf{p}_1^T \boldsymbol{\Sigma} \mathbf{p}_1 \tag{2.2}$$

where $E\{t_1\} = 0$ from equation (2.1). Since the variance can become unbounded with an increase in the magnitude of the vector $\mathbf{p}_1$, it is necessary to constrain the magnitude of vector $\mathbf{p}_1$. The mathematical problem of PCA, therefore, can be formulated as a constrained optimization problem:

$$\max_{\mathbf{p}_1} E\{t_1{}^2\} = \max_{\mathbf{p}_1} E\{\mathbf{p}_1{}^T \mathbf{x}\mathbf{x}^T \mathbf{p}_1\} \qquad \|\mathbf{p}_1\| = 1 \tag{2.3}$$

The constrained optimization problem can be solved using the Lagrange multiplier for which the Lagrangian is:

$$J = \mathbf{p}_1^T \boldsymbol{\Sigma} \mathbf{p}_1 + \lambda(1 - \mathbf{p}_1^T \mathbf{p}_1) \tag{2.4}$$

where λ is a Lagrange multiplier. Taking the derivative of J with respect to $\mathbf{p}_1$ and equating the result to zero gives:

$$\boldsymbol{\Sigma} \mathbf{p}_1 = \lambda \mathbf{p}_1 \tag{2.5}$$

Equation (2.5) shows that $\mathbf{p}_1$ is a normalized eigenvector of $\boldsymbol{\Sigma}$ corresponding to the eigenvalue λ. Pre-multiplying equation (2.5) by $\mathbf{p}_1^T$ gives:

$$\mathbf{p}_1^T \boldsymbol{\Sigma} \mathbf{p}_1 = \lambda \mathbf{p}_1^T \mathbf{p}_1 = \lambda \tag{2.6}$$

From equation (2.2), it can be seen that the left hand side of equation (2.6) represents the variance of t_1. Thus the variance will be a maximum if the eigenvalue λ is a maximum. The solution $\mathbf{p}_1$ of the optimization problem is therefore, the normalized eigenvector of the covariance matrix $\boldsymbol{\Sigma}$ corresponding to the largest eigenvalue. The vector $\mathbf{p}_1$ is known as the (first) loading vector and the projection t_1 is the (first) principal component or latent variable. The above solution can be interpreted as a set of K variables contained in a vector $\mathbf{x}$ projected onto a single principal component t_1 that includes maximum information with respect to the variance. In most situations one principal component t_1 may not be sufficient to explain most of the information contained in the vector $\mathbf{x}$. Therefore, there is a need to extract more latent variables. To extract the second principal component t_2, it is required that t_2 and t_1 are orthogonal (uncorrelated). The idea behind the orthogonality constraint is that the information contained in principal components t_2 and t_1 should be mutually exclusive. Therefore extraction of the second principal component requires determining the loading vector $\mathbf{p}_2$ with unit norm such that the projection $t_2 = \mathbf{x}^T \mathbf{p}_2$ has maximum variance with the constraint that t_2 and t_1 are orthogonal. It can be shown (Anderson, 1984) that the loading vector $\mathbf{p}_2$ is the normalized eigenvector associated with the second largest eigenvalue of the covariance matrix $\boldsymbol{\Sigma}$. In general, the loading vector $\mathbf{p}_i$ corresponding to the i^{th} principal component t_i, where t_i is orthogonal to all other principal components, is given by the normalized eigenvector of the covariance matrix corresponding to the i^{th} largest eigenvalue. If A $(A \leq K)$ principal components are required to retain a 'sufficient' proportion of the information contained in the measurements of variables vector $\mathbf{x}$ then the subspace spanned by the loading vectors $\mathbf{p}_1, \mathbf{p}_2 ..., \mathbf{p}_A$ is known as the signal subspace of dimension A and the subspace spanned by the loading vectors $\mathbf{p}_{A+1} ... \mathbf{p}_K$ is the noise subspace.

The computational methods for computing the PCA solutions (eigenvectors and eigenvalues of the covariance matrix $\mathbf{\Sigma}$) can be divided into two categories. The first consists of batch methods, where the eigenvectors and eigenvalues of a matrix are computed through a single matrix operation. The most important batch method is the application of Singular Value Decomposition (SVD) (Golub and Loan, 1996) to the matrix $\mathbf{X}$ that contains N observations of the variables vector $\mathbf{x}$. The second category includes methods that compute the eigenvalues and eigenvectors of the covariance matrix in an iterative manner. The latter method is useful where not all eigenvalues and eigenvectors of the matrix are required. One of the popular iterative methods for computing the principal components is the iterative least square method proposed by Wold (1966(a)) that was later applied to partial least squares (Geladi and Kowalsky, 1986). Another iterative method is the Power method (Golub and Loan, 1996).

2.2.2 Properties of Principal Component Analysis

The key properties of PCA include:

1. The variance of principal component t_i is λ_i, i.e. the i^{th} largest eigenvalue of the covariance matrix:

$$var(t_i) = E\{\mathbf{p}_i^T \mathbf{x}\,\mathbf{x}^T \mathbf{p}_i\} = \mathbf{p}_i^T E\{\mathbf{x}\mathbf{x}^T\}\mathbf{p}_i = \mathbf{p}_i^T \mathbf{\Sigma}\,\mathbf{p}_i \qquad (2.7)$$

Since $\mathbf{p}_i$ is an eigenvector of the matrix $\mathbf{\Sigma}$ corresponding to the eigenvalue λ_i,

$$\mathbf{\Sigma}\,\mathbf{p}_i = \lambda_i \mathbf{p}_i \qquad (2.8)$$

Substituting equation (2.8) back into equation (2.7) and noting that $\mathbf{p}_i^T \mathbf{p}_i = 1$:

$$var(t_i) = \mathbf{p}_i^T \lambda_i \mathbf{p}_i = \lambda_i \mathbf{p}_i^T \mathbf{p}_i = \lambda_i \qquad (2.9)$$

2. Any two principal components are orthogonal (uncorrelated):

$$\begin{aligned} E\{t_i t_j\} &= E\{\mathbf{p}_i^T \mathbf{x}\,\mathbf{x}^T \mathbf{p}_j\} \\ &= \mathbf{p}_i^T \mathbf{\Sigma}\,\mathbf{p}_j = \mathbf{p}_i^T \lambda_j \mathbf{p}_j = \lambda_j \mathbf{p}_i^T \mathbf{p}_j = 0 \quad \text{for } i \neq j \end{aligned} \qquad (2.10)$$

3. $\frac{\lambda_1 + \lambda_2 + \ldots . \lambda_A}{\lambda_1 + \lambda_2 + \ldots . \lambda_K}$, defines the percentage of total variance explained by the first A principal components.
4. No linear combination of the vector $\mathbf{x}$ has a larger variance than λ_1. This is a result of the objective function, given in equation (2.3) being maximized when defining the principal components.
5. The principal components are not scale-invariant.
6. If the covariance matrix $\boldsymbol{\Sigma}$ has rank $R < K$, then the total variance can be explained by first R principal components.
7. PCA also minimizes the mean square error $\mathrm{E}\left\{\left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2\right\}$, where $\hat{\mathbf{x}}$ is the lower dimensional subspace approximation of $\mathbf{x}$.

2.2.3 Sample Principal Component Analysis

In the previous sub-section, it was assumed that the population covariance matrix $\boldsymbol{\Sigma}$ is available for computing the eigenvalues and eigenvectors. In most practical situations, the population covariance matrix is unknown and is estimated from N observations of a random vector $\mathbf{x}$ collected into a matrix $\mathbf{X}$. The unbiased estimate of the (sample) covariance matrix is computed as:

$$\mathbf{S} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N-1} \tag{2.11}$$

where it is assumed that the matrix $\mathbf{X}$ is mean centred. In practice, the loading vectors $\mathbf{p}_1, \mathbf{p}_2 \ldots, \mathbf{p}_K$ are computed as the normalized eigenvectors of the sample covariance matrix $\mathbf{S}$. It is, therefore, necessary to understand how the parameters of the sample PCA (eigenvectors and eigenvalues of the sample covariance matrix $\mathbf{S}$) relate to the parameters of the population PCA. The relationships are as follows:

1. If $\mathbf{x}$ is a (multivariate) Gaussian random vector with (population) covariance matrix $\boldsymbol{\Sigma}$ with distinct eigenvalues, then the eigenvectors and eigenvalues of the

sample covariance matrix are the maximum likelihood estimates of the corresponding population parameters (Anderson, 1984).

2. It can be proved (Mardia et al., 1979; Anderson, 1984) that the sample eigenvalues and eigenvectors are asymptotically normally distributed.

2.3 Principal Component Regression

In many applications, building a mathematical model of the system requires establishing a causal relationship between the measurements on the input variables **X**, also known as independent variables or process variables, and the output variables **y**, also known as the dependent or quality variable. Assuming a linear relationship exists between **y** and **X**, that is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{2.12}$$

where $\boldsymbol{\beta}$ is a vector of regression coefficients and **e** is the prediction error. Ordinary Least Squares (OLS) can be applied to find the estimate of $\boldsymbol{\beta}$ (Draper and Smith, 1998):

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2.13}$$

Properties of OLS include the fact that it is known to give the Best Linear Unbiased Estimate (BLUE) of the parameters (regression vector $\boldsymbol{\beta}$) when the Gauss-Markov assumptions (Montgomery and Peck, 1982) are satisfied. That is, of all possible linear unbiased estimates of the regression coefficients, the estimates given by OLS have the smallest variance.

One limitation of OLS is where the input variables are strongly correlated. This problem is often referred to as multicollinearity, and if OLS is used to construct a model in such a situation then the parameter estimates will be unstable. By instability it is meant that the parameters will be highly sensitive to small changes in the data, for example, the presence of an outlier. Also the standard error (deviation) of the parameter estimates will be high making them unreliable.

Several methods e.g. stepwise regression, ridge regression and variable selection techniques (Montgomery and Peck, 1982) have been proposed in the literature to overcome this problem. However, the techniques that have received significant attention to solve the problem of

multicollinearity in the regression modelling are known as subspace projection methods. The basic idea behind these techniques is to project the original correlated variables onto orthogonal latent variables such that the loss of 'information' is minimized. In this section one such technique, Principal Component Regression (PCR) is introduced.

Principal component regression involves first performing PCA on the predictor variables matrix **X** and then using the principal components in place of the predictor variables in the regression analysis. Since the principal components are mutually orthogonal, the issue of multicollinearity is addressed. It can be proven that if all the principal components are retained when building the regression model, the solution is equivalent to the OLS solution and thus the problem of the large variance of the OLS estimates in the presence of multicollinearity is not addressed. In practice only a few principal components are included in the regression model which leads to a reduction in the variance of the estimates but the cost of reducing the variance of the estimates is that of biased parameter estimates. The mathematical theory behind PCR is now discussed. The values of the principal components (referred to as t-scores) for each observation of the input variables are given by:

$$\mathbf{T} = \mathbf{XP} \tag{2.14}$$

where $\mathbf{P}$ is a $(K \times K)$ loading matrix and $\mathbf{T}$ is a score matrix of order $(N \times K)$. Since $\mathbf{P}$ is an orthogonal matrix, $\mathbf{X\beta}$ can be written as:

$$\mathbf{X\beta} = \mathbf{XPP}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{T}\boldsymbol{\gamma} \tag{2.15}$$

where

$$\boldsymbol{\gamma} = \mathbf{P}^{\mathrm{T}}\boldsymbol{\beta} \tag{2.16}$$

Substitution of equation (2.15) into equation (2.12) gives:

$$\mathbf{y} = \mathbf{T}\boldsymbol{\gamma} + \mathbf{e} \tag{2.17}$$

The least square estimate of the new regression vector $\boldsymbol{\gamma}$ is given by:

$$\hat{\boldsymbol{\gamma}} = (\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{y} \tag{2.18}$$

Since the matrix $\mathbf{T}$ is an orthogonal matrix, $(\mathbf{T}^{\mathrm{T}}\mathbf{T})$ is a diagonal matrix. The estimate of the regression vector $\boldsymbol{\beta}$ from equation (2.16) is given by:

$$\hat{\boldsymbol{\beta}}_{\mathrm{PCR}} = \mathbf{P}\,\hat{\boldsymbol{\gamma}} \tag{2.19}$$

It can be proven that the solution $\hat{\boldsymbol{\beta}}_{\mathrm{PCR}}$ is equal to $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$. Substituting equation (2.18) into (2.19) and using equation (2.14) gives:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\mathrm{PCR}} &= \mathbf{P}(\mathbf{P}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{P}\mathbf{P}^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}(\mathbf{P}^{\mathrm{T}})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} \\ &= (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \hat{\boldsymbol{\beta}}_{\mathrm{OLS}}\end{aligned} \tag{2.20}$$

It can be seen from equation (2.20) that if all the principal components are retained in the model, there is no advantage to using PCR except that the computation is simplified:

$$\hat{\boldsymbol{\beta}}_{\mathrm{PCR}} = \mathbf{P}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{y} \tag{2.21}$$

Since $(\mathbf{T}^{\mathrm{T}}\mathbf{T})$ is a diagonal matrix, and if d_i denotes the i^{th} diagonal element of this diagonal matrix and $\mathbf{p}_i$ denotes the i^{th} column of $\mathbf{P}$, then equation (2.21) can be written as:

$$\hat{\boldsymbol{\beta}}_{\mathrm{PCR}} = \sum_{i=1}^{K} d_i^{-1}\mathbf{p}_i\mathbf{p}_i^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{2.22}$$

Assuming that the observations of the output variable are uncorrelated and each has the same variance, σ^2, then the variance-covariance matrix of the estimate $\hat{\boldsymbol{\beta}}_{\mathrm{PCR}}$ is given by:

$$\begin{aligned}Cov(\hat{\boldsymbol{\beta}}_{\mathrm{PCR}}) &= \sigma^2\mathbf{P}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}(\mathbf{T}^{\mathrm{T}}\mathbf{T})(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{P}^{\mathrm{T}} = \sigma^2\mathbf{P}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{P}^{\mathrm{T}} \\ &= \sigma^2\sum_{i=1}^{K} d_i^{-1}\mathbf{p}_i\mathbf{p}_i^{\mathrm{T}}\end{aligned} \tag{2.23}$$

where $Cov(.)$ denotes the covariance function. From equation (2.23) it can be seen that multicollinearity leads to large variances for the elements of $\hat{\boldsymbol{\beta}}_{PCR}$. Since the variance of the i^{th} principal component is proportional to d_i, multicollinearity results in one or more of the d_i's in equation (2.23) being very small resulting in large variances for the elements of the estimated parameter vector $\hat{\boldsymbol{\beta}}_{PCR}$. One approach to reducing the variance of the elements of $\hat{\boldsymbol{\beta}}_{PCR}$ is to delete terms in equation (2.22) that correspond to very small values of d_i. If $A < K$ terms are retained in equation (2.23), the estimator becomes:

$$\tilde{\boldsymbol{\beta}}_{PCR} = \sum_{i=1}^{A} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{y} \qquad (2.24)$$

where it is assumed that $d_{A+1}, d_{A+2}, \ldots, d_K$ are very small. It can be shown (Jolliffe, 1986) that the covariance matrix of $\tilde{\boldsymbol{\beta}}_{PCR}$ is given by:

$$Cov(\hat{\boldsymbol{\beta}}_{PCR}) = \sigma^2 \sum_{i=1}^{A} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \qquad (2.25)$$

Comparison of equations (2.25) and (2.23) show that the PCR model with fewer principal components being retained lead to smaller variances for the estimated parameters. But this reduction in variance comes at the price of introducing bias into the estimates. From equations (2.22) and (2.24), the model parameter $\hat{\boldsymbol{\beta}}_{PCR}$, which is equal to the OLS solution $\hat{\boldsymbol{\beta}}_{OLS}$, and the reduced model parameter $\tilde{\boldsymbol{\beta}}_{PCR}$ can be related as:

$$\tilde{\boldsymbol{\beta}}_{PCR} = \tilde{\boldsymbol{\beta}}_{OLS} - \sum_{i=A+1}^{K} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{y} \qquad (2.26)$$

The statistical expectation of the second term on the right hand side of equation (2.26) is given as:

$$E\left\{ \sum_{i=A+1}^{K} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{y} \right\} = \sum_{i=A+1}^{K} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T E\{\mathbf{y}\} = \sum_{i=A+1}^{K} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{X}\boldsymbol{\beta} \qquad (2.27)$$

$\mathbf{X}^T\mathbf{X}$ can be decomposed using singular value decomposition:

$$\mathbf{X}^T\mathbf{X} = \sum_{i=1}^{K} d_i \mathbf{p}_i \mathbf{p}_i^T \tag{2.28}$$

Substituting (2.28) into (2.27) and noting that the vectors $\mathbf{p}_i$ are orthonormal:

$$E\left\{ \sum_{i=A+1}^{K} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{y} \right\} = \sum_{i=A+1}^{K} \mathbf{p}_i \mathbf{p}_i^T \boldsymbol{\beta} \tag{2.29}$$

Taking the statistical expectation of both sides of equation (2.26) and using equation (2.29) gives:

$$E\{\tilde{\boldsymbol{\beta}}_{PCR}\} = \boldsymbol{\beta} - \sum_{i=A+1}^{K} \mathbf{p}_i \mathbf{p}_i^T \boldsymbol{\beta} \tag{2.30}$$

where $E\{\hat{\boldsymbol{\beta}}_{OLS}\} = \boldsymbol{\beta}$. Since the second term on the right hand side of equation (2.30) is typically not zero, the estimate will be biased.

It is also important to note that it is not always a good strategy to retain the first A principal components and delete the remaining $(K\text{-}A)$ principal components which have small variances. A principal component that has a small variance can be highly correlated with the output variable and therefore it would be desirable to include this principal component in the regression model. Taking this into consideration, a PCR model can be defined as in equation (2.12), where the estimate of the regression coefficient $\boldsymbol{\beta}$ is computed from:

$$\tilde{\boldsymbol{\beta}}_{PCR} = \sum_{z} d_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \mathbf{X}^T \mathbf{y} \tag{2.31}$$

where Z is an appropriate subset of the principal components.

2.4 Partial Least Squares

A limitation of PCR is that the direction in which the input variables are projected is determined so as to explain the maximum variance of **X**. The objective of a regression model, however, is to explain "maximally" the output variables **Y**. The directions which explain the maximal variance of **X** need not necessarily be those which explain the maximal variance of **Y**. This limitation is overcome by the partial least squares algorithm that was developed in the 1960's.

2.4.1 Literature Review and Historical Details of PLS

The history of PLS as a modelling tool began in the 1960's when Herman Wold (1966 (a); 1966(b)) proposed an iterative algorithm for extracting latent variables both for PCA and the two-block situation. This algorithm was first known as NILES (Non-Linear Estimation by Least Squares) and was later termed NIPALS (Non-Linear Iterative Partial Least Squares) (Geladi, 1988). The initial applications of PLS were in econometrics (Fornell and Bookstein, 1982 ; Dijkstra, 1983) with the range of applications broadening out to include the disciplines of psychology, management, education, political science, environmental science and analytical chemistry (Geladi, 1988; Sellin, 1995; Hulland, 1999).

A key pioneer of the application of PLS in chemometrics was Svante Wold. Some of the earlier publications in chemometrics involving the application of PLS include (Wold et al., 1983 (a); 1983(b); Wold et al., 1984; Lorber et al., 1987; Frank, 1987). The reasons for the popularity of PLS in chemometrics are a consequence of the fact that in the chemical and process industries, a large number of variables are measured that are highly correlated thereby giving rise to the multicollinearity problem. PLS not only effectively handles multicollinearity but it can also describe the variation of the predictor and response variables using a reduced set of variables. The second reason is that PLS can identify the causal relationship between the predictor and response variables even when the number of observations is less than the number of variables. This situation is common in spectroscopic data where the number of wavelengths can significantly exceed the number of samples.

In the late 1980's and 1990's a number of researchers Höskuldsson (1988) and Kaspar and Ray (1993 (b)) addressed some of the theoretical challenges of PLS including the definition of the properties of PLS. Additionally, in this period, a number of modifications to PLS to

identify non-linear models (Wold et al., 1989; Wold, 1992; Frank, 1990) were proposed. Furthermore recursive versions of PLS were proposed (Helland et al., 1992; Qin, 1993; 1998; Dayal and MacGregor, 1997 (b)), where the PLS model was updated on-line to help realize the modelling of nonstationary data. Dynamic versions of PLS have also been proposed (Kaspar and Ray, 1992; 1993(a); Lakshminarayan et al., 1997) to take into consideration the dynamics of the process. One important application of PLS based dynamic models has been in process control (Lakshminarayan et al., 1997; Patwardhan et al., 1998).

2.4.2 Partial Least Squares -The Algorithms

Let $\mathbf{X}$ be a $(N \times K)$ matrix containing N observations on K predictor variables and let $\mathbf{Y}$ be a $(N \times M)$ matrix comprising N observations on M response variables. PLS seeks to find two vectors $\mathbf{w}_1 \in \mathrm{R}^K$ in the row space of $\mathbf{X}$ and $\mathbf{v}_1 \in \mathrm{R}^M$ in the row space of $\mathbf{Y}$ such that the vectors $\mathbf{t}_1$ and $\mathbf{u}_1$ in the column space of $\mathbf{X}$ and $\mathbf{Y}$ respectively, given by

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \qquad \mathbf{u}_1 = \mathbf{Y}\mathbf{v}_1 \tag{2.32}$$

have maximum covariance. The vectors $\mathbf{t}_1$ and $\mathbf{u}_1$ in R^N are known as t-scores and u-scores respectively. The estimate of the covariance between $\mathbf{t}_1$ and $\mathbf{u}_1$ is given by:

$$Cov(\mathbf{t}_1, \mathbf{u}_1) = \mathbf{t}_1^{\mathrm{T}}\mathbf{u}_1 \tag{2.33}$$

It should be noted from equations (2.32) and (2.33) that if there is no constraint on the magnitude of $\mathbf{w}_1$ and $\mathbf{v}_1$, then the magnitude of the covariance can be made arbitrarily large by choosing suitable $\mathbf{w}_1$ and $\mathbf{v}_1$. To keep the magnitude of covariance bounded, the constraint of unit norm is placed on $\mathbf{w}_1$ and $\mathbf{v}_1$. Mathematically, the problem of PLS can be stated as:

$$\mathrm{J} = \underset{\mathbf{w}_1, \mathbf{v}_1}{max} \ (\mathbf{t}_1{}^{\mathrm{T}}\mathbf{u}_1) \quad \text{subject to } \|\mathbf{w}_1\| = \|\mathbf{v}_1\| = 1 \tag{2.34}$$

The above constrained optimization problem can be solved using the Lagrangian multiplier method with the Lagrangian being given by:

$$L(\mathbf{w}_1, \mathbf{v}_1, \mu, \lambda) = (\mathbf{t}_1{}^T \mathbf{u}_1) + \mu(1 - \mathbf{w}_1{}^T \mathbf{w}_1) + \sigma(1 - \mathbf{v}_1{}^T \mathbf{v}_1)$$
$$= \mathbf{w}_1^T \mathbf{X}^T \mathbf{Y} \mathbf{v}_1 + \mu(1 - \mathbf{w}_1{}^T \mathbf{w}_1) + \sigma(1 - \mathbf{v}_1{}^T \mathbf{v}_1) \quad (2.35)$$

where σ and μ are the Lagrangian multipliers. The optimal solution is found by setting derivatives of the Lagrangian with respect to parameters, $\mathbf{w}_1$ and $\mathbf{v}_1$ and the Lagrangian multipliers, σ and μ, to zero:

$$\frac{\partial}{\partial \mathbf{w}_1} L(\mathbf{w}_1, \mathbf{v}_1, \mu, \sigma) = (\mathbf{X}^T \mathbf{Y} \mathbf{v}_1) - 2\mu \mathbf{w}_1 = 0 \quad (2.36)$$

$$\frac{\partial}{\partial \mathbf{v}_1} L(\mathbf{w}_1, \mathbf{v}_1, \mu, \sigma) = (\mathbf{Y}^T \mathbf{X} \mathbf{w}_1) - 2\sigma \mathbf{v}_1 = 0 \quad (2.37)$$

$$\frac{\partial}{\partial \mu} L(\mathbf{w}_1, \mathbf{v}_1, \mu, \sigma) = \mathbf{w}_1^T \mathbf{w}_1 - 1 = 0$$
$$\frac{\partial}{\partial \sigma} L(\mathbf{w}_1, \mathbf{v}_1, \mu, \sigma) = \mathbf{v}_1^T \mathbf{v}_1 - 1 = 0 \quad (2.38)$$

From equation (2.37):

$$\mathbf{v}_1 = \frac{1}{2\sigma} (\mathbf{Y}^T \mathbf{X} \mathbf{w}_1) \quad (2.39)$$

Substituting equation (2.39) into (2.36) gives

$$\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w}_1 = 4\sigma\mu \mathbf{w}_1 = \lambda_1 \mathbf{w}_1 \quad (2.40)$$

From equation (2.40), it can be concluded that the weight vector $\mathbf{w}_1$ is an eigenvector of the matrix $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$ with eigenvalue λ_1. Furthermore $\mathbf{w}_1$ is an eigenvector corresponding to the largest eigenvalue of $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$. This is because the covariance functions in equation (2.33) can be re-written by incorporating equation (2.39):

$$Cov(\mathbf{t}_1, \mathbf{u}_1) = \mathbf{t}_1^T \mathbf{u}_1 = \mathbf{w}_1^T \mathbf{X}^T \mathbf{Y} \mathbf{v}_1 = \frac{1}{2\sigma}\left(\mathbf{w}_1^T \mathbf{X}^T \mathbf{Y}\mathbf{Y}^T \mathbf{X} \mathbf{w}_1\right) \quad (2.41)$$

Pre-multiplying equation (2.40) by $\mathbf{w}_1^T$

$$\mathbf{w}_1^T \mathbf{X}^T \mathbf{Y}\mathbf{Y}^T \mathbf{X} \mathbf{w}_1 = \lambda_1 \mathbf{w}_1^T \mathbf{w}_1 = \lambda_1 \quad (2.42)$$

and then combining equations (2.41) and (2.42) gives:

$$Cov(\mathbf{t}_1, \mathbf{u}_1) = \frac{\lambda_1}{2\sigma} \quad (2.43)$$

Since the *Cov* (.) function is proportional to λ_1, the eigenvalue of $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ that gives maximum value of the covariance is the eigenvector corresponding to the maximum value of eigenvalue. It can similarly be proved that the weight vector $\mathbf{v}_1$ is an eigenvector of $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$ corresponding to the largest eigenvalue.

2.4.2.1 Computation of the Weight Vectors

One method to compute the weight vectors, $\mathbf{w}_1$ and $\mathbf{v}_1$, is to make use of the result proven in the previous section which states that the weight vectors can be computed by solving the eigenvalue-eigenvector problem. However, instead of separately computing the eigenvectors of the two matrices $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}$, the two weight vectors can be computed by applying Singular Value Decomposition (SVD) (Kaspar and Ray, 1993(b)) to the cross covariance matrix $\mathbf{X}^T\mathbf{Y}$ with the weight vector $\mathbf{w}_1$ being equal to the left singular vector and $\mathbf{v}_1$ being equal to the right singular vector associated with the largest singular value. Mathematically, SVD decomposes the matrix $\mathbf{X}^T\mathbf{Y}$ as:

$$\mathbf{X}^T\mathbf{Y} = \mathbf{W}\mathbf{D}\mathbf{V}^T \quad (2.44)$$

where **W** is a matrix of left orthonormal singular vectors, **V** is a matrix of right orthonormal vectors and **D** is a diagonal matrix of singular values. The key step to computing the weight vectors is:

$$\mathbf{w}_1 = \mathbf{W}(:,1) \qquad \mathbf{v}_1 = \mathbf{V}(:,1) \tag{2.45}$$

The weight vectors can also be computed using an iterative method which is at the heart of the NIPALS algorithm. The theory behind the iterative computation is now described.

To simplify the situation, it is first assumed that the eigenvalue problem equation (2.40) is solved, thus the weight vector $\mathbf{w}_1$ is known. Knowing $\mathbf{w}_1$, the t-scores vector can be calculated as:

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \tag{2.46}$$

And the weight vector $\mathbf{v}_1$ can be calculated using equation (2.39):

$$\mathbf{v}_1 = \frac{\mathbf{Y}^T\mathbf{X}\mathbf{w}_1}{2\sigma} = \frac{\mathbf{Y}^T\mathbf{t}_1}{2\sigma} \tag{2.47}$$

Since the weight vector $\mathbf{v}_1$ should be of unit norm as per the requirements of the objective function of PLS, the normalized weight vector is given by:

$$\mathbf{v}_1 = \frac{\mathbf{Y}^T\mathbf{t}_1}{\left\|\mathbf{Y}^T\mathbf{t}_1\right\|} \tag{2.48}$$

It should be noted from equation (2.48) that normalization of the weight vector $\mathbf{v}_1$ to unit norm eliminates the constant σ which appeared in the expression for $\mathbf{v}_1$ in equation (2.47). After $\mathbf{v}_1$ is determined, the u-score vector can be determined as:

$$\mathbf{u}_1 = \mathbf{Y}\mathbf{v}_1 \tag{2.49}$$

The iteration is completed by calculating $\mathbf{w}_1$ from the u-scores $\mathbf{u}_1$ by using equation (2.36):

$$\mathbf{w}_1 = \frac{\mathbf{X}^T\mathbf{Y}\mathbf{v}_1}{2\mu} = \frac{\mathbf{X}^T\mathbf{u}_1}{2\mu} \tag{2.50}$$

The constant μ is eliminated when the weight vector $\mathbf{w}_1$ is normalized to unit norm:

$$\mathbf{w}_1 = \frac{\mathbf{X}^T\mathbf{u}_1}{\left\|\mathbf{X}^T\mathbf{u}_1\right\|} \tag{2.51}$$

The cycle of computation can be thus summarized as:

$$\mathbf{w}_1 \rightarrow \mathbf{t}_1 \rightarrow \mathbf{v}_1 \rightarrow \mathbf{u}_1 \rightarrow \mathbf{w}_1 \tag{2.52}$$

The algorithm described started by defining $\mathbf{w}_1$ as the eigenvector of $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ and therefore, the algorithm converges in one iteration. If an arbitrary vector $\mathbf{w}_1 \in R^K$, is used as the starting point, the algorithm will take, in general, more than one iteration to converge. It should also be noted that it is not necessary to start with the value of $\mathbf{w}_1$ to reach the solution. In fact, it is possible to start from anywhere in the cycle given in equation (2.52). For example, an arbitrary vector $\mathbf{u}_1$ can be first selected and then $\mathbf{w}_1$ is computed using equation (2.51), followed by the computation of $\mathbf{t}_1$ and $\mathbf{v}_1$ using equations (2.46) and (2.48) respectively with the cycle (iteration) ending by computing a new value of u-scores $\mathbf{u}_1$, using equation (2.49). If the new value of $\mathbf{u}_1$ is sufficiently close to the initial value, the algorithm is terminated, otherwise the procedure is repeated. The complete iterative procedure is summarized below.

Given: Matrices $\mathbf{X}$ and $\mathbf{Y}$

1. Select an arbitrary u-scores vector $\mathbf{u}_1 \in R^N$. For example, any column of the matrix $\mathbf{Y}$

2. Compute $\mathbf{w}_1 = \mathbf{X}^T\mathbf{u}_1$.
3. Normalize $\mathbf{w}_1$ to unit length.
4. Compute $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$
5. Compute $\mathbf{v}_1 = \mathbf{Y}^T\mathbf{t}_1$
6. Normalize $\mathbf{v}_1$ to unit length
7. Compute $\mathbf{u}_1 = \mathbf{Y}\mathbf{v}_1$
8. If the distance between the $\mathbf{u}_1$ vectors computed in step 7 and step 1 is less than a predefined value, stop otherwise return to step 1 and repeat the procedure until convergence is attained

To build a predictive model between matrices **X** and **Y**, a linear relationship between the scores $\mathbf{t}_1$ and $\mathbf{u}_1$ is fitted using ordinary least squares regression:

$$\mathbf{u}_1 = b_1\mathbf{t}_1 + \mathbf{e}_1 \tag{2.53}$$

where b_1 is the regression coefficient:

$$b_1 = \frac{\mathbf{u}_1{}^T\mathbf{t}_1}{\mathbf{t}_1{}^T\mathbf{t}_1} \tag{2.54}$$

Equation (2.53) defines the so called inner relationship of the PLS model. Since it is only the original variables that have physical significance, it is important to establish the outer relationship (between the latent variables and the original input and output variables). To identify the outer relationship, it should be noted that the score $\mathbf{t}_1$ contains information about **X** (as the score $\mathbf{t}_1$ is a linear combination of the columns of **X**) and therefore can be used to predict matrix **X**. This can be achieved by selecting vector $\mathbf{p}_1 \in R^K$ such that:

$$\mathbf{X} = \mathbf{t}_1\,\mathbf{p}_1{}^T + \mathbf{E}_1 \tag{2.55}$$

The vector $\mathbf{p}_1$ is determined such that the norm of the prediction error $\mathbf{E}_1$ is a minimum. Applying least squares, the regression vector $\mathbf{p}_1$ is given by:

$$\mathbf{p}_1 = \frac{\mathbf{X}^T \mathbf{t}_1}{\mathbf{t}_1^T \mathbf{t}_1} \tag{2.56}$$

The scores vector $\mathbf{t}_1$ can also be used to predict matrix **Y**. This can be done by first using $\mathbf{t}_1$ to predict the u-scores $\hat{\mathbf{u}}_1 = b_1 \mathbf{t}_1$, and then using the predicted u-scores to predict matrix **Y** by finding $\mathbf{q}_1$ such that:

$$\mathbf{Y} = \hat{\mathbf{u}}_1 \mathbf{q}_1^T + \mathbf{F}_1 = b_1 \mathbf{t}_1 \mathbf{q}_1^T + \mathbf{F}_1 \tag{2.57}$$

the norm of $\mathbf{F}_1$ is a minimum. This can be determined using least squares:

$$\mathbf{q}_1 = \frac{\mathbf{Y}^T \hat{\mathbf{u}}_1}{\hat{\mathbf{u}}_1^T \hat{\mathbf{u}}_1} \tag{2.58}$$

Equations (2.55) and (2.57) collectively define the outer relationship of the PLS model. In the terminology of PLS, the vectors $\mathbf{p}_1$ and $\mathbf{q}_1$ are known as the loading vectors and determine the contributions of the scores vector $\mathbf{t}_1$ to the input and output matrices.

2.4.2.2 Motivation for the Deflation Procedure

In general, one latent variable is not sufficient to predict the matrix **Y** (and also **X**) and therefore more than one latent variable will be included in the PLS model. The philosophy behind extracting more than one latent variable is that the latent variables should contain 'independent' information about the input and output measurements. Therefore, to extract the second latent variable which contains information other than that included in the first set of latent variables, the contribution of the first latent variables towards the input and output matrices must be subtracted from matrices **X** and **Y**. This procedure is known as deflation. From equations (2.55) and (2.57), it can be observed that the contributions of the first latent variable $\mathbf{t}_1$ to matrices **X** and **Y** is $\mathbf{t}_1 \mathbf{p}_1^T$ and $b_1 \mathbf{t}_1 \mathbf{q}_1^T$ respectively. Therefore, the deflated matrices $\mathbf{X}_2$ and $\mathbf{Y}_2$ for extracting the second latent variables are given by:

$$\mathbf{X}_2 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T$$
$$\mathbf{Y}_2 = \mathbf{Y} - b_1\mathbf{t}_1\mathbf{q}_1^T \quad (2.59)$$

and the second latent variables are given by $\mathbf{t}_2 = \mathbf{X}_2\mathbf{w}_2$ and $\mathbf{u}_2 = \mathbf{Y}_2\mathbf{v}_2$, where the vectors $\mathbf{w}_2$ and $\mathbf{v}_2$ are the eigenvectors corresponding to the largest eigenvalues of the matrices $\mathbf{X}_2^T\mathbf{Y}_2\mathbf{Y}_2^T\mathbf{X}_2$ and $\mathbf{Y}_2^T\mathbf{X}_2\mathbf{X}_2^T\mathbf{Y}_2$ so as to maximize the covariance between the latent variables $\mathbf{t}_2$ and $\mathbf{u}_2$. The inner relationship between the scores is given by:

$$\mathbf{u}_2 = b_2\mathbf{t}_2 + \mathbf{e}_2 \quad (2.60)$$

where b_2 is the regression coefficient and is determined from equation (2.54) by replacing $\mathbf{t}_1$ and $\mathbf{u}_1$ with $\mathbf{t}_2$ and $\mathbf{u}_2$ respectively. The outer relationship is similarly denoted as:

$$\mathbf{X}_2 = \mathbf{t}_2\mathbf{p}_2^T + \mathbf{E}_2$$
$$\mathbf{Y}_2 = \hat{\mathbf{u}}_2\mathbf{q}_2^T + \mathbf{F}_2 \quad (2.61)$$

The loading vectors $\mathbf{p}_2$ and $\mathbf{q}_2$ for the second latent variable can be determined from equations (2.56) and (2.58) by replacing $\mathbf{X}$ and $\mathbf{Y}$ with $\mathbf{X}_2$ and $\mathbf{Y}_2$ and $\mathbf{t}_1$ and $\hat{\mathbf{u}}_1$ with $\mathbf{t}_2$ and $\hat{\mathbf{u}}_2$ respectively. The decomposition of matrices $\mathbf{X}$ and $\mathbf{Y}$, after the extraction of two latent variables, can be obtained by substituting (2.61) into equation (2.59):

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \mathbf{E}_2$$
$$\mathbf{Y} = \hat{\mathbf{u}}_1\mathbf{q}_1^T + \hat{\mathbf{u}}_2\mathbf{q}_2^T + \mathbf{F}_2 \quad (2.62)$$

In general, if A latent variables are required to build the PLS model, then the matrices $\mathbf{X}$ and $\mathbf{Y}$ can be written as:

$$\mathbf{X} = \sum_{i=1}^{A}\mathbf{t}_i\mathbf{p}_i^T + \mathbf{E}$$
$$\mathbf{Y} = \sum_{i=1}^{A}\hat{\mathbf{u}}_i\mathbf{q}_i^T + \mathbf{F} \quad (2.63)$$

From the above it can be noted that PLS decomposes matrices **X** and **Y** into the summation of A rank-one matrices. The matrices **E** and **F** are the residual matrices for matrices **X** and **Y** respectively when the PLS model is build using A latent variable. Each pair of latent variables account for a certain percentage of variance for both **X** and **Y** with most of the variability in **X** and **Y** being explained by $(A < K)$ latent variables. The remaining variability typically accounts for the noise in the data. The decision of how many latent variables should be retained in the PLS model can be made using cross-validation (Wold, 1978).

Geometrically the loading vectors $\mathbf{p}_i$ and $\mathbf{q}_i$ represent the basis vectors of the input and output space respectively. It is therefore desirable to normalize these vectors to unit length. This mathematical adjustment can be made to the algorithm by reformulating the scores and inner regression coefficients. Let $\mathbf{t}_i^*, \mathbf{u}_i^*, \mathbf{p}_i^*, \mathbf{q}_i^*$ and b_i^* denote the quantities defined above for the i^{th} latent variable but redefined so that the norms of the loading vectors are of unit length. The normalized loading vector $\mathbf{p}_i^*$ is given by:

$$\mathbf{p}_i^* = \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|} \tag{2.64}$$

and the contribution, $\mathbf{t}_i\mathbf{p}_i^T$ of the i^{th} latent variable to the matrix **X** is re-written as:

$$\mathbf{t}_i\mathbf{p}_i^T = \mathbf{t}_i\|\mathbf{p}_i\|\mathbf{p}_i^{*T} = \mathbf{t}_i^*\mathbf{p}_i^{*T} \tag{2.65}$$

where the redefined score vector $\mathbf{t}_i^*$ is given by:

$$\mathbf{t}_i^* = \mathbf{t}_i\|\mathbf{p}_i\| \tag{2.66}$$

The unit norm loading vector $\mathbf{q}_i^*$ is given by:

$$\mathbf{q}_i^* = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} \tag{2.67}$$

The contribution to the **Y**-matrix can be similarly re-written as:

$$\hat{\mathbf{u}}_i \mathbf{q}_i^T = b_i \mathbf{t}_i \mathbf{q}_i^T = b_i \frac{\mathbf{t}_i^*}{\|\mathbf{p}_i\|} \mathbf{q}_i^{*T} \|\mathbf{q}_i\| \tag{2.68}$$

and the inner regression coefficient can thus be defined as:

$$b_i^* = b_i \frac{\|\mathbf{q}_i\|}{\|\mathbf{p}_i\|} \tag{2.69}$$

Substituting equation (2.69) into equation (2.68), gives

$$\hat{\mathbf{u}}_i \mathbf{q}_i^T = b_i \mathbf{t}_i \mathbf{q}_i^T = b_i^* \mathbf{t}_i^* \mathbf{q}_i^{*T} \tag{2.70}$$

with the regression coefficient in terms of the scaled scores being given by:

$$b_i^* = \frac{\mathbf{u}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i} \frac{\|\mathbf{q}_i\|}{\|\mathbf{p}_i\|} = \frac{\mathbf{u}_i^T \mathbf{t}_i^*}{\mathbf{t}_i^{*T} \mathbf{t}_i^*} \tag{2.71}$$

The rescaled scores vector $\mathbf{t}_i^*$ can be calculated by rescaling the weight vector $\mathbf{w}_i$:

$$\mathbf{t}_i^* = \mathbf{X}_i \mathbf{w}_i \|\mathbf{p}_i\| = \mathbf{X}_i \mathbf{w}_i^* \tag{2.72}$$

where

$$\mathbf{w}_i^* = \mathbf{w}_i \|\mathbf{p}_i\| \tag{2.73}$$

is the rescaled weight vector

One important property that can be proven from the above is that the output weight vectors $\mathbf{v}_i$ and the output loading vectors $\mathbf{q}_i^*$ are the same. Without loss of generality, this is shown

for the first weight vector $\mathbf{v}_1$ and loading vector $\mathbf{q}_1^*$. From equation (2.58), the loading vector $\mathbf{q}_1$ is given by:

$$\mathbf{q}_1 = \frac{\mathbf{Y}^T\hat{\mathbf{u}}_1}{\hat{\mathbf{u}}_1^T\hat{\mathbf{u}}_1} = \frac{b_1\mathbf{Y}^T\mathbf{t}_1}{b_1^2\mathbf{t}_1^T\mathbf{t}_1} = \frac{\mathbf{Y}^T\mathbf{t}_1}{b_1\mathbf{t}_1^T\mathbf{t}_1} \tag{2.74}$$

and the normalized loading vector is given by:

$$\mathbf{q}_1^* = \frac{\mathbf{Y}^T\mathbf{t}_1}{\left\|\mathbf{Y}^T\mathbf{t}_1\right\|} \tag{2.75}$$

Comparing this with equation (2.48) for the weight vector $\mathbf{v}_1$, it can be seen that they are equivalent.

2.4.2.3 The NIPALS Algorithm

The concepts previously explained are collated into the NIPALS (Non-linear Iterative Partial Least Squares) algorithm as proposed by Wold (1966(a); 1966(b)). The complete algorithm is summarized in Table 2.1.

Table 2.1: NIPALS Algorithm

Step	Description	Equation
1	**Given:** Matrices **X** and **Y** Mean centre and scale each variable to unit variance. Set i (number of latent variable) = 1; j (number of iteration)=1 and $\mathbf{X}_1 = \mathbf{X}$ $\mathbf{Y}_1 = \mathbf{Y}$	
2	Initialize the u-scores vector, **u**	$\mathbf{u}_i$ = some column of $\mathbf{Y}_i$
3	Calculate the **w**-weight vector	$\mathbf{w}_{j,i} = \frac{\mathbf{X}_i^T\mathbf{u}_{j,i}}{\mathbf{u}_{j,i}^T\mathbf{u}_{j,i}}$
4	Normalize the **w**-weight vector	$\mathbf{w}_{j,i} = \frac{\mathbf{w}_{j,i}}{\left\|\mathbf{w}_{j,i}\right\|}$

5	Calculate the t-scores	$\mathbf{t}_{j,i} = \mathbf{X}_i \mathbf{w}_{j,i}$
6	Fit the inner relation	$\mathbf{u}_{j,i} = b_{j,i}\mathbf{t}_{j,i} + \mathbf{e}_{j,i}$
7.	Calculate the prediction of the u-scores	$\hat{\mathbf{u}}_{j,i} = b_{j,i}\mathbf{t}_{j,i}$
8	Calculate the $\mathbf{q}$-loading vector	$\mathbf{q}_{j,i} = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_{j,i}}{\mathbf{t}_{i,k}^T \mathbf{t}_{i,k}}$
9	Determine the **v**- weight vector	$\mathbf{v}_{j,i} = \frac{\mathbf{q}_{j,i}}{\left\|\mathbf{q}_{j,i}\right\|}$
10	Calculate the new u-scores	$\mathbf{u}_{j+1,i} = \mathbf{Y}_i \mathbf{v}_{j,i}$
11	Check for convergence	If $\left\|\mathbf{u}_{j+1,i} - \mathbf{u}_{j,i}\right\| \geqslant \in$, $j = j+1$, go to step 3, else go to step 12
12	Fit the linear inner relation	$\mathbf{u}_i = b_i \mathbf{t}_i + \mathbf{e}_i$
13	Predict the u-scores	$\hat{\mathbf{u}}_i = b_i \mathbf{t}_i$
14	Determine the **p**-loading vector	$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$
15	Determine the **q**-loading vector	$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \mathbf{u}_i}$
16	Deflate the predictor matrix	$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$
17	Deflate the response matrix	$\mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{t}_i \mathbf{q}_i^T$
18	If additional latent variables are required, repeat steps 2-17 by replacing $\mathbf{X}_i$ and $\mathbf{Y}_i$ with $\mathbf{X}_{i+1}$ and $\mathbf{Y}_{i+1}$ respectively.	$i = i+1$

Further details of the PLS algorithm can be found in (Geladi and Kowalsky, 1986; Höskuldsson, 1988; Martens and Næs, 1989; Wold et al., 2001(a); Helland, 2001). Historical details leading to the development of PLS and its impact are given in (Geladi, 1992; Wold, 2001; Martens, 2001) with more recent developments in the algorithm being summarised in (Wold et al., 2001(b)).

2.4.2.4 Properties of Partial Least Squares

In this section the properties of the weight and loading vectors in the PLS algorithm are summarised. These properties were first comprehensively proven by Höskuldsson (1988). All the properties of PLS follow from the way the deflated matrix $\mathbf{X}_j$ is computed from the previous deflated matrices. The relationship between $\mathbf{X}_j$ and $\mathbf{X}_i$ for (i< j) can be derived as follows. From the deflation procedure of the NIPALS algorithm:

$$\begin{aligned}\mathbf{X}_j &= \mathbf{X}_{j-1} - \mathbf{t}_{j-1}\mathbf{p}_{j-1}^T \\ &= \mathbf{X}_{j-1} - \left[\frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{\mathbf{t}_{j-1}^T\mathbf{t}_{j-1}}\right]\mathbf{X}_{j-1} \\ &= \left[\mathbf{I} - \frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{\mathbf{t}_{j-1}^T\mathbf{t}_{j-1}}\right]\mathbf{X}_{j-1} \\ &= \left[\mathbf{I} - \frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{\mathbf{t}_{j-1}^T\mathbf{t}_{j-1}}\right]\left[\mathbf{X}_{j-2} - \frac{\mathbf{t}_{j-2}\mathbf{t}_{j-2}^T}{\mathbf{t}_{j-2}^T\mathbf{t}_{j-2}}\mathbf{X}_{j-2}\right] \\ &= \mathbf{Z}\left[\mathbf{X}_{j-2} - \frac{\mathbf{t}_{j-2}\mathbf{t}_{j-2}^T}{\mathbf{t}_{j-2}^T\mathbf{t}_{j-2}}\mathbf{X}_{j-2}\right]\end{aligned} \tag{2.76}$$

where

$$\mathbf{Z} = \left[\mathbf{I} - \frac{\mathbf{t}_{j-1}\mathbf{t}_{j-1}^T}{\mathbf{t}_{j-1}^T\mathbf{t}_{j-1}}\right] \tag{2.77}$$

By following the above recursive procedure, the relationship between $\mathbf{X}_j$ and $\mathbf{X}_i$ can be established:

$$\mathbf{X}_j = \mathbf{Z}\left[\mathbf{X}_i - \frac{\mathbf{t}_i\mathbf{t}_i^T}{\mathbf{t}_i^T\mathbf{t}_i}\right] \quad \text{for } (i < j) \tag{2.78}$$

where $\mathbf{Z}$ is a matrix written as a cascade of the matrices of the form given in equation (2.77).

Property 1: The weight vectors $\mathbf{w}_i$'s are mutually orthogonal, i.e.

$$\mathbf{w}_i^T \mathbf{w}_j = 0 \qquad \text{for } i \neq j \tag{2.79}$$

Proof: First it is shown that

$$\mathbf{X}_j \mathbf{w}_i = \mathbf{0} \qquad \text{for } (i < j) \tag{2.80}$$

From equation (2.78)

$$\mathbf{X}_j \mathbf{w}_i = \mathbf{Z}\left[\mathbf{X}_i - \left(\frac{\mathbf{t}_i \mathbf{t}_i^T}{\mathbf{t}_i^T \mathbf{t}_i}\right)\mathbf{X}_i\right]\mathbf{w}_i = \mathbf{Z}\left[\mathbf{t}_i - \left(\frac{\mathbf{t}_i \mathbf{t}_i^T}{\mathbf{t}_i^T \mathbf{t}_i}\right)\mathbf{t}_i\right] = \mathbf{0} \tag{2.81}$$

It is also known that the weight vector $\mathbf{w}_j$ can be calculated by solving an eigenvector problem:

$$\mathbf{X}_j^T \mathbf{Y}_j \mathbf{Y}_j^T \mathbf{X}_j \mathbf{w}_j = \lambda_j \mathbf{w}_j \tag{2.82}$$

Taking the transpose of both sides of equation (2.82)

$$\mathbf{w}_j^T = \mathbf{w}_j^T \mathbf{X}_j^T \mathbf{Y}_j \mathbf{Y}_j^T \mathbf{X}_j \tag{2.83}$$

and then using equation (2.80)

$$\mathbf{w}_j^T \mathbf{w}_i = \mathbf{w}_j^T \mathbf{X}_j^T \mathbf{Y}_j \mathbf{Y}_j^T \mathbf{X}_j \mathbf{w}_i / \lambda_j = 0 \tag{2.84}$$

Hence the required proof.

Property 2: The scores are mutually orthogonal, i.e.

$$\mathbf{t}_i^T \mathbf{t}_j = 0 \tag{2.85}$$

Proof: This can be proven by recalling the deflation procedure in the PLS algorithm.

$$\begin{aligned}\mathbf{X}_j &= \mathbf{X}_{j-1} - \mathbf{t}_{j-1}\mathbf{p}_{j-1}^T \\ &= \mathbf{X}_{j-1} - \left[\frac{\mathbf{X}_{j-1}\mathbf{w}_{j-1}\mathbf{t}_{j-1}^T\mathbf{X}_{j-1}}{\mathbf{t}_{j-1}^T\mathbf{t}_{j-1}}\right] \\ &= \mathbf{X}_{j-1}\left[\mathbf{I} - \frac{\mathbf{w}_{j-1}\mathbf{t}_{j-1}^T\mathbf{X}_{j-1}}{(\mathbf{t}_{j-1}^T\mathbf{t}_{j-1})}\right] \\ &= \mathbf{X}_{i+1}\mathbf{Z} \qquad \text{for } i < j \\ &= \left[\mathbf{X}_i - \frac{\mathbf{t}_i\mathbf{t}_i^T}{\mathbf{t}_i^T\mathbf{t}_i}\mathbf{X}_i\right]\mathbf{Z}\end{aligned} \tag{2.86}$$

where $\mathbf{Z}$ is a matrix product that satisfies the matrix equation. From equation (2.86)

$$\mathbf{t}_i^T\mathbf{X}_j = \left[\mathbf{t}_i^T\mathbf{X}_i - \frac{\mathbf{t}_i^T\mathbf{t}_i\mathbf{t}_i^T\mathbf{X}_i}{\mathbf{t}_i^T\mathbf{t}_i}\right]\mathbf{Z} = \mathbf{0} \quad \text{for } i < j \tag{2.87}$$

and then by post multiplying by $\mathbf{w}_i$ on both sides of equation (2.87):

$$\mathbf{t}_i^T\mathbf{X}_j\mathbf{w}_j = \mathbf{t}_i^T\mathbf{t}_j = 0 \tag{2.88}$$

Hence the required proof.

2.4.2.5 PLS Regression Matrix

The PLS regression matrix $\mathbf{B}_{PLS}$ establishes the link between the input variables matrix $\mathbf{X}$ and the output variables matrix $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{B}_{PLS} + \mathbf{F} \tag{2.89}$$

If the PLS model is identified using latent variables, there exist different expressions for the regression matrix $\mathbf{B}_{PLS}$. One simple expression, given below, was derived by Lindgren et al., (1993):

$$\mathbf{B}_{PLS} = \mathbf{w}_1^* \mathbf{q}_1^T + \mathbf{w}_2^* \mathbf{q}_2^T + \ldots + \mathbf{w}_A^* \mathbf{q}_A^T \quad (2.90)$$

where

$$\begin{aligned} \mathbf{w}_1^* &= \mathbf{w}_1 \\ \mathbf{w}_2^* &= (\mathbf{I} - \mathbf{w}_1 \mathbf{p}_1^T)\mathbf{w}_2 \\ &\cdot \\ &\cdot \\ \mathbf{w}_A^* &= (\mathbf{I} - \mathbf{w}_1 \mathbf{p}_1^T)(\mathbf{I} - \mathbf{w}_2 \mathbf{p}_2^T) \ldots (\mathbf{I} - \mathbf{w}_{A-1} \mathbf{p}_{A-1}^T)\mathbf{w}_A \end{aligned} \quad (2.91)$$

Another important expression for the PLS regression matrix was derived by Manne (1987) and Helland, (1988):

$$\mathbf{B}_{PLS} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\ \mathbf{Q}^T \quad (2.92)$$

where

$$\begin{aligned} \mathbf{W} &= [\mathbf{w}_1\ \mathbf{w}_2 \ldots\ \mathbf{w}_A] \\ \mathbf{P} &= [\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_A] \\ \mathbf{Q} &= [\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_A] \end{aligned} \quad (2.93)$$

It is also worth noting here that a distinction is made between the two PLS algorithms depending on the number of output variables. If the number of output variables is one, then the algorithm is referred to as PLS1 whilst for the case when there are multiple output variables the algorithm is designated PLS2. It is observed that the former algorithm is simpler, has optimal properties and is easier to handle theoretically which makes it suitable for comparison with other regression methods.

2.4.2.6 Kernel Algorithms - Modifications of the NIPALS Algorithm

The kernel algorithm, as a modification of the NIPALS algorithm, was proposed by Lindgren et al., (1993). The modification was motivated particularly for situations where the number of observations is much larger than the number of input and/or output variables. The direct

application of the NIPALS algorithm in such a situation would not only require large memory for the storage of the scores vectors (as the size of a scores vector is equal to the number of observations) but the computational effort is also significant. One typical application where the number of objects is much larger than the number of variables is multivariate image analysis (Geladi and Grahn, 1997). Each pixel represents an object and therefore in a 512 × 512 image the number of objects is 262144 which is much larger than the number of variables (which is equal to number of wavelengths and typically lies between 5 and 25). The basic idea is to compute the parameters of PLS, namely the weight vectors, $\mathbf{w}_i$ and $\mathbf{v}_i$, the loading vectors, $\mathbf{p}_i$ and the regression matrix, $\mathbf{B}_{PLS}$, without calculating the scores.

From the previous discussion, the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$ can be determined as the eigenvectors of the matrices $\mathbf{X}_i^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{X}_i$ and $\mathbf{Y}_i^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_i$ where $\mathbf{X}_i$ and $\mathbf{Y}_i$ represent the deflated matrices at the i^{th} step of the iteration with $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{Y}_1 = \mathbf{Y}$. The order of the matrices $\mathbf{X}_i^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{X}_i$ and $\mathbf{Y}_i^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_i$, which are known as the kernel matrices, are $(K\times K)$ and $(M\times M)$ respectively and is independent of the number of observations (objects). Therefore, the resources (speed and memory of computing devices) required for the computation of the weight vectors are unaffected by a large number of observations. Since only the eigenvector corresponding to the maximum eigenvalue of the kernel matrices is required, any iterative method for calculating the eigenvalue of the square matrix, e.g. power method (Golub and Loan, 1996) can be used to determine the weight vectors. However, only the weight vector $\mathbf{w}_i$ need to be determined iteratively as the weight vector $\mathbf{v}_i$ and the loading vector $\mathbf{p}_i$ can be determined from knowledge of the weight vector $\mathbf{w}_i$. That is, from steps 9, 14 and 15 of the NIPALS algorithm given in section 2.4.2.1:

$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T\mathbf{t}_i}{\mathbf{t}_i^T\mathbf{t}_i} = \frac{\mathbf{Y}_i^T\mathbf{X}_i\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{X}_i^T\mathbf{X}_i\mathbf{w}_i} \qquad (2.94)$$

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|}$$

$$\mathbf{p}_i = \frac{\mathbf{X}_i^T\mathbf{t}_i}{\mathbf{t}_i^T\mathbf{t}_i} = \frac{\mathbf{X}_i^T\mathbf{X}_i\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{X}_i^T\mathbf{X}_i\mathbf{w}_i}$$

It should be noted that matrices $\mathbf{Y}_i^T\mathbf{X}_i$ and $\mathbf{X}_i^T\mathbf{X}_i$ are required to calculate the loading vectors $\mathbf{q}_i$ (and weight vector $\mathbf{v}_i$) and $\mathbf{p}_i$. Therefore, it can be concluded that determination

of the PLS parameters depends on three matrices $\mathbf{X}_i^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{X}_i$, $\mathbf{Y}_i^T\mathbf{X}_i$ and $\mathbf{X}_i^T\mathbf{X}_i$. To determine these matrices, a deflation procedure that does not require the calculation of scores is required. Lindgren et al., (1993) proposed a deflation procedure by taking into consideration the fact that the matrix **Y** need not be deflated (Höskuldsson, 1988) and that matrix **X** can be deflated by post multiplying it by a matrix of order *(K× K)* (this is again independent of the number of observations *N*):

$$\mathbf{X}_i = \mathbf{X}(\mathbf{I} - \mathbf{w}_1\mathbf{p}_1^T)(\mathbf{I} - \mathbf{w}_2\mathbf{p}_2^T)\ldots(\mathbf{I} - \mathbf{w}_{i-1}\mathbf{p}_{i-1}^T) \qquad (2.95)$$

Since the matrix **Y** need not be deflated, the three kernel matrices can be written as $\mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i$, $\mathbf{Y}^T\mathbf{X}_i$, $\mathbf{X}_i^T\mathbf{X}_i$. Adopting the notation:

$$(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})_i = \mathbf{X}_i^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}_i \qquad (2.96)$$

the kernel matrices can be computed recursively (Lindgren et al., 1993):

$$\begin{aligned}(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})_{i+1} &= (\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)^T(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})_i(\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)\\ (\mathbf{X}^T\mathbf{X})_{i+1} &= (\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)^T(\mathbf{X}^T\mathbf{X})_i(\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)\\ (\mathbf{Y}^T\mathbf{X})_{i+1} &= (\mathbf{Y}^T\mathbf{X})_i(\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)\end{aligned} \qquad (2.97)$$

After the weight vectors and the loading vectors have been determined, the regression matrix can be calculated using the formula given in equation (2.92). This formula is only dependent on the weight vectors and the loading vectors and, therefore, the regression matrix can be determined without calculating the scores. The disadvantage of this formula is that it requires the calculation of an inverse, which can be computationally expensive. Lindgren et al. (1993) also derived a formula for the regression matrix which does not require the inversion of a matrix. The formula is given in equations (2.90) and (2.91).

Another Kernel algorithm was proposed by Rannar et al., (1994) and was motivated by applications where the number of observations (objects) is fewer than the number of variables. This situation is common in analytical chemistry, e.g. spectroscopic data. Since in this situation the dimension of the score vectors is less than the weight vectors, the score vectors are first determined as the eigenvectors of the kernel matrices (Höskuldsson, 1988) and the weight vectors are then derived from the score vectors.

Some improvements to the Kernel algorithm were proposed by Jong and Braak (1994) so as to increase the speed of computation. They proposed a procedure for the deflation of the kernel matrices which is computationally less expensive than that given in equation (2.95) for the kernel algorithm of Lindgren et al., (1993). The core of the argument in (Jong and Braak, 1994) is that if the input matrix $\mathbf{X}$ has rank A (say), then matrices $\mathbf{X}$ and $\mathbf{Y}$ can be decomposed as:

$$\begin{aligned}\mathbf{X} &= \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \ldots \mathbf{t}_A\mathbf{p}_A^T = \mathbf{T}\mathbf{P}^T \\ \mathbf{Y} &= \mathbf{t}_1\mathbf{q}_1^T + \mathbf{t}_2\mathbf{q}_2^T + \ldots \mathbf{t}_A\mathbf{q}_A^T + \mathbf{F} = \mathbf{T}\mathbf{Q}^T + \mathbf{F}\end{aligned} \tag{2.98}$$

where $\mathbf{T}$ is the score matrix, $\mathbf{P}$ and $\mathbf{Q}$ are the loading matrices for $\mathbf{X}$ and $\mathbf{Y}$ and $\mathbf{F}$ is the error matrix. Now

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= \mathbf{P}\mathbf{T}^T\mathbf{T}\mathbf{P}^T = (\mathbf{t}_1^T\mathbf{t}_1)\mathbf{p}_1\mathbf{p}_1^T + (\mathbf{t}_2^T\mathbf{t}_2)\mathbf{p}_2\mathbf{p}_2^T + \ldots + (\mathbf{t}_A^T\mathbf{t}_A)\mathbf{p}_A\mathbf{p}_A^T \\ \mathbf{X}^T\mathbf{Y} &= \mathbf{Q}\mathbf{T}^T\mathbf{T}\mathbf{Q}^T = (\mathbf{t}_1^T\mathbf{t}_1)\mathbf{q}_1\mathbf{p}_1^T + (\mathbf{t}_2^T\mathbf{t}_2)\mathbf{q}_2\mathbf{p}_2^T + \ldots + (\mathbf{t}_A^T\mathbf{t}_A)\mathbf{q}_A\mathbf{p}_A^T\end{aligned} \tag{2.99}$$

The above equations suggest a deflation procedure as follows:

$$\begin{aligned}\left(\mathbf{X}^T\mathbf{X}\right)_{i+1} &= (\mathbf{X}^T\mathbf{X})_i - (\mathbf{t}_i^T\mathbf{t}_i)\mathbf{p}_i\mathbf{p}_i^T \\ \left(\mathbf{X}^T\mathbf{Y}\right)_{i+1} &= (\mathbf{X}^T\mathbf{Y})_i - (\mathbf{t}_i^T\mathbf{t}_i)\mathbf{q}_i\mathbf{p}_i^T\end{aligned} \tag{2.100}$$

Deflation by using equations (2.100) is less expensive compared to equation (2.95) of the kernel algorithm of Lindgren et al., (1993) since it avoids the multiplication of $\mathbf{X}_i^T\mathbf{X}_i$ and $\mathbf{X}_i^T\mathbf{Y}$ by the factor $(\mathbf{I} - \mathbf{w}_i\mathbf{p}_i^T)$. Another computational saving is proposed by starting the iteration with the calculation of the output weight (and loading vector) $\mathbf{q}_i$ rather than with the calculation of the input weight vector $\mathbf{w}_i$. The logic behind this is that the dimension of the matrix $\mathbf{Y}_i^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_i$ is usually smaller than that of $\mathbf{X}_i^T\mathbf{Y}\ \mathbf{Y}^T\mathbf{X}_i$ as the number of output variables is usually smaller than the number of input variables. The main steps of the algorithm are:

1. Calculate the weight vector $\mathbf{q}_i$ through either the eigenanalysis of $\mathbf{Y}_i^T\mathbf{X}_i\mathbf{X}_i^T\mathbf{Y}_i$ or by SVD of $(\mathbf{Y}_i^T\mathbf{X}_i)$

2. Calculate the weight vector $\mathbf{w}_i$:

$$\mathbf{w}_i = \mathbf{X}_i^T \mathbf{Y}_i \mathbf{q}_i = (\mathbf{X}^T\mathbf{Y})_i \mathbf{q}_i \qquad (2.101)$$
$$\mathbf{w}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}$$

3. Calculate the loading vector:

$$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_i}{\mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_i} = \frac{(\mathbf{X}^T\mathbf{X})_i \mathbf{w}_i}{\mathbf{w}_i^T (\mathbf{X}^T\mathbf{X})_i \mathbf{w}_i} \qquad (2.102)$$

4. Calculate the deflation:

$$(\mathbf{X}^T\mathbf{X})_{i+1} = (\mathbf{X}^T\mathbf{X})_i - (\mathbf{t}_i^T\mathbf{t}_i)\mathbf{p}_i\mathbf{p}_i^T \qquad (2.103)$$
$$(\mathbf{X}^T\mathbf{Y})_{i+1} = (\mathbf{X}^T\mathbf{Y})_i - (\mathbf{t}_i^T\mathbf{t}_i)\mathbf{q}_i\mathbf{p}_i^T$$

The factors $(\mathbf{t}_i^T\mathbf{t}_i)$ in the deflation equation (2.103) can be calculated as:

$$\mathbf{t}_i^T\mathbf{t}_i = \mathbf{w}_i^T\mathbf{X}_i^T\mathbf{X}_i\mathbf{w}_i = \mathbf{w}_i^T(\mathbf{X}^T\mathbf{X})_i\mathbf{w}_i \qquad (2.104)$$

Finally the regression matrix can be calculated as previously by using either equation (2.90) or (2.92).

Some further modifications were proposed by Dayal and MacGregor (1997(b)). They proved in their paper that to get the PLS solution either **X** or **Y** need to be deflated. The potential of this approach is that the user can select which matrix needs to be deflated. For example, if the input matrix has more variables than the output variables then it is advantageous to deflate matrix **Y**. Alternatively, if the number of output variables exceeds that of the input variables then **X** should be deflated. In the situation where matrix **X** is not deflated, the orthogonal scores can be calculated from the original (undeflated matrix) by finding a transformation matrix **R**, whose columns $\mathbf{r}_1, \mathbf{r}_2 \ldots \mathbf{r}_A$ can be computed recursively (Höskuldsson, 1988; Dayal and MacGregor, 1997(b)):

$$\mathbf{r}_1 = \mathbf{w}_1$$
$$\mathbf{r}_i = \mathbf{w}_i - \mathbf{p}_1^T\mathbf{w}_i\mathbf{r}_1 - \mathbf{p}_2^T\mathbf{w}_i\mathbf{r}_2 - \ldots\mathbf{p}_{i-1}^T\mathbf{w}_i\mathbf{r}_{i-1} \quad \text{for } i > 1 \tag{2.105}$$

2.5 Comparison of the Predictive Ability of PCR and PLS

Partial least squares and principal component regression are widely applied tools for the modelling of multivariate data. In the literature, their predictive ability has been compared from two perspectives: through simulations and by analytical means. One of the earlier comparisons was made by Næs and Martens (1985) who compared PLS and PCR using an artificially generated data set and (real) spectral data. It was shown that for both data sets, PLS performed better than PCR when the number of latent variables was less than a particular number (dependent on the data set) and PCR performed better than PLS when the number of latent variables was more than this number. The disadvantage of comparing the two methods by simulation is that while it is possible to demonstrate the superiority of one method over the other for a single data set, it is difficult to generalize the result. To address this problem for spectroscopic data, Thomas and Haalland (1990) designed a series of experiments to generate simulated data sets that resembled typical data and compared the performance of PLS, PCR and other two least squares methods Classical Least Squares (CLS) or the **K**- matrix method and the Inverted Least Squares (ILS) or **P**-matrix method. It was concluded that the performance of PCR and PLS were 'similar' except that PLS was found to be more suitable (in terms of prediction) over a wide range of conditions (e.g. presence of random baseline, presence of noise in the measure variables, etc.). For an extensive discussion on the relationship of PLS to other spectroscopic modelling techniques, see (Haaland and Thomas, 1988(a); 1988(b)).

Although the approach of Thomas and Haaland (1990) answered some of the important questions about the predictive ability of the two methods, it was still difficult to generalize this result as they were specific for spectroscopic data. Helland and Almoy (1994) were the first to derive the mathematical formulae for the mean square error. They assumed that *'there exist a number A such that A eigenvectors of the covariance matrix* **S** *of input variables, all corresponding to different eigenvalues, are related with the output variable* **y***'* such that the cross-covariance vector between the output and input variables can be written as:

$$\mathbf{s}_{xy} = \sum_{i=1}^{A} \alpha_i \mathbf{p}_i \tag{2.106}$$

where $\mathbf{s}_{xy}$ is the cross-covariance vector between the (multivariate) input variables and a single output variable, $\mathbf{p}_i$ are the eigenvectors of the covariance matrix **S** of the input variables and α_i are scalar constants.

The eigenvectors, which are correlated with the output variable **y** are called relevant eigenvectors and the corresponding eigenvalues are the relevant eigenvalues. The rest of the eigenvectors and eigenvalues are known as irrelevant. The formula for the average prediction error F_{PCR}, in a PCR model when the PCR model is identified with $a \geq A$ components is given by (Helland and Almoy, 1994):

$$F_{PCR} = \sigma^2\left(1+\frac{a+1}{N}\right)+\frac{1}{N}\sum_{j=a+1}^{K}\lambda_j^2\left(\sum_{r=1}^{A}\frac{\alpha_r^2}{\lambda_r(\lambda_r-\lambda_j)^2}\right)+o\left(\frac{1}{N}\right) \qquad (2.107)$$

The corresponding formula for PLS when the model is identified using A latent variable is given by:

$$F_{PLS} = \sigma^2\left(1+\frac{A+1}{N}\right)+\frac{1}{N}\sum_{j=A+1}^{K}\lambda_j^2 \times \left\{\omega_j^2\sum_{r=1}^{A}\frac{\alpha_r^2}{\lambda_r(\lambda_r-\lambda_j)^2}+\sigma^2\left(\frac{1-\omega_j}{\lambda_j}\right)^2\right\}+o\left(\frac{1}{N}\right) \qquad (2.108)$$

where σ^2 is the variance of the output variable, λ_i are the eigenvalues of the matrix **S** and

$$\omega_j = \prod_{i=1}^{A}\left(\frac{(\lambda_i-\lambda_j)}{\lambda_i}\right) \qquad (2.109)$$

From the formulae given in equations (2.107) and (2.108), Helland and Almoy (1994), provided the following conclusions:

1. If all the irrelevant eigenvalues (corresponding to the eigenvectors of the covariance matrix of the input variables have no or weak correlations with the output variable) are small, then there is not much practical difference between the prediction ability of PCR and PLS with both of them giving good predictions when the irrelevant components are excluded. Also the smaller the size of irrelevant eigenvalues (that is smaller magnitudes of the irrelevant eigenvalues), the better is the performance of PCR over PLS.
2. As the size of the irrelevant eigenvalue/s increases and approaches the smallest relevant eigenvalue or if an irrelevant eigenvalue is close in magnitude to any other relevant eigenvalue, then the performance of PCR is poor and in this situation PLS is better.
3. If the irrelevant eigenvalues lie between the smallest and largest relevant eigenvalues, then it is difficult to determine which approach is the best and very much depends on other parameters
4. When the irrelevant eigenvalues are quite high (that is, larger than the highest relevant eigenvalue) then, PCR performs better than PLS

The final conclusions that can be drawn are that PCR is best when either the irrelevant eigenvalues are small or very large and PLS is best for intermediate irrelevant eigenvalues. Since the difference between PCR and PLS is quite small when the irrelevant eigenvalues are small, and large irrelevant eigenvalues rarely occur in practical data sets, Helland and Almoy (1994) concluded that PLS is the method of choice in most cases. PLS also has the advantage that it only requires a decision on the number of components A to be included in the model whereas in PCR, not only is the selection of the number of components A is required but it also requires which of the A components should be included in the model. This further justifies the choice of PLS over PCR.

2.6 PLS as a Parameter Estimator

In most application of PLS in chemometrics, it has been primarily used for prediction. A related problem in chemical and process engineering is parameter estimation (Englezos and Kalogerakis, 2000) where the objective is to identify the parameter such that it is as 'close' as possible to the true parameter value. In this section the performance of PLS1 when it is used for parameter estimation is studied. The objective of parameter estimation is to estimate the regression vector $\boldsymbol{\beta}$ in the linear regression equation:

$$y = \mathbf{x}^T\boldsymbol{\beta} + e \tag{2.110}$$

as 'accurately' as possible. The common method for estimating the regression vector $\boldsymbol{\beta}$ is to use Ordinary Least Squares (OLS). The problem with OLS estimates, as mentioned earlier, is that if the variables are strongly correlated then the variance of the estimate is high leading to unreliable estimates. Alternatively PLS can be used for parameter estimation. The expression for the estimate $\hat{\boldsymbol{\beta}}_{PLS}$ is given by (Helland, 1988):

$$\hat{\boldsymbol{\beta}}_{PLS} = \mathbf{W}_A\left(\mathbf{W}_A^T\mathbf{X}^T\mathbf{X}\mathbf{W}_A\right)^{-1}\mathbf{W}_A^T\mathbf{X}^T\mathbf{y} \tag{2.111}$$

It should be noted from equation (2.111) that the estimator for PLS is fundamentally different to the OLS and PCR estimators in that the estimate is a non-linear function of the output variable **y**. It is well known that the two parameters that are used to evaluate the quality of an estimator are 'bias' and variance. Because of the non-linearity of the estimator function it is more difficult to analyse the PLS estimator as compared to OLS and PCR estimators. However, from equation (2.111) it can be proven that when the number of latent variables A in PLS is equal to the number of variables K (number of columns) in the **X** matrix, the estimate $\hat{\boldsymbol{\beta}}_{PLS}$ given by PLS is equal to the least square solution, and therefore, is an unbiased estimate. However, when A ($A < K$) latent variables are retained in the PLS model then the estimate, in general, is biased.

Several attempts have been made to estimate the variance (covariance) of the PLS estimator. Phatak et al., (1993) linearized the non-linear estimator to estimate the variance and used this estimate to find the prediction intervals of the estimate. Denham (1997) suggested three methods namely bootstrapping, cross validation and local linearization of the non-linear function to estimate the variance of the estimate and the prediction intervals of the predicted value. Another approach to estimating the covariance matrix of estimates is based on matrix differential calculus (Phatak et al., 2002).

There is also some disagreement among researchers regarding the significance of regression coefficients in PLS regression. One group of researchers view the PLS regression coefficients as a causal link between the observations **X** and **y** as in conventional linear regression whereas the other group views it as a latent variable model (Burnham, et al., 2001) where the observations **X** and **y** are seen as being generated by a common set of latent variables.

Some comparisons of the PLS estimator with other estimators, namely, OLS estimator and PCR estimators have also been performed. De Jong (1995) showed that the Euclidean norm of the PLS estimate is less than the OLS estimate while Stoica et al., (1995) observed that the PLS and PCR estimates are equivalent to within a first order approximation.

2.6.1 Unbiased Estimate using Partial Least Squares

In the above section it was noted that the PLS estimate is biased when A ($A < K$) latent variables are retained. In this subsection condition, other than $A = K$, under which the PLS estimate is unbiased is considered. It is known that if the input vector $\mathbf{x}$ and the output variable y are jointly normally distributed, then equation (2.110) represents the best predictor of the output variable under the quadratic loss function with the regression parameter vector $\boldsymbol{\beta}$ given as (Therrien, 1992):

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}_{xy} \tag{2.112}$$

where $\boldsymbol{\Sigma}$ is the (population) covariance matrix of the input variables vector $\mathbf{x}$ and $\boldsymbol{\sigma}_{xy}$ is the (population) cross-covariance vector between the input variables $\mathbf{x}$ and the output variable y. The matrix $\boldsymbol{\Sigma}$ can be decomposed using singular value decomposition as:

$$\boldsymbol{\Sigma} = \sum_{i=1}^{K} \lambda_i \mathbf{p}_i \mathbf{p}_i^T \tag{2.113}$$

Substituting equation (2.113) into equation (2.112) gives:

$$\boldsymbol{\beta} = \sum_{i=1}^{K} \lambda_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \boldsymbol{\sigma}_{xy} \tag{2.114}$$

However, it may be that not all the directions, $\mathbf{p}_i$, in the input space are correlated with $\boldsymbol{\sigma}_{xy}$ (and hence the output variable). Consider the case where the first A ($A < K$) directions $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_A$ are correlated with $\boldsymbol{\sigma}_{xy}$ and the remaining (K-A) directions are orthogonal to $\boldsymbol{\sigma}_{xy}$, that is:

$$\mathbf{p}_i^T \boldsymbol{\sigma}_{xy} \neq 0 \text{ for } i \leq A$$
$$\mathbf{p}_i^T \boldsymbol{\sigma}_{xy} = 0 \text{ for } i > A \quad (2.115)$$

Using equation (2.115) in equation (2.114) gives:

$$\boldsymbol{\beta} = \sum_{i=1}^{A} \lambda_i^{-1} \mathbf{p}_i \mathbf{p}_i^T \boldsymbol{\sigma}_{xy} \quad (2.116)$$

From equation (2.120) it can be noted that under the condition described in equation (2.115), the true regression vector $\boldsymbol{\beta}$ lies in the subspace spanned by the eigenvectors $(\mathbf{p}_1, \mathbf{p}_2 \cdots \mathbf{p}_A)$. Helland (1990) (Theorem (2c)) proved that the weight matrix $\mathbf{W}_A$ and the eigenvectors, $(\mathbf{p}_1, \mathbf{p}_2 \cdots \mathbf{p}_A)$, span the same space. Using this theorem, it follows that $\boldsymbol{\beta}$ lies in the space spanned by the weight matrix $\mathbf{W}_A$. Also from equation (2.111), it can be noted that the PLS parameter estimate $\hat{\boldsymbol{\beta}}_{PLS}$ lies in the (column) space spanned by $\mathbf{W}_A$. Since the true regression parameter vector $\boldsymbol{\beta}$ and the estimate $\hat{\boldsymbol{\beta}}_{PLS}$ lie in the same space, the estimate is unbiased. This is demonstrated using the following simulation example.

2.6.1.1 Example

The properties of PLS as a parameter estimator under the assumption of equation (2.115) are illustrated by an example. Two cases are considered, namely when the measured variables are strongly correlated, and when they are mutually orthogonal.

Case 1: Collinear data

In this case 1000 observations of 5 measured variables are generated using 2 principal components (latent variables), $\mathbf{t}_1$ and $\mathbf{t}_2$ as

$$\mathbf{X} = [\mathbf{t}_1 \ \mathbf{t}_2][\mathbf{p}_1 \ \mathbf{p}_2]^T + \delta\mathbf{E} \quad (2.117)$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are orthogonal column vectors each of order (1000×1) containing samples drawn from a Gaussian distribution with mean zero and unit variance. The vectors, $\mathbf{p}_1$ and $\mathbf{p}_2 \in R^5$, are orthonormal loading vectors given as:

$$\mathbf{p}_1^T = [-0.1694 \quad -0.1429 \quad 0.3308 \quad 0.8321 \quad 0.3860]^T$$
$$\mathbf{p}_2^T = [0.7796 \quad 0.6079 \quad 0.0902 \quad 0.1058 \quad 0.0586]^T \qquad (2.118)$$

δ is a scalar and controls the multicollinearity in the matrix $\mathbf{X}$. For example, when δ is near zero, the rank of the matrix is 2 and the variables are highly collinear. As δ increases, collinearity becomes less severe. The matrix $\mathbf{E}$ in this example is considered to be fixed and its columns are independent and Gaussian distributed with mean zero and unit variance. Now if it is assumed that only (first) two directions are relevant then the regression vector $\boldsymbol{\beta}$ lies in the subspace spanned by the first two loading vectors and can be generated as a linear combination of the first two vectors. The linear combination in the example is chosen as:

$$\boldsymbol{\beta} = 5\mathbf{p}_1 + 7\mathbf{p}_2 \qquad (2.119)$$

The observations of output variables can be generated as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \qquad (2.120)$$

where $\boldsymbol{\varepsilon}$ represents measurement noise in the output variable and is assumed to be Gaussian distributed with variance 0.25.

Four data sets each consisting of 1000 data points were generated corresponding to δ = 0.0001, δ = 0.001, δ = 0.01 and δ = 0.1. The regression parameters are determined using OLS and PLS for each of the data sets. Tables 2.2 to 2.5 show the mean and standard deviation of the OLS and PLS estimates calculated over 10000 trials for each of the data sets.

Case 2: Independent input variables

In this case, the five measured variables in matrix $\mathbf{X}$ are orthogonal to each other (that is, no correlation exists between the variables). The matrix $\mathbf{X}$ is generated as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T \tag{2.121}$$

where $\mathbf{T}$ is an orthogonal matrix of order (1000×5) and $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2 ... \mathbf{p}_5]$ is an orthonormal matrix of order (5×5) given by:

$$\mathbf{P} = \begin{bmatrix} -0.1694 & 0.7796 & -0.3210 & -0.5381 & 0.0136 \\ -0.1429 & 0.6079 & 0.2917 & 0.7180 & 0.0016 \\ 0.3308 & 0.0902 & 0.8402 & -0.4070 & -0.0726 \\ 0.8321 & 0.1058 & -0.2562 & 0.1181 & 0.4458 \\ 0.3860 & 0.0586 & -0.2008 & 0.1239 & -0.8921 \end{bmatrix} \tag{2.122}$$

The regression vector $\boldsymbol{\beta}$ in this case is also assumed to lie in the space spanned by the loading vectors $\mathbf{p}_1$ and $\mathbf{p}_2$, as in Case1, and is generated by the linear combination given in equation (2.119). The observations of output variable are generated as in equation (2.120). A data set comprising one thousand data points is generated and again the parameters are estimated using OLS and PLS. Table 2.6 shows the mean and standard deviation of the OLS and PLS estimates calculated over 10000 trials.

Table 2.2: Mean and standard deviation of OLS and PLS estimates with $\delta = 0.0001$

Actual Parameter Vector	Average of estimated vector over 10000 trials		Standard deviation of estimated vector over 10000 trials	
	OLS	PLS	OLS	PLS
$\begin{bmatrix} -0.4831 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} 0.7900 \\ 0.4382 \\ 6.4343 \\ 4.9687 \\ 5.5875 \end{bmatrix}$	$\begin{bmatrix} -0.4383 \\ -0.4395 \\ 3.0529 \\ 7.2775 \\ 3.3599 \end{bmatrix}$	$\begin{bmatrix} 104.9964 \\ 156.1201 \\ 124.4889 \\ 82.7206 \\ 133.1420 \end{bmatrix}$	$\begin{bmatrix} 0.0118 \\ 0.0031 \\ 0.0097 \\ 0.0132 \\ 0.0085 \end{bmatrix}$

Table 2.3: Mean and standard deviation of OLS and PLS estimates with $\delta = 0.001$

Actual Parameter Vector	Average of estimated vector over 10000 trials		Standard deviation of estimated vector over 10000 trials	
	OLS	PLS	OLS	PLS
$\begin{bmatrix} -0.4831 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} -0.5882 \\ -0.1276 \\ 2.9949 \\ 7.3349 \\ 3.3143 \end{bmatrix}$	$\begin{bmatrix} -0.4832 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} 10.5958 \\ 15.5904 \\ 12.4929 \\ 8.1911 \\ 13.1648 \end{bmatrix}$	$\begin{bmatrix} 0.0119 \\ 0.0032 \\ 0.0098 \\ 0.0134 \\ 0.0085 \end{bmatrix}$

Table 2.4: Mean and standard deviation of OLS and PLS estimates with $\delta = 0.01$

Actual Parameter Vector	Average of estimated vector over 10000 trials		Standard deviation of estimated vector over 10000 trials	
	OLS	PLS	OLS	PLS
$\begin{bmatrix} -0.4831 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} -0.4625 \\ -0.4595 \\ 3.0773 \\ 7.2648 \\ 3.3663 \end{bmatrix}$	$\begin{bmatrix} -0.4830 \\ -0.4394 \\ 3.0528 \\ 7.2778 \\ 3.3601 \end{bmatrix}$	$\begin{bmatrix} 1.0650 \\ 1.5568 \\ 1.2569 \\ 0.8265 \\ 1.3332 \end{bmatrix}$	$\begin{bmatrix} 0.0120 \\ 0.0041 \\ 0.0101 \\ 0.0134 \\ 0.0087 \end{bmatrix}$

Table 2.5: Mean and standard deviation of OLS and PLS estimates with $\delta = 0.1$

Actual Parameter Vector	Average of estimated vector over 10000 trials		Standard deviation of estimated vector over 10000 trials	
	OLS	PLS	OLS	PLS
$\begin{bmatrix} -0.4831 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} -0.4835 \\ -0.4390 \\ 3.0529 \\ 7.2775 \\ 3.3604 \end{bmatrix}$	$\begin{bmatrix} -0.4834 \\ -0.4391 \\ 3.0525 \\ 7.2775 \\ 3.3599 \end{bmatrix}$	$\begin{bmatrix} 0.1061 \\ 0.1553 \\ 0.1242 \\ 0.0830 \\ 0.1334 \end{bmatrix}$	$\begin{bmatrix} 0.0215 \\ 0.0268 \\ 0.0236 \\ 0.0194 \\ 0.0244 \end{bmatrix}$

Table 2.6: Mean and standard deviation of OLS and PLS estimates for case 2

Actual Parameter Vector	Average of estimated vector over 10000 trials		Standard deviation of estimated vector over 10000 trials	
	OLS	PLS	OLS	PLS
$\begin{bmatrix} -0.4831 \\ -0.4394 \\ 3.0529 \\ 7.2776 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} -0.4834 \\ -0.4393 \\ 3.0531 \\ 7.2777 \\ 3.3600 \end{bmatrix}$	$\begin{bmatrix} -0.4831 \\ -0.4392 \\ 3.0531 \\ 7.2774 \\ 3.3598 \end{bmatrix}$	$\begin{bmatrix} 0.0156 \\ 0.0157 \\ 0.0157 \\ 0.0156 \\ 0.0158 \end{bmatrix}$	$\begin{bmatrix} 0.0217 \\ 0.0215 \\ 0.0211 \\ 0.0175 \\ 0.0208 \end{bmatrix}$

The following conclusions can be drawn from the above tables:

1. The parameter estimates determined using PLS is unbiased in all five cases. This is contrary to the common perception that when a PLS model is identified using fewer latent variables than the number of input variables, the PLS estimate is biased. The intuitive explanation for this is that under the assumption that A directions, where A is in general less than the number of input variables, in the $\mathbf{X}$ space are correlated with the output variable y and when the PLS model is built using A latent variables, no variance of y is left unexplained by the A latent variables and the estimate is therefore unbiased.

2. The OLS estimates appear to be biased when the variables are highly collinear (Tables 2.2 and 2.3). This is, however, contrary to the well known fact that the estimates in OLS are always unbiased. The explanation behind this anomaly is that when the variables are highly collinear, the variance, which is a measure of the uncertainty in parameters, is very high as seen in Tables 2.2 and 2.3 and therefore the average calculation over a finite data may not lead to the true parameter.

3. The variance estimate of the parameters in PLS is less than the variance of the estimates using OLS except in the last case where the variance of the PLS estimates is slightly greater than the OLS estimates. Again this conclusion seems to be contrary to the well known fact about OLS that it is the best estimate in the sense that no estimator can have less variance than the OLS estimate. The explanation behind this aspect is that OLS is a linear estimator in the sense that the estimate is a linear

function of the observations of the output variable y. When OLS is said to best, it is best among all linear estimators. A PLS estimator on the other hand is a non-linear estimator and therefore can have a smaller variance than the OLS estimator.

4. When a PLS model is identified for prediction purpose, the number of latent variables is not decided based on the fact that the maximum variance of y is to be explained. This is because by adopting this approach noise in the model will be fitted and therefore model performance will be poor on unseen data. Therefore, methods like cross validation are used to select the number of latent variables in the PLS model. On the other hand when PLS is to be used as a parameter estimator, these rules for selecting the number of latent variables may not be appropriate.

5. When using PLS as a parameter estimator, the pre-processing of data can have a serious effect on the estimates. For example, normally the data is auto-scaled before the model is identified. Auto-scaling of the data, which can be modelled as a linear operation on the data, can have serious effect on the performance of PLS estimator since PLS is a non-linear estimator and, therefore, the effect of auto-scaling on the estimates may be irreversible.

2.7 Conclusions

In a typical process a large number of strongly correlated variables are measured. To identify a model for the process from the measured data, it is useful to project the variables onto a set of orthogonal variables such that the new variables retain most of the information contained in the original data. Two projection techniques that have been widely used in modelling multivariate data from chemical and process industries are principal component analysis (and its application in regression, known as principal component regression) and partial least squares. In this chapter the basic theory behind these techniques has been reviewed and a literature review has been undertaken.

It can be very difficult for the user to decide whether to use PCR or PLS model. These techniques have been compared with respect to their prediction capability and a number of guidelines have been proposed for selecting between PCR and PLS.

PLS has been most widely applied in chemometrics for prediction. In this chapter an alternative application of PLS has been proposed namely in parameter estimation. It has been

shown from simulation study that under the assumption that if exactly A ($A < K$) directions in the input space are correlated with the output variables and the PLS model is built using A latent variables then PLS not only gives unbiased estimates of the parameters but also identifies them with lesser variance than that given by OLS estimator.

PLS forms the basis of the discussion of the next two chapters where the aim is to modify the basic PLS algorithm to make it more suitable for handling non-linearity and process dynamics.

CHAPTER 3

Non-linear Partial Least Squares

3.1. Introduction

In practice, when dealing with real chemical and physical systems, linear PLS cannot always be used to model the underlying structure since it may exhibit significant non-linear characteristics. A number of non-linear extensions to partial least squares have been proposed over the last decade to integrate non-linear features within the PLS framework. In this chapter, following an extensive review of non-linear PLS, the existing non-linear partial least squares algorithms are classified into three categories namely covariance based, quick and dirty and error based, on the basis of the underlying objective function. More specifically, a detailed mathematical analysis of the error based non-linear PLS algorithm proposed by Baffi et al., (1999(a)) is undertaken and it is proven that it is a non-linear extension of Reduced Rank Regression (RRR).

It has been widely reported that linear PLS is based on the maximization of the covariance between the t-and u-scores. This covariance based criterion realises a straight forward approach to the calculation of the scores variables and model parameters, as well as providing statistical interpretation of the parameters. This is essential in terms of assisting in the understanding of the behaviour of the underlying system. In this chapter it is argued that a 'true' non-linear PLS algorithm should be based on the maximization of a 'non-linear covariance' function. Following a detailed study of the algorithm by Wold et al., (1989), it is shown that although it has been considered as 'complicated', it is the only algorithm that attempts to maximize the non-linear covariance function. The optimization problem solved by Wold et al., (1989) is however, severely constrained in the sense that not all the parameters that influence the non-linear covariance function are used to optimize the objective function. To overcome this limitation, two new non-linear PLS algorithms are proposed that make use of a different set of constraints to maximize the non-linear covariance function. The performance of the proposed algorithms is evaluated on two artificial data sets and a benchmark simulation of a pH process.

3. 2. Literature Review

There are basically two approaches to extending linear PLS to its non-linear form. In the first approach, the input variables are first non-linearly transformed and linear PLS is applied to the transformed data set. For example, to model a quadratic non-linearity, the input data matrix **X** can be extended with the square terms, x_i^2 and the cross terms ($x_i x_j$), where x_i and x_j for $i, j = 1,2...K$, denote the K input variables (Ganadeskian, 1977). This method, which suffers from the disadvantage of making the size of the augmented matrix **X** large, was reviewed by Berglund and Wold (1997). They showed that in quadratic PLS, by including the squared terms in the data matrix **X**, both square and cross terms of the latent variables are implicitly included in the resulting PLS model. The implication of this result is that if a latent structure is present in the data, that is, if the measured variables are assumed to be generated by a set of hidden or latent variables then it is not necessary to include the cross terms in the augmented matrix serving to reduce the size of augmented matrix. The latest development in this class of algorithms is that of Reproducing Kernel Hilbert Space (RKHS) PLS (Rosipal and Trejo, 2001). The data in this algorithm is first transformed to a feature space using a reproducing kernel (Aronszajn, 1950) and then linear PLS is performed in the feature space. The focus of this chapter is however on the class of algorithms where a non-linear model is fitted to the (inner) latent variables.

Wold et al., (1989) in their seminal paper proposed that a non-linear relationship be introduced through the scores rather than through the predictor variables and suggested updating the weights of the outer relationship in an iterative manner thereby integrating the non-linearity within the PLS framework. Although they described the approach using a quadratic non-linear relationship, they stated that the method was applicable for any differentiable non-linear function. However their algorithm for weight updating was, in their own words 'complicated' and required to be 'improved by better algorithms'. Along with this algorithm they also proposed a method, which they termed 'quick and dirty'. For this approach the outer weights are determined by the standard (linear) PLS algorithm and a non-linear relationship is then fitted between the corresponding pair of t- and u-scores. This method, they conjectured, was appropriate for situations where the non-linearity involved was weak. More flexible non-linear models were proposed by Frank (1990) and Wold (1992). Whilst the former work included the use of a smoothing procedure, the technique of Wold (1992) was based on the use of spline functions. These methods, however, require a

number of parameters to be decided by the user including degree of spline and number of knots.

Qin and McAvoy (1992) proposed fitting a feed forward neural network between the corresponding pairs of scores. Since a feed forward neural network with one hidden layer of sigmoidal units can approximate any continuous function with arbitrary accuracy (Cybenko, 1989), the method of Qin and McAvoy can be used to approximate any non-linear relation between the latent variables, and is therefore widely applicable. It should, however, be noted that for this method the outer weights are not updated as an integral part of the non-linear relationship. The outer weights are determined as per linear PLS, i.e. this is a 'quick and dirty' method of identifying a non-linear PLS model. Other approaches to building non-linear PLS models using neural networks have also been reported. Wilson et al., (1997) described an approach whereby a Radial Basis Function (RBF) network was used to model the non-linear relationship between the scores. The methodology was applied to model the Tennessee Eastman process.

Walczak and Massart (1996) used a RBF network to first non-linearly transform the input variables prior to applying PLS. A further approach proposed by Malthouse (Malthouse, 1995; Malthouse et al., 1997) to generalize linear PLS to its non-linear form was to project the predictor variables onto curves (which were parameterized by a feed forward neural network) instead of lines as in linear PLS. The neural network parameters were determined by minimizing the sum of squares of the prediction errors between the actual values of the input and output variables and their corresponding approximations obtained from the projections. In this way the latent variables are determined so that a compromise is achieved between the predictably of the output variable and the approximation of the input variables from the latent variables. One of the limitations of this algorithm is that the latent variables in the input variables space are not orthogonal as is the case for linear PLS. Doymaz et al., (2003) proposed a modified version of the algorithm of Malthouse et al., (1997) which retains the orthogonal property of the latent variables in the input space.

A revision to the approach of Wold et al., (1989) was proposed by Baffi et al., (1999(a)) whereby the non-linear model was fully integrated within the framework of PLS by updating the outer weights using the prediction error of the inner scores model. This algorithmic approach also formed the basis of identifying a non-linear dynamic PLS model (Baffi et al., 2000). Other approaches to non-linear PLS include the use of Hammerstein and Wiener filters (Patwardhan et al., 1998), genetic programming (Hiden, et al., 1998) and the Box-

Tidwell transformation (Li et al., 2001). Min et al., (2002) suggested using a modified back propagation algorithm to integrate a feedforward neural network within the PLS framework. The iterative backpropagation algorithm, they argued, would circumvent the problem of calculating the pseudo-inverse for updating the weights in the algorithm proposed by Baffi et al., (1999).

3.3 Comments on Linear PLS

Before undertaking an analysis of non-linear PLS algorithms, some facts about linear PLS are stated that will be useful later in the chapter.

Remark 3.1: The vectors, $\mathbf{v}_i$, $\mathbf{q}_i$ and $\mathbf{p}_i$ in the PLS algorithm are functions of the weight vector $\mathbf{w}_i$ for $i = 1,2,...K$

Proof: The response variables projection direction, $\mathbf{v}_i$, is given by:

$$\mathbf{v}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\left\| \mathbf{Y}_i^T \hat{\mathbf{u}}_i \right\|} = \frac{\mathbf{Y}_i^T b_i \mathbf{t}_i}{\left\| \mathbf{Y}_i^T b_i \mathbf{t}_i \right\|} = \frac{\mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i}{\left\| \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i \right\|} \tag{3.1}$$

and loading vectors, $\mathbf{p}_i$ and $\mathbf{q}_i$, are given by:

$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \hat{\mathbf{u}}_i} = \frac{\mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i}{b_i \, \mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_i} \tag{3.2}$$

$$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i} = \frac{\mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_i}{\mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_i} \tag{3.3}$$

The above equations demonstrate the dependency of $\mathbf{v}_i$, $\mathbf{q}_i$ and $\mathbf{p}_i$ on $\mathbf{w}_i$. Thus it can be concluded that the weight vectors, $\mathbf{w}_i$ for $i = 1,2,...K$, completely characterize the PLS algorithm in the sense that all other parameters of PLS can be derived from them.

Remark 3.2: The vectors $\mathbf{v}_i$ and $\mathbf{q}_i$ lie in the same direction in the response space, i.e. $\mathbf{v}_i$ is proportional to $\mathbf{q}_i$.

Proof: It follows from equations (5.1) and (5.2) that $\mathbf{v}_i$ and $\mathbf{q}_i$ are related as:

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} \tag{3.4}$$

which proves that $\mathbf{v}_i$ and $\mathbf{q}_i$ are oriented in the same direction.

Remark 3.3: The scores vectors, $\mathbf{t}_i$ and $\mathbf{u}_i$, are determined to have maximum covariance.

Proof: This follows from the objective function of PLS stated in Chapter 2 (section 2.4.2)

3. 4 Review of Error Based Non-Linear Partial Least Squares

This class of algorithms for non-linear PLS was proposed by Baffi et al., (1999) and assumes that the t- and u-scores are related by a non-linear function, **f**:

$$\mathbf{u}_i = \mathbf{f}(\mathbf{t}_i) + \mathbf{e}_i \tag{3.5}$$

Wold et al., (1989) and Baffi et al., (1999(a)) took **f** to be a quadratic polynomial whilst a more general function for **f** in the form of a feedforward neural network was proposed by Baffi et al., (1999(b)). In these algorithms, the basic framework of PLS formed the basis of the approach. For example, both the property of orthogonality of the t-scores and the constraint on $\mathbf{v}_i$ to have the same orientation as that of $\mathbf{q}_i$ were retained in the non-linear extensions. The algorithms differ from linear PLS in the way that the weights $\mathbf{w}_i$ are determined. As mentioned above, while the maximization of the covariance between the t-scores and u-scores is the objective function for determining the weight vectors, $\mathbf{w}_i$, the approach of Baffi et al., (1999(a)) was based on the minimization of sum of squares of prediction error of the u-scores. Mathematically, the objective function of Error Based Non-Linear PLS (EBNPLS) proposed by Baffi et al., (1999(a)) is:

$$J_{EBNPLS} = \min_{\mathbf{w}_i} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|_2 = \min_{\mathbf{w}_i} \|\mathbf{u}_i - \mathbf{f}(\mathbf{t}_i)\|_2 = \min_{\mathbf{w}_i} \|\mathbf{Y}_i\mathbf{v}_i - \mathbf{f}(\mathbf{X}_i\mathbf{w}_i)\|_2 \quad (3.6)$$

Since the direction, $\mathbf{v}_i$, in the objective function is constrained to lie in the same direction as that of $\mathbf{q}_i$, the weight vectors $\mathbf{w}_i$ are solutions of the following constrained optimization problem:

$$J_{EBNPLS} = \min_{\mathbf{w}_i} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|_2 = \min_{\mathbf{w}_i} \|\mathbf{u}_i - \mathbf{f}(\mathbf{t}_i)\|_2 = \min_{\mathbf{w}_i} \|\mathbf{Y}_i\mathbf{v}_i - \mathbf{f}(\mathbf{X}_i\mathbf{w}_i)\|_2 \quad (3.7)$$

subject to

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} = \frac{\mathbf{Y}_i^T\hat{\mathbf{u}}_i}{\|\mathbf{Y}_i^T\hat{\mathbf{u}}_i\|} = \frac{\mathbf{Y}_i^T\mathbf{f}(\mathbf{t}_i)}{\|\mathbf{Y}_i^T\mathbf{f}(\mathbf{t}_i)\|} = \frac{\mathbf{Y}_i^T\mathbf{f}(\mathbf{X}_i\mathbf{w}_i)}{\|\mathbf{Y}_i^T\mathbf{f}(\mathbf{X}_i\mathbf{w}_i)\|}$$

$$\|\mathbf{w}_i\| = 1$$

The weight updating procedure for the weight vector, $\mathbf{w}_i$, is calculated through Newton-Raphson linearization of the function **f**:

$$\Delta\mathbf{w}_{j+1,i} = [\mathbf{Z}_{j,i}^T\mathbf{Z}_{j,i}]^{-1}\mathbf{Z}_{j,i}^T\mathbf{e}_{j,i} \quad (3.8)$$

$$\mathbf{w}_{j+1,i} = \frac{\mathbf{w}_{j,i} + \Delta\mathbf{w}_{j+1,i}}{\|\mathbf{w}_{j,i} + \Delta\mathbf{w}_{j+1,i}\|_2}$$

where $\mathbf{Z}_{j,i}$ is a matrix where the first order differential of the non-linear function of the i^{th} t-scores with respect to the weight vector, $\mathbf{w}_i$, are stored and $\Delta\mathbf{w}_{j+1,i}$ is the incremental change in the weight vector $\mathbf{w}_i$ for the j^{th} iteration. The complete error-based non-linear PLS algorithm is summarised in Table 3.1.

Table 3.1: Error based non-linear partial least squares (Baffi et al., 1999(a))

Step	Description	Equation
1	**Given**: Matrices **X** and **Y** Mean centre and scale each variable to unit variance. Set i (number of latent variable) = 1; j (number of iteration)=1 and $\mathbf{X}_1 = \mathbf{X}$ $\mathbf{Y}_1 = \mathbf{Y}$	

2	Initialize the u-scores vector **u**	$\mathbf{u}_i$ = some column of $\mathbf{Y}_i$
3	Calculate the **w**-weight vector	$\mathbf{w}_{j,i} = \frac{\mathbf{X}_i^T \mathbf{u}_{j,i}}{\mathbf{u}_{j,i}^T \mathbf{u}_{j,i}}$
4	Normalize the **w**-weight vector	$\mathbf{w}_{j,i} = \frac{\mathbf{w}_{j,i}}{\|\mathbf{w}_{j,i}\|}$
5	Calculate the t-scores	$\mathbf{t}_{j,i} = \mathbf{X}_i \mathbf{w}_{j,i}$
6	Fit the non-linear inner relationship	$\mathbf{u}_{j,i} = \mathbf{f}(\mathbf{t}_{j,i}) + \mathbf{e}_{j,i}$
7	Calculate the prediction of the u-scores	$\hat{\mathbf{u}}_{j,i} = \mathbf{f}(\mathbf{t}_{j,i})$
8	Calculate the **q**-loading vector	$\mathbf{q}_{j,i} = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_{j,i}}{\mathbf{t}_{i,k}^T \mathbf{t}_{i,k}}$
9	Determine the **v**-weight vector	$\mathbf{v}_{j,i} = \frac{\mathbf{q}_{j,i}}{\|\mathbf{q}_{j,i}\|}$
10	Calculate the new u-scores	$\mathbf{u}_{j+1,i} = \mathbf{Y}_i \mathbf{v}_{j,i}$
11	Update and normalize the weight vector **w** (equation 3.8)	
12	Calculate the new t-scores	$\mathbf{t}_{j+1,i} = \mathbf{X}_i \mathbf{w}_{j+1,i}$
13	Check for convergence	If $\|\mathbf{t}_{j+1,i} - \mathbf{t}_{j,i}\| \geqslant \in$, $j = j+1$, go to step 3, else go to step 14
14	Fit the non-linear inner relationship	$\mathbf{u}_i = \mathbf{f}(\mathbf{t}_i) + \mathbf{e}_i$
15	Predict the u-scores	$\hat{\mathbf{u}}_i = \mathbf{f}(\mathbf{t}_i)$
16	Determine the **p**-loading vector	$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$
17	Determine the **q**-loading vector	$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \mathbf{u}_i}$
18	Deflate the predictor matrix	$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$

19	Deflate the response matrix	$\mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{t}_i\mathbf{q}_i^T$
20	If additional latent variables are required, repeat steps 2-19 by replacing $\mathbf{X}_i$ and $\mathbf{Y}_i$ with $\mathbf{X}_{i+1}$ and $\mathbf{Y}_{i+1}$ respectively.	

The issue is that this algorithm is a non-linear extension of Reduced Rank Regression (RRR) rather than a non-linear extension of partial least squares. A brief review of reduced rank regression is given in the next section prior to providing this proof. For more details refer to (Reinsel and Velu, 1998).

3.5 Brief Overview of Reduced Rank Regression

In (linear) reduced rank regression the objective is to determine the weight vector, $\mathbf{w}_i$, such that the t-scores vector $\mathbf{t}_i$:

$$\mathbf{t}_i = \mathbf{X}_i\mathbf{w}_i \tag{3.9}$$

has maximum contribution to the response matrix $\mathbf{Y}_i$. The prediction error of the response matrix is defined as:

$$\mathbf{E}_i = \mathbf{Y}_i - \mathbf{t}_i\mathbf{q}_i^T \tag{3.10}$$

where $\mathbf{q}_i$ is a vector that is to be determined such that the norm of the error matrix $\mathbf{E}_i$ is a minimum. The vector $\mathbf{q}_i$ that has maximum contribution to $\mathbf{Y}_i$ can be determined using least squares:

$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T\mathbf{t}_i}{\mathbf{t}_i^T\mathbf{t}_i} \tag{3.11}$$

Since the vector $\mathbf{q}_i$ depends on vector $\mathbf{w}_i$ through $\mathbf{t}_i$, the prediction error in equation (3.10) is completely determined by the weight vector $\mathbf{w}_i$. The objective function of reduced rank regression can therefore be stated as:

$$J_{RRR} = \min_{\mathbf{w}_i} \left\| \mathbf{Y}_i - \mathbf{t}_i \mathbf{q}_i^T \right\| = \min_{\mathbf{w}_i} \left(\text{trace} \left(\mathbf{E}_i^T \mathbf{E}_i \right) \right) \quad (3.12)$$

Although the objective function in equation (3.12) does not require any constraint to be imposed on the weight vector $\mathbf{w}_i$ in order to keep it bounded, a unit norm constraint is placed on the weight vector $\mathbf{w}_i$ by multiplying the t-scores vector, $\mathbf{t}_i$, by a constant b_i which is determined in a similar manner as for $\mathbf{q}_i$ (minimization of the prediction error in the response matrix). Equation (3.10) can thus be written as:

$$\mathbf{Y}_i = b_i \mathbf{t}_i \mathbf{q}_i^T + \mathbf{E}_i = \hat{\mathbf{u}}_i \mathbf{q}_i^T + \mathbf{E}_i \quad (3.13)$$

where $\hat{\mathbf{u}}_i = b_i \mathbf{t}_i$. The objective function of reduced rank regression can be re-stated as:

$$J_{RRR} = \min_{\mathbf{w}_i} \left\| \mathbf{Y}_i - \hat{\mathbf{u}}_i \mathbf{q}_i^T \right\| = \min_{\mathbf{w}_i} \left(\text{trace} \left(\mathbf{E}_i^T \mathbf{E}_i \right) \right) \quad (3.14)$$
$$\text{subject to } \|\mathbf{w}_i\| = 1$$

3.6. Analysis of Error-Based Non-linear Partial Least Squares

To prove the equivalence between reduced rank regression and the algorithm of Baffi et al., (1999), the following theorem for reduced rank regression is proven.

Theorem 3.1: The objective function of reduced rank regression can be formulated in terms of minimizing the (sum of squares) u-scores prediction error with the constraint that the response variables projection direction $\mathbf{v}_i$ and the vector $\mathbf{q}_i$ lie in the same direction. That is, the objective function of reduced rank regression can be written mathematically as:

$$J_{RRR} = \min_{\mathbf{w}_i} \left\| \mathbf{u}_i - \hat{\mathbf{u}}_i \right\|_2 = \min_{\mathbf{w}_i} \left\| \mathbf{u}_i - b_i \mathbf{t}_i \right\|_2 \quad (3.15)$$
$$\text{subject to}$$
$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\left\| \mathbf{Y}_i^T \hat{\mathbf{u}}_i \right\|} = \frac{\mathbf{Y}_i^T b_i \mathbf{t}_i}{\left\| \mathbf{Y}_i^T b_i \mathbf{t}_i \right\|} = \frac{\mathbf{Y}_i^T \mathbf{t}_i}{\left\| \mathbf{Y}_i^T \mathbf{t}_i \right\|} \quad \text{and} \quad \|\mathbf{w}_i\| = 1$$

Proof: Define vector $\mathbf{v}_i$ as:

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} \tag{3.16}$$

Multiplying both sides of equation (3.13) by $\mathbf{v}_i$ gives:

$$\mathbf{Y}_i\mathbf{v}_i = \hat{\mathbf{u}}_i\mathbf{q}_i^T\mathbf{v}_i + \mathbf{E}_i\mathbf{v}_i \tag{3.17}$$

Substituting equation (3.16) into (3.17) gives:

$$\mathbf{Y}_i\mathbf{v}_i = \hat{\mathbf{u}}_i \frac{\mathbf{q}_i^T\mathbf{q}_i}{\|\mathbf{q}_i\|} + \mathbf{E}_i\mathbf{v}_i \tag{3.18}$$

Now taking $\mathbf{Y}_i\mathbf{v}_i = \mathbf{u}_i$, the u-scores, and $\mathbf{E}_i\mathbf{v}_i = \mathbf{e}_i$:

$$\mathbf{u}_i = \hat{\mathbf{u}}_i + \mathbf{e}_i \tag{3.19}$$

The weight vectors $\mathbf{w}_i$ in the RRR can, therefore, also be determined by minimizing the objective function:

$$J_{RRR} = \min_{\mathbf{w}_i} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|_2 \tag{3.20}$$

provided that the projection direction $\mathbf{v}_i$ is given as:

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} = \frac{\mathbf{Y}_i^T\hat{\mathbf{u}}_i}{\|\mathbf{Y}_i^T\hat{\mathbf{u}}_i\|} = \frac{\mathbf{Y}_i^T\mathbf{t}_i}{\|\mathbf{Y}_i^T\mathbf{t}_i\|} \tag{3.21}$$

The objective function of (linear) RRR can, therefore, be stated explicitly as in equation (3.15).

In the theorems below, the equivalence between the linear version of Baffi's algorithm and reduced rank regression is established.

Theorem 3.2: When the function, **f**, is linear, the objective function minimized in the algorithm of Baffi et al., (1999(a)) is the same as for reduced rank regression.

Proof: Replacing the non-linear function, **f**, in the inner scores by a linear function gives:

$$\hat{\mathbf{u}}_i = b_i \mathbf{t}_i \tag{3.22}$$

The objective function for the algorithm proposed by Baffi et al., (1999 (a)) given in equation (3.7) under the assumption of a linear relationship between the scores reduces to:

$$J = \min_{\mathbf{w}_i} \left\| \mathbf{u}_i - \hat{\mathbf{u}}_i \right\|_2 \tag{3.23}$$

subject to

$$\mathbf{v}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|} = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\left\|\mathbf{Y}_i^T \hat{\mathbf{u}}_i\right\|} = \frac{\mathbf{Y}_i^T b_i \mathbf{t}_i}{\left\|\mathbf{Y}_i^T b_i \mathbf{t}_i\right\|} = \frac{\mathbf{Y}_i^T \mathbf{t}_i}{\left\|\mathbf{Y}_i^T \mathbf{t}_i\right\|} \quad \text{and} \quad \|\mathbf{w}_i\| = 1$$

Comparing equations (3.15) and (3.23), the two objective functions are observed to be equivalent.

Theorem 3.3: When the function **f** in the algorithm of Baffi et al., (1999(a)) is linear, the weight vector, $\mathbf{w}_i$, in the iterative algorithm converges to the eigenvector corresponding to the largest eigenvalue of the matrix expression $[\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i$ and is equal to the weight vector determined by reduced rank regression.

Proof: From Table 3.1, the weight vector, $\mathbf{v}_i$, the regression coefficient, b_i, and the residual vector, $\mathbf{e}_i$, for the j^{th} iteration, can be expressed as a function of $\mathbf{w}_i$ as follows:

$$\mathbf{v}_{j,i} = \frac{\mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i}}{\left\| \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|}$$

$$b_{j,i} = \frac{\mathbf{w}_{j,i}^T \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i}{\left\| \mathbf{w}_{j,i}^T \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i \right\| \left\| \mathbf{w}_{j,i}^T \mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|}$$

$$\mathbf{e}_{j,i} = \frac{\mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i}}{\left\| \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|} - \mathbf{X}_i \mathbf{w}_{j,i} \frac{\mathbf{w}_{j,i}^T \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_i}{\mathbf{w}_{j,i}^T \mathbf{X}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \left\| \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|} \qquad (3.24)$$

The change in the weight $\Delta \mathbf{w}_{j+1,i}$ from equation (3.8) is given by:

$$\Delta \mathbf{w}_{j+1,i} = [\mathbf{X}_i^T \mathbf{X}_i]^{-1} \frac{\mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \left\| \mathbf{X}_i \mathbf{w}_{j,i} \right\|^2}{\left\| \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|^2} - \mathbf{w}_{j,i} \qquad (3.25)$$

Consequently the normalized weight vector for the $(j + 1)^{th}$ iteration is given as:

$$\mathbf{w}_{j+1,i} = \frac{[\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i}}{\left\| [\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i \mathbf{w}_{j,i} \right\|} \qquad (3.26)$$

The above iteration is equivalent to the Power method (Golub and Loan, 1996) for determining the eigenvector corresponding to the largest eigenvalue of the matrix $[\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i$. It is also known (Reinsel and Velu, 1998) that the weight vector $\mathbf{w}_i$ determined by the reduced rank regression is equal to the eigenvector corresponding to the largest eigenvalue of the matrix $[\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{X}_i$.

Theorems 3.2 and 3.3 demonstrate that the error based algorithm proposed by Baffi et al., (1999(a)) is equivalent to classical reduced rank regression for the special case of where a linear relationship is assumed between the scores.

Remark 3.4: Since the error based iteration procedure converges to the RRR solution, it is not guaranteed that the scores vectors, $\mathbf{t}_i$ and $\mathbf{u}_i$, have maximum covariance.

Remark 3.5: The constraint of $\mathbf{v}_i$ being aligned in the same direction as $\mathbf{q}_i$ in the error based cost function results in the residuals of the response variables being minimized instead of the residuals of the i^{th} scores model being minimized.

Corollary 3.1: The non-linear error based PLS algorithm proposed by Baffi et al., (1999(a)) is a non-linear extension of reduced rank regression.

Proof: This follows from Theorems 3.2 and 3.3 and Remark 3.5. The constraint, that the directions of the projection direction, $\mathbf{v}_i$, and the loading vector, $\mathbf{q}_i$, are equivalent, is retained in the algorithm of Baffi et al., (1999(a)) for the non-linear case. Therefore, the consequence is that the maximum amount of variance in the response matrix is explained, which is in keeping with the spirit of (non-linear) reduced rank regression.

3.7 Analysis of the Algorithm of Wold et al., (1989)

Having analyzed the algorithm of Baffi et al., (1999(a)), the algorithm of Wold et al., (1989) is now considered and it is proven that this algorithm attempts to maximize the non-linear covariance function. Before the algorithm is analyzed, a brief overview of the algorithm is provided.

Wold et al., (1989) extended linear PLS by incorporating a non-linear (quadratic) relationship:

$$\mathbf{u}_i = \mathbf{f}(\mathbf{t}_i) = \mathbf{f}(\mathbf{X}_i, \mathbf{w}_i, \mathbf{c}_i) = c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2 + \mathbf{e}_i \tag{3.27}$$

Updating of the weight vector $\mathbf{w}_i$ is performed by linearising $\mathbf{f}(\mathbf{t}_i)$ about the current weight vector $\mathbf{w}_i^0$ (or the t-scores vector $\mathbf{t}_i^0$) and the parameter vector $\mathbf{c}_i^0$ using a first order Taylor series expansion of the quadratic function:

$$\mathbf{u}_i = \mathbf{f}(\mathbf{t}_i^0) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_i}\right|_{\mathbf{w}_i^0} \Delta\mathbf{w}_i + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_i}\right|_{\mathbf{c}_i^0} \Delta\mathbf{c}_i + \mathbf{e}_i \tag{3.28}$$

The increment, $\Delta\mathbf{w}_i$, for the current weight vector $\mathbf{w}_i^0$ is calculated as follows:

1. Define a matrix $\mathbf{Z}_i$ and a vector $\mathbf{d}_i$ as:

$$\mathbf{Z}_i = \left[\left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}_i} \right|_{\mathbf{w}_i^0} \quad \left. \frac{\partial \mathbf{f}}{\partial \mathbf{c}_i} \right|_{\mathbf{c}_i^0} \right]$$

$$\mathbf{d}_i = \begin{bmatrix} \Delta \mathbf{w}_i \\ \Delta \mathbf{c}_i \end{bmatrix}$$

2. Determine the column vector $\mathbf{d}_i$ as:

$$\mathbf{d}_i = \frac{\mathbf{Z}_i^T \mathbf{u}_i}{\mathbf{u}_i^T \mathbf{u}_i}$$

3. Normalize $\mathbf{d}_i$ to unit norm:

$$\mathbf{d}_i = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$$

4. Evaluate a column vector $\mathbf{s}_i$:

$$\mathbf{s}_i = \mathbf{Z}_i \mathbf{d}_i$$

5. Regress $\mathbf{u}_i$ on $\mathbf{s}_i$:

$$b_i = \frac{\mathbf{s}_i^T \mathbf{u}_i}{\mathbf{u}_i^T \mathbf{u}_i}$$

6. The incremental weight vector is then determined as:

$$\Delta \mathbf{w}_i = b_i \mathbf{d}_i(1{:}K)$$

where $\mathbf{d}_i(1{:}K)$ denotes the first K elements of the vector $\mathbf{d}_i$

7. Update the weight vector $\mathbf{w}_i^0$ as:

$$\mathbf{w}_{i+1} = \mathbf{w}_i^0 + \Delta\mathbf{w}_i$$

Wold et al., (1989) stated that their algorithm 'is fairly complicated and converges slowly when the data lack structure'. Baffi et al., (1999(a)) while proposing a 'simplified' version of the algorithm of this algorithm raised the following three questions on the weight updating procedure:

1. Why is the vector $\mathbf{d}_i$ determined as $\mathbf{d}_i = \dfrac{\mathbf{Z}_i^T\mathbf{u}_i}{\mathbf{u}_i^T\mathbf{u}_i}$, that is as if $\mathbf{Z}_i$ is being regressed on $\mathbf{u}_i$ according to $\mathbf{Z}_i = \mathbf{u}_i\mathbf{d}_i^T$ (Step 2 above) instead of $\mathbf{u}_i = \mathbf{Z}_i\mathbf{d}_i$?
2. Why is the vector $\mathbf{d}_i$ scaled by a constant b_i (step 6 above) to determine the incremental weight vector $\Delta\mathbf{w}_i$?
3. Why is the first order differential of the function $\mathbf{c}_i$ included in the matrix $\mathbf{Z}_i$ if the incremental function parameter vector $\Delta\mathbf{c}_i$ is not to be used for updating the weight vector $\mathbf{c}_i$?

However, Baffi et al., (1999) did not address these questions. Answers to these questions are provided in the following lemmas.

Lemma 3.1: The vector $\mathbf{d}_i^T = [\Delta\mathbf{w}_i^T \;\; \Delta\mathbf{c}_i^T]$ in the updating procedure of Wold et al., (1989) is determined based on maximizing the covariance between the u-scores vector, $\mathbf{u}_i$, and the non-linearly transformed t-scores vector, $\mathbf{f}(\mathbf{t}_i)$, with a unit norm constraint on vector $\mathbf{d}_i$.

Proof: The covariance between $\mathbf{u}_i$ and $\mathbf{f}(\mathbf{t}_i)$ is given by:

$$\mathbf{u}_i^T\mathbf{f}(\mathbf{t}_i) = \mathbf{u}_i^T\left(\mathbf{f}(\mathbf{t}_i^0) + \left.\frac{\partial\mathbf{f}}{\partial\mathbf{w}_i}\right|_{\mathbf{w}_i^0}\Delta\mathbf{w}_i + \left.\frac{\partial\mathbf{f}}{\partial\mathbf{c}_i}\right|_{\mathbf{c}_i^0}\Delta\mathbf{c}_i\right) \qquad (3.29)$$

Since the covariance between two random variables does not change if a constant value is added to either of the two variables, $\mathbf{f}(\mathbf{t}_i^0)$ can be removed from the above expression. Consequently equation (3.29) can be written as:

$$\mathbf{u}_i^T \mathbf{f}(\mathbf{t}_i) = \mathbf{u}_i^T \left[\left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}_i} \right|_{\mathbf{w}_i^0} \quad \left. \frac{\partial \mathbf{f}}{\partial \mathbf{c}_i} \right|_{\mathbf{c}_i^0} \right] \begin{bmatrix} \Delta \mathbf{w}_i \\ \Delta \mathbf{c}_i \end{bmatrix} = \mathbf{u}_i^T \mathbf{Z}_i \mathbf{d}_i \tag{3.30}$$

If the objective function for determining $\mathbf{d}_i$ is taken as the maximization of the covariance function given in equation (3.30), subjected to unit norm constraint on $\mathbf{d}_i$

$$\max_{\mathbf{d}_i} (\mathbf{d}_i \mathbf{Z}_i^T \mathbf{u}_i) \quad \text{subjected to } \|\mathbf{d}_i\| = 1 \tag{3.31}$$

then the solution to the above cost function is given by the conventional (linear) PLS solution for one response variable, i.e. PLS1:

$$\mathbf{d}_i = \frac{\mathbf{Z}_i^T \mathbf{u}_i}{\mathbf{u}_i^T \mathbf{u}_i} \tag{3.32}$$

This equation for determining $\mathbf{d}_i$ is the same as that used in the algorithm of Wold et al., (1989) (step 2 in the summary given above)

Lemma 3.2: The incremental function parameter vector $\Delta \mathbf{c}_i$ is a vector of slack variables, (that is, the variables that are used to optimize the objective function but that are not used in the model) for the covariance maximization and is used to guarantee the convergence of the algorithm.

Proof: Since $\Delta \mathbf{c}_i$, determined by solving the optimization problem in equation (3.31) is not used to update $\mathbf{c}_i$, it is clearly a slack variable. If $\Delta \mathbf{c}_i$ is not included in the optimization problem, the incremental weight vector $\Delta \mathbf{w}_i$ will always be of unit norm since $\mathbf{d}_i$ is constrained to be of unit norm. The consequence of this is that the algorithm will not

converge since $\Delta \mathbf{w}_i$ cannot tend to zero. Inclusion of $\Delta \mathbf{c}_i$, therefore enables the algorithm to converge so that after convergence $\Delta \mathbf{w}_i = \mathbf{0}$ and $\Delta \mathbf{c}_i$ is of unit norm so that the constraint of unit norm on $\mathbf{d}_i$ is satisfied.

It can therefore be concluded from the two lemmas that the parameters in the algorithm by Wold et al., (1989) are adjusted using a covariance maximization criterion.

Although the weight vectors $\mathbf{w}_i$ in the algorithm of Wold et al., (1989) are obtained such that the covariance between the u-and the non-linearly transformed t-scores is maximized, the algorithm suffers from the following limitations:

1. The quadratic function parameter vector, $\mathbf{c}_i$, is determined so as to minimize the prediction error of the inner scores model. However, $\mathbf{c}_i$ also influences the covariance between the u-scores $\mathbf{u}_i$ and the non-linearly transformed t-scores, $\mathbf{f}(\mathbf{t}_i)$. It is therefore, necessary to determine $\mathbf{c}_i$ along with $\mathbf{w}_i$ to maximize the covariance.
2. Wold et al., (1989) proposed the use of a first order Taylor series expansion for it to align with the iterative framework of linear PLS. In each iteration, a constraint is placed on $\Delta \mathbf{w}_i$ by placing a unit norm constraint on $\mathbf{d}_i$ and also on the updated weight vector $\mathbf{w}_i$ by normalizing it to unit norm. Since $\mathbf{w}_i$ is constrained to be of unit norm, there is no need to impose any constraint on $\Delta \mathbf{w}_i$ or $\mathbf{d}_i$.

The above problems associated with the algorithm of Wold et al., (1989) are overcome in the two non-linear PLS algorithms proposed in section 3.9. Prior to discussing the new algorithms, a classification of existing non-linear PLS algorithms is undertaken.

3.8 Classification of Existing Non-linear PLS Algorithms

This section categorizes the non-linear PLS (NLPLS) algorithms described in the literature into three categories (1) covariance based (2) quick and dirty algorithms and (3) error based.

3.8.1 Covariance based Non-linear PLS

The algorithm by Wold et al., (1989) belongs to this category but has a number of limitations as discussed in section 3.7.

3.8.2 Quick and Dirty Methods

This category includes those algorithms whereby linear PLS is used to determine the t-and u-scores prior to a non-linear model being fitted between the t- and u-scores. The algorithms of Frank (1990), Qin et al., (1992), Wilson et al., (1997) are members of this category.

These algorithms clearly do not represent the true non-linear PLS as the outer weights of the algorithm are not determined as per the non-linearity in the data.

3.8.3 Error Based Non-linear PLS Algorithms

This category of algorithms obtain scores variables that are projected onto the lines as in linear PLS, and a non-linear relationship is fitted between the corresponding pair of t- and u-scores. The parameters (the outer weights as well as the inner non-linear model parameters) are simultaneously updated and are determined so as to minimize the prediction error of the inner model. The algorithms of Hiden et al., (1998), Baffi et al., (1999), and Li et al., (2001) are examples of this approach.

It should be noted, however, that the minimization of the prediction error in the inner scores model does not guarantee the maximization of the covariance between the t-and u-scores. As analysed in section 3.7, these algorithms are in fact, a non-linear version of reduced rank regression and therefore do not represent a true non-linear representation of PLS.

The categorization of the key non-linear PLS algorithms mentioned above is summarized in the Table 3.2.

Table 3.2: Categorization of the proposed non-linear PLS algorithms

Category		
Covariance Based	**Quick and dirty**	**Error based**
Wold et al., (1989) Wold (1992)	Frank (1990) Qin et al., (1992) Wilson et al., (1997) Patwardhan et al., (1998)	Hiden et al., (1998) Baffi et al., (1999) Li et al., (2001) Min et al., (2002)

3.9 Non-linear Partial Least Squares using Covariance Maximization

PLS is based on the maximization of a linear covariance function (Höskuldsson, 1988) between the t-scores and u-scores. This criterion not only provides a straightforward calculation of the scores vectors and the model parameters, it also offers statistical interpretation which is helpful in understanding the underlying structure of the system. Maximization of the covariance function determines scores variables that are a statistical compromise between the approximations of the predictor (process) variables and the prediction of the response (quality) variables. This is in contrast to other regression techniques such as Multiple Linear Regression (MLR) and Reduced Rank Regression (RRR)) whose objective is to predict the response variables from the predictor variables by minimizing the prediction error. It has been demonstrated that maximising the covariance between pairs of scores variables is of benefit in applications such as multivariate statistical process performance monitoring (MacGregor et al., 1991; Martin et al., 1996). In addition, a number of comparisons have identified the benefits of PLS in terms of deriving and encapsulating important qualitative information from chemical data (Haaland and Thomas, 1988(a); 1988(b); Martens and Næs, 1989; Næs et al., 1986; Næs and Martens, 1985; Wold, et al., 1983(a); 1983(b); Wold et al., 1984).

Considering the importance of covariance maximization in linear PLS, any 'true' non-linear PLS should be based on the maximization of the 'non-linear covariance function' which reduces to a (linear) covariance function when the non-linear function is replaced by linear function. The non-linear covariance function is defined as:

$$J_{NLPLS} = Cov(\mathbf{f}(\mathbf{t}_i), \mathbf{u}_i) = Cov(\mathbf{f}(\mathbf{X}_i\mathbf{w}_i, \mathbf{Y}_i\mathbf{v}_i)) \quad (3.33)$$

where *Cov* is the usual covariance function of the two vectors, **f** is a non-linear function which is assumed to be quadratic:

$$\tau_i = \mathbf{f}(\mathbf{t}_i) = c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2 \qquad (3.34)$$

It can be seen that this definition of the non-linear covariance function is a generalization of the usual covariance function since if the non-linear function is replaced by a linear function, the definition reduces to the conventional covariance function.

Based on the definition of the non-linear covariance function in equation (3.34), the objective function of non-linear PLS is to determine the weight vectors $\mathbf{v}_i$ and $\mathbf{w}_i$, and the scores vectors such that the non-linear covariance function J_{NLPLS} is maximized. By selecting a quadratic non-linear function **f**, it is not only the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$ that are to be optimized but also the parameters $\mathbf{c}_i$ of the function **f**. The objective function of non-linear PLS can be stated as:

$$J_{NLPLS} = \max_{\mathbf{w}_i,\mathbf{v}_i,\mathbf{c}_i} Cov(\mathbf{f}(\mathbf{c}_i,\mathbf{X}_i\mathbf{w}_i), \mathbf{Y}_i\mathbf{v}_i) \qquad (3.35)$$

It should be noted, however, that in the objective function given above, it is necessary to introduce constraints on the magnitude of the parameter vector $\mathbf{c}_i$ in addition to having a constraint of unity norm on the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$, as otherwise J_{NLPLS} will be unbounded. This gives rise to two non-linear PLS algorithms depending on how the parameter vector $\mathbf{c}_i$ is constrained. The two algorithms are detailed below.

3.9.1 Non-linear PLS Algorithm Number 1 (NLPLS1)

In the first version of the algorithm, the parameter vector $\mathbf{c}_i$ like the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$, is constrained to have unit norm. This algorithm is termed NLPLS1 and the objective function is given by:

$$J_{NLPLS1} = \max_{\mathbf{w}_i, \mathbf{v}_i, \mathbf{c}_i} \mathbf{f}(\mathbf{t}_i)^T \mathbf{u}_i$$

subject to

$$\|\mathbf{w}_1\| = 1 \quad \|\mathbf{v}_i\| = 1 \quad \|\mathbf{c}_i\| = 1$$

(3.36)

3.9.2 Non-linear PLS Algorithm Number 2 (NLPLS2)

In the second version of the algorithm, the magnitude of the parameter vector is indirectly constrained by placing a constraint on the magnitude of the function **f**. Since non-linear PLS should be a generalization of linear PLS, the constraint selected is that the length of **f** is the same as that for **t**:

$$\|\boldsymbol{\tau}_i\| = \|\mathbf{f}(\mathbf{t}_i)\| = \|\mathbf{t}_i\| \tag{3.37}$$

The algorithm is denoted as NLPLS2 and its objective function is as follows:

$$J_{NLPLS2} = \max_{\mathbf{w}_i, \mathbf{v}_i, \mathbf{c}_i} \mathbf{f}(\mathbf{t}_i)^T \mathbf{u}_i$$

subject to

$$\|\mathbf{w}_i\| = 1 \quad \|\mathbf{v}_i\| = 1$$

$$\|\boldsymbol{\tau}_i\| = \|\mathbf{f}(\mathbf{t}_i)\| = \|\mathbf{t}_i\|$$

(3.38)

The optimization functions for both non-linear PLS algorithms can be optimized using gradient ascent algorithm. The gradients of the objective function with respect to parameters for NLPLS1 (which are also equal to NLPLS2) can be computed as follows:

$$\frac{\partial J_{NLPLS1}}{\partial \mathbf{w}_i} = \mathbf{X}_i^T((c_{1,i} + 2c_{2,i}\mathbf{t}_i)^T \mathbf{u}_i)$$

$$\frac{\partial J_{NLPLS1}}{\partial \mathbf{c}_i} = ([1 \quad \mathbf{t} \quad \mathbf{t}_i^2]^T \mathbf{u}_i)$$

$$\frac{\partial J_{NLPLS1}}{\partial \mathbf{v}_i} = \mathbf{Y}_i^T[c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2]$$

(3.39)

Once the gradients are computed, the objective function can be optimized by updating the parameters until convergence using the following equations:

$$\mathbf{c}_i\,(n+1) = \mathbf{c}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{c}_k(n)} \qquad (3.40)$$

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{w}_i(n)}$$

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{v}_i(n)}$$

where η is the learning rate. The constraints on the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$ and the parameter vector $\mathbf{c}_i$ are taken into consideration after each updating. For example, for the NLPLS1 algorithm, the updating equations are:

$$\tilde{\mathbf{c}}_i\,(n+1) = \tilde{\mathbf{c}}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{c}_i(n)} \qquad (3.41)$$

$$\mathbf{c}_i\,(n+1) = \frac{\tilde{\mathbf{c}}_i(n+1)}{\|\tilde{\mathbf{c}}_i(n+1)\|}$$

$$\tilde{\mathbf{w}}_i(n+1) = \mathbf{w}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{w}_i(n)}$$

$$\mathbf{w}_i(n+1) = \frac{\tilde{\mathbf{w}}_i}{\|\tilde{\mathbf{w}}_i\|}$$

$$\tilde{\mathbf{v}}_i(n+1) = \mathbf{v}_i(n) + \eta \frac{\partial J_{NLPLS\ 1}}{\partial \mathbf{v}_i(n)}$$

$$\mathbf{v}_i(n+1) = \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|}$$

For the NLPLS2 algorithm, the constraints are implemented as follows:

$$\mathbf{t}_i = \mathbf{X}_i\mathbf{w}_i \qquad (3.42)$$

$$\mathbf{u}_i = \mathbf{Y}_i\mathbf{v}_i$$

$$\boldsymbol{\tau}_i = \mathbf{f}(\mathbf{t}_i) = c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2$$

$$\boldsymbol{\tau}_i = \frac{\boldsymbol{\tau}_i\|\mathbf{t}\|}{\|\boldsymbol{\tau}_i\|}$$

The optimization method given in equation (3.41) uses a gradient based method and may be slow in convergence. To increase the speed of convergence, second order methods (e.g Newton's method) can be used.

After the parameters have been optimized, a linear relationship between $\mathbf{f}(\mathbf{t}_i)$ and $\mathbf{u}_i$ can be fitted:

$$\mathbf{u}_i = b_{1,i}\mathbf{f}(\mathbf{t}_i) + b_{0,i} + \mathbf{e}_i \qquad (3.43)$$

The parameters, $b_{1,i}$ and $b_{0,i}$ can be determined using ordinary least squares. The prediction of the response variables can now be calculated from the predicted u-scores, $\hat{\mathbf{u}}_i$ by calculating the loading vector $\mathbf{q}_i$:

$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \hat{\mathbf{u}}_i} \qquad (3.44)$$

The prediction of the response variables is given by:

$$\hat{\mathbf{Y}} = \hat{\mathbf{u}}_i \mathbf{q}_i^T \qquad (3.45)$$

After the calculation of the first latent variable, the percentage variance explained by this latent variable is calculated and in case more latent variables are needed, the above procedure is repeated by deflating the matrices **X** and **Y** as in conventional PLS.

3.10 Summary of the Algorithms

The two algorithms, NLPLS1 and NLPLS2 are summarized below.

3.10.1 NLPLS1 Algorithm

Given: Input matrix **X** and output matrix **Y**.

Mean centre and auto scale the two matrices. Set $i = 1$ and $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{Y}_1 = \mathbf{Y}$

Step 1: Initialize the weight vectors $\mathbf{w}_i$, $\mathbf{v}_i$ and non-linear function parameter $\mathbf{c}_i$ to random values.

Step 2: Compute t-and u-scores

$$\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$$
$$\mathbf{u}_i = \mathbf{Y}_i \mathbf{v}_i$$

Step 3: Compute the gradients

$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{w}_i} = \mathbf{X}_i^T((c_{1,i} + 2c_{2,i}\mathbf{t}_i)^T \mathbf{u}_i)$$
$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{c}_i} = ([\mathbf{1} \quad \mathbf{t} \quad \mathbf{t}_i^2]^T \mathbf{u}_i)$$
$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{v}_i} = \mathbf{Y}_i^T[c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2]$$

Step 4: Update the parameters

$$\tilde{\mathbf{c}}_i\,(n+1) = \tilde{\mathbf{c}}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{c}_i(n)}$$
$$\mathbf{c}_i\,(n+1) = \frac{\tilde{\mathbf{c}}_i(n+1)}{\|\tilde{\mathbf{c}}_i(n+1)\|}$$
$$\tilde{\mathbf{w}}_i(n+1) = \mathbf{w}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{w}_i(n)}$$
$$\mathbf{w}_i(n+1) = \frac{\tilde{\mathbf{w}}_i}{\|\tilde{\mathbf{w}}_i\|}$$
$$\tilde{\mathbf{v}}_i(n+1) = \mathbf{v}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{v}_i(n)}$$
$$\mathbf{v}_i(n+1) = \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|}$$

Step 5: Repeat steps 2 and 3 until convergence

Step 6: Fit a linear relationship between the u-scores $\mathbf{u}_i$ and the non-linearly transformed t-scores $\mathbf{f}(\mathbf{t}_i)$:

$$\mathbf{u}_i = b_{1,i}\mathbf{f}(\mathbf{t}_i) + b_{0,i} + \mathbf{e}_i$$

Step 7: Calculate predicted u-scores $\hat{\mathbf{u}}_i$

$$\hat{\mathbf{u}}_i = b_{1,i}\mathbf{f}(\mathbf{t}_i) + b_{0,i}$$

Step 8: Determine the loading vectors $\mathbf{p}_i$ and $\mathbf{q}_i$

$$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$$
$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \hat{\mathbf{u}}_i}$$

Step 9: Deflate the matrices

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$$
$$\mathbf{Y}_{i+1} = \mathbf{Y}_i - \hat{\mathbf{u}}_i \mathbf{q}_i^T$$

Step 10: Repeat steps 1 to 8, with $i = i + 1$ if another latent variable is required.

3.10.2 NLPLS2 Algorithm

Given: Input matrix **X** and output matrix **Y**.

Mean centre and auto scale the two matrices. Set $i = 1$ and $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{Y}_1 = \mathbf{Y}$

Step 1: Initialize the weight vectors $\mathbf{w}_i$, $\mathbf{v}_i$ and non-linear function parameter $\mathbf{c}_i$ to random values

Step 2: Compute t-and u-scores

$$\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$$
$$\mathbf{u}_i = \mathbf{Y}_i \mathbf{v}_i$$
$$\boldsymbol{\tau}_i = \mathbf{f}(\mathbf{t}_i) = c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2$$
$$\boldsymbol{\tau}_i = \frac{\boldsymbol{\tau}_i \|\mathbf{t}\|}{\|\boldsymbol{\tau}_i\|}$$

Step 3: Compute the gradients

$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{w}_i} = \mathbf{X}_i^T((c_{1,i} + 2c_{2,i}\mathbf{t}_i)^T \mathbf{u}_i)$$
$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{c}_i} = ([\mathbf{1} \quad \mathbf{t} \quad \mathbf{t}_i^2]^T \mathbf{u}_i)$$
$$\frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{v}_i} = \mathbf{Y}_i^T[c_{0,i} + c_{1,i}\mathbf{t}_i + c_{2,i}\mathbf{t}_i^2]$$

Step 4: Update the parameters

$$\mathbf{c}_i\,(n+1) = \mathbf{c}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{c}_i(n)}$$
$$\tilde{\mathbf{w}}_i(n+1) = \mathbf{w}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{w}_i(n)}$$
$$\mathbf{w}_i(n+1) = \frac{\tilde{\mathbf{w}}_i}{\|\tilde{\mathbf{w}}_i\|}$$
$$\tilde{\mathbf{v}}_i(n+1) = \mathbf{v}_i(n) + \eta \frac{\partial J_{\text{NLPLS 1}}}{\partial \mathbf{v}_i(n)}$$
$$\mathbf{v}_i(n+1) = \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|}$$

Step 5: Repeat steps 2 and 3 until convergence

Step 6: Fit a linear relationship between u-scores $\mathbf{u}_i$ and the non-linearly transformed t-scores $\mathbf{f}(\mathbf{t}_i)$:

$$\mathbf{u}_i = b_{1,i}\mathbf{f}(\mathbf{t}_i) + b_{0,i} + \mathbf{e}_i$$

Step 7: Calculate the predicted u-scores $\hat{\mathbf{u}}_i$

$$\hat{\mathbf{u}}_i = b_{1,i}\mathbf{f}(\mathbf{t}_i) + b_{0,i}$$

Step 8: Determine the loading vectors $\mathbf{p}_i$ and $\mathbf{q}_i$

$$\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$$

$$\mathbf{q}_i = \frac{\mathbf{Y}_i^T \hat{\mathbf{u}}_i}{\hat{\mathbf{u}}_i^T \hat{\mathbf{u}}_i}$$

Step 9: Deflate the matrices

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$$

$$\mathbf{Y}_{i+1} = \mathbf{Y}_i - \hat{\mathbf{u}}_i \mathbf{q}_i^T$$

Step 10: Repeat steps 1 to 8, with $i = i + 1$ if another latent variable is required

3.11 Application Studies

The algorithms given above are now tested on three data sets, two artificial data sets and a simulation of a pH neutralization process.

3.11.1 Example 1

In this example, data from a non-linear function described by Cherkassky et al., (1996) and also used by Baffi et al., (1999(a)) forms the basis of study. The function has four uncorrelated random inputs which are uniformly distributed in the interval [-0.25 0.25]. The single output variable is related to the input variables as:

$$y = \exp(2x_1 \sin(\pi x_4)) + \sin(x_2 x_3) \qquad (3.46)$$

$$x_i \in [-0.25 \;\; 0.25] \quad \text{for } i = 1,2,3,4$$

A data set of 800 samples was generated and was divided into a training data set (600 samples) and a validation data set (200 samples). After the training data set was auto-scaled, the NLPLS1 and NLPLS2 algorithms with a quadratic function as the inner non-linear function were applied to identify non-linear models. The performance of the NLPLS1 and NLPLS2 algorithms on the training data set was assessed using the percentage contribution of each latent variable and their cumulative percentage contribution to the predictor and response matrices. Furthermore, the Mean Square Prediction Error (MSPE) for the training and validation data sets were also calculated. The quantitative performance of the NLPLS1 and NLPLS2 algorithms, as evaluated by these performance indices is summarized in Tables 3.3 and 3.4 respectively. The qualitative performance of the model is shown in Figures 3.1 and 3.3 in term of plots of the measured and predicted values of the response using four latent variables. The corresponding time series plots of the residuals for the two algorithms are shown in Figures 3.2 and 3.4 respectively. To compare the performance of NLPLS1 and NLPLS2, the performances of linear PLS, non-linear PLS algorithm of Wold et al., (1989) and non-linear PLS algorithm of Baffi et al., (1999(a)) using a quadratic non-linearity are summarized in Tables 3.5, 3.6 and 3.7 respectively.

Table 3.3: Performance of NLPLS1 algorithm (example 1)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	26.56	26.56	76.29	76.29	0.2367	0.2228
2	24.17	50.73	16.68	92.98	0.0701	0.0692
3	25.80	76.53	1.90	94.88	0.0511	0.0513
4	23.47	100.00	0.43	95.31	0.0469	0.0475

Table 3.4: Performance of NLPLS2 algorithm (example 1)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	26.37	26.37	77.89	77.89	0.2207	0.2093
2	24.02	50.39	15.97	93.86	0.0613	0.0609
3	26.00	76.39	1.90	95.76	0.0423	0.0457
4	23.61	100.00	0.43	96.19	0.0380	0.0392

Table 3.5: Performance of linear PLS algorithm (example 1)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	27.40	27.40	0.52	0.52	1.07	0.8493
2	25.82	53.22	0.00	0.52	1.07	0.8493
3	22.52	75.74	0.00	0.52	1.07	0.8493
4	24.26	100.00	0.00	0.52	1.07	0.8493

Table 3.6: Performance of Wold et al., (1989) algorithm (example 1)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	26.72	26.72	74.22	74.22	0.2574	0.2437
2	24.48	51.20	2.65	76.87	0.2309	0.2424
3	25.10	76.30	4.90	81.77	0.1819	0.1892
4	23.70	100.0	0.63	82.40	0.1757	0.1856

Table 3.7: Performance of Baffi et al., (1999(a)) algorithm (example 1)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	26.26	26.26	78.05	78.05	0.2091	0.2082
2	23.93	50.19	15.80	93.86	0.0613	0.0608
3	26.10	76.33	1.91	95.77	0.0423	0.0457
4	23.67	100.0	0.41	96.17	0.0382	0.0392

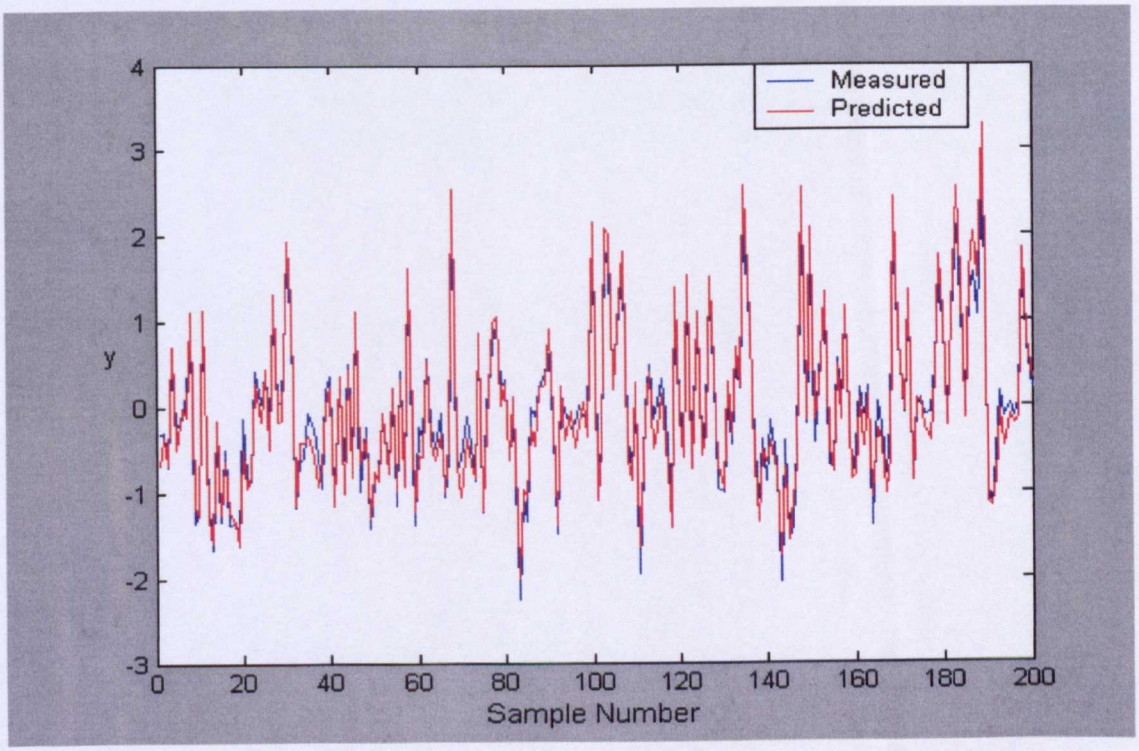


Figure 3.1: Prediction of response variable using NLPLS1 algorithm (example 1)

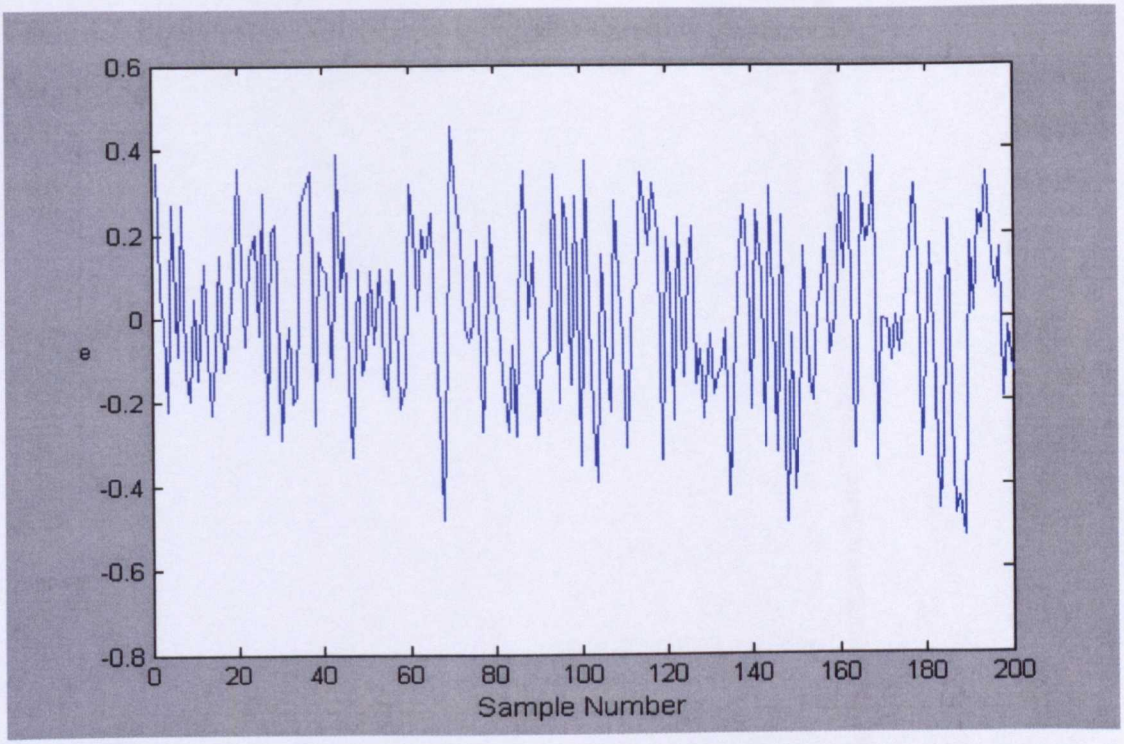


Figure 3.2: Time series plot of the residuals using NLPLS1 algorithm (example 1)

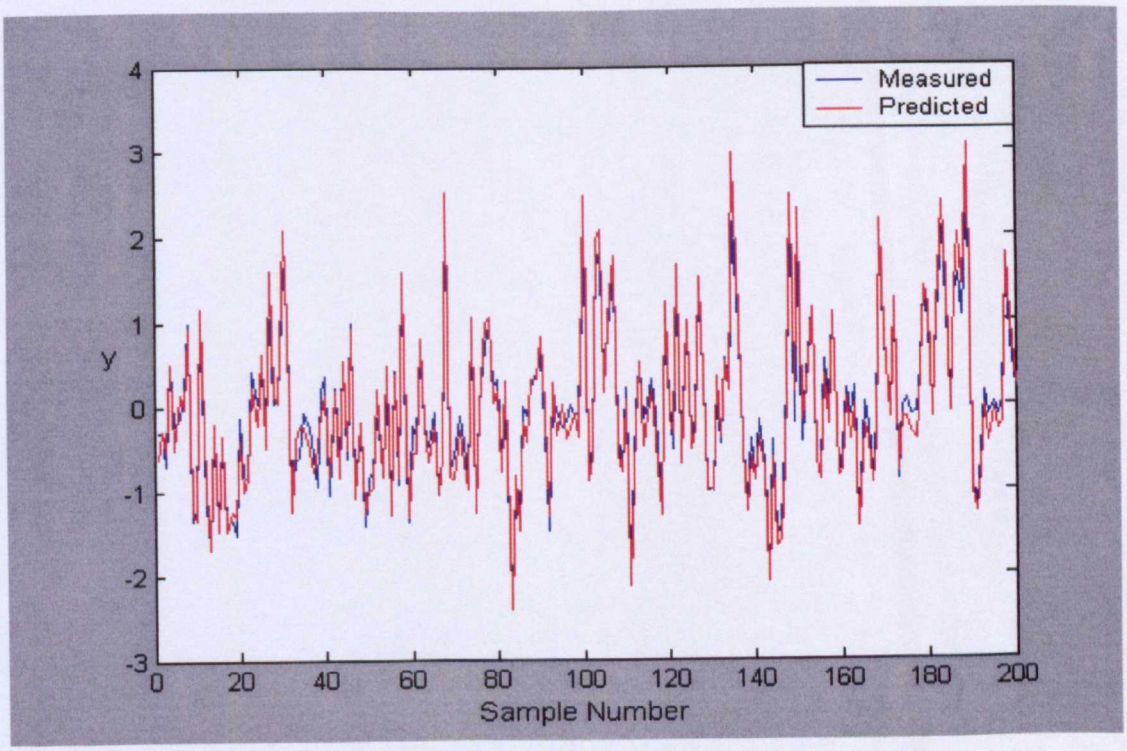


Figure 3.3: Prediction of response variable using NLPLS2 algorithm (example 1)

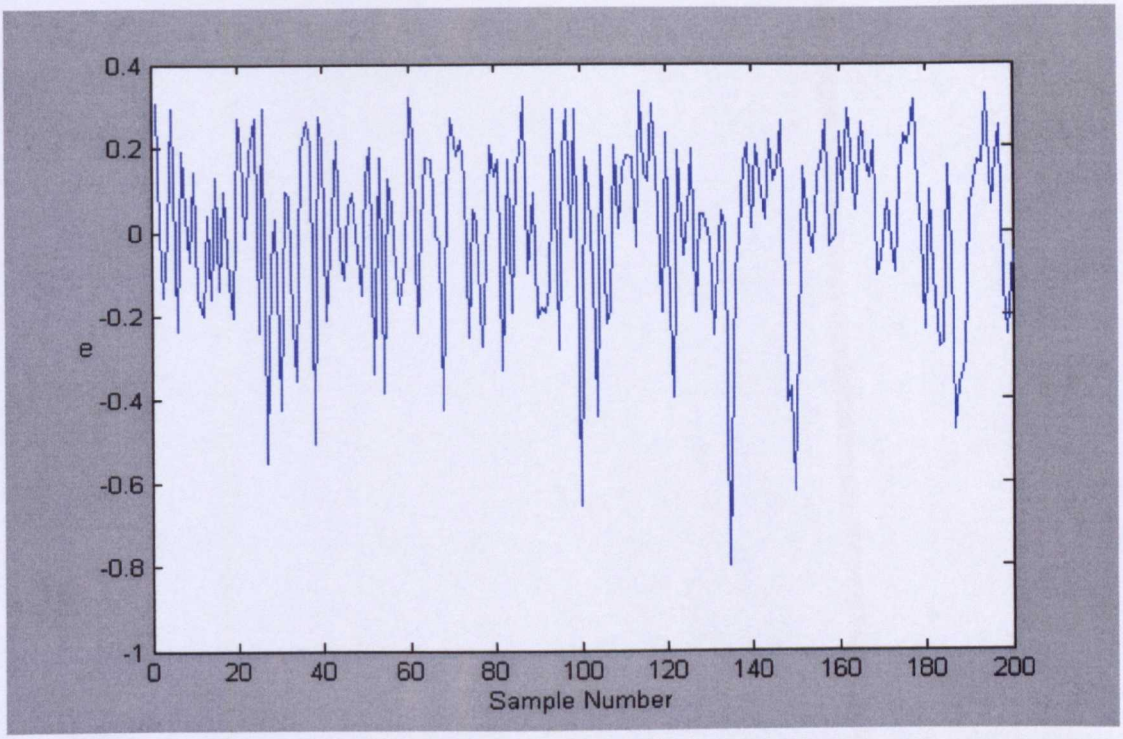


Figure 3.4: Time series plot of the residuals using NLPLS2 algorithm (example 1)

The following observations can be made by comparing the performances of linear PLS, the algorithm of Wold et al., (1989), the algorithm of Baffi et al., (1999(a)) and the NLPLS1 and NLPLS2 algorithms.

1. The percentage variance of **X** explained is approximately equal for each of the four latent variables. This is true for all algorithms considered. The reason for this is that the four input variables are uncorrelated and each of them is uniformly distributed over the same interval. Each of the four directions in the input space, therefore account for equal variance.
2. All four latent variables in linear PLS explain only 0.52% of the variance of **Y**, and therefore, is unable to model the data. This is understandable as the response variable, y, is a sum of an exponential and sine function of the input variables and therefore a non-linear model is required to explain y as a function of the input variables
3. The application of the algorithm of Wold et al., (1989) with a quadratic inner non-linearity improves the model. The percentage variance of **Y** explained by the first latent variable, for example, increases from 0.52% in the linear PLS model to 74.22%. The second and higher order latent variables, however, do not contribute

significantly to the variance of **Y**. The improvement in the model identified by the algorithm of Wold et al., (1989) from that identified by linear PLS is also reflected in the lower values of mean square prediction error (MSPE) for both the training and validation data sets.

4. The NLPLS1 and NLPLS2 algorithms with a quadratic inner non-linearity further improve the non-linear model for the data. Application of NLPLS1 and NLPLS2 algorithms not only increases (by about 4% from the algorithm of Wold et al., (1989)) the contribution of the first latent variable to **Y**, but the contribution of the second latent variable is increased by a significant amount. If a non-linear PLS model is built using two latent variables, then the resulting MSPE for the training and validation data sets are much lower for the model identified using NLPLS1 and NLPLS2 algorithms than that identified by the algorithm of Wold et al., (1989). The poor performance of the algorithm of Wold et al., (1989) can be explained, as mentioned in section 3.7, by the fact that while maximizing the covariance between the u-scores and the non-linearly transformed t-scores, all the parameters, in particular the non-linear function parameters, that influence this covariance function are not optimized. In the NLPLS1 and NLPLS2 algorithms, on the other hand, all the parameters (outer weights and inner model parameters) are determined such that the non-linear covariance function is maximized leading to better performance of the algorithms.
5. NLPLS2 performs slightly better than NLPLS1. This may be due to the fact that a unit norm constraint on the non-linear function parameter in NLPLS1 is more severe than the constraint of the non-linearly transformed t-scores having the same norm as that of the t-scores in NLPLS2.
6. The algorithm of Baffi et al., (1999(a)) performs better than all the other algorithms considered in terms of percentage variance of **Y** explained for a given set of latent variables and mean square prediction error on the training and validation data sets for this algorithm. The performance of the NLPLS2 algorithm is very close to this algorithm. The first latent variable, for example, in the algorithm of Baffi et al., (1999(a)) explains 78.05 % variance of **Y** which is slightly higher than the corresponding value (77.89 %) explained in the NLPLS2 algorithm. The second and higher latent variables, however, explains the same amount of variance of **Y** in the NLPLS2 and the algorithm of Baffi et al., (1999(a)). The better predictive capability of the algorithm of Baffi et al., (1999(a)) can be explained from the fact that parameters of the model in this algorithm are determined so as to minimize the prediction error in the response variables.

3.11.2 Example 2

In this example the input matrix **X** is assumed to have a latent structure, that is first latent variables are generated and then the measured variables are generated as a function of the latent variables:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \mathbf{t}_3\mathbf{p}_3^T \qquad (3.47)$$

where $\mathbf{t}_1$, $\mathbf{t}_2$ and $\mathbf{t}_3$ contain one thousand independent observations drawn from a normal distribution with zero mean and unit variance. The three columns, $\mathbf{t}_i$, are mutually independent and the vectors $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ form a set of orthonormal vectors in R^5. The matrix **X**, therefore, comprises five (measured) variables and 1000 data points. An augmented matrix $\mathbf{X}_{aug}$ of **X** is generated by including the squares and cross terms of the original **X** so that $\mathbf{X}_{aug}$ is of order (1000×20). A regression matrix **B** of order (20×3) is now generated and 3 output variables with 1000 observations are calculated using the augmented matrix as follows:

$$\mathbf{Y} = \mathbf{X}_{aug}\mathbf{B} \qquad (3.48)$$

The data set is divided into a training data set consisting of 800 data points that is used for model identification with the remaining 200 data points forming the data set for model validation.

After the training data set is mean centred and scaled to unit variance, the NLPLS1 and NLPLS2 algorithms with a quadratic function as the inner non-linear function are applied to identify a non-linear model for the data set. The performance of the NLPLS1 and NLPLS2 algorithms was measured using the same performance indices as in example 1. The numerical values of the performance indices for the NLPLS1 and NLPLS2 algorithms is summarized in Tables 3.8 and 3.9 respectively and Figures 3.5 and 3.7 show the prediction of the response variables using the NLPLS1 and NLPLS2 algorithms respectively with five latent variables retained in the model. The corresponding time series plots of the residuals for the two algorithms are shown in Figures 3.6 and 3.8 respectively. The performances of linear PLS,

algorithm of Wold et al., (1989) and the algorithm of Baffi et al., (1999(a)) are given in Tables 3.9, 3.10 and 3.11 respectively.

1. All four algorithms require three latent variables to explain 100% variance of **X**. This is because there are three latent variables that generate the matrix **X**.
2. NLPLS1 and NLPLS2 algorithms explain a higher percentage of variance of **Y** for a given set of latent variables. For example, for two latent variables, the percentage variance of **Y** explained for the NLPLS1 and NLPLS2 algorithms are approximately 80% and 85% respectively whereas the corresponding figures for linear PLS and the algorithm of Wold et al., (1989) are approximately 6% and 57% respectively.
3. NLPLS1 and NLPLS2 give lower values of mean squares prediction errors for the training and validation data sets. For two latent variables, for example, the MSPE on the validation data set for the NLPLS1 and NLPLS2 algorithms are 0.65 and 0.49 respectively and the corresponding figures for linear PLS and the algorithm of Wold et al., (1989) are 3.03 and 1.33 respectively. The reasons for better performance of NLPLS1 and NLPLS2 as compared to the algorithm of Wold et al., (1989) are as given in example 1.
4. Also as noted in example 1, NLPLS2 performs slightly better than NLPLS1 in terms of prediction ability of response variables.
5. As far as the prediction ability of response variables is concerned, the algorithm of Baffi et al., (1999(a)) is the best among all the algorithms considered. However, NLPLS1 and NLPLS2 (and also the linear PLS and the algorithm of Wold et al., (1989)) performs better than Baffi et al., (19999(a))'s algorithm in terms of approximation ability of input variables (**X**) for a given set of latent variables. For example, the percentage variance of **X** explained by two latent variables in the NLPLS1 algorithm is 90.18 % which is higher than the corresponding value (77.86 %) in the Baffi et al., (1999(a)) algorithm. This can be explained by the fact that the latent variables in the Baffi et al., (1999(a)) algorithm are determined so as to minimize the prediction error of response variables without any consideration for the approximation of input variables. The latent variables in the NLPLS1 and NLPLS2 algorithms on the other hand are determined so as to maximize the covariance between the t-and u-scores resulting in a compromise between the predictive ability of response variables and the approximation of input variables. Since the compromise between the prediction of response variables and the approximation of input variables is at the heart of conventional linear PLS algorithm, this example illustrates that NLPLS1 and NLPLS2 represent the 'true' non-linear extension of linear PLS.

Table 3.8: Performance of NLPLS1 algorithm (example 2)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	54.02	54.02	45.24	45.58	1.6305	1.4240
2	36.16	90.18	34.95	80.19	0.6552	0.6140
3	9.82	100.00	2.33	82.53	0.5772	0.5583
4	0.00	100.00	0.01	82.54	0.5719	0.5583
5	0.00	100.00	0.00	82.54	0.5718	0.5583

Table 3.9: Performance of NLPLS2 algorithm (example 2)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	53.67	53.67	46.64	46.64	1.5988	1.3875
2	28.50	82.17	38.88	85.02	0.4488	0.4925
3	17.83	100.00	2.31	87.33	0.3796	0.4115
4	0.00	100.00	0.09	87.42	0.3769	0.4112
5	0.00	100.00	0.00	87.42	0.3767	0.4109

Table 3.10: Performance of linear PLS algorithm (example 2)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	54.01	54.01	4.51	4.51	2.8512	3.0559
2	31.23	85.25	1.45	5.95	2.8054	3.0312
3	14.75	100.00	0.05	6.01	2.8037	3.0340
4	0.00	100.00	0.00	6.01	2.8037	3.0133
5	0.00	100.00	0.00	6.01	2.8037	3.0133

Table 3.11: Performance of Wold et al., (1989) algorithm (example 2)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	54.3448	54.3448	42.9243	42.943	1.7101	1.5509
2	31.4290	85.7738	14.2637	57.1879	1.2828	1.3390
3	14.2262	100.00	8.3476	65.5356	1.0326	1.2836
4	0.00	100.0	0.3112	65.8466	1.0233	1.2798
5	0.00	100.0	0.0588	65.9096	1.0216	1.2744

Table 3.12: Performance of Baffi et al., (1999(a)) algorithm (example 2)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	53.01	53.01	47.12	47.12	1.5845	1.3632
2	24.85	77.86	41.62	88.73	0.3376	0.3923
3	22.13	99.99	2.79	91.52	0.2540	0.2873
4	0.01	100.0	0.06	91.59	0.2521	0.2882
5	0.00	100.0	0.01	91.59	0.2519	0.2890

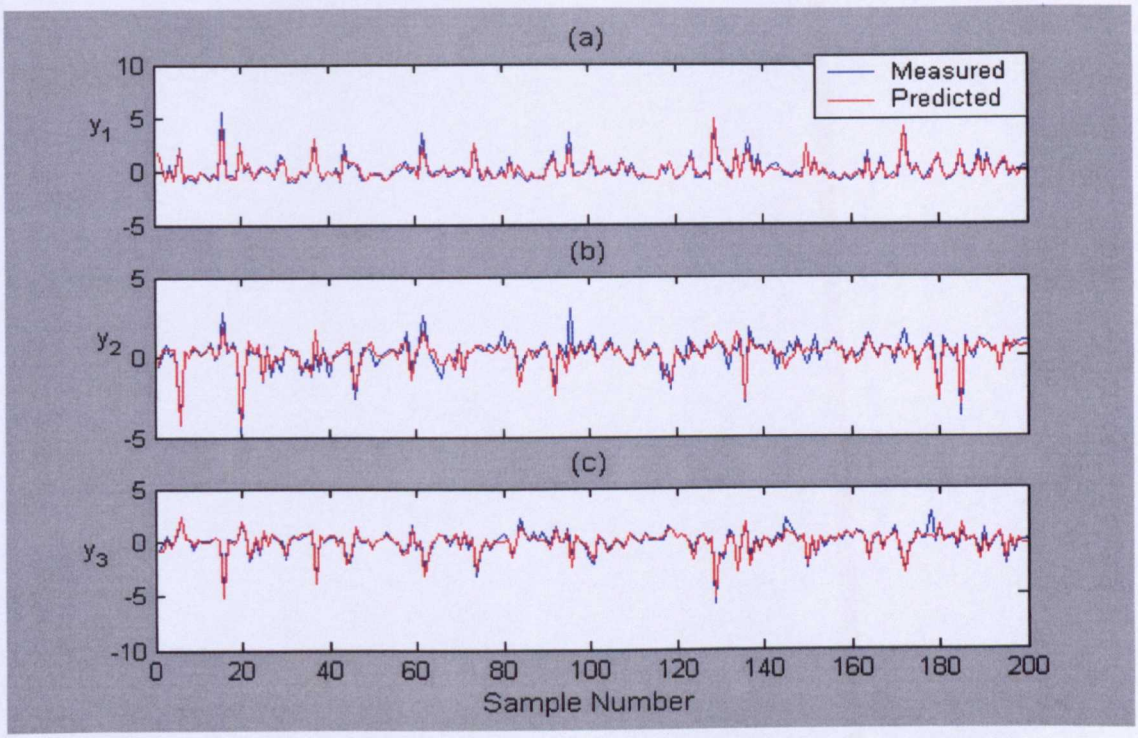


Figure 3.5: Prediction of response variables using NLPLS1 algorithm (example 2).

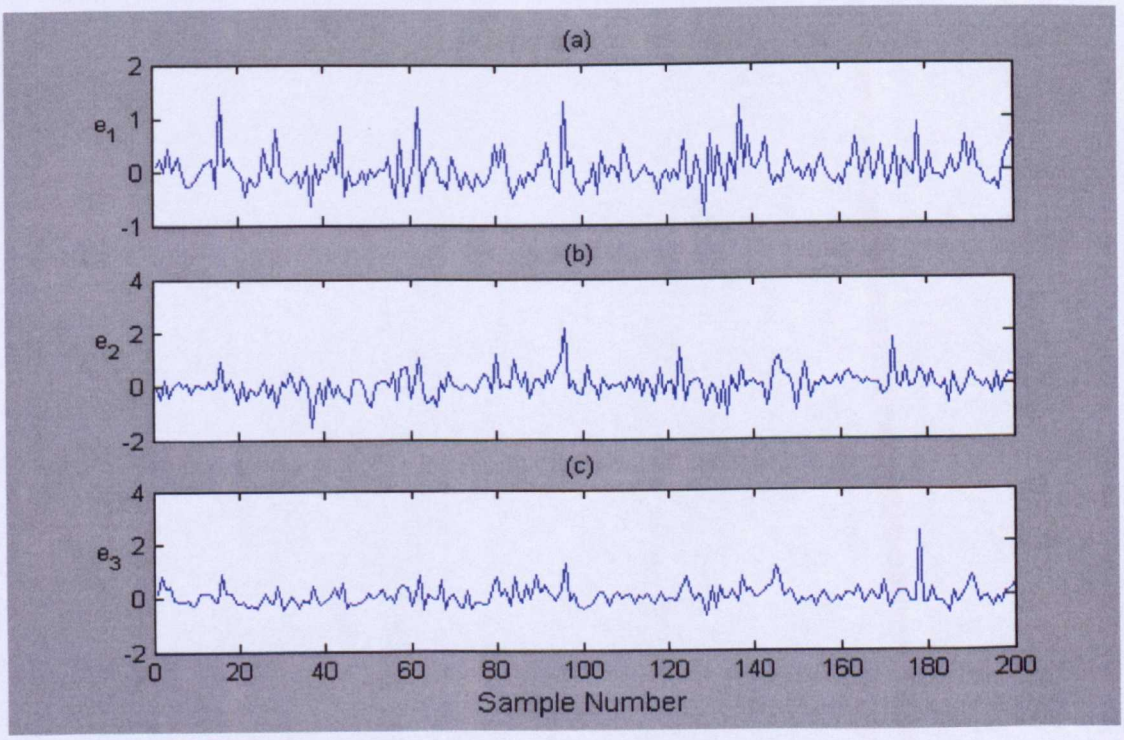


Figure 3.6: Time series plots of the residuals using NLPLS1 algorithm (example 2)

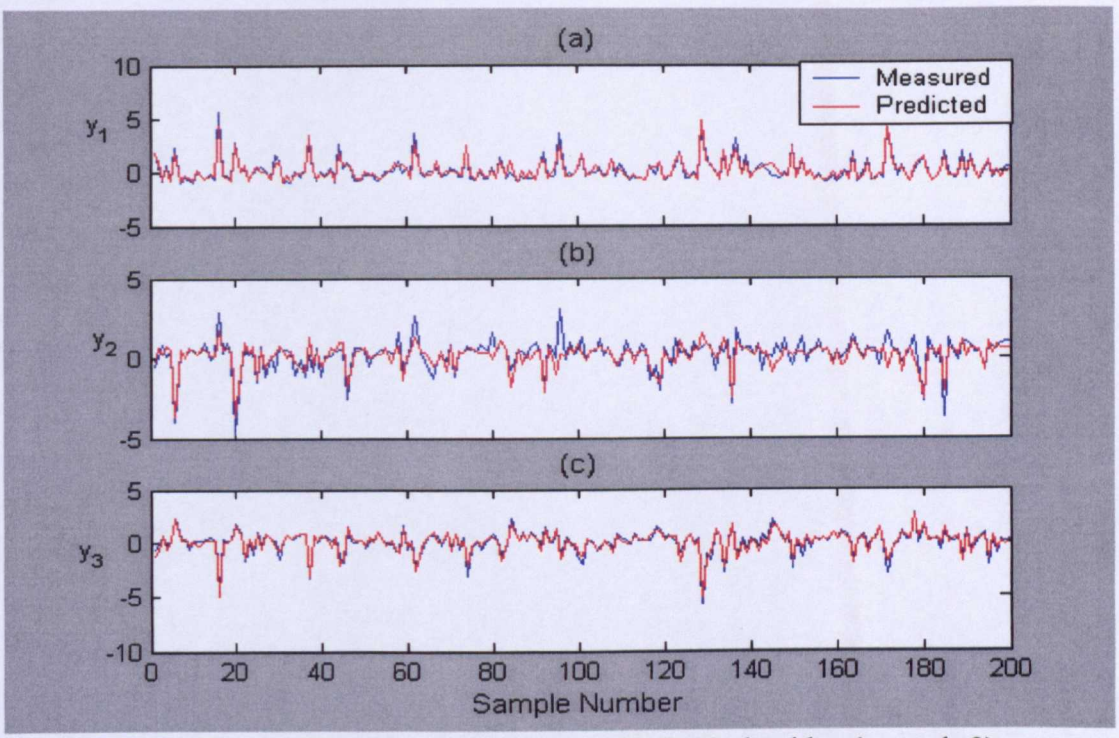


Figure 3.7: Prediction of Response variables using NLPLS2 algorithm (example 2).

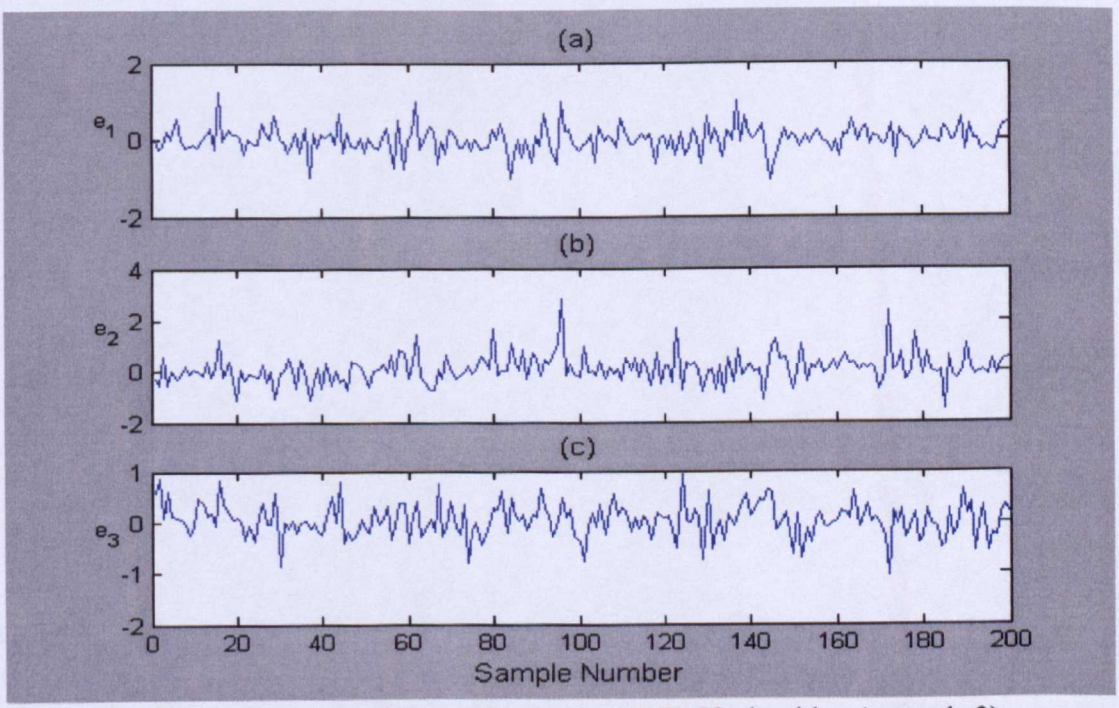


Figure 3.8: Time series plots of the residuals using NLPLS2 algorithm (example 2)

3.11.3 Example 3: pH neutralization process

In the third and final application, data from a pH neutralization process is considered. This process has been used as a benchmark process for testing the performance of different control algorithms (Henson and Seborg, 1994; Johansen and Foss, 1997). The process consists of a tank where a strong acid (nitric acid) is neutralized by a strong base such as sodium hydroxide. A dynamic model of the process was developed by Henson and Seborg (1994). To collect steady state data, the flow rates are kept fixed until the process reaches steady state. Three flow rates Q_1, Q_2 and Q_3 are used as the predictor variables and the three variables namely pH value, level of tank and the output flow rate are used as the response variables. A data set consisting of 1000 samples is collected and divided into a training data set (800 samples) and a validation data set (200 samples). The performance of the NLPLS1 and NLPLS2 algorithms is given in Tables 3.13 and 3.14 respectively. Figures 3.9 and 3.11 show the prediction of the response variables using two latent variables and the corresponding residuals are shown in Figures 3.10 and 3.12 respectively. The performances of linear PLS, the algorithm of Wold et al., (1989) and the algorithm of Baffi et al., (1999(a)) are given in Tables 3.15, 3.16 and 3.17 respectively.

Table 3.13: Performance of NLPLS1 algorithm (example 3)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	35.79	35.79	63.46	63.46	1.1348	1.1720
2	31.00	66.79	26.97	90.43	0.2061	0.1923
3	33.21	100.00	0.13	90.56	0.2060	0.1923

Table 3.14: Performance of NLPLS2 algorithm (example 3)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	35.72	35.72	65.20	65.20	1.0239	1.0632
2	31.03	66.75	28.60	93.82	0.1833	0.1615
3	33.25	100.00	0.11	93.93	0.1804	0.1586

Table 3.15: Performance of linear PLS algorithm (example 3)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	35.80	35.80	44.22	44.22	1.7162	1.7443
2	30.97	66.77	2.65	46.87	1.7053	1.7231
3	33.23	100.00	4.90	51.77	1.6982	1.7187

Table 3.16: Performance of Wold et al., (1989) algorithm (example 3)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	30.96	30.96	28.05	28.05	2.1558	1.7511
2	33.69	64.65	1.56	29.61	2.1088	1.7031
3	35.35	100.00	58.93	88.54	0.3428	0.2856

Table 3.17: Performance of Baffi et al., (1999(a)) algorithm (example 3)

No. of LV	% Variance explained (X)	Cumulative % variance explained (X)	% Variance explained (Y)	Cumulative % variance explained (Y)	MSPE (Training Data)	MSPE (Validation Data)
1	35.74	35.74	65.89	65.89	1.0220	1.0616
2	30.83	66.56	28.11	94.00	0.1799	0.1579
3	33.44	100.00	0.00	94.00	0.1798	0.1580

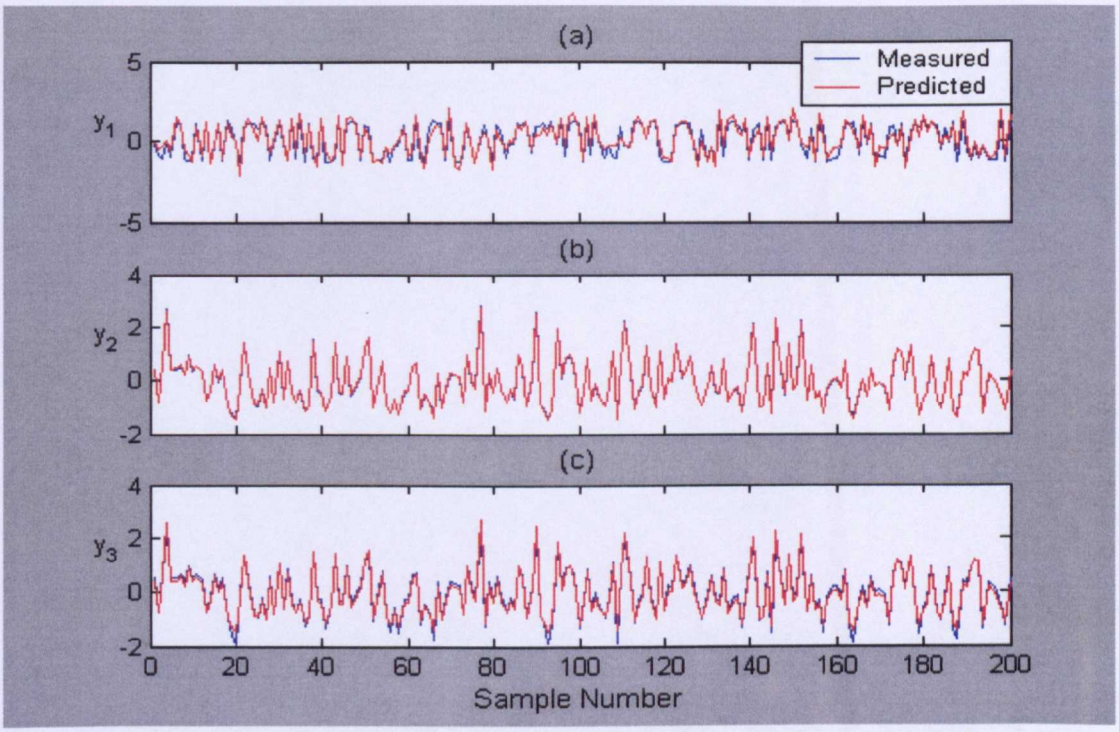


Figure 3.9: Prediction of response variables using NLPLS1 algorithm (example 3)

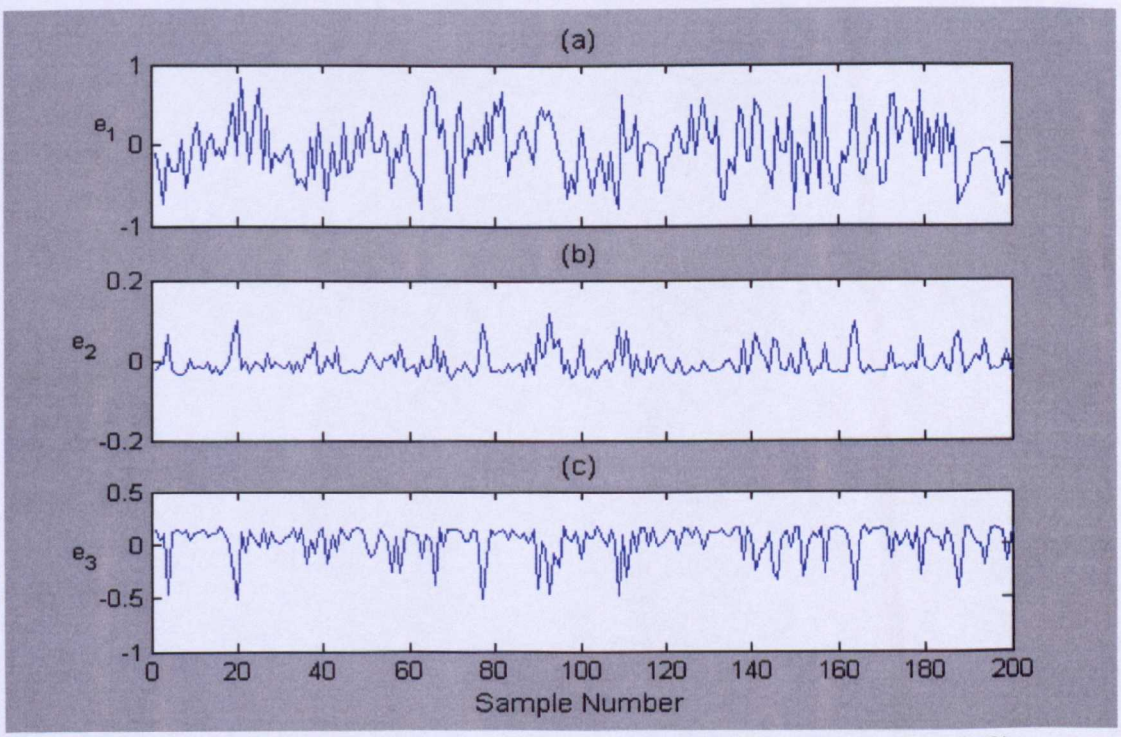


Figure 3.10: Time series plots of the residuals using NLPLS1 algorithm (example 3)

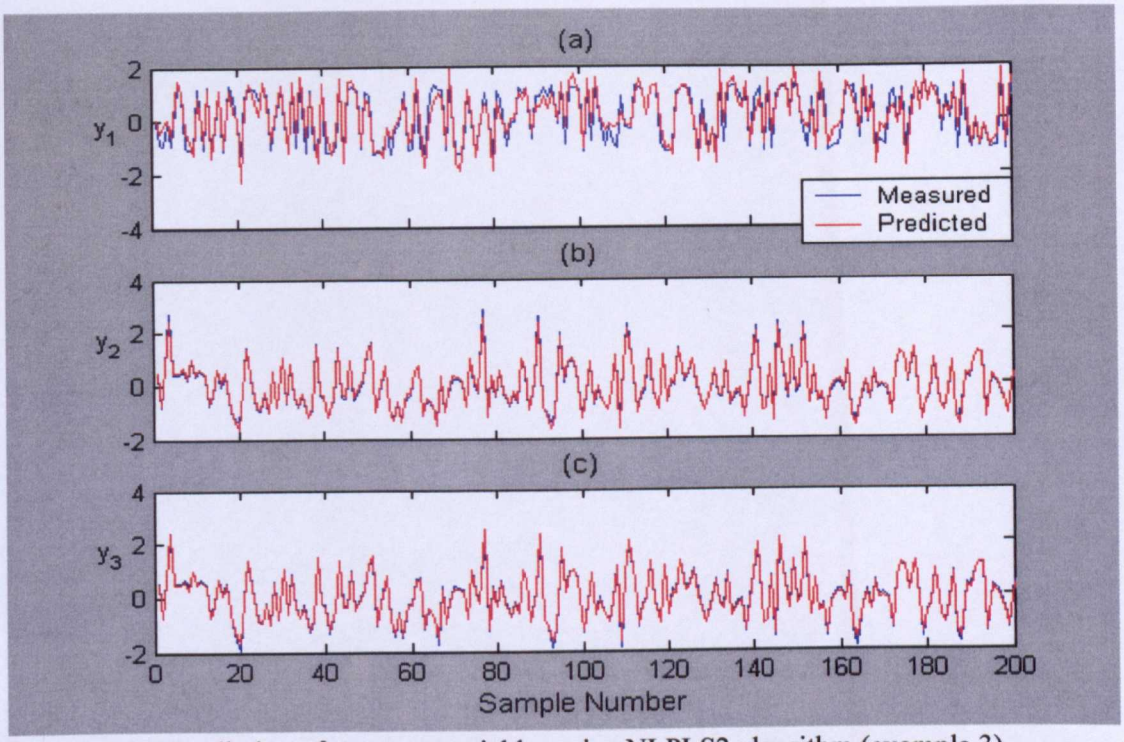


Figure 3.11: Prediction of response variables using NLPLS2 algorithm (example 3)

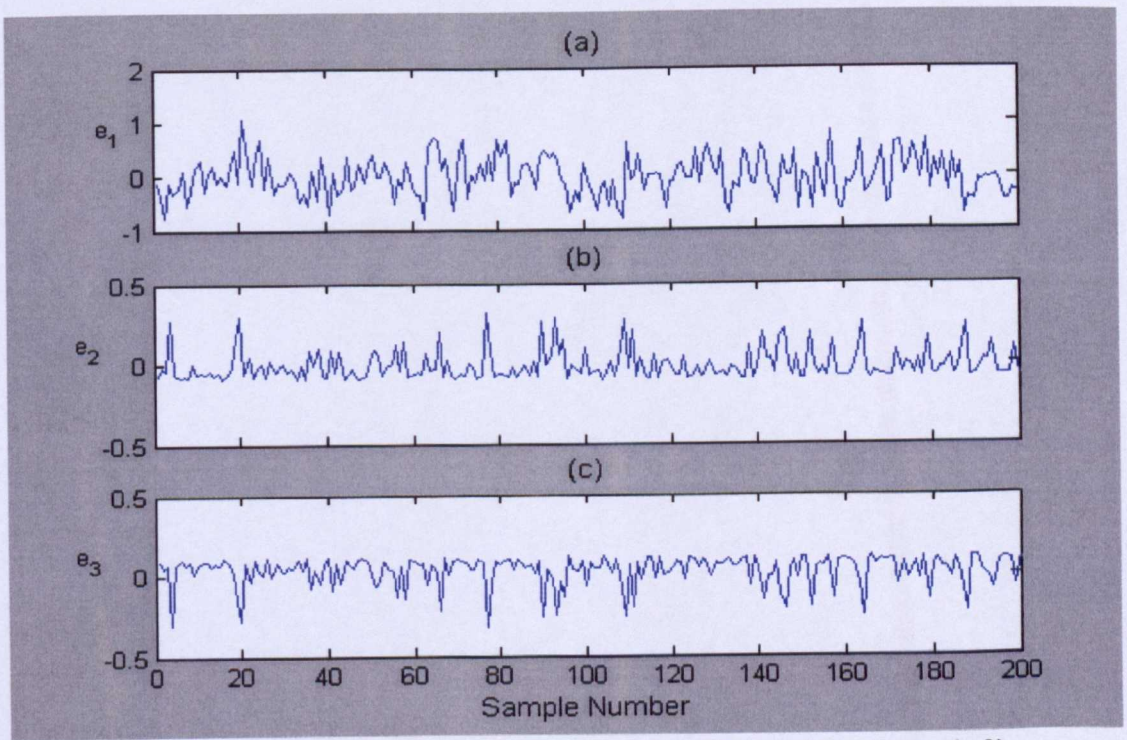


Figure 3.12: Time series plot of the residuals using NLPLS2 algorithm (example 3)

The following conclusions can be drawn from the above results:

1. The percentage variance of Y explained by the NLPLS1 and NLPLS2 algorithms on the training data set is higher than the linear PLS and the Wold et al., (1989) algorithm for a given set of latent variables.
2. The mean square prediction error (MSPE) for NLPLS1 and NLPLS2 algorithms is lower than the corresponding values of MSPE for linear PLS and Wold et al., (1989) algorithms.
3. NLPLS2 performs slightly better than NLPLS1 algorithm in terms of predictive ability of response variables.
4. The algorithm of Baffi et al., (1999(a)) is the best among all the algorithms considered in terms of prediction ability.

The reasons for these observations are the same as in example 1 and example 2.

3.12 Conclusions

A number of non-linear extensions of PLS have been proposed in the literature. In this chapter, following a review of the existing algorithms, it is proven that the error based non-linear PLS algorithm proposed by Baffi et al., (1999(a)) maximizes the accuracy with which the response variables are predicted and is, therefore, a non-linear extension of reduced rank regression. It is argued that a 'true' non-linear PLS algorithm should be based on the maximization of the 'non-linear covariance' function so as to preserve the spirit of linear PLS. After careful investigation, it is proven that the algorithm of Wold et al., (1989) makes attempts to achieve this objective but has several limitations. To overcome these limitations, two non-linear PLS algorithms which maximize the non-linear covariance function are proposed. The performance of these algorithms is compared and contrasted with linear PLS and the Wold et al., (1989) algorithm. In addition, all the non-linear PLS algorithms in the literature have been classified into three categories namely covariance based, quick and dirty and error based depending on the underlying objective functions.

Having incorporated the non-linear feature within the conventional linear PLS algorithm, the next step is to take into consideration the process dynamics so that a dynamic (and linear) model can be identified. This is the focus of the next chapter.

CHAPTER 4

Dynamic Partial Least Squares

4.1. Introduction

Partial Least Squares assumes that a linear and static (algebraic) relationship exists between the variables. In a typical process, however, the data collected for building the empirical model may exhibit serial correlation and therefore, the application of PLS will be inappropriate. Also if the model is to be used for process control then it is essential that process dynamics are included. A number of approaches have been proposed to modify PLS to take into consideration the dynamics of the process. One possible way is to first apply static PLS to the input matrices **X** and **Y** and then fit a dynamic relationship between the scores. This approach, which is investigated in this chapter, was used by Lakshminarayan et al., (1997) to identify and control a multivariate process. This approach, however, has the disadvantage that the weights of the outer relationship in the PLS model are not determined utilising the dynamics of the process. The contribution of this chapter is to propose a method that fully integrates a dynamic model within the PLS framework and determine all the parameters (outer weights and inner dynamic model parameters) of the PLS model as dictated by the dynamics of the process.

4.2. Literature Review

There have been two main approaches to introducing the dynamics of a process into the basic PLS algorithm. One approach is to include lagged values of the input and /or output variables in the input data matrix **X** and then use the basic PLS algorithm to identify the dynamic relationship between the input and output matrices. This method was adopted by Ricker (1988), but only the lagged values of input variables were included in the input matrix to identify the impulse response of a model. Qin and McAvoy (1992(a)) used lagged values of both input and output variables to identify a multivariate Autoregressive Moving Average (ARMA) model. Mathematically, this method can be denoted as:

$$\mathbf{Y} = \mathbf{X}_{dyn}\mathbf{B}_{dyn} + \mathbf{E} \qquad (4.1)$$

where $\mathbf{X}_{dyn}$ is a matrix consisting of measurements of the input variables and lagged values of the input and/or output variables and $\mathbf{B}_{dyn}$ is the regression matrix relating the matrix $\mathbf{X}_{dyn}$ to the output matrix $\mathbf{Y}$.

Although simpler to understand, this method has the disadvantage that through the inclusion of lagged values of the variables in the input matrix, the dimensionality of the input matrix becomes extremely large, particularly, for Multi-Input and Multi-Output (MIMO) systems in which the number of input and/or output variables is large, a direct consequence of which is an increase in computational burden.

To overcome the need to include lagged values of the input and/or output variables, Kaspar and Ray (1992; 1993(a)) proposed a method whereby the dynamics of the data are taken care of by first filtering the data using a suitable dynamic filter. Their argument was that in this way, the dynamics of the data are removed and hence the relationship between the output of the filter and the output variables is algebraic. The filter selected was either based on prior process knowledge or was designed by optimizing an objective function. The approach was applied for the identification and design of controllers for a simulated process (Kaspar and Ray, 1993(a))

Another approach, investigated in this chapter, is where the inner relationship of the basic PLS algorithm is modified (Lakshminarayan et al., 1997). Instead of using a static relationship between the scores, the relationship between the scores is replaced by an appropriate dynamic relationship. The methodology involves first performing PLS on the input and output data matrices without including lagged values of the input and output variables in the input matrix and then fitting a dynamic relationship between the resulting scores. Mathematically, if t_i and u_i $(i = 1,2...A)$ denote the latent variables obtained by applying PLS to matrices $\mathbf{X}$ and $\mathbf{Y}$ and $G_i(t_i)$ denotes the dynamic model fitted between the latent variables t_i and u_i, then the dynamic PLS decomposes the matrix $\mathbf{Y}$ as follows:

$$\mathbf{Y} = G_1(\mathbf{t}_1)\mathbf{q}_1^T + G_2(\mathbf{t}_2)\mathbf{q}_2^T + \ldots G_A(\mathbf{t}_A)\mathbf{q}_A^T + \mathbf{E} \qquad (4.2)$$

where $\mathbf{q}_i$ $(i = 1,2...A)$ are the loading vectors

The main advantage of this approach is that the problem of identifying a MIMO model can be reduced to the problem of identifying multiple SISO (Single Input Single Output) models. This strategy, therefore, realises the use of the wealth of identification and control algorithms that have been developed for SISO systems. This is, particularly useful for non-linear (dynamic) systems since for example, the structure selection and training of a MIMO neural network based non-linear model is much more difficult than for (a series of) SISO models. This approach was successfully used by Lakshminarayan et al., (1997) for identifying and controlling a multivariate process. However, this method has one shortcoming; the weights in the outer relationship $\mathbf{w}_i$ and $\mathbf{q}_i$ are not determined by the dynamics of the process. The approach, therefore, may be suboptimal in terms of the predictive ability of the model. This is, in particular, important for processes that have fast dynamics. In this chapter a method for determining the outer weights is proposed so that the dynamic relationship between the scores can be fully integrated within the PLS model. It should be noted, however that a general limitation of fitting an inner dynamic model between the scores is that it is difficult to determine the number of delays and the magnitude of the serial correlation of the scores from knowledge of the delays and serial correlation of measured variables. Each input variable, for example, may have a different autocorrelation function and since a latent variable is the weighted sum of the input variables, it may be difficult to determine the autocorrelation function of the latent variable given the autocorrelation function of the input variables. This makes it difficult to determine the order of the inner dynamic scores model from the serial correlation of the measured variables.

4.3. Modified Dynamic Partial Least Squares

Let $\mathbf{x}(n)$ and $\mathbf{y}(n)$ denote the n^{th} sample of the input and output variables respectively so that the latent variables for the same sampling instant are given by:

$$t_1(n) = \mathbf{x}(n)^T \mathbf{w}_1(n) \tag{4.4}$$

and

$$u_1(n) = \mathbf{y}(n)^T \mathbf{v}_1(n) \tag{4.5}$$

where $\mathbf{w}_1(n)$ and $\mathbf{v}_1(n)$ are the outer weights at sampling instant n. Let the ARX (p, q, d) model between t_1 and u_1 be given by:

$$u_1(n) = a_1(n)u_1(n-1) + a_2(n)u_1(n-2) + \ldots + a_p(n)u_1(n-p) + b_0(n)t_1(n-d) + b_1(n)t_1(n-d-1) + \ldots + b_q(n)t_1(n-d-q) + e_1(n) \quad (4.6)$$

with the prediction of the u-scores from equation (4.6) given as:

$$\hat{u}_1(n) = a_1(n)u_1(n-1) + a_2(n)u_1(n-2) + \ldots + a_p(n)u_1(n-p) + b_0(n)t_1(n-d) + b_1(n)t_1(n-d-1) + \ldots + b_q(n)t_1(n-d-q) \quad (4.7)$$

Equation (4.7) can be re-written as:

$$\hat{u}(n) = \boldsymbol{\varphi}_1^T(n)\boldsymbol{\theta}_1(n) \quad (4.8)$$

where

$$\boldsymbol{\varphi}_1(n) = [u_1(n-1)\ u_1(n-2)\ldots u_1(n-p)\ t_1(n-d)\ t_1(n-d-1)\ldots t_1(n-d-q)\ ]^T$$
$$\boldsymbol{\theta}_1(n) = [a_1(n)\ a_2(n)\ldots a_p(n)\ b_0(n)\ b_1(n)\ldots b_q(n)]^T \quad (4.9)$$

To integrate the ARX model within the PLS framework, the weight vectors $\mathbf{w}_1(n), \mathbf{v}_1(n)$ and the ARX model parameter vector $\boldsymbol{\theta}_1(n)$ are determined such that average of the square prediction error, J is a minimum:

$$J = E\{e_1^2(n)\} = E\{(u_1(n) - \hat{u}_1(n))^2\} \quad (4.10)$$

Taking the instantaneous value $e_1^2(n)$ as the estimate of $E\{e_1^2(n)\}$ for the on-line optimization of the objective function (Widrow, 1985), the derivatives can be computed as follows:

$$\frac{\partial J}{\partial a_i(n)} = -2(e_1(n)u_1(n-i)) \quad \text{for} \quad i = 1,2...p$$
$$\frac{\partial J}{\partial b_j(n)} = -2(e_1(n)t_1(n-d-j)) \quad j = 0,1,...q \qquad (4.11)$$

From equation (4.11), the derivative of the objective function with respect to the parameter vector $\boldsymbol{\theta}_1$ can be written as:

$$\frac{\partial J}{\partial \boldsymbol{\theta}_1} = -(e_1(n)\varphi_1(n)) \qquad (4.12)$$

Also

$$\frac{\partial J}{\partial \mathbf{w}_1(n)} = -2\left(e_1(n)\frac{\partial e_1(n)}{\partial \mathbf{w}_1(n)}\right)$$
$$\frac{\partial J}{\partial \mathbf{v}_1(n)} = -2\left(e_1(n)\frac{\partial e_1(n)}{\partial \mathbf{v}_1(n)}\right) \qquad (4.13)$$

Now

$$\frac{\partial e_1(n)}{\partial \mathbf{w}_1(n)} = \frac{\partial(u_1(n)-\hat{u}_1(n))}{\partial \mathbf{w}_1(n)} = -\frac{\partial \hat{u}_1(n)}{\partial \mathbf{w}_1(n)}$$
$$\frac{\partial e_1(n)}{\partial \mathbf{v}_1(n)} = \frac{\partial(u_1(n)-\hat{u}_1(n))}{\partial \mathbf{v}_1(n)} = \mathbf{y}(n) - \frac{\partial \hat{u}_1(n)}{\partial \mathbf{v}_1(n)} \qquad (4.14)$$

Equation (4.14) requires the computation of the differential of the predicted u-scores with respect to the outer weight vectors. This can be computed as follows. From equation (4.7):

$$\frac{\partial \hat{u}_1}{\partial \mathbf{w}_1(n)} = b_0(n)\frac{\partial t_1(n-d)}{\partial \mathbf{w}_1(n)} + b_1(n)\frac{\partial t_1(n-d-1)}{\partial \mathbf{w}_1(n)} + b_2(n)\frac{\partial t_1(n-d-2)}{\partial \mathbf{w}_1(n)} + ...$$
$$..+ b_q(n)\frac{\partial t_1(n-d-q)}{\partial \mathbf{w}_1(n)} \qquad (4.15)$$

where

$$\frac{\partial u_1(n-i)}{\partial \mathbf{w}_1(n)} = 0 \quad \text{for } i = 1, 2, \ldots p \tag{4.16}$$

To compute the differential of the predicted u-scores with respect to the present weight vector $\mathbf{w}_1(n)$, the differentials of the past scores $t_1(n-i)$ with respect to the present weight vector $\mathbf{w}_1(n)$ require to be computed. It is important to note that the past scores also depend on the present weight vector $\mathbf{w}_1(n)$. This is because the past scores $t(n-i)$ $(i = 1,2 \ldots q)$ depend on the past weight vectors $\mathbf{w}_1(n-i)$ which in turn are related to the present weight vector $\mathbf{w}_1(n)$ through the recursive weight updating equation (4.20).

To compute the differentials in equation (4.15), the following approximation is used. If the learning rate η is small, $\mathbf{w}_1(n) \approx \mathbf{w}_1(n-1) \ldots \approx \mathbf{w}_1(n-q)$. This assumption is particularly justified where the order q (and p) of the ARX model is small. Introducing this assumption into equation (4.14) gives:

$$\frac{\partial \hat{u}_1}{\partial \mathbf{w}_1(n)} \approx b_0(n)\frac{\partial t_1(n-d)}{\partial \mathbf{w}_1(n-1)} + b_1(n)\frac{\partial t_1(n-d-1)}{\partial \mathbf{w}_1(n-1)} + b_2(n)\frac{\partial t_1(n-d-2)}{\partial \mathbf{w}_1(n-2)} + \ldots + b_q(n)\frac{\partial t_1(n-d-q)}{\partial \mathbf{w}_1(n-q)} \tag{4.17}$$

From equation (4.4):

$$\frac{\partial t_1(n-d-i)}{\partial \mathbf{w}_1(n\text{-}i)} = \mathbf{x}(n-d-i) \quad \text{for } i = 0,1,2,\ldots q \tag{4.18}$$

Now substituting this back into equation (4.14) gives:

$$\frac{\partial \hat{u}_1}{\partial \mathbf{w}_1(n)} \approx b_0(n)\,\mathbf{x}(n-d) + b_1(n)\,\mathbf{x}(n-d-1) + \ldots + b_q(n)\,\mathbf{x}(n-d-q) \tag{4.19}$$

Similarly

$$\frac{\partial \hat{u}_1}{\partial \mathbf{v}_1(n)} \approx a_1(n)\,\mathbf{y}(n\text{-}1) + a_2(n)\,\mathbf{y}(n\text{-}2) + \ldots + a_p(n)\,\mathbf{y}(n\text{-}q) \tag{4.20}$$

Using equations (4.13) (4.16) and (4.17) in equation (4.12):

$$\frac{\partial J}{\partial \mathbf{w}_1(n)} = -\left(e_1(n)\left(b_0(n)\,\mathbf{x}(n-d) + b_1(n)\,\mathbf{x}(n-d-1) + \ldots + b_q(n)\,\mathbf{x}(n-d-q)\right)\right)$$
$$\frac{\partial J}{\partial \mathbf{v}_1(n)} = -\left(e_1(n)\left(\mathbf{y}(n) - a_1(n)\,\mathbf{y}(n-1) - a_2(n)\,\mathbf{y}(n-2) - \ldots - a_p(n)\,\mathbf{y}(n-p)\right)\right) \tag{4.21}$$

Once the differentials of the objective function are known, the parameters $\boldsymbol{\theta}_1, \mathbf{w}_1, \mathbf{v}_1$ can be updated using the gradient descent rule:

$$\boldsymbol{\theta}_1(n+1) = \boldsymbol{\theta}_1(n) - \eta \frac{\partial J}{\partial \boldsymbol{\theta}_1(n)}$$
$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) - \eta \frac{\partial J}{\partial \mathbf{w}_1(n)}$$
$$\mathbf{v}_1(n+1) = \mathbf{v}_1(n) - \eta \frac{\partial J}{\partial \mathbf{v}_1(n)} \tag{4.22}$$

where η is the learning rate and the gradients are given in equations (4.21) and (4.12).

4.3.1 Transfer Function and Prediction

To find the transfer function for the first set of latent variables, the Z-Transform (Oppenheim et al., 1989) is applied to both sides of equation (4.7):

$$\hat{U}_1(z) = (a_1(n)\,z^{-1} + a_2(n)z^{-2} + \ldots a_p(n)z^{-p})U_1(z) + (b_0(n)\,z^{-d} + b_1(n)z^{-d-1} + \ldots b_q(n)z^{-d-q})T_1(z) \tag{4.23}$$

where $\hat{U}_1(z), U_1(z)$ and $T_1(z)$ denote the Z-transform of $\hat{u}_1(n), u_1(n)$ and $t_1(n)$ respectively. Once the parameters a_i and b_i of the ARX model have converged, they are independent of time and therefore, the time index n in equation (4.23) can be dropped. Denoting

$$A_1(z) = a_1 z^{-1} + a_2 z^{-2} + ...a_p z^{-p}$$
$$B_1(z) = b_1 z^{-d} + b_2 z^{-d-1} + ...b_q z^{-d-q} \quad (4.24)$$

then, equation (4.23) can be re-written as:

$$\hat{U}_1(z) = A_1(z)U_1(z) + B_1(z)T_1(z) \quad (4.25)$$

It should be noted from equation (4.25) that to predict the u-score at sampling time n, the past scores $u_1(n-j)$ for j =1,2...p and $t_1(n-d-i)$ for i =0,1,2,...q are required. These scores in turn require the past outputs $\mathbf{y}(n-j)$ and inputs $\mathbf{x}(n-d-i)$ to be measured on-line. In some processes, however, measurements of the output variables are not available on-line and it would, therefore, be useful if only past values of the inputs $\mathbf{x}(n-d-i)$ and hence past t-scores $t_1(n-d-i)$, are used to predict the u-scores. This can be done if the past predicted u-scores are used instead of the actual u-scores in equation (4.25). Replacing $U_1(z)$ in equation by $\hat{U}_1(z)$, the transfer function between the latent variables is given by:

$$H_1(z) = \frac{B_1(z)}{1 - A_1(z)} \quad (4.26)$$

Once the u-scores has been predicted, the prediction of the output variables **y** can be achieved by finding the loading vector $\mathbf{q}_1$ given by:

$$\mathbf{q}_1 = \frac{\mathbf{Y}^T \hat{\mathbf{u}}_1}{\hat{\mathbf{u}}_1^T \hat{\mathbf{u}}_1} \quad (4.27)$$

where $\hat{\mathbf{u}}_1$ is a vector containing the predictions of the u-scores for all the observations. The prediction $\hat{\mathbf{Y}}_1$ of **Y** is thus given by:

$$\hat{\mathbf{Y}} = \hat{\mathbf{u}}_1 \mathbf{q}_1^T \quad (4.28)$$

4.3.2 Computation of More than One Latent Variable

The second set of latent variables can be obtained by repeating the above procedure on the deflated matrices computed as follows:

$$\mathbf{X}_2 = \left(\mathbf{I} - \frac{\mathbf{t}_1 \mathbf{t}_1^T}{\mathbf{t}_1^T \mathbf{t}_1} \right) \mathbf{X} \qquad (4.29)$$

$$\mathbf{Y}_2 = \mathbf{Y} - \hat{\mathbf{Y}}$$

Higher latent variables can be computed similarly.

4.4 Summary of the Algorithm

Given: A matrix **X** of order $N \times K$, and **Y** of order $N \times M$

Autoscale each variable of **X** and **Y**

Step 1: Initialize the weight vectors $\mathbf{w}_1, \mathbf{v}_1$, and the parameter vector $\boldsymbol{\theta}_1$ to random values. Also

chose suitable values for the inner ARX model order, p, q, and d.

Step 2: Compute at time n

$$t_1(n) = \mathbf{x}(n)^T \mathbf{w}_1(n)$$

$$u_1(n) = \mathbf{y}(n)^T \mathbf{v}_1(n)$$

$$\hat{u}_1(n) = a_1(n)u_1(n-1) + a_2(n)u_1(n-2) + \ldots + a_p(n)u_1(n-p) + b_0(n)u_1(n-d) + b_1(n)t_1(n-d-1) + \ldots + b_q(n)t_1(n-d-q)$$

$$e_1(n) = u_1(n) - \hat{u}_1(n)$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}_1} = -\left(e_1(n)\varphi_1(n)\right)$$

$$\frac{\partial J}{\partial \mathbf{w}_1(n)} = -(e_1(n)(b_0(n)\,\mathbf{x}(n-d) + b_1(n)\,\mathbf{x}(n-d-1) + \ldots + b_q(n)\,\mathbf{x}(n-d-q))$$

$$\frac{\partial J}{\partial \mathbf{v}_1(n)} = -(e_1(n)(\mathbf{y}(n) - a_1(n)\,\mathbf{y}(n-1) - a_2(n)\,\mathbf{y}(n-2) - \ldots - a_p(n)\,\mathbf{y}(n-p))$$

Step 3: Update the parameters

$$\boldsymbol{\theta}_1(n+1) = \boldsymbol{\theta}_1(n) - \eta \frac{\partial J}{\partial \boldsymbol{\theta}_1(n)}$$

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) - \eta \frac{\partial J}{\partial \mathbf{w}_1(n)}$$

$$\mathbf{v}_1(n+1) = \mathbf{v}_1(n) - \eta \frac{\partial J}{\partial \mathbf{v}_1(n)}$$

Step 4: Repeat steps 2 and 3 for all sampling times $n = 1,2 \ldots N$.

Step 5: Repeat steps 2, 3 and 4 until convergence.

Step 6: Compute t-score $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$

Step 7: Predict u-score at each time:

$$\hat{u}_1(n) = a_1(n)u_1(n-1) + a_2(n)u_1(n-2) + \ldots + a_p(n)u_1(n-p) + b_0(n)u_1(n-d) + b_1(n)t_1(n-d-1) + \ldots + b_q(n)t_1(n-d-q)$$

and store all predictions in vector $\hat{\mathbf{u}}_1$

Step 8: Determine the loading vectors

$$\mathbf{p}_1 = \frac{\mathbf{X}^T\mathbf{t}_1}{\mathbf{t}_1^T\mathbf{t}_1} \qquad \mathbf{q}_1 = \frac{\mathbf{Y}^T\hat{\mathbf{u}}_1}{\hat{\mathbf{u}}_1^T\hat{\mathbf{u}}_1}$$

Step 9: Deflate the input and output matrices

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

$$\mathbf{Y} = \mathbf{Y} - \hat{\mathbf{u}}_1\mathbf{q}_1^T$$

Step 10: Validate the inner model on the validation data set. Change the values for the inner model order (p, q, d) and repeat steps 1 to 9. Select the best model order that explains maximal variance of the response variables for the validation data set.

Step 11: If additional latent variables are required, repeat steps 1 to 10 on the deflated matrices computed in step 9.

4.5 Simulation Studies

In this section, the proposed method is applied to identify a dynamic PLS model, first for data generated from an artificial system and then for a copolymerization process.

4.5.1 Example 1: Artificial data set

Consider a hypothetical dynamic process having two inputs and two outputs, described by the following state and measurement equations (Ku et al., 1995).

$$\mathbf{x}(n) = \begin{bmatrix} 0.811 & 0.226 \\ 0.477 & 0.415 \end{bmatrix}\mathbf{x}(n-1) + \begin{bmatrix} 0.193 & 0.689 \\ 0.320 & 0.749 \end{bmatrix}\mathbf{e}(n-1) \qquad (4.30)$$

$$\mathbf{u}(n) = \begin{bmatrix} 0.118 & 0.191 \\ 0.847 & 0.264 \end{bmatrix}\mathbf{u}(n-1) + \begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{bmatrix}\mathbf{x}(n-1)$$

$$\mathbf{y}(n) = \mathbf{u}(n) + \mathbf{h}(n)$$

where **u**, **x** and **y** $\in R^2$ are the state, input and output vectors respectively; **e** and **h** are zero mean Gaussian random vectors consisting of two independent random variables. The variance of each random variable in **e** is unity and for **h** is 0.1.

A data set consisting of 1500 samples is generated and divided into two sets. The first set (training data set) comprises 1000 samples while the second set (validation data set) comprises 500 samples. After the training data set is auto-scaled, the algorithm described in

section 4.5 is applied with an ARX (2, 1, 1) model fitted to each pair scores. The order of ARX (2, 1, 1) was determined by exploring the predictive capability first on the training data set and then validated on the validation data set. While the lower order models performed poorly, higher order models did not show significant increase in terms of their ability to predict. The percentage of variance captured by each latent variable is listed in Table 4.1. A PLS model with 2 latent variables is then built.

Table 4.1: Percent variance captured by PLS model (example 1)

No. of LV	% Variation explained (X)	Cumulative % variance explained (X)	% Variation explained (Y)	Cumulative % variance explained (Y)
1	50.71	50.71	57.04	57.04
2	49.29	100.00	39.41	96.45

The transfer functions $H_1(z)$ (between t_1 and u_1) and $H_2(z)$ (between t_2 and u_2) are:

$$H_1(z) = \frac{z^{-1} - 0.7437z^{-2}}{1 - 0.7876z^{-1} + 0.0111z^{-2}}$$
$$H_2(z) = \frac{0.0259z^{-1} + 0.1771z^{-2}}{1 - 0.1792z^{-1} + 0.1795z^{-2}} \quad (4.31)$$

Plots of the predictions for the two outputs y_1 and y_2 for the model validation data set are shown in Figures 4.1 and 4.2 respectively. The lower panel in each of these figures show a time series plot of the residuals. To test if the model is a good fit to the data, a bivariate plot of the residuals versus fitted values for the training data set for each of the two outputs is shown in Figure 4.3. The figure shows that no more 'information' is left in the residuals and therefore the model is a good fit to the data.

The performance of the algorithm (on the model validation data set) is quantitatively evaluated by two statistics namely the R-statistic, which is defined as the ratio of the Sum of Squares (SSQ) of the prediction error (for each individual output) to the SSQ of the measured signal:

$$R = \frac{\text{SSQ(Prediction Error)}}{\text{SSQ(Measured (original)Signal)}} \quad (4.32)$$

and the mean square error (MSE), which is defined as:

$$MSE = \frac{SSQ(\text{Prediction Error})}{\text{Number of samples}} \qquad (4.33)$$

The idea behind defining these two statistics is that while the MSE measures the absolute value of the variance in the error, the R-statistic measures the variance in the error expressed as a fraction of the variance of the original signal.

The values of the statistics, R and MSE, for each of the two outputs on the validation data set are given in Table 4.2. To investigate the impact of weight updating, inner dynamic models having the same order as above were built but this time without the outer weights being updated (Lakshminarayan et al., 1997). For comparison, the values of R and MSE for this case are also given in Table 4.2

Table 4.2: Summary of values of the statistics, R and MSE, (example 1)

Method	R		MSE	
	Output y_1	Output y_2	Output y_1	Output y_2
Integrated dynamic PLS	0.0279	0.0304	0.0360	0.0275
No-weight updating	0.1696	0.1379	0.1539	0.1228

It is seen from Table 4.2 that updating the weights in the PLS model according to the dynamics of the process has a considerable impact on model performance. For example, the value of R for the output y_1 when the outer weights are determined by the dynamics of process is 0.1696. This means that about 17 % of the variance in y_1 is left unexplained by the model. This figure reduces to about 2.8 % when the outer weights are used to capture the dynamics along with the inner model parameters. The higher predictive capability of the dynamic PLS model when the outer weights are updated is also reflected in the lower values of the mean square errors. The MSE of output y_1 when the outer weights are not updated is 0.1539 which drops to 0.0360 when the weights are updated. Similar conclusions about y_2 can also be made.

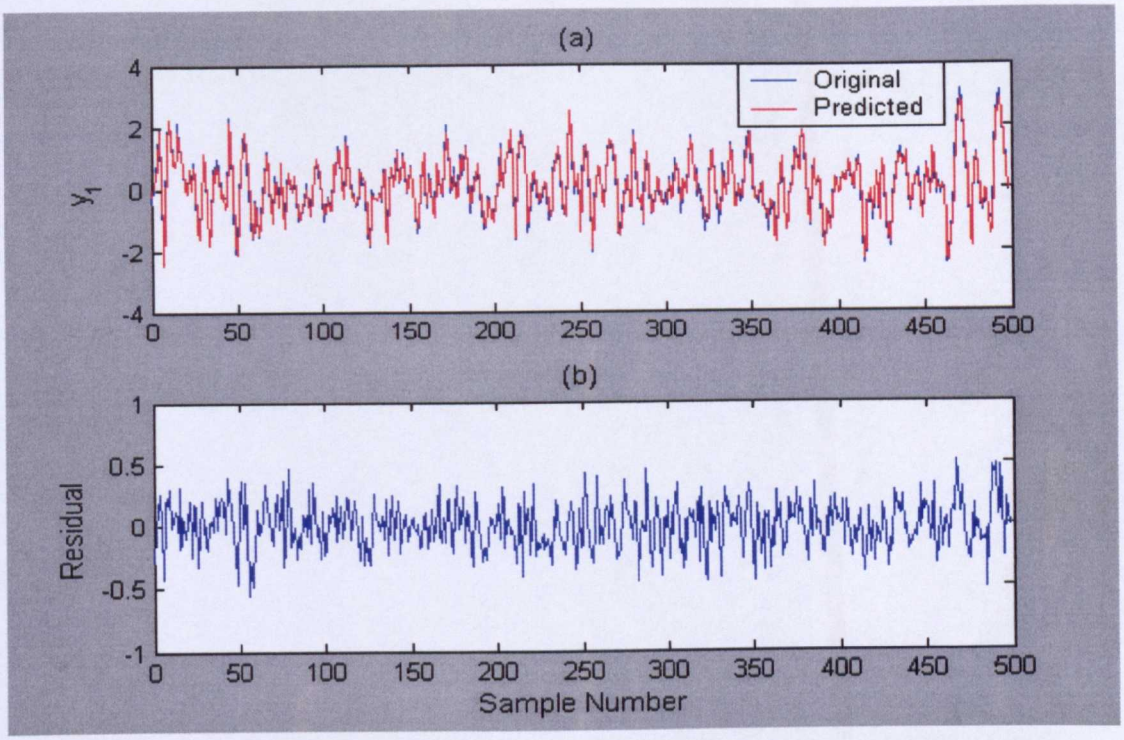


Figure 4.1: Time series plots of (a) the original and predicted values for the first output y_1 and (b) the residuals (example 1)

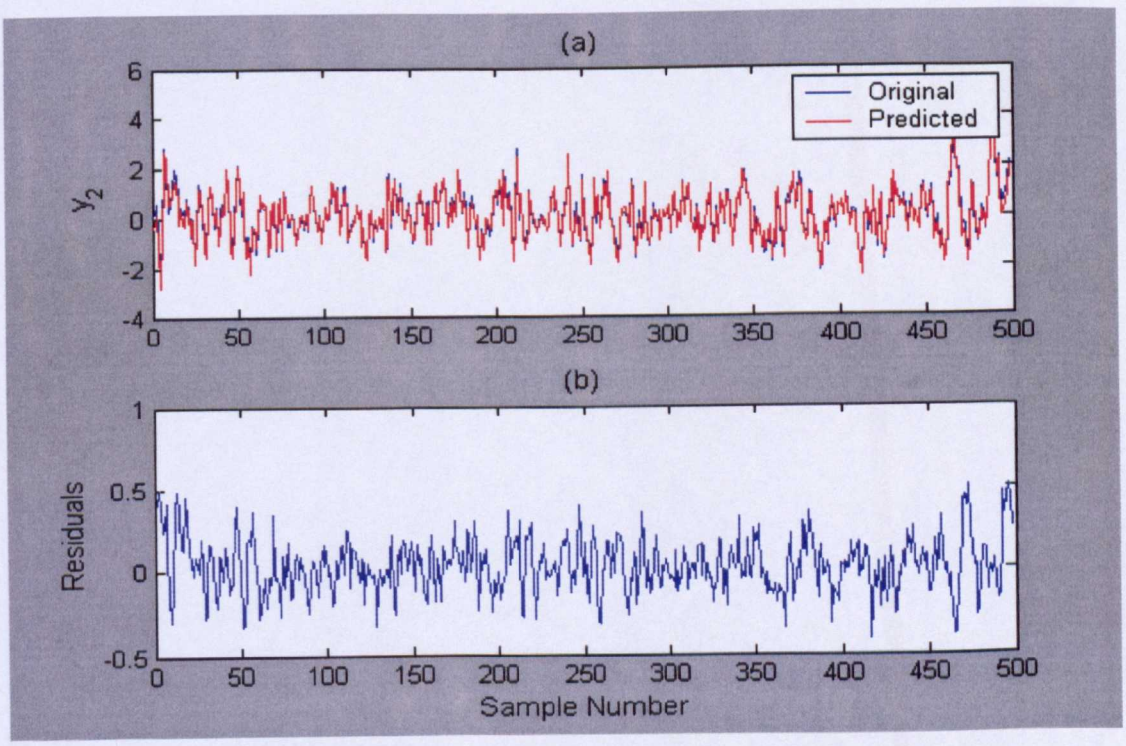


Figure 4.2: Time series plots of (a) the original and predicted values for the second output y_2 (b) the residuals (example 1)

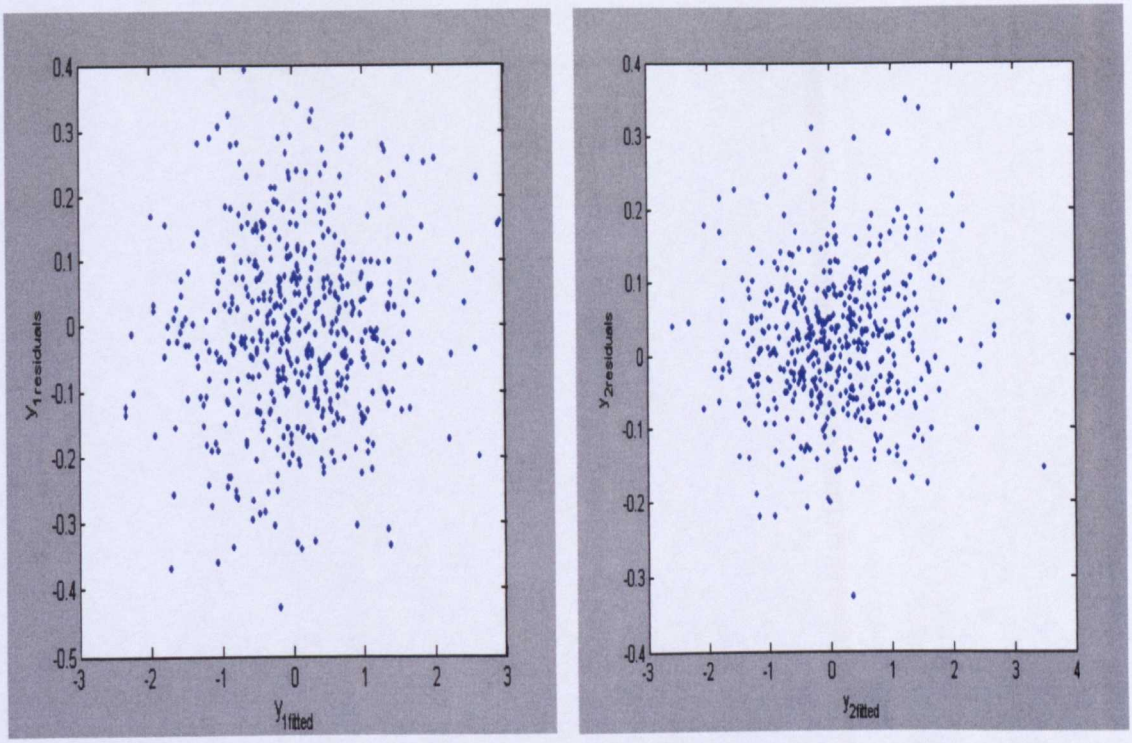


Figure 4.3: Bivariate plots of residuals versus fitted values for the two outputs (example 1)

4.5.2 Example 2: Co-polymerization Reactor

The integrated dynamic PLS model was finally applied to a comprehensive simulation of a continuous stirred tank copolymerization reactor (Achilias & Kiparssides, 1994). In the reactor monomers methyl methacrylate and vinyl acetate are continuously added to a perfectly mixed tank along with initiator azobisiobutyronitrile, solvent benzene, and chain transfer agent acetaldehyde and inhibitor m-dinitrobenzene. The process consists of four inputs

1. Feed concentration of monomer methyl methacrylate
2. Feed concentration of vinyl acetate
3. Feed concentration of chain transfer agent
4. Coolant temperature in the jacket

and four outputs

1. Reactor temperature
2. Polymerization rate
3. Composition of copolymer
4. Weight average molecular weight of copolymer

A nominal data set consisting of 1500 samples was generated by exciting the process with multi-level Pseudo Random Binary Signal (PRBS). The signal to noise ratio was set at 10 by adding measurement noise. Of the 1500 data samples, 1000 were used to identify the dynamic PLS model and the remaining 500 were used for model validation. After the data was autoscaled, the integrated dynamic PLS algorithm was applied with ARX(2,1,1), ARX(2,1,1), ARX(4,1,1) and ARX(5,1,1) structures chosen as the inner dynamic models. The choice of these parameters was determined as in example 1. The transfer functions identified for the inner dynamic models are:

$$H_1(z) = \frac{-0.017z^{-1} + 0.2529z^{-2}}{1 - 0.7564z^{-1} - 0.0361z^{-2}}$$

$$H_2(z) = \frac{0.2927z^{-1} + 0.0951z^{-2}}{1 - 0.5450z^{-1} - 0.1502z^{-2}}$$

$$H_3(z) = \frac{0.0077z^{-1} + 0.1221z^{-2}}{1 - 0.4830z^{-1} - 0.3819z^{-2} - 0.1215z^{-3} + 0.0691z^{-4}}$$

$$H_4(z) = \frac{0.0414z^{-1} + 0.1327z^{-2}}{1 - 0.4759z^{-1} - 0.3552z^{-2} - 0.2120z^{-3} - 0.0345z^{-4} + 0.1767z^{-5}} \quad (4.34)$$

Table 4.3: Percent variance captured by PLS model (example 2, Co-polymerization reactor)

No. of LV	% Variation explained (X)	Cumulative % variance explained (X)	% Variation explained (Y)	Cumulative % variance explained (Y)
1	28.38	28.38	59.09	59.09
2	25.42	53.80	23.56	82.65
3	25.04	78.84	8.41	91.06
4	21.16	100.00	0.97	92.03

A PLS model using 3 latent variables was built based on cross-validation. Figures 4.4, 4.5, 4.6 and 4.7 show the prediction of the four outputs on the model validation data set. The bivariate plots of the fitted values versus residuals for each of the four outputs on the training data set are shown in Figure 4.8. The values of the R-statistic and the mean square error for the four outputs on the validation data set are given in Table 4.4. As can be seen from Table 4.4, the conclusions derived in example 1 also hold for this example.

Table 4.4: Summary of values R-statistic and MSE (example 2, Co-polymerization Reactor)

Method	R				MSE			
	Output 1	Output 2	Output 3	Output 4	Output 1	Output 2	Output 3	Output4
Integrated dynamic PLS	0.0984	0.0454	0.1220	0.0339	0.0067	0.0064	4.54E-6	1.96E+3
No weight updating	0.3388	0.1592	0.3064	0.3041	0.0232	0.0170	1.22E-5	2.04E+4

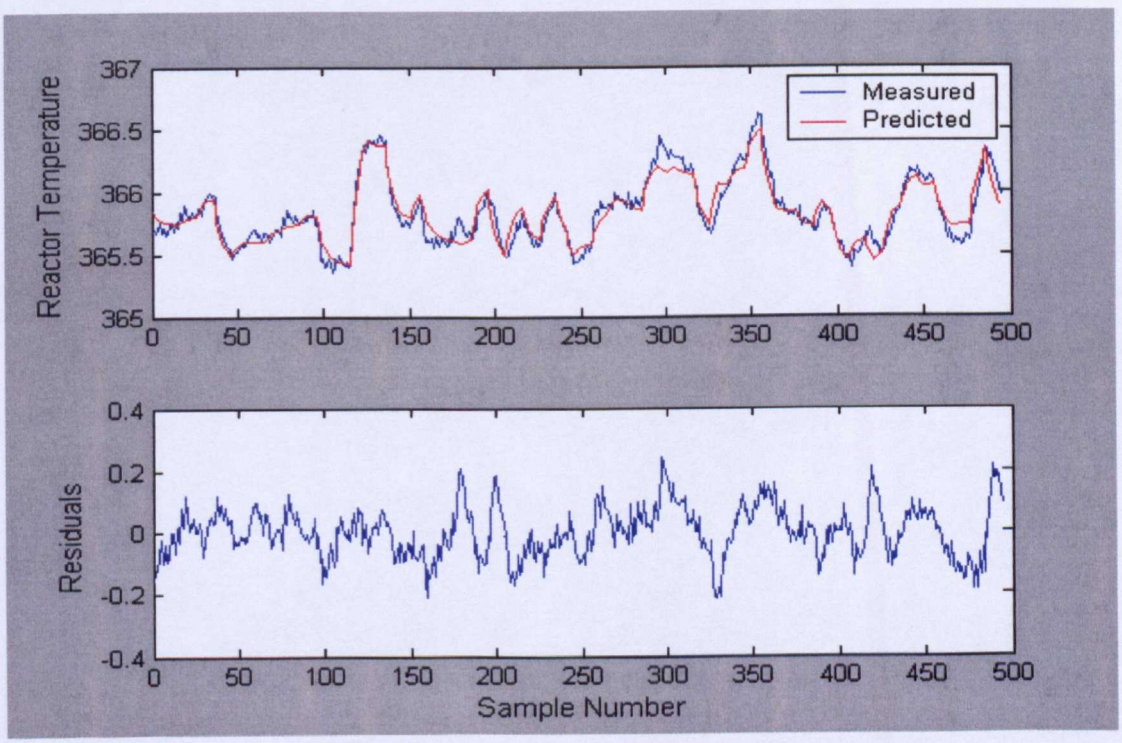


Figure 4.4: Time series plots of (a) the measured and predicted value of the reactor temperature and (b) the residuals (example 2)

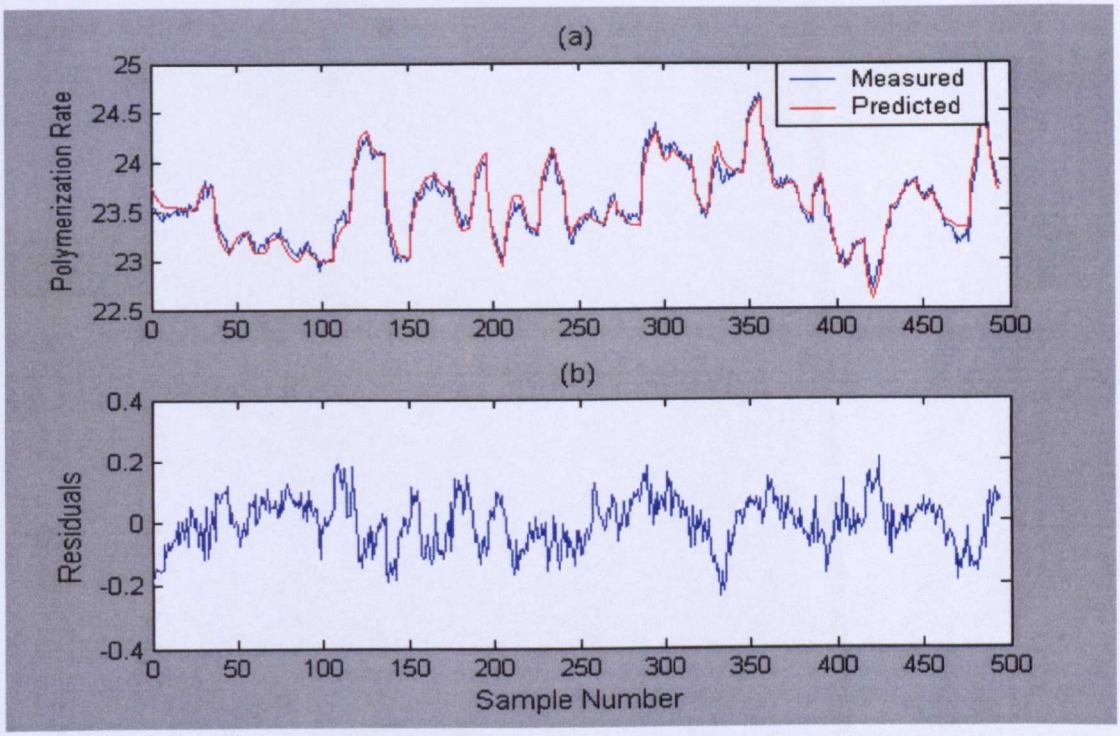


Figure 4.5: Time series plots of (a) the measured and predicted value of the polymerization rate and (b) the residuals (example 2)

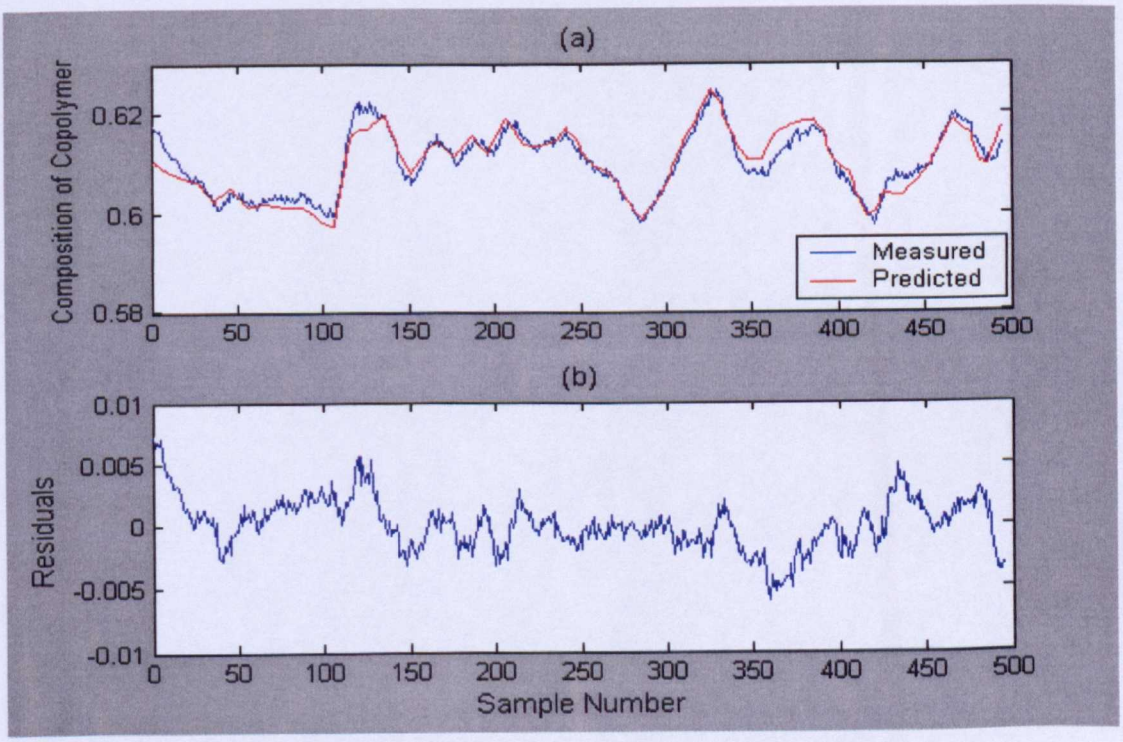


Figure 4.6: Time series plots of (a) the measured and predicted values of the copolymer composition and (b) the residuals (example 2)

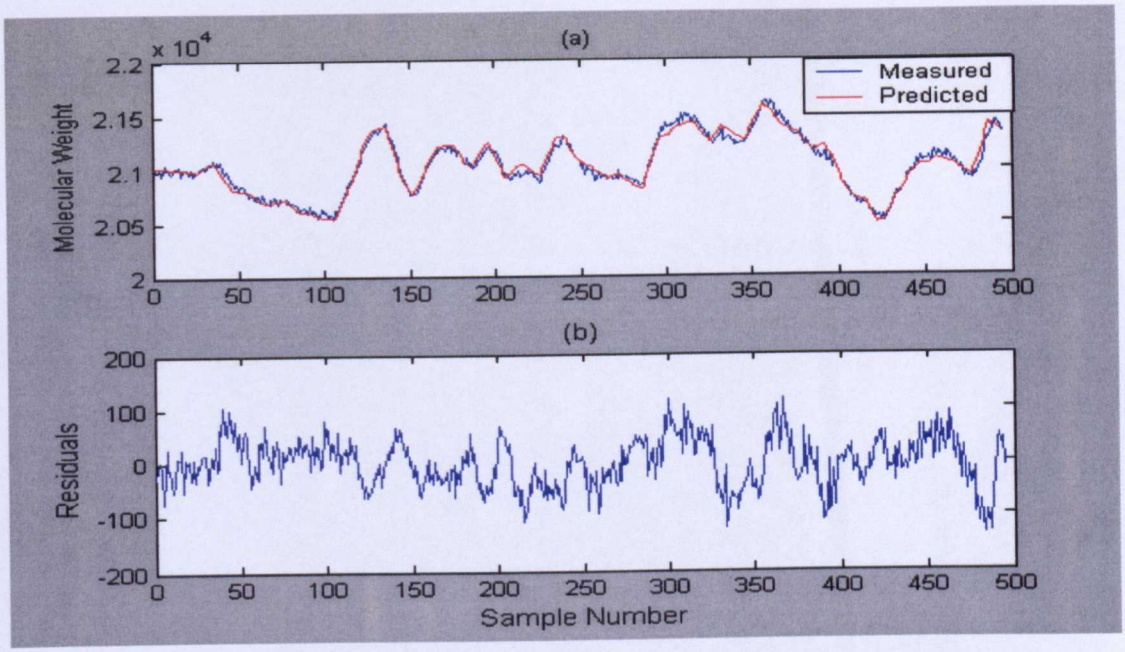


Figure 4.7: Time series plots of (a) the measured and predicted values of the weight average molecular weight and (b) the residuals (example 2).

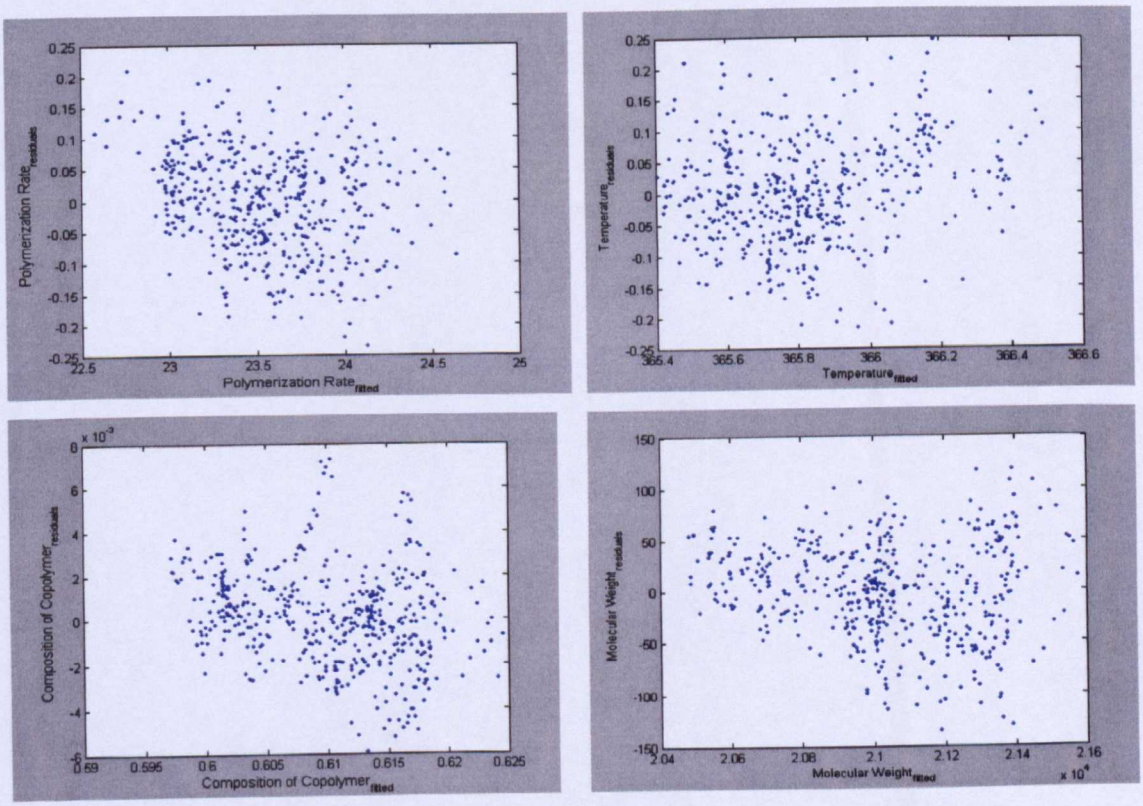


Figure 4.8: Bivariate plots of the residuals versus the fitted values (example 2)

4.6 Conclusions

In this chapter a method has been proposed to integrate a dynamic model within the PLS framework. The advantage of the method is that the task of identification of a MIMO model reduces to multiple SISO model identifications. The method differs from that previously proposed by Lakshminarayan et al., (1997) in that the determination of the outer weights and the inner dynamic relationship between the scores are integrated. The advantage of the determination of the outer weights according to the dynamics of process was illustrated using two examples. It was shown that the prediction capability of the model increases if all the parameters (outer weights and the inner model parameters) are determined in accordance with the dynamics of the process.

CHAPTER 5

Review of Statistical Process Monitoring Techniques

5.1 Introduction

A major challenge facing the process industries is to consistently manufacture good quality product. In practice, it is well known that there will be some degree of "inherent or natural" variability in any production process. This variation, which is caused by unknown factors, is termed common cause variation. However, other types of variability, known as "assignable cause variability", may occasionally be present in the process. This variability arises because of the occurrence of some changes in normal performance e.g. machine errors, operator errors or defective raw material. Such variability is generally large when compared to the natural variability and represents an unacceptable level of performance in terms of the final product. A process that is operating in the presence of assignable cause variability is said to be "out-of-statistical control".

To achieve tighter control over the critical process steps, and to monitor the performance of industrial process over time to detect any systematic drift of the process from its normal operating mode, a set of techniques are commonly employed. These techniques can be grouped under the heading of statistical process control (SPC) (Montgomery, 1991). The idea behind SPC is to use variables measurements in a process to detect changes in the equipment or process. In a typical SPC scheme, variables are first measured and then either the actual measurements or a statistic derived from them is plotted along with the associated confidence limits that are known as the warning or action limits. The resulting representation is known as a control chart. If the trace of the measurements or statistic lies within the confidence limits, it indicates the process is under statistical control whilst if the point lies outside the limit, this potentially indicates the occurrence of some abnormal events in the process, i.e. the process is out-of-statistical-control.

SPC techniques can be divided into two categories depending on how a given set of variables is monitored. If the variables are monitored individually without taking into consideration the interrelationship between the variables then the scheme is termed univariate. If the given set of variables is handled collectively, the methodology is termed as Multivariate Statistical Process Control (MSPC). The aim of this chapter is to give a brief overview of univariate and

multivariate monitoring schemes. But before an overview is undertaken, the statistical basis of SPC techniques is first summarised.

5.2 Statistical Basis of Control Charts

There is a strong link between control charts and hypothesis testing. In essence, the control chart is equivalent to a hypothesis test in which the null and alternative hypotheses are:

H_0: The process is under statistical control

H_1: The process is out-of-statistical control

A point on a control chart within the control limits indicates that the process is in statistical control and thus is equivalent to accepting the null hypothesis. A point outside the control limit is equivalent to rejecting the null hypothesis and this indicates that the process is not in statistical control. Similar to hypothesis testing, Type I and II errors can be defined in the context of control charts. A Type I error occurs if the null hypothesis is rejected when it is actually true (that is, a process is concluded to be out-of-statistical-control when it is really in statistical control). A type II error occurs when the null hypothesis is not rejected when it is in fact false (that is, a process is concluded to be in statistical control when actually it is out of statistical control)

The Type I error determines the false alarms rate. Commonly used values for Type I errors are 0.05 and 0.01, which means that on average, 5%(1%) of samples on the control chart are expected to lie outside the control limits even when the process is in control. The Type II error determines the delay (difference between the time point at which the change occurs and the time point at which the change is detected) in detecting the change. A more useful concept that unifies both errors is the Average Run Length (ARL). A run length is defined as the number of observations that pass from the time at which the change has occurred until the control chart gives a signal indicating the change. The average of run lengths is calculated (either theoretically or empirically) to determine the ARL. The number of samples that occur between the occurrence of a change and its detection is known as the out-of-control ARL. Since a chart gives a signal indicating the process is out of statistical control even if the process is in statistical control (a false alarm), the average run length between two false alarms is known as the in-control ARL.

It is desirable that the in-control ARL should be as high as possible (so that there are few false alarms) whereas the out-of-control ARL should be as low as possible (so that there is less delay in the detection of a change). Both these objectives, however, cannot be achieved simultaneously and a trade-off between the false alarm rate and the delay is required for the implementation of a control chart. Tighter control limits will give a small value for the out-of-control ARL but at the expense of an increase in the number of false alarms. Conversely, wider control limits will give fewer false alarms but at the expense of an increase in the delay in detection of a change in the process.

5.3 Univariate Monitoring Schemes

Three univariate monitoring schemes namely Shewhart, Cumulative Sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) are now briefly described.

5.3.1 Shewhart Chart

The Shewhart chart is one of the earliest control charts and is still popular today in the process industries. It was first introduced by Shewhart (1931) at the Bell Laboratories and by Dudding and Jenett in Britain in 1937 (Banks, 1993). In a typical Shewhart chart for monitoring the mean, a sample of N measurements on a quality variable are taken and the sample mean is plotted on the chart along with the confidence limits. A typical Shewhart control chart, with 95% and 99% confidence limits, is shown in Figure 5.1, where the sample mean for 100 batches, each consisting of 20 samples, is plotted.

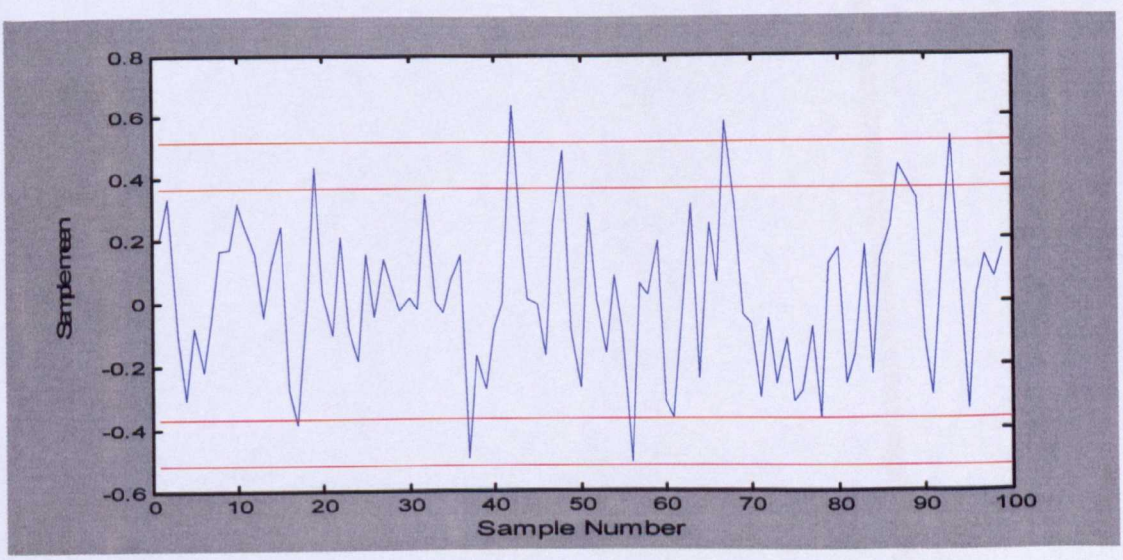


Figure 5.1: A typical Shewhart chart

A Shewhart chart for the standard deviation can likewise be plotted. It is known that the Shewhart chart is suitable for detecting larger shifts in the mean which are of the order of two or more standard deviations (Montgomery, 1991). To detect smaller changes, the cumulative sum (CUSUM) chart is more appropriate.

5.3.2 Cumulative Sum (CUSUM) Chart

The introduction of the CUSUM chart was driven by the need to detect small changes in the mean value of a process/quality variable. This chart was first proposed by Page (1954) and is a modification of the Sequential Probability Ratio Test (SPRT) introduced by Wald (1947). In a CUSUM chart, a cumulative sum of the deviations between the measurement (or the statistic) and the target value is plotted. Mathematically, if x_n denotes the current measurement and S_{n-1} denotes the cumulative sum of the deviations between the measurements and the target value for the past (n-1) observations, then the statistic for the CUSUM is computed as:

$$S_n = max\{0, S_{n-1} + (x_n - T)\} \quad (5.1)$$

where T is the target value. Equation (5.1) detects if the shift in the process is above the target value. A shift below the target value can be detected by plotting the statistic $S_{n(low)}$:

$$S_{n(low)} = min\{0, S_{n-1(low)} - (x_n - T)\} \quad (5.2)$$

The properties of CUSUM chart have been investigated extensively (Philips, 1969; Hinkely, 1969; 1970; 1971; Moustakides, 1986). One of the attractive properties of the CUSUM chart, which makes it popular, is its optimal property that was proved by Lorden (1971; 1973). This property states that the CUSUM chart minimizes the average delay for a given false alarm rate.

5.3.3 Exponentially Weighted Moving Average (EWMA) Chart

Another popular univariate monitoring scheme is the Exponentially Weighted Moving Average (EWMA) chart which was first introduced in the late 1950's by Roberts (1959). Later, Lucas and Saccucci, (1990) investigated the properties and suggested further enhancements. The EWMA chart is expressed as:

$z(n) = \lambda\, x(n) + (1-\lambda) z(n-1)$	(5.3)

where x(n) and z(n) are the sample value and weighted sum at time n, λ is a scalar lying between 0 and 1 and is known as the weighting parameter. The properties and design procedures for constructing EWMA charts can be found in the literature (Lucas and Saccucci, 1990; Montgomery, 1991; Christer and Wang, 1995)

5.4 Limitations of Univariate Control Charts

The difficulty with using independent control charts for each variable in a multivariate setting can be illustrated with the help of Figure 5.2. Suppose that when the process is running under normal operating conditions, two quality variables, denoted x_1 and x_2, follow a bivariate normal distribution each with mean zero and unit variance. Also, let the two variables be correlated with correlation coefficient, $\rho_{x_1x_2}$ equal to 0.8. A scatter plot of one hundred observations drawn from this bivariate normal population is shown in Figure 5.2(a). The ellipse in Figure 5.2(a) represents the 99% confidence bound for the in-control process. An independent Shewhart chart, with 95% and 99% confidence bounds for each variable is also plotted in Figures 5.2(b) and 5.2(c). It should be noted that following the inspection of both Shewhart charts reveals that the process is in a state of statistical control and none of the observations violate the confidence bound. However, a customer could complain about the quality of product corresponding to observation number 51. If only univariate charts were used for quality control, then this problem cannot be detected as corresponding to this observation the Shewhart charts for both variables are within the confidence bounds. The problem is detetcted in the bivariate plot of x_1 and x_2, where the point corresponding to sample number 51 lies outside the confidence bound which indicates that the quality of this product is *different* from rest of the products.

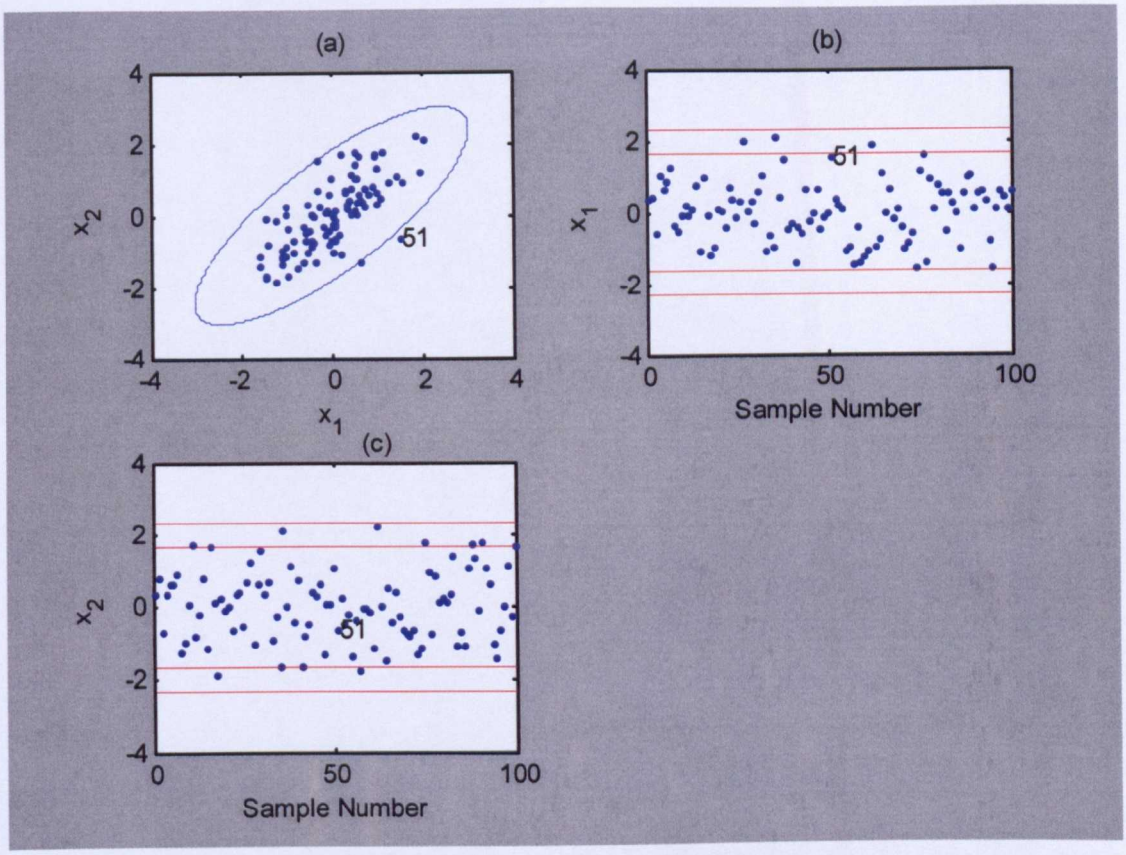


Figure 5.2: Illustration of problem of using independent control charts in a multivariate setting: (a) Scatter plot of two correlated variable with 99% confidence bound (b) Shewhart control chart with 95% and 99 % confidence bounds for x_1 and (c) x_2

Monitoring of each variable independently, in a multivariate setting, can also be misleading even if the variables are independent. If for example the variables, x_1 and x_2 considered above are independent, then the confidence bound for each variable under normal operating conditions, with a given probability of type I error equal to α, is given by:

$$0 - Z_{\frac{\alpha}{2}} \sigma_{x_1} \leq x_1 \leq 0 + Z_{\frac{\alpha}{2}} \sigma_{x_1}$$
$$0 - Z_{\frac{\alpha}{2}} \sigma_{x_2} \leq x_2 \leq 0 + Z_{\frac{\alpha}{2}} \sigma_{x_2} \quad (5.4)$$

where $Z_{\frac{\alpha}{2}}$ is the point of the standard normal distribution such that the probability of standard normal random variable z,

$$prob\left(z \geq Z_{\frac{\alpha}{2}}\right) = \frac{\alpha}{2} \qquad (5.5)$$

and σ_{x_i} is the standard deviation for variable x_i for $i = 1,2$. Since the variables are independent, the probability that both variables lie within their respective confidence bounds (and hence the probability that the process operates in normal operating conditions) is given by:

$$prob(\text{both } x_1 \text{ and } x_2 \text{ lie within their respective confidence bounds}) = (1-\alpha)^2 \qquad (5.6)$$

If $\alpha = 0.05$, then the probability in equation (5.5) is equal to $(1-0.05)^2 = 0.9025$ and therefore the probability of a false alarm is equal to $1 - 0.9025 = 0.0975$. It is therefore, observed that the probability of a false alarm increases from 0.05 to 0.0975 when the two variables are monitored independently. In general, if K variables in a process are monitored independently, the probability of a false alarm is equal to $1-(1-\alpha)^K$. For example, for $K = 10$, this value is 0.40. It can, therefore, be concluded that false alarms are much too frequent if a process consists of a large number of variables and each is monitored independently.

5.5 Multivariate Statistical Process Control

In a typical process industry, a large number of process variables e.g. temperatures, pressures, flows etc. are measured with high sampling frequency. The quality variables on the other hand are available at a much lower frequency. Since the quality of the final product eventually depends on the process variables, it would therefore be advantageous to use the data from the process variables to determine if the process is running under normal conditions. One way to do this is to monitor each process variable independently. But as noted in the previous section, this can be highly misleading. An alternative is to develop a monitoring scheme, where all the variables are dealt with collectively.

One characteristic property of the data collected on process variables is that they are highly (cross) correlated. This is because only a few independent events drive the whole process. To take into consideration the (cross) correlation, subspace projection techniques (PCA and

PLS) are commonly used to model the process data. The advantages of using projection techniques to model correlated data, as mentioned in Chapter 2, include dimensionality reduction and noise filtering. It is therefore highly desirable if these techniques could also be used for process monitoring. In the section given below, PCA based monitoring of a multivariate process is described.

5.5.1 Principal Component Analysis based Process Monitoring Scheme

The idea behind using projection techniques to monitor a process is to examine the behaviour of data in a subspace defined by a reduced number of variables (known as latent variables or principal components). For example in PCA, if a vector $\mathbf{x} \in \mathrm{R}^K$ is projected onto A ($A < K$) principal components, then the subspace is defined by the set of orthogonal variables, $t_1, t_2 ... t_A$:

$$t_i = \mathbf{x}^T \mathbf{p}_i \quad \text{for } i = 1, 2 ... \ A \tag{5.7}$$

Once the subspace is defined, a statistic is then defined to detect any abnormal deviation in the subspace. One commonly used statistic is Hotelling T^2, which in general, is defined as:

$$T^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{5.8}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix of vector $\mathbf{x}$. Since the latent variables which define the subspace are orthogonal, the definition of Hotelling T^2 in equation 5.8, for detecting changes in a subspace reduces to:

$$T^2 = \mathbf{t}^T \boldsymbol{\Sigma}_t^{-1} \mathbf{t} = \sum_{i=1}^{A} \frac{t_i^2}{s_i^2} \tag{5.9}$$

where $\mathbf{t}$ is a vector of principal components $\mathbf{t} = [t_1 \ t_2 ... t_A]^T$, $\boldsymbol{\Sigma}_t$ is the covariance matrix of $\mathbf{t}$ and s_i is the standard deviation of the i^{th} principal component. It should be noted that in equation (5.8) it is assumed that each variable is mean centred.

Aside from keeping track of deviations in the subspace, it is also important to monitor the residuals between the actual observations and that predicted by the projections onto the subspace. The residual vector **e** is defined as:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{5.10}$$

where $\hat{\mathbf{x}}$ is the component predicted by the PCA model. A statistic that is used to monitor the residuals is known as the Q-statistic and is equal to the sum of squares of the components of the residual vector **e**. Mathematically, it is given by:

$$Q = \mathbf{e}^T\mathbf{e} \tag{5.11}$$

Figure 5.3 shows a geometrical interpretation of Hotelling T^2 and Q-statistic. The ellipse shows the subspace spanned by two principal components PC1 and PC2. Hotelling T^2 measures the square of the distance of a point (marked * in the figure), within the subspace from the origin (or in general from the mean value). On the other hand, the Q-statistic measures the square of the distance of a point (marked **o** in the figure) orthogonal to the subspace spanned by the principal components.

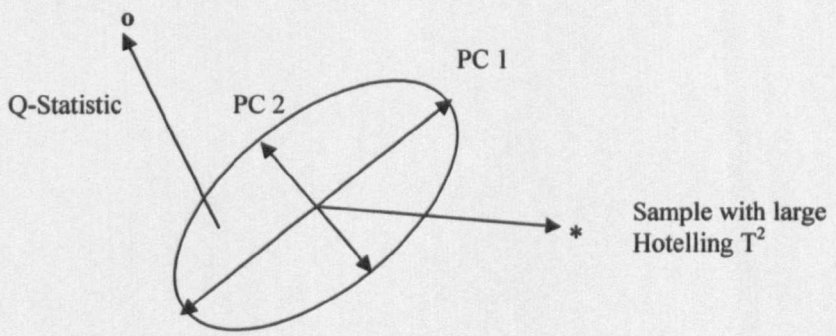


Figure 5.3: Geometrical interpretation of Hotelling T^2 and the Q-statistic

To design and analyse a change detection algorithm, the distribution functions of Hotelling T^2 and the Q-statistic are required. It has been proven(Mardia et al., 1979; Jackson, 1991) that Hotelling T^2 follows an F distribution:

$$T^2 = \frac{A(N-1)}{N-A} F_{A,N-A} \tag{5.12}$$

where N is the number of observations and $F_{A,N-A}$ is a F-distribution function with A and N-A degrees of freedoms. The confidence limit for the Q-statistic can be shown to be equal to (Jackson and Mudholkar, 1979):

$$Q = \theta_1 \left[1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + \frac{c\sqrt{2\theta_2 h_0^2}}{\theta_1} \right]^{\frac{1}{h_0}} \tag{5.13}$$

where $\theta_1 = \sum_{i=A+1}^{K} \lambda_i$, $\theta_2 = \sum_{i=A+1}^{K} \lambda_i^2$, $\theta_3 = \sum_{i=A+1}^{K} \lambda_i^3$ and $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$

5.5.2 Literature Review

Initial applications of PCA for the monitoring of quality variables were reported by Hotelling (1947; 1957), Jackson (1956; 1959; 1980) and Jackson and Morris (1957). In these approaches only the quality variables of the products were monitored. The application of projection techniques to monitor the process variables was first reported in the late 1980's and the beginning of 1990's. Kresta et al., (1989; 1991) demonstrated the application of PCA and PLS for monitoring simulated data collected from a fluidized bed reactor. Numerous papers on applications of PCA and PLS for the monitoring of continuous processes have been reported since (MacGregor et al., 1991; MacGregor, 1994; MacGregor and Kourti, 1995; Martin et al., 1996; Kourti and MacGregor, 1994; 1995). The application of these techniques to real industrial application have also been reported (Piovoso and Kosanovich, 1991; 1992; 1994; Kourti, et al., 1996; Morud, 1996; Wikstrom et al., 1998). To diagnose the cause of the occurrence of abnormal events in the process, use is made of contribution plots (Miller et al., 1993).

Some chemical processes e.g. pharmaceutical, operate in batch mode rather than in continuous mode. The applications of PCA for the monitoring of batch processes have also been reported in the literature. Since the pioneering work of MacGregor and Nomikos (MacGregor and Nomikos, 1992; MacGregor et al, 1994; Nomikos and MacGregor, 1994; 1995) in using PCA for monitoring batch processes, several modifications have been

proposed. Wold et al., (1998) proposed a different way of unfolding the batch data to that used by Nomikos and MacGregor. Louwerse and Smilde (1999) introduced the use of PARAFAC and three way models to monitor a batch process. Applications of PCA based scheme to monitor batch process in real industrial applications have been reported (Lennox et al., 2000; 2001)

The classical PCA based monitoring scheme assumes that the data is collected in a two dimensional matrix with a linear relationship between the variables and that statistical independence exists between the observations. These assumptions are not always valid. To overcome these limitations, a number of alternative PCA based monitoring scheme have been proposed. The non-linear relationship between the variables is dealt with through the application of non-linear PCA which can be implemented using neural networks (Kramer, 1991) and can be further used in monitoring applications (Jia et al., 1998; 2000). Furthermore to account for serial correlation, dynamic version of PCA has been proposed and used in process monitoring Ku et.al., (1995).

The data measured in a typical process does not correspond to one scale. This is because the events occurring in a process occupy different regions in the time-frequency (or time-scale) space (Bakshi, 1999). To account for the multiscale nature of the data, multiscale PCA using wavelets was developed and used in process monitoring (Bakshi, 1998; Shao et.al., 1999).

In a practical situation, slow and normal changes can occur in a process. If the process is monitored with a fixed model, it will give rise to false alarms. The time varying nature of the process has been taken into consideration in yet another version of PCA known as adaptive or recursive PCA (Li et al., 2000; Lane et al., 2003) in which the PCA model is updated after every observation or after a block of observations.

5.6 Conclusions

In this chapter a brief literature review of statistical process monitoring has been presented. It is shown that independent monitoring of process variables in a multivarite process can be misleading. It is therefore, recommended to use multivariate monitoring schemes which handle all the variables collectively. Since, subspace projection techqniues are useful in identifying a compact model from the cross correlated measurements of process variables, it is desirable to extend their application to process monitoring. Two statistics, Hotelling T^2

and the Q-statistic, are used in a PCA based performance monitoring scheme. A literature review for the application of PCA and PLS in process monitoring was also undertaken

Process monitoring forms the basis of the next two chapters. It is shown that a monitoring scheme based on Hotelling T^2 and the Q-statistic in PCA and PLS is particularly insensitive to a class of changes which lead to a change in the covariance structure of the process variables. Two new monitoring methods are then proposed to these changes. In Chapter 6, the focus is on PCA and the statistic is derived from the theory of PCA model identification. In Chapter 7, a PLS based monitoring scheme is considered. A recursive algorithm for PLS is first derived and a monitoring statistic is developed.

CHAPTER 6

Detection of Changes in Covariance Structure

6.1 Introduction

Assuming the distribution of the process variables to be multivariate Gaussian, the process is completely characterized by the mean and variance-covariance matrix of the process variables. In this situation it is possible to distinguish between two classes of changes depending on whether the change affects the mean or variance-covariance structure of the process variables. Although not standard nomenclature, changes affecting the mean value of one or more of the process variables is termed class 1 and those that affect the variance-covariance structure of the process variables are denoted, class 2. A class 1 change can occur if, for example, a (constant) sensor bias is present whilst the second class is associated with fluctuations (larger than what is observed normally) about the mean value of the variable.

The conventional monitoring scheme for detecting changes in the normal operating conditions of a process using subspace projection techniques is based on two statistics, Hotelling T^2 and the Q-statistic. The poor sensitivity of these statistics to detect small changes in the variance- covariance structure of the process variables (class 2 changes) has previously been reported in the literature (Kano et al., 2001). Although some work (Kano et al., 2001), has been proposed to detect these changes more efficiently, there is still a need for an algorithm that detects small changes with limited delay. The aim of this chapter is to propose an algorithm which is "nearly optimal" in terms of the detection of the second class of changes. An optimal algorithm is defined as the one that detects a given change with the smallest possible delay for a given false alarm rate

6.2 Literature Review

It should be recalled that the parameters of a PCA model, the loading vectors, depend on the variance-covariance structure of the process data. More specifically, the loading vectors are the eigenvectors of the variance-covariance matrix. The problem of abnormal change detection in the variance-covariance structure of the process is thus equivalent to the detection of abnormal changes in the PCA model parameter vector denoted by θ.

One of the desirable characteristics of a monitoring statistic is that it captures the complete information encapsulated within the data. A statistic with this characteristic is known as a sufficient statistic (Basseville and Nikiforov, 1993). It is known, particularly in the context of single-input single-output systems, that the prediction error based statistic, for example the sum of the squares of the prediction error, is not sufficient for the detection of abnormal changes in the parameters of the system. This result, the formal proof of which is given in Basseville and Nikiforov (1993) forms the basis of a series of papers (Benveniste et al., 1987; Zhang, et al, 1994; Basseville, 1998) where research has been undertaken to identify a sufficient statistic that detects abnormal changes in the parameters of the system. In the works cited, the monitoring statistic is derived from system identification algorithms. While Benveniste et al., (1987) derived the monitoring statistic from a recursive algorithm for the estimation of parameters, Zhang et al., (1994) extended this work to include non-recursive algorithms in this framework.

The poor sensitivity of Hotelling T^2 and the Q-statistic to detect abnormal changes in the variance-covariance structure (and hence the PCA model parameters) was first reported in the chemometrics literature by Kano et al., (2001). Since the Q-statistic is the sum of squares of the prediction errors, the poor sensitivity of the Q-statistic based on the work of Basseville and Nikiforov (1993) can be easily understood. Kano et al., (2001), in their paper proposed a new scheme based on determining the revised loading vectors for a moving window and then calculating the 'distance' between the new loading vectors and the reference loading vectors as determined under normal operating conditions. The distance was quantified using the dot product between the calculated and the reference loading vectors. There are limitations associated with this method. The first is that a number of statistics are required to monitor the system. For example, consider a process that includes 4 variables, consequently the covariance matrix is of order 4×4 and hence there are 4 loadings vectors. Kano et al., (2001) proposed monitoring this process by calculating the dot product between the i^{th} new loading vector and its corresponding normal condition loading vector, therefore, 4 metrics require to be monitored for the detection of changes in the loading vectors. In addition to these four metrics, they also proposed monitoring the combined subspace spanned by different combinations of the loadings. In this simple example, two additional statistics for monitoring the subspace spanned by the first two and first three loading vectors are additionally derived, therefore, the total number of metrics to monitor a process comprising 4 variables is 6. A clear limitation of this scheme is that it is not efficient for the monitoring of a process consisting of a large number of variables.

Secondly, the parameters of the model, namely the loadings, require to be estimated on-line to determine the 'distance' between the new parameters and the reference parameters. A more straight forward approach would be if the change in the parameter vector of the model could be detected by using the 'distance' between the reference model parameter vector and the data, that is, there is no requirement to re-estimate the model parameters.

Finally, the determination of the confidence limits for the statistic has not been addressed. Kano et al., (2001) determined the confidence limits by calculating the statistic over a large number of data sets. Consequently the determination of the confidence limits of the statistic where the number of data sets is limited, which is the situation most often, is not viable.

6.3 Poor Sensitivity of Hotelling T^2 and the Q-statistic: An Intuitive Explanation

Consider a hypothetical process which has (say) 6 correlated process variables. Applying a PCA model to the process and retaining 3 principal components, the model comprises 6×3 =18 parameters (each loading vector being 6-dimensional). Now suppose the covariance structure of the process variables has changed which results (in general) in a change in all the loading vectors. This change, therefore, takes place in an 18-dimensional vector space. However, the residual vector of the process considered is 6-dimensional and, therefore, may not capture the "real" extent of the change. Since the Q-statistic is based on the sum of the squares of the residuals, the statistic will be less sensitive, in particular, to small changes in the variance-covariance structure.

The poor sensitivity of Hotelling T^2 can be understood with the help of Figure 6.1. Recall that Hotelling T^2 measures the distance of projection from the origin (under the assumption that the data is mean centred) within the subspace spanned by the loading vectors. For the purpose of illustration, a two dimensional space is considered. Under normal operating conditions, let the loading vectors be $\mathbf{p}_1$ and $\mathbf{p}_2$ (Figure 6.1). Hotelling T^2 for samples generated from this population (labelled population 1) defines a limit for this data with certain confidence (95 or 99% (Figure 6.1)). Now suppose the covariance structure changes, the loading vectors are now $\mathbf{p}_1'$ and $\mathbf{p}_2'$, such that the new data is represented by a smaller ellipse in Figure 6.1. Since the smaller ellipse lies within the larger ellipse, Hotelling T^2 will not give a alarm despite the fact that the covariance structure has changed.

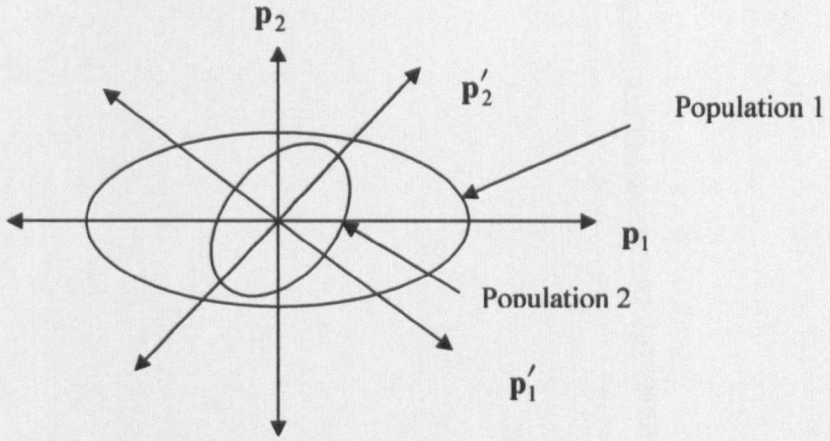


Figure 6.1: Graphical illustration of the poor sensitivity of Hotelling T^2 to a change in variance-covariance structure

6.4. A New Monitoring Statistic

Let the process be characterized by parameter vector Θ. Now consider this parameter vector more specifically. Assume that the process variables are multivariate Gaussian and are independent and identically distributed and recalling that a multivariate Gaussian distribution is completely characterized by its mean vector and covariance matrix, the parameter vector Θ will collectively represent the true (population) mean and covariance matrix. Since in this study changes in the mean value are not considered, it is assumed that that the mean value of the variables is known (it is taken as zero without loss of generality), consequently the parameter vector Θ represents the variance-covariance matrix of the process variables and determines the behaviour of the process. Suppose that under normal operating conditions $\Theta = \Theta_0$ and when Θ takes values other than Θ_0, abnormal system behaviour is indicated. The problem of change or abnormality detection in a system can be formulated in the framework of a hypotheses testing problem. Given a set of observations $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$, at time n, from a process with parameter vector Θ, it is necessary to decide whether to reject the null hypothesis H_0:

H_0: $\Theta = \Theta_0$ for t =1,2, . . . n	(6.1)
H_1: $\exists$ an instance r $(1 \le r \le n)$ such that $\begin{cases} \Theta = \Theta_0 & \text{for } t = 1,2,\ldots r\text{-}1 \\ \Theta \neq \Theta_0 & \text{for } t = r, r+1,\ldots n \end{cases}$	

r is the sampling instance at which the fault occurs.

Let the PCA representation for a process be characterized by the parameter vector $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is essentially the loading matrix of the PCA representation arranged in a vector form (columns placed one above the other). Note that $\boldsymbol{\Theta}$ and $\boldsymbol{\theta}$ are normally not equal and may belong to vector spaces of different dimensions (for example, in a process with 5 variables the covariance matrix is of order 5×5 and $\boldsymbol{\Theta}$ is a vector $\in R^{25}$. If a PCA representation is built using 3 principal components, say, then $\boldsymbol{\theta}$ is a vector of order R^{15}). When the process operates under normal operating conditions, let $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Assuming a mapping $\mathbf{f}$ exists between the true process parameter vector $\boldsymbol{\Theta}$ and the PCA model parameter vector $\boldsymbol{\theta}$, that is:

$$\mathbf{f}(\boldsymbol{\Theta}) = \boldsymbol{\theta} \quad \forall \quad \boldsymbol{\Theta} \text{ and } \boldsymbol{\theta} \tag{6.2}$$

the abnormality detection problem equation (6.1) can be reformulated as:

$$H_0: \mathbf{f}(\boldsymbol{\Theta}) = \boldsymbol{\theta}_0 \text{ for } t = 1,2.....n \tag{6.3}$$

$$H_1: \exists \text{ an instance } r \ (1 \le r \le n) \text{ such that}$$

$$\begin{cases} \mathbf{f}(\boldsymbol{\Theta}) = \boldsymbol{\theta}_0 & \text{for } t = 1,2,\ldots r\text{-}1 \\ \mathbf{f}(\boldsymbol{\Theta}) \neq \boldsymbol{\theta}_0 & \text{for } t = r, r+1,\ldots n \end{cases}$$

To determine the new statistic, recall that the first loading vector of the PCA model under normal operating conditions is determined by maximizing the variance of the latent variable t_1, that is:

$$\mathbf{p}_1 = \arg\max_{\mathbf{p}} \left[E\{t_1^2\} - \lambda\left(\mathbf{p}^T\mathbf{p} - 1\right) \right] \tag{6.4}$$

Using the properties associated with finding the maxima of a function, the loading vector $\mathbf{p}_1$ is given by the solution of the following equation:

$$\mathbf{p}_1 = \arg \frac{\partial}{\partial \mathbf{p}} [E\{t_1^2\} - \lambda(\mathbf{p}^T\mathbf{p} - 1)] = \mathbf{0} \tag{6.5}$$

Taking the differential operator inside the statistical expectation operator:

$$\mathbf{p}_1 = arg[\mathrm{E}\{2\,\mathrm{t}_1\mathbf{x}\} - 2\lambda\mathbf{p}] = arg[\mathrm{E}\{2\,\mathrm{t}_1\mathbf{x} - 2\lambda\mathbf{p}\}] = \mathbf{0} \qquad (6.6)$$

and letting

$$\mathbf{k}_1 = 2\,\mathrm{t}_1\mathbf{x} - 2\lambda\mathbf{p} \qquad (6.7)$$

gives

$$\mathbf{p}_1 = \arg \mathrm{E}\{\mathbf{k}_1\} = \mathbf{0} \qquad (6.8)$$

This can be written as:

$$\mathrm{E}\{\mathbf{k}_1\}\big|_{\mathbf{p} = \mathbf{p}_1} = \mathbf{0} \qquad (6.9)$$

Also in the neighbourhood of $\mathbf{p}_1$, $\omega(\mathbf{p}_1)$ (which does not contain $\mathbf{p}_1$):

$$\mathrm{E}\{\mathbf{k}_1\}\big|_{\mathbf{p} \in \omega(\mathbf{p}_1)} \neq \mathbf{0} \qquad (6.10)$$

From equations (6.9) and (6.10), it follows that if the (first) loading vector of the PCA model remains equal to $\mathbf{p}_1$, the mean of the statistic $\mathbf{k}_1 = 2\,\mathrm{t}_1\mathbf{x} - 2\lambda\mathbf{p}_1$ is equal to zero. A non-zero value for the mean of $\mathbf{k}_1$ indicates that the (first) loading vector of the model is no longer equal to that determined under normal operating conditions. It is worth noting that λ in equation (6.7) for $\mathbf{k}_1$ is the eigenvalue corresponding to the first loading (Chapter 2, section 2.1). The second and higher loading vectors of the PCA model are determined in a similar way.

Corresponding to each loading, $\mathbf{p}_i$, a corresponding statistic $\mathbf{k}_i = 2\,\mathrm{t}_i\mathbf{x} - 2\lambda_i\mathbf{p}_i$ can be determined such that the mean of $\mathbf{k}_i = \mathbf{0}$, when $\mathbf{p}_i$ relates to normal operating conditions. The mean thus deviates from zero when the loading vector drifts away from normal operating

conditions. If all the loading vectors are arranged into one column, the vector, $\boldsymbol{\theta}_0$, of the parameters corresponding to the normal operating conditions for a PCA model is given by:

$$\boldsymbol{\theta}_0 = [\mathbf{p}_1 \mathbf{p}_2 \cdots \mathbf{p}_A]^T \tag{6.11}$$

The corresponding augmented vector $\mathbf{k}$ of the statistics is:

$$\mathbf{k} = [\mathbf{k}_1 \ \mathbf{k}_2 \ldots \mathbf{k}_A]^T \tag{6.12}$$

Since each component of the vector **k** has zero mean when the corresponding loading vector relates to normal operating conditions, it follows that the mean of the augmented vector **k** is zero when the PCA model parameter vector $\boldsymbol{\theta}_0$ corresponds to normal operating conditions. When any, or all, of the loading vectors change, the mean of vector **k** deviates from zero. The problem of detecting changes in the PCA model, therefore, reduces to detecting a change in the mean of **k**.

For the design and analysis of a change detection algorithm based on the statistic **k**, the underlying probability density function is required. Determination of the distribution function of **k** is not easy to determine theoretically. To overcome this problem, the local approach of hypotheses testing is considered.

6.5. Local Approach to Hypothesis Testing: An Introduction

The basic statistic for detecting a change in the parameter vector from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}$ is the log-likelihood ratio (Basseville and Nikiforov, 1993):

$$\mathrm{LR}_n(\boldsymbol{\theta}_0, \boldsymbol{\theta}) = ln \frac{p_{\boldsymbol{\theta}}(\mathbf{X}_n)}{p_{\boldsymbol{\theta}_0}(\mathbf{X}_n)} \tag{6.13}$$

where $\mathbf{X}_n$ is a matrix containing observations from time point 1 to n, $p_{\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}_0}$ are the probability density functions with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$ respectively and *ln* is the natural logarithm. Although known to be a sufficient statistic, the problem with the log-likelihood ratio statistic is that its distribution function is difficult to determine for all probability

density functions, p_θ. One solution is to assume that the parameters θ and θ_0 are 'close' to each other, that is, $\theta = \theta_0 + \frac{\gamma}{n}$, where γ is a fixed but unknown vector and its magnitude (divided by the sample size, n) represents the amount by which the parameter vector θ_0 has changed. For a large sample size, the parameter vector θ lies close (or locally) to θ_0 and the approach to testing a hypothesis under this assumption is known as the local approach of hypothesis testing. Mathematically, this approach decides between the null and alternative hypothesis which are defined as:

$H_0: \theta = \theta_0$ for $t = 1,2.....n$

$H_1: \exists$ an instance r $(1 \le r \le n)$ such that

$$\begin{cases} \theta = \theta_0 & \text{for } t = 1,2,\ldots r\text{-}1 \\ \theta = \theta_0 + \frac{\gamma}{n} & \text{for } t = r, r+1,\ldots n \end{cases}$$

(6.14)

Assuming the local approach, the log of the distribution function, p_θ, can be expanded around, p_{θ_0} using second order Taylor expansion:

$$ln\, p_\theta(.) \approx ln\, p_{\theta_0}(.) + \frac{\gamma^T}{\sqrt{n}} \frac{\partial(ln\, p_\theta(.))}{\partial\theta}\bigg|_{\theta=\theta_0} + \frac{1}{2}\frac{\gamma^T}{\sqrt{n}} \frac{\partial^2(ln\, p_\theta(.))}{\partial\theta} \frac{\gamma}{\sqrt{n}}\bigg|_{\theta=\theta_0} \quad (6.15)$$

The log-likelihood ratio can be expanded by substituting equation (6.15) into equation (6.13)

$$LR_n(\theta_0, \theta) \approx \frac{\gamma^T}{\sqrt{n}} \frac{\partial(ln\, p_\theta(.))}{\partial\theta}\bigg|_{\theta=\theta_0} + \frac{1}{2}\frac{\gamma^T}{\sqrt{n}} \frac{\partial^2(ln\, p_\theta(.))}{\partial\theta} \frac{\gamma}{\sqrt{n}}\bigg|_{\theta=\theta_0} \quad (6.16)$$

Using the definitions of *efficient score*, $\mathbf{z}_n$ and *information matrix*, $\mathbf{I}_n$ (Basseville and Nikiforov, 1993):

$$\mathbf{z}_n(\boldsymbol{\theta}_0) = \frac{1}{\sqrt{n}} \frac{\partial(ln\, p_{\boldsymbol{\theta}}(.))}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{6.17}$$

$$\mathbf{I}_n(\boldsymbol{\theta}_0) = -\frac{1}{n} \frac{\partial^2(ln\, p_{\boldsymbol{\theta}}(.))}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \tag{6.18}$$

the log-likelihood ratio in equation (6.16) can be re-written as:

$$LR_n(\boldsymbol{\theta}_0, \boldsymbol{\theta}) \approx \boldsymbol{\gamma}^T \mathbf{z}_n(\boldsymbol{\theta}_0) - \frac{1}{2} \boldsymbol{\gamma}^T \mathbf{I}_n(\boldsymbol{\theta}_0) \boldsymbol{\gamma} \tag{6.19}$$

The distribution function of the log-likelihood ratio in equation (6.19) was determined by Cam (1986) by proving the following central limit theorem:

$$\begin{aligned} LR_n(\boldsymbol{\theta}_0, \boldsymbol{\theta}) &\rightarrow G\left(0.5\, \boldsymbol{\gamma}^T \mathbf{I}(\boldsymbol{\theta}_0)\, \boldsymbol{\gamma},\ \boldsymbol{\gamma}^T \mathbf{I}(\boldsymbol{\theta}_0) \boldsymbol{\gamma}\right) \quad \text{under } p_{\boldsymbol{\theta}_0} \\ &\rightarrow G\left(-0.5\, \boldsymbol{\gamma}^T \mathbf{I}(\boldsymbol{\theta}_0)\, \boldsymbol{\gamma},\ \boldsymbol{\gamma}^T \mathbf{I}(\boldsymbol{\theta}_0) \boldsymbol{\gamma}\right) \text{ under } p_{\boldsymbol{\theta}} \end{aligned} \tag{6.20}$$

where $G(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

It can be seen from equation (6.20) that a log-likelihood ratio has a Gaussian distribution under both normal and modified conditions and a deviation in the parameter vector is reflected as a change in the sign of the mean value of the log-likelihood ratio. It is also important to note that the variance under both conditions is the same. The local approach thus has transformed the problem from the detection of a change in the parameter vector to the detection of a change in the mean value of a Gaussian random variable.

Similar to the expansion of the log-likelihood ratio, under the local hypothesis, the efficient scores can also be expanded using Taylor series expansion (Basseville and Nikiforov, 1993):

$$\mathbf{z}_n(\boldsymbol{\theta}) = \mathbf{z}_n(\boldsymbol{\theta}_0) + \frac{1}{n} \frac{\partial^2\, ln\, p_{\boldsymbol{\theta}}(.)}{\partial \boldsymbol{\theta}^2}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \boldsymbol{\gamma} \tag{6.21}$$

Based on the condition that the maximum likelihood principle is used for parameter identification, the central limit theorem for the efficient scores states that (Basseville and Nikiforov, 1993):

$$\mathbf{z}_n(\boldsymbol{\theta}_0, \boldsymbol{\theta}) \rightarrow G(\mathbf{0}, \mathbf{I}(\boldsymbol{\theta}_0)) \text{ under } p_{\boldsymbol{\theta}_0}$$
$$\rightarrow G(\mathbf{I}(\boldsymbol{\theta}_0)\boldsymbol{\gamma}, \mathbf{I}(\boldsymbol{\theta}_0)) \text{ under } p_{\boldsymbol{\theta}} \qquad (6.22)$$

Thus a change in the parameter vector is reflected as a change in the mean value of the efficient scores with covariance matrix remaining the same under both process states.

6.5.1 Generalization to other Monitoring Functions

Although the expansion of the log-likelihood ratio under the local approach has been known since 1980's, it is the expansion of the efficient scores (equations 6.21 and 6.22) that has led to the recent popularity of the local approach. An important result, (Benveniste et al., (1987); Zhang et al., 1994) was established whereby it was shown that the central limit theorem in equation (6.22) holds not only for the efficient scores but for a large class of estimating functions (a function $\mathbf{k}(\boldsymbol{\theta}, .)$ is termed an estimating function for the parameter vector $\boldsymbol{\theta}_0$ if the parameter vector $\boldsymbol{\theta}_0$ is equal to the roots of the equation $\mathbf{k}(\boldsymbol{\theta}, .) = \mathbf{0}$). Such estimating functions when used for change detection are known as primary residuals (Basseville, 1997). The conditions for a finite dimensional vector-valued function $\mathbf{k}(\boldsymbol{\theta}, .)$ to be primary residuals are:

1. Average value of $\mathbf{k}(\boldsymbol{\theta}, .)$ should be equal to zero when $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, that is

   $E_{\boldsymbol{\theta}}\{\mathbf{k}(\boldsymbol{\theta}, .\} = \mathbf{0}$ when $\boldsymbol{\theta} = \boldsymbol{\theta}_0$

2. Average is non-zero when $\boldsymbol{\theta}$ is different from $\boldsymbol{\theta}_0$

   $E_{\boldsymbol{\theta}}\{\mathbf{k}(\boldsymbol{\theta}, .\} \neq \mathbf{0}$ when $\boldsymbol{\theta} \neq \boldsymbol{\theta}_0$

For a PCA model, the function $\mathbf{k}$ defined in equation 6.12 satisfies both conditions and therefore is a valid primary residual. The distribution function of $\mathbf{k}$ can be determined by generalizing equation (6.22). Specifically if $\mathbf{k}(.)$ is a primary residual then Benveniste et al., (1987) and Zhang et al., (1994) proved that the function $\mathbf{r}_n$, also known as improved residuals, and defined as:

$$\mathbf{r}_n = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \mathbf{k}(t) \qquad (6.23)$$

satisfies the following central limit theorem under the local approach of hypothesis testing:

$$\begin{aligned} \mathbf{r}_n(\boldsymbol{\theta}_0, \boldsymbol{\theta}) &\rightarrow G\big(\mathbf{0},\ \mathbf{M}(\boldsymbol{\theta}_0)\big) && \text{under } p_{\boldsymbol{\theta}_0} \\ &\rightarrow G\big(\mathbf{M}(\boldsymbol{\theta}_0)\,\boldsymbol{\gamma},\ \mathbf{M}(\boldsymbol{\theta}_0)\big) && \text{under } p_{\boldsymbol{\theta}} \end{aligned} \qquad (6.24)$$

where $\mathbf{M}$ is a Jacobian matrix defined as $\mathbf{M} = E_{\boldsymbol{\theta}_0}\left\{ \frac{\partial \mathbf{k}(\boldsymbol{\theta}, .)}{\partial \boldsymbol{\theta}} \right\}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$.

Once the improved residuals vector, $\mathbf{r}_n$, is determined, the optimal test for detecting change is given by computing a scalar S_n:

$$S_n = \mathbf{r}_n^T \, \boldsymbol{\Sigma}_r^{-1} \, \mathbf{r}_n \qquad (6.25)$$

where $\boldsymbol{\Sigma}_r$ is the covariance matrix of the improved residuals under normal operating conditions. The decision rule for detecting a change is given by:

Decide in favour of H_1 if $S_n > t_0$

Decide in favour of H_0 if $S_n \leq t_0$

where t_0 is a threshold and is determined by noting that (under the null hypothesis) S_n is χ^2 distributed with degrees of freedom equal to the dimension of $\mathbf{r}$.

It should be noted from equation (6.23) that the size of the window over which $\mathbf{r}_n$ is computed, tends to infinity as time increases. From a practical point of view, it has been proposed (Zhang et al., 1994) to compute $\mathbf{r}_n$ over a fixed size window, that is, a fixed value n_0 is selected such that:

$$\mathbf{r}_n = \frac{1}{n_0 + 1} \sum_{t=n-n_0}^{n} \mathbf{k}(t) \qquad (6.26)$$

The choice of n_0 is determined as a compromise between a large false alarm rate and the magnitude of the delay in detection. It is known (Zhang et al., 1994) that a smaller window size (that is a larger value of n_0) reduces the number of false alarms but introduces a delay in change detection. A larger size of window increases the speed of change detection but results in a higher false alarm rate.

Also since at the start of the algorithm, the value of n is small, and since the local approach is asymptotic, it is proposed (Zhang et al., 1994) that the algorithm starts after the first n_1 samples, where n_1 is suitably selected and is generally of the order of 30-50 samples.

6.6 Summary of the Algorithm

The steps of the local approach based scheme are now summarized:

Given: Matrix **X** of size $N \times K$, containing N observations on K variables corresponding to the normal operating condition of a process

Mean centre and scale each variable to unit variance.

Step 1: Build a PCA model using A principal components

Step 2: Compute the primary residuals for each principal component and each sample time point: $\mathbf{k}_i(t) = 2\, t_i(t)\, \mathbf{x}(t) - 2\lambda_i \mathbf{p}_i$, where $\mathbf{x}(t)$ is the observation vector at sample time point t, λ_i eigenvalue corresponding to i^{th} loading $\mathbf{p}_i$ for $i = 1$ to A

Step 3: Determine the augmented vector $\mathbf{k}(t) = [\mathbf{k}_1(t)\ \mathbf{k}_2(t) \ldots \mathbf{k}_A(t)]^T$

Step 4: Remove the bias, i.e, mean centre $\mathbf{k}(t)$

Step 5: Select the window parameter n_0 (Typical value is in the range 300-500 samples) and n_1 (30-50) samples.

Step 6: Compute the improved residuals at each sample time:

$$\mathbf{r}_n = \frac{1}{n_0 + 1} \sum_{t=n-n_0}^{n} \mathbf{k}(t)$$

Step 7: Calculate the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{r}}$ of the improved residuals

Step 8: Compute the local statistics at each sample time:

$$S_n = \mathbf{r}_n^T \boldsymbol{\Sigma}_{\mathbf{r}}^{-1} \mathbf{r}_n$$

Step 9: Determine the confidence limits (95%, 99%), t_0

Step 10: If there are large numbers of false alarms, change the window parameter n_0 and repeat steps 6-9

Step 11: Finally apply the algorithm to new (experimental) data set by scaling it using the same values that were used to scale the nominal data set.

6.7 Simulation Studies

The methodology described above is tested first on two artificial data sets and is then applied to detect abnormal changes in the performance of a continuous stirred tank reactor.

6.7.1 Example 1

In this example a normal data set comprising 2000 samples and two variables was generated from a population of zero mean and covariance matrix:

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 55.25 & -1.57 \\ -1.57 & 18.50 \end{bmatrix} \tag{6.27}$$

An experimental data set consisting of 2000 samples was then generated with the first one thousand samples drawn from the normal population and the second one thousand samples, corresponding to a faulty data set, drawn from a population with zero mean and covariance matrix

$$\Sigma_f = \begin{bmatrix} 18.50 & 1.57 \\ 1.57 & 55.25 \end{bmatrix} \quad (6.28)$$

It should be noted that the eigenvectors of Σ_f

$$\mathbf{p}_{1f} = [0.0426 \quad 0.9910]^T$$
$$\mathbf{p}_{2f} = [-0.9910 \quad 0.0426]^T \quad (6.29)$$

are 90-degrees rotated with respect to the corresponding eigenvectors of original covariance matrix Σ_0 whose eigenvectors are

$$\mathbf{p}_1 = [-0.9910 \quad 0.0426]^T$$
$$\mathbf{p}_2 = [-0.0426 \quad -0.9910]^T \quad (6.30)$$

The faulty data therefore correspond to a modified covariance structure in which the eigenvectors have rotated through 90-degrees from the eigenvectors corresponding to the normal operating mode. After the normal data is auto-scaled, a PCA model with one principal component was built explaining 53.4% of the total variance. The local approach based algorithm described in the previous section was then applied to the experimental data set. The size of the window was tuned to 300 and the value of n_1 was adjusted to 50. The plot of the statistic, *S,* for the experimental data set is shown in Figure 6.2(a). Figure 6.2(b) shows a plot of the statistic for the first one thousand samples (corresponding to the normal operating conditions) of the experimental data set. The sample number at which the change is detected is 1052 and hence a delay of 52 samples in detecting the change is incurred.

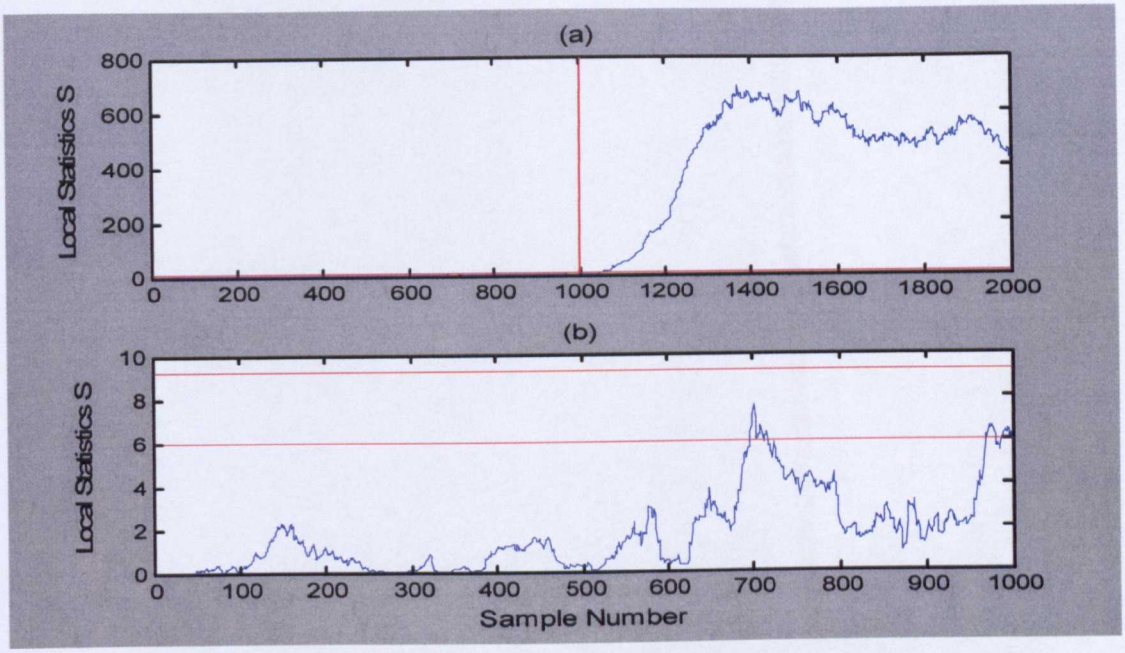


Figure 6.2: Plot of local statistic versus sample number for (a) the whole experimental data set and (b) the normal operating condition component of the experimental data set, when one principal component is retained in the PCA model (example 1).

The plots of Hotelling T^2 and Q-statistic for the experimental data set are shown in Figures 6.3(a) and 6.3(b) respectively. It can be seen by comparing Figures 6.2 and 6.3 that while the conventional monitoring scheme based on Hotelling T^2 and the Q-statistic fail to detect the change, the local approach based scheme successfully identifies the change in the covariance structure.

The procedure is now repeated by increasing the number of principal components in the PCA model to two and thus the model accounts for 100% of the variance of the nominal data set. The plot of the local statistic, *S*, for the experimental data set is shown in Figure 6.4(a) with Figure 6.4(b) showing a zoomed-in portion of Figure 6.4(a) corresponding to the plot of the statistic *S* for the first one thousand samples of the experimental data. The plots of Hotelling T^2 and Q-statistic are shown in Figure 6.5. It can be seen from Figure 6.4 that the local approach based statistic detects the change at sample point 1034 and therefore the delay in detecting the change is 34 sample points.

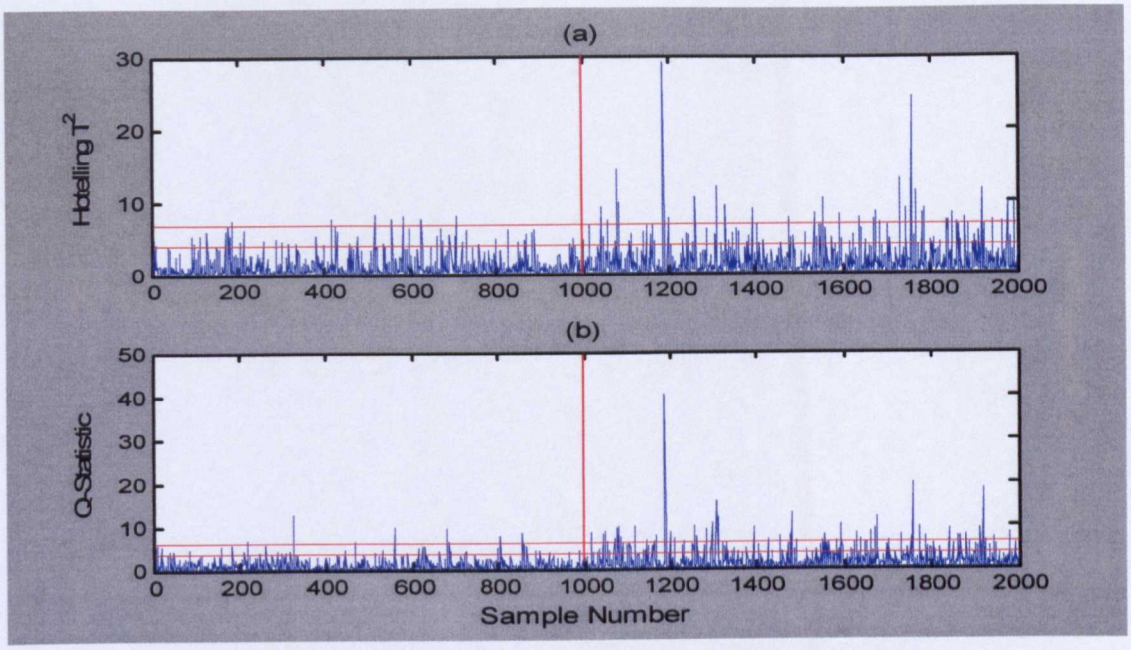


Figure 6.3: Plot of (a) Hotelling T^2 and the (b) Q-statistic for the experimental data set, when one principal component is retained in the PCA model (example 1).

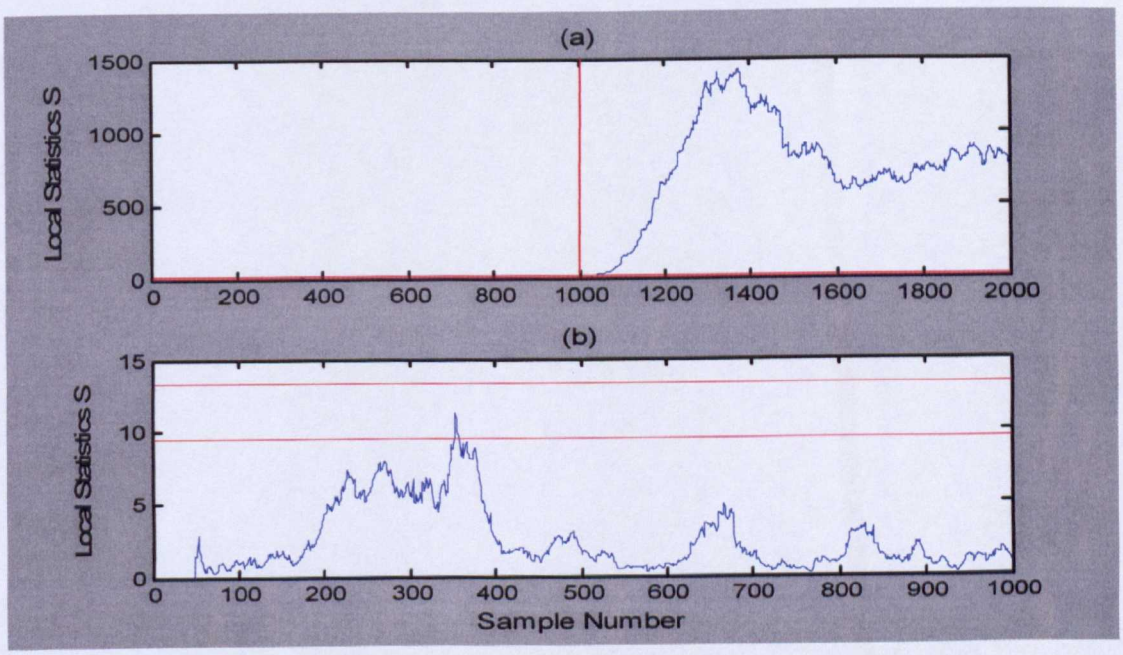


Figure 6.4: Plot of local statistic versus sample number for (a) the total experimental data set and (b) the normal operating condition component of the experimental data set, when two principal components are retained in the PCA model (example 1).

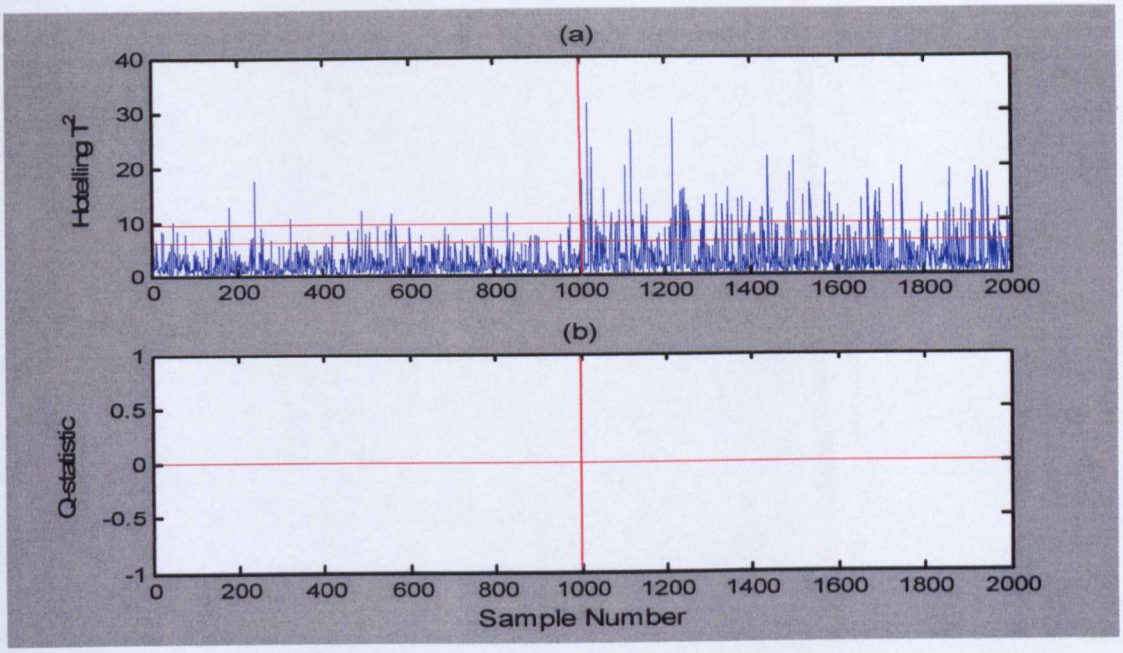


Figure 6.5: Plot of (a) Hotelling T^2 and (b) Q-statistic for the experimental data set, when two principal components are retained in the PCA model (example 1).

In the conventional monitoring scheme, although Hotelling T^2 shows an upward shift after the introduction of the change, it is not sufficient to identify the occurrence of a change in the process. It is also interesting to note that the Q-statistic remains equal to zero both before and after the occurrence of the change. While the Q-statistic is expected to be equal to zero before the change as 100% variability of the data is explained by the PCA model, a value exactly equal to zero even after the change is not obvious. This can be explained by recalling the fact that if $\mathbf{p}_1$ and $\mathbf{p}_2 \in R^2$ form an orthonormal basis of the vector space R^2, then any vector $\mathbf{x} \in R^2$ can be written as :

$$\mathbf{x} = t_1\mathbf{p}_1 + t_2\mathbf{p}_2 \qquad (6.31)$$

where t_1 and t_2 are the projection of vector $\mathbf{x}$ onto $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively. Therefore, even if $\mathbf{x}$ and $\mathbf{x}'$ belong to different populations (normal and faulty respectively in the example above), both of them can written as a linear combination of the same basis functions (the loadings of the PCA model). That is, the vector $\mathbf{x}'$ can be written as:

$$\mathbf{x}' = t'_1\mathbf{p}_1 + t'_2\mathbf{p}_2 \qquad (6.32)$$

where t_1' and t_2' are the projections of $\mathbf{x}'$ on $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively. Since the (prediction) error in both cases is zero, the Q-statistic is zero not only for normal operating conditions but also for the faulty condition.

6.7.2 Simulation Example 2

Consider a 2×2 process described by the following state and measurement equations. This model is taken from Ku et al., (1995) and was considered by Kano et al., (2001). By utilising this model, a comparison between the performance of the proposed monitoring scheme and that reported by Kano et al., (2001) is possible.

$$\mathbf{x}(t) = \begin{bmatrix} 0.811 & 0.226 \\ 0.477 & 0.415 \end{bmatrix} \mathbf{x}(t-1) + \begin{bmatrix} 0.193 & 0.689 \\ 0.320 & 0.749 \end{bmatrix} \mathbf{x}(t-1)$$

$$\mathbf{u}(t) = \begin{bmatrix} 0.118 & 0.191 \\ 0.847 & 0.264 \end{bmatrix} \mathbf{u}(t-1) + \begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{bmatrix} \mathbf{x}(t-1) \qquad (6.33)$$

$$\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{h}(t)$$

where $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y} \in R^2$ are the state, input and output vectors respectively, $\mathbf{e}$ and $\mathbf{h}$ are zero mean Gaussian random vectors comprising two independent random variables. The variance of each random variable in $\mathbf{e}$ is unity and for $\mathbf{h}$ is 0.1.

Kano et al., (2001) simulated abnormal changes in the parameters of the above system by changing the coefficient relating the second state variable, u_2 , to the first input, x_1, (the value of this coefficient under normal operating condition is 3). Three changes, small, medium, and large were considered which correspond to change from 3.0 to 2.5, 2.0 and 1.0 respectively. These changes are summarized in Table 6.1. The objective is to compare the proposed monitoring scheme with the conventional PCA based monitoring scheme before comparing it with the scheme proposed by Kano et al., (2001).

Table 6.1: Abnormal changes in the artificial system

Case	Type	Size
1 (Small)	Change of parameter from x_1 to u_2	$3.0 \rightarrow 2.5$
2 (Medium)	Change of parameter from x_1 to u_2	$3.0 \rightarrow 2.0$
3 (Large)	Change of parameter from x_1 to u_2	$3.0 \rightarrow 1.0$

6.7.2.1 Monitoring using Static PCA

The system described in equation (6.26) is a dynamic system and therefore a dynamic model would be most appropriate to model the data generated from this system. However, as a first step, monitoring of this system based on a static PCA model is first studied. Two thousand measurements corresponding to the normal operating conditions of four variables namely two output and two input variables are collated into a matrix. The data is auto-scaled and PCA was performed. Table 6.2 lists the percentage contribution of each principal component to the total variance of the data.

Table 6.2: Variance contribution for static PCA (example 1)

Number of PC	Eigenvalue	% variance explained	Cumulative % variance explained
1	1.9478	48.69	48.69
2	1.3408	33.52	82.21
3	0.6489	16.22	98.43
4	0.0624	1.57	100.00

A PCA model using three principal components (selected using cross-validation) was built. A further data set (experimental data) comprising two thousand samples was generated with the first one thousand corresponding to normal operating conditions and the remaining one thousand corresponding to an abnormal change in the value of the parameter from 3.0 to 2.5 (case1, small). The proposed monitoring scheme based on the local approach was applied with the window parameter n_0 and n_1 tuned to 350 and 50 respectively. The plot of the local statistics for the experimental data set is shown in Figure 6.6(a) with Figure 6.6(b) showing the plot of the statistic for the first one thousand samples of the experimental data set. The performance of conventional monitoring scheme is given in Figure 6.7. The procedure is repeated for the other two changes (medium and large) and the corresponding

plots for the local statistics for the medium and large changes are shown in Figures 6.8 and 6.10 with Figures 6.9 and 6.11 showing the corresponding performance of the conventional monitoring scheme based on these changes. The following conclusions can be drawn.

First, the local approach based scheme is able to detect all three changes with delays of 29, 23, and 18 samples for the small, medium and large changes respectively. The conventional monitoring scheme, on the other hand, is almost insensitive to the small and medium changes but the Q-statistic for the large change does show an upward shift but is not sufficient to give a clear indication of the change. Secondly, there are some false alarms both in the proposed monitoring scheme and in the conventional monitoring scheme. For the conventional monitoring scheme they can be attributed to the serial correlation in the data but for the local based monitoring scheme, false alarms are due to the fact that the local approach is asymptotic, that is, it assumes (ideally) an infinite data set but practice the data set is finite. False alarms can be reduced (1) by tuning the window size parameter n_0 appropriately or (2) Zhang et al., (1994) also suggested increasing the theoretical confidence bound by an 'appropriate' amount to account for the asymptotic nature of the local approach. For example, the theoretical limit calculated by Zhang et al., (1994) for an example given in their paper was 26.21 but they 'upgraded' the limit to 40 to reduce the false alarm rate.

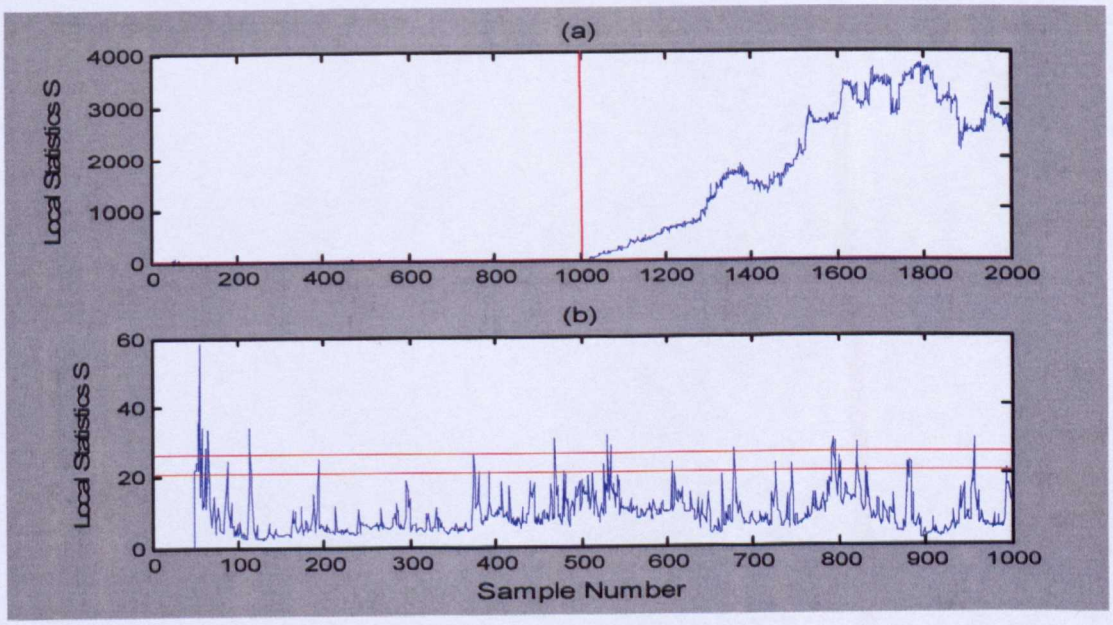


Figure 6.6: Plot of the local statistics versus sample number for a static PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (b) the normal operating condition component of the experimental data, (example 2)

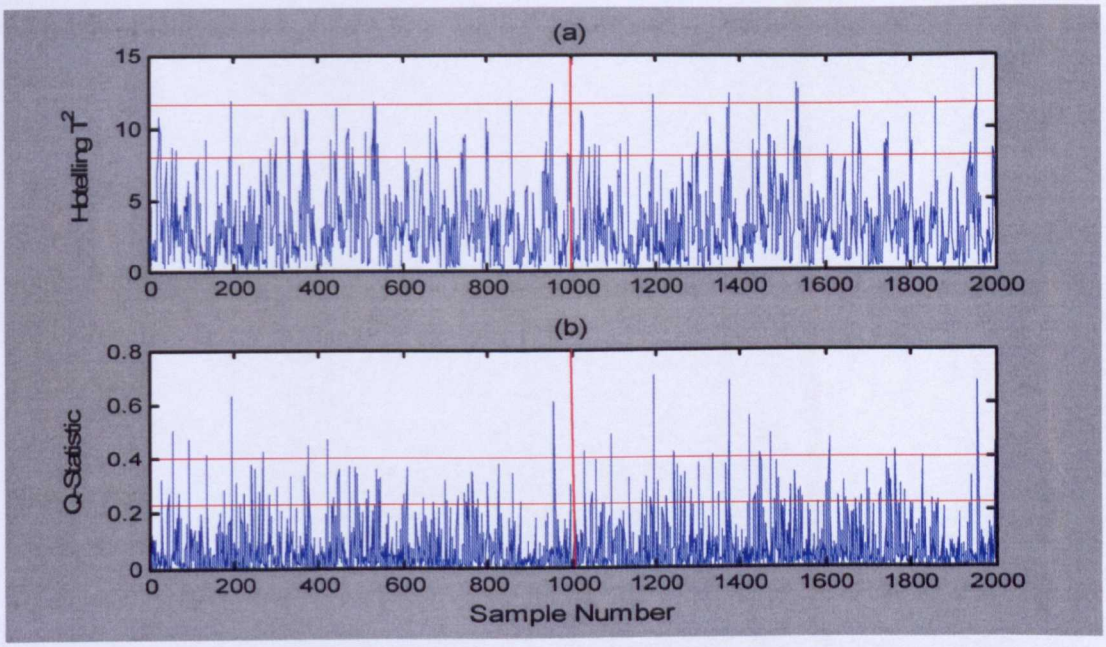


Figure 6.7: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for a static PCA based conventional monitoring scheme when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (example 2)

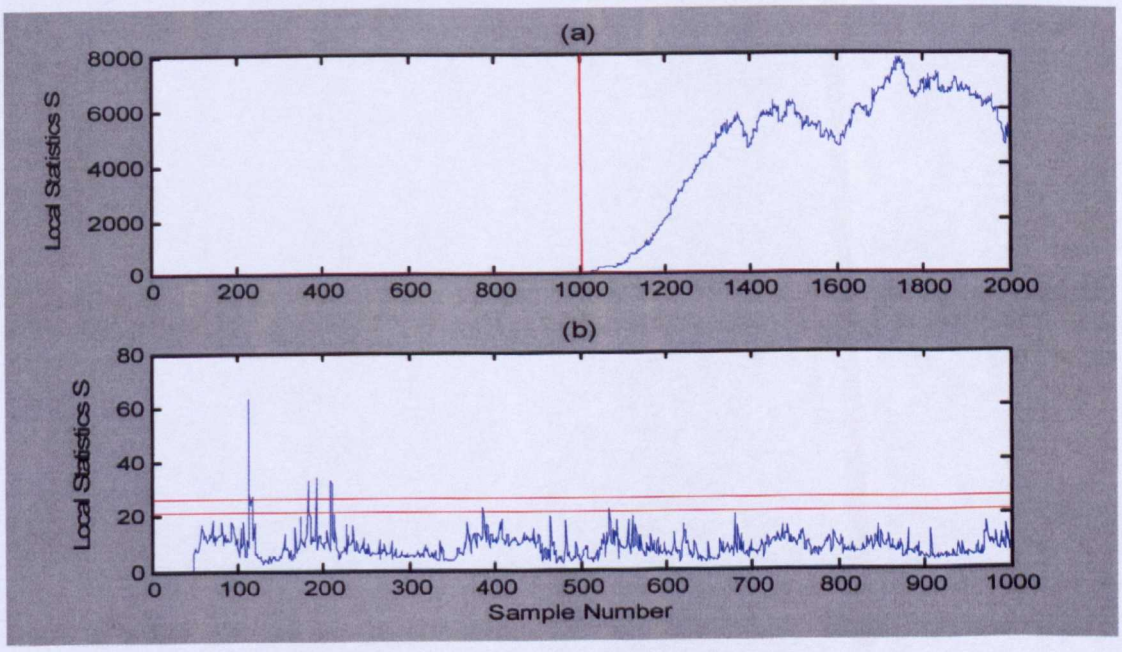


Figure 6.8: Plot of the local statistics versus sample number for a static PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 2)

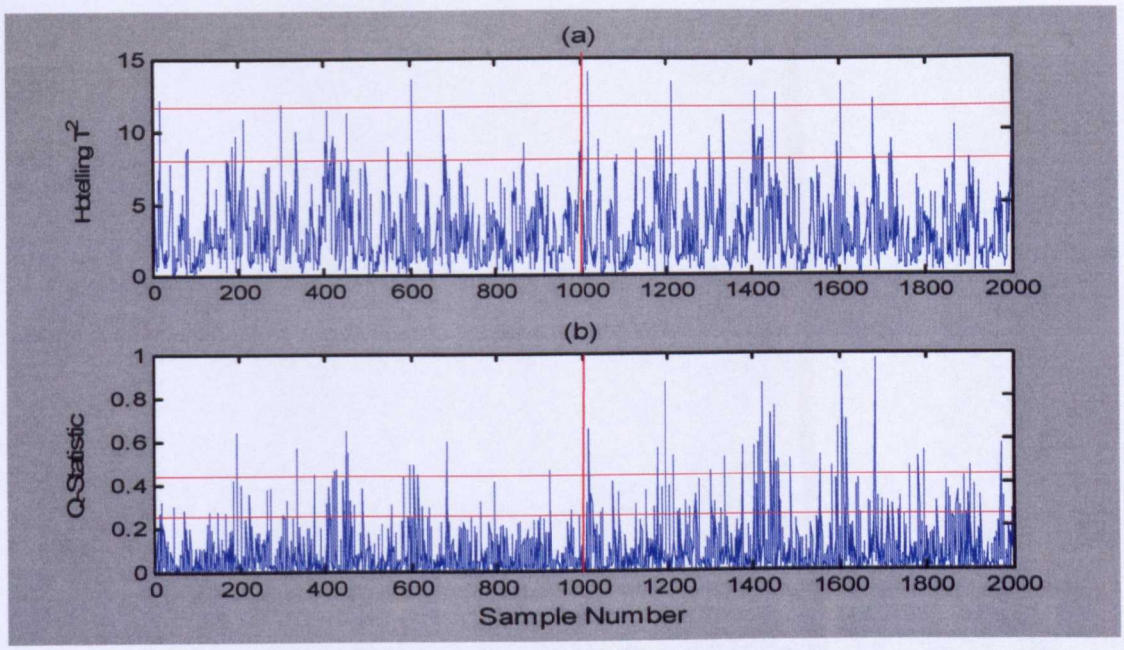


Figure 6.9: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for a static PCA based monitoring scheme when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (example 2)

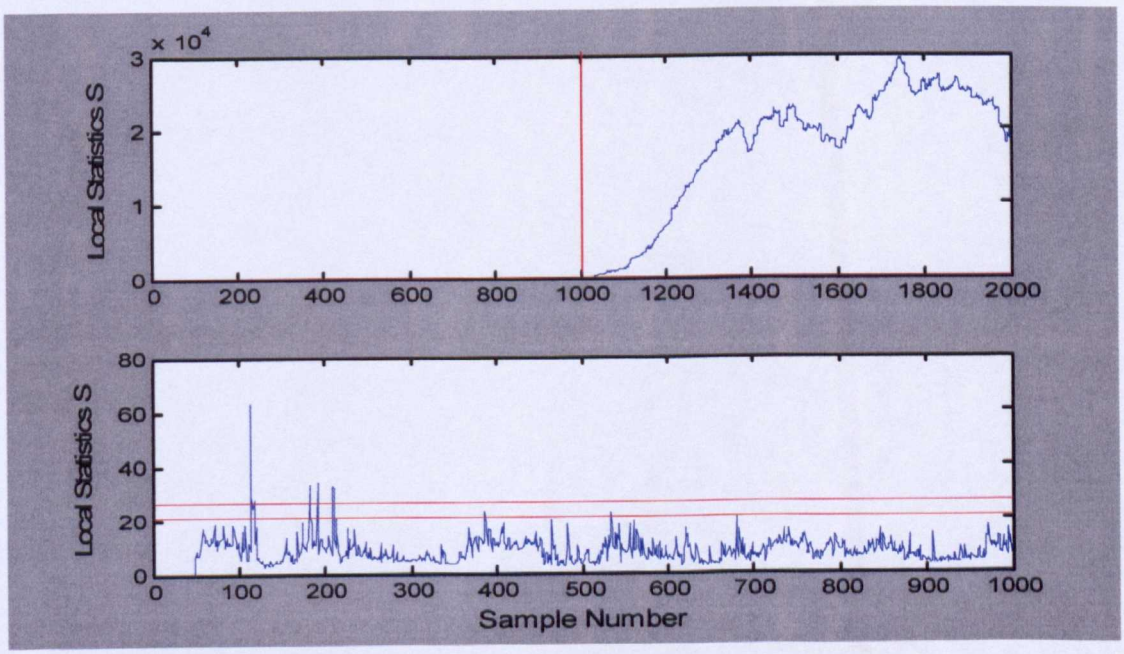


Figure 6.10: Plot of the local statistics versus sample number for a static PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 2)

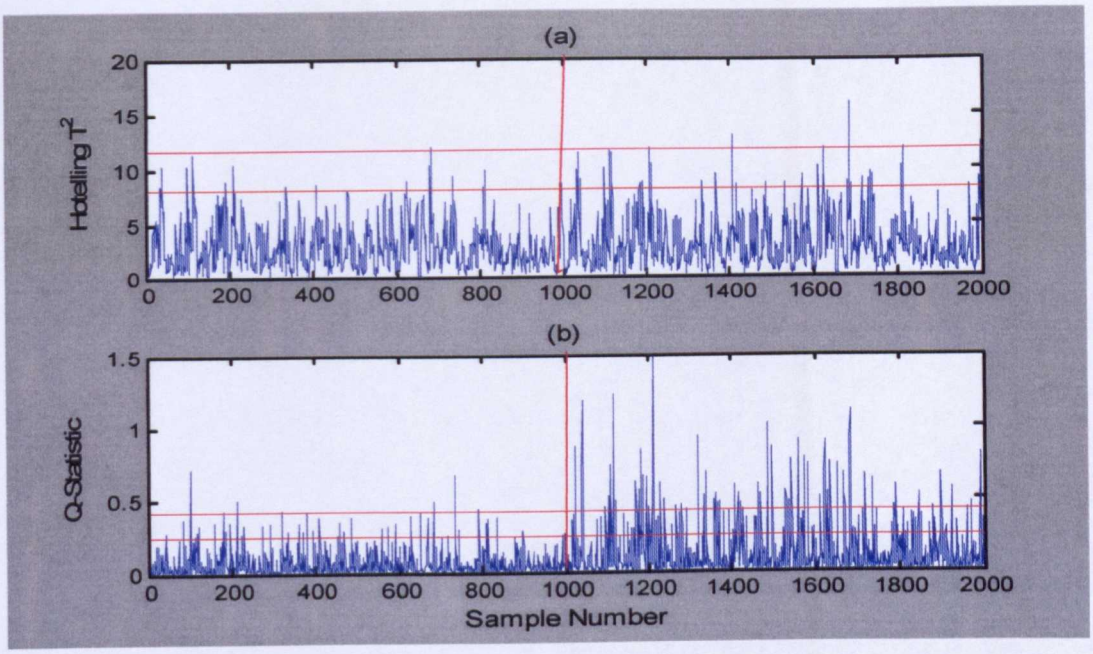


Figure 6.11: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for a static PCA based conventional monitoring scheme when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (example 2)

To compare the proposed scheme with that proposed by Kano et al., (2001), the following procedure was used (Kano et al., 2001).

1. Data is collected from the process when operating under normal conditions and the control limits for the monitoring statistic are calculated for a given confidence (95% and 99%).
2. For the data generated after the occurrence of the fault, the percentage of samples lying outside the control limit is calculated for each simulation. This percentage is termed 'reliability' and depends on the number of samples used for the calculation. In the reported study, 100 samples were considered.
3. The average reliability for the 1000 data sets is calculated for each case. This produces a performance index for the monitoring scheme.

The average reliability reported by Kano et al., (2001) for the conventional (static) PCA based monitoring scheme for the three changes mentioned are summarised in Table 6.3.

Table 6.3: Average Reliability (%) for the static PCA based conventional MSPC scheme

Monitoring statistic	Case		
	1	2	3
Hotelling T^2	1.2	1.3	2.0
Q-statistic	1.6	3.2	9.5

It is clear from Table 6.3 that the conventional monitoring scheme is poor in terms of detecting the different levels of change, with the maximum average reliability being less than 10%. To improve the reliability, Kano et al., (2001) proposed monitoring the change in the covariance structure by monitoring the 'distance' between the eigenvectors (loadings) of the new (experimental) data collected from a moving window and the reference (nominal) eigenvectors. The distance was measured using the dot product between the new and the reference loading vectors. Since the reliability depends on the window size, Kano et al., (2001) reported the average reliability for two window sizes (100 and 200 samples). The maximum average reliability (where the maximum is calculated over window size) for this scheme for each of the three changes is given in Table 6.4. It is seen that although the reliability has improved considerably for large change (case 3), it is still low for the small change (case 1). To see how the local approach based scheme performs in comparison to the scheme of Kano et.al (2001), the average reliability for the local approach is calculated for the three changes and is given in Table 6.5

Table 6.4: Average reliability (%) for the static PCA based scheme of Kano et al., (2001)

Monitoring statistic	Case		
	1	2	3
Proposed by Kano et.al (2001)	17.3	50.2	75.2

Table 6.5: Average reliability (%) for the static PCA based local monitoring scheme

Monitoring statistic	Case		
	1	2	3
Based upon Local Approach	76.9	83.4	87.28

From Tables 6.4 and 6.5, it can be observed that the average reliability for the local approach based scheme has improved over the approach proposed by Kano et al., (2001). More importantly, is that, there is a much greater increase (over 4 times) in the reliability for the

small change. This illustrates the fact that the local approach is especially suitable for detecting small changes in a system. This can be explained from the theory of the local approach (section 6.5) which assumes that the normal and changed parameters are 'close to one another'.

It is also important to recall that the statistic S_n, which essentially detects a change in the mean value of the improved residuals $\mathbf{r}_n$, is an asymptotically optimal statistic (Basseville and Nikiforov, 1993). But since, in practical situations, the sample size is finite and $\mathbf{r}_n$ is calculated by summing the primary residuals over a finite window of size n_0, the algorithm loses its optimal properties. Experience shows that the algorithm works well when the window size n_0 lies in the range 300 or higher. An analytical study of the effect of window size on the optimal property, however, needs to be undertaken. This is identified in chapter 8 as an area of future work.

6.7.2.2 Monitoring using Dynamic PCA

A static PCA model assumes that the observations collected are statistically independent. Since the data used in this example is generated from a dynamic system, the observations are serially correlated as is also evident from Figure 6.12 which shows the autocorrelation function plot for each of the four variables. Ku et al., (1995) addressed the issue of serial correlation by including lagged variables in the observation matrix and then applying PCA. The number of lagged variables to be included can be decided by determining how many past observations influence the current observation. If $\mathbf{x}(t)$ denotes the current observation vector, then the number of lagged variables is equal to *a,* which is determined such that an autoregressive (AR) model of order *a* is a 'good' fit to the data:

$$\mathbf{x}(t) = \sum_{i=0}^{a} \mathbf{D}_i \mathbf{x}(t-i) + \mathbf{e}(t) \qquad (6.27)$$

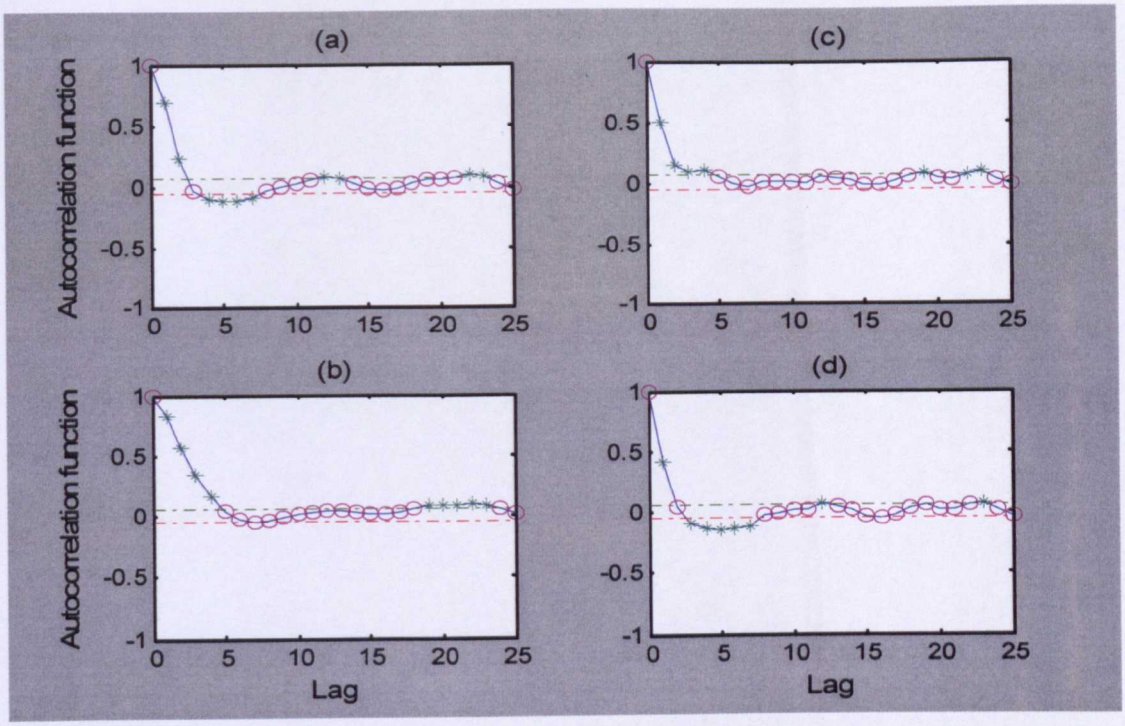


Figure 6.12: Autocorrelation function plots for the four measured variables (example 2)

There exist different criteria (Ljung, 1999) for selecting the order of an autoregressive model. One such criterion is the log of Akaike's Final Prediction Error (AFPE) (Neumaier and Schneider, 2001). The plot of AFPE for an AR model order 0 to 10 is shown in Figure 6.13. It can be seen that although the AFPE is a minimum at model order 2, there is not a significant decrease in the value of AFPE when the model order increases from 1 to 2. It is, therefore, decided to include one lagged variable in the observation matrix to reduce the cost (number of parameters) of the model.

A data set comprising four variables and two thousand samples corresponding to the normal operating conditions are collated into matrix **X**. The matrix is then augmented with one lagged value of each variable so that the size of the augmented matrix $\mathbf{X}_{aug}$ is 1999×8. The matrix is scaled to unit variance and zero mean and PCA is performed. The percentage variance contribution of each principal component to the total variance of the data is listed in Table 6.6. A PCA model is built using four principal components. This was identified using cross-validation. Three additional (experimental) data sets each consisting of 2000 samples are generated in which the first one thousand samples correspond to normal conditions and the remaining one thousand correspond to one of three changes. Figures 6.14, 6.16 and 6.18 show the plots of the results following the application of the local statistics to the

experimental data with the results of the conventional monitoring scheme given in Figures 6.15, 6.17 and 6.19. It is seen from the figures that the local approach based scheme successfully detects all three changes. The delays for the small, medium and large change are 2, 2 and 1 samples respectively. The conventional monitoring scheme is almost insensitive to small and medium changes but the Q-statistic for the large change clearly indicates the occurrence of a change

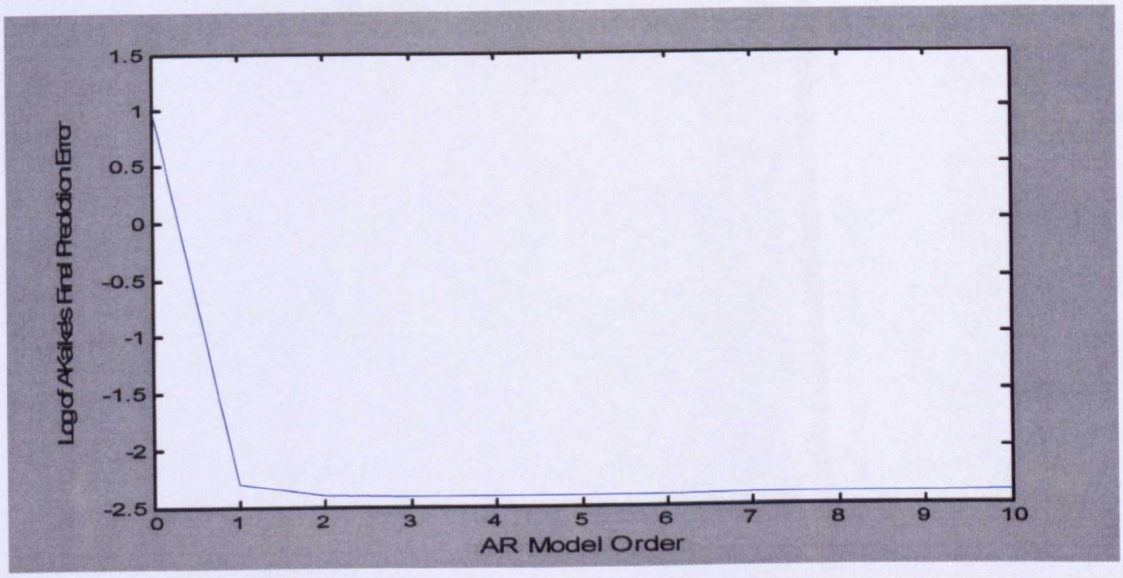


Figure 6.13: Plot of logarithm of Akaike's Final Prediction Error (FPE) versus model order (example 2)

Table 6.6: Variance contribution for dynamic PCA

Number of PC	Eigenvalue	% variance explained	Cumulative % variance explained
1	3.399	42.49	42.49
2	2.807	35.10	77.59
3	0.9589	11.99	89.57
4	0.7109	8.89	98.46
5	0.1085	1.36	99.82
6	0.0107	0.13	99.85
7	0.0024	0.03	99.98
8	0.0013	0.02	100.00

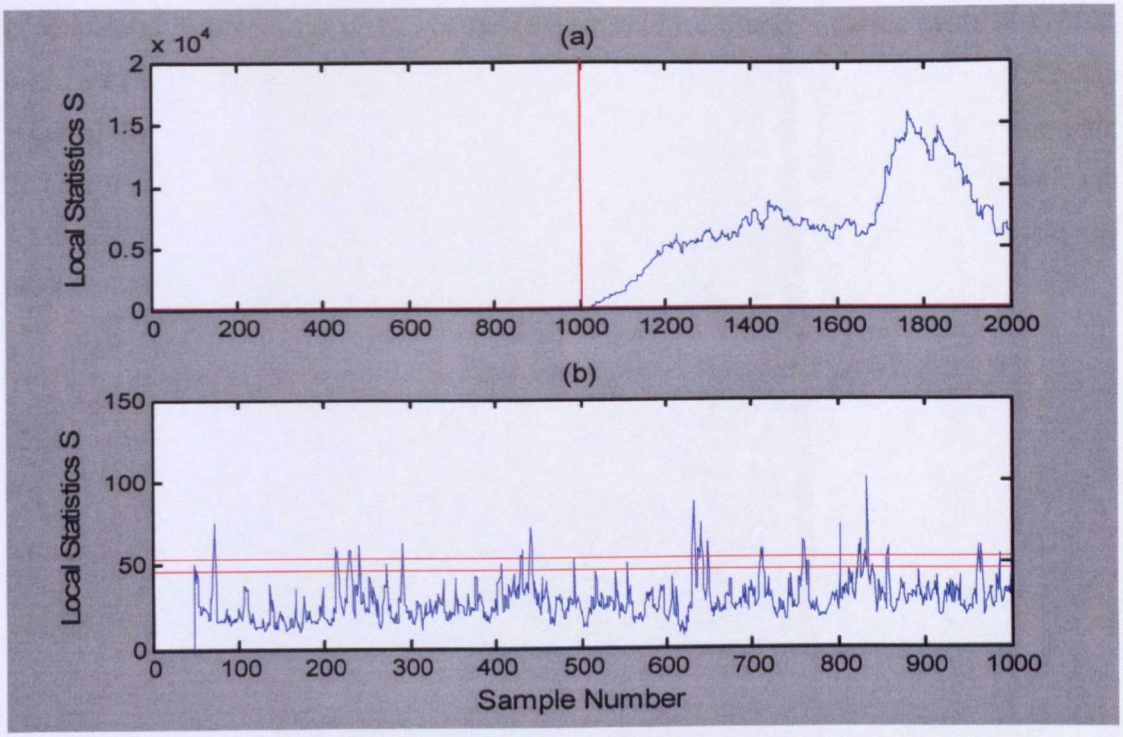


Figure 6.14: Plot of the local statistics versus sample number for the dynamic PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (b) the normal operating condition component of the experimental data (example 2)

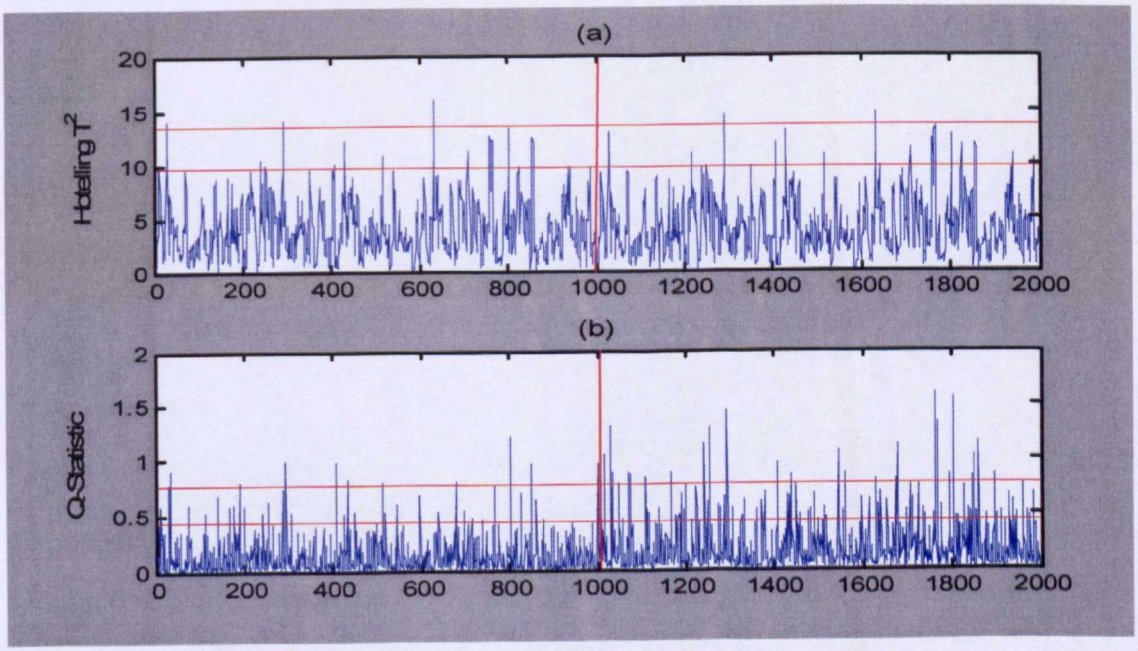


Figure 6.15: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for the dynamic PCA based conventional monitoring scheme when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (example 2)

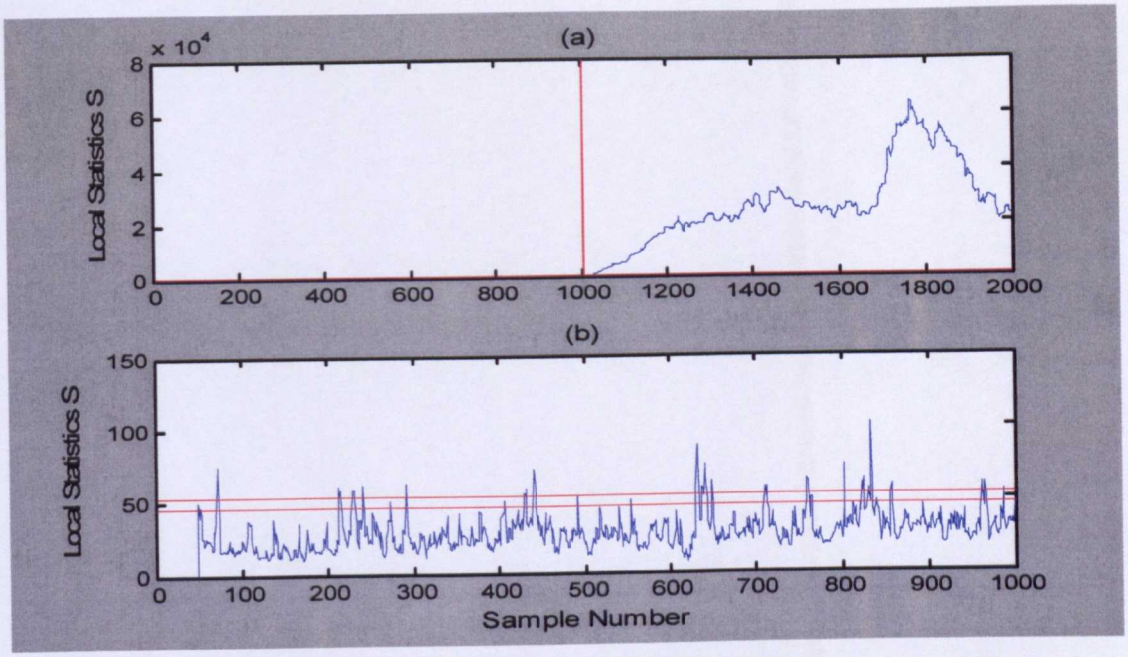


Figure 6.16: Plot of the local statistics versus sample number for the dynamic PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 2)

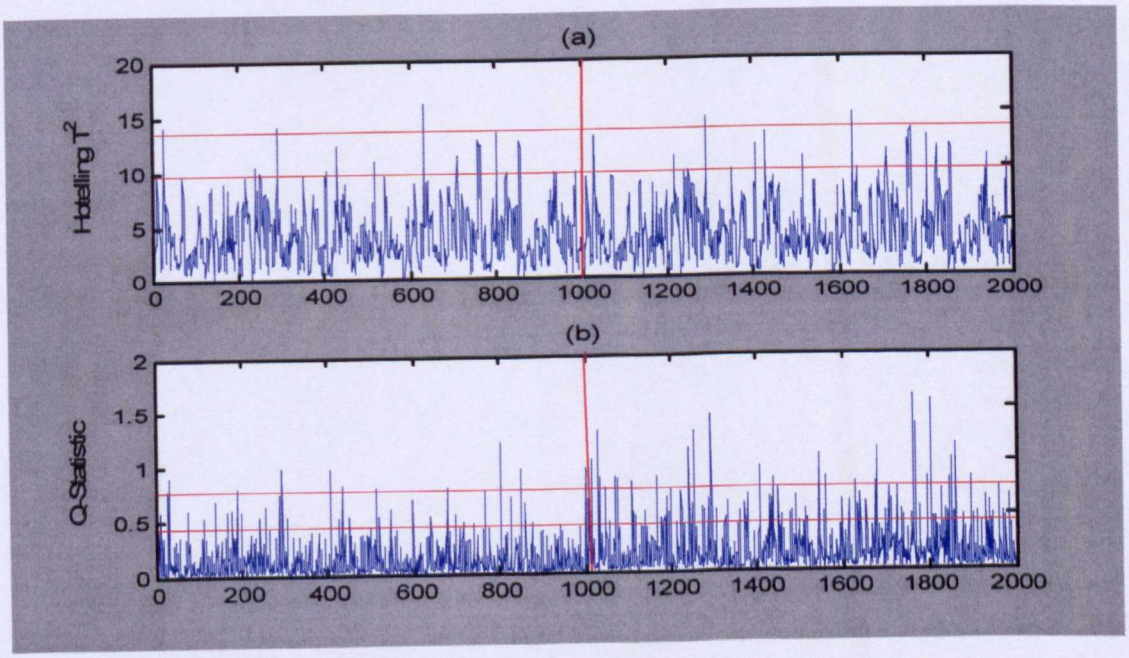


Figure 6.17: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for the dynamic PCA based conventional monitoring scheme when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (example 2)

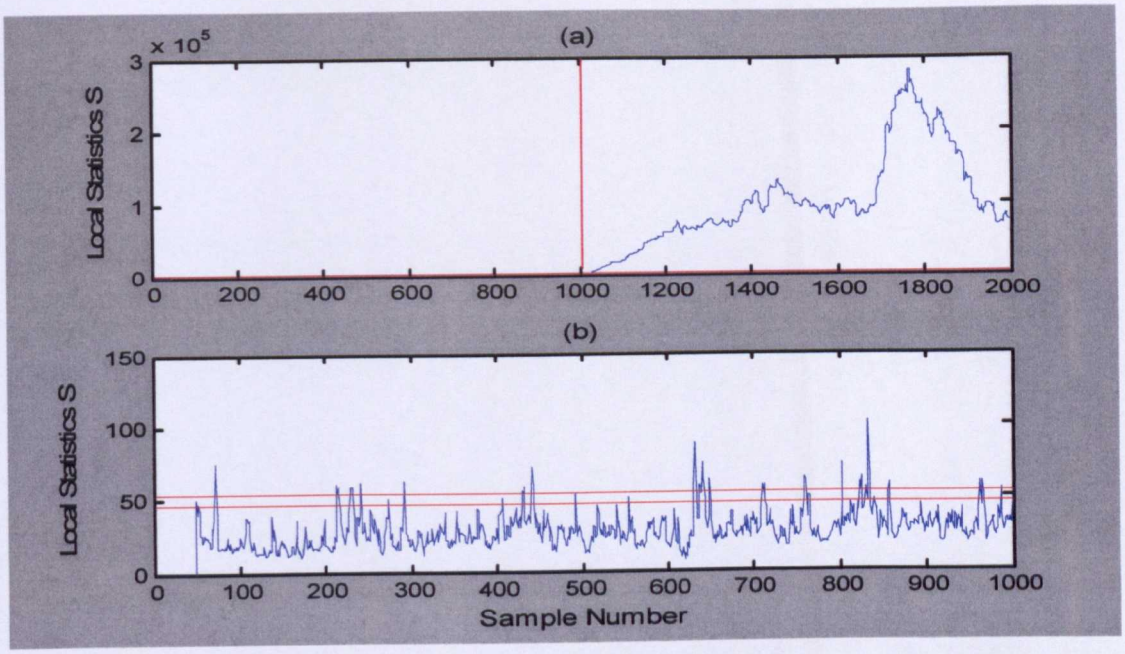


Figure 6.18: Plot of the local statistics versus sample number for the dynamic PCA based monitoring (a) the whole experimental data set when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 2)

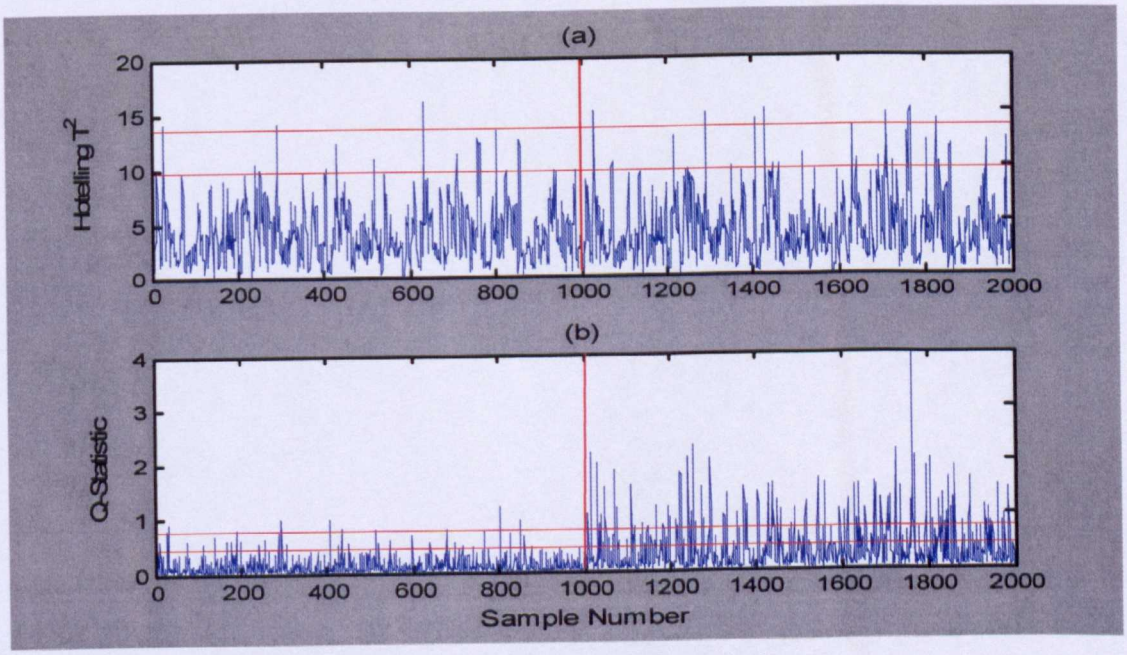


Figure 6.19: Plot of (a) Hotelling T^2 and (b) Q-statistic versus sample number for the whole experimental data set for the dynamic PCA based conventional monitoring when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (example 2)

The average reliability for the conventional monitoring scheme based on dynamic PCA model reported, calculated as per the procedure given by Kano et al., (2001) is given in Table 6.7

Table 6.7: Average reliability (%) of the dynamic PCA based conventional MSPC

Monitoring statistic	Case		
	1	2	3
Hotelling T^2	1.6	3.8	14.2
Q-statistic	13.6	39.3	65.5

Comparison of Tables 6.3 and 6.7 highlights the importance of selecting an appropriate model for monitoring. When a dynamic system is monitored using a static model, the performance of the (conventional) monitoring scheme is much poorer (Table 6.3) than when a dynamic model based scheme is applied (Table 6.7). For example, the average reliability for the large change situation for a static PCA model is 9.5% and the corresponding value for a dynamic PCA based monitoring scheme is 65.5%, approximately a 6 fold increase. Although the incorporation of the dynamics into the model has considerably increased the performance (especially for the large change situation) of the conventional monitoring scheme, its reliability is still poor for the small and medium change cases. The average reliability for the scheme proposed by Kano et al., (2001) is given in Table 6.8. The corresponding figures for the local approach based scheme are given in Table 6.9.

Table 6.8: Average reliability (%) for the dynamic PCA based scheme of Kano et al., (2001)

Monitoring statistic	Case		
	1	2	3
Proposed by Kano et.al (2001)	291.	81.6	95.4

Table 6.9: Average reliability (%) for the dynamic PCA based local monitoring scheme.

Monitoring statistic	Case		
	1	2	3
Based upon Local Approach	96.79	97.34	98.96

It can be observed that the local approach based scheme not only detects the large change case successfully (with an average delay of one sample) but is almost equally efficient in detecting the small and medium change cases.

6.7.3. Example 3: Fault Detection in Continuous Stirred Tank Reactor

The proposed statistic for PCA model change detection is now applied to detect a fault in a continuous stirred tank reactor (CSTR). A schematic diagram of the CSTR is shown in Figure 6.20 (Zhang, 1991). In the reactor an irreversible heterogeneous catalytic exothermic reaction $A \rightarrow B$ takes place. The objective of the process is to maintain the product concentration at a desired level by controlling the temperature of the reactor, the height in the reactor and the reactor mixing conditions. Temperature in the reactor is controlled by manipulating the flow rate of the feed cold water to the heat exchanger via a cascade control system. Manipulating the product flow rate controls level in the reactor. The mixing conditions are controlled by manipulating the recycle flow rate. A SIMULINK based simulator for this process was developed by Lane (2000).

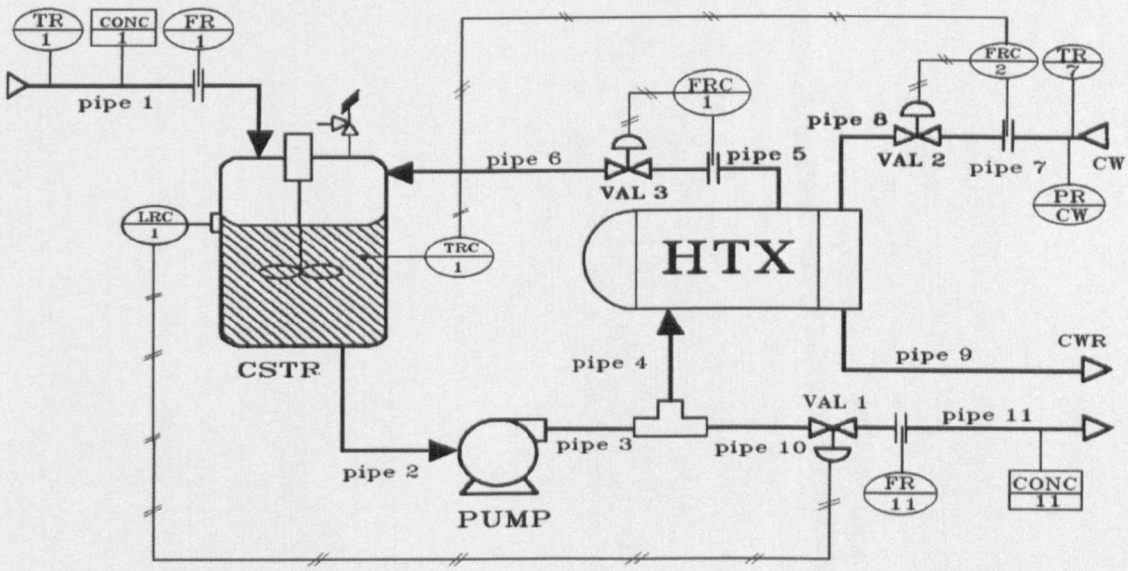


Figure 6.20: Continuous Stirred Tank Reactor Schematic

A nominal data set consisting of 12 process variables and one thousand samples was generated. The process variables measured were:

1. Feed flow rate
2. Temperature of feed
3. Concentration of reactant A in feed
4. Pressure of cooling water
5. Temperature of cooling water
6. Control signal to recycling flow valve
7. Height in reactor
8. Temperature in reactor
9. Recycle flow rate
10. Cooling water flow rate
11. Recycle Temperature
12. Product flow rate

After the data was auto-scaled, PCA was performed. The percentage contribution of each principal component towards the total variance of data is summarized in Table 6.10

Table 6.10: Variance contribution for PCA on CSTR data

Number of PC	Eigenvalue	% variance explained	Cumulative % variance explained
1	2.65	22.09	22.09
2	2.03	16.93	39.02
3	1.37	11.38	50.40
4	1.09	9.10	59.50
5	1.07	8.95	68.45
6	1.06	8.87	77.31
7	0.963	8.03	85.34
8	0.832	6.93	92.27
9	0.727	6.06	98.33
10	0.143	1.20	99.53
11	0.0564	0.47	100.0
12	0.0243	0.00	100.0

A PCA model was built using 8 principal components, as determined from cross validation. The fault studied was the fouling of the heat exchanger and was simulated by reducing the

heat transfer coefficient of the heat exchanger from its nominal value. Three experimental data sets, each comprising 1500 samples with the first one thousand samples corresponding to normal conditions and the remaining 500 samples corresponding to three different magnitudes of fouling (i) small (2%) (ii) medium (3%) and large (5%), were generated. The local approach based monitoring scheme with window parameters n_0 and n_1 equal to 300 and 50 respectively was applied to each of the experimental data sets. Figures 6.21, 6.23 and 6.25 show the plots of the local statistics for the cases of small, medium and large fouling respectively. The corresponding performance of the conventional monitoring scheme is shown in Figures 6.22, 6.24 and 6.26. It can be seen that while the local approach based scheme detects all these changes without any delay, the conventional approach is almost insensitive in the cases relating to small and medium changes but is able to detect the case of large fouling in the heat exchanger.

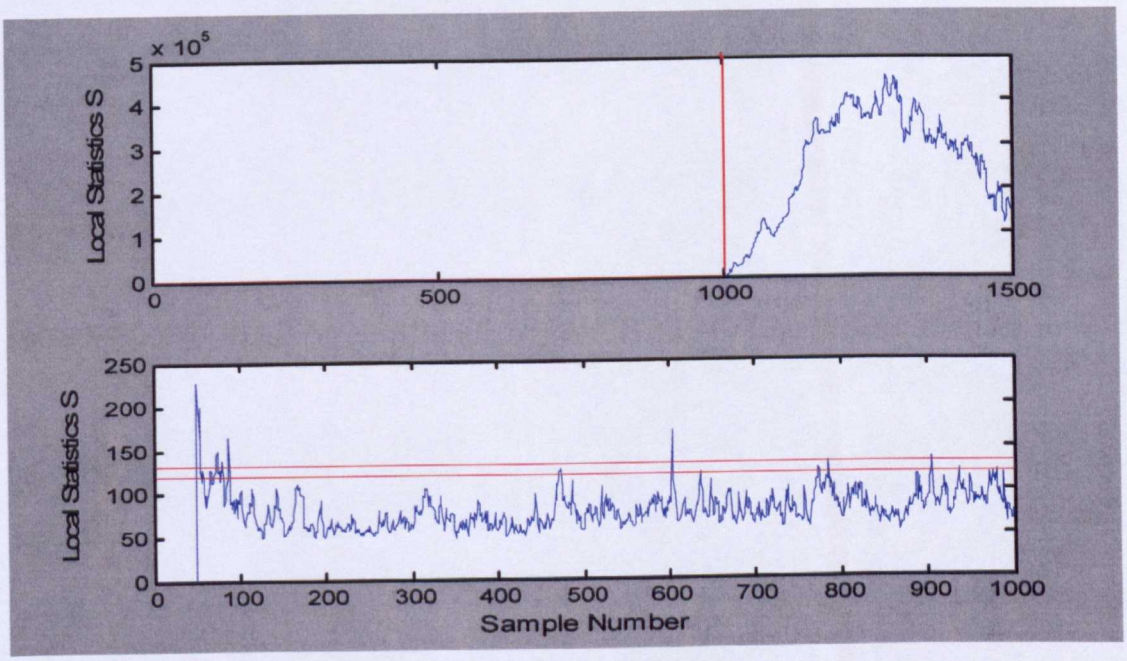


Figure 6.21: Plot of the local statistics versus sample number for (a) the whole experimental data set when is fouling is increased by 2% at sample number 1000(b) the normal operating condition component of the experimental data set (example 3).

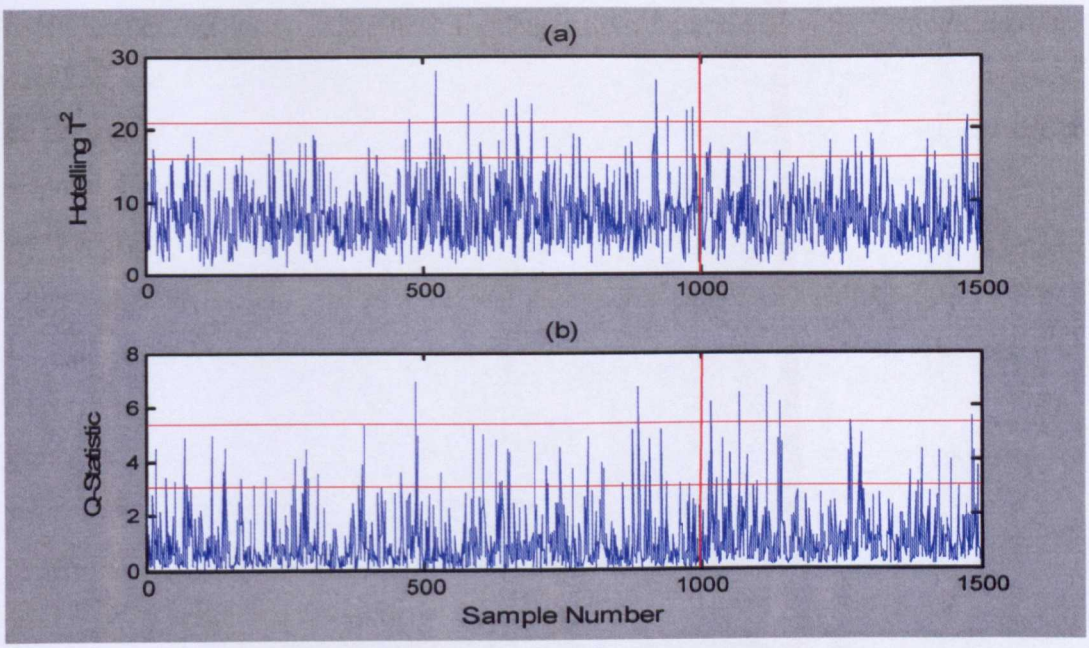


Figure 6.22: Plot of (a) Hotelling T^2 and (b) Q-statistic for the experimental data set when the fouling is increased by 2% (example 3)

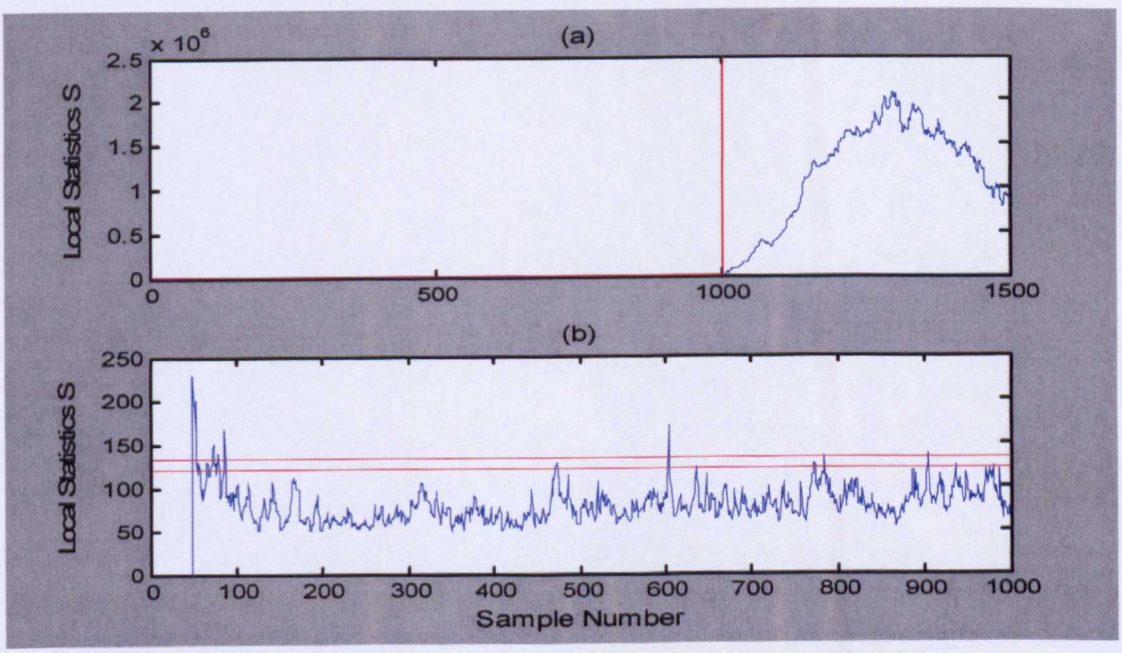


Figure 6.23: Plot of the local statistics versus sample number for (a) the whole experimental data set when is fouling is increased by 3% at sample number 1000 (b) the normal operating condition component of the experimental data set (example 3)

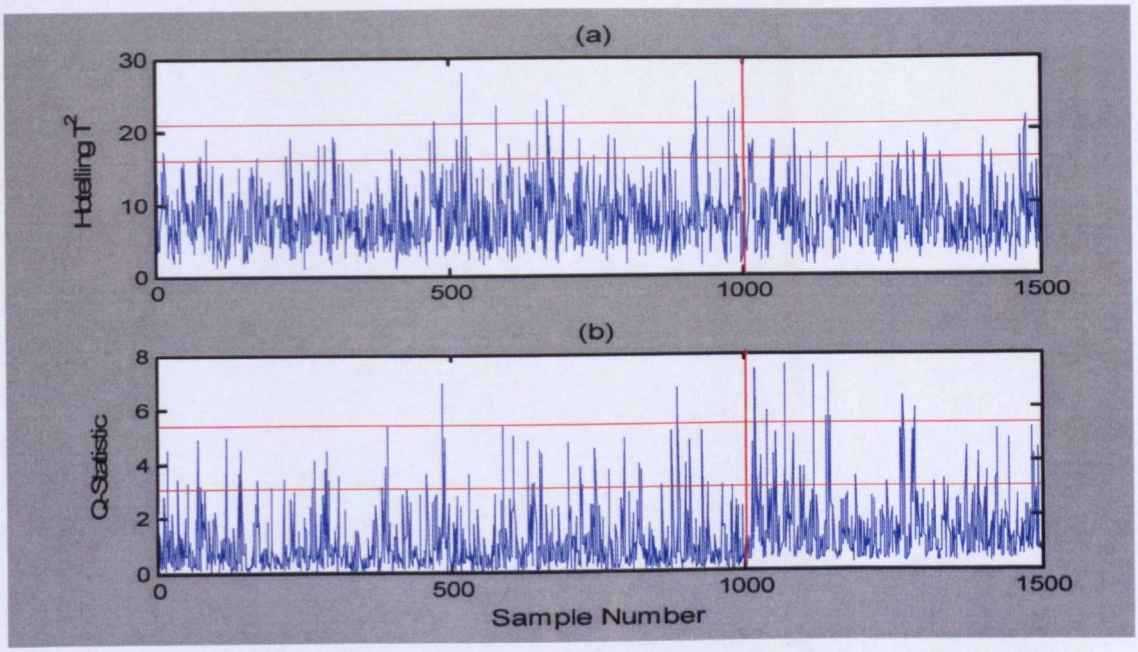


Figure 6.24: Plot of (a) Hotelling T^2 and (b) Q-statistic for the experimental data set when the fouling is increased by 3% (example 3)

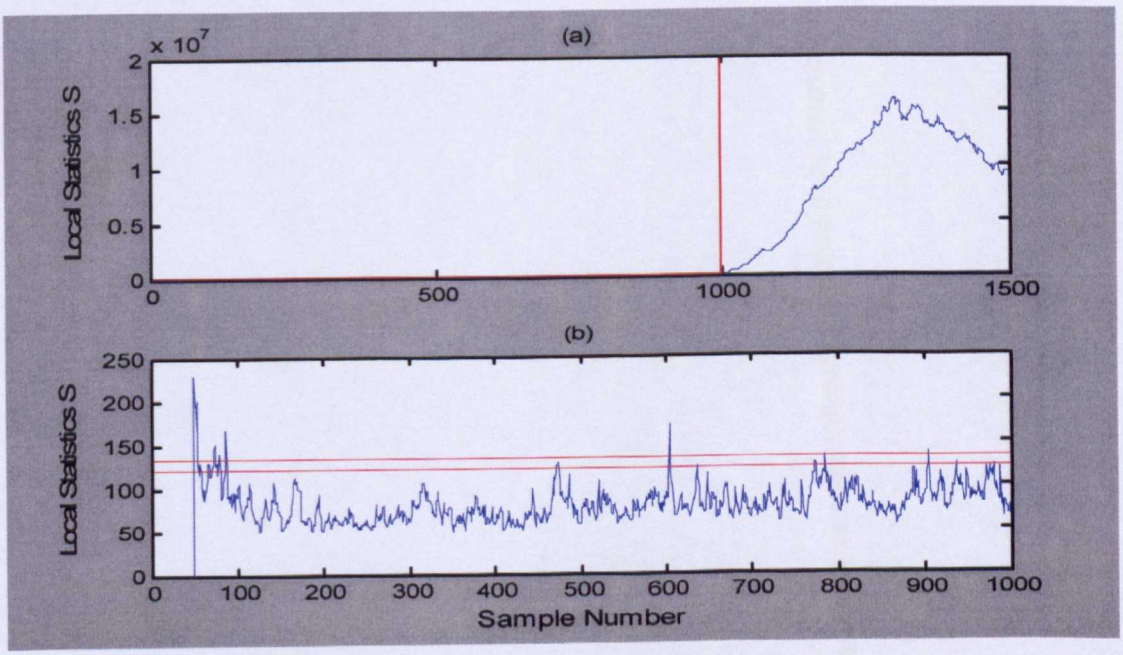


Figure 6.25: Plot of the local statistics versus sample number for (a) the whole experimental data set when is fouling is increased by 5% at sample number 1000 (b) the normal operating condition component of the experimental data set (example 3)

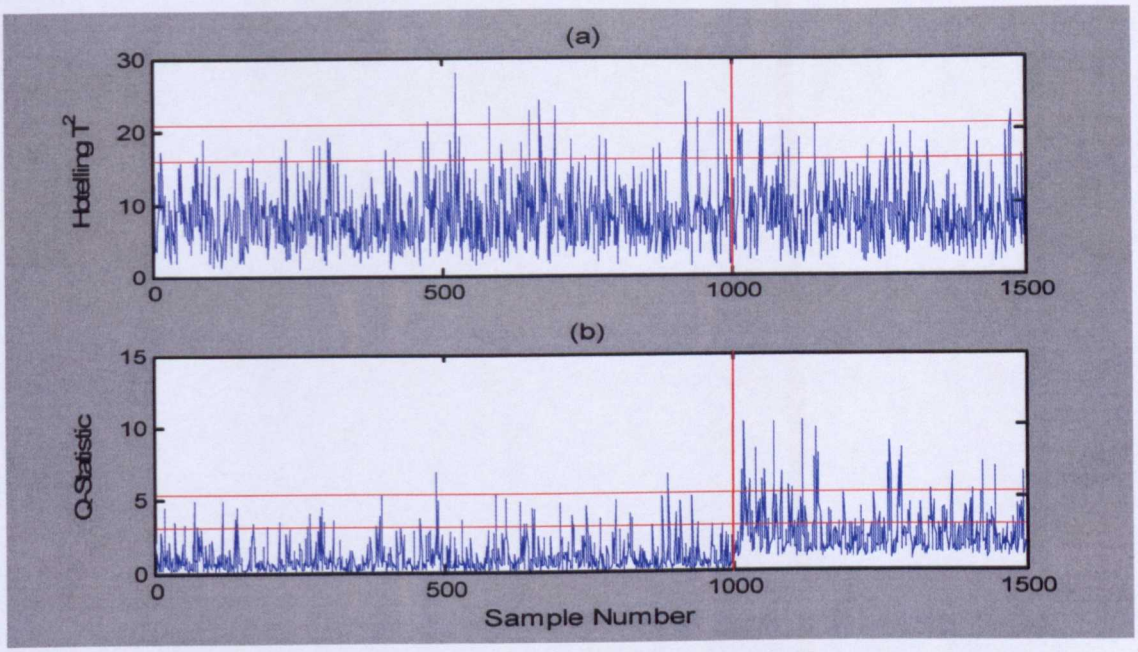


Figure 6.26: Plot of (a) Hotelling T^2 and (b) Q-statistic for the experimental data set when the fouling is increased by 5% (example 3)

6.8. Conclusions

In this chapter the abnormal changes that can occur in identical and independent multivariate Gaussian process variables have been divided into two categories namely (i) a change in the mean vector and (ii) a change in variance-covariance structure. It has been shown that although a conventional PCA based monitoring scheme, in particular the Q-statistic, can detect large changes in the variance-covariance structure, it is not sensitive to small changes. A new monitoring statistic based on the estimation function of PCA is derived. To derive the distribution function of the statistic, use is made of the local approach of hypothesis testing. The performance of this new statistic is tested and compared with a conventional monitoring scheme for detecting changes in two artificial data sets. It was found that the proposed scheme detects not only changes of large magnitudes, but is especially suitable for detecting small changes. The proposed scheme was also compared to the scheme recently proposed by Kano et al., (2001) on the basis of the performance index proposed by Kano et al., (2001) and is observed to outperform this scheme. The scheme was finally applied to detect fouling of heat exchanger in continuous stirred tank reactor.

CHAPTER 7

Recursive Partial Least Squares with Application to Process Monitoring

7.1. Introduction

In the last chapter, it was shown that a conventional PCA based monitoring scheme is particularly insensitive for the detection of small changes in the variance-covariance structure of the variables and a new monitoring scheme based on a PCA model identification procedure was derived. In this chapter, the focus is on the monitoring of cross-covariance (i.e. between the input and output variables) through a PLS based monitoring scheme. A recursive algorithm for identifying a PLS model is first developed and then use is made of this algorithm to derive a monitoring statistic.

7.2 Recursive Partial Least Squares

The most common method for identifying a PLS model is the batch method. It is a two step procedure (1) the collation of the data into matrices **X** and **Y**, and (2) the calculation of the eigenvalue-eigenvectors of suitable combinations of the matrices **X** and **Y** through the application of the NIPALS algorithm. This method has two limitations. First it can be shown that the computational complexity of this approach increases at least quadratically with the dimensionality of the data (Partridge and Calvo, 1998). This can make the method impractical when the data set is large. Secondly, if the data is nonstationary and the PLS model requires to be updated regularly, then the single PLS model with constant parameters, as identified by the batch method, is inefficient. To overcome these limitations, adaptive methods, also known as on-line or recursive methods, have been proposed. In contrast to the conventional batch method, adaptive methods do not require the prior storage of data and the PLS model is updated as and when a sample of the data becomes available.

In general, there are two methods for the computation of recursive subspace projection techniques for PCA and PLS. In the first class of algorithms, the covariance matrix (for PCA) and cross-covariance matrix (for PLS) is updated on-line by a rank one modification procedure and then eigenvalues-eigenvectors of combinations of the updated matrices are calculated. This method was proposed by Li et al., (2000) and Dayal and MacGregor (1997(b)) for recursive PCA and PLS respectively. In the second class of algorithms, a

recursive equation for updating the eigenvalues and eigenvectors is derived directly from the data. This approach for PCA has attracted a great deal of attention in the research community since it has the additional advantage that the algorithm can be implemented using a neural network architecture. The latter method is, therefore, sometimes referred to as neural PCA (Oja, 1982).

The objective of this section is to propose a recursive PLS algorithm, which belongs to the second class of approaches. Although the proposed algorithm can be used to update the parameters of the PLS model, the objective is to derive a statistic that can be used to detect changes in the cross-covariance structure.

7.2.1 Literature Review

The literature on neural PCA is extensive. Neural PCA methods are based on the biologically motivated unsupervised Hebbian learning rule, which was first proposed by Hebb in his seminal book 'The Organization of Behaviour' (1949). Hebb hypothesised that "*when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both of the cells such that A's efficiency, as one of the cells firing B, is increased.* " Putting it more simply, the rule states that when cells (neurons) A and B are simultaneously excited, the strength of connection between the two is increased. Oja (1982) first showed that the normalized version of the Hebbian rule when applied to a neural network consisting of a single linear neuron converges to the principal eigenvector of the covariance matrix. This work by Oja (1982) attracted a lot of attention from the neural network community and several researchers, Sanger (1989), Oja (1989), Foldiak (1989), Kung and Diamantaras (1994), extended Oja's methodology to extract multiple components of PCA using a neural network consisting of multiple linear neurons. However, one limitation was that the speed of convergence was quite slow. To increase the speed, modifications to the Hebbian learning rule were proposed by Partridge and Calvo (1998) and Bannour and Sadjadi (1995). Diamantaras (1994) extended the Hebbian learning rule to extract principal components where two sets of variables $\mathbf{x}$ and $\mathbf{y}$ were available. This work was further generalized by Feng et al., (1998) to extract the singular components of any general matrix. A comprehensive review of the neural network implementation of PCA can be found in Diamantaras and Kung (1996).

The chemometrics research community have also attempted to develop an adaptive version of Partial Least Squares, but these mainly belong to the first class of algorithms. The

recursive algorithm for PLS was first proposed by Helland et al., (1991). In this algorithm, the old data is captured by the PLS model loadings (**P** and **Q**) and the new data is augmented with these loadings matrice. The model is then updated by performing the NIPALS algorithm on these augmented matrices. Qin (1993, 1998) modified this algorithm to identify a dynamic process model. Wold (1994) proposed exponentially weighted algorithms for both PCA and PLS. His approach is based on performing NIPALS algorithm on the augmented data set (**X** and **Y**) every time a new data sample becomes available. Dayal and MacGregor (1997(a)) proposed an improved version of this algorithm, in which the covariance and cross-covariance matrices (and not the data matrices) of **x** and **y** are updated using an exponential forgetting factor. The kernel algorithm (Dayal and MacGregor, 1997(b)) is then used to calculate the parameters of the new model. Lane et al., (2003) and Wang et al., (2003) more recently used recursive PCA and PLS respectively to monitor time varying chemical processes.

The recursive algorithm for PLS is now derived in the next sub-section. First the recursive equations for the extraction of the first latent variable are derived (section 7.2.2) and then the algorithm is extended to extract $A\ (\geq 1)$ latent variables (section 7.2.3).

7.2.2 Extraction of First Latent Variable

Let $\mathbf{x}(n) \in R^K$ and $\mathbf{y}(n) \in R^M$ be a sample of process and response variables respectively at time point n. As mentioned in Chapter 2, PLS seeks to find two vectors $\mathbf{w}_1 \in R^K$ and $\mathbf{v}_1 \in R^M$ such that

$$t_1 = \mathbf{w}_1^T \mathbf{x} \qquad (7.1)$$

and

$$\mathbf{u}_1 = \mathbf{v}_1^T \mathbf{y} \qquad (7.2)$$

have maximum covariance with the constraint $\|\mathbf{w}_1\| = 1$ and $\|\mathbf{v}_1\| = 1$. The covariance J between t_1 and u_1 is given by:

$$J(\mathbf{w}_1, \mathbf{v}_1) = E\{t_1 u_1\} \tag{7.3}$$

where $E\{.\}$ denotes the statistical expectation operator and it is assumed that t_1 and u_1 have zero mean. The problem, therefore, can be stated as:

$$\max_{\mathbf{w}_1, \mathbf{v}_1} (J) = \max_{\mathbf{w}_1, \mathbf{v}_1} E\{\mathbf{w}_1^T \mathbf{x} \mathbf{y}^T \mathbf{v}_1\} \quad \text{subject to } \|\mathbf{w}_1\| = 1 \ \|\mathbf{v}_1\| = 1 \tag{7.4}$$

Differentiating the objective function with respect to the weight vector $\mathbf{w}_1$ gives:

$$\frac{\partial}{\partial \mathbf{w}_1}(J(\mathbf{w}_1, \mathbf{v}_1)) = E\{\mathbf{x}\,\mathbf{y}^T \mathbf{v}_1\} = E\{\mathbf{x}\, u_1\} \tag{7.5}$$

Likewise differentiating the objective function with respect to the weight vector $\mathbf{v}_1$ gives:

$$\frac{\partial}{\partial \mathbf{v}_1}(J(\mathbf{w}_1, \mathbf{v}_1)) = E\{\mathbf{y}\,\mathbf{x}^T \mathbf{w}_1\} = E\{\mathbf{y}\, t_1\} \tag{7.6}$$

The gradient ascent rules for updating the weights $\mathbf{w}_1$ and $\mathbf{v}_1$ are:

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \eta \frac{\partial}{\partial \mathbf{w}_1}(J) = \mathbf{w}_1(n) + \eta E\{\mathbf{x}(n)\, u_1(n)\}$$
$$\mathbf{v}_1(n+1) = \mathbf{v}_1(n) + \eta \frac{\partial}{\partial \mathbf{v}_1}(J) = \mathbf{v}_1(n) + \eta E\{\mathbf{y}(n)\, t_1(n)\} \tag{7.7}$$

where η is the learning rate. To implement the recursive equations in (7.7), the statistical expectation requires to be estimated. Taking the instantaneous values $\mathbf{x}(n)u_1(n)$ and $\mathbf{y}(n)t_1(n)$ as the estimates of $E\{\mathbf{x}(n)u_1(n)\}$ and $E\{\mathbf{y}(n)t_1(n)\}$ respectively, as in the Least Mean Square (LMS) algorithm (Haykin, 1995), the recursive equations in (7.7) reduce to:

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \eta \mathbf{x}(n)\, u_1(n)$$
$$\mathbf{v}_1(n+1) = \mathbf{v}_1(n) + \eta\, \mathbf{y}(n)\, t_1(n) \tag{7.8}$$

It should be noted that the above equations for updating the weight vectors do not take into consideration the unit norm constraints on the weight vectors $\mathbf{w}_1$ and $\mathbf{v}_1$. Two approaches can be adopted to take into account these constraints. The first is to normalize the weight vectors to unit norm after each updating of the weight vectors. This gives the following updating equations:

$$\begin{aligned}
\tilde{\mathbf{w}}_1(n+1) &= \mathbf{w}_1(n) + \eta \mathbf{x}(n)\, u_1(n) \\
\mathbf{w}_1(n+1) &= \frac{\tilde{\mathbf{w}}_1(n+1)}{\|\tilde{\mathbf{w}}_1(n+1)\|} \\
\tilde{\mathbf{v}}_1(n+1) &= \mathbf{v}_1(n) + \eta \mathbf{y}(n)\, t_1(n) \\
\mathbf{v}_1(n+1) &= \frac{\tilde{\mathbf{v}}_1(n+1)}{\|\tilde{\mathbf{v}}_1(n+1)\|}
\end{aligned} \tag{7.9}$$

An alternative approach is to use the first order technique adopted in (Oja, 1982). Taylor series expansions of $\frac{1}{\|\tilde{\mathbf{w}}_1(n+1)\|}$ and $\frac{1}{\|\tilde{\mathbf{v}}_1(n+1)\|}$ are calculated and the second and higher order terms in the learning rate η are neglected. For example, the first order Taylor series expansion of $\frac{1}{\|\tilde{\mathbf{w}}_1(n+1)\|}$ is given by:

$$\frac{1}{\|\tilde{\mathbf{w}}_1(n+1)\|} = 1 - \eta_1\, t_1(n)\, u_1(n) \tag{7.10}$$

Therefore, the updating equation for weight vector $\mathbf{w}_1$ becomes:

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \eta_1 (\mathbf{x}(n) - \mathbf{w}_1(n)\, t_1(n)) u_1(n) \tag{7.11}$$

This can also be written as:

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \eta \mathbf{x}'(n)\, u_1(n) \tag{7.12}$$

where

$$\mathbf{x}'(n) = \mathbf{x}(n) - \mathbf{w}_1(n)t_1(n) \tag{7.13}$$

Similarly the updating equation for the weight vector $\mathbf{v}_1$ is given by:

$$\mathbf{v}_1(n+1) = \mathbf{v}_1(n) + \eta(\mathbf{y}(n) - \mathbf{v}_1(n)u_1(n))\,t_1(n) \tag{7.14}$$

If b_1 is the inner regression coefficient between u_1 and t_1, that is:

$$u_1 = b_1 t_1 + e_1 \tag{7.15}$$

then it can be computed recursively using the LMS rule (Haykin, 1995):

$$b_1(n+1) = b_1(n) + \eta e_1(n)t_1(n) = b_1(n) + \eta(u_1(n) - \hat{u}_1(n))t_1(n) \tag{7.16}$$

where

$$\hat{u}_1(n) = b_1(n)t_1(n) \tag{7.17}$$

7.2.3 Extraction of More Than One Latent Variable

The second and higher order latent variables of PLS in the NIPALS algorithm are found by deflating matrices **X** and **Y**. Since the recursive algorithm deals with vectors instead of matrices, the next step is to deflate the vectors. The deflated vector $\mathbf{x}_2(n)$ for computing the second latent variable can be obtained by re-writing equation (2.59) in terms of a single observation of input variables:

$$\mathbf{x}_2(n) = \mathbf{x}(n) - t_1(n)\mathbf{p}_1 \tag{7.18}$$

where $\mathbf{p}_1$ is the loading vector. Equation (7.18) shows that to compute the deflated vector $\mathbf{x}_2(n)$ the loading vector $\mathbf{p}_1$ needs to be calculated recursively. From equations (2.55) and (2.56), the loading vector $\mathbf{p}_1$ is identified as the parameter vector for predicting the input variables $\mathbf{x}$ from the latent variable t_1 :

$\mathbf{x} = t_1\mathbf{p}_1 + \mathbf{e}_1$	(7.19)

and is determined such that the norm of the prediction error $\mathbf{e}_1$ is a minimum. Using the LMS rule to determine the loading vector $\mathbf{p}_1$ recursively:

$\mathbf{p}_1(n+1) = \mathbf{p}_1(n) + \eta\mathbf{e}_1(n)\, t_1(n)$	(7.20)

where

$\mathbf{e}_1(n) = \mathbf{x}(n) - t_1(n)\mathbf{p}_1(n)$	(7.21)

The deflated vector $\mathbf{y}_2$ (from equation 2.59) is given by:

$\mathbf{y}_2(n) = \mathbf{y}(n) - \hat{u}_1(n)\, \mathbf{v}_1$	(7.22)

Once the deflated vectors $\mathbf{x}_2$ and $\mathbf{y}_2$ are available, the second latent variables t_2 and u_2 can be calculated by determining the weight vectors $\mathbf{w}_2$ and $\mathbf{v}_2$. The updating equations for these vectors can be obtained from equations (7.12) and (7.14) by replacing $\mathbf{x}$ and $\mathbf{y}$ with $\mathbf{x}_2$ and $\mathbf{y}_2$ respectively. Rewriting the updating equations in terms of $\mathbf{x}$ and $\mathbf{y}$ requires making use of the fact that it is not necessary to deflate both $\mathbf{x}$ and $\mathbf{y}$. It was proven in Höskuldsson (1988) that only $\mathbf{x}$ needs to be deflated, this was later extended by Dayal et al., (1997(a)) by who proved that either $\mathbf{x}$ or $\mathbf{y}$ can be deflated. Letting:

$t_2(n) = \mathbf{w}_2{}^{T}(n)\mathbf{x}(n)$ $u_2(n) = \mathbf{v}_2{}^{T}(n)\mathbf{y}(n)$	(7.23)

and

$t_2'(n) = \mathbf{w}_2{}^{T}(n)\mathbf{x}_2(n)$ $u_2'(n) = \mathbf{v}_2{}^{T}(n)\mathbf{y}_2(n)$	(7.24)

The relationship between t_2, u_2 and t'_2, u'_2 can be derived by substituting equations (7.18) and (7.22) into equation (7.24):

$$t'_2(n) = \mathbf{w}_2{}^T(n)(\mathbf{x}(n) - t_1(n)\mathbf{p}_1(n))$$
$$u'_2(n) = \mathbf{v}_2{}^T(n)(\mathbf{y}(n) - \hat{u}_1(n)\mathbf{v}_1(n)) \quad (7.25)$$

These equations can be further simplified to give:

$$t'_2(n) = t_2(n) - t_1(n)\, d_{12}(n)$$
$$u'_2(n) = u_2(n) - \hat{u}_1(n)\, r_{12}(n) \quad (7.26)$$

where

$$d_{12}(n) = \mathbf{w}_2{}^T(n)\, \mathbf{p}_1(n)$$
$$r_{12}(n) = \mathbf{v}_2{}^T(n)\, \mathbf{v}_1(n) \quad (7.27)$$

Now the updating equation for $\mathbf{w}_2$, assuming that $\mathbf{x}$ is not deflated but $\mathbf{y}$ is deflated, in accordance with equation (7.12) can be written as:

$$\mathbf{w}_2(n+1) = \mathbf{w}_2(n) + \eta(\mathbf{x}(n) - \mathbf{w}_2(n)\, t_2(n))\, u'_2(n)) \quad (7.28)$$

Similarly the recursive equation for $\mathbf{v}_2$, assuming $\mathbf{y}$ is not deflated but that $\mathbf{x}$ is deflated, can be written as:

$$\mathbf{v}_2(n+1) = \mathbf{v}_2(n) + \eta(\mathbf{y}(n) - \mathbf{v}_2(n)\, u_2(n))\, t'_2(n) \quad (7.29)$$

The recursive equations for computing d_{12} and r_{12} can also be derived. Multiplying both sides of equation (7.20) by $\mathbf{w}_2^T$ and incorporating equation (7.21) gives:

$$d_{12}(n+1) = d_{12}(n) + \eta(t_2(n) - t_1(n) d_{12}(n))\, t_1(n) \quad (7.30)$$

Similarly, r_{12} can be computed by multiplying equation (7.29) on both sides by $\mathbf{v}_1^T$ and using equations (7.26), (7.23) and (7.24), thus:

$$r_{12}(n+1) = r_{12}(n) + \eta(u_1(n) - r_{12}(n)u_2(n))t_2'(n) \quad (7.31)$$

The inner regression coefficient, b_2 for the second set of latent variables can be computed as:

$$b_2(n+1) = b_2(n) + \eta(u_2' - b_2(n)t_2'(n))t_2'(n) \quad (7.32)$$

The above scheme can be extended to extract, in general, the A^{th} latent variable as follows:

$$\mathbf{w}_A(n+1) = \mathbf{w}_A(n) + \eta(\mathbf{x}(n) - \mathbf{w}_A(n)\, t_A(n))\, u_A'(n)) \quad (7.33)$$

$$\mathbf{v}_A(n+1) = \mathbf{v}_A(n) + \eta(\mathbf{y}(n) - \mathbf{v}_A(n)\, u_A(n))\, t_A'(n) \quad (7.34)$$

where

$$t_A = \mathbf{w}_A{}^T\mathbf{x}$$
$$u_A = \mathbf{v}_A{}^T\mathbf{y} \quad (7.35)$$

$$t_A' = t_A - \sum_{i=1}^{A-1} d_{iA}t_i'$$
$$u_A' = u_A - \sum_{i=1}^{A-1} r_{iA}\hat{u}_i' \quad (7.36)$$

Also

$$d_{iA}(n+1) = d_{iA}(n) + \eta_3\, t_A'(n)\, t_i'(n) \qquad \text{for } i < A \quad (7.37)$$

$$r_{iA}(n+1) = r_{iA}(n) + \eta_2(u_A(n) - r_{iA}(n)u_A(n))t_A'(n) \quad \text{for } i < A \quad (7.38)$$

7.3 Summary of the Algorithm

A summary of the algorithm for computing A latent variables is given below:

Step1: Initialize weight vectors $\mathbf{w}_i, \mathbf{v}_i$, inner regression coeffiecient b_i for $i = 1,2...A$ and d_{ij}, r_{ij} for $i = 1,2...A$; $j = 1,2..i-1$ to random values.

Step 2: Compute at time point n

for $i = 1,2,...A$

$$t_i(n) = \mathbf{x}(n)^T \mathbf{w}_i(n)$$

$$u_i(n) = \mathbf{y}(n)^T \mathbf{v}_i(n)$$

if $i = 1$

$$t_i'(n) = t_i(n)$$

$$u_i'(n) = u_i(n)$$

$$\hat{u}_i'(n) = b_i(n) t_i'(n)$$

else

$$t_i'(n) = t_i(n) - \sum_{j=1}^{i-1} d_{ji}(n) t_j'(n)$$

$$\hat{u}_i'(n) = b_i(n) t_i'(n)$$

$$u_i'(n) = u_i(n) - \sum_{j=1}^{i-1} r_{ji}(n) \hat{u}_j'(n)$$

Step 3: Update the parameters

for $i = 1,2,...A$

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(\mathbf{x}(n) - \mathbf{w}_i(n)\, t_i(n))\, u_i'(n)$$

$$\mathbf{v}_i(n+1) = \mathbf{v}_i(n) + \eta(\mathbf{y}(n) - \mathbf{v}_i(n)\, u_i(n))\, t_i'(n)$$

$$b_i(n+1) = b_i(n) + \eta(u_i' - b_i(n) t_i'(n)) t_i'(n)$$

$$d_{ji}(n+1) = d_{ji}(n) + \eta t_i'(n)\, t_j'(n) \qquad \text{for each } j < i-1$$

$$r_{ji}(n+1) = r_{ji}(n) + \eta(u_i(n) - r_{ji}(n)u_j(n))t_i'(n) \quad \text{for each } j < i-1$$

Step 4: Repeat steps 2 and 3 until convergence

7.4. Simulation Study

To test the above method, an artificial data set was generated. The vector **x** consists of 5 variables generated as follows: x_1, x_2, x_4 and x_5 are distributed normally with zero mean and a variance of unity and $x_3 = x_1 + x_2$. Measurement noise, which is Gaussian with zero mean and a variance of 0.1 is added to each of the input variables. The output vector **y** consists of 4 variables with the component variables generated as: $y_1 = 2\ x_1$, $y_2 = x_1 + x_2 + x_3$, $y_3 = 4\ x_4$ and $y_4 = x_2 + x_3 + x_4 + x_5$. Gaussian measurement noise with a mean of zero and variance of 0.1 was added to these output variables.

A data set consisting of 200 samples was generated. After the data was auto-scaled, the recursive algorithm was applied to extract 3 latent variables, that is $A = 3$. The learning rates in all the recursive equation was set equal to a fixed value of 0.01. The choice of the learning rate was determined as a compromise between the speed of convergence and instability (oscillations around the minima). A High learning rate leads to fast convergence but may not converge to the minima (solution). On the other hand a small value for the learning rate makes the algorithm converge more slowly. This is illustrated in Appendix 1 for two value of learning rates, 0.001 and 0.04. The convergence of the first three solutions for the weight vectors **w** and **v** are shown in Figures 7.1 and 7.2 respectively. Figure 7.3 shows the convergence of the inner regression coefficients. It can be seen from the figures that for the example considered, approximately 5 iterations are required for the first solution to converge. The successive solutions, however, require fewer number of iterations because they are computed in parallel with the first one.

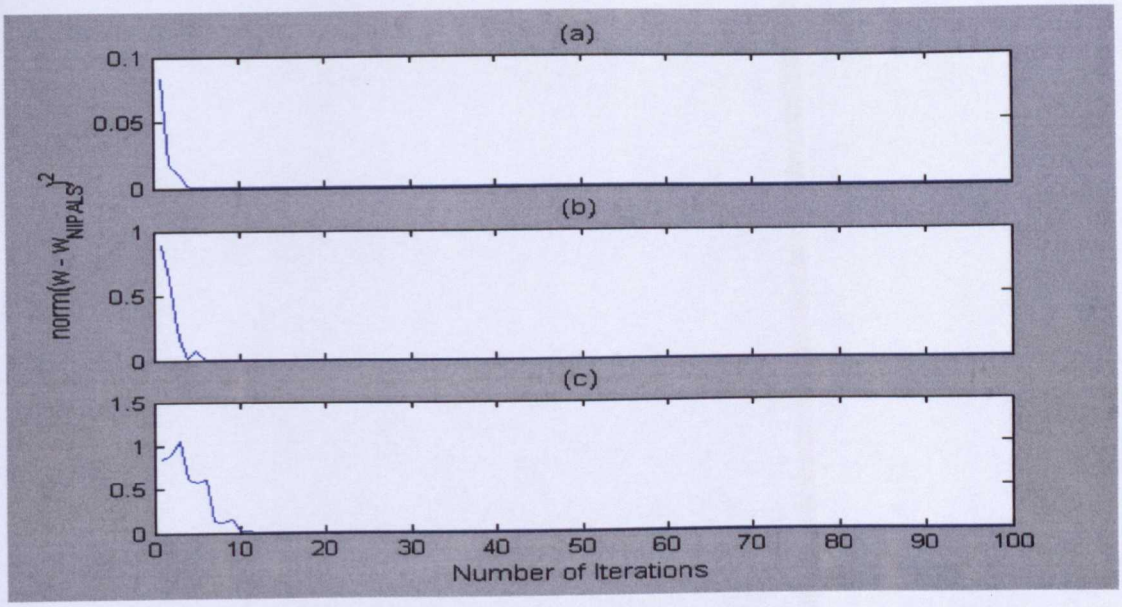


Figure 7.1: Plot of estimation error $\|\mathbf{w}-\mathbf{w}_{NIPALS}\|^2$, where $\mathbf{w}_{NIPALS}$ is the PLS solution from the NIPALS algorithm versus number of iterations for the first three solutions of $\mathbf{w}$ (a) $\mathbf{w}_1$ (b) $\mathbf{w}_2$ (c) $\mathbf{w}_3$

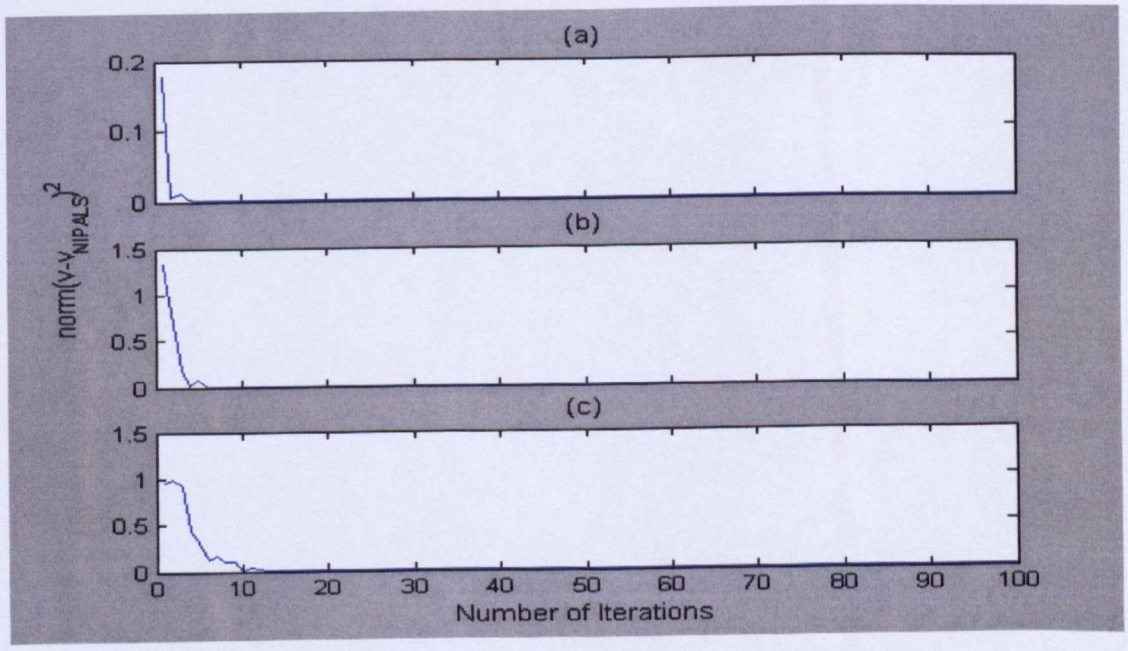


Figure 7.2: Plot of estimation error $\|\mathbf{v}-\mathbf{v}_{NIPALS}\|^2$, where $\mathbf{v}_{NIPALS}$ is the PLS solution from the NIPALS algorithm, against number of iterations for the first three solutions of $\mathbf{v}$ (a) $\mathbf{v}_1$ (b) $\mathbf{v}_2$ (c) $\mathbf{v}_3$

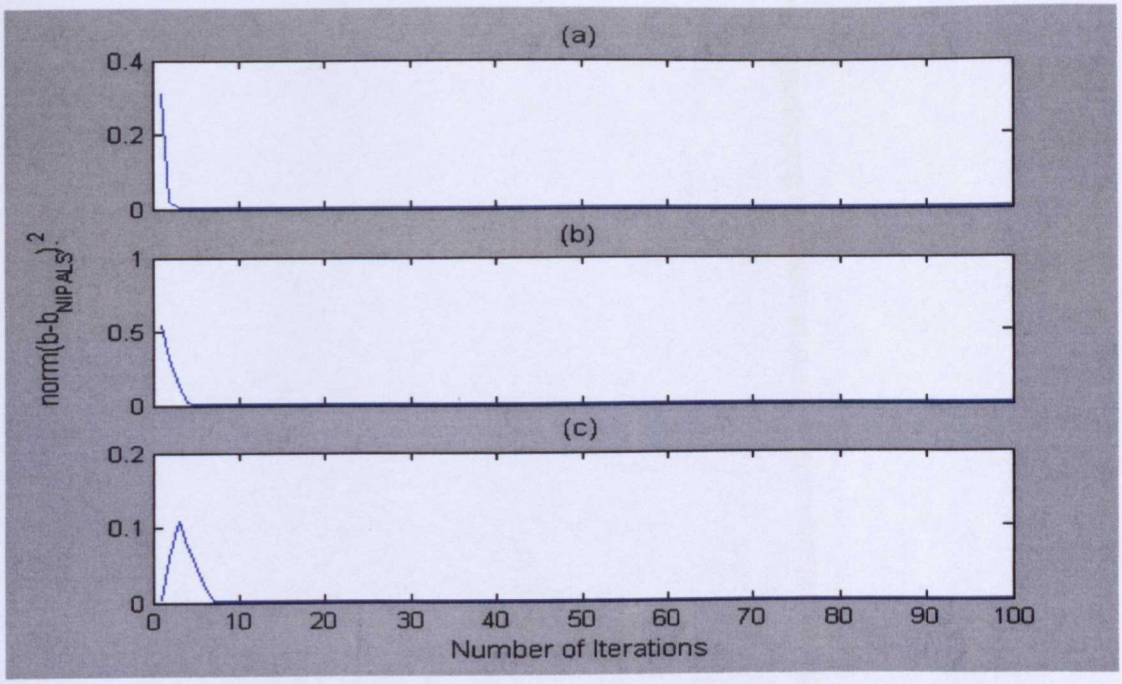


Figure 7.3: Plot of estimation error $\|b - b_{NIPALS}\|^2$, where b_{NIPALS} is the PLS inner regression coefficient from the NIPALS algorithm, versus number of iterations for the first three inner regression coefficients(a) b_1 (b) b_2 (c) b_3

7.5 Application to Process Performance Monitoring

In general, a recursive algorithm for estimating a parameter vector $\boldsymbol{\theta}$ can be written as:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \eta \mathbf{k}_1\left(\boldsymbol{\theta}(n), \mathbf{x}(n)\right) \tag{7.39}$$

where $\mathbf{k}_1$ (multiplied by the learning rate η) represents the change in the parameter vector $\boldsymbol{\theta}$ at the current time. For the LMS algorithm, $\mathbf{k}_1$ is given by the gradient of the instantaneous estimate of the objective function:

$$\mathbf{k}_1 = \frac{\partial}{\partial \boldsymbol{\theta}}\left(J_n\left(\boldsymbol{\theta}, \mathbf{x}(n)\right)\right)\Big|_{\boldsymbol{\theta}(n)} \tag{7.40}$$

where J_n is an instantaneous estimate of the objective function J, the optimization of which determines the parameter vector $\boldsymbol{\theta}$. For recursive PLS, for example, $\mathbf{k}_1$ for estimating the first weight vector $\mathbf{w}_1$ can be determined from equation (7.11):

$$\mathbf{k}_1 = (\mathbf{x}(n) - \mathbf{w}_1(n)\, t_1(n))u_1(n) \tag{7.41}$$

Taking the statistical expectation on both sides of equation (7.40) gives:

$$E\{\mathbf{k}_1\} = E\left\{\frac{\partial}{\partial \boldsymbol{\theta}}\left(J_n(\boldsymbol{\theta}, \mathbf{x}(n))\Big|_{\boldsymbol{\theta}(n)}\right)\right\} = \frac{\partial}{\partial \boldsymbol{\theta}} E\left\{\left(J_n(\boldsymbol{\theta}, \mathbf{x}(n))\Big|_{\boldsymbol{\theta}(n)}\right)\right\} \tag{7.42}$$

Now assuming that the nominal model parameter of the system is known and is equal to $\boldsymbol{\theta}_0$, if the measurements $\mathbf{x}(n)$ from the system correspond to the nominal model parameter $\boldsymbol{\theta}_0$, then the right hand side of equation (7.42) evaluated at $\boldsymbol{\theta}_0$ must be equal to zero:

$$E\{\mathbf{k}_1\}\Big|_{\boldsymbol{\theta}_0} = \mathbf{0} \tag{7.43}$$

This is because, the nominal model parameter $\boldsymbol{\theta}_0$ corresponds to the optimization of the objective function $E\{J_n\}$, and therefore its gradient at $\boldsymbol{\theta}_0$ is equal to zero. When the system parameters do not correspond to the nominal parameter $\boldsymbol{\theta}_0$, the statistical expectation of $\mathbf{k}_1$ will be non-zero. The change detection in the parameters of the system is, therefore, equivalent to detecting a change in the mean of $\mathbf{k}_1$.

The weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$ ($\text{for } i = 1,2 \ldots A$) in a PLS model depend on the cross-covariance (between the input and output variables) structure of the process variables. Specifically, $\mathbf{w}_i$ and $\mathbf{v}_i$ are obtained by singular value decomposition of the cross-covariance matrix. A change in the cross-covariance, therefore, can be detected by detecting a change in the weight vectors $\mathbf{w}_i$ and $\mathbf{v}_i$. It can, however, be proven (Chapter, 3, remark 1) that vector $\mathbf{v}_i$ is related to $\mathbf{w}_i$ and therefore it is sufficient to detect a change in $\mathbf{w}_i$ in order to detect a change in the cross-covariance structure.

As in equation (7.40), corresponding to each $\mathbf{w}_i$, a statistic $\mathbf{k}_i$ can be derived from the recursive equation for $\mathbf{w}_i$, such that the mean of $\mathbf{k}_i$ is zero under normal conditions but becomes non-zero when the cross-covariance structure changes from normal conditions. The expression for $\mathbf{k}_i$ from equation (7.33) is:

$$\mathbf{k}_i = (\mathbf{x}(n) - \mathbf{w}_i(n)\, t_i(n))\, u_i'(n) \tag{7.44}$$

If all the weight vectors are arranged in one column, then the vector $\boldsymbol{\theta}_0$ of the parameters corresponding to the normal operating conditions of a PLS model is given as:

$$\boldsymbol{\theta}_0 = [\mathbf{w}_1\, \mathbf{w}_2 \ldots \mathbf{w}_A]^T \tag{7.45}$$

The corresponding augmented vector $\mathbf{k}$ of the statistic is given by:

$$\mathbf{k} = [\mathbf{k}_1\ \mathbf{k}_2 \ldots \mathbf{k}_A]^T \tag{7.46}$$

Since each component of the vector $\mathbf{k}$ has zero mean when the corresponding loading vector corresponds to normal operating conditions, it therefore follows that the mean of the augmented vector $\mathbf{k}$ is zero when the PLS model parameter vector $\boldsymbol{\theta}_0$ corresponds to normal operating conditions. When any or all of the weight vectors change, the mean of vector $\mathbf{k}$ deviates from zero. The vector $\mathbf{k}$, therefore is a primary residual (section 6.5.1). After the primary residual is determined, the local approach of hypothesis testing (described in Chapter, 6) can be used to design an algorithm to detect a change in its mean value.

It should be noted from equation (7.44) that calculation of $\mathbf{k}_i$ requires u_i', which in turns requires the measurement of the quality variables $\mathbf{y}$ on-line. In some processes, the quality variables are not available as frequently as the process variables. To determine the monitoring statistic in such a situation, u_i' can be replaced by its predicted value $\hat{u}_i'$. The justification for this is that under normal conditions, the covariance between the t-and u-scores is the same as the covariance between t-and predicted u-scores, that is:

$$E\{t_i' u_i'\} = E\{t_i' (\hat{u}_i' + e)\} = E\{t_i' \hat{u}_i'\} \tag{7.47}$$

The expression for $\mathbf{k}_i$ can, therefore, be written as:

$$\mathbf{k}_i = (\mathbf{x}(n) - \mathbf{w}_i(n)t_i(n))\hat{u}_i'(n) \quad (7.48)$$

7.5.1 Summary of the Change Detection Algorithm

Given: Matrices **X** and **Y** corresponding to the normal operating condition of a process

Mean centre and scale each variable to unit variance.

Step 1: Build a PLS model using A latent variables

Step 2: Compute the primary residuals for each principal component at each sample time:

$$\mathbf{k}_i(n) = (\mathbf{x}(n) - \mathbf{w}_i t_i(n))\hat{u}_i'(n)$$

Step 3: Determine the augmented vector $\mathbf{k}(n) = [\mathbf{k}_1(n)\ \mathbf{k}_2(n) \ldots \mathbf{k}_A(n)]^T$

Step 4: Remove the bias, i.e., mean centre $\mathbf{k}(n)$

Step 5: Select the window size parameter n_0 (typical value is in the range 300-500) and n_1 (30-50).

Step 6: Compute improved residuals at each sample time:

$$\mathbf{r}_n = \frac{1}{n_0 + 1} \sum_{i=n-n_0}^{n} \mathbf{k}(i)$$

Step 7: Calculate the covariance matrix $\boldsymbol{\Sigma}_\mathbf{r}$ of the improved residuals

Step 8: Compute the local statistics at each sample time:

$S_n = \mathbf{r}_n^T \, \boldsymbol{\Sigma}_r^{-1} \, \mathbf{r}_n$	

Step 9: Determine the confidence limit (95 %, 99%) t_0

Step 10: If there are a large number of false alarms, change the window parameter n_0 and repeat steps 6-9

Step 11: Finally apply the algorithm to new (experimental) data set by scaling it with the same values that were used in the scaling of the nominal data set.

7.5.2 Simulation Studies

The algorithm described above is first applied to detect a change in the parameter of an artificial system and is then applied to detect a fault in a continuous stirred tank reactor.

7.5.2.1 Example 1: Detection of a change in the parameters in an artificial system

In this example the artificial system described in section 6.7.2 is considered. A normal data set comprising 2000 samples of two input variables **x** and two output variables **y** is generated and stored in matrices **X** and **Y**. The matrix **X** is augmented with one lagged value for each of the input and output variable so that the size of the augmented matrix $\mathbf{X}_{aug}$ is 1999×6. The matrices $\mathbf{X}_{aug}$ and **Y** are auto-scaled and the NIPALS algorithm is applied to the data. The percentage contribution of different latent variables is summarised in Table 7.1. A PLS model using three latent variables is then built.

Three (experimental) data sets, each comprising 2000 samples, corresponding to the three changes (listed in Table 6.1) in the coefficient relating the second state variable, u_2, to the first input, x_1, are generated. The first one thousand sample of each data set correspond to normal operating conditions and the remaining one thousand correspond to a change in the parameter. The change detection algorithm with parameters n_0 and n_1 equal to 300 and 50 respectively was then applied.

Table 7.1: Percent variance captured by PLS model (example 1)

No. of LV	% variance explained (X)	Cumulative % variance explained (X)	% Variation explained (Y)	Cumulative % variance explained (Y)
1	37.14	37.14	51.48	51.48
2	36.86	74.01	34.02	85.50
3	13.20	87.21	7.48	92.98
4	11.47	98.68	3.54	96.52
5	1.29	99.97	2.16	98.68
6	0.03	100.00	0.04	98.72

Figures 7.4(a), 7.6(a) and 7.8(a) show plots of the local statistic for the experimental data set corresponding to small, medium and large changes respectively. The lower panel in each of these figures correspond to the normal operating conditions of the experimental data set. The performance of conventional PLS based monitoring scheme, which is based on three statistics namely the Q-statistic in the input space, the Q-statistic in the output space and Hotelling T^2, is shown in Figures 7.5, 7.7 and 7.9. It is seen from these figures that the proposed monitoring scheme successfully detects all the changes. The delays in detecting small, medium and large change in the system parameters for the proposed algorithm are 23, 17 and 5 samples respectively. The conventional monitoring scheme, in comparison, is insensitive to small and medium changes but the Q-statistic in the output space does show an upward shift for the large change after the occurrence of the change. This shift, however, is not sufficient to give a clear signal of the change.

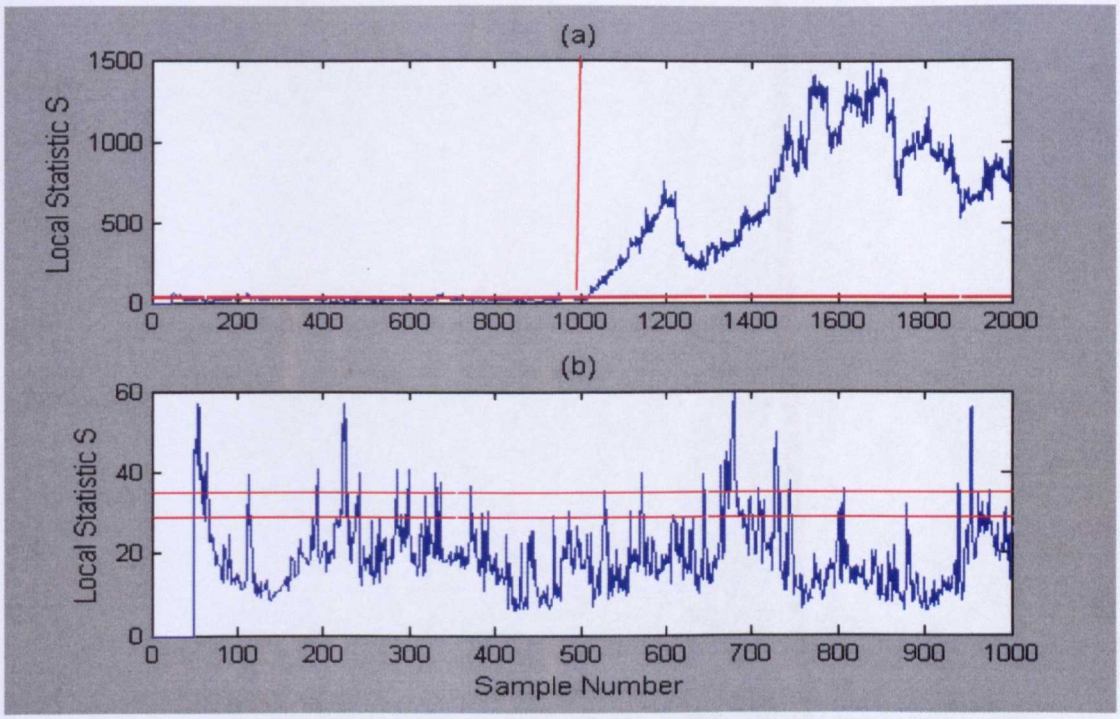


Figure 7.4: Plot of the local statistics versus sample number for (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (b) the normal operating condition component of the experimental data (example 1)

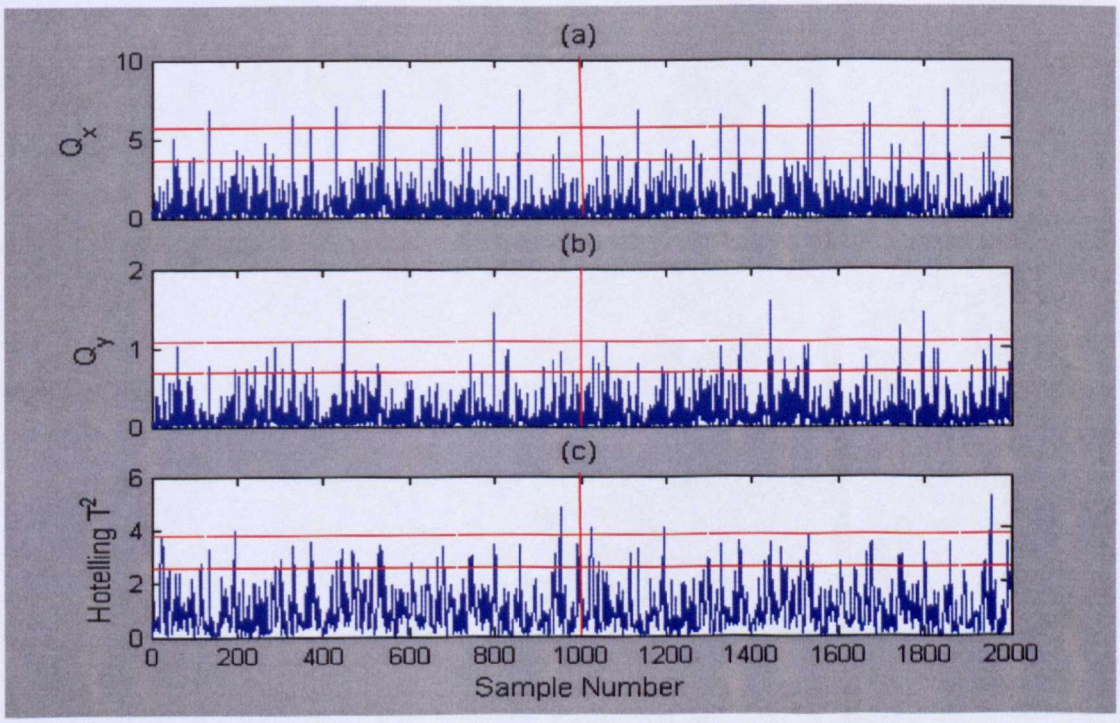


Figure 7.5: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space versus sample number for the whole experimental data set when the system parameter is changed from 3.0 to 2.5 at sample number 1000 (example 1)

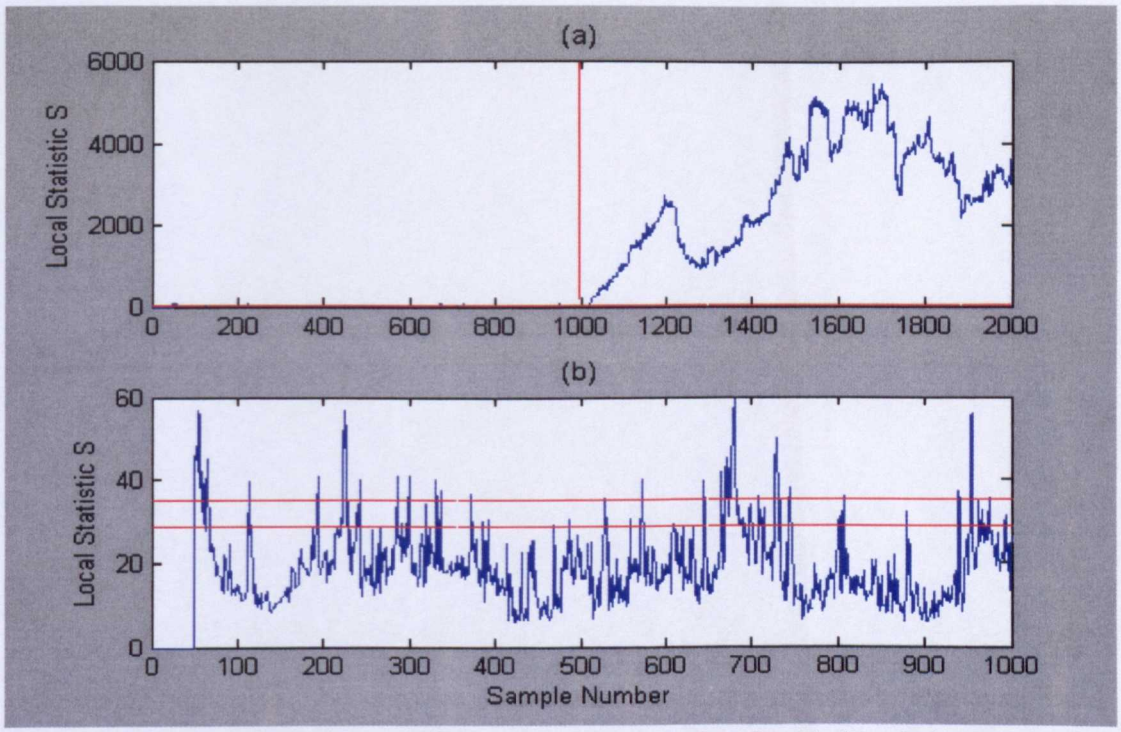


Figure 7.6: Plot of the local statistics versus sample number for (a) the whole experimental data set when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 1)

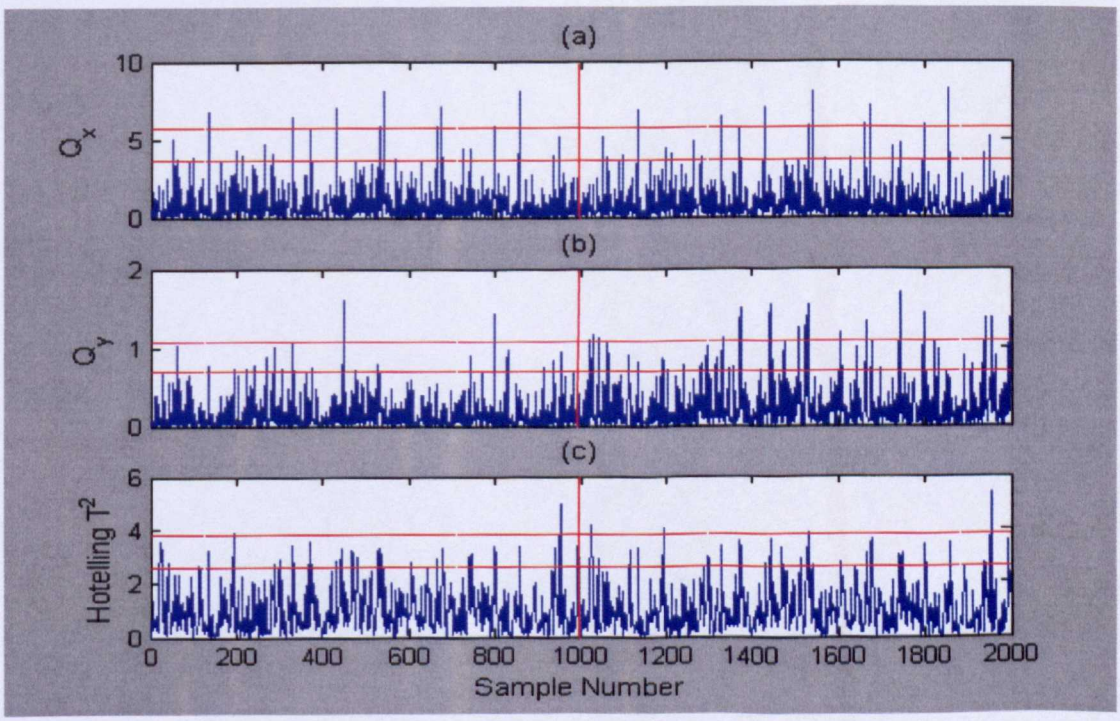


Figure 7.7: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space versus sample number for the whole experimental data set when the system parameter is changed from 3.0 to 2.0 at sample number 1000 (example 1)

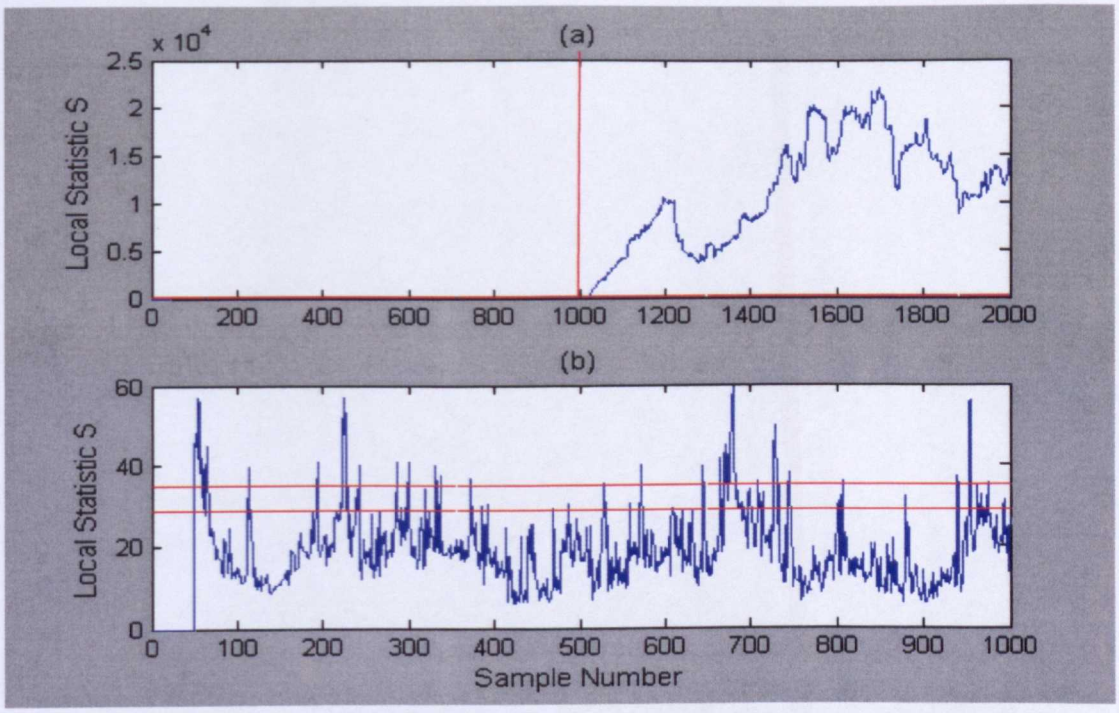


Figure 7.8: Plot of the local statistics versus sample number for (a) the whole experimental data set when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (b) the normal operating condition component of the experimental data (example 1)

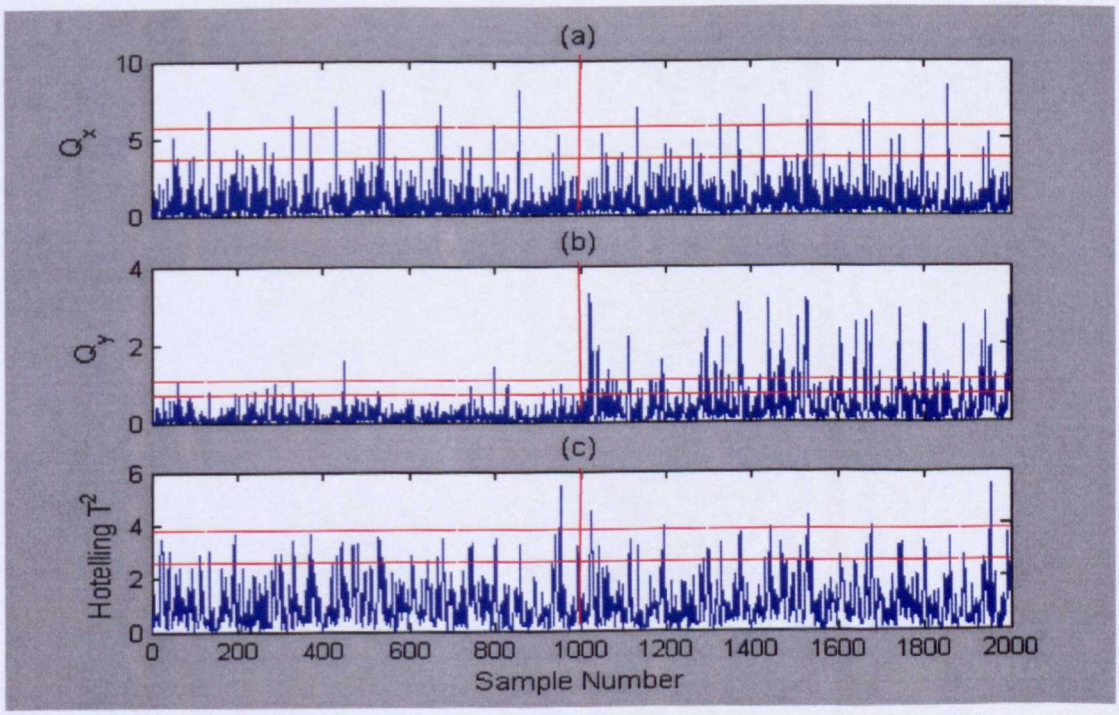


Figure 7.9: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space versus sample number for the whole experimental data set when the system parameter is changed from 3.0 to 1.0 at sample number 1000 (example 1)

The reasons for better performance of the proposed algorithm and the poor sensitivity of conventional monitoring scheme can be summarized as follows:

(1) Any change in the parameters of the system is reflected as a change in the mean value of the statistic, **k**, which is of the same dimension as that of the parameter vector of the model of the system. For example, in the PLS model, the parameter vector (equation 7.45) is of dimension $(6 \times 3 = 18)$, which is same as that of the statistic **k** (equation 7.46). The dimension of the residual vector, on which the Q-statistics are calculated, is 6 (for the residual in the input space) and 2 (for the residual in the output space). The residual vectors, therefore, may not be able to capture the full information about the change.

(2) The proposed algorithm is based on the local approach of hypothesis testing which, as explained in chapter 6, is especially suitable for detecting small changes in the parameters

(3) The change in the mean value of statistic **k** is detected by an algorithm which is "nearly optimal" in the sense that it minimizes the delay for a given false alarm rate.

7.5.2.2 Example 2: Fault detection in a continuous stirred tank reactor

The proposed scheme is finally applied to detect a fault in a continuous stirred tank reactor described previously in Chapter 6 (section 6.7.3). Of the 12 measured variables, three variables namely temperature in the reactor, height in the reactor and product flow rate are taken as the output variables and the remaining 9 are taken as input variables. A normal data set consisting of 2000 samples is generated from the SIMULINK based simulator of the CSTR system. Partial least squares was performed after the data was auto-scaled. The percentage variance captured by the different latent variables is shown in Table 7.2. A PLS model using 6 latent variables was built as determined by cross-validation.

Three experimental data sets, each comprising 2000 samples with the first one thousand samples corresponding to normal conditions and the remaining 1000 samples corresponding to three different variants of fouling (i) small (2%) (ii) medium (3%) and large (5%), were generated. The performance of the proposed algorithm for change detection is shown in Figures 7.10, 7.12 and 7.13. The corresponding performance of conventional PLS based monitoring scheme is shown in Figures 7.11, 7.13 and 7.15. These results once again show that while the proposed algorithm detects all these changes without any delay, the

conventional monitoring scheme only detects the situation where the level of fouling is large while remaining insensitive to small and medium levels of faults. The reasons for the better performance of the proposed scheme are the same as given in example 1.

Table 7.2: Percent variance captured by PLS model (example 2)

No. of LV	% variance explained (X)	Cumulative % variation explained (X)	% variance explained (Y)	Cumulative % variance explained (Y)
1	14.86	14.86	36.91	36.91
2	17.71	32.57	9.46	46.37
3	9.77	42.34	4.63	51.00
4	12.82	55.16	3.67	54.67
5	3.80	58.96	21.00	75.67
6	6.59	65.55	9.70	85.37
7	10.88	76.43	0.23	84.60
8	11.57	88.00	0.07	85.67
9	12.00	100.00	0.00	85.67

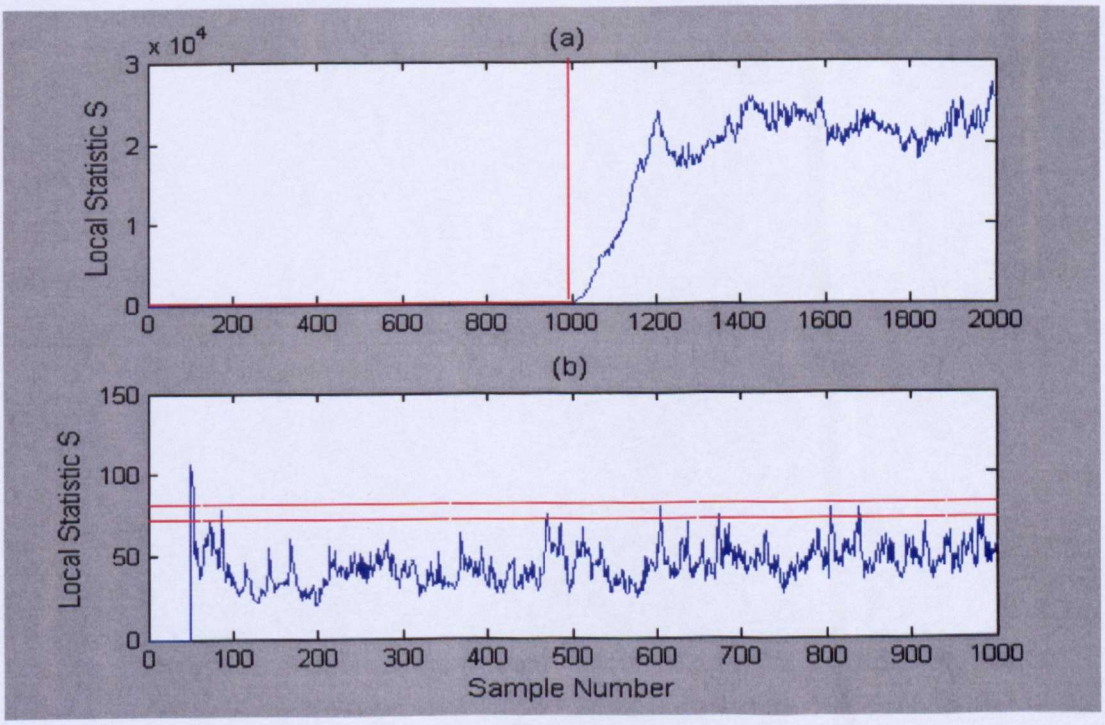


Figure 7.10: Plot of the local statistics versus sample number for (a) the whole experimental data set when fouling is increased by 2% at sample number 1000 (b) the normal operating condition component of the experimental data set (example 2)

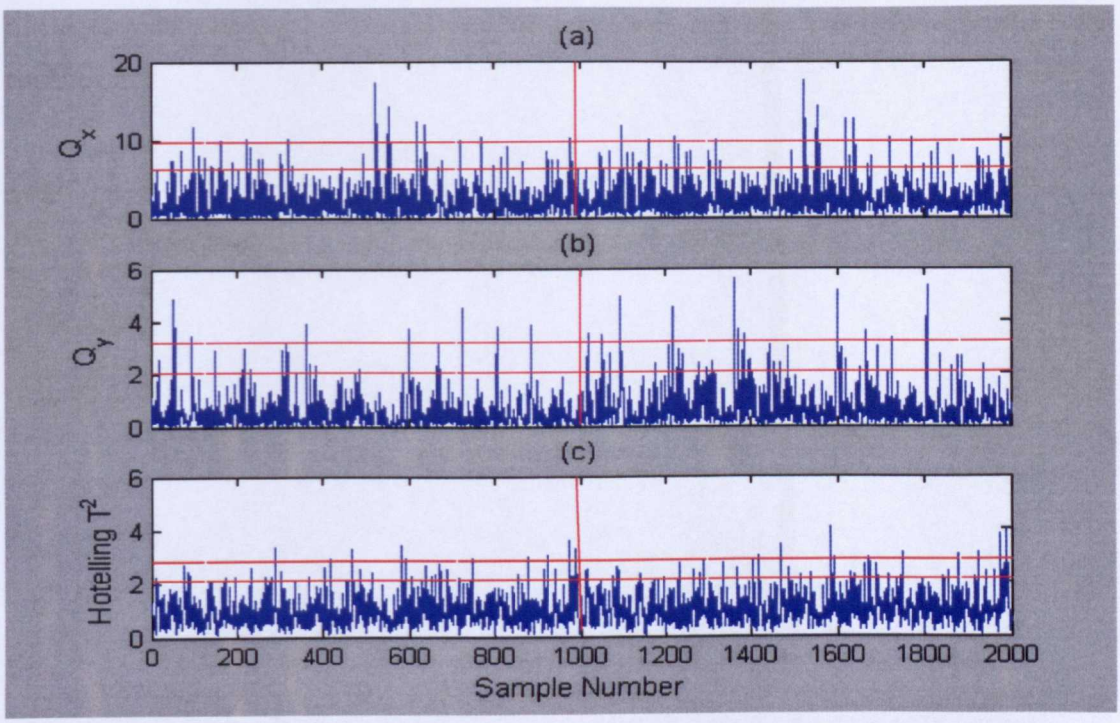


Figure 7.11: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space for the experimental data set when fouling is increased by 2% (example 2)

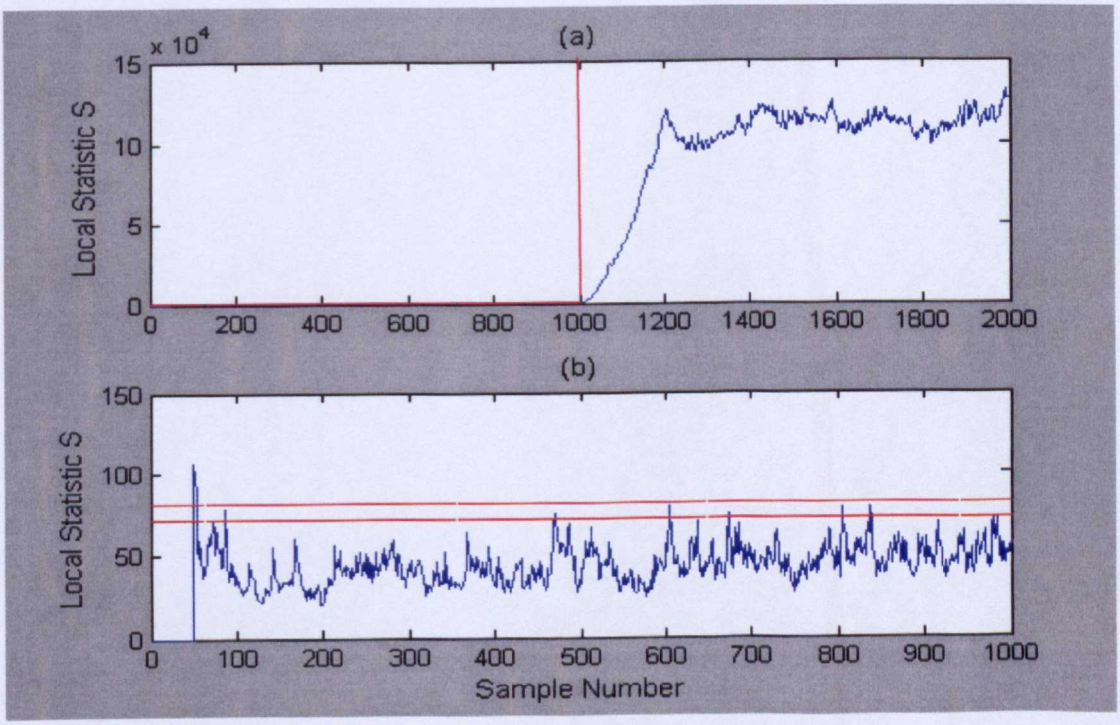


Figure 7.12: Plot of the local statistics versus sample number for (a) the whole experimental data set when fouling is increased by 3% at sample number 1000 (b) the normal operating condition component of the experimental data set (example 2)

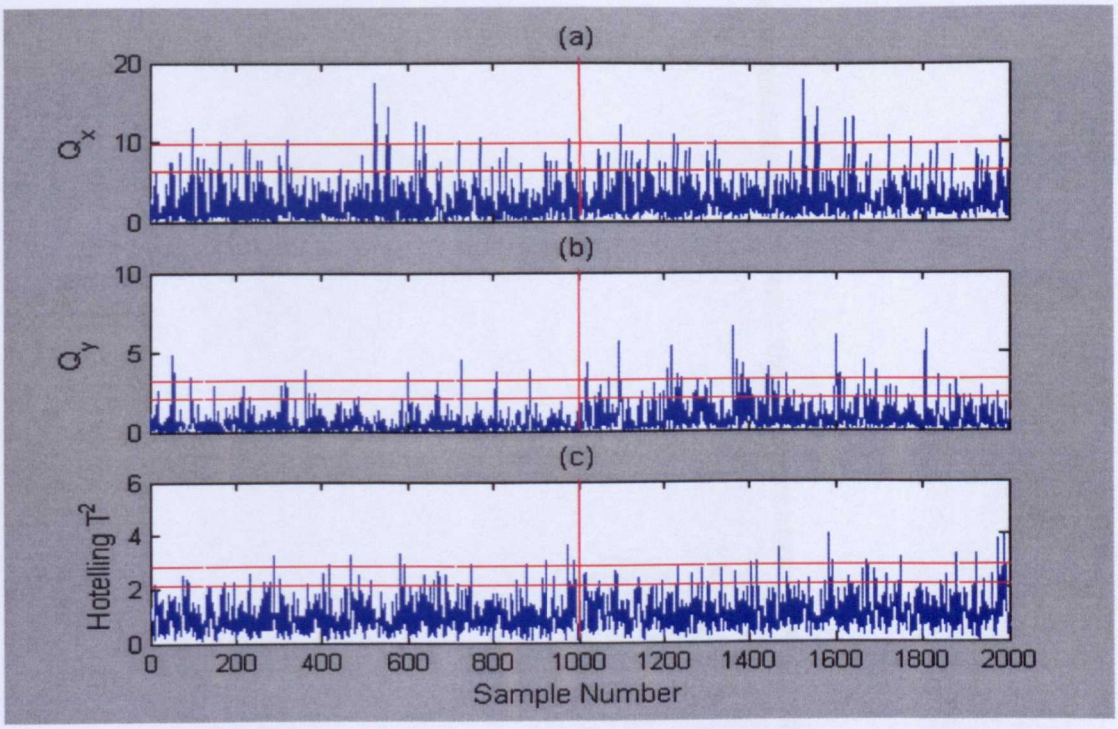


Figure 7.13: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space for the experimental data set when fouling is increased by 3% (example 2)

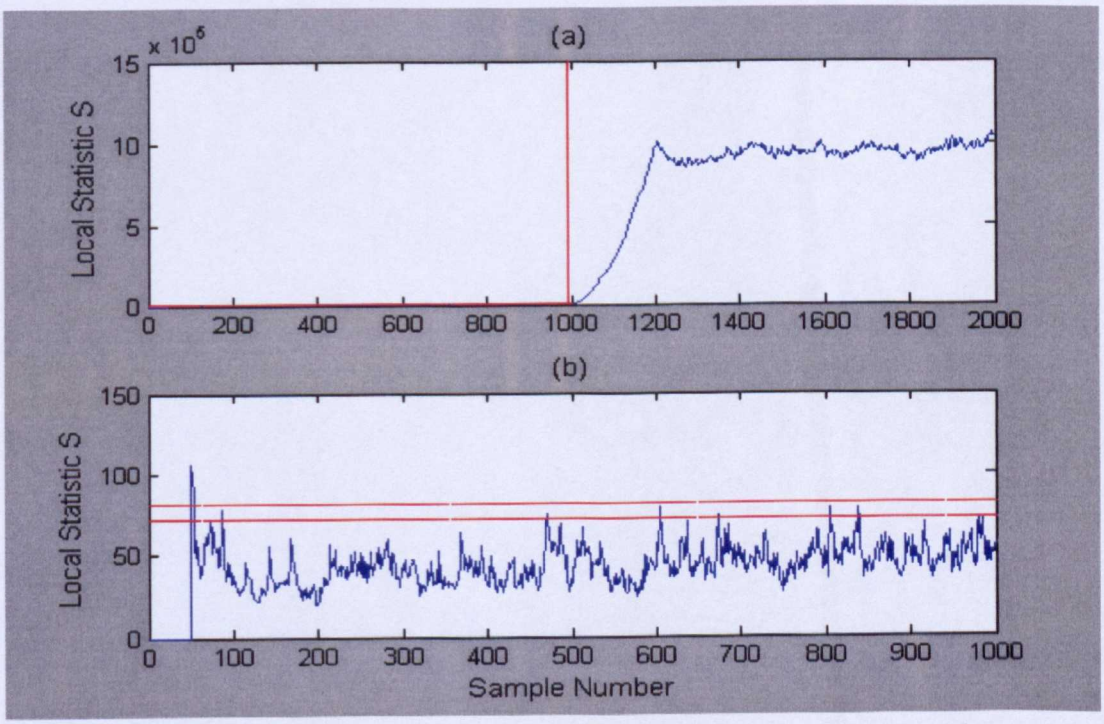


Figure 7.14: Plot of the local statistics versus sample number for (a) the whole experimental data set when fouling is increased by 5% at sample number 1000 (b) the normal operating condition component of the experimental data set (example 2)

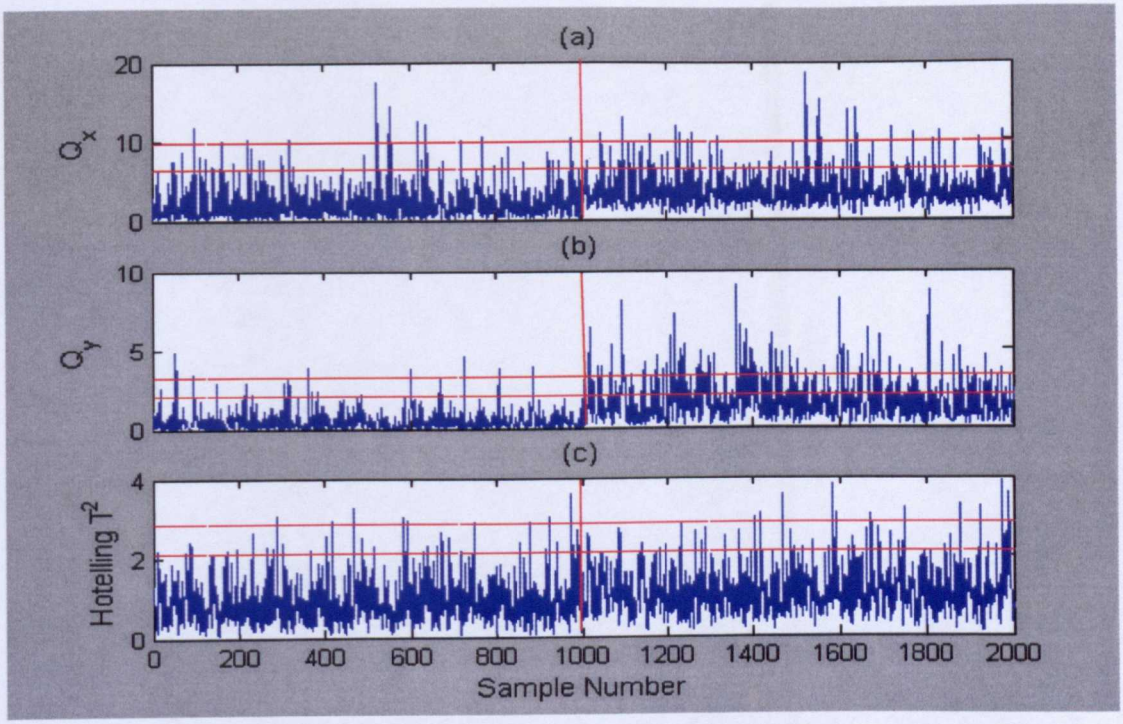


Figure 7.15: Plot of (a) Hotelling T^2 and (b) Q-statistic in the output space (c) Q-statistic in the input space for the experimental data set when fouling is increased by 5% (example 2)

7.6 Conclusions

In this chapter, a recursive algorithm, which computes parameters for all the latent variables in parallel, is proposed. The algorithm is shown to converge to the parameters computed by the NIPALS algorithm. A statistic was then derived from this recursive algorithm to monitor a change in the cross-covariance (between the output and the input variables) structure of the measured variables. The monitoring statistic is especially suitable for detecting small changes in the covariance structure that cannot be detected by a conventional PLS monitoring scheme. The proposed monitoring scheme was first tested to detect a change in the parameter of an artificial system before using it to detect different levels of fouling in a continuous stirred tank reactor.

CHAPTER 8

Conclusions and Recommendations

8.1 Introduction

This thesis contributes to the two disciplines of modelling and monitoring of multivariate signals. Specifically, in the first part of the thesis, issues relating to the extension of the partial least squares algorithm to more complex situations where the data exhibits non-linear and dynamic behaviour were investigated. The second part was concerned with the detection of abnormal changes in the variance-covariance structure of the data in PCA and PLS based monitoring schemes. The contribution and main results of the thesis are summarized in section 8.2. Recommendations for the future work are given in section 8.3

8.2 Main Contributions and Results

In most applications of PLS, the objective is to predict the response variables as accurately as possible. An alternative application of PLS is that of parameter estimation where the objective is to estimate the parameters from the data in such a way that they are 'close' to the 'true' parameters. It is known that PLS gives biased estimates of the parameters when the number of latent variables retained in the model is less than the number of input variables. However, it is shown that when a subspace of dimension A ($< K$, number of input variables) is correlated with the output variable and a PLS1 model is built using A latent variables then PLS1 gives unbiased estimates of the parameters. Furthermore, the variance of the PLS1 estimates can be less than the variance of the estimates using ordinary least squares.

Several non-linear extensions of PLS have been proposed in the literature to model the non-linear behaviour of complex processes. A detailed investigation of the non-linear PLS algorithms of Baffi et al., (1999(a)) revealed that this algorithm represents a non-linear extension of reduced rank regression. Conventional PLS is based on the maximization of the covariance between the t- and u-scores. It is thus argued that a 'true' non-linear PLS should be a generalization of linear PLS in the sense that when the non-linear function is replaced by a linear function, non-linear PLS should reduce to linear PLS. A 'true' non-linear PLS algorithm, therefore, should be based on the maximization of the 'non-linear' covariance function. After a detailed investigation of the algorithm of Wold et al. (1989) it is proven that, of all the algorithms existing in the literature, it is the only algorithm that attempts to

maximize the covariance based function to identify a non-linear PLS model. Further analysis of the Wold et al., (1989) algorithm revealed some limitations of the algorithm in that all the parameters that influence the non-linear covariance function are not determined so as to maximize the covariance function. To overcome this limitation, two new non-linear PLS algorithms, NLPLS1 and NLPLS2, are proposed. In these algorithms, the 'non-linear covariance' function is maximized over all the parameters (outer weights and inner non-linear model parameters). The difference between NLPLS1 and NLPLS2 being that they use a different set of constraints to make the non-linear covariance function bounded. The application of NLPLS1 and NLPLS2 algorithms to two artificial data sets and a data generated from a simulation of a pH neutralization process showed that these two algorithms perform better than the Wold et al., (1989) algorithm in terms of explaining the variance of the response matrix **Y** and the prediction of the response variables for a given number of latent variables. Of NLPLS1 and NLPLS2, it was observed that the NLPLS2 performs slightly better than NLPLS1. Following a critical analysis, the existing non-linear PLS algorithms are divided into three categories namely quick and dirty, covariance based and error based depending on the underlying objective function used to determine the model parameters.

Conventional linear PLS assumes a static relationship between the input and output variables and therefore, is not suitable in situations where a dynamic model of the process is required. One approach to extending PLS to take into consideration the dynamics of the process is to replace the inner static relationship between the t- and u-scores of conventional PLS by a dynamic relationship. In this approach, linear PLS is first performed on the data matrices **X** and **Y** and a dynamic relationship is then fitted between each pair of t- and u-scores (Lakshminarayan et al., 1997). The limitation of this methodology is that the outer weights are not determined by the dynamics of the process and the dynamic PLS model thus identified may not be optimal. To overcome this limitation, an integrated dynamic PLS is proposed where the dynamic model is fully integrated within the framework of PLS in the sense that all the parameters (outer weights and inner model parameters) of the dynamic PLS model are determined as dictated by the dynamics of the data. The application of this algorithm to model the data collected from an artificial dynamic system and data generated from a co-polymer reactor simulator showed that the integration of the outer weights and inner model parameter result in the dynamic model explaining more variance of the response matrix **Y** and gives better prediction of the response variables for a given set of latent variables.

The abnormal changes in a multivariate Gaussian process can be divided into two categories. While the first category comprises changes that result in the mean vector of the process shifting away from its value defined under normal operating conditions, the changes in the second category are reflected as a change in the variance-covariance structure of the variables. It is shown that a conventional PCA based monitoring scheme, which uses the two statistics, Hotelling T^2 and the Q-statistic, to detect any systematic shift in the variables, is particularly insensitive to small changes in the variance-covariance structure of variables. To overcome this limitation, a new monitoring scheme that derives a monitoring statistic from the PCA model identification procedure is proposed. The key advantage of the proposed scheme is that a change in the variance-covariance structure is reflected as a change in the mean value of a statistic that can be detected optimally. To derive the distribution function of the statistic, and thus to design the change detection algorithm, use is made of the local approach of hypothesis testing. Another important property of the proposed scheme is that it is especially suitable for detecting changes of small magnitude. The application of the proposed scheme to detect changes in the covariance structure of two artificial data sets showed that while the conventional PCA based monitoring scheme failed to detect the small changes, the proposed scheme successfully detected these changes. The scheme was finally applied to detect three different magnitudes of fouling in a heat exchanger in a continuous stirred tank reactor system. It was observed that while the proposed scheme detects all three magnitudes of fouling without any delay, the conventional PCA based monitoring scheme is almost insensitive to small and medium magnitudes of fouling but does give an indication of change, although weak, when the magnitude of fouling is large.

To detect changes in the cross-covariance (between **X** and **Y**) structure, a partial least squares based performance monitoring scheme is proposed. In this scheme, the derivation of a monitoring statistic requires that a recursive algorithm exists for identifying the PLS model parameters. A new recursive PLS algorithm is first derived using the Least Mean Squares (LMS) algorithm. The algorithm is tested on an artificial data set and is observed to converge to the solution of the NIPALS algorithm. A monitoring statistic is then derived from this algorithm. The key properties of the statistic derived from the recursive algorithms are (1) a change in the cross-covariance structure is reflected as a change in the mean value of the statistic and (2) it is especially suitable for detecting small changes in the cross-covariance structure. The distribution function of the statistic derived from the recursive PLS algorithm is determined using the local approach of hypothesis testing. The proposed scheme was first applied to detect changes in the parameter of an artificial system before applying it for the detection of fouling in a heat exchanger in a continuous stirred tank reactor. It was observed

that while the proposed scheme detects changes of all magnitudes, the conventional PLS based monitoring scheme can detect changes of large magnitudes only and remains almost insensitive to changes of small magnitudes.

8.3 Recommendations

Based on the research undertaken in this thesis, certain issues need to be investigated and explored further. Some recommendations for future directions are given below.

In Chapter 2, the performance of PLS as a parameter estimator has been evaluated on an artificial data set only. The application of PLS based parameter estimation to a practical physical/chemical data remains to be addressed.

In Chapter 3, two non-linear PLS algorithms that are based on the 'non-linear covariance maximization' have been proposed. The non-linearity considered is quadratic and thus the issue of generalizing the algorithm to a more general non-linearity e.g. feedforward neural network with a one or more hidden layers, needs to be addressed. Furthermore, extensions to the modelling of non-linear and dynamic data also need to be explored. Finally 'non-linear covariance' based algorithms need to be applied for process monitoring and control.

In Chapter 4, the order of the inner dynamic models was selected using a subjective approach. The reason is that there is no relationship between the number of lags and delays of the measured variables and the lags and delays of the latent variables, which makes the selection of the number of lags and delays in the inner dynamic model difficult. This issue needs to be investigated further. Also the scheme for integrating the outer weights needs to be extended to the situation when the inner model is non-linear and dynamic, e.g. Hammerstein and Weiner models.

In Chapter 6, a new statistic to detect a change in the variance-covariance structure has been proposed. The practical application of the scheme requires the selection of window parameters which in this thesis has been selected using adhoc approaches. The issue of systematically selecting the window size parameters and their effect on the optimality of the change detection algorithm need to be investigated further. Additionally, the extension of this scheme to monitor the cross-covariance structure in a PLS based monitoring scheme needs to be undertaken. Also since the derivation of the monitoring statistic requires a model

identification procedure, the scheme can be extended to detect faults in a non-linear and/or dynamic PCA based monitoring scheme.

Finally in Chapter 7, a recursive version of the PLS algorithm is proposed. Although the algorithm can be used to update the parameters of a PLS model on-line in a non-stationary environment, the issue of its comparison with other recursive PLS algorithm with respect to speed still remains to be addressed.

Appendix 1

A.1 Learning rate_ $\eta = 0.001$

The plots of the square of the norm of the error versus number of iterations for the first three **w**-weight vectors, **v**-weight vectors and the inner regression coefficients with the learning rate parameter, $\eta = 0.001$ are shown in Figures A.1, A.2 and A.3 respectively. It is seen from these figures that for this learning rate convergence for the parameters is slow and some of the parameters ($\mathbf{w}_2$, $\mathbf{w}_3$, $\mathbf{v}_2$, $\mathbf{v}_3$ and b_3) have not converge even after 100 iterations.

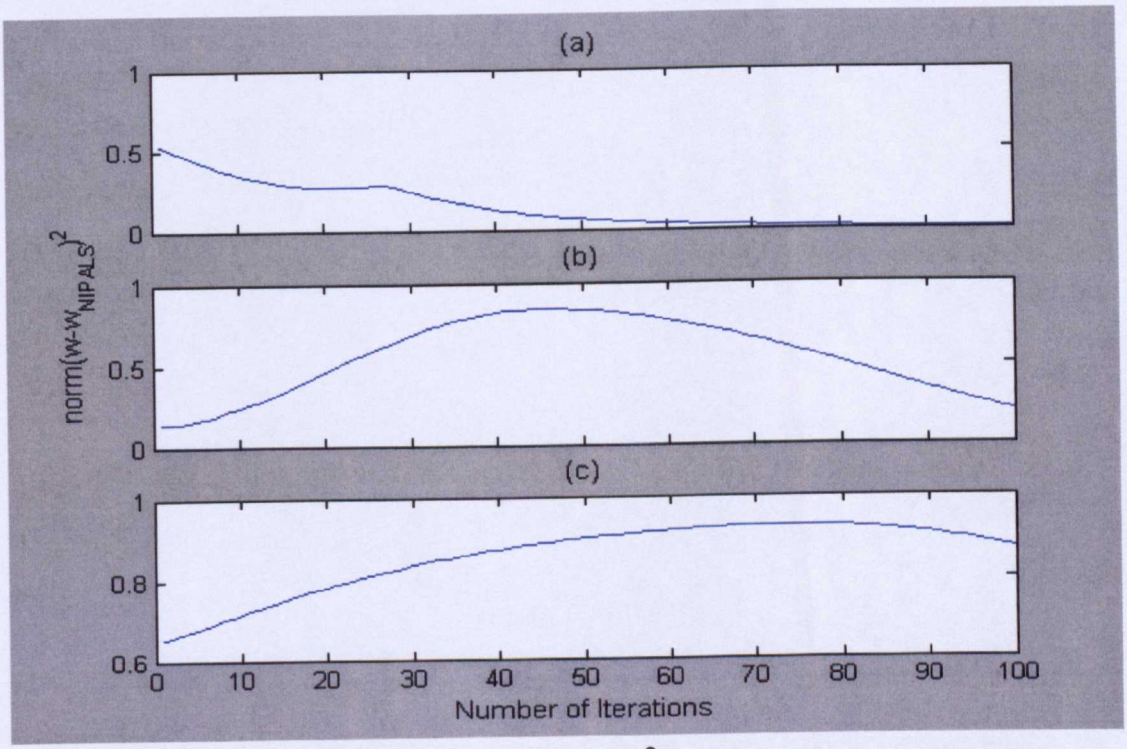


Figure A.1: Plot of estimation error $\|\mathbf{w}\text{-}\mathbf{w}_{\text{NIPALS}}\|^2$, where $\mathbf{w}_{\text{NIPALS}}$ is the PLS solution from the NIPALS algorithm versus number of iterations for the first three solutions of **w** (a) $\mathbf{w}_1$ (b) $\mathbf{w}_2$ (c) $\mathbf{w}_3$ for learning rate $\eta = 0.001$.

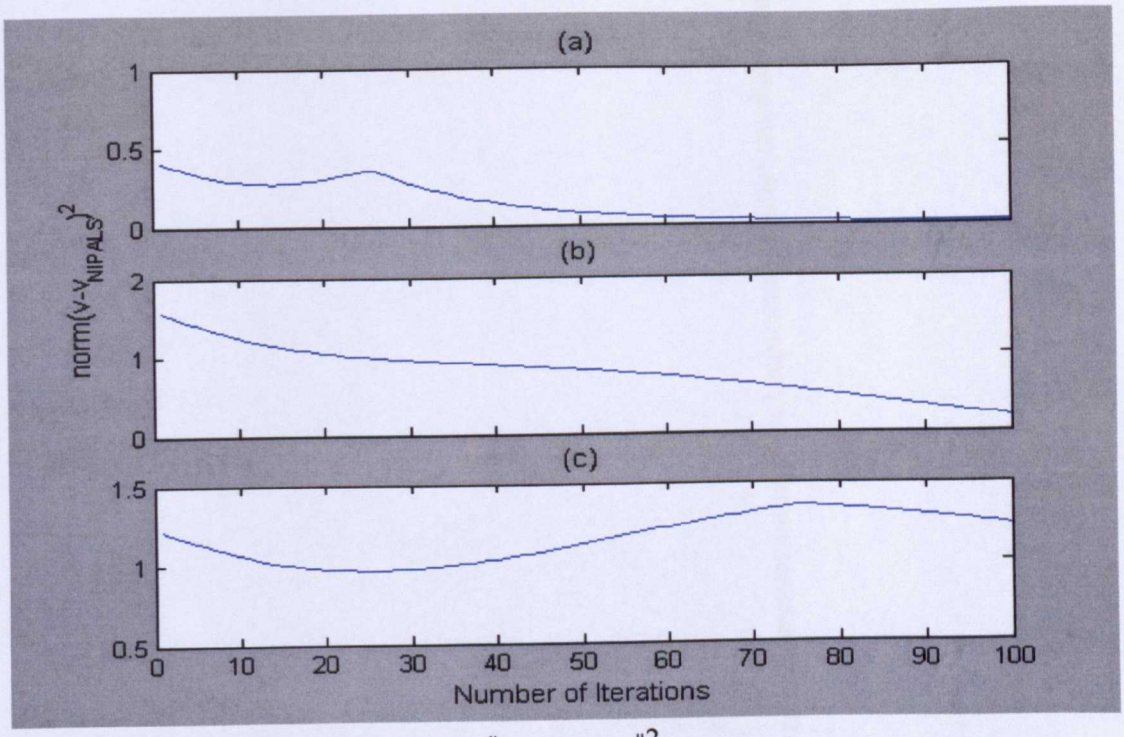


Figure A.2: Plot of estimation error $\|\mathbf{v} - \mathbf{v}_{NIPALS}\|^2$, where $\mathbf{v}_{NIPALS}$ is the PLS solution from the NIPALS algorithm, against number of iterations for the first three solutions of v (a) $\mathbf{v}_1$ (b) $\mathbf{v}_2$ (c) $\mathbf{v}_3$ for learning rate $\eta = 0.001$

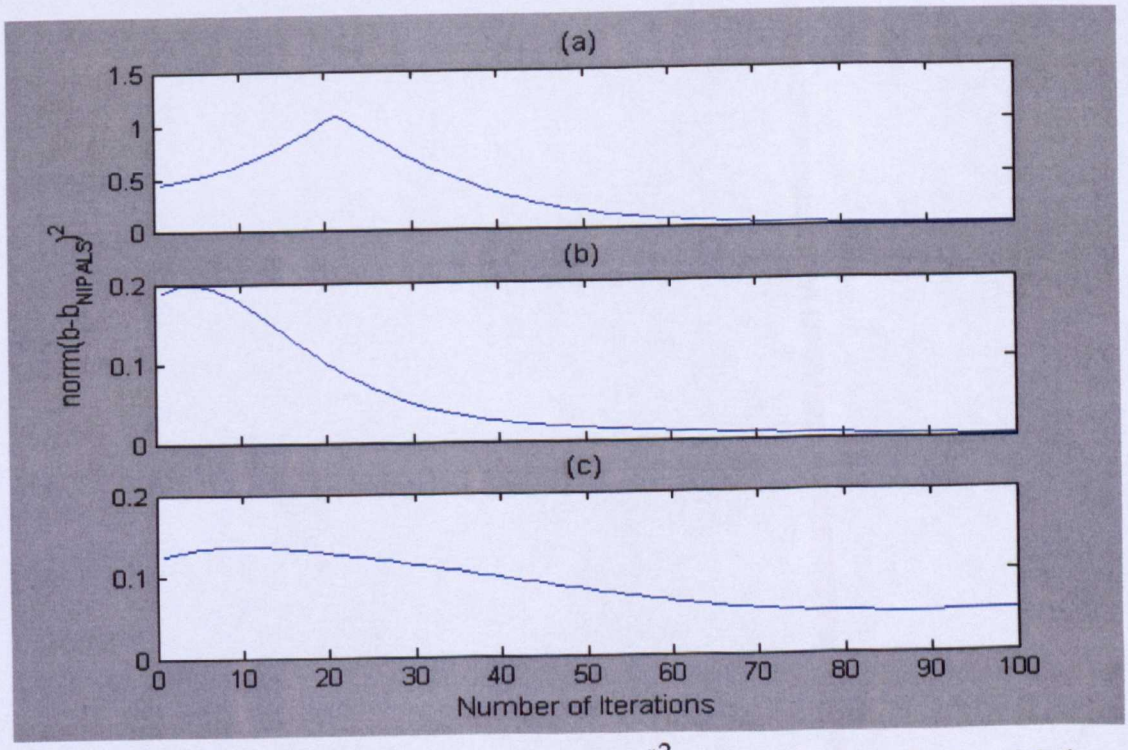


Figure A.3: Plot of estimation error $\|b - b_{NIPALS}\|^2$, where b_{NIPALS} is the PLS inner regression coefficient from the NIPALS algorithm, versus number of iterations for the first three inner regression coefficients(a) b_1 (b) b_2 (c) b_3 for $\eta = 0.001$

A.2 Learning rate_$\eta = 0.04$

The plots of the square of the norm of the error versus number of iterations for the first three **w**-weight vectors, **v**-weight vectors and the inner regression coefficients with the learning rate parameter $\eta = 0.04$ are shown in Figures A.4, A.5 and A.6 respectively. It is seen from the figures that error in many of the parameters $(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, b_1, b_2)$ does not become zero after convergence, which implies that the algorithm has not converged to the 'true' (NIPALS) parameters.

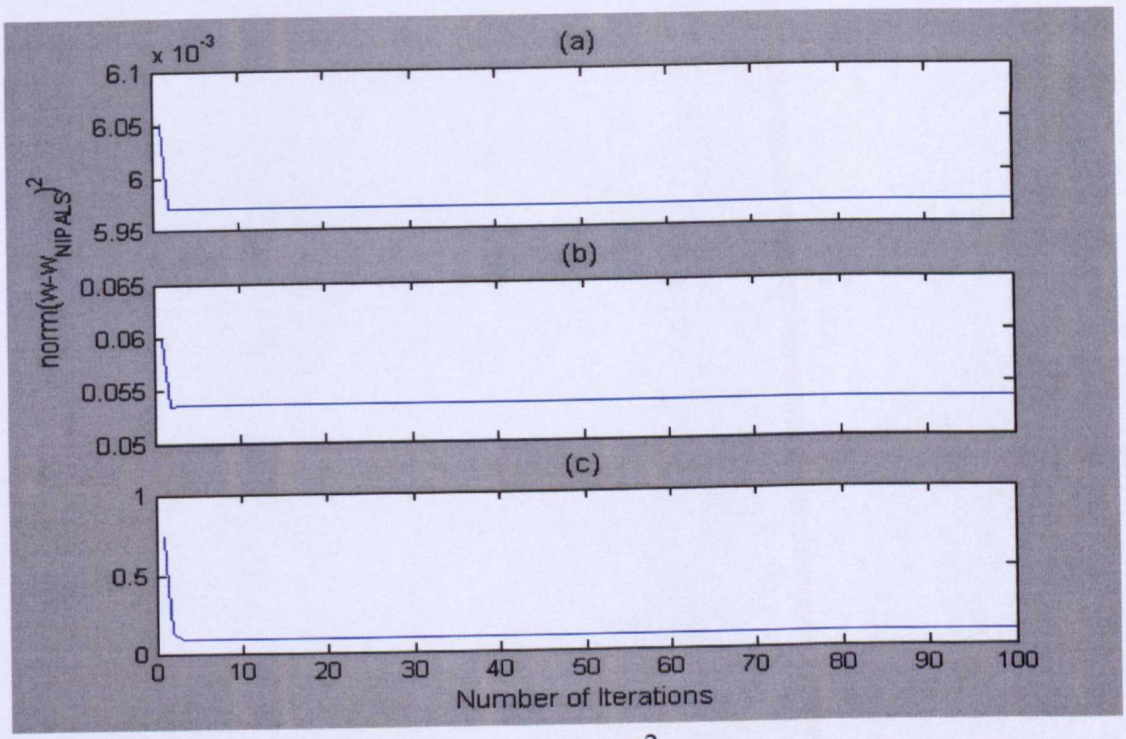


Figure A.4: Plot of estimation error $\|\mathbf{w}\text{-}\mathbf{w}_{\text{NIPALS}}\|^2$, where $\mathbf{w}_{\text{NIPALS}}$ is the PLS solution from the NIPALS algorithm versus number of iterations for the first three solutions of **w** (a) $\mathbf{w}_1$ (b) $\mathbf{w}_2$ (c) $\mathbf{w}_3$ for $\eta = 0.04$

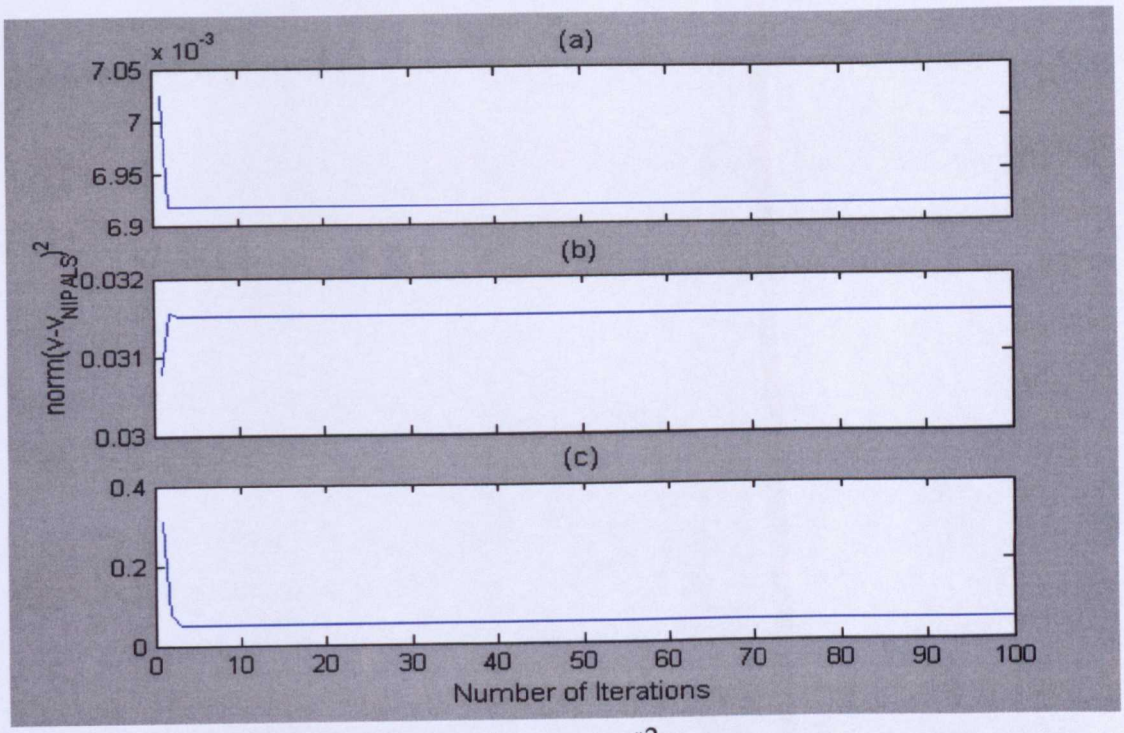


Figure A.5: Plot of estimation error $\|\mathbf{v} - \mathbf{v}_{NIPALS}\|^2$, where $\mathbf{v}_{NIPALS}$ is the PLS solution from the NIPALS algorithm, against number of iterations for the first three solutions of v (a) $\mathbf{v}_1$ (b) $\mathbf{v}_2$ (c) $\mathbf{v}_3$ for $\eta = 0.04$

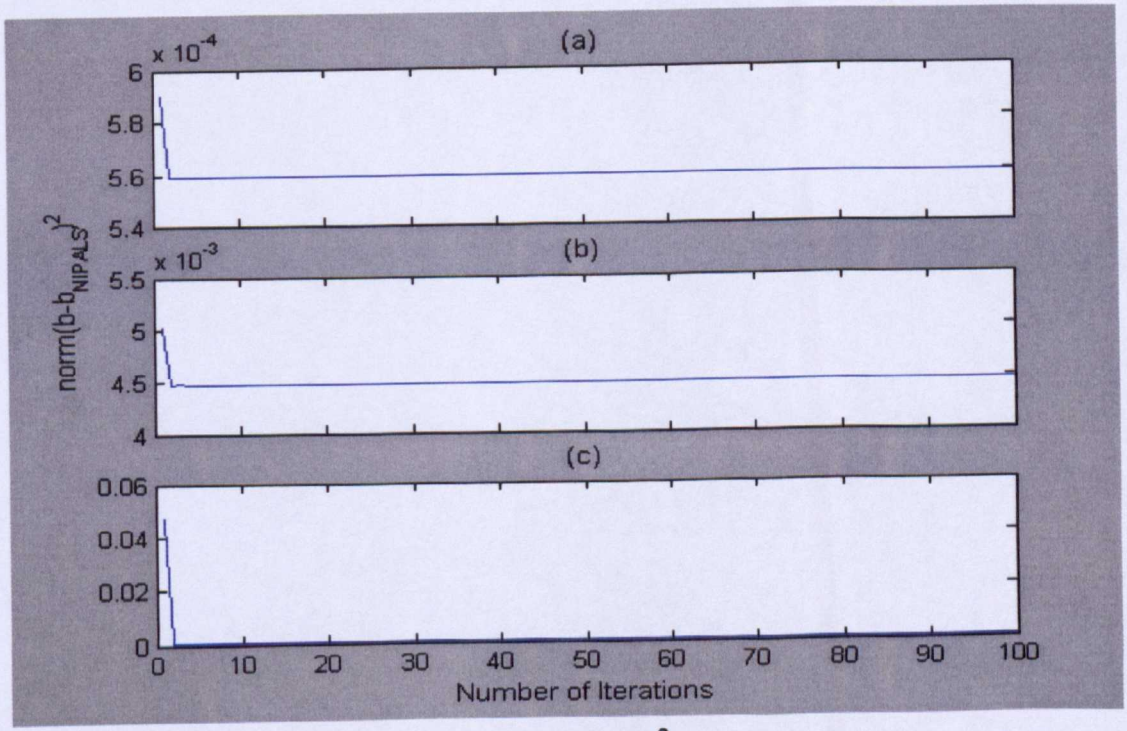


Figure A.6: Plot of estimation error $\|b - b_{NIPALS}\|^2$, where b_{NIPALS} is the PLS inner regression coefficient from the NIPALS algorithm, versus number of iterations for the first three inner regression coefficients(a) b_1 (b) b_2 (c) b_3 for $\eta = 0.04$

References

Achilias, D.S. and Kiparissides, C. (1994). On the validity of the steady state approximations in high conversion diffusion-controlled free-radical copolymerization reactions. *Polymer,* 35(8), 1714 - 1721.

Anderson, T.W. (1984). *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons.

Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68, 337-404

Baffi, G., Martin, E.B., Morris, A.J. (1999(a)). Non-linear projection to latent structures revisited, the quadratic PLS algorithm. *Computers and Chemical Engineering*, 23, 395-411.

Baffi, G., Martin, E.B., Morris, A.J. (1999(b)). Non-linear projection to latent structures revisited, the neural network PLS algorithm. *Computers and Chemical Engineering*, 23(9), 1293-1307.

Baffi, G., Martin, E.B., Morris, A.J. (2000). Non-linear dynamic projection to latent structures modelling. *Chemometrics and Intelligent Laboratory Systems*, 52(1), 5-22.

Bakshi, B.R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE*, 44(7), 1596-1610.

Bakshi, B.R. (1999). Multiscale analysis and modelling using wavelets. *Journal of Chemometrics*, 13, 415-434.

Banks, D. (1993). Is industrial statistics out of control?. *Statistical Science,* 8(4), 356-409.

Bannour, S. and Sadjadi, M.R.A. (1995). Principal component extraction using recursive least squares. *IEEE Transactions on Neural Networks*, 6(2), 457-469.

Basseville, M. and Nikiforov, I. (1993). *Detection of abrupt changes - Theory and applications*. Prentice Hall Information and System Sciences Series, Englewood Cliffs, NJ.

Basseville, M. (1998). On-board component fault detection and isolation using the statistical local approach. *Automatica*, 34(11), 1391-1416.

Benveniste, A., Basseville, M., Moustakides, G.V. (1987). The asymptotic local approach to change detection and model validation. *IEEE Transactions on Automatic Control*, 32(7), 583-592.

Berglund, A. and Wold, S. (1997). INLR, Implicit non-linear latent variable regression. *Journal of Chemometrics*, 11, 141-156.

Braak, T.C.J.F. and Jong, D.S. (1998). The objective function of partial least squares. *Journal of Chemometrics*, 12, 41 - 54.

Burnham, A.J., MacGregor, J.F., Viveros, R. (2001). Interpretation of regression coefficients under a latent variable regression model. *Journal of Chemometrics*, 15, 265-284.

Cam, L. (1986). *Asymptotic methods in statistical decision theory*. Springer, New-York.

Cherkassky, V., Gehring, D., Mulier, F. (1996). Comparison of adaptive methods for function estimation from samples. *IEEE Transactions on Neural Networks*, 7(4), 969-984.

Christer, A.H and Wang, W. (1995). Simple condition monitoring model for a direct monitoring process. European Journal of Operation Research, 82(2), 258-269.

Cybenko (1989). Approximation by superposition of a sigmoidal function. *Math. Control, Signals, Systems*, 2, 303 -314.

Dayal, B.S. and MacGregor, J.F. (1997 (a)). Recursive exponentially weighted PLS and its application to adaptive control and prediction. *Journal of Process Control*, 7(3), 169-179.

Dayal, B.S. and MacGregor, J.F. (1997(b)). Improved PLS algorithms. *Journal of Chemometrics*, 11, 73-85.

Denham, M.C. (1997). Prediction Interval for Partial Least Squares. *Journal of Chemometrics*, 11, 39-52.

Diamantaras, K.I. (1994). Cross correlation neural network models. *IEEE Transactions on Signal Processing*, *42*(11), 3218-3223.

Diamantaras, K.I. and Kung, S.Y. (1996). *Principal component neural networks- Theory and applications*. John Wiley & Sons, INC, New York.

Dijkstra, T. (1983). Some comments on maximum likelihood and partial least squares. *Journal of Econometrics*, (22), 67-90.

Doymaz, F., Palazoglu, A., Romagnoli, J.A. (2003). Orthogonal non-linear partial least squares regression. *Industrial Engineering Chemistry Research*, 42, 5836-5849.

Draper, N.R. and Smith, H. (1998). *Applied regression analysis*. Wiley-Interscience.

Dunia, R., Qin, S.J., Edgar, T.F., McAvoy, T.J. (1996). Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42(10), 2797-2811.

Englezos, P. and Kalogerakis, N. (2000). *Applied parameter estimation for chemical engineers*. Marcel Dekkar.

Feng, D.Z., Bao, Z., Shi, W.X. (1998). Cross-correlation neural network models for the smallest singular component of general matrix. *Signal Processing*, 64, 333-346.

Foldiak, P. (1989). Adaptive network for optimal linear feature extraction. *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, Washington, D.C. June 18-22, 401-405.

Fornell, C., and Bookstein, F. (1982). Two structural equation models, LISREL and PLS applied to consumer exit -voice theory. *Journal of Marketing Research*, 19(4), 440-452.

Frank, I. (1987). Intermediate Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems*, 1(3), 233-242.

Frank, I.E. (1990). A non-linear PLS model. *Chemometrics and Intelligent Laboratory Systems*, 8, 109-119.

Ganadesikan, R. (1977). *Methods for statistical data analysis of multivariate observations.* Wiley, New York.

Geladi, P. (1988). Notes on the history and nature of partial least squares modelling. *Journal of Chemometrics*, 2, 231-246.

Geladi, P. (1992). Herman Wold, the father of PLS. *Chemometrics and Intelligent Laboratory Systems*, 15(1), R7-R8.

Geladi, P. and Grahn, H. (1997). *Multivariate Image Analysis*. John Wiley and Sons.

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

Golub, G.H. and Loan, C.F.V (1996). *Matrix Computation*. The Johns Hopkins University Press.

Haaland, D.M. and Thomas, E.V (1988(a)). Partial least squares methods for spectral analyses-1, Relation to other quantitative calibration methods and extraction of qualitative information. *Analytical Chemistry*, 60, 1193-1202.

Haaland, D.M. and Thomas, E.V (1988(b)). Partial least squares methods for spectral analyses-2, Application to simulated and glass spectral data. *Analytical Chemistry*, (60), 1202-1208.

Haykin, S. (1995). *Adaptive filter theory*. Prentice Hall, Englewood Cliffs, NJ.

Hebb, D.O. (1949). *The Organization of Behaviour*. Wiley, New-York.

Helland, I.S. (1988). On the structure of partial least squares regression. *Communications in Statistics – Simulations*, 17(2), 581-607.

Helland, I.S. (1990). Partial least squares and statistical models. *Scandinavian Journal of Statistics*, 17, 97-114.

Helland, I.S. (2001). Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 97-107.

Helland, I.S. and Almoy, T. (1994). Comparison of prediction methods when only a few components are relevant. *Journal of the American Statistical Association*, 29, 583-591.

Helland, K., Bernsten, H.E., Borgen, O.S, Martens, H. (1992). Recursive algorithm for partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 14, 129-137.

Helland, K., Berntsen, H.E., Borgen, O.S., Martens, H. (1991). Recursive algorithm for partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 14, 129-137.

Henson, M.A. and Seborg, D.E (1994). Adaptive non-linear control of a pH neutralization process. *IEEE Transactions on Control Systems Technology*, 3, 169-183.

Hiden, H., McKay, B., Willis, M., Montague, G. (1998). Non-linear partial least squares using genetic programming. *Proceedings of the 2nd Annual Conference on Genetic Programming, Madison, Wisconsin, USA*, 128-133.

Hill, R.C., Fombay, T.B., Johnson, S.R. (1977). Component selection norms for principal component regression. *Communications in Statistics, Theory and Methods*, 6, 309-334.

Hinkley, D.V. (1969). Inference about the intersection in two-phase regression. *Biometrika*, 56(3), 495-504

Hinkley, D.V. (1970). Inference about the change point in a sequence of random variables. *Biometrika*, 57(1), 1-17.

Hinkley, D.V. (1971). Inference about the change point from cumulative-sum tests. *Biometrika*, 58(3), 509-523.

Holcomb, T.R. and Morari, M. (1992). PLS/Neural Networks. *Computers and Chemical Engineering*, 16(4), 393 - 411.

Höskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics*, 2, 211-228.

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

Hotelling, H. (1933). Analysis of complex statistical variables into principal components. *The Journal of Educational Psychology*, 24, 417-441.

Hotelling, H. (1947). *Multivariate quality control, Techniques of statistical analysis*. Edited by Eisenhart, M.W.H.C., Wallis, W.A., McGraw- Hill, New York, 111-184.

Hotelling, H. (1957). The relations of newer statistical methods to factor analysis. *British Journal of Statistical Psychology*, 10, 69-79.

Hulland, J. (1999). Use of PLS in strategic management research, A review of four recent studies. *Strategic Management Journal*, 20(2), 195-204.

Jackson, J.E. (1956). Quality control methods for two related variables. *Industrial Quality Control*, 12(7), 2-6.

Jackson, J.E. (1959). Quality control methods for several variables. *Technometrics*, 1(4), 359-377.

Jackson, J.E. (1980). Principal components and factor Analysis-I, Principal components. *Journal of Quality Technology*, 12(4), 201.

Jackson, J.E. (1991). *A user's guide to principal components*. John Wiley & Sons, New York.

Jackson, J.E. and Morris, R.H. (1957). An application of multivariate quality control to photographic processing. *Journal of the American Statistical Association*, 52, 186-199.

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

Jain, A.K. (1989). *Fundamentals of digital image processing*. Prentice-Hall, Englewood Cliffs, NJ.

Jia, F., Martin, E.B., Morris, A.J. (1998). Non-linear multiway principal component analysis for process monitoring. *Computers and Chemical Engineering*, 22, S851-S854.

Jia, F., Martin, E.B., Morris, A.J. (2000). Non-linear principal component analysis with application to fault detection. *International Journal of System Science*, 11, 1473-1487.

Johansen, T.A. and Foss, B.A. (1997). Operating regime based process modelling and identification. *Computers and Chemical Engineering*, 21, 159-176.

Jolliffe, I.T. (1986). *Principal Component Analysis*. Springer-Verlag, New York.

Jong, D.S. (1995). PLS Shrinks. *Journal of Chemometrics*, 9, Short Communication, 323-326

Jong, D.S. and Braak, C.J.F.T. (1994). Comments on the PLS kernel algorithm. *Journal of Chemometrics*, 8, 169-174.

Kano, M., Haebe, S., Hashimoto, I., Ohno, H. (2001). A new multivariate statistical process monitoring method using principal component analysis. *Computers and Chemical Engineering*, 25, 1103-1113.

Kaspar, M.H. and Ray, W.H (1992). Chemometrics methods for process monitoring and high performance controller design. *AIChE Journal*, 38(10), 1593-1608.

Kaspar, M.H. and Ray, W.H. (1993(a)). Dynamic PLS modelling for process control. *Chemical Engineering Science*, 48(20), 3447-3461.

Kaspar, M.H. and Ray, W.H. (1993(b)). Partial least squares modelling as successive singular value decomposition. *Computers and Chemical Engineering*, 17(10), 985-989.

Kourti, T. and MacGregor, J.F (1995). Process analysis, monitoring and diagnosis using multivariate methods. *Chemometrics and Intelligent Laboratory Systems*, 20, S745-S750.

Kourti, T. and MacGregor, J.F. (1994). Multivariate SPC methods for monitoring and diagnosis of process performance. *Proceedings of PSE*, Kyungu, S.Korea, 739-746.

Kourti, T., Lee, J., MacGregor, J.F. (1996). Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering*, 20, 745-750.

Kramer, M.A. (1991). Non-linear principal component analysis using auto-associative neural networks. *AIChE Journal*, 37(2), 233-243.

Kramer, M.A. and Mah, R.S.H., (1994). Model based monitoring. *International Conference on Foundations of Computer Aided Process Operations*, Austin, TX.

Kresta, J.V., MacGregor, J.F., Marlin, T.E. (1989). Multivariate statistical monitoring of process performance. *AIChE Annual Meeting*, San Francisco, CA.

Kresta, J.V., MacGregor, J.F., Marlin, T.E. (1991). Multivariate statistical monitoring of process operating performance. *Can. Journal of Chem. Engg.*, 69, 35-47.

Ku, W., Storer, R.H., Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30, 179-196.

Kung, S.Y., Diamantaras, K.I, Taur, J.S (1994). Adaptive principal component extraction (APEX) and applications. *IEEE Transactions on Signal Processing*, 42(5), 1202-1217.

Lakshminarayan, S., Shah, S.L, Nandkumar, K. (1997). Modelling and control of multivariable processes, The dynamic projection to latent structure approach. *AIChE Journal*, 43, 2307-2323.

Lane, S., Martin, E.B., Morris, A.J., Gower, P. (2003). Application of exponentially weighted principal component analysis for the monitoring of a polymer film manufacturing process. *Transactions of the Institute of Measurement and Control*, 25, 17-35.

Lane, S. (2000). *Statistical performance monitoring using group models*. PhD Thesis, School of Chemical Engineering and Advanced Materials, University of Newcastle, Newcastle upon Tyne (UK).

Lennox, B., Hiden, H.G., Montague, G.A., Kornfield, G., Goulding, P.R. (2000). Application of multivariate statistical process control to batch operations. *Computers and Chemical Engineering*, 24, 291-296.

Lennox, B., Hiden, H.G., Montague, G.A., Kornfield, G., Goulding, P.R. (2001). Process monitoring of an industrial fed-batch fermentation. Biotechnology and Bioengineering, 74, 125-135.

Li, B., Martin, E.B., Morris, A.J. (2001). Box-Tidwell transformation based partial least squares regression. *Computers and Chemical Engineering*, 25, 1219-1233.

Li, W., Yu, H., Valle, C.S., Qin, S.J. (2000). Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10, 471-486.

Lindgren, F., Geladi, P., Wold, S. (1993). The kernel algorithms for PLS. *Journal of Chemometrics*, 7, 45-59.

Ljung, L. (1999). *System Identification, Theory for the User*. Prentice Hall, New York.

Ljung, L. and Soderstrom, T. (1983). *Theory and practice of recursive identification*. MIT Press, London.

Lorber, A., Wangen, L., Kowalski, B. (1987). A theoretical foundation for the PLS algorithm. *Journal of Chemometrics*, 1, 19-31

Lorden, G. (1971). Procedures for reacting to a change in distribution. *Annals Mathematical Statistics*, 44, 1897-1908.

Lorden, G. (1973). Open-ended tests for Koopman- Darmois families. *Annals Statistics*, 1(4), 633-643.

Louwerse, D.J and Smilde, A.K (1999). Multivariate statistical process control of batch proceses based on three way models. *Chemical Engineering Science*, 55, 1225-1235.

Lucas, J.M. and Saccucci, M.S. (1990). Exponential weighted moving average control schemes, Properties and enhancements. *Technometrics*, 32(1), 1-12.

MacGregor, J.F (1994). Statistical process control of multivariate processes. *IFAC, ADCHEM*, Kyoto, Japan.

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

MacGregor, J.F. and Nomikos, P. (1992). Monitoring batch processes. *Proceedings NATO*, Advanced study Institute for Batch Processes, Antalya, Turkey.

MacGregor, J.F., Marlin, T.E., Kresta, J., Skagerberg, B. (1991). Multivariate statistical methods in process analysis and control. *Proceedings of the AIChE Symposium, Fourth International Conference on Chemical Process Control*, New York.

MacGregor, J.F., Nomikos, P., Kourti, T. (1994). Multivariate Statistical Process Control of Batch Processes using PCA and PLS. *IFAC ADCHEM*, Kyoto, Japan.

Malthouse, E.C. (1995). *Non-linear partial least squares*. PhD Thesis, Department of Statistics, Northwestern University, Evanston, IL.

Malthouse, E.C., Tamhane, A.C., Mah, R.S.H (1997). Non-linear partial least squares. *Computers and Chemical Engineering*, 21(8), 875-890.

Manne, R. (1987). Analysis of two partial least squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2, 187-197.

Mardia, K.V., Kent, J.T., Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press.

Martens H., and Næs, T., (1987). Multivariate Calibration by Data Compression, Williams P. (ed.), *NIR Analysis*, American Association of Cereal Chemistry, St. Paul, Minnesota

Martens, H. (2001). Reliable and relevant modelling of real world data, a personal account of the development of PLS regression. *Chemometrics and Intelligent Laboratory Systems*, 58, 85-95.

Martens, H. and Næs, T. (1989). *Multivariate calibration*. Wiley, Chichester.

Martin, E.B., Morris, A.J., Zhang, J. (1996). Process performance monitoring using multivariate statistical process control. *Proceedings of IEE Control Theory and Applications*, Part D, 143(2), 132-144.

Miller, P., Swanson, R.E., Heckler, C.F. (1993). Contribution plots: The missing link in multivariate quality control. Proceedings of the 37th Annual Fall Conference, Rochester, New-York.

Min, K.G., Han, I., Han, C. (2002). Iterative error-based non-linear PLS method for non-linear chemical process modelling. *Journal of Chemical Engineering of Japan*, 35(7), 613-625.

Montgomery, D.C. (1991). *Introduction to Statistical Quality Control*. John Wiley & Sons, Singapore.

Montgomery, D.C. and Peck, E.A. (1982). *Introduction to Linear Regression Analysis*. John Wiley & Sons, Inc.

Morud, T.E. (1996). Multivariate statistical process control: Example from the chemical process industry. Journal of Chemometrics, 10, 669-675.

Moustakides, G.V. (1986). Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4), 1379-1387.

Næs T., Irgens, C., Martens, H., (1986). Comparison of linear statistical methods for calibration of NIR instruments. *Applied Statistics*, 35(2), 195-206.

Næs, T. and Martens, H. (1985). Comparison of prediction methods for collinear data. *Communication in Statistics - Simulations and Computation*, 14, 545-576.

Neumaier, A. and Schneider, T. (2001). Estimation of parameters and eigenmodes of multivariate autoregressive models. ACM Transactions on Mathematical Software, 27(1), 27-57.

Nomikos, P. and MacGregor, J.F. (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40(8), 1361-1375

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

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15, 267-273.

Oja, E. (1989). Neural networks, principal components and subspaces. *International Journal of Neural Systems*, 1(1), 61-68.

Oppenheim, A.V. and Schafer, R.W. (1989). *Discrete -time signal processing*. Prentice Hall New Jersey.

Page, E.S. (1954). Continuous inspection schemes. *Biometrika*, 41, 100-114.

Partridge, M. and Calvo, R.A. (1998). Fast dimensionality reduction and simple PCA. *Intelligent Data Analysis*, (2), 203-214.

Patwardhan, R., Lakshminarayan, S., Shah, S. (1998). Constrained non-linear MPC using Hammerstein and Wiener models: PLS Framework. *AIChE Journal*, 44(7), 1611-1622.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 559-572.

Phatak, A., Reily, P.M., Penlidis, A. (1993). An approach to interval estimation in partial least squares regression. *Analytica Chimica Acta*, 277, 495-501.

Phatak, A., Reily, P.M., Penlidis, A. (2002). The asymptotic variance of the univariate PLS estimator. *Linear Algebra and its Applications*, 354, 245-253.

Philips, M.J. (1969). A survey of sampling procedures for continuous production. *Journal of Royal Statistical Society*, A-132(2), 205-228.

Piovoso, M.J and Kosanovich, K.A. (1991). Monitoring process performance in real time. *American Control Conference*, Chicago, Illinois.

Piovoso, M.J and Kosanovich, K.A (1992). Process data chemometrics. *IEEE Transactions on Instrumentation and Measurement*, 41(2), 262-268.

Piovoso, M.J and Kosanovich, K.A (1994). Application of multivariate statistical methods to process monitoring and controller design. *International Journal of Control*, 59(3), 743-765.

Qin, S.J. (1993). A recursive PLS algorithm for system identification. *AIChE Annual Meeting*.

Qin, S.J. (1998). Recursive PLS algorithm for adaptive data modelling. *Computers and Chemical Engineering*, 22(4/5), 503-514.

Qin, S.J. and McAvoy, T.J (1992(a)). A data based process modelling approach and its applications. *Proceedings of IFAC conference on dynamics and control of reactors* (DYCORD 92).

Qin, S.J. and McAvoy, T.J. (1992(b)). Non-linear PLS modelling using neural networks. *Computers and Chemical Engineering*, 16(8), 379-391.

Rannar, S., Lindgren, F., Geladi, P., Wold, S. (1994). A PLS kernel algorithm for data sets with many variables and fewer objects- Part 1: Theory and algorithm. *Journal of Chemometrics*, 8, 111-125.

Reinsel, G. and Velu, R.P. (1998). *Multivariate reduced rank regression: Theory and applications*: Springer-Verlag.

Ricker, N.L. (1988). the use of biased least - squares estimators for parameters in discrete - time pulse response models. *Industrial and Engineering Chemistry Research*, 27, 343 - 350.

Roberts, S.W. (1959). Control chart tests based on geometric moving average. *Technometrics*, 1, 239-250.

Rosipal and Trejo, L.J (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2, 97-123.

Sanger, T.D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6), 459-473.

Sellin, N. (1995). Partial least Squares modelling in research on educational achievement. In Bos, W and Lehmann, R.H.(eds.), *Reflections on Educational Achievement*. Papers in honour of T.Neville Postlethwaite, Waxmann, New-York.

Shao, R., Jia, F., Martin, E.B, Morris, A.J. (1999). Wavelets and non-linear principal component analysis for process monitoring. *Control Engineering Practice*, 7, 865-879.

Shewhart, W.A. (1931). *Economic control of quality of manufactured product*. MacMillan & Co. Limited, London.

Soderstrom, T. and Stoica, P. (1988). *System Identification*. Prentice Hall New York.

Stoica, P. and Soderstrom, T. (1998). Partial least squares: A first-order analysis. *Scandinavian Journal of Statistics*, 25(1), 17-25.

Therrein, C.W. (1992). *Discrete random signals and statistical signal processing*. Pentice-Hall, Englewood Cliffs, NJ.

Thomas, E.V. and. Haaland, D.M., (1990). Comparison of multivariate methods for quantitative spectral analysis. *Analytical Chemistry*, 62, 1091-1099.

Walczak, B. and Massart, D.L. (1996). Radial basis functions- partial least squares approach as a flexible non-linear regression technique. *Analytica Chimica Acta*, 331, 177-185.

Wald, A. (1947). *Sequential Analysis*. John Wiley & Sons, New York.

Wang, X., Kruger, U., Lennox, B. (2003). Recursive partial least squares algorithms for monitoring complex industrial processes. *Control Engineering Practice*, 11, 613-632.

Wentzell, P.D. and Montoto, L.V. (2003). Comparison of principal component regression and partial least squares through generic simulations of complex mixtures. *Chemometrics and Intelligent Laboratory Systems*, 65, 257-279.

Wickstrom, C., Albano, C., Eriksson, L., Friden, H., Johansson, E., Nordahl, A., Rannar, S., Sandberg, M., Wold, K.N, Wold, S. (1998). Multivariate process and quality control

monitoring applied to an electrolysis process- part 1: Process supervision with multivariate control charts. *Chemometrics and Intelligent Laboratory Systems*, 42, 221-231.

Widrow, B. and Stearns, S.D (1985). *Adaptive Signal Processing*. Prentice Hall, New Jersey.

Wilson, D.J.H., Irwin, G.W., Lightbody, G. (1997). Non-linear PLS modelling using radial basis functions. *Proceedings of IEEE American Control Conference.*

Wold, H. (1966 (a)). Estimation of principal components and related models by iterative least squares, In Krishnaiah, P.R (eds.), *Multivariate Analysis*, Academic Press, New York, 391-420.

Wold, H. (1966 (b)). Non-linear estimation by iterative least Squares. *Research Papers in Statistics*. F. David, John Wiley New York, 411-444.

Wold, S. (1978). Cross-validatory estimation of number of principal components in factor and principal component models. *Technometrics*, 20, 397-404.

Wold, S. (1992). Non-linear partial least squares modelling-II: spline inner function. *Chemometrics and Intelligent Laboratory Systems*, 14, 71-84.

Wold, S. (1994). Exponentially weighted moving principal component analysis and projections to latent structures. *Chemometrics and Intelligent Laboratory Systems*, 23, 149-161

Wold, S. (2001). Personal memories of the early PLS development. *Chemometrics and Intelligent Laboratory Systems*, 58, 83-84.

Wold, S., Martens, H., Wold, H. (1983 (a)). The multivariate calibration problem in chemistry solved by the PLS method. *Lecture Notes in Mathematics*, 973. Springer-Verlag, Heidelberg, 286-293.

Wold, S., Martens, H., Wold, H. (1983 (b)). A multivariate calibration problem in analytical chemistry solved by partial least squares models in latent variables. *Analytica Chimica Acta*, 150, 61-70.

Wold, S., Ruhe, A., Wold, H., Dunn, W.J. (1984). The collinearity problem in linear regression: The PLS approach to generalized inverses. *SIAM Journal of Scientific Statistical Computations*, 5, 735-743.

Wold, S., Sjostrom, M., Eriksson, L. (2001). PLS regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 109-130.

Wold, S., Trygg, J., Berglund, A., Antti, H. (2001). Some recent developments in PLS modelling. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 131-150.

Wold, S., Wold, K.N, Skagerberg, B (1989). Non-linear PLS modelling. *Chemometrics and Intelligent Laboratory Systems*, 7, 53-65.

Wold, S., Kettaneh, N., Friden, H., Holmberg, A. (1998). Modelling and diagnosis of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, 44, 331-340.

Zhang, Q., Basseville, M., Benveniste, A. (1994). Early warning of slight changes in systems and plants with application to condition based monitoring. *Automatica*, 30(1), 95-114.

Zhang, J. (1991). *Application of expert systems in on-line process control and fault diagnosis*. PhD Thesis, Department of Electrical, Electronic and Information Engineering, City University, London (U.K).