Moving PCA FOR PROCESS FAULT DETECTION – A PERFORMANCE AND SENSITIVITY STUDY

DOAN XUAN TIEN

NATIONAL UNIVERSITY OF SINGAPORE
2005
Moving PCA FOR PROCESS FAULT DETECTION – A PERFORMANCE AND SENSITIVITY STUDY

DOAN XUAN TIEN
(B.Eng.(Hons.), University of Sydney)

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2005
Acknowledgements

First of all, I would like to thank my supervisor, A/Prof. Lim Khiang Wee for his invaluable guidance, support and encouragement throughout my time here. He has given advices not only from an academic point of view but also from practical senses which I have not experienced. He has actively searched for better ways to support me and in the end he encourages and helps me planning the first step in my career. For all of that and much more, I would like to express my deepest gratitude to him.

I am also grateful to Dr. Liu Jun for his time and efforts in evaluating my progress and managing ICES–related issues. With his support, my life in ICES could not get more enjoyable.

I would also thank the Institute of Chemical and Engineering Sciences (ICES) for granting me the research scholarship and funds needed to pursue my Master degree.

It has been a wonderful experience for me in ICES and I look forward to continuing working here.

I would like to dedicate this to my parents, my sisters and brothers-in-law for their understandings and supports over these years.
Contents

1 Fault detection approaches – An overview 1

1.1 Fault detection – A definition ................................. 1

1.2 Why fault detection is critical ............................... 2

1.3 Current FDI approaches .................................... 4

1.3.1 Model-based FDI approaches ............................... 6

1.3.2 Process history–based FDI approaches ...................... 9

1.4 Principal Component Analysis (PCA) ........................ 13

1.4.1 Model development ....................................... 13

1.4.2 Number of Principal components (PCs) .................. 16

1.4.3 Conventional multivariate statistics ........................ 18

1.4.4 Performance criteria ....................................... 22

1.5 Thesis objectives ............................................. 23

2 PCA for monitoring processes with multiple operation modes 26
2.1 Motivation ......................................................... 26

2.2 Moving Principal Component Analysis .......................... 33
  2.2.1 Alternative scaling approach ............................... 33
  2.2.2 Practical issues ............................................. 37
  2.2.3 Detection rule ............................................... 38
  2.2.4 MPCA algorithm .............................................. 40

2.3 Algorithms for conventional PCA, APCA, and EWPCA ......... 42
  2.3.1 Conventional PCA ........................................... 42
  2.3.2 APCA .......................................................... 44
  2.3.3 EWPCA ....................................................... 45

2.4 A preliminary comparison between algorithms .................. 47

2.5 Simulation studies ................................................. 48
  2.5.1 Tennessee Eastman Process (TEP) .......................... 48
  2.5.2 Methodology ................................................ 50
  2.5.3 Results ..................................................... 51

2.6 Industrial case study .............................................. 55
  2.6.1 Process description ......................................... 55
  2.6.2 Results ..................................................... 57

2.7 Chapter conclusion ............................................... 61
3 Evaluation of MPCA Robustness

3.1 Introduction .......................................................... 63

3.2 Moving window size .................................................. 65

3.3 Number of principal components retained a ..................... 70

3.4 Confidence limit ...................................................... 77

3.5 Monitoring indices .................................................. 80

  3.5.1 Theory and implementation .................................... 80

  3.5.2 Comparative results ............................................. 84

3.6 Conclusion .......................................................... 87

4 Conclusion ............................................................ 88

A Process time constants ............................................. 96

  A.1 TEP ................................................................. 96

  A.2 Industrial case study ............................................... 96
Executive Summary

Process monitoring and fault detection is critical for economic, environmental as well as safety reasons. According to how a–priori knowledge of process is used, fault detection (and isolation) methods can be classified as process model–based or process history based or somewhere in between. Although the choice is often context–dependent, the use of process history based methods has become more popular due to the fact that massive databases of online process measurements are available for analysis.

This thesis evaluates the Principal Component Analysis approach (PCA), one of many process history–based methods for process monitoring and fault detection using operating data from an oil refinery and simulation data from a well–known research case study. Although successful applications of PCA have been extensively reported, it has the major limitation of being less effective with time–varying and/or non–stationary processes or processes with multiple operation modes. To address the limitation, this thesis proposes a *Moving Principal Component Analysis* (MPCA), which is based on the idea that updating scaling parameters (mean and standard deviation) from a moving window is adequate for handling the process variation between different operation modes. MPCA performance is compared with other published approaches including conventional PCA, adaptive PCA, and Exponentially Weighted PCA in monitoring Tennessee Eastman Process (TEP) simulation and analyzing an industrial data set. It is shown that the proposed MPCA method performs better than the other approaches when performance is measured by missed detection, false alarms, time delay and computational requirement.
Sensitivity of MPCA performance is also investigated empirically by varying critical parameters including moving window size, number of principal components retained, and confidence limits. The results indicate that MPCA method is not sensitive to those parameters in monitoring TEP process. Its performance does not change significantly with varying the size of moving window, number of principal components retained, or confidence limits. However, tuning of parameters is necessary for industrial application of MPCA. It has also been found that reasonable MPCA performance could be achieved using moving window size of 1 – 2 process time constant, 2 PCs, and 99% – 99.9% confidence limits. In addition, several monitoring indices including conventional statistics ($T^2$ and $Q$), combined $QT$ and standardized $Q$ index are also implemented in MPCA. It is shown that MPCA performance does not depend much on the form of the monitoring index being employed. All of the indices perform well although the standardized $Q$ statistic requires more computation time.
List of Figures

1.1 Transformations in a fault detection system ........................................... 4
1.2 Classification of FDI methods ................................................................. 5

2.1 Original operation data from a Singapore petrochemical plant. $\mathbf{X}_{16}$ and $\mathbf{X}_{08}$, correspond to two different periods of plant operation. The plant is in normal steady state in $\mathbf{X}_{16}$ but appears to experience some disturbance in $\mathbf{X}_{08}$................................................................. 27

2.2 Conventional PCA ($- T^2$ statistic) monitoring results: test data $\mathbf{X}_{08}$ is scaled against the mean and standard deviation of the training data $\mathbf{X}_{16}$ and subsequently analyzed by a PCA model derived from $\mathbf{X}_{16}$ ......................................................... 28

2.3 Conventional PCA ($- Q$ statistic) monitoring results: test data $\mathbf{X}_{08}$ is scaled against the mean and standard deviation of the training data $\mathbf{X}_{16}$ and subsequently analyzed by a PCA model derived from $\mathbf{X}_{16}$ ......................................................... 29

2.4 Monitoring by $T^2$ statistic for test data: $\mathbf{X}_{08}$ is initially scaled against its mean and standard deviation (ie. auto-scaled) and then analyzed by a PCA model derived from $\mathbf{X}_{16}$ ......................................................... 35
2.5 Monitoring by $Q$ statistic for test data: $X_{08}$ is initially scaled against its mean and standard deviation (ie. auto–scaled) and then analyzed by a PCA model derived from $X_{16}$. 

2.6 MPCA implementation .................................................. 40

2.7 MPCA schematic diagram ............................................... 41

2.8 Conventional PCA implementation ................................. 42

2.9 Schematic diagram for conventional PCA method ............... 43

2.10 Implementation of APCA method ................................. 45

2.11 APCA schematic diagram ............................................ 46

2.12 Tennessee Eastman Process ........................................... 49

2.13 Performance of four PCA methods in monitoring TEP–$T^2$ statistic. Simulated faults include $idv(1)$ (feed composition), $idv(4)$ (reactor cooling water inlet temperature) and $idv(8)$ (feed composition) at 3000–4000, 7000–8000, 10000–11000, respectively. ................................. 52

2.14 Performance of four PCA methods in monitoring TEP–$Q$ statistic. Simulated faults include $idv(1)$ (feed composition), $idv(4)$ (reactor cooling water inlet temperature) and $idv(8)$ (feed composition) at 3000–4000, 7000–8000, 10000–11000, respectively. ................................. 53

2.15 Process diagram for the industrial case study ..................... 56

2.16 Performance of four PCA methods in industrial case study – $T^2$ statistic .......... 58

2.17 Performance of four PCA methods in industrial case study – $Q$ statistic .... 59
3.1 False alarms for different moving window sizes – TEP simulation . . . . . 67
3.2 False alarms for different number of PCs retained – TEP simulation . . 71
3.3 Scree plot – TEP simulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
3.4 Scree plot – industrial case study . . . . . . . . . . . . . . . . . . . . . . . . 76
3.5 Algorithm for MPCA approach using QT statistic . . . . . . . . . . . . 82
A.1 TEP step response . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
A.2 Step response for the industrial case study . . . . . . . . . . . . . . . . . 98
# List of Tables

2.1 Cross-validation study of TEP simulation data .......................... 39  
2.2 Process disturbances ............................................................. 50  
2.3 Performance in TEP simulation using $T^2$ statistic ...................... 51  
2.4 Performance in TEP simulation using $Q$ statistic ....................... 54  
2.5 Performance in industrial case study – $T^2$ statistic .................... 57  
2.6 Performance in industrial case study – $Q$ statistic ....................... 57  
3.1 MPCA robustness to window size in TEP simulation–$T^2$ statistic .... 65  
3.2 MPCA robustness to window size in TEP simulation–$Q$ statistic .... 66  
3.3 MPCA robustness to window size in industrial case study ............... 69  
3.4 Sensitivity to number of PCs retained in TEP simulation–$T^2$ statistic . 72  
3.5 Sensitivity to number of PCs retained in TEP simulation–$Q$ statistic . 72  
3.6 Sensitivity to number of PCs retained in industrial case study–$T^2$ statistic 75  
3.7 Sensitivity to number of PCs retained in industrial case study–$Q$ statistic 75
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8</td>
<td>MPCA performance using different confidence limits – TEP simulation</td>
<td>78</td>
</tr>
<tr>
<td>3.9</td>
<td>MPCA performance using different confidence limits – industrial case</td>
<td>79</td>
</tr>
<tr>
<td>3.10</td>
<td>Parameter settings for MPCA using $QT$ index</td>
<td>81</td>
</tr>
<tr>
<td>3.11</td>
<td>Parameter settings for MPCA using Johan’s standardized $Q$ index</td>
<td>84</td>
</tr>
<tr>
<td>3.12</td>
<td>Comparative study of MPCA performance – TEP study</td>
<td>85</td>
</tr>
<tr>
<td>3.13</td>
<td>Comparative study of MPCA performance – industrial case study</td>
<td>86</td>
</tr>
</tbody>
</table>
Chapter 1

Fault detection approaches – An overview

1.1 Fault detection – A definition

Generally, fault detection is defined as the “determination of the faults present in a system and the time of detection” [14]. It is therefore to ascertain whether or not (and if so, when) a fault has occurred. A fault can be thought of as any change in a process that prevents it from operating in a proper pre-specified manner. Since performance of a process is usually characterized by a number of variables and parameters, a fault can also be defined to be any departure from an acceptable range of observed process variables and/or parameters. The term fault is generally used in synonym with failure which is of a physical/mechanical nature. More precisely, a failure is a catastrophic or complete breakdown of a component or function in a process that will definitely lead to a process fault even though a fault presence itself might not
indicate a component failure [37].

Other comprehensive definitions recognize that fault detection is more appropriate than change detection in describing the cause of performance degradation and that a fault can be either a failure in a physical component, or a change in process performance [37]. From a pattern recognition point of view, fault detection is in effect a binary classification: to classify a process data as either normal (conforming) or faulty (nonconforming). Consequently, fault detection is at the heart of a process monitoring system, which continuously determines the state of the process in real-time.

1.2 Why fault detection is critical

Any industrial process is liable to fault or failure. In all but the most trivial cases, the existence of a fault may lead to situations with human safety and health, financial, environmental and/or legal implications. The cost of poor product quality, schedule delays, equipment damage and others caused by process faults and failures was estimated to be approximately 20 billion USD for the US petrochemical industry alone every year [12]. It would be even higher when similar estimates for other industries such as pharmaceutical, specialty chemicals, power and so on, are accounted for. Similarly, the British economy incurred 27 billion USD annually due to poor management of process faults and failures [38]. Worst still, process upsets might contribute to chemical accidents which might in turn kill or injure people, and damage environment. Such accidents as Union Carbide’s Bhopal, India (1984) and
Occidental Petroleum’s Piper Alpha (1988) have not only lead to enormous financial liability but also resulted in tragic human loss.

Although proper design and operating practice might help to prevent process upsets from occurring, there are technical as well as human causes which make a monitoring system vital to effective and efficient process operation. Today, technology has not only made feasible highly complex and integrated processes operating at extreme conditions but also brought about an issue commonly referred to as “alarm flooding”. Ten of thousands of sensors are often monitored in a modern plant. Even in normal operation, 30 to 60 of these measurements may be in alarm per hour [26]. According to a survey undertaken in 1998 for the Health and Safety Executive, UK government, these figures were not untypical [3]. Given this “alarm flooding” issue and the complexity of process plants, it should come as no surprise that human operators tend to make erroneous decisions and take actions which make matters even worse. Industrial statistics show that human errors account for 70% of industrial accidents [38]. The 1994 explosion at Texaco’s Milford Haven refinery in south Wales is one of the well–published cases illustrating this. In the five hours before the explosion which cost £48 million and injured 26 people, two operators had to handle alarms triggered at an unmanageable rate of one alarm every 2 – 3 seconds [3]. The “alarm flooding” issue and the human error factor have raised the challenge to develop more effective methods for process monitoring and fault detection.
1.3 Current FDI approaches

In general, fault detection and isolation (FDI) tasks can be considered as a series of transformations or mappings on process measurements (see Fig. 1.1).

In Fig. 1.1 (reproduced from [38]), the measurement space is a space of finite number of measurements $\mathbf{x} = [x_1, x_2, \ldots, x_N]$, with no a priori problem knowledge relating these measurements. The feature space is a space of points $\mathbf{y} = [y_1, y_2, \ldots, y_M]$, where $y_i$ is the $i^{th}$ feature obtained as a function of the measurements utilizing a priori problem knowledge. The purpose of transforming the measurement space into

---

Since fault detection is the first stage in any FDI approach, it is more complete to review FDI approaches in general, rather than fault detection separately.
feature space is to improve performance or to reduce the complexity of the problem.
The mapping from the feature space to the decision space is usually designed to meet some objective function, such as minimizing the missed detections or false detections. In most cases, the decision space and the class space are one and the same, though in some other cases it is desired to maintain them as separate.

To explain these transformations more clearly, let consider Principal Component Analysis (PCA) method for fault detection problem. The dimension of the measurement space is the number of measurements available for analyzing. The transformation from the measurement space into the feature space, which is commonly referred to as score space in PCA terminology, is mathematically a linear transformation. It is accomplished by a vector–matrix multiplication between the measurements vector and the loading matrix $P$ (see Section 1.4), in which a priori process knowledge is embedded. The decision space could be seen as containing the statistical index chosen for monitoring purpose. The transformation from the feature space into the
decision space is a functional mapping and is very much dependent on the statistical index used. Lastly, the class space for fault detection has two values: 0 for normal and 1 for fault. A threshold function maps the decision space into the class space. Again, a priori process knowledge plays an important role here in determining the statistical threshold.

As seen, a priori process knowledge is the key component in any FDI approach. It affects two out of three transformations in Fig. 1.1. As a result, the type of a priori knowledge used is the most important distinguishing feature in FDI approaches [38]. A priori process knowledge which is developed from a fundamental understanding of the process using first–principles knowledge is referred to as deep, causal or model–based knowledge. On the other hand, it may be learned from past experience with the process and is referred to as shallow, compiled, evidential or process history–based knowledge. In addition, a priori process knowledge can also be classified as either quantitative or qualitative depending on whether it is described by quantitative or qualitative functions.

Based on the classification of a priori process knowledge, FDI approaches can be classified accordingly in Fig. 1.2 (reproduced from [38]).

1.3.1 Model–based FDI approaches

In general, a model is usually developed based on some fundamental understanding of the process. In that aspect, model–based FDI approaches can be broadly classified as quantitative or qualitative, depending on the type of model they make use.
Quantitative approaches

Quantitative model–based FDI approaches require two components: an explicit mathematical model of the process and some form of redundancy. There is a wide variety of quantitative model types that have been considered in FDI, and in all of them, the knowledge about the process physics is expressed in terms of mathematical functional relationships. They include first–principle models, frequency response models, input–output and state–space models. The first–principle models have not been very popular in fault diagnosis studies because of the difficulty in building these models and the computational complexity involved in utilizing them in real–time application. So far, the most important class of models that have been heavily investigated are the input–output or state–space models [38].

Once an explicit model of the monitored plant is available, all model–based FDI methods require two steps: generate inconsistencies (ie. residuals) between the actual and expected behavior of the plant and evaluate the inconsistencies to make a decision. In the first step, some form of redundancy is required. There are basically two types of redundancies including hardware redundancy and analytical redundancy. The former requires redundant sensors and its applicability is limited because of the extra cost and additional space required [38]. On the other hand, analytical redundancy, also referred to as functional, inherent or artificial redundancy is derived from the functional dependence among the process variables. In the second step, the generated inconsistencies are usually checked against some thresholds which might be derived from statistical tests such as generalized likelihood ratio test.
Extensive research over the past two decades has resulted in various FDI model–based techniques. The most frequently used include diagnostic observers, parity relations and Kalman filters. A detailed review of those techniques and relevant research is beyond the scope of this study. Interested readers are referred to the three–part review in [38]. It was also discussed in [38] that most of the research in quantitative model–based approaches have been in the aerospace, mechanical and electrical engineering literature. Model–based technique for chemical engineering has not received the same attention. This might be attributed to the unavailability/complexity of high fidelity models and the essential nonlinear nature of these models for chemical process. Several other factors such as high dimensionality, modelling uncertainty, parameter ambiguity could also limit the usefulness of the quantitative model–based approach in chemical industrial processes.

**Qualitative approaches**

Unlike quantitative approaches, the qualitative model–based ones require a model of the process in a qualitative form. In other words, the fundamental relationships between process variables are expressed in terms of qualitative functions. Depending on the form of model knowledge, qualitative approaches can be further classified as either qualitative causal models or abstraction hierarchies.

Qualitative causal model contains reasoning about the cause and effect relationships in the process. The most commonly used ones are digraphs, fault trees and qualitative physics, where the underlying relationships are represented graphically, logically, and in qualitative equations respectively.
Alternatively, in abstraction hierarchies, the process system is decomposed into its process units. The idea of decomposition is to be able to draw inferences about the overall process behavior solely from the laws that govern the behavior of its subsystems. There are two dimensions along which the decomposition can be done, which result in structural hierarchy and functional hierarchy. The former contains the connectivity information, while the later represents the means-end relationships between the process and its subsystems.

Qualitative model–based FDI approaches have a number of advantages as well as disadvantages. One of the major advantages is that qualitative models do not require exact, precise information about the process. Qualitative behaviors can be derived even if an accurate mathematical process model is not available. Furthermore, qualitative model–based methods can provide an explanation of the fault propagation through the process, which is indispensable when it comes to operator support in decision making [40]. However, the major disadvantage is the generation of spurious solutions resulting from the ambiguity in qualitative reasoning. Significant amount of research has been carried out to improve qualitative approaches. Interested readers are referred to [39] for extensive review and references.

1.3.2 Process history–based FDI approaches

In contrast to model–based approaches where some form of a process model is required, process history–based methods make use of historical process data. Based on feature extraction – the way in which the data is transformed into features and presented to the system – process history–based approaches can be viewed as quan-
Qualitative approaches

Two important methods that employ qualitative feature extraction are the expert system and trend modelling approaches.

The main components in an expert system development are knowledge acquisition, knowledge representation, the coding of knowledge in a knowledge base, inference machine, human–machine interface. When a fault arises, an expert system can reason out the causes of the fault (and propose correct methods to deal with it rapidly), based on a number of process measurement and the experience knowledge contained in the knowledge base. Using expert systems for diagnostic problem–solving has a number of advantages including ease of development, transparent reasoning, the ability to reason under uncertainty and the ability to provide explanations for the solutions provided [40].

Alternatively, qualitative trend modelling approaches to fault diagnosis can use a methodology based on a multi–scale extraction of process trends [30]. The monitoring and diagnostic methodology has three main components: the language used to represent the sensor trends, the method used for identifying the fundamental elements of the language from the sensor data and their use for performing fault diagnosis. Qualitative representation of the process trends has fundamental elements called primitives. Identification of primitives can be based on first and second derivatives of the process trend calculated using finite difference method or based on the use of an artificial neural network. However, the use of primitives from first– and second–order
trend requires numerous parameters (for shape comparison). In addition, qualitative trends alone might not be sufficient for monitoring process transitions because they do not contain time and magnitude information [1]. Enhanced trend analysis proposed in [1] uses only first-order primitives but incorporate additional information on the evolution and magnitude of process variables.

**Quantitative approaches**

Quantitative process history–based approaches can be further classified as either statistical or non–statistical. Artificial neural networks (ANN) are an important class of non–statistical approaches while principal component analysis (PCA)/projection to latent structure (PLS) are two of the most widely used statistical classifiers.

ANN has been utilized for pattern classification and function approximation problems. There are numerous studies reported where ANN is used for FDI (see [40]). The ability of ANN to construct nonlinear decision boundaries or mappings and accurately generalize the relationship learnt, in the presence of noisy or incomplete data, are very desirable qualities. Comparison between ANN and some conventional classification algorithm, such as Bayes’ rule and the nearest-neighbor rule, have shown that neural networks classify as well as the conventional methods. In general, ANN can be classified as either supervised learning or unsupervised learning depending on how it is trained.

In supervised learning strategies, synaptic connection weights are learned by explicitly utilizing the mismatch between the desired and actual values. Depending on the specific ANN topology, learning algorithms vary from universal approxima-
tors such as stochastic approximation (ie. back-propagation) curve fitting (ie. radial basis function) to method of structural risk minimization. The most popular supervised learning strategy has been the back-propagation algorithm. On the other hand, unsupervised learning ANN, also known as self organizing maps, have not been as effective in FDI. However, their ability to classify data autonomously is very interesting and useful when industrial processes are considered [25].

Statistical techniques such as PCA/PLS represent alternative approaches to FDI problem viewed from a quality control standpoint. Statistical Process Control (SPC) and subsequently Multivariate Statistical Process Control (MSPC) have been widely used in process systems for maintaining quality and recently in process monitoring and fault detection. Successful applications of MSPC techniques, PCA in particular have been extensively reported in the literature (see [40] and reference therein). PCA enables a reduction in the dimension of the plant data by the use of linear dependencies among the process variables. Process data are described adequately, in a simpler and more meaningful way in a reduced space defined by the first few principal components. Details of fundamental PCA technique are covered in the next section.

Despite successful applications, PCA is not a problem–free technique in the FDI field. One of the major limitations of PCA–based monitoring is that the PCA model is time–invariant, while most of real processes are time–varying to a certain degree [40]. Consequently, it might not work effectively with time–varying, non–stationary processes. In addition, because it is essentially a linear technique, its best applications are limited to steady state data with linear relationships between variables.
[24]. Other factors which might discourage the use of PCA in monitoring and fault
detection are related to data quality (characteristics of outliers/noise [8]); process
nature (batch/continuous) and practical issues such as selecting the monitoring in-
dex, number of principal components to retain etc.)

1.4 Principal Component Analysis (PCA)

1.4.1 Model development

Statisticians first used PCA over 100 years ago for multivariate analysis. The original
paper on PCA described the analysis, from a geometrical viewpoint, as determining
“lines and planes of closest fit to systems of points in space” [33]. Hotelling was the
first to use PCA in analyzing correlated data sets and since then, it has been used
extensively in analytical chemistry, social and economic sciences, and more recently
in chemical engineering.

PCA is a linear dimensionality reduction technique, optimal in terms of capturing
the variability of the data. It determines a set of orthogonal vectors, called loading
vectors, ordered by the amount of variance explained in the loading vector directions.
The new variables, often referred to as principal components are uncorrelated (with
each other) and are weighted, linear combinations of the original ones. The total
variance of the variables remains unchanged from before to after the transformation.
Rather, it is redistributed so that the most variance is explained in the first principal
component (PC), the next largest amount goes to the second PC and so on. In such
a redistribution of total variance, the least number of PCs is required to account for
the most variability of the data sets.

The development of PCA model, which can be found in numerous published literature including [21, 33] is summarized as follows. For a given data matrix $X^o$ (raw data), which has $n$ samples and $m$ process variables as in (1.1), each row $X_i^T$ is a sample of $m$ variables associated with a given time. The training data matrix contains values of process variables during “good”, “normal” operation.

$$
X^o = \begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1m} \\
x_{21} & x_{22} & \cdots & x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \cdots & x_{nm}
\end{pmatrix}
$$

(1.1)

Where: $x_{ij}$ is the data value for the $j^{th}$ variable at the $i^{th}$ sample.

Initially, some scaling is usually required for the training data set. The most common approach is to scale the data using its mean and standard deviation

$$
X = (X^o - 1_n\mu^T)\Sigma^{-1}
$$

(1.2)

Where: $X^o$ is a $n \times m$ data set of $m$ process variables and $n$ samples.

$\mu$ is the $m \times 1$ mean vector of the dataset.

$1_n = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$.

$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_m)$ whose $i^{th}$ element is standard deviation of the $i^{th}$ variable.

After appropriate scaling, the training data can used to determine loading vectors by solving the stationary points (where the first derivative is zero) of the optimization problem:

$$
\max_{\nu \neq 0} \frac{\nu^T X^T X \nu}{\nu^T \nu}
$$

(1.3)
However, the stationary points are better computed via the singular value decomposition (SVD) of the data matrix

\[
\frac{1}{\sqrt{n-1}}X = U\Sigma V^T
\]  

(1.4)

Where: \( U \in R^{n \times n} \) and \( V \in R^{m \times m} \) are unitary matrices.

\( \Sigma \in R^{n \times m} \) is diagonal matrix.

The matrix \( \Sigma \) contains the nonnegative real singular values of decreasing magnitude along its main diagonal \( (\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\text{min}(m,n)}) \), and zero off-diagonal elements. Column vectors in the matrix \( V \) are the loading vectors. Upon retaining the first \( a \) singular values, the loading matrix \( P \in R^{m \times a} \) is obtained by selecting the corresponding loading vectors.

The projections of the observations in \( X \) into the lower dimensional space are contained in the score matrix

\[
T = XP
\]  

(1.5)

and the projection \( \hat{X} \) of \( T \) back into the \( m \)-dimensional observation space

\[
\hat{X} = TP^T
\]  

(1.6)

The residual matrix \( E \) is the difference between \( X \) and \( \hat{X} \)

\[
E = X - \hat{X}
\]  

(1.7)

The residual matrix \( E \) contains that part of the data not explained by the PCA model with \( a \) principal components and usually associated “noise”, the uncontrolled process and/or instrument variation arising from random influences. The removal of this data from \( X \) can produce a more accurate representation of the process, \( \hat{X} \) [21].
1.4.2 Number of Principal components (PCs)

As the portion of the PCA space corresponding to the larger singular values describes most of the *systematic* or *state* variations occurring in the process, and the random noise is largely contained in the portion corresponding to the smaller singular values, appropriately determining the number of principal components, $a$, to retain in the PCA model can decouple the two portions and enable separate monitoring of the two types of variations [21]. Retention of too many PCs might incorporate process noise unnecessarily and lead to slow and ineffective fault detection, especially for faults with smaller magnitude. On the other hand, too few PCs could result in a greater frequency of false alarms as the important process variation might not be fully accounted for by the PCA model [11].

Several techniques exist for selecting the optimal number of principal components to retain in a PCA model including: the percent variance test, the scree test and cross validation technique.

The percent variance method is based on the fact that each of the PCs is representative of a portion of the process variance, measured by the square of its corresponding singular value. The method determines the optimal value $a$ by choosing the smallest number of loading vectors needed to explain a specific minimum percentage of the total variance. Its popularity lies in the fact that it is easy to understand and automate for online applications [7]. However, the method is not recommended because it suffers from a disadvantage that the inherent variability of a chemical process is generally unknown and hence unaccounted for. A decision based solely on an arbitrarily chosen minimum percentage variance is unlikely to yield the optimal number

The scree test was developed by Cattell who observed that plots of the eigenvalues of the covariance matrix versus their respective component number had a characteristic shape [11]. The eigenvalues tend to drop off quickly at first, decreasing to a break in the curve. The remaining eigenvalues, which are assumed to correspond to the random noise, forms a linear profile. The number of principal components to retain is determined by identifying the break in the scree plot. Although this method has become quite popular, there can be a few problems with it. Particularly, identification of the break in scree plots can be ambiguous [21] as they might have no break or multiple breaks [7]. Consequently, this method can not be recommended, especially in automatic online applications.

Cross validation technique starts with zero principal components to be retained. Then, for each additional PC, it evaluates a prediction sum of squares (also known as PRESS statistic). As PRESS statistic for a data set is computed based on increasing dimensions of the score space using other data sets, the statistic is a measure of the predictive power of the model. When the PRESS is not significantly reduced compared to the residual sum of squares (RSS) of the previous dimension, the additional PC is considered unnecessary and the model building is stopped [33]. Intuitively, cross validation technique requires much more data and computational resource and hence might not be suitable for online implementation.

In short, although the techniques just described are used commonly, they all have some disadvantages in theoretical basis (percent variance method) or in online implementation (scree plot, cross validation). As a result, this study takes an empirical
approach where the number of PCs is increased from 1 until satisfactory performance
of PCA model in process monitoring and fault detection is obtained. (Performance
comparison in Section 3.3 indicates the superiority of empirical approach over the
percent variance method and the scree plot method.)

1.4.3 Conventional multivariate statistics

Once a PCA model based on normal, “in–control” performance is obtained, upon
new data becoming available, several multivariate statistics can be used to monitor
and detect faults. The conventional ones include Hotelling’s $T^2$ statistic and squared
prediction error (SPE) statistic (also known as $Q$ statistic). In this section, these
statistical monitoring indices are briefly reviewed.

**Hotelling’s $T^2$ statistic**

$T^2$ statistic, introduced by and named after Hotelling in 1947, is a scaled squared
2–norm of an observation vector $x$ (from its mean). The scaling on $x$ is in the
direction of the eigenvectors and is inversely proportional to the standard deviation
along those directions ie. the Mahalanobis distance.

$$z = \Lambda^{-\frac{1}{2}} V^T x$$

$$T^2 = z^T z = x^T V (\Sigma^T \Sigma)^{-1} V^T x$$

By retaining only the first $a$ largest singular values and substituting $P$ the loading
matrix for $V$, the $T^2$ statistics for the PC subspace is:

$$T^2 = x^T P \Sigma_a^{-2} P^T x$$
To determine whether or not a fault has occurred, appropriate thresholds for the $T^2$ statistic based on the level of significance $\alpha$, are required. These control limits can be evaluated by assuming the projection of measurement $\mathbf{x}$, is randomly sampled from a multivariate normal distribution. If it is assumed additionally that the sample mean vector and covariance matrix for normal/ “in–control” operations are equal to the actual population counterparts, then the $T^2$ statistic follows a $\chi^2$ distribution with $a$ degrees of freedom

$$ T^2_{\alpha} = \chi^2(a) $$

(1.11)

Where: $\alpha$ is the level of significance.

$\chi^2(a)$ is $\chi^2$ distribution with $a$ degrees of freedom.

However, most of the time, the actual mean and covariance matrix are estimated by the sample counterparts. The $T^2$ statistic threshold in these cases is:

$$ T^2_{\alpha} = \frac{a(n - 1)(n + 1)}{n(n - a)} F_{\alpha}(a, n - a) $$

(1.12)

Where: $F_{\alpha}(a, n - a)$ is the upper 100$\alpha$% critical point of the $F$–distribution with $a$ and $n - a$ degrees of freedom.

If the number of data points $n$ is so large that the mean and covariance matrix estimated from data are accurate enough, the two thresholds above approach each other. Even though the control limits for $T^2$ statistic are derived assuming that the observations are statistically independent and identically distributed, provided that there are enough data in the training set to capture the normal process variations, $T^2$ statistic can perform effectively in process monitoring even if mild deviations from those assumptions exist [21].
In conclusion, given a level of significance $\alpha$, the process operation is considered normal/“in-control” if $T^2 \leq T^2_\alpha$, which is an elliptical confidence region in the PCA space.

**Squared Prediction Error (SPE) – $Q$ statistic**

$Q$ statistic also known as squared prediction error ($SPE$) is mathematically the total sum of residual prediction errors.

$$Q = e^T e$$  \hspace{1cm} (1.13)

Where: $e = (I - PP^T)x$ is the row vector in the residual matrix $E$ (see Equation 1.7).

The upper control limit for $Q$ statistic with a significance level $\alpha$ was developed by Jackson and Mudholkar [6]

$$Q_\alpha = \theta_1 \left[ \frac{h_0 c_\alpha \sqrt{2 \theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0}$$  \hspace{1cm} (1.14)

Where:

$$\theta_i = \sum_{j=a+1}^{m} \sigma_j^{2i}$$  \hspace{1cm} (1.15)

$$h_0 = 1 - \frac{2 \theta_1 \theta_3}{3 \theta_2^2}$$  \hspace{1cm} (1.16)

$c_\alpha$ is the normal deviate corresponding to the $(1 - \alpha)$ percentile.

$\sigma_j^2$ is the $j^{th}$ eigenvalue.

Alternatively, the upper control limit for $Q$ statistic can be determined as follows according to [28]
\[ Q_\alpha = g \chi^2_{k,\alpha} \] (1.17)

Where:

\[ g = \frac{\theta_2}{\theta_1} \] (1.18)
\[ h = \frac{\theta_1}{\theta_2} \] (1.19)

All of these control limits for \( Q \) statistic were derived based on assumptions that the residual vector \( \mathbf{e} \) follows a multivariate normal distribution and \( \theta_1 \) is very large \([2, 16]\)

**\( T^2 \) or \( Q \) statistics**

Although both \( T^2 \) and \( Q \) statistics are used in industrial applications \([22]\), it is necessary to point out that they actually measure different situations of the process, and hence they detect different types of faults.

The \( Q \) statistic is a measure of deviation from the PCA model in which normal process correlation is embedded. Provided that the PCA model is valid, exceeding the control limit for the \( Q \) index indicates that the normal correlation is broken and hence it is very likely that a fault has occurred. On the other hand, the \( T^2 \) index measures the distance to the origin in the PC subspace. In other words, it is a measure of how far the current observation is away from the mean of the training set which captures the normal process variations. If the \( T^2 \) threshold is exceeded, it could be due to a fault but it might very well be due to a change in the operating region which is not necessarily a fault.
Furthermore, as the PC subspace typically contains normal process variations with large variance and the residual subspace contains mainly noise, the normal region defined by the $T^2$ threshold is usually much larger than that defined by the $Q$ threshold. As a result, it usually takes a much larger fault magnitude to exceed the control limit for $T^2$ statistic [16].

As $T^2$ and $Q$ statistics along with their appropriate thresholds detect different types of faults, the advantages of both monitoring indices can be fully utilized by employing the two measures together [21].

### 1.4.4 Performance criteria

In order to compare various fault detection methods, it is useful to identify a set of desirable criteria based on which performance of a fault detection system can be evaluated. A common set of such criteria or standards for any fault detection approach includes detection errors, timely detection, and computational requirements.

The first criterion is the classification error in fault detection. This includes missed detection rate and false alarm rate. The former refers to the number of actual faults that occurred but are not detected while the later is the number of normal, in–controlled data samples that are declared as faults by the monitoring approach.

The second criterion is time delay in fault detection. The monitoring system should respond quickly in detecting process malfunctions. The less time a method takes to detect a fault, the better it is. However, there is a tradeoff between timely detection and sensitivity of the method. A monitoring method that is designed to respond
quickly to a failure will be sensitive to high frequency influences. This makes the
method likely to be vulnerable to noise and lead to frequent false alarms during
normal operation.

Last but not least, storage and computational requirements also plays an important
role in evaluating the performance of a fault detection method, especially in an
online context. Usually, quick real-time fault detection would require algorithms
and implementations which are computationally less complex, but might impose a
high storage requirements. It is therefore desirable to employ a method that offers
a reasonable balance between online (real-time) computational requirement versus
storage/data requirement.

1.5 Thesis objectives

Given all the available techniques for fault detection, the question of which one
should be used does not have a trivial answer, and is often very much context–
dependent. However, the use of process history–based techniques has become more
and more popular for a number of reasons. One reason is that it may be difficult,
time–consuming, tedious and even expensive to develop a first–principle model of
the process accurate enough to be used for process monitoring and fault detection
[4]. Even when such a process model can be obtained, its validity over a range of op-
erating conditions is questionable due to the unavoidable estimation of certain model
parameters. Secondly, the popularity of process history–based approaches has been
supported by an ever–increasing availability of computer control and new sensors,
installed and used in process monitoring (data acquisition) system, thus creating massive databases of process measurements, which require efficient analytical methods for their interpretation. [19].

This thesis studies PCA techniques in process monitoring and fault detection. As mentioned previously, PCA might not perform well with time–varying and/or non–stationary processes or continuous process with multiple operation modes. Various modifications have been proposed to improve its performance. This work explores an alternative scaling approach and studies the performance of a new *Moving Principal Component Analysis* (MPCA) approach in dealing with process variation between different process operation modes.

The thesis is organized as follows. Chapter 1 serves as an introduction to the context of process monitoring and fault detection. It explains what and why fault detection is necessary and then gives an overview of the current FDI approaches. It then describes fundamentals of PCA technique including model development, selecting the number of principal components (PCs), conventional Hotelling’s $T^2$ and $Q$ statistics, performance criteria. Chapter 1 ends with an outline of the thesis.

Chapter 2 proposes a new *Moving Principal Component Analysis* (MPCA) approach, and compares its performance with other approaches for monitoring processes with multiple operation modes. The chapter initially describes the limitation of conventional PCA technique in dealing with time–varying, non–stationary processes and briefly review modifications which have been published in literature. A new MPCA approach is proposed for monitoring processes with multiple operation modes, which are locally time–invariant and stationary. Implementation of the newly proposed
MPCA approach as well as other PCA–based methods including conventional PCA, adaptive PCA (APCA) and exponentially weighted PCA (EWPCA) are carried out to evaluate their performance both in a single–mode TEP simulation and in analyzing data sets from different operation modes of an industrial process. Chapter 2 concludes that based on the criteria set out previously, MPCA performs better than the other methods in both of the contexts.

In Chapter 3, the sensitivity of the proposed MPCA approach is studied empirically. The parameters subjected to study include moving window size, number of PCs retained, and confidence limits. In addition, Chapter 3 also implements a number of monitoring indices including conventional Hotelling’s $T^2$ and $Q$ statistics, modified $Q$ statistic and combined $QT$ index in order to search for the optimal index to be used with MPCA monitoring. Finally, a conclusion and recommendations for further work are presented in Chapter 4.
Consider the use of conventional PCA to analyze operation data from an industrial process. Analysis is carried out on data sets extracted from an operational database of a Singapore petrochemical plant. Although the training and test data sets are in a chronological order, they are from two separate operation intervals. The data sets are shown in Figure 2.1 (their description are presented shortly). A PCA–based model is built using the training data set, retaining two principal components. The test data set is scaled using the mean and standard deviation of the training set as in Equation (2.1). $T^2$ and $Q$ statistics with 99% and 99.9% respectively are used to analyze the test set for potential process disturbances. The results are shown in Figures 2.2 and 2.3.
Figure 2.1: Original operation data from a Singapore petrochemical plant. $\mathbf{X}_{16}$ and $\mathbf{X}_{08}$, correspond to two different periods of plant operation. The plant is in normal steady state in $\mathbf{X}_{16}$ but appears to experience some disturbance in $\mathbf{X}_{08}$. 
Figure 2.2: Conventional PCA ($T^2$ statistic) monitoring results: test data $X_{08}$ is scaled against the mean and standard deviation of the training data $X_{16}$ and subsequently analyzed by a PCA model derived from $X_{16}$. 
Figure 2.3: Conventional PCA (− Q statistic) monitoring results: test data \( \mathbf{X}_{08} \) is scaled against the mean and standard deviation of the training data \( \mathbf{X}_{16} \) and subsequently analyzed by a PCA model derived from \( \mathbf{X}_{16} \).
Figure 2.1 shows two things. Firstly, the training and testing data sets are from two different operation modes, indicated by different values of process variables at steady state. Secondly, although disturbance occurs in the testing data set, a significant part of the data set indicates normal/in–control process operation. However, as seen in Figures 2.2 and 2.3, two statistical indices $T^2$ and $Q$ always exceed their corresponding control limits. In other words, the results indicate that the testing set appears completely faulty, which is obviously not reliable.

This illustrates a major limitation of using conventional PCA for process monitoring: that conventional PCA–based model is time–invariant. What is captured in such a PCA model is only an instant of the process dynamics and its statistical characteristics, being conveyed by the training data set. However, most industrial processes are time–varying and/or non–stationary. Their dynamics and/or statistical characteristics changes with time and hence their monitoring requires an adaptive rather than a fixed model [42].

By definition, a process is time–invariant if its output response to a fixed input does not depend on time [23]. Hence, continuous chemical processes which frequently undergo operation mode changes, are not strictly time–invariant. The time–varying behaviors come in when transition between operation modes occurs. In addition, at steady state, the statistical characteristics such as process variable mean and variance vary from one mode to another although the process dynamics may be the same. As a result, a conventional PCA–based model for monitoring a continuous process with multiple operation modes would produce an excessive number of false alarms (as in Figures 2.2 and 2.3) or alternatively, missed detection of process faults.
In either case, reliability of the monitoring system is significantly compromised [41].

The problem of time–invariant PCA–based monitoring has been recognized in the literature [40, 41, 42]. Feasible solutions for continuous processes with multiple operation modes are available in basically three different classes [5]:

1. develop one PCA model for all the different operation modes

2. adaptively update the PCA model to account for mode changes

3. develop a local monitoring model for each mode

Approaches in the first class try to accommodate the mode–to–mode variation and mode transition using one overall PCA model. A conventional PCA model could become too conservative, and consequently leads to erroneous monitoring (see Figures 2.2 and 2.3). Hwang and Han [5] proposed a two–stage approach where different operating modes are initially identified using hierarchical clustering then a super–PCA model is built for monitoring processes with multi–mode operation. Their assumption is that the different operating modes do not introduce significant nonlinearities in the process behavior which is not true for modes corresponding to different feed grades, product slates, or throughput [1].

The second class of approaches has so far been explored the most extensively and is also applicable to time–varying processes. Wold [35] presented the use of exponentially weighted moving average (EWMA) filters in conjunction with PCA and PLS to dynamically update the PCA/PLS models. Li et al., [41] used algorithms utilizing rank–one modification and Lauczos tridiagonalization for recursive PCA and considered other essential issues including recursive update of scaling factors,
number of principal components retained, and confidence limits. Lane [34] applied exponentially weighted PCA (EWPCA) to monitoring of a polymer film manufacturing process. Other applications was demonstrated on adaptive monitoring of a rapid thermal annealing process [40]. For comparison, two representative PCA–based methods from this class are studied. They are adaptive PCA and exponentially weighted PCA.

In the last class, the general strategy is to build a local PCA model for monitoring each operating mode. Intuitively, approaches in this class would give better performance as local PCA models offers higher monitoring resolution. However, they do suffer from a number of disadvantages. The most serious one would be that it is usually difficult to identify effectively when mode transitions in an industrial process occur. Inaccurate mode transition identification could result in the process characteristics being miss–represented and consequently lead to poor monitoring performance [5]. This issue of identification and monitoring of mode–to–mode transitions was studied extensively in [1, 31, 32]. In addition, there might not be sufficient training data from each operating mode to capture its true covariance structure for local PCA model development, which results in unsatisfactory performance. Furthermore, these approaches are considered as being time consuming and not cost–effective and require more computational and storage requirements, as switching between operation modes could be frequent in industrial processes. This study deals with those disadvantages by demonstrating that one local PCA model could be used effectively and efficiently in monitoring different operating modes if proper scaling is employed. The proposed scaling is based on a moving window and hence the method is referred
The rest of Chapter 2 is organized as follows. Section 2.2 introduces the proposed MPCA approach using alternative scaling based on a moving window. Section 2.3 describes general frameworks for carrying out different PCA implementations including conventional PCA, adaptive PCA (APCA) and exponentially weighted PCA (EWPCA) which are published in literature. TEP simulation and an industrial case study are presented in Sections 2.5 and 2.6 respectively. In both of these sections, the performance of MPCA is analyzed and compared with that of the other methods discussed previously. Chapter 2 ends with a concluding section 2.7.

2.2 Moving Principal Component Analysis

2.2.1 Alternative scaling approach

The first step in applying PCA is to preprocess the data matrix \( X \). PCA is scale dependent i.e. critically depends on the units of measurement used for each process variable. If the differences between the measurement units of variables are very large, the first few principal components from PCA analysis will be dominated by those variables with larger magnitudes. As a result, different scaling approaches can produce different results and care needs to be taken in choosing an appropriate scaling method [17].

As mentioned previously, the most common scaling method is to normalize the data using the mean and standard deviation of the training set which is used to built the
PCA model

\[ \tilde{x} = \frac{x - \mu}{\sigma} \]  \hspace{1cm} (2.1)

Where: \( \mu \) is the mean of the training data set.

\( \sigma \) is standard deviation of the training data set.

However, it intuitively and verifiably (as seen in Figures 2.2 and 2.3) does not work well for data that comes from other operating conditions than the training set.

In addition, it has been observed that

1. for continuous process with multiple operation modes, the mode-to-mode variation seems not to introduce significant changes in process dynamics at steady state. In other words, process covariance structures for different mode share common characteristics [5], which could be effectively captured by one PCA model.

2. mode-to-mode variation is reflected in the changes in process variable mean and standard deviation.

To demonstrate the two observations, consider the earlier example which is shown in Figures 2.2 and 2.3. Instead of scaling the test data set by the mean and standard deviation of the training data set, the corresponding parameters of the whole test set are used. In other words, the test data is scaled using the mean and standard deviation of the test data set. The results are shown in Figures 2.4 and 2.5.

Clearly, the results appear much more reliable than previously with scaling parameters from the training data set. Hence, the changes due to switching between
Figure 2.4: Monitoring by $T^2$ statistic for test data: $\mathbf{X08}$ is initially scaled against its mean and standard deviation (ie. auto–scaled) and then analyzed by a PCA model derived from $\mathbf{X16}$. 
Figure 2.5: Monitoring by $Q$ statistic for test data: $X_{08}$ is initially scaled against its mean and standard deviation (i.e. auto-scaled) and then analyzed by a PCA model derived from $X_{16}$. 
operation modes are likely to be captured by adapting the scaling parameters, while other process dynamics properties can be described by one fixed PCA model.

Based on the observations, it would be recommended that the mean and standard deviation of the test set itself should be used in scaling of the test sample. However, in real–time monitoring, these parameters are unknown. As a result, it is proposed here that the corresponding parameters for a recent moving window of data could be used instead.

The proposed approach maintains a moving window in real–time (hence it is termed moving PCA) for updating the scaling parameters, which could account for the statistical variation between operation modes. As the mode–to–mode variation is dealt with, one conventional PCA model would be sufficient and effective in monitoring of continuous process with multiple operation modes.

2.2.2 Practical issues

It is noted that the proposed MPCA approach comes with new challenges. To achieve acceptable performance using this method, a number of practical issues must be addressed. Questions such as how moving window should be updated or whether MPCA works best using a particular monitoring index and if so, what the optimal index is among numerous statistical indices that have been proposed in literature. Parameters including the length of the moving window, number of principal components retained, significance levels and weighting factors used must be specified carefully. Empirical study as well as a–priori knowledge of the process at hand would help to form a basis in choosing the length of the moving window so as to it
can be a representative description of the process dynamics (while being robust). In
addition, weighting factors for data samples are usually taken to be one i.e., all data
(new or old) in the set have the same influence even though in certain cases, using
different weighting scheme could improve or degrade the monitoring performance.

2.2.3 Detection rule

A detection rule is devised in order to deal with excessive number of redundant
alarms and to reduce the number of false alarms (due to process noise). (A similar
rule was presented in [21]). It is based on the reasoning that from the time a fault
occurs to the time it is corrected, the process measurements are likely to exceed the
upper control limits. As all of those alarms correspond to the same fault, they are,
except for the very first one, considered redundant. In addition, measurements from
chemical processes are constantly polluted with random noise, which might make it
to exceed the upper control limit. Normally, this would constitute a (false) alarm.
Due to the random nature of the process noise, and given that there is no process
fault present, the next measurement is unlikely to stay above the upper control limit
i.e., it must go back to the normal region. Therefore, a fault is likely to present if
there are two consecutive warnings. It is even more likely to occur if there are three
consecutive warnings.

Continuing on this argument, a question must be asked: how many consecutive
warnings would be required before a fault is announced. To answer this question, a
cross-validation study on Tennessee Eastman Process (TEP) simulation (cf. Section
2.5 is carried out. In this study, TEP simulation is run in a normal base case i.e.,
without any process faults/disturbances. The 2401 data samples are recorded and used to build a PCA model, which is in turn used to test its training data. \( T^2 \) statistic is used with 99% confidence limit. The following results are presented in Table 2.1.

Table 2.1: Cross-validation study of TEP simulation data

<table>
<thead>
<tr>
<th>No. of consecutive warnings used</th>
<th>No. of false alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

As Table 2.1 shows, when one warning implies one alarm, there are 15 false alarms. However, if 4 consecutive warnings indicate one alarm, then there is only 1 false alarm. Consequently, the following detection rule is proposed:

**Warning** when the monitoring index exceeds its corresponding control limit

**Fault** detected when there are at least *four* consecutive warnings.

This detection rule is applied for MPCA as well as for other PCA-based methods. Even though certain faults/outliers which occur for period of less than four samples, might be missed, those faults/outliers are deemed insignificant in overall process performance. In addition, detection delay is also increased (by 4 samples) due to the rule. Nevertheless, the great reduction in false alarms definitely offsets the two shortcomings of the proposed detection rule.
2.2.4 MPCA algorithm

*Moving PCA* (MPCA) attempts to deal with mode–to–mode process variation by updating the scaling parameters each time a new process measurement becomes available. It does that by moving and updating a data window in real–time. New test samples when available is scaled against the mean and standard deviation of that moving data window. MPCA schematic diagram is shown in Figure 2.7. Its algorithm follows (see Figure 2.6).

Figure 2.6: MPCA implementation

1. Acquire training data which represents normal process operations. Auto–scale the training data using its mean and standard deviation. Then carry out sin-
singular value decomposition (SVD) operation to obtain a PCA model for the process. Two principal components are retained (refer to 3.3). Determine the control limits for $T^2_{stat}$ and $Q_{stat}$. Levels of significance are set at 99% and 99.9% respectively as suggested in [2].

2. Obtain the next test sample $x$. Scale $x$ using scaling factors from the moving data window. For initialization of the moving window, corresponding values for training data set could be used.

3. Evaluate the monitoring indices ($T^2_{stat}$ and $Q_{stat}$) for the scaled test sample using the PCA model obtained in Step 1. Check with the corresponding control limits. If not exceeded, the sample $x$ is considered normal.

4. If $x$ is normal, update the moving window by deleting the oldest data sample and concatenating the new one $x$ to the moving window. Repeat from Step 2.
2.3 Algorithms for conventional PCA, APCA, and EWPCA

The performance of three PCA-based approaches including conventional PCA, Adaptive PCA (APCA), and Exponentially Weighted PCA (EWPCA) are compared to that of the proposed MPCA method. The procedures to implement those methods are described in the following.

2.3.1 Conventional PCA

Figure 2.8: Conventional PCA implementation
The algorithm and schematic diagram for conventional PCA are shown in Figures 2.8 and 2.9 respectively. Its implementation consists of the following steps

1. Acquire training data which represents normal process operations. Scale the training data using its mean and standard deviation. Then carry out singular value decomposition (SVD) to obtain a PCA model for the process. Two principal components are retained. Determine the upper control limits for $T^2_{\text{stat}}$ and $Q_{\text{stat}}$. Levels of significance are set at 99% for $T^2_{\text{stat}}$ and 99.9% for $Q_{\text{stat}}$.

2. Obtain the next test sample $\mathbf{x}$. Scale $\mathbf{x}$ using scaling factors from Step 1.

3. Evaluate the monitoring indices $T^2_{\text{stat}}$ and $Q_{\text{stat}}$ for the scaled test sample using the PCA model obtained in Step 1. Check with the corresponding control limits. If not exceeded, the sample $\mathbf{x}$ is considered normal.
4. Repeat from Step 2

2.3.2 APCA

Unlike MPCA method which updates only the scaling factors, APCA updates the model to deal with time-varying, non-stationary characteristics. In APCA approach, the PCA model is updated each time a new sample is available. In doing so, it attempts to capture the most recent normal data variation. Algorithms for online adapting PCA model have been studied in [41] although there is a little computational difference between the algorithm described there and the one that is implemented in this study. Li et al. [41] presented the adaptive PCA in a recursive way while the approach here is to re-evaluate the PCA model for a moving window of fixed length each time the window itself is updated with a new data sample. Algorithm for APCA approach is described following (see Figure 2.10), while its schematic diagram is shown in Figure 2.11

1. Acquire training data, which represents normal process operations for initializing the moving window.

2. Scale the moving window data using its mean and standard deviation. Then carry out SVD to obtain a PCA model. Two principal components are retained. Determine the control limits for $T^2_{stat}$ and $Q_{stat}$. Levels of significance are set at 99% and 99.9%, respectively.

3. Obtain the next test sample $\mathbf{x}$. Scale $\mathbf{x}$ using scaling factors from Step 2. Evaluate the monitoring indices ($T^2_{stat}$ and $Q_{stat}$) for the scaled test sample using
the current PCA model. Check the monitoring indices with the corresponding control limits. If the control limits are not exceeded, the sample $x$ is considered normal.

4. if $x$ is normal, update the moving window by deleting the oldest data sample and concatenating the new one $x$ to the moving window. Repeat from Step 2

Figure 2.10: Implementation of APCA method

### 2.3.3 EWPCA

Similar to APCA approach, the EWPCA updates the PCA model as a new sample becomes available. However, as implied by its name, the latter approach implements
an exponentially weighting scheme to put more emphasis on more recent data. A number of studies in EWPCA have been published in literature including [13, 34, 35, 41]. The following procedure is adopted from [34].

1. **Initializing** Acquire training data, which represents normal process operations. Scale the data using its mean $\mu_0$ and standard deviation $\sigma_0$. Initialize all other parameters as follows

   \[
   R_{t=0} = \frac{(X^TX)_{t=0}}{n_0 - 1} \quad (2.2)
   \]

   \[
   \lambda_0 = \left(1 - \frac{1}{n_0}\right) \quad (2.3)
   \]

   Where: $R_{t=0}$ is the correlation matrix. $\lambda_0$ is the initial weighting factor. $n_0$ is the initial sample size

2. **Testing** Obtain the next testing sample $x$ and then scale it. Update the cor-

![Figure 2.11: APCA schematic diagram](image-url)
relation matrix
\[ R_t = \lambda_{t-1} R_{t-1} + (1 - \lambda_{t-1})(xx^T) \]  
(2.4)

Update the PCA model based on the new correlation matrix \( R \), then evaluate \( T^2 \) stat and \( Q \) stat for the scaled test sample \( x \). Check the monitoring index \( T^2 \) stat with its corresponding upper limit.

3. Updating If the sample \( x \) is considered normal, the data matrix \( X \) is updated as follows
\[ X_t = \lambda_{t-1}[X_{t-1}|x^T] \]  
(2.5)

For other parameters,
\[
\mu_t = \lambda_{t-1}\mu_{t-1} + (1 - \lambda_{t-1})x \\
(2.6)
\]
\[
\sigma_t(i) = \{\lambda_{t-1}[\sigma_{t-1}^2(i) + (\mu_t(i) - \mu_{t-1}(i))^2] + (1 - \lambda_{t-1})\|x - \mu\|^2\}^{0.5} \\
(2.7)
\]
\[
\lambda_t = 1 - \frac{(1 - \frac{T^2_m}{m})SPE_t}{\sqrt{n_t - 1}} \\
n_t = \lambda_{t-1}n_{t-1} + 1 \\
(2.8)(2.9)
\]

Where: \( T^2_t \) and \( SPE_t \) are respectively the values of \( T^2 \) and \( Q \) statistics for sample \( x \).

Return to Step 2 and repeat the iteration for the next online sample.

2.4 A preliminary comparison between algorithms

In all schematic diagrams (Figures 2.7, 2.9, and 2.11), any red box indicates that the values inside the box are processed online. As can be seen from those figures,
APCA (and similarly EWPCA) requires the most online calculation while conventional PCA requires the least. The major difference between MPCA and APCA (and EWPCA) is that the former approach evaluates and stores its training data and PCA model (except for the scaling parameters) off-line (ie. only once) while the latter processes its data and evaluates PCA model completely online (eg. once every new sample). In other words, in terms of modelling, the only calculation that MPCA has to do online is evaluating scaling parameters from a moving window. On the other hand, APCA (and EWPCA) must additionally re-evaluate its PCA model (ie. principal component vectors, control limits etc.). This makes MPCA simpler and more attractive as much less online computation and storage are necessary.

2.5 Simulation studies

2.5.1 Tennessee Eastman Process (TEP)

TEP (see Figure 2.12, reproduced from [15]) was developed by Downs and Vogel of the Eastman Company to provide a realistic simulation for evaluating process control and monitoring methods [15]. It has become perhaps the most important and commonly used benchmark simulation for the development of plant-wide control [29]. There are five major units in TEP simulation: a reactor, a separator, a stripper, a condenser, and a compressor. The process here consists of 12 manipulated variables from the controller and 41 measurements, which all have Gaussian noise. Corresponding to different production rates, there are six modes of process operation. As TEP is unstable on its own, a control system is required to stabilize
Figure 2.12: Tennessee Eastman Process
Table 2.2: Process disturbances

<table>
<thead>
<tr>
<th>Variable number</th>
<th>Process variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDV (1)</td>
<td>A/C feed ratio, B composition constant (stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (2)</td>
<td>B composition, A/C ratio constant (stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (3)</td>
<td>D feed temperature (stream 2)</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (4)</td>
<td>Reactor cooling water inlet temperature</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (5)</td>
<td>Condenser cooling water inlet temperature</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (6)</td>
<td>A feed loss (stream 1)</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (7)</td>
<td>C header pressure loss—reduced availability (stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>IDV (8)</td>
<td>A, B, C feed composition (stream 4)</td>
<td>Random variation</td>
</tr>
<tr>
<td>IDV (9)</td>
<td>D feed temperature (stream 2)</td>
<td>Random variation</td>
</tr>
<tr>
<td>IDV (10)</td>
<td>C feed temperature (stream 4)</td>
<td>Random variation</td>
</tr>
<tr>
<td>IDV (11)</td>
<td>Reactor cooling water inlet temperature</td>
<td>Random variation</td>
</tr>
<tr>
<td>IDV (12)</td>
<td>Condenser cooling water inlet temperature</td>
<td>Random variation</td>
</tr>
<tr>
<td>IDV (13)</td>
<td>Reaction kinetics</td>
<td>Slow drift</td>
</tr>
<tr>
<td>IDV (14)</td>
<td>Reactor cooling water valve</td>
<td>Sticking</td>
</tr>
<tr>
<td>IDV (15)</td>
<td>Condenser cooling water valve</td>
<td>Sticking</td>
</tr>
<tr>
<td>IDV (16)</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>IDV (17)</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>IDV (18)</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>IDV (19)</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>IDV (20)</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

Disturbances 14–20 should be used in conjunction with another disturbance from this table or a setpoint change. To realize the full effect of these disturbances, we suggest a simulation time of 24–48 h.

The output from TEP simulation containing 41 measurements forms training and test sets. A total of 20 process disturbances, of which 15 are known, is available in the simulated process (see Table 2.2, reproduced from [15]). It is suggested in [15] that for comparing simulation results, 4 load disturbances including idv(1), idv(4), idv(8), idv(12) and idv(15) should be introduced into the simulated process at the base case (Mode 1).

2.5.2 Methodology

Simulation time is set at 140 hours. Training and test data sets are obtained by running TEP simulation in Matlab Simulink which is available for download from [20]. Of 20 preprogrammed load disturbances that are available, three faults are simulated as suggested in [15] (idv(1), idv(4), and idv(8)). Fault idv(1) is introduced at sample 3000\textsuperscript{th} and corrected at 4000\textsuperscript{th}. Fault idv(4) is introduced at 7000\textsuperscript{th} and
corrected at 8000th. Similarly for fault ide(8) at 10000th and 11000th respectively. Based on sensitivity analysis (Section 3.2), moving window length is set at 600 samples; number of PCs retained \( a = 2 \); levels of significance for \( T^2 \) and \( Q \) statistics are set at 99% and 99.9% respectively.

The results of MPCA, APCA and EWPCA approaches are shown in Figures 2.13, 2.14 and tabulated in Tables 2.3, 2.4. Results from conventional PCA method are also presented for comparison.

### 2.5.3 Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (s/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional PCA</td>
<td>0</td>
<td>13</td>
<td>90</td>
<td>1.1 \cdot 10^{-4}</td>
</tr>
<tr>
<td>MPCA</td>
<td>0</td>
<td>4</td>
<td>92</td>
<td>2.6 \cdot 10^{-3}</td>
</tr>
<tr>
<td>APCA</td>
<td>0</td>
<td>4</td>
<td>95</td>
<td>2.0 \cdot 10^{-2}</td>
</tr>
<tr>
<td>EWPCA</td>
<td>1</td>
<td>3</td>
<td>145</td>
<td>6.0 \cdot 10^{-2}</td>
</tr>
</tbody>
</table>

The results shown in Figures 2.13, 2.14 as well as Tables 2.3, 2.4 indicate that all methods except EWPCA perform reasonably well. Every simulated fault is detected. There are no more than 18 false alarms during 140 simulation hours, corresponding to 0.13 alarm (per hr). This is quite low compared to similar industrial figures which range from 0.3 to more than 100 alarms (per hr) during normal operation [18]. Maximum delay appears a bit excessive: the faults are detected after around 0.90
Figure 2.13: Performance of four PCA methods in monitoring TEP– $T^2$ statistic. Simulated faults include $idv(1)$ (feed composition), $idv(4)$ (reactor cooling water inlet temperature) and $idv(8)$ (feed composition) at 3000–4000, 7000–8000, 10000–11000, respectively.
Figure 2.14: Performance of four PCA methods in monitoring TEP – $Q$ statistic. Simulated faults include \textit{idv}(1) (feed composition), \textit{idv}(4) (reactor cooling water inlet temperature) and \textit{idv}(8) (feed composition) at 3000–4000, 7000–8000, 10000–11000, respectively.
Table 2.4: Performance in TEP simulation using $Q$ statistic

<table>
<thead>
<tr>
<th>Methods</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (s/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional PCA</td>
<td>0</td>
<td>18</td>
<td>85</td>
<td>$1.6 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>MPCA</td>
<td>0</td>
<td>3</td>
<td>88</td>
<td>$3.0 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>APCA</td>
<td>0</td>
<td>3</td>
<td>88</td>
<td>$2.0 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>EWPCA</td>
<td>0</td>
<td>8</td>
<td>91</td>
<td>$6.1 \cdot 10^{-2}$</td>
</tr>
</tbody>
</table>

hour $\equiv 54$ minutes. Nevertheless, this result is even slightly better than those published in [21], where the corresponding delays are from 60 to 69 minutes for different PCA–based monitoring methods. Numerical results reveal that Fault $idv(8)$, which involves random variation in the feed composition, is responsible for the maximum delay. It is believed that the excessive delay is due to the fault nature and compensation action of the control system. Moreover, it is noted in Figure 2.13 that Fault $idv(4)$ actually causes two alarms. The reason for it might be that the fault effect is not significant and have been cancelled out by the controller well before the corrective action is introduced. As a result, the corrective action plays instead the role of a new disturbance and results in the second alarm (which is considered as a false alarm). Visual inspection of the simulation data supports the above explanation.

Among the four approaches, EWPCA produces the worst result. It is the only method having missed detection. Using $T^2$ statistic, Fault $idv(4)$ is missed while Fault $idv(8)$ is detected only after 145 samples $\equiv 87$ minutes. Moreover, it also requires about 20 times more online computation time compared to MPCA for one sample as seen in Table 2.3 where the $cpu$ time is measured on a Pentium 4 computer.
(CPU 3.40 GHz; RAM 512 MB). Similar performance is observed for EWPCA with $Q$ statistic. Although there is no missed detection and having fewer false alarms compared to conventional PCA method, EWPCA requires more than 300 times online computational time.

Conventional PCA works comparatively well in this case. This is because TEP in the base mode (Mode 1) is a time-invariant, stationary process and no change in operation mode occurs during the simulation. MPCA appears to be the best among four approaches, having detected all three faults, giving fewest false alarms, similar detection delay and requiring less online computation. APCA performance is similar to that of MPCA but requires about 10 times more online computational resource.

2.6 Industrial case study

2.6.1 Process description

Diagram of the process which is used in this case study is shown in Figure 2.15. It corresponds to the reaction section of a petrochemical plant located in Singapore. In this process, preheated benzene and hydrogen enter the primary reactor containing a catalyst suspended in liquid cyclohexane. The exothermic reaction occurs between benzene and hydrogen to produce cyclohexane. The heat released in the reaction is removed and used to generate steam in the steam drum. Overhead product stream from the primary reactor is feed into the secondary packed bed reactor to further hydrogenate the remaining benzene. Cyclohexane product stream from the secondary reactor is cooled down before dueing for further processing.
Analysis is carried out using two sets of data. Each set corresponds to one-week operation of the plant (from 14 September 1999 to 20 September 1999 and from 07 February 2000 to 12 February 2000). Seven variables, which are related to the steam drum operation, were recorded every minute and used for this study. These variables (5 of them are shown in Figure 2.15; the other two are manipulated variables for the temperature and level controllers) are temperature (3 variables, °C), pressure (1 variable, kPa), volumetric flow (1 variable, m³/h) and level percentage (2 variables, %).

Figure 2.15: Process diagram for the industrial case study

One-day data (X16, corresponding to data recorded on 16 September 1999) from the September set is used as the training set in all of the techniques (except for adaptive PCA which used the data recorded just before the test data set (X08),
corresponding to data recorded on 08 February 2000). Moving window length is set at 48 samples. Only the monitoring results of one–day data in the February set (X08) is displayed. Upon inspection of the typical variables, it is believed that there are process upsets at around 180–600\textsuperscript{th} measurements, an outlier around 880\textsuperscript{th} and a process drift from around 980\textsuperscript{th} sample.

2.6.2 Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (s/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional PCA</td>
<td>3</td>
<td>1</td>
<td>–</td>
<td>$5.4 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>MPCA</td>
<td>0</td>
<td>3</td>
<td>22</td>
<td>$2.5 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>APCA</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>$4.7 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>EWPCA</td>
<td>2</td>
<td>0</td>
<td>42</td>
<td>$5.0 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (s/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional PCA</td>
<td>3</td>
<td>1</td>
<td>–</td>
<td>$2.2 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>MPCA</td>
<td>0</td>
<td>2</td>
<td>22</td>
<td>$2.4 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>APCA</td>
<td>2</td>
<td>0</td>
<td>22</td>
<td>$8.3 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>EWPCA</td>
<td>2</td>
<td>0</td>
<td>40</td>
<td>$4.5 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Figures 2.16, 2.17 and Tables 2.5, 2.6 show the results of analyzing the industrial data.
Figure 2.16: Performance of four PCA methods in industrial case study – $T^2$ statistic
Figure 2.17: Performance of four PCA methods in industrial case study – $Q$ statistic
sets using conventional PCA method and three PCA–based approaches described earlier. The results have confirmed observation from simulation study above: MPCA appears to be the better approach of all three. It detects the disturbed region around 180–600\textsuperscript{th} measurements, the outlier 880\textsuperscript{th} and the drifting from 980\textsuperscript{th}. A number of false alarms occurred in MPCA at early stage of the monitoring, which might be due to initialization of the online algorithm. Nevertheless, the false alarm rate – 3 alarms for 24 hours or equivalent to 0.13 alarm per hour, is much smaller than corresponding industrial figures (from 0.3 alarm per hour to 100 alarms per hour) published in [18]. In addition, even though the quantitative difference in performance between MPCA and APCA is marginal, being a simpler approach, MPCA requires much less computational resource than APCA (see Section 2.4). In addition, APCA appears less sensitive and has missed detection in both $T^2$ and $Q$ cases. In every fault that is missed, the indices do exceed their corresponding limits but fewer than 4 consecutive times. As a result, the detection rule is not satisfied and alarm is not announced. EWPCA is the least sensitive approach compared to MPCA and APCA because at the outlier sample and the drifting fault, the monitoring indices are well below the control limits.

Contrary to the results from TEP simulation study, the performance of conventional PCA has become totally unreliable. As seen from Figures 2.16 and 2.17, conventional PCA approach would classify every test sample as faulty sample in this situation where the training and test data are from different operating regions. This limitation of conventional PCA approach has been widely acknowledged and it is in fact the motivation for the development of other PCA–based methods including
MPCA, APCA and EWPCA. Regarding MPCA approach, it is interesting that by using a simple scaling method, a PCA model built upon some five-month old data (in September 1999) can actually be used 5 months later (in February 2000) for monitoring an industrial process operating at different conditions/modes.

2.7 Chapter conclusion

It is demonstrated in this Chapter 2 and acknowledged in literature elsewhere (eg. [40, 41, 42]) that conventional PCA approach might not work satisfactorily with processes with multiple operation modes. Extensive research in dealing with this PCA disadvantage have resulted in various modified PCA–based approaches. The most extensive research direction has been to accommodate the process time dependence by updating the PCA monitoring model. Two representative methods of this strategy are studied including adaptive PCA (APCA) and exponentially weighted PCA (EWPCA).

This chapter proposes a simpler approach in dealing with processes with multiple operation modes. It is termed moving PCA (MPCA) and is based on maintaining a moving window and using the mean and standard deviation of that moving window for scaling purpose. The basis of MPCA development lies in the two observations: changes in the mean and standard deviation reflect the process variation between different operation modes and that process covariance structure for different operating modes shares common structure [5] which could be captured by one single PCA model.
MPCA performance is compared with that of APCA, EWPCA as well as conventional PCA in monitoring TEP simulation and analyzing an industrial data set. In monitoring TEP, which is a simulated time–invariant process, results indicate that conventional PCA, APCA and MPCA methods perform reasonably well compared to published industrial statistics. MPCA appears to be the best with fewer false alarms and lower computational requirement although APCA performance is marginally close. In analyzing the industrial data set, conventional PCA has become unreliable and this is expected as the training and test data sets are from different operating regions. MPCA, having no missed detection and few false alarms, outperforms the other approaches. APCA and EWPCA become less sensitive and require more online computation.

Overall, MPCA is a PCA–based monitoring method which is simpler, yet outperforms APCA, and EWPCA in monitoring time–invariant process (TEP simulation) and especially industrial processes operating at different regions/modes.
Chapter 3

Evaluation of MPCA Robustness

3.1 Introduction

In order to implement MPCA approach, a number of parameters have to be selected. They include moving window size, the number of principal components (PCs) retained, and confidence level. This chapter investigates the robustness of MPCA method to variation in those parameters.

The approach taken is empirical i.e., observe changes in MPCA performance while varying the parameters. In terms of moving window, a size range from approximately 1 to 4 process time constant is studied. The reason is that within that range, essential process dynamics would be captured by such a moving window. Regarding the number of PCs retained, there are a number of methods to determine the optimal number of PCs that should be used (see Section 1.4.2). In this study, the number of PCs retained is increased from 1 to the smallest number needed to capture about
90% process variance. (The process variance associated with a PC is equal to the square of its corresponding singular value). In addition, confidence limit is also one of the most influential parameters in MPCA performance. It is directly related to the upper control limit, which in turn affects the monitoring results in two ways: firstly whether the test sample is normal/in-control or faulty and secondly as a result, whether the moving window should be updated. There is no formal analytical approach to determining the optimal confidence limit. As a result, this study will set the parameter at values which are either used commonly in practice or suggested in published literature.

Moreover, as many monitoring indices are available, the question of which one is best used for MPCA approach deserves a comparative study. Typically, Hotelling’s $T^2$ statistic and $Q$ statistic, which is also known as the squared prediction error (SPE), are used for process monitoring and fault detection. These are two of the most common monitoring indices. Owing to the complementary nature of these two indices, the combined $QT$ index has been proposed as reviewed in [16]. In addition, to improve the sensitivity of the $Q$ statistic, J. A. Westerhuis introduced the standardized $Q$ statistic by taking into account the expected variation of the residuals [2]. All these newly proposed indices were reported to perform better conventional statistics (Hotelling’s $T^2$ and $Q$) in one way or another [2, 16]. In this chapter, performance of MPCA approach using these indices including conventional $T^2$, $Q$, combined $QT$ and standardized $Q$ statistics is studied and evaluated.
3.2 Moving window size

Sensitivity of MPCA performance is investigated by applying the method to both TEP simulation and the industrial case study over a range of moving window sizes. As the time constants \( \tau \) for TEP simulation and for the industrial case study are approximately \( \tau_T = 500 \) samples (5 hours simulation time) and \( \tau_c = 50 \) minutes respectively (see Appendix A), the moving window size range is from 200 to 2000 samples for TEP simulation study and from 36 to 200 samples for the industrial case study. The results are shown in Tables 3.1–3.3 for \( T^2 \) and \( Q \) statistics in both TEP simulation and in the industrial case study.

**TEP simulation**

<table>
<thead>
<tr>
<th>Window size (samples)</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 (0.4( \tau_T ))</td>
<td>0</td>
<td>10</td>
<td>91 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>300 (0.6( \tau_T ))</td>
<td>0</td>
<td>7</td>
<td>91 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>400 (0.8( \tau_T ))</td>
<td>0</td>
<td>6</td>
<td>92 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>500 (1.0( \tau_T ))</td>
<td>0</td>
<td>4</td>
<td>92 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>600 (1.2( \tau_T ))</td>
<td>0</td>
<td>4</td>
<td>92 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>1000 (2.0( \tau_T ))</td>
<td>0</td>
<td>4</td>
<td>91 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>1500 (3.0( \tau_T ))</td>
<td>0</td>
<td>3</td>
<td>91 (0.18( \tau_T ))</td>
</tr>
<tr>
<td>2000 (4.0( \tau_T ))</td>
<td>1</td>
<td>3</td>
<td>91 (0.18( \tau_T ))</td>
</tr>
</tbody>
</table>

Results shown in Tables 3.1 and 3.2 and Figure 3.1 indicate that there exists an
optimal moving window size around $\tau_T$ (500 samples) to $2\tau_T$ (1000 samples). For $T^2$ statistic, the number of false alarms starts at 10, decreases to 4 using $\tau_T - 2\tau_T$ window. Fewer false alarms is possible but might not be recommended as missed detection begins to occur (as with moving window of $4\tau_T$). On the other hand, for $Q$ statistic, minimum number of false alarms is 2 using $\tau_T - 1.2\tau_T$ moving window size. The existence of a minimum is interesting because it had been expected that MPCA with a bigger moving window would be less sensitive compared to a smaller window size, leading to fewer false alarms and more missed detection for bigger more windows. Instead, more false alarms can be observed in Table 3.2. In some cases, as the sensitivity decreases with increasing window size, one false alarm might be broken down and form two (false) alarms. However, in other cases, there has been no clear reason for why it was so.

More importantly, MPCA performance measured in terms of missed detection and
false alarms, deviates little from its optimal performance (at the minimum) over a wide range of window size. As can be seen in Figure 3.1 (as well as in Tables 3.1 and 3.2), for moving windows of 500 up to 1500 samples using $T^2$ index, and 500 to 1000 samples using $Q$ statistic, the number of false alarms varies only by 2 while the missed detection stays at 0. In this respect, it could be said that $T^2$ statistic would be a safer choice due to a wider range of moving window size, even though it might give slightly higher false alarm rate.

In terms of detection delay, effects of moving window size are less obvious. As the size changes from 200 to 2500 samples, the maximum detection delays stay at around 90
samples (equivalent to about 0.9 hour or 54 minutes). Numerical results point to the third fault $idv(8)$, which is a random disturbance in TEP feed component, for causing the maximum delays. Similar detection delays are observed and published in [21], where the corresponding delays are from 60 to 69 minutes for the same fault using different PCA–based methods. As a result, it is believed the excessive delay is due to the fault nature. Visual inspection reveals that after Fault $idv(8)$ is introduced, it took about 80 samples (or equivalently 48 minutes) for the process to exhibit any significant effect while the earlier manifestation might have been absorbed by the control system. Consequently, the maximum delay is considered to be in reasonable range.

**Industrial case study**

Corresponding results from analyzing the industrial data set are shown in Table 3.3. For reference, the time constant of the industrial process is estimated $\tau_c \approx 50$ minutes (see Appendix A) and sampling time is 1 minute/sample.

Contrary to the TEP study, the result from the industrial case study does not reveal an optimal window size where both the numbers of false alarms and missed detection are both at minimum. Instead, it only seems that a bigger moving window is less sensitive as clearly shown for $T^2$ statistic in Table 3.3. As the window size grows from 48 samples (the base case), the number of missed detection increases to 2, while the false alarm decreases to 1. Similar trend appears for $Q$ statistic even though the number of false alarms fluctuates a bit before decreasing. For moving window of as few as 36 samples, MPCA does not work well probably because dynamics of the
<table>
<thead>
<tr>
<th>Window size</th>
<th>$T^2$ missed detection</th>
<th>$T^2$ false alarms</th>
<th>$Q$ missed detection</th>
<th>$Q$ false alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>36 ($0.7\tau_c$)</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>42 ($0.8\tau_c$)</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>48 ($1.0\tau_c$)</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>54 ($1.1\tau_c$)</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>60 ($1.2\tau_c$)</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>70 ($1.4\tau_c$)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>80 ($1.6\tau_c$)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>100 ($2.0\tau_c$)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>150 ($3.0\tau_c$)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>200 ($4.0\tau_c$)</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.3: MPCA robustness to window size in industrial case study

process have not been effectively captured by the moving window. Interestingly, for 42 samples, $Q$ statistic gives the best performance while it is one of the worst for $T^2$ index in terms of missed detection and false alarms.

It should be noted that MPCA in analyzing the industrial case study seems to always achieve reasonable false alarm rate. With only fewer than 4 alarms for 24 hours, it is even better than results published by a survey by UK Safety and Health Executive [18], which reported that typical false alarm rates following plant upset were 600 alarms in 7 hours. (However, it is acknowledged that the number of process variables being monitored could be an important factor influencing alarm rates). Nevertheless, giving that performance measured by number of missed detection and
false alarms, the optimal window size in the industrial case study for $T^2$ statistic would be 48 samples ($1.0 \tau_c$) while for $Q$ statistic it appears 42 – 54 samples ($0.8 \tau_c$ – $1.1 \tau_c$). As seen from Table 3.3, any deviation from this optimal size likely results in significant changes in MPCA performance, especially for $T^2$ statistic. This is in contrast with observation from TEP study where the optimal performance would deteriorate slowly over a much wider range of window sizes.

In conclusion, the sensitivity analysis suggests that the optimal moving window size for MPCA monitoring is about $\tau$ to $2 \tau$ ($\tau$ is the process time constant), where minimum number of missed detection and false alarms can be achieved. Overall, MPCA is more sensitive to moving window size in the industrial case study than in TEP simulation study.

### 3.3 Number of principal components retained

MPCA robustness to variation in the number of principal components retained is studied. In TEP simulation, PCA models retain 1 up to 30 PCs. In the industrial case study, 1 up to 5 PCs are used. Numerical results are shown in Tables 3.4–3.7 and Figure 3.2. Cumulative variance, which is the percentage of total process variance captured by retaining the first $a$ principal components in the PCA model, is also shown.
Tables 3.4 and 3.5 show the numerical results from MPCA approach using 1-30 PCs in TEP simulation. As can be seen, the number of missed detection is not very sensitive to the number of PCs retained. Using any number of PCs, all simulated faults are detected except for $T^2$ statistic with 1 and 3 PCs where one fault is missed. In addition, when there is no missed fault, the maximum detection delays hardly change and appears reasonable as discussed previously in Section 3.2. When $T^2$ statistic with only 1 principal vector is used, essential process variation might not be explained in the PCA score space and hence lead to missed detection of Fault

Figure 3.2: False alarms for different number of PCs retained – TEP simulation

**TEP simulation**
Table 3.4: Sensitivity to number of PCs retained in TEP simulation–$T^2$ statistic

<table>
<thead>
<tr>
<th>No. of PCS</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cumulative variance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>–</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>92 ($0.18\tau_T$)</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>91 ($0.18\tau_T$)</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2</td>
<td>91 ($0.18\tau_T$)</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
<td>91 ($0.18\tau_T$)</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>5</td>
<td>89 ($0.18\tau_T$)</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>3</td>
<td>89 ($0.18\tau_T$)</td>
<td>68</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>10</td>
<td>89 ($0.18\tau_T$)</td>
<td>89</td>
</tr>
</tbody>
</table>

Table 3.5: Sensitivity to number of PCs retained in TEP simulation–$Q$ statistic

<table>
<thead>
<tr>
<th>No. of PCS</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cumulative variance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>88 ($0.18\tau_T$)</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>88 ($0.18\tau_T$)</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>88 ($0.18\tau_T$)</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2</td>
<td>88 ($0.18\tau_T$)</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2</td>
<td>88 ($0.18\tau_T$)</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>3</td>
<td>89 ($0.18\tau_T$)</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>5</td>
<td>88 ($0.18\tau_T$)</td>
<td>68</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>9</td>
<td>84 ($0.17\tau_T$)</td>
<td>89</td>
</tr>
</tbody>
</table>
When 3 PCs are used, $T^2$ statistic misses Fault $idv(4)$. The reason is due to the detection rule: the statistical index exceeds the control limit less than 4 times consecutively.

However, as can be seen from Figure 3.2, the number of PCs retained $a$ seems to have a greater effect on the number of false alarms, which would increase significantly when as few as 1 or more than 20 PCs are retained. The reason is that when 1 PC is retained, the PCA model is not an accurate description of the TEP process as a significant part of process variation is not captured by the retained PCs. Hence it results in more false alarms and possibly missed detection (as in $T^2$ case). On the other hand, when more than 20 PCs are retained, PCA model might try to describe more process variation than necessary (for monitoring purpose), as well as the noise involved. As a result, it is not able to differentiate between a fault and a noise–corrupted data sample, which might, depending on the statistical significance level, lead to excessive number of either false alarms or missed detection.

As shown in Tables 3.4 and 3.5, the optimal values for $a$ are 4 or 5 for both $T^2$ and $Q$ statistics. The base case scenario is set at 2 PCs. This is also another optimal settings for using $Q$ statistic as shown in Table 3.5. However, as suggested in Table 3.4 for $T^2$ statistic, $a = 2$ PCs retained leads to more false alarms than if 4 or 5 PCs are used. Interestingly, by retaining 2 to 5 PCs, the PCA model would explain less than 30% of the total process variance, yet achieve good performance in process monitoring and fault detection. This implies that the percentage of overall process variance captured by the PCA model has little or no direct influence on process monitoring performance in this case. This observation seems not to support the percent variance
method to determine the number of PCs retained (discussed earlier in Section 1.4.2. Similarly, the scree test shown in Figure 3.3 suggests that the number of PCs retained would be 3. Clearly, it is not the optimal and even leads to missed detection (in $T^2$ case).

Industrial case study

In the industrial case study, MPCA appears much more sensitive to the number of PCs retained. Again, 2 PCs are retained in the base case scenario. As can be seen in Tables 3.6 and 3.7, any deviation from this scenario increases the number of missed
Table 3.6: Sensitivity to number of PCs retained in industrial case study—$T^2$ statistic

<table>
<thead>
<tr>
<th>No. of PCS</th>
<th>missed detection</th>
<th>false alarms</th>
<th>cumulative variance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>62</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>87</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>95</td>
</tr>
</tbody>
</table>

Table 3.7: Sensitivity to number of PCs retained in industrial case study—$Q$ statistic

<table>
<thead>
<tr>
<th>No. of PCS</th>
<th>missed detection</th>
<th>false alarms</th>
<th>cumulative variance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>62</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>87</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>95</td>
</tr>
</tbody>
</table>

detection. For example, if 1 or 3 PCs are used, out of the three identified faults none is detected using $Q$ statistic. Similar observations can be seen in Table 3.6 for MPCA using $T^2$ statistic with 1 or more than 3 PCs.

In terms of false alarms, effect of the number of PCs retained is not clear. In $T^2$ case (Table 3.6), the number of false alarms decreases to 1 (for 3 PCs) then increases to 5 (for 5 PCs). On the other hand, the number of false alarms in $Q$ statistic (Table 3.7) seems to slowly decrease with increasing number of PCs retained.

It is interesting that in both TEP simulation and the industrial data set, 2 PCs are
sufficient for a PCA model used in monitoring and fault detection, which is supported by a heuristic rule in [17] that only 2 or 3 PCs should be used. The scree test shown in Figure 3.4 suggests retaining the first 3 PCs for optimal performance. However, numerical results in Tables 3.6 and 3.7 do not support this suggestion. In addition, those tables also show that only 62% of process variation is captured using 2 PCs. This seems to agree with the earlier observation that the process variation captured in the PC score space does not have significant influence on MPCA performance.

In conclusion, MPCA performance is very sensitive to the number of PCs retained in analyzing the industrial data set but not in monitoring TEP simulation. This has implied that extensive tuning might be required to achieve optimal performance for
the proposed MPCA approach. Validity of scree test method and percent variance method combined with MPCA approach might be questionable as they could not provide the number of PCs that should be retained for optimal MPCA performance. Instead of using those methods, it is shown that retaining 2 PCs works reasonably well for both TEP simulation and the industrial case study although further tuning might (or might not) improve the monitoring performance.

3.4 Confidence limit

Common practice often sets the significant levels at 0.05 and 0.01, which correspond to the confidence limits of 95% and 99% respectively. However, there exists published literature which suggests 99.9% for the confidence limit to avoid excessive number of false alarms [27]. As a result, sensitivity of MPCA performance is carried out with 4 different confidence limits including 95.00%, 99.00%, 99.90% and 99.99%. The results are shown in Table 3.8 for TEP simulation and in Table 3.9 for the industrial case study.

TEP simulation

As seen in the results for TEP simulation, there seems to be an obvious relationship between the confidence limit and the MPCA performance in terms of false alarms and missed detection. Table 3.8 indicates that as the confidence limit increases, there are fewer false alarms but more faults might be missed. This is due to increasing the confidence levels lead to higher upper control limits, which would in turn give
fewer false alarms possibly at the expense of missed detection. The secondary effect of confidence limits (on updating the moving window) seems not very significant, which is due to two factors. Firstly, the TEP simulation does not experience any changes in operating conditions/mode (except for the faults introduced). Secondly, the moving window captured effectively the process dynamics. Both of these factors might have made updating the window have little effect on overall MPCA performance. From Table 3.8, the optimal confidence limits for $T^2$ statistic and $Q$ statistic should be around 99.00% and 99.90% respectively for optimal MPCA performance. In fact, they are chosen for the base case scenarios.

Table 3.8: MPCA performance using different confidence limits – TEP simulation

<table>
<thead>
<tr>
<th>Confidence limits</th>
<th>$T^2$ missed detection</th>
<th>$T^2$ false alarms</th>
<th>$Q$ missed detection</th>
<th>$Q$ false alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.00%</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>99.00%</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>99.90%</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>99.99%</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**Industrial case study**

Similar observations can not be seen from the results for analyzing the industrial data set. It would be expected that a higher confidence limit would give fewer false alarms and possibly more missed detection. This is partly observed in Table 3.9, especially in results for $T^2$ statistic. However, the reverse is not true for using low confidence limit. For example, at 95% confidence level, both $T^2$ and $Q$ statistics
have 3 missed detection. The reason is that using low confidence level would on one hand set a low control limit which leads to more false alarms. On the other hand, it would prevent the data which might contain essential process variation but is falsely classified as abnormal, from being used to update the moving window. In doing so, it would invalidate the attempt to capture process dynamics using the moving window, consequently leading to poor MPCA performance.

The base case scenarios are 99.00% confidence for $T^2$ statistic and 99.90% confidence for $Q$ statistic. Table 3.9 shows that if any other confidence limits had been used, MPCA would perform poorly. It is therefore concluded that MPCA performance is very sensitive to the confidence limits used.

Table 3.9: MPCA performance using different confidence limits – industrial case

<table>
<thead>
<tr>
<th>Confidence limits</th>
<th>$T^2$ missed detection</th>
<th>$T^2$ false alarms</th>
<th>$Q$ missed detection</th>
<th>$Q$ false alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.00%</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>99.00%</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>99.90%</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>99.99%</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
3.5 Monitoring indices

3.5.1 Theory and implementation

Combined QT statistics

Although both Hotelling’s $T^2$ and $Q$ statistics are commonly used for process monitoring, it is widely acknowledged that they detect different types of faults [16, 21] and their role are somewhat complementary. To take advantages of both statistics, combined indices have been proposed.

Yue and Qin [10] introduced a combined index for $T^2$ and $Q$ as follows

$$\varphi = \frac{T^2(x)}{\chi^2_{2,\alpha}} + \frac{SPE(x)}{\delta^2_{\alpha}} = x^T \Phi x$$

$$\Phi = \frac{PA^{-1}P^T}{\chi^2_{2,\alpha}} + \frac{I - PP^T}{\delta^2_{\alpha}}$$

(3.1) (3.2)

Where: $\chi^2_{2,\alpha}$ and $\delta^2_{\alpha}$ are the upper control limits for $T^2$ and $Q$ statistics respectively at the corresponding significant level $\alpha$.

In order to apply this index for process monitoring, the upper control limit for $\varphi$ is required. It was derived in [10] using an approximate distribution of quadratic forms as follows

$$\varphi = x^T \Phi x \sim g\chi^2_{h}$$

(3.3)

Where:

$$g = \frac{tr(S\Phi)^2}{tr(S\Phi)} \quad h = \frac{[tr(S\Phi)]^2}{tr(S\Phi)^2}$$

(3.4)
It was shown in [16] that

\[ tr(\mathbf{S\Phi}) = \frac{a}{\chi^2_{a,\alpha}} + \frac{\sum_{i=a+1}^{m} \lambda_i}{\delta^2_{\alpha}} \]  

(3.5)

\[ tr(\mathbf{S\Phi})^2 = \frac{a}{\chi^4_{a,\alpha}} + \frac{\sum_{i=a+1}^{m} \lambda_i^2}{\delta^4_{\alpha}} \]  

(3.6)

Where:  \( a \) is the number of PCs retained.

\( \lambda \) is the corresponding eigenvalues

After \( g \) and \( h \) are calculated, the upper control limit of \( \phi \) for a given significance level \( \alpha \) can be obtained \( g\chi^2_{h,\alpha} \).

The procedure to implement MPCA using the combined \( QT \) index in Figure 3.5 is very similar to that of MPCA using \( T^2 \) or \( Q \) statistic. The only difference is quite obvious: \( QT \) index needs to be evaluated and used for checking instead of \( T^2 \) or \( Q \).

The following settings is chosen as empirical analysis showed that it appears to be optimal for MPCA using \( QT \) index.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>TEP simulation</th>
<th>industrial case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of PCs</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Window size (samples)</td>
<td>600</td>
<td>78</td>
</tr>
<tr>
<td>Confidence limit</td>
<td>99.99%</td>
<td>99.99%</td>
</tr>
</tbody>
</table>

**Johan’s standardized \( Q \) statistics**

As suggested by the name, this monitoring index is a modified version of the \( Q \) statistic. In the original \( Q \), all squared residuals are summed without taking into
Figure 3.5: Algorithm for MPCA approach using $QT$ statistic
account the relative size of the residuals with respect to their expected variation. From this observation and being inspired by the use of standardized residuals for diagnostic purposes, Westerhuis J. A. et al. [2] proposed the standardized $Q$ statistic in an attempt to improve the sensitivity of the monitoring index by considering the residuals relative to their expected variation.

Scaled residual was introduced as follows

$$
\tilde{e}_{ij} = \frac{e_{ij}}{s_j}
$$

Where: $e_{ij}$ is the $i^{th}$ residual for the $j^{th}$ variable.
$s_j$ is the standard deviation of the normal operating condition residual for the $j^{th}$ variable.

The new Johan’s standardized $Q$ statistic:

$$
Q_J = \tilde{\theta}^T \tilde{\theta} \sim \tilde{g} \chi^2_{\tilde{h}}
$$

Where: $\tilde{g}$ and $\tilde{h}$ are the corresponding moments of the $\chi^2$ distribution of the scaled residuals $\tilde{e}$

Let $\tilde{E}$ be the matrix of the scaled residuals from the training data set, and $\tilde{\Sigma}$ be the covariance matrix of $\tilde{E}$. Then,

$$
\tilde{g} = \frac{\tilde{\theta}_2}{\tilde{\theta}_1}
$$

$$
\tilde{h} = \frac{\tilde{\theta}_1^2}{\tilde{\theta}_2}
$$

Where:

$$
\tilde{\theta}_i = trace(\tilde{\Sigma}^i) \quad i = 1, 2
$$
The implementation of the newly proposed MPCA approach with the Johan’s standardized $Q$ statistic is very similar to MPCA using the original $Q$ index. The difference is that the residuals need to be replaced by the scaled ones and the upper control limit $\tilde{g}\chi^2_{k,\alpha}$ for a chosen significant level $\alpha$ must be calculated accordingly. In addition, the moving window is taken as the normal operating condition and hence $s_j$ is evaluated (online) based on the moving window. The following parameter settings is chosen as a result of empirical analysis.

Table 3.11: Parameter settings for MPCA using Johan’s standardized $Q$ index

<table>
<thead>
<tr>
<th>Parameters</th>
<th>TEP simulation</th>
<th>industrial case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of PCs</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Window size (samples)</td>
<td>700</td>
<td>80</td>
</tr>
<tr>
<td>Confidence limit</td>
<td>99.99%</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

3.5.2 Comparative results

The comparative study of MPCA approach using different indices is meant to serve two purposes. The first one is to determine whether or not the performance of the proposed MPCA approach strongly depends upon the index it employs. The second goal is to identify if possible the index which would yield the best overall performance. The results obtained from analyzing TEP simulation and the industrial data set, are tabulated in Tables 3.12 and 3.13. Besides false alarms, missed detection, and maximum detection delay, cpu time in $10^{-3}$s is also shown. The additional num-
ber gives how much time (per sample) the online processing would require (on a Dell Pentium IV 3.4G, 512Mb RAM PC) and hence is a relative measure of computational requirement for each method.

<table>
<thead>
<tr>
<th>Indices</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (ms/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional $T^2$</td>
<td>0</td>
<td>4</td>
<td>92</td>
<td>2.8</td>
</tr>
<tr>
<td>conventional $Q$</td>
<td>0</td>
<td>2</td>
<td>88</td>
<td>2.6</td>
</tr>
<tr>
<td>combined $QT$</td>
<td>0</td>
<td>2</td>
<td>90</td>
<td>2.7</td>
</tr>
<tr>
<td>standardized $Q$</td>
<td>0</td>
<td>2</td>
<td>84</td>
<td>5.5</td>
</tr>
</tbody>
</table>

As can be seen in Table 3.12 for TEP simulation, there is not much difference between MPCA performance using different statistics although $T^2$ has a few more false alarms than any other indices. All three simulated faults are detected; maximum detection delays for every indices are roughly the same; the time to analyze one data sample for each statistic are also similar (except for standardized $Q$ which takes about twice longer).

However, as shown in Table 3.13, the difference between those statistics become more significant when applying to the industrial data set. In terms of missed detection and false alarms, the combined $QT$ index appears to be the least sensitive with no false alarms and 1 missed detection. Conventional $T^2$ statistic continues to be the most sensitive index producing 3 false alarms. Interestingly, the standardized $Q$, which is modified to improve its sensitivity, does not seem more sensitive than the other indices including the conventional $Q$ statistic.
Table 3.13: Comparative study of MPCA performance – industrial case study

<table>
<thead>
<tr>
<th>Indices</th>
<th>missed detection</th>
<th>false alarms</th>
<th>max delay (samples)</th>
<th>cpu time (ms/sample)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional $T^2$</td>
<td>0</td>
<td>3</td>
<td>22</td>
<td>0.23</td>
</tr>
<tr>
<td>conventional $Q$</td>
<td>0</td>
<td>2</td>
<td>31</td>
<td>0.24</td>
</tr>
<tr>
<td>combined $QT$</td>
<td>1</td>
<td>0</td>
<td>33</td>
<td>0.21</td>
</tr>
<tr>
<td>standardized $Q$</td>
<td>0</td>
<td>2</td>
<td>40</td>
<td>0.38</td>
</tr>
</tbody>
</table>

The standardized $Q$ index is also the method which requires the most computational resource. Its *cpu time* per sample is about twice that of any other indices. The reason is that in implementing MPCA with the standardized $Q$ index, the residual is scaled against the standard deviation of the residuals from the moving window which needs to be evaluated online. This obviously takes more time to carry out and hence is shown up in the standardized $Q$ *cpu time* per sample.

Overall, MPCA performance in terms of false alarms and missed detection is considered reasonable in four cases with different monitoring indices. The worst false alarm rate is 3 out of 1440 samples, equivalent to $2.1 \cdot 10^{-3}$, which is about one order magnitude smaller compare to other methods [9]. Similarly for the missed detection rate. (The corresponding rates for MPCA in TEP simulation study are even smaller). These numbers are even slightly better than similar industrial statistics reported in [18].
3.6 Conclusion

The chapter has studied the sensitivity of the proposed MPCA approach to changes in a number of parameters including the moving window size, the number of PCs retained and the significant level for setting the upper control limit. It is found that in TEP simulation, MPCA performance does not change significantly to variations in the above parameters. On the contrary, it appears that MPCA is much more sensitive in the industrial case study. Any deviation from the base case scenario is likely to degrade MPCA performance. The results also indicate that the moving window size seems to be less important than the number of PCs retained and the confidence level.

In addition, the chapter also implements the proposed MPCA with different monitoring indices including conventional statistics ($T^2$ and $Q$), combined $QT$ and standardized $Q$. Comparison between MPCA using different monitoring indices indicates that conventional statistics including $T^2$ and $Q$ appear to work reasonably well compared to either combined $QT$ or standardized $Q$ index. Both false alarm and missed detection rates are very much lower than those from other published methods. The two newly proposed statistics including the combined $QT$ and standardized $Q$ need further modifications to improve their performance.
Chapter 4

Conclusion

To address the time–invariant limitation of PCA technique, this thesis proposes a Moving Principal Component Analysis (MPCA). Its performance is compared with that of other published approaches including conventional PCA, adaptive PCA, and Exponentially Weighted PCA in monitoring Tennessee Eastman Process (TEP) simulation and analyzing an industrial operation data set. MPCA is based on the idea that updating scaling parameters (mean and standard deviation from a moving window) is adequate for handling the process variation between different operation modes. It is shown that the proposed MPCA method is simpler and requires less online computation and storage, yet performs better than the other approaches when performance is measured by missed detection, false alarms, time delay and computational requirement.

MPCA robustness is investigated empirically by changing critical parameters including moving window size, number of principal components retained, and confidence limits. The results indicate that MPCA method is generally more sensitive in an-
alyzing the industrial data set. In addition, moving window size seems to be less critical than the other two parameters. Moreover, it is also shown that conventional statistics works better than the combined $QT$ and standardized $Q$ indices.

**Future work**

The proposed MPCA method is developed for monitoring processes at different operating modes. Further modification of MPCA is required for monitoring transition between process modes. One obvious solution is to integrate MPCA with other scheme such as described in R. Srinivasan et. al. ([1, 32]). In this way, the monitoring system would switch from MPCA to Srinivasan’s scheme when process transition occurs and back to MPCA when the process reaches new operating modes. Clearly, further work is required for designing such a switching mechanism.

In addition, it might be possible for MPCA to deal with process transition by using a different method in updating the moving window. Currently, it is updated when the new data is diagnosed as normal/in–control. Consequently, it captures normal process dynamics in an operation mode but the dynamics during mode transition is discarded. An alternative updating scheme for moving window would be: to update the window as soon as the new data becomes available regardless whether it is faulty or not. Consequently, a more robust approach in estimating the scaling parameters from the moving window needs to be employed and a mechanism to differentiate between a fault and a process transition is needed.
Bibliography


Appendix A

Process time constants

A.1 TEP

Initial steady state at 3000th sample: 9.2

New steady state: 10.1

After 1 time constant $\tau$: $9.2 + 0.632 \cdot (10.1 - 9.2) = 9.8$ which corresponds to 3500th sample.

The estimated time constant for TEP simulation is

$\tau_T = 3500 - 3000 = 500$ samples or 5 simulation hours.

A.2 Industrial case study

Similar calculation gives rough estimated time constant of the case study $\tau_c = 50$ minutes (or samples).
Figure A.1: TEP step response
Figure A.2: Step response for the industrial case study