University of Alberta

The treatment of Missing Data in Process Monitoring and Identification

by

Syed Ahmad Imtiaz 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta
Spring 2007

Your file Votre référence
ISBN: 978-0-494-29690-5
Our file Notre référence
ISBN: 978-0-494-29690-5

To My Parents
And
My Wife "Shoma"

Abstract

Process data suffers from many different types of imperfections, for example, bad data due to sensor problems, multi-rate data, outliers, data compression etc. Since most modeling and data analysis methods are developed to analyze regularly sampled and well conditioned data sets there is a need for pre-treatment of data. Traditionally these imperfections have been viewed as unrelated problems and dealt individually. In this thesis we treat these diverse problems under the general framework of 'treatment of missing data'. A vast amount of literature on the statistical analysis of data with missing values has flourished over last three decades mainly dealing with statistical surveys and biomedical data analysis. Therefore, the objectives of this study are to: (i) establish the link between the missing data literature and the process data analysis, so that the process engineering community can take advantage of these methods, (ii) extend some of the commonly used process data analysis tools using these formal methods for building models from data matrix with missing values and (iii) implement novel applications of missing data handling techniques in solving problems which may not appear as missing data problem directly.

This thesis has two main parts. Part-I of this thesis deals with 'off-line' modeling of 'latent variable models'. Principal Component Analysis (PCA), Iterative-PCA (IPCA) and Maximum Likelihood Factor Analysis (MLFA) are extended to the Data Augmentation framework for dealing with missing values. Missing data handling techniques have been applied to synchronize uneven length batch process data and recover the correlation between compressed signals. Data pre-processing issues other than missing values have been dealt with in relation to an industrial case study where PCA was used to detect sheet-breaks in a paper mill.

Part-II of the thesis deals with the 'on-line' filtering problem. The Sequential Monte Carlo (SMC) filter is extended to a Multiple Imputation framework for updat-

ing the filter with multi-rate measurements.

The improved performance of the proposed methods have been demonstrated using simulated examples, experimental data and industrial case study.

Acknowledgements

All praise and glory to Almighty ALLAH who gave me the knowledge and ability to bring this enormous task to completion. This in no way the end of seeking knowledge, not even the beginning of the end but merely the end of the beginning.

In this journey I was fortunate to have wonderful people at different stages who took me in their hands and gently placed me on one step ahead. I take this opportunity to thank all those hands, all my teachers from elementary to graduate studies who's dedication and help brought me here. Specially for this thesis, I gratefully acknowledge the supervision and support of Prof. Sirish Lalji Shah. I started my PhD as a half motivated person, but once I came in contact with Prof. Shah all my indecision vanished, I learned the joy of research and learning. He gave me ample of freedom and time to explore all the alleys of my interests. With his encouragement and support research was never strenuous on me. It was the most wonderful years in my learning. I thank him for giving me such an opportunity. I also thank Prof. Shankar Narasimhan of IIT, Madras, for his guidance and invigorating discussions during the initial period of my thesis which helped me in shaping my ideas. I am thankful to my course teachers and members of the supervisory committee, Prof. Peter Hooper and Prof. Biao Huang for introducing me to tow of the core areas of my thesis, multivariate statistics and system identification respectively.

I was also lucky to have wonderful colleagues, who always helped me in my needs and listened to my many dim ideas without a hint to it. To name a few, I would like to thank Dr. Kallol Roy, Dr. Weihua Li, Dr. Arun Tangirala, Dr. Shoukat Choudhury, Dr. Harigopal Raghavan, Dr. Zhengang Han, Dr. Salim Ahmed, David Chang, Rumana Sharmin, Monjur Murshed, Amar-Bin-Halim, Enayet-Bin-Halim, Haeli Zhang and Phanindra Jampana for their support and many helpful discussions.

I also thank all the office staff and the support staff of the department for their diligent support whenever I needed.

I gratefully acknowledge the financial support of NSERC, Matrikon, Sucor, i-CORE and ASRA.

The cold winters of Edmonton and the strains of graduate studies were much eased because of the warmth from the Bangladeshi community of Edmonton. It was like a big family where I could fall back in all my needs. My dream for graduate studies would have never come to reality without the sacrifice and encouragement of my family. I cannot thank enough my parents, brothers and sisters for their support throughout my life. I would also like to thank my parent-in-laws and sister-in-laws for their encouragement during the last crucial four years. Finally, I thank my loving

wife 'Shoma' for being so patient when I was stressed out from research and not in the my best mood. Without her care and gentle push it would have taken another year to finish the work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data driven methods are now extensively used in process industries for identification and monitoring purposes. Such methods require well conditioned data, i. e., regularly sampled, uncompressed or raw data that is properly time synchronized, has no bad values and without any outliers. However, it is common to encounter imperfection in process data. Some of the common causes that lead to imperfect data are:

Bad values due to sensor problems: Measurements may get corrupted due to failure of the measurement device and/or errors in data management. Common causes that lead to bad data are, sensor breakdown, measurement outside the range of the sensor, data acquisition system malfunction, energy black outs, interruption of transmission lines, wrong format in logged data, glitches in data management software, data storage errors etc. Some of these phenomena are illustrated in Figure 1.1. It shows a pressure signal where the measurement occasionally exceeded the range of the sensor, with missing values on a few occasions.

Multi-rate data: Sometimes data may not be available at the required time interval because of the nature of the sensors or the strategy of sampling, for example, quality measurements that come from an analyzer have lower sampling frequency compared to measurements such as temperature, flowrates, and pressure. Figure 1.2 shows composition measurement from an analyzer with 5 mins sampling interval where the rest of the variables are logged in every 1 min. In the Distributed Control System (DCS) a 'zero order hold' is applied to the measurements and measurements are supplied to the controller at every 1 min interval. In case of lab analysis where samples need to be collected manually, the measurement frequency may be completely irregular, asynchronous and there will also be a time lag between the instant the measurement is available and the actual sampling time. Measurements with different sampling rates are often collected in a single data matrix for monitoring or identification purpose and referred to as multi-rate data.

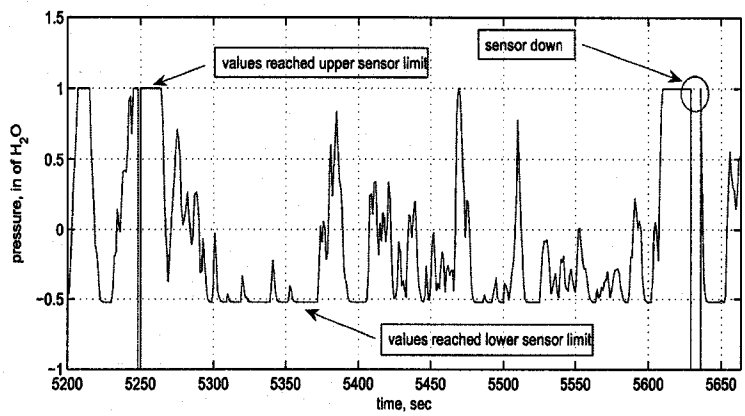


Figure 1.1: *A pressure signal with occasional censoring and bad values*

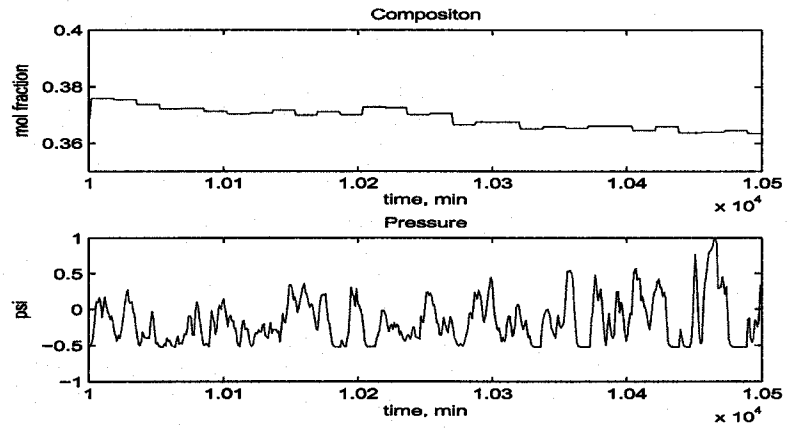


Figure 1.2: *A quality variable from an analyzer*

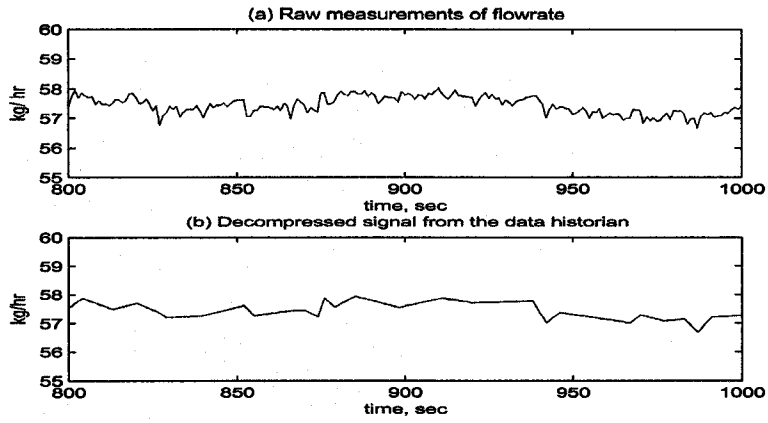


Figure 1.3: *Typical signature of compression; the decompressed signal has many linear segments*

Outliers in data: In robust analysis of data, outliers or observed values which lie far from the normal trend of the data, are often discarded. Often these missing values are filled using linear interpolation or using the adjacent data point. However, depending on the dynamics of the process these reconstructions may not be optimal.

Data Compression: Compression is encountered if archived data from the data historian are used for analysis. In order to save disk space or enhance data transmission, process data are usually compressed before archiving in the data historian. Various reconstruction methods are used to decompress such data. The typical signature of compression is shown in Figure 1.3. The reconstructed signal has many linear segments. Only the inflection points are raw values, all other points are interpolated values from these raw values. It has been shown that such reconstructed data are not even suitable for calculating basic statistics or performance index calculations (Thornhill *et al.* 2004).

Until now these problems have been viewed as separate problems and dealt with individually. The general approach to these problems is to discard the imperfect data totally (i. e., discard the complete row meaning all values in a particular time stamp even if a single value is corrupted in a row) and only work with the complete data set. Sometimes 'linear interpolation' or 'zero order hold' devices are used to give the data a complete makeup. Most of the commercial software also deal with the problem in a similar manner. The justification for such an approach is that the sampling rates in the process industries are quite high and with the improvements in instrumentation and IT infrastructure large archived data banks are available.

Therefore, one can afford to discard incomplete portions of the data and build a model using only complete data. However, in reality this may not always be possible, for example, if multiple grades are produced in a process then there is a need to divide the data according to different grades, or in time varying processes the model needs to be updated with the most recent data, or when only few laboratory measurements are available to build an inferential sensor. Moreover, it is now commonplace to use multivariate rather than univariate techniques for process monitoring and control. Deletion of the entire row or time stamp for a single missing value or only a few missing values can lead to discarding a large portion of the otherwise rich data matrix. In this context it is also important to use statistically valid methods so that the true correlation structure of data is retained. Such methods must be capable of dealing with missing values in a multivariate framework. Alternative less rigorous or *ad hoc* methods such as, straight line interpolation, zero order hold, substitution of conditional expectation etc. to reconstruct imperfect data give good results in some situations while in other cases may completely destroy the correlation structure and badly bias the estimated parameters.

In this study we propose a unified approach for dealing with this diverse set of problems. Despite the apparent differences all these causes affect the data matrix in a similar manner and fall broadly under the topic of 'Treatment of Missing Data'.

The treatment of missing data has received much attention in the statistical literature. Missing values occur mainly in statistical surveys when respondents may choose not to answer all the questions. The properties of missing data and its impact on data analysis have been studied rigorously and many formal methods have been developed to deal with this problem, for example, Expectation Maximization (EM), Data Augmentation (DA), Multiple Imputation (MI) etc. These methods are commensurate with the objectives of multivariate analysis, they conserve the correlation between the variables and have been used to solve a wide range of problems, even problems unrelated with missing data. For example, the Expectation Maximization (EM) algorithm which was developed for dealing with missing data has also been used to estimate latent variables, where the unobserved latent variables are considered as missing. In this work we have adopted ideas and techniques of missing data handling methods from statistics and surveys. However, there are major differences in the treatment of missing data between surveys and the process industry. In surveys the data collector and the analyst are two different entities and the collected data ends up in large data base. Typically the data collector has access to more information than the analyst. In filling the missing values, the data collector uses this information and provides a complete 'make-up' to the data set, so that a variety of potential users can reach cor-

rect conclusions irrespective of the analytical tools. In contrast to this, in the process industries data is automatically archived in the process historian and typically the engineer/analyst is the only user of the data. Therefore it is not important to fill the data, rather it is important to treat missing data in a way that is compatible with multivariate analysis. The main objective of the current study is to modify commonly used process monitoring methods to handle missing data in a statistically correct and meaningful way that will lead to correct inference from our analysis. Our goal in this work is to develop and propose methods that are easily understandable and have the potential of being accepted by the engineering community at large. Therefore, in developing different methods our emphasis is to keep the amount of statistical theory and derivation to minimal and use different computational based methods instead. Because of the availability of powerful computational devices in recent years there has been a renewed interest in the use of computational techniques, for example, Markov Chain Monte Carlo (MCMC) methods, various re-sampling methods such as Sequential Important Re-sampling (SIR) etc. Throughout the thesis we have used these ideas extensively for developing the new algorithms. The main areas of research are:

1.1 Missing data handling in latent variable models

In this thesis we are mainly concerned with the impact of missing data on process monitoring tools especially because these are common cause models and identified from normal operation data. It is more likely that there would be bad data during normal operation than periods when an identification experiment is being carried out. Principal Component Analysis (PCA) is the most commonly used method for process monitoring. Therefore a main focus of this study is on problems arising due to missing data in PCA based modeling. PCA deals with 'measurement errors' and 'latent variables' and thus belongs to a broader class 'measurement error latent variable models'. Besides PCA several other methods are available to deal with such class of problems. Though PCA is the most popular method in process monitoring it may not be the optimal method depending on the nature of the problem. Each of these methods has implicit or explicit assumptions and performs better over the range where the process best matches the assumptions of the method. Before extending the methods for dealing with missing data it is important to classify the methods and explore different explicit and implicit assumptions. This is essential as different assumptions require different kinds of treatment and the missing data extension of the algorithms

are also different. We also classify processes in the context of process monitoring and recommend latent variable methods that are best suited for each class. In this study several limitations of the current methods for dealing with missing data in PCA are pointed out. To overcome these limitations we extend PCA to the Data Augmentation framework for dealing with missing data. We also use similar methodology to extend Iterative PCA (IPCA) and Maximum Likelihood Factor Analysis (MLFA) for dealing with missing data. These methods have better optimality criteria and the use of these methods can be advantageous in certain situations.

We also demonstrate the application of these latent variable based missing data handling techniques in two different situations which are not apparently missing data problems. Our objective is to show that the missing data handling technique is not necessarily limited to dealing with missing values but it can also be useful for a wide variety of problems. The first example is the synchronization of the uneven length batch process data. Monitoring of batch processing is important as typically batch processes are used for producing high value added products. In batch monitoring the objective is to monitor the variation of a new batch and compare it with the average batch trajectory. Therefore data from different batches need to be included in the analysis. However, different batches may have different completion times and therefore the data lengths may be different. So there is a need for synchronizing uneven length batch data records for building an effective monitoring scheme. Synchronization of uneven length batch data sets creates some gaps in data matrices from shorter batch runs. Most of the methods currently use 'zero order hold' or 'linear interpolation' for filling these gaps. These may not be satisfactory for multivariate analysis. The objective of this study is to investigate the utility of iterative missing data technique for building multi-way PCA from such data.

1.2 Compression as a missing data problem

The other novel application of missing data handling technique is to restore the correlation structure of compressed data and then build models from such data sets. Process data are commonly archived in compressed form in the data historian. Data compression has been known to distort univariate statistics. However, the effect of compression on multivariate data analysis has not been studied so far. In this study a systematic study has been carried out on the impact of compression on multivariate data analysis including its correlation structure, the fidelity of PCA model and the quality of the model in terms of its capability to accurately represent process dynamics etc. In this study we characterize data compression from a missing data point of view

and formulate data compression as a missing data problem. We also apply an PCA based iterative missing data handling technique to restore the true correlation of compressed data set.

1.3 Pre-processing of data

In addition to dealing with missing data there are other issues that need to be addressed during the data pre-processing stage, for example, variable selection, selection of normal operating regions, accessing data quality and choosing the right data segment with appropriate excitation pattern, scaling, handling of multiple grades etc. Though often overlooked, the success of a monitoring scheme largely depends on how these issues are resolved during the preprocessing stage. We address these issues in connection with an industrial case study. This study is concerned with the analysis of process data to diagnose causes of sheet-breaks and therefore significant down times. PCA is used to model the process and a combined index based on the Hotelling T^2 and Squared Prediction Error (SPE) is developed as a sheet-break predictor. As the process is subject to external disturbances, grade changes and frequent interruptions, pre-processing of the data played an important role in getting consistent results. We use several novel techniques for data selection, scaling and modeling. The models are validated using a large validation data set with known fault conditions. The developed model, data visualization tool and engineering judgement are used for off-line diagnosis of root causes of sheet-breaks. Several operational changes were recommended and implemented on the process, resulting in significantly reduced sheet-breaks. Key Performance Indicators calculated before and after the changes show the significant economic gain as a result of this 'data-mining' project.

1.4 Missing data handling in SMC methods

Until now we have discussed the missing data and pre-processing issues of 'model-free' and 'off-line' analysis. Handling of missing data efficiently is also equally important for 'model-based' and 'on-line' analysis. On-line estimation of unmeasured states is an important problem in process industries, primarily because knowledge of such states lead to better control. Due to the advancement of computational devices there is renewed interest in methods based on sampling theory. Calculations in the sample space has several advantages, e. g., it can handle non-linearity and non-Gaussianity in the data very efficiently. However, the methods are not yet popular in process industries as the application details are missing in the literature. We investigate the

issues related to the tuning of the Particle Filter and also extend state estimation method to Multiple Imputation (MI) framework for dealing with multi-rate data.

1.5 Organization of the thesis

The dissertation is written in 'paper' or 'publication' format. Much of the material developed in this thesis have appeared in three peer reviewed conference papers and two journal papers. At the time of this thesis preparation another journal paper is in review stage. Since much of the material has been taken and edited from these publications there are some repetitions and overlaps between different chapters. However, each chapter is self contained and can be read more or less on its own. Chapter 1 provides a general introduction and describes the scope of the thesis; Chapter 2 contains the literature review of important definitions, concepts and techniques of missing data handling techniques. A majority of the references reviewed are from the field of statistics, surveys and economics. An effort has been made to explain these concepts in easily understandable terms to the engineering community. The relevance of these concepts to process data analysis has also been explored. Chapters 3 to 6 contain the new developments and contributions of this thesis. We can divide these chapters in to two parts: Part-I is on the 'model-free' analysis (i.e., no model has been identified previously) and Part-II, model-based analysis. In Part-I of the thesis, Chapters 3 through 5, the main focus is 'measurement error latent variable models'. Handling of missing data as well as the impact of the pre-processing step in general are the main topics of these chapters. In Chapter 3 we explore the characteristics of the 'measurement error latent variable models' and extend them in the Data Augmentation framework to deal with missing data. In Chapter 4 the problem of reconstruction of data from compressed data records has been dealt. Data compression has been formulated as a missing data problem and the correlation between variables are restored by using the missing data handling technique. Chapter 5 deals with different data pre-processing techniques other than missing data. This is essentially an industrial case study investigating the root causes of sheet-break. Part-II of the thesis comprises of Chapter 6 where the focus is on investigation of 'model-based' and 'on-line' methods. In this chapter different implementation issues of Sequential Importance Re-sampling (SIR) or Particle Filter are described. Particle Filters can estimate the states of non-linear and non-Gaussian systems without any approximation when the number of particles tends to infinity. However, the method is not popular in industry because the tuning of the particle filter is not easy and its implementation details are missing in the literature. We investigate several implementation issues and propose

novel techniques for tuning the Particle filter and dealing with multi-rate data in the application of Particle Filter. Finally, Chapter 7 is the concluding chapter where we summarize the main contributions of this thesis.

Chapter 2

Review of Missing Data Handling Techniques

Treatment of missing data is usually not the main focus of analysis. It is something that has to be dealt with during analysis because most data analysis or modelling methods are originally not designed to handle missing values in the data matrix. Use of *ad hoc* methods to deal with missing data may produce biased, inefficient and unreliable results. Over the last three decades a vast amount of literature has emerged in the statistical literature dealing with the treatment of missing data with applications in biostatistics, chemometrics, etc. Several important definitions and concepts have emerged to systematically analyze missing data. Commonly used methods for treating missing values include Complete Case Analysis(CCA), Available Case Analysis (ACA), Single Imputation, Multiple Imputation, Expectation Maximization(EM) and Data Augmentation (DA). The main objective of this chapter is to review these concepts and the methods for dealing with missing data and establish the link with process data analysis. Some of the relevant concepts and methods are described below:

2.1 Patterns of Missing Data

Sometimes it is beneficial to sort the columns containing missing values into orderly patterns. Historically, survey methodologists have classified different patterns of missing data encountered in surveys. For example, *unit nonresponse*, which occurs when the entire data collection procedure fails (because the sampled person is not at home, refuses to participate, etc.), *item nonresponse* which means partial data are available (e.g., the person participates but does not respond to certain individual items) are two such classifications. In longitudinal studies (e.g., for drug trials) towards the end

there are dropouts for different reasons and the collected data from subjects are of uneven length. Such data can be ordered to a monotone pattern (Little and Rubin 2002).

In process data analysis missing values are encountered because of reasons different than surveys, however the patterns can be very similar to those found in a survey. The different patterns commonly encountered in survey studies are shown graphically in Figure 2.1 where Y denote an$(N \times k)$ rectangular data set without missing value. The i-th row is denoted by $y_i = (y_{i1}, ..., y_{ik})$ and any element y_{ij} is the value of variable Y_j for subject i. Figure 2.1(a) is an example when all the missing values belong to one variable. In process industries this typically happens when the sensor breaks down for a long period of time. Figure 2.1(b) is an example of unit non-response, an analogous situation in process industries is when the process is down due to a fault condition (e.g., sheet-break in a pulp and paper mill), the only available information are the time stamps. This is especially a problem for building dynamic process model. The uneven length batch data can be arranged into the monotone pattern as shown in Figure 2.1(c), each of the columns would be a matrix of variables from different batch runs. Figure 2.1(d) is a general pattern where the short missing points may due to outliers removed for robust analysis and the long periods can be due to sensor downtime. The orderly pattern of missing values shown in Figure 2.1(e) is a unique signature of multi-rate data. The variable with missing value may be a quality variable such as, concentration which are measured less frequently in process industries. Figure 2.1(f) is a special representation of measurement error or latent variable model in missing data format where all the unobserved true values, X are considered as missing.

2.2 Mechanism for Missing Data

Missing values occur for reasons beyond our control. In statistical surveys often the data analysts do not have the information of what may have caused the data to be missing. However, for analysis purpose assumptions are made about the reason for missing data. These assumptions are usually untestable. If the assumptions are good then similar conclusions will follow from a variety of realistic alternative assumptions. Rubin (1976) laid out a probabilistic framework for the missingness mechanism and obtained the weakest condition under which it is appropriate to ignore the process that may have been the cause of missing data.

Let $Y = (y_{ij})$ denote an $(N \times k)$ rectangular data set of which some of the values are missing. Missing values are denoted by Y_{mis} and observed values are denoted by

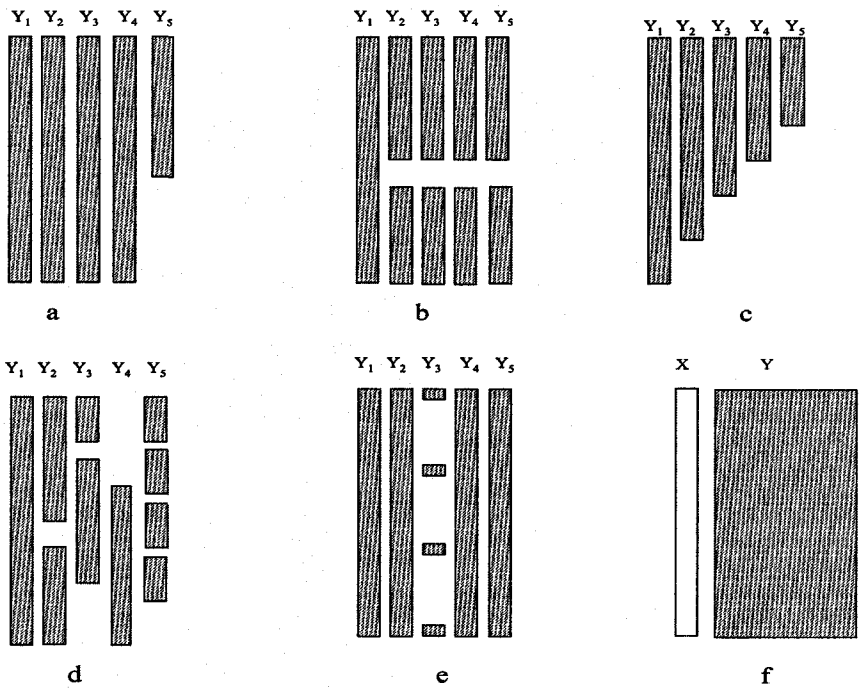


Figure 2.1: *Examples of Missing Data patterns. Rows correspond to observations*

Y_{obs}. In concise form it is represented as $Y = [Y_{obs}, Y_{mis}]$. This is a general notation used in missing data literature which is also followed in this thesis. It does not mean that observed and missing data are in two different blocks, rather missing values are distributed all over the data matrix. For any data set, a matrix $M = (m_{ij})$, referred as missingness matrix, identifies what is known and what is missing. Each element of M is usually a single binary item indicating whether y_{ij} is observed ($m_{ij} = 1$) or missing ($m_{ij} = 0$). In the statistics literature, missingness is treated as random phenomena. The distribution of M, called missingness mechanism, is characterized by the conditional distribution of M given Y, $p(M|Y, \phi)$, where ϕ denotes parameters unrelated with Y. Based on different conditionality the mechanism of missingness have been classified into three classes (Rubin 1976):

1. Missing Completely At Random (MCAR)

   In this case missingness does not depend on any part of the data Y either missing or observed.

   $$p(M|Y, \phi) = p(M|\phi)$$

   This does not mean that the pattern has to be random, rather the pattern does not depend on the values of the data matrix. Some examples from process data would be, regularly sampled multi-rate data, missing data due to sensor failure etc.

2. Missing At Random (MAR)

   This is a less restrictive assumption than MCAR and the weakest condition under which the missingness mechanism can be safely ignored while reconstructing the missing data. In this case missingness depends only on the observed component Y_{obs} and not on the missing component Y_{mis}.

   $$p(M|Y, \phi) = p(M|Y_{obs}, \phi)$$

   For example, in some cases where measurement of quality variables are costly and time consuming, the condition measurements are taken regularly and the quality variables are measured only when these condition variables indicate that the process is drifting away from the normal operating region. Thus missing values are not systematically different from observed values during normal operation and a model based on observed data can be used to estimate the quality variables.

3. Non Ignorable Mechanism (NI)

   If the mechanism of missingness is dependent on values of both the observed and the missing part of the data then the mechanism is Non Ignorable. This is the most restrictive assumption. In this case the underlying reason that caused the missing data has an effect on the inference and has to be included in any analysis. For example, in the process industry sometimes data are not recorded because they are outside the range of the sensor. These censored data may be systematically different from observed data and a model based on the observed data is often not valid in that region.

A graphical representation presented by Schafer (1997) showing the difference among the above three classes is very appealing to intuition. Here we present it (Figure 2.2) with a slight change in notation and in a more generalized form. The straight line represents the link between different components, an absence of straight line signifies that the components are unrelated. Figure 2.2(i) is a representation of MCAR showing that any missing value in j-th variable is related with observed values of k-th variable. However, the missingness mechanism or the distribution of M is not related to the values of either Y_{obs} or Y_{mis}. The distribution of M is related to a cause ϕ which is completely unrelated with Y_{obs} or Y_{mis}. The mechanisms that lead to missing values in process data are limited. In many cases it is possible to make a sound guess about the mechanisms of missingness by looking at the pattern or the signature of the data. The header information of the tags and the log books may also provide additional information about the cause that lead to missing data. Therefore, often it is not required to perform tests to classify the mechanism of missingness. Rather using logical deduction one can decide whether the missing values can be predicted directly from the observed values or the causes of missingness (i.e., ϕ) need to be included in the analysis.

2.3 Methods for Treating Missing Data

Whenever data analysts come across missing values they adopt different methods to give the data a complete look. Some of these methods have been formalized and appear in widely used statistical software. While some of them are useful, many serves only specific purpose, lack theoretical justification and may not provide sound solutions in more general circumstances. On the other hand, several methods have also been developed which have firm statistical basis and general applicability. In this section we will review all methods starting from the very basic data editing to the

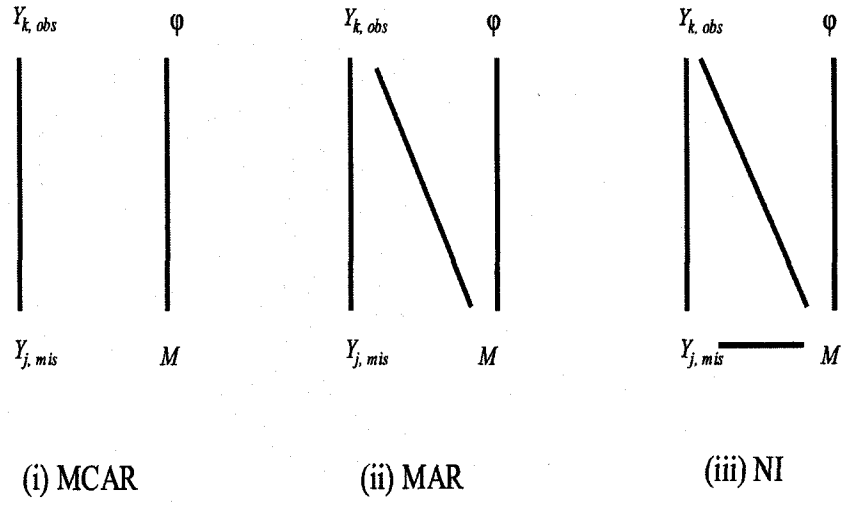


Figure 2.2: *Graphical representation of (i)missing completely at random(MCAR),(ii) missing at random(MAR) and (iii) non ignorable (NI) missingness mechanism, ϕ represents component or causes of missingness unrelated with Y*

most advanced methods.

2.3.1 Complete Case Analysis(CCA)

This is more data editing than a method itself, also known as listwise deletion. In a multivariate data matrix only those rows which do not contain a missing value are kept for analysis while the rest of the values are deleted. The advantages of this method are (i) simplicity, since standard complete data statistical analysis can be applied without modifications, (ii) provides unbiased estimate of regression parameters when data is MCAR or MAR in many cases and (iii) provides unbiased estimate of mean when data is MCAR (Allison 2002). Disadvantages stem from potential loss of information by discarding incomplete cases. This loss has two aspects, loss of precision and biasing the estimated parameters when the missing-data mechanism is not MCAR, and the complete cases are not random samples of from the data matrix rather a block of data from a particular section. The loss in efficiency can be particularly large for data sets involving a large number of variables. For example, if there are 20 variables and each variable independently has 10% chance of being missing , then the expected portion of complete cases is $0.9^{20} \doteq 0.12$. That is only about $0.12/0.9 = 13\%$ of the total data values will be retained (Schafer and Graham 2002). Process data is often dynamic in nature and listwise deletion will distort

the whole dynamic relationship. In complete case analysis the estimated variance is higher than the actual variance. This inflation can be mostly attributed to the small sample size (Little and Rubin 2002). Listwise deletion is widely used in process data analysis software.

2.3.2 Available Case Analysis(ACA)

Available Case Analysis stemmed from the idea that, one should use as much data as possible even for calculation of some of the univariate and bivariate statistics. For example, for calculating the mean and the variance of a variable all the available data of the specific variable can be used even if the corresponding measurements of the other variables may not be available. While calculating the correlation of two variables only the common values between the two variables are used. Under MCAR the Available Case Analysis gives consistent estimate of most of the univariate and the bivariate statistics. However, when considered collectively the estimated covariance matrix may not be positive definite. In regression, this creates severe problem when the covariates are highly correlated and the performance is remarkably inferior to the Complete Case Analysis (Haitovsky 1968). On the other hand, the performance of ACA is superior to CCA when the correlation between the covariates is weak (Kim and Curry 1977). In Maximum Likelihood (ML) estimate of regression parameters ACA gives better estimates even when the normality assumptions are violated (Azen and Van Guilder 1981, Little 1988).

2.3.3 Single Imputation Methods

Imputation, literally meaning attribution, is the practice of filling in missing values. These methods first fill in the missing values and give the data a complete look and then estimate the desired parameters. Sometimes these two steps are performed iteratively in order to have better estimates. Imputation is a general and flexible method for handling missing data problems. Imputation produces an apparently complete data and allows the data user to analyze using standard methods and software. When data are analyzed by multiple users, imputing once prior to all analysis ensures that same set of data is being analyzed by each analyst, which facilitates comparison of results. However, imputation can give a false sense of completeness of data and the implication can be potentially dangerous. Some *ad hoc* imputation methods can distort data distributions and relationships. Single Imputation methods can be broadly divided into two classes:

1. Explicit Modelling: Imputation models are based on formal statistical mod-

els (e.g. multivariate model), and hence the assumptions are explicit. Some popular methods of this class are:

Imputing Unconditional Mean

Missing values are replaced by the average of the observed values for that variable. This preserves the mean of the variable, but other aspects of distribution are altered with potentially serious ramifications. For example, the 95% confidence interval of the population mean of the variable y is:

$$\overline{y} \pm 1.96\frac{\sqrt{S^2}}{N}$$

where $\overline{y}$ and S^2 are the sample mean and variance and N is sample size. Mean substitution narrows this interval in two ways: by introducing a downward bias in S^2 and by overstating N. In addition to reducing variances, the method also distorts the correlation structure of the data.

Imputing Conditional Mean

Conditional mean is essentially the regression estimate of a variable based on other variables. These regressed values are used to fill the missing values. Let us consider a regression model for predicting Y from $X = (X_1, ..., X_p)$. In a general setting, values can be missing in both X and Y. If values are missing in Y the model is first fitted to the cases for which Y is known. Regression coefficients estimated from these complete cases are used to estimate the missing Y. Conditional mean imputation is nearly optimal if special corrections are made to standard errors (Schafer and Schenker 2000). If data values are missing only in the dependent variable Complete Case Analysis gives optimal estimate of the regression coefficient. However, the imputed data matrix is not suitable for analysis of variance and correlations because the method overstates the relationship between X and Y. In cases where missing values are in X as well, two approaches have been used: (i) imputing missing X by linear regression on the observed X's, regression coefficients are estimated from the complete case. For example, if X_1 is observed for m cases $(i = 1, ..., m)$ and missing for $(N - m)$ cases then the regression equation can be written in the following form:

$$E(Y_i|X_{i2}, ..., X_{ip}) = \beta_0 + \beta_1 X^*_{i1} + \sum \beta_j X_{ij}$$

where $X^*_{i1} = E(X_{i1}|X_{i2}, ..., X_{ip})$. Thus, if conditional means X^*_{i1} are substituted for missing value of X_{i1}, then Least Squares (LS) estimate on the filled-in data produces consistent estimates of the regression coefficients, assuming

MCAR; (ii) imputing missing X by linear regression on the observed X's and Y. If the partial correlation of Y and the missing X's given the observed X's is high, then better imputations can be obtained by including Y in the predictor along with the X's. However, the LS estimate of regression coefficients of Y on X's are biased. Several bias corrected methods have also been studied (Afifi and Elashoff 1966, Buck 1960). Buck (1960) imputed missing X's by regressing missing X's on the observed X's and Y, with coefficients based on the complete cases. To correct for the variance estimates the residuals estimated from regression equation were added to the sum squares of the missing variables. Later corrections were also applied to the covariance estimates (Beale and Little 1975). The corrected version of Buck's method is closely related to the normal Maximum Likelihood Estimates.

Imputing from Conditional Distribution

The distortion in the covariance structure can be minimized if a sample drawn from the conditional distribution is imputed instead of the conditional mean. In the regression equation instead of imputing $\hat{Y}$ a residual error drawn from a normal distribution with mean zero and variance estimated by the residual mean square is added to the predicted value.

$$\hat{Y}_i = \beta_0 + \beta_1 X_{i1}^* + \sum \beta_j X_{ij} + z_i$$

where $z_i \sim N(0, \sigma_i^2)$, with variance σ_i^2 estimated from the residuals of regression of Y on X based on complete cases. The addition of the random normal deviate makes the imputation a draw from the predictive distribution rather than a mean (Schafer 1983).

2. Implicit Modelling: Mostly clustering algorithms are used to find similar data sets from the observed cases and fill the missing values. This implies that the underlying model and the assumptions are implicit. Popular techniques of this class are: K Nearest Neighbor (KNN), Hot Deck Imputation etc.

   In KNN-based methods, K other samples which have similar profile (based on the observed variables) are selected. A weighted average is calculated for this missing value from the K nearest samples. The weights are the inverse of the similarity distance. Distance measures, such as, Pearson correlation, Euclidean distance, variance minimization have been used for weighting (Troyanskaya *et al.* 2001).

   Hot Deck imputation is mainly used in surveys. The data from related cases in the same survey are used to fill the missing values. To fill a missing value first

a pool of candidate values are selected. Several distance measures are used to measure the similarity, for example, maximum deviation, Mahalanobis distance etc. This pool of candidate samples are known as adjustment cell. To further select the missing value from within the cell a regression estimate of the missing value is calculated. Regression coefficients are based on complete case analysis. The value within the cell which is closest to the regression estimate is used to fill the missing value. Hot Deck differs from KNN in the sense that, it does not use an average instead it uses a sampled value of the specific variable from the adjustment cell to fill in the missing value. Several other versions of this algorithm are also available (Rubin 1973*a*, Rubin 1973*b*, Cochran and Rubin 1973, Colledge and Sande 1978).

2.3.4 Multiple Imputation

The basic idea behind multiple imputation is to assess the additional variability introduced because of imputations of missing values (Rubin 1977, Rubin 1978a). The main feature of multiple imputation is, for each missing data point several values (for instance, m samples from the conditional distribution) are imputed (Figure 2.3). Thus there would be m complete data sets. Each complete data set is analyzed using standard complete-data procedure just as if the imputed values were real data. In a survey setting this is most appropriate because the data collector and the analyst are often two different identity and the data collector has more information than those that are reported in the data base. Based on the additional information the data base constructor can think of different imputation model and use them to fill the data. So the analyst will have a chance to use all of these different sets and use it to do sensitivity analysis.

The simplest method for combining the results of m analysis is Rubin's Rule (Rubin 1987). Suppose that $\mathbf{Q}$ represents a population quantity (e.g. regression coefficient) to be estimated. Let $\hat{\mathbf{Q}}$ and $\sqrt{\mathbf{U}}$ denote the estimate of $\mathbf{Q}$ and the standard error one will use if no data were missing. The method assumes that the sample is large enough that $\frac{(\hat{\mathbf{Q}}-\mathbf{Q})}{\sqrt{\mathbf{U}}}$ has approximately a normal distribution, therefore $\hat{\mathbf{Q}} \pm 1.96\sqrt{\mathbf{U}}$ has approximately 95% coverage. In the presence of missing data, using multiple imputation (MI) m different data sets are created, subsequently there will be m different estimates of $\mathbf{Q}$ and $\mathbf{U}$, $[\hat{\mathbf{Q}}^{(j)}, \mathbf{U}^{(j)}, j = 1, ...m]$. Rubin's overall estimate is simply the average of the m estimates,

$$\bar{\mathbf{Q}} = m^{-1} \sum_{j=1}^{m} \hat{\mathbf{Q}}^{(j)} \tag{2.1}$$

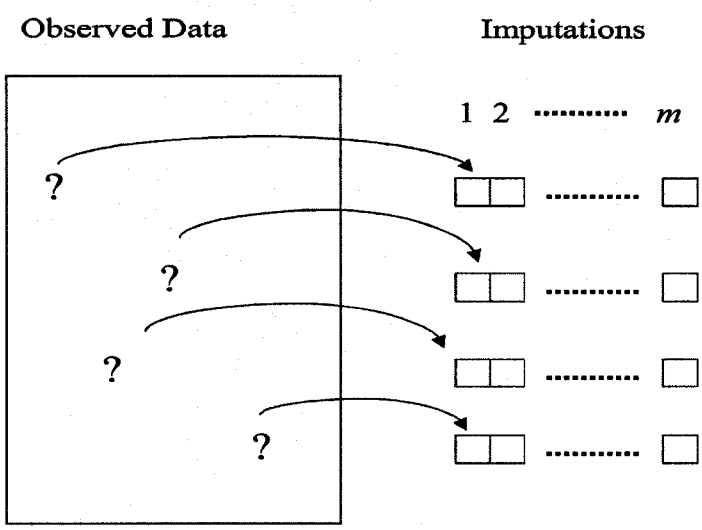


Figure 2.3: *Schematic representation of multiple imputation, where m is the number of imputation*

The uncertainty in $\bar{\mathbf{Q}}$ has two parts: the average within imputation variance,

$$\bar{\mathbf{U}} = m^{-1} \sum_{j=1}^{m} \mathbf{U}^j \tag{2.2}$$

and between-imputations variance,

$$\mathbf{B} = (m-1)^{-1} \sum_{j=1}^{m} [\hat{\mathbf{Q}}^{(j)} - \bar{\mathbf{Q}}]^2 \tag{2.3}$$

The total variance is a modified sum of the two components,

$$\mathbf{T} = \bar{\mathbf{U}} + (1 + m^{-1})\mathbf{B} \tag{2.4}$$

where the factor $(1 + m^{-1})$ is a correction for the finite number of imputations. The relative increase in variance due to nonresponse is given by the ratio $\tau = (1 + m^{-1})\mathbf{B}/\mathbf{U}$

One of the main requirement for proper multiple imputation is that the parameters used for estimating missing values should also be sampled from the distribution to reflect the uncertainty about the parameters of the model. So it is natural to motivate multiple imputation from a Bayesian perspective, where estimating the posterior distribution of a parameter is an integral part of the analysis (Schafer and Schenker 2002). As a result, it is widely accepted that multiple imputation using Bayesian

method of analysis is generally proper (Rubin 1987). However, with a variety of examples Nielsen (2003) has shown that the Bayesian method does not generally lead to proper multiple imputation and even in cases when it is proper the Bayesian method may estimate a variance which may go either way, be much higher or much smaller than the actual estimate. In response to that it has been argued that the examples were pathological cases and multiple imputation has a self correcting nature that leads to approximately valid statistical inference (Rubin 2003, Zhang 2003).

2.3.5 Expectation Maximization

The Expectation Maximization (EM) algorithm is a general method for obtaining maximum likelihood estimates of parameters in problems with incomplete data. Expectation Maximization (EM) is based on the intuitive idea of estimating the missing values and iteratively re-estimating the parameters using the estimated missing values. The origin of EM algorithm has been traced back to Fisher (1925) and McKendrick (1926). Some of the important contributions along the way were made by Hartley (1958), Baum and Weiss (1970), Orchard and Woodbury (1972), Sundberg (1974, 1976). Baum and Weiss (1970) proved the monotone convergence of EM algorithm and Sundberg (1974) provided an easily understandable theory underlying EM algorithm and illustrated using several iterative examples. However, the popularity of the method is due to the seminal paper of Dempster and Rubin (1977). The word Expectation Maximization(EM) was also coined by them. As pointed out by VanDyk and Meng (2001), this paper has two main contributions which popularized the method. First, they gave the algorithm an informative title identifying the key steps, the Expectation step (E-step) and Maximization Step (M-Step). Second, they demonstrated how it can be implemented to solve a wide class of problems. Some of them were never thought of previously, for example, Factor Analysis (VanDyk and Meng 2001). The steps of Expectation Maximization are described below. Let Y denote the complete data matrix with density $p(Y|\theta)$ where $\theta \epsilon \Re^{d\times 1}$. If Y were observed completely the objective would be to maximize the complete-data likelihood function of θ

$$L(\theta|Y) \propto p(Y|\theta) \tag{2.5}$$

In the presence of missing data, however, only part of Y, Y_{obs} is observed. In a convenient but imprecise notation we write $Y = (Y_{obs}Y_{mis})$ where Y_{mis} denotes the unobserved or missing part of the data. For simplicity we assume that data are missing at random (MAR), so that the likelihood for θ based on the observed data is,

$$L_{obs}(\theta|Y_{obs}) \propto \int p(Y_{obs}\ Y_{mis}|\theta)dY_{mis} \tag{2.6}$$

Because of the integration maximizing L_{obs} can be difficult even when maximizing L is trivial. The EM algorithm maximizes L_{obs} by maximizing the expected value of complete-data likelihood. The likelihood of the observed data increases with each iteration of the EM algorithm until converging to a local or global maximum (Dempster and Rubin 1977). The rate of convergence is directly related to the amount of unobserved information in the data matrix, i.e., convergence becomes slow with greater amount of missing data. The algorithm starts at some value of parameters, $\theta^{(t)}$ and iterates between the following two steps:

Expectation Step: In E-step we find the expectation of the logarithm of the complete-data likelihood given the observed data and the current estimate of the parameters.

$$Q(\theta|\theta^{(t)}) = \int L(\theta|Y_{obs}\ Y_{mis})p(Y_{mis}|Y_{obs},\ \theta(t))dY_{mis} \tag{2.7}$$

Maximization Step: In the M-step we find the $\theta^{(t+1)}$ to maximize $Q(\theta^{(t+1)}|\theta^{(t)})$ such that,

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta^{(t)}|\theta^{(t-1)}) for t = [1....n] \tag{2.8}$$

Although the general theory of EM applies to any model, it is particularly useful for data which comes from any exponential family of density functions. In such case the E-step reduces to finding the expected value of the sufficient statistics of the complete-data likelihood. Also if the closed form solution for the parameters are not available the M-step becomes complicated and the simplicity of the algorithm gets lost in the implementation. In the following section we explain the steps of EM algorithm with the simplest case of parameter estimation of a univariate Gaussian distribution.

Example Suppose $(y_1, y_2, ..., y_n)$ have a univariate normal distribution with mean μ and variance σ^2. We write $y = (y_{obs},\ y_{mis})$ where y represent random samples of size n, $y_{obs} = [y_1, ...y_r]$ is the set of observed values and $y_{mis} = [y_{r+1}, ...y_n]$ the missing data. The log-likelihood based on the complete-data is:

$$l(\mu, \sigma^2|y) = const - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\sum_{i=1}^{n}\frac{(y_i - \mu)^2}{\sigma^2} \tag{2.9}$$

$$= const - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\sum_{i=1}^{n}\frac{(y_i^2 - 2y_i\mu + \mu^2)}{\sigma^2} \tag{2.10}$$

$$= const - \frac{n}{2}\ln(\sigma^2) - \frac{n}{2}\frac{(\bar{y} - \mu)^2}{\sigma^2} - \frac{n}{2}\frac{s^2}{\sigma^2} \tag{2.11}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $s^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$. The sufficient statistics of the loglikelihood function is $s_1 = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $s_2 = \frac{1}{n}\sum_{i=1}^{n} y_i^2$. Log-likelihood is linear in the sufficient statistics. If all the data values are available the closed form solution for the parameters are given by, $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu})^2$. Following exactly the above description the steps of the algorithm are:

E-step: We calculate the expected value of the log-likelihood function over the distribution of the missing values.

$$Q = E(l(\mu, \sigma^2|y)) = E_{p(y_{mis}|y_{obs},[\mu(t),\sigma^2])}(const - \frac{n}{2}\ln(\sigma^2) - \frac{n}{2}\frac{(\bar{y}-\mu)^2}{\sigma^2} - \frac{n}{2}\frac{s^2}{\sigma^2}) \quad (2.12)$$

M-step: In this step we maximize the expectation of the log-likelihood with respect to the parameters.

$$\begin{aligned}
\frac{\partial Q}{\partial \mu} &= -E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\mu - \frac{1}{n}\sum_{i=1}^{n} y_i\right) = 0 \\
\Rightarrow \mu &= E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=1}^{r} y_{obs,i}\right) + E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=r+1}^{n} y_{mis,i}\right) \\
\Rightarrow \mu &= \left(\frac{1}{n}\sum_{i=1}^{r} y_{obs,i}\right) + E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=r+1}^{n} y_{mis,i}\right)
\end{aligned} \quad (2.13)$$

$$
\begin{aligned}
\frac{\partial Q}{\partial \sigma^2} &= E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(-\frac{n}{2\sigma^2}+\frac{(\bar{y}-\mu)^2}{2\sigma^4}+\frac{ns^2}{2\sigma^4}\right)=0 \\
\Rightarrow \sigma^2 &= E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(s^2\right) \\
\Rightarrow \sigma^2 &= E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2\right) \\
\Rightarrow \sigma^2 &= E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=1}^{r}(y_{obs,i}-\mu)^2\right)+E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=r+1}^{n}(y_{mis,i}-\mu)^2\right) \\
\Rightarrow \sigma^2 &= \left(\frac{1}{n}\sum_{i=1}^{r}(y_{obs,i}-\mu)^2\right)+E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=r+1}^{n}(y_{mis,i}-\mu)^2\right) \\
\Rightarrow \sigma^2 &= \left(\frac{1}{n}\sum_{i=1}^{r}(y_{obs,i}-\mu)^2\right) \\
&\quad + \frac{1}{n}\left[E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n}y_{mis,i}^2-2\mu E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n}y_{mis,i}+\mu^2\right]
\end{aligned}
\tag{2.14}
$$

Clearly it is evident from Equation 2.13 and 2.14 that calculation of the expectation of the log-likelihood is not necessary. Rather in the E-step we need to calculate the expectation of the sufficient statistics of the log-likelihood. Therefore the calculations become much simpler. First, in the E-step the expected values of the sufficient statistics are given by,

$$
E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n}y_{mis,i} = (n-r)\mu^{(t)} \tag{2.15}
$$

$$
E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n}y_{mis,i}^2 = (n-r)\mu^{(t)^2}+(n-r)\sigma^{2(t)} \tag{2.16}
$$

Substituting these expected values we obtain the estimates of the parameters in the M-step,

$$
\mu^{(t+1)} = \left(\frac{1}{n}\sum_{i=1}^{r}y_{obs,i}\right)+E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\left(\frac{1}{n}\sum_{i=r+1}^{n}y_{mis,i}\right) \tag{2.17}
$$

$$
= \frac{1}{n}\left(\sum_{i=1}^{r}y_{obs,i}+(n-r)\mu^{(t)}\right) \tag{2.18}
$$

$$(\sigma^2)^{(t+1)} = \frac{1}{n}\left(\sum_{i=1}^{r}\left(y_{obs,i} - \mu^{(t+1)}\right)^2 + E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n}\left(y_{mis,i} - \mu^{(t+1)}\right)^2\right)$$

$$\Rightarrow (\sigma^2)^{(t+1)} = \frac{1}{n}\sum_{i=1}^{r}\left(y_{obs,i} - \mu^{(t+1)}\right)^2$$

$$+\frac{1}{n}\left(E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n} y_{mis,i}^2 - 2\mu^{(t+1)}E_{p(y_{mis}|y_{obs},[\mu,\sigma^2])}\sum_{i=r+1}^{n} y_{mis,i} + (n-r)\left(\mu^{(t+1)}\right)^2\right)$$

$$\Rightarrow (\sigma^2)^{(t+1)} = \frac{1}{n}\sum_{i=1}^{r}\left(y_{obs,i} - \mu^{(t+1)}\right)^2$$

$$+\frac{1}{n}\left[(n-r)\left(\left(\mu^{(t+1)}\right)^2 + \left(\sigma^{(t)}\right)^2\right) - 2\mu^{(t+1)}(n-r)\mu^{(t+1)} + (n-r)\left(\mu^{(t+1)}\right)^2\right]$$

$$\Rightarrow (\sigma^2)^{(t+1)} = \frac{1}{n}\left[\sum_{i=1}^{r}\left(y_{obs,i} - \mu^{(t+1)}\right)^2 + (n-r)\left(\sigma^{(t)}\right)^2\right]$$

Remarks

- In the EM algorithm, 'missing data' are not directly replaced in the log likelihood function, rather expected values of the 'sufficient statistics of likelihood' are replaced in the function. Simple substitution of $\mu^{(t)}$ would lead to omission of the term $(n-r)(\sigma^2)^{(t)}$. This is the main difference between EM algorithm and other naive methods such as substitution of 'estimated missing value' and re-estimation of parameters. For each imputation of the missing value correction is also done in the covariance so that the error and covariance structures remain the same.

- If the sufficient statistics of log-likelihood function are linear in the data (e.g., multinomial distribution) E-step is simply estimation of the conditional expectation of the missing values and naive methods such as substitution of the estimated 'missing value' and re-estimation of parameters is equivalent to the EM algorithm.

EM has two major limitations: (i) in some cases with a large fraction of missing values it can be very slow to converge. (ii) cases where M-step is difficult (e.g., does not have any closed form), the theoretical simplicity of the algorithm does not convert to practical simplicity. Two types of extensions of the EM algorithm have been done to speed up the convergence. The first type which are more like the EM algorithm, retains the monotone convergence properties of EM by keeping the E-step unchanged and mostly modifying the M-step of the algorithm. The basic idea is to replace the M-step with several conditional maximization steps where a closed form for the M-step is not available. Several methods have been developed along this line. Expectation Conditional Maximization (ECM) (Meng and Rubin 1993),

Expectation Conditional Maximization Either (ECME) (Liu and Rubin 1994), Alternating Expectation Conditional Maximization (AECM) (Meng and van Dyk 1997), Parameter-Expanded EM(PX-EM) (Liu and Rubin 1994) are some of the notable extensions. The other type is based on the idea of speeding the algorithm by combining it with Newton-Raphson type updates commonly known as Hybrid EM algorithm (Jennrich and Sampson 1966, Laird and Ware 1982) However, these algorithms can be categorized under ECME as well.

2.3.6 Expectation Conditional Maximization (ECM)

ECM replaces the M-step of the EM algorithm with a series of conditional maximization steps. Like EM this also maximizes the expectation of the loglikelihood in the M-step. For cases where a closed form solution for all the parameters are not available it splits the maximization step to several conditional maximization (CM) steps. The idea is that the closed form solutions for such CM steps are easy to derive. Suppose parameter vector $\theta = (\theta_1, \theta_2, ..., \theta_s)$. The s-th CM step is to maximize expectation of the loglikelihood function with respect to θ_s keeping all other parameters, g_s (a vector containing all parameters other than θ_s) fixed at their previous estimated values. When the set of g_s is 'space-filling' in the sense of allowing unconstrained maximization over θ in its parameter space, ECM converges to a stationary point under essentially the same conditions that guarantee the convergence of EM (Little and Rubin 2002, Meng and Rubin 1993).

Example *Multivariate normal regression model with incomplete data*

Suppose we have n independent observations from the following k-variate normal model.

$$Y_i \sim N(X_i\beta, \Sigma) \tag{2.19}$$

Where X_i is a known $(k \times p)$ design matrix for the i-th observation, β is a $(p \times 1)$ vector of regression coefficients, and Σ is a $(k \times k)$ unknown variance-covariance matrix. The maximum likelihood estimation of $\theta = (\beta, \Sigma)$ is not available in closed form except for special cases when $\Sigma = \sigma^2 I$.

*E-Step:*The E-step of the algorithm is similar to the EM algorithm.

M-step is divided into two CM steps:

CM1: If Σ were known, say $\Sigma = \Sigma^{(t)}$, then the conditional maximum likelihood estimate of β would be simply the weighted least-squares estimate:

$$\beta^{(t+1)} = \{\sum_{i=1}^{n} X_i^T (\Sigma^{(t)})^{-1} X_i\}^{-1} \{\sum_{i=1}^{n} X_i^T (\Sigma^{(t)})^{-1} Y_i\} \tag{2.20}$$

CM2: Given $\beta = \beta^{(t+1)}$, the conditional maximum likelihood estimate of Σ can be directly obtained from the cross product of the residuals:

$$\Sigma^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - X_i \beta^{(t+1)})(Y_i - X_i \beta^{(t+1)})^T \tag{2.21}$$

2.3.7 Expectation Conditional Maximization Either(ECME)

ECME is an extension of ECM. Similar to ECM, E-step remains the same as Expectation Maximization. However, the M-step is substantially different from the EM or ECM. Instead of maximizing the *expectation of loglikelihood function* here the actual *loglikelihood function* is maximized in several CM steps. If an explicit expression is not available for any of these CM steps, parameters can be estimated by using optimization schemes like Newton-Rapson method. This speeds up the convergence greatly. In that case the algorithm becomes very similar to the Hybrid methods (Little and Rubin 2002).

2.3.8 Data Augmentation

The term *Data Augmentation* refers to methods for iterative optimization or sampling algorithms via the introduction of unobserved data or latent variable. In the statistics literature data augmentation was made popular by Tannner and Wong for posterior distributions of parameters (Tanner and Wong 1990). From a different perspective for applying multiple imputation to missing values a similar algorithm was developed by Li (1985). Important methodological and theoretical papers on Data Augmentation include Damien, Wakefield and Walker (1999), Higdon (1999), Mira and Tierney (1997), Neal (1997), Roberts and Rosenthal (1997) and VanDyk and Meng (2001). In the physics literature Data Augmentation is referred to as *method of auxiliary variable* (Swendsen and Wang 1987). Auxiliary variables are adopted to improve the speed of simulation, important contribution include Edwards and Sokal(1993).

If the missing data mechanism is ignorable then all the relevant information about the parameters are contained in the observed-data likelihood $L(\theta|Y_{obs})$ or observed-data posterior $P(\theta|Y_{obs})$. Except for some special cases, these tend to be complicated functions of θ, and extracting summaries like parameter estimates require special

computation tools, Data Augmentation (DA) can be very useful in this respect. The basis of Data Augmentation is Bayes' Rule for estimating joint density:

$$P(\theta, Y_{mis}|Y_{obs}) = P(Y_{mis}|Y_{obs})\, P(\theta|Y_{obs}, Y_{mis})$$

Integrating both sides over the missing data space gives the desired posterior density:

$$P(\theta|Y_{obs}) = \int_{Y_{mis}} P(Y_{mis}|Y_{obs})\, P(\theta|Y_{obs}, Y_{mis})\, dY_{mis} \tag{2.22}$$

If Y_{mis} is sampled from the posterior distribution $P(Y_{mis}|Y_{obs})$ then in discrete form Equation(2.22) can be written as,

$$P(\theta|Y_{obs}) = 1/m \sum_{i=1}^{m} P\left(\theta|Y_{mis}^{(i)}, Y_{obs}\right) \tag{2.23}$$

Similarly,

$$P(\theta, Y_{mis}|Y_{obs}) = P(\theta|Y_{obs})\, P(Y_{mis}|\theta, Y_{obs})$$

Integrating both sides over the parameter space gives the posterior density:

$$P(Y_{mis}|Y_{obs}) = \int_{\theta} P(\theta|Y_{obs})\, P(Y_{mis}|\theta, Y_{obs})\, d\theta \tag{2.24}$$

$$P(Y_{mis}|Y_{obs}) = 1/m \sum_{i=1}^{m} P\left(Y_{mis}|\theta^{(i)}, Y_{obs}\right) \tag{2.25}$$

Equation (2.23) and (2.25) suggest an iterative scheme. The key idea behind Data Augmentation is to solve the incomplete-data problem by repeatedly solving tractable complete-data problem. In Data Augmentation Y_{obs} is augmented by an assumed value of the Y_{mis}. The resulting complete-data posterior $P(\theta|Y_{obs}, Y_{mis})$ becomes much easier to handle. The solution is further improved by the iterative implementation or the following two steps:

Imputation Step: Given a current guess $\theta^{(t)}$ of the parameters, first a value of the missing data is drawn from the conditional predictive distribution of Y_{mis},

$$Y_{mis}^{t+1} \sim P\left(Y_{mis}|Y_{obs}, \theta^{t}\right) \tag{2.26}$$

Posterior Step: Conditioned on Y_{mis}^{t+1} a new value of θ is drawn from its complete-data posterior,

$$\theta^{t+1} \sim P\left(\theta|Y_{obs}, Y_{mis}^{t+1}\right) \tag{2.27}$$

Repeating the above steps from a starting value of $\theta^{(0)}$ yields a stochastic sequence $\{\theta^{(t)}, Y_{mis}^{(t)} : t = 1, 2, ...\}$ whose stationary distribution is $P(\theta, Y_{mis}|Y_{obs})$, and subsequences $\{\theta^{(t)} : t = 1, 2, ...\}$ and $\{Y_{mis}^{(t)} : t = 1, 2, ...\}$ have $P(\theta|Y_{obs})$ and $P(Y_{mis}|Y_{obs})$ as their respective stationary distribution. For a value of t that is substantially large, θ^t can be regarded as an approximate draw from $P(\theta|Y_{obs})$ and $Y_{mis}^{(t)}$ as an approximate draw from $P(Y_{mis}|Y_{obs})$. Data Augmentation may be viewed as a stochastic counterpart of Expectation Maximization where the Imputation step is similar to the Expectation step and the Posterior step is equivalent to the Maximization step of the EM algorithm (VanDyk and Meng 2001). Data Augmentation method is very closely related to an iterative method called the Gibbs' Sampler. The Gibbs' Sampler is a Markov Chain Monte Carlo (MCMC) method used to generate samples from joint distribution of a set of variables where it is difficult to sample from the joint distribution directly, but relatively easy to sample from the conditional distribution. Data Augmentation may be viewed as a Gibbs' Sampler with two parameter vectors (i. e., vector one containing model parameters, vector two containing the missing values) (Gelman and Rubin 2004). The main advantages of Data Augmentation are: it is intuitive, steps are easy to follow and implementation is easy for a wide variety of problems. The method has also good convergence property.

2.4 Concluding Remarks

The methods discussed at the beginning of this chapter (e.g., Complete Case Analysis, Available Case Analysis, Conditional and Unconditional Mean Substitution, Single Imputation) are more of historical approaches to missing data problems. Although these methods are quick fix to the problem, they are not based on a rigorous framework and tend to have an *ad hoc* character. On the other hand, Expectation Maximization (EM), Data Augmentation (DA) and Multiple Imputation (MI) are systematic approaches to the analysis of data with missing values, where inferences are based on likelihoods derived from formal statistical models for data generation (Little and Rubin 2002). Among these methods, the EM algorithm has been used extensively for building time series models from data with missing values (Shumway and Stoffer 1982). Although the EM algorithm requires implementation of two easy steps, if the closed form solution for the parameters are not available the M-step becomes complicated and the simplicity of the algorithm gets lost in the implementation. On the other hand, Data Augmentation and Multiple Imputation are computational based

methods. These methods can be applied with a wide range of model structures for building models from data containing missing values. The focus of this thesis is mainly on Data Augmentation and Multiple Imputation. In Chapter 3 latent variable models are extended to Data Augmentation framework for building models from data matrix containing missing values and Multiple Imputation is applied to update Particle filters from multi-rate data matrix in Chapter 6.

Chapter 3

Missing Data Treatment in Latent Variable Models

Modern chemical processes have become increasingly complex and are well equipped with instrumentation. Typically hundreds of variables may be monitored in a process and measurements are recorded at a high sampling rate. This results in overwhelming amount of data sets. Often there exists redundancy in the measurements leading to collinearity or the variables are correlated by the physical laws. In this context latent variable models are used extensively in process industries to eliminate the redundancy and capture the useful process information in a lower dimensional space. One of the major difficulty in applying these methods is that there may be imperfection in process data and most of these methods are not originally designed to handle missing or imperfect values in the data matrix. In the absence of such capabilities often *ad hoc* or heuristics are used to get around the problem. This can give inefficient results, which may put the reliability of the whole monitoring scheme into question. Therefore it is important to adopt a 'formal' method of treating missing data while building such models. The main objective of this study is to extend commonly used monitoring tools in a statistically valid manner, so that they can handle missing data to lead to correct inferences.

In this study we mainly restrict ourselves to analyzing latent variable models which are used for monitoring purpose. From a structural point of view such models are symmetric (i.e., data are not divided into dependent and independent variable sets). We explore different characteristics and inherent assumptions of this group of latent variable models and based on the inherent assumptions of the models, we classify these methods in different classes. This is crucial for extending the methods to effectively handle missing data. We propose algorithms to handle missing data in the following methods: (i) Principal Component Analysis (PCA) (ii) Iterative Principal

Component Analysis (IPCA) and (iii) Maximum Likelihood Factor Analysis (MLFA).

The organization of this chapter is as follows: in Section 3.1 we define the problem that is encountered in process monitoring, and classify the problem into two different classes: structural latent variable problem and functional latent variable problem. In Section 3.2 we explore the characteristics of some of the symmetrical latent variable models. Next in Section 3.3.1 we provide a brief overview on PCA with emphasis on different methods for handling missing values in a data matrix and point out some of the limitations of the currently used methods in handling missing values. Iterative Principal Component Analysis (IPCA) is introduced in Section 3.3.2 for dealing with 'Functional Latent Variable Problems' and the method has been extended in the Data Augmentation (DA) framework for handling missing data. In Section 3.4.1 we introduce the Maximum Likelihood Factor Analysis for dealing with 'Structural Latent Variable Problems' and extend the method in a Data Augmentation framework for handling missing values in data matrix. Throughout this study it has been assumed that data is Missing Completely At Random (MCAR) or Missing At Random (MAR). Treatment of missing data with Non Ignorable (NI) mechanism is beyond the scope of this study. In Section 3.5 the implementation and performance of the methods are demonstrated using a simulated flownetwork system. Finally, in Section 3.6 we give a novel application of the missing data handling technique, we apply a PCA based missing data handling technique for synchronizing uneven length batch process data.

3.1 Problem definition

Consider a measurement matrix $Y \epsilon \Re^{N \times n}$ where N is the number of samples and n represents the number of variables. The measurement at sampling instant i, $y_i^{1 \times n}$ can be decomposed as follows:

$$y_i = x_i + \varepsilon_i \tag{3.1}$$

where ε_i is the measurement error and x_i is the noise free true variable. For building latent variable model, measurements from a particular section or unit are collected in a data matrix. After collecting N samples we can write it in the following matrix form:

$$Y = X + \varepsilon \tag{3.2}$$

where $X \epsilon \Re^{N \times n}$ is the noise free true values and $\varepsilon \epsilon \Re^{N \times n}$ is the measurement error matrix. These are routine operational data and represent the normal variation in the

process. Process measurements usually demonstrate strong correlation under normal conditions arising from physical relationships or instrumentation redundancy. The correlation between the measurements provides necessary redundancy to detect and identify any fault in the process. However, these constraints are hidden under the noise and not visible apparently. As a result though the data matrix Y may appear as full rank, the true signal part of the measurement, X is likely to be rank deficient in most cases. The relationships between the variables can be expressed in two ways:

i)Linear factor form

$$X = TP^T$$

where $T \epsilon \Re^{N \times m}$ contains the uncorrelated latent variables and $P \epsilon \Re^{n \times m}$ is the loading matrix, which contains the basis vectors of the lower dimensional space. The rank of matrix T and P are lower than the rank of the noisy measurements Y (i.e, $m < n$).

ii)Linear relation form

$$AX^T = 0$$

where A is the constraint equation.

So the modelling problem is essentially the estimation of loading matrix P or the constraint equation A from the training data set. Either forms of these models can be used for detection and isolation of faults. Using loading matrix P, the test data is projected onto the lower dimensional space. The residuals of a test sample are calculated as follows:

$$r_i = y_i - \hat{x}_i \tag{3.3}$$

$$= y_i - y_i \hat{P} \hat{P}^T \tag{3.4}$$

Residuals in a lower dimensional space can also be calculated using the following equation:

$$r_i = A y_i^T \tag{3.5}$$

Once the residuals have been calculated, various statistical tests are performed on these residuals to detect and isolate faults in the process.

3.1.1 Characterization of Error

The measurement noise ε is an additive noise due to the inaccuracy in the measuring devices. This is inherent to the measuring device and the characteristics of the measurement error do not change from observation to observation. Therefore the measurement errors at different sampling events belong to the same distribution. For

characterization purpose errors are often assumed to be normally distributed, for example, zero mean multivariate normal $\varepsilon_i \sim N(0, \Omega_\varepsilon)$; $i = [1, ..., N]$. The covariance of the measurement error is given by,

$$\Omega_\varepsilon = cov(\varepsilon_i) \tag{3.6}$$

$$= E(\varepsilon_i^T \varepsilon_i) \tag{3.7}$$

where $E(.)$ represents the expectation.

The structure of the covariance matrix will depend on the correlation between the measurement errors in the variable direction. We assume that the structure of the covariance matrix, Ω_ε is diagonal with unequal diagonal elements. Since we are dealing with steady state modeling, measurement errors are usually uncorrelated with each other. The only exception is if some derived variables are used, for example, enthalpy instead of flowrate and temperature, then the errors of enthalpy will be correlated with flowrate and temperature. In that case, there is a good chance that flowrate and temperature may not be included in the model. However, measurements come from different sensors, for example, flowmeter, thermocouple, level sensor etc. Each sensor works on different principles, measures different quantity and has its own precision level. It is highly unlikely that the errors of the measurements will have equal variances. Therefore the diagonal elements of the covariance matrix Ω_ε is assumed unequal. This structure is sufficiently general to capture the behavior of measurement errors from most steady state processes.

3.1.2 Characterization of Underlying Signal

The assumptions regarding the true signal give rise to two different problems which are well known in statistical literature. If we assume that x_i follows a multivariate normal distribution with mean μ and covariance Σ_x then we obtain what is referred to as a Structural Latent Variable Problem. On the other hand if we make a less restrictive assumption that the true values of the variables follow an arbitrary deterministic sequence then we obtain a Functional Latent Variable Problem (Fuller 1987). Noise-free signals from chemical processes are in general deterministic, for example, if we consider that x_i represents different steady state operating points then it is more appropriate to use a functional model. However, in many circumstances the behavior of the signal can be approximated very well by a structural model. Since for monitoring purposes normal operation data are used for building the model, the excitation is due to random disturbances entering into the process and the measured signals behave like filtered random signals. This is illustrated in Figure 3.1 where a random

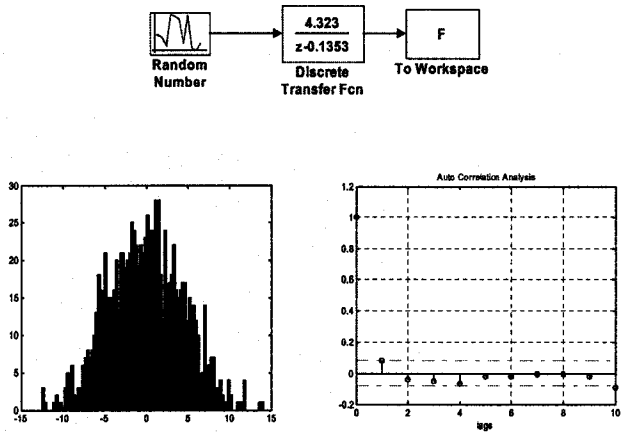


Figure 3.1: *A filtered random noise behaving as a structural signal*

Gaussian signal passes through an autoregressive process. The output signal has a Gaussian Distribution and the correlation between the samples is also minimal. The degree of correlation will depend on the coefficients of the auto regressive process. The point to note here is that, in a chemical process we can also expect Gaussian distribution of the noise free signals in some situations. So, solutions of both the structural and the functional problems are useful for process monitoring.

3.2 Classification of Latent Variable Models

Principal Component Analysis (PCA) is the most common latent variable method used for process monitoring. Besides PCA there are other similar methods which can extract the latent structure from the data. Several modifications of PCA have also been developed to attain better optimality criteria and solve specific problems. Some of the popular latent variable methods are Maximum Likelihood Factor Analysis (MLFA), Bayesian PCA (BPCA), Iterative PCA (IPCA), Probabilistic PCA (PPCA). Based on the discussions in the previous section we classify these methods into different groups. The classification of the methods and their salient properties are summarized in Table 3.1. The classification of methods under this framework is important for extending the methods for dealing with missing data, since structural and functional models need different kind of treatment.

From the above list of latent variable models we choose to study the impact of

Method	Optimality	Underlying Signal	Noise Structure	Scaling	Comment
PCA	Least Squares	Functional	Isotropic Noise	Dependent on Scaling	Data requirement is small.
BPCA	Least Squares	Structural	Isotropic Noise	Dependent on Scaling	Data requirement is small. Prior information is required.
PPCA	Least Squares	Structural	Isotropic Noise	Dependent on Scaling	
IPCA	Weighted Least Squares	Functional	Non-Isotropic Noise	Standard deviations of measurement noise are used for scaling. Scaling Invariant	Estimates the variance of measurement error from the data. Data requirement is moderate.
MLPCA	Maximum Likelihood	Functional	Very general noise structure with correlation in both time and variable direction.	Standard deviations of measurement noise are used for scaling. Scaling Invariant	Requires the knowledge of covariance of measurement error.
MLFA	Maximum Likelihood	Structural	Non-Isotropic noise	Standard deviations of measurement noise are used for scaling. Scaling Invariant	Estimates the variance of measurement error from the data. Data requirement is very high.

Table 3.1: *Salient properties of latent variable models*

missing data in few representative models. We choose Principal Component Analysis (PCA) and Iterative Principal Component Analysis (IPCA) as representative cases from the functional class and Maximum Likelihood Factor Analysis (MLFA) as a representative case from the structural class of models. We propose modifications of these algorithms based on Data Augmentation and Bootstrap methods to deal with missing data.

3.3 Functional Latent Variable Model

3.3.1 Principal Component Analysis

Review of PCA

PCA is widely used to build models from a large number of highly correlated variables. In process monitoring, PCA is extensively used for tracking operating performance and detecting impending faults. Extensive review on the theory and practice of PCA is available in Jackson (1991). In PCA it is assumed that the noise free signals, X can be expressed as a product of two factors.

$$X = TP^T$$

where T is the score, and P is the loading matrix both have rank, r and $r < n$. The objective of PCA is to get the solution of T and P by minimizing the sum of squared errors, E.

$$E = tr((Y - TP^T)^T(Y - TP^T))$$

However, this leaves indeterminacy in the estimation of T and P. A restriction is placed on the length of P (i.e. $PP^T = I$) to make it an identifiable problem and the modified objective function is given by,

$$L = tr((Y - TP^T)^T(Y - TP^T)) + \mu\, tr((P^TP - I))$$

The minimization of the objective function leads to an eigenvector solution and the error is minimized when P contains the eigenvectors corresponding to the significant eigenvalues of Y^TY and also from definition $T = YP$.

This solution is closely related to the *Singular Value Decomposition* (SVD) of the matrix Y. Using SVD the data matrix Y of dimension $N \times n$ can be decomposed to the following form:

$$Y = U\Lambda V^T \tag{3.8}$$

where $UU^T = VV^T = I_n$ and Λ is a positive definite diagonal matrix. Equation 3.8 implies that $Y^TY = V\Lambda^2V^T$, and finding U, Λ and V can be seen as an eigenvalue problem. The columns of V are the eigenvectors and the diagonal matrix Λ^2 will contain the eigenvalues in descending order, $\lambda_1 \geq \lambda_2 \geq \lambda_3... \geq \lambda_r \geq ....\lambda_n$. If the noise-free signal X has rank $r < n$, the eigenvalues $\lambda_{r+1}....\lambda_n$ will be substantially smaller than the first r eigenvalues and the variance explained by these eigenvalues can be attributed to the error E. Also the first r columns of V give the loading matrix P, and the first r eigenvalues explain the variance of the noise free signal X.

Missing Data Handling in PCA

Missing values in the data matrix pose difficulties at two stages in PCA based process monitoring. Firstly, in building a model from the historical data sets. Secondly, during the monitoring phase for calculating the scores and the residuals. Methods developed for score calculation in the presence of missing values include Trimmed Score (TRI), Single Component Projection (SCP), Conditional Mean Replacement (CMR), Projection to Model Plane (PMP). These are all single step methods and essentially the implementation of Equation 3.12 (Nelson and MacGregor 1996, Artega and Ferrer 2002). In this study we focus only on the issues that arise during the off-line modelling stage due to the presence of missing values in the data matrix. A brief review of these methods is in order.

Originally the NIPALS algorithm was used for building principal component models. Christofferson (1970) extended the NIPALS algorithm for finding first and second principal components in the presence of missing values in the data matrix. The method has been generalized for finding multiple PCs in the presence of missing data (Grung and Manne 1998). It uses a least square minimization criteria to estimate the scores and the loadings in successive steps. Let Y be a full matrix and Z is the known part of Y. In matrix Z the missing values are represented by zeroes. The relation between Z and Y can be conveniently expressed with the help of missingness indicator M which has same dimensions as Y with elements $m_{ij} = 1$ for y_{ij} known, and $m_{ij} = 0$ for missing y_{ij}. Consequently the relation between Z and Y is $z_{ij} = m_{ij}y_{ij}$. The objective function may be written in the following form:

$$F = \sum_{i=1}^{N}\sum_{j=1}^{n} m_{ij}(y_{ij} - \sum_{k=1}^{A} t_{ik}p_{jk})^2 = \sum_{i=1}^{N}\sum_{j=1}^{n}(z_{ij} - \sum_{k=1}^{A} t_{ik}m_{ij}p_{jk})^2 \tag{3.9}$$

For the i-th row the objective function is given by,

$$F_i = \sum_{j}(z_{ij} - \sum_{k=1} t_{ik}m_{ij}p_{jk})^2 \tag{3.10}$$

Defining $A^{(i)}$ with elements $a_{jk}^{(i)} = m_{ij}p_{jk}$; $z^{(i)}$ as the i-th row of Z; and $t^{(i)}$ as the i-th row of T, Equation(3.10) can be written in the following form:

$$F_i = \sum_j (z_{ij} - \sum_k t_{ik} a_{jk}^{(i)})^2 \tag{3.11}$$

If elements of the loadings matrix, p_{ij} are known the solution to the regression problem is:

$$t^{(i)} = z^{(i)} A^{(i)} \left(A^{(i)^T} A^{(i)} \right)^{-1} \tag{3.12}$$

Similarly, defining $z^{(j)}$ as the j-th column of Z containing elements z_{ij}; $p^{(j)}$ as j-th column of P containing elements p_{ij} and the matrix $B^{(j)}$ with the elements $b_{ik}^{(j)} = t_{ik}m_{ij}$ the loadings matrix can be found by ordinary least-squares regression as:

$$p^{(j)} = \left(B^{(j)^T} B^{(j)} \right)^{-1} B^{(j)^T} z^{(j)} \tag{3.13}$$

The algorithm is an iterative implementation of Equation 3.12 and Equation 3.13. However, the estimated scores are not orthogonal to each other. To get orthogonal scores after convergence of the algorithm, SVD is performed on the data matrix with imputed values.

Now it has become more customary to use the Singular Value Decomposition (SVD) algorithm to extract the PCs. In the presence of missing values in the data matrix, an iterative imputation approach is used to fill the missing values and estimate the PCs. In this study we refer to this algorithm as Principal Component Analysis Imputation Algorithm (PCAIA) (Grung and Manne 1998, Troyanskaya *et al.* 2001, Walczak and Massart 2001). The algorithm is described below:

1. Initially missing values of the data matrix are filled with the unconditional mean of the variables. For example, missing values are filled by the column averages of Y_{obs}, which gives the augmented data matrix $Y_{aug} = [Y_{obs}, \hat{Y}_{mis}]$ where $\hat{Y}_{mis} = mean(Y_{obs})$.

2. Singular Value Decomposition (SVD) is performed on the augmented data matrix. The loading matrix, $\hat{P}$ is used to predict the noise free values $\hat{X} = Y_{aug}\hat{P}\hat{P}^T$.

3. Missing values are filled with predicted $\hat{X}$, and the augmented data matrix, $Y_{aug} = [Y_{obs}, \hat{X}_{mis}]$.

4. Convergence is monitored by observing sum of squared errors of the observed values and corresponding predicted values from step (2).

$$SSE_{obs} = \sum_{i=1}^{N}\sum_{j=1}^{n}(y_{ij} - \hat{x}_{ij})^2_{obs}$$

Step (2) and step(3) are repeated until convergence.

Treatment of missing data based on the NIPALS algorithm and SVD essentially minimize the same least squared objective function as in Equation 3.9. However, the NIPALS algorithm converges at a faster rate than the iterative imputation method. Some of the limitations of PCAIA are discussed below. Methods based on the NIPALS algorithm also suffer from similar limitations in the presence of missing data.

Limitations of PCAIA

Distortion of Covariance Structure

Measurements have two parts: a deterministic part, X and a random noise, ε. Therefore, the covariance of the measurement matrix Y can be divided into two parts, $\Sigma_Y = \Sigma_X + \Omega_\varepsilon$. In filling the missing values the method ignores the random error part of Y. Missing values are filled by the conditional expectation of the missing values, $\hat{X}$. As a result Ω_ε is underestimated and the estimate of Σ_Y from such imputed data matrix gets distorted. The degree of distortion will depend on the percentage of missing data and the relative magnitude of the measurement error. Also this type of imputations over emphasize the linear relationships between the variables, therefore the imputed data set is not suitable for analysis of variance.

Model Order Selection

The rank of the loading matrix P or equivalently the number of major PCs in the model is known as the order of the model. The loadings are given by the eigenvectors, and the corresponding eigenvalues indicate the variances explained by the eigenvectors. Ideally one would like to include the minimum number of eigenvectors necessary to explain the total variance of the deterministic part, X. Methods commonly used for model order selection are, SCREE-plot, Broken root, Cross validation, Significance test etc. In selecting the number of PCs, except for cross validation all these methods make use of the ratio of the variance explained by major PCs to the total variance explained by all PCs.

$$\%\ Variance\ Explained\ by\ Major\ PCs = \frac{\lambda_1 + \cdots + \lambda r}{\lambda_1 + \cdots + \lambda r + \cdots + \lambda n} \qquad (3.14)$$

Once the ratio has been calculated, the user decides on how much of the variance needs to be attributed to X. The number of PCs necessary to capture the specified variance information will determine the model order.

Distortion of the covariance matrix has a direct impact on the selection of model order. The error variances are attenuated because of missing values in the data matrix which leads to the shrinkage of the denominator term of the ratio defined in Equation 3.14 . Therefore the percentage of total variance explained by major PCs will no longer remain constant for a data set, rather it will depend on the fraction of missing values present in the data matrix.

Extending PCA to the Data Augmentation Framework

The limitations of Principal Component Analysis Iterative Algorithm (PCAIA) arise due to the fact that the missing values are replaced with the conditional expected values of the missing values, i.e., while imputing the missing values the errors in the measurements are ignored. Depending on the magnitude of the measurement errors the covariance matrix of the imputed data set may get distorted from the covariance matrix of the original data set. Therefore it is important to take the measurement error into consideration during the imputation phase. In this section we propose an algorithm which combines PCAIA with the ideas of Bootstrap re-sampling and Data Augmentation strategies. The proposed algorithm is named PCA-Data Augmentation (PCADA).

The basic idea is to iteratively implement the imputation and posterior steps described by Equation (2.26) and (2.27) as discussed in Section 2.3.8. Suppose that at the i-th iterative step, the available data set is Y_{obs} and the missing values are randomly distributed throughout the data matrix. The data set can be given a complete makeup with some initial estimates of the missing values, e.g., filling the missing values with the mean of the observed values $\hat{Y}_{mis} \Leftarrow mean(Y_{obs})$. The complete data set is given as, $Y_{aug} = [Y_{obs} \;\; \hat{Y}_{mis}]$. The parameters or the loading matrix, $\hat{P}$ can be calculated by applying Singular Value Decomposition (SVD) on the augmented data matrix Y_{aug}. After the initial estimation, the imputation and the posterior steps are carried out as follows:

Imputation Step

Imputation step requires that the missing values are sampled from the distribution conditioned on the observed values and the parameters. Using the estimate of the loading matrix, $\hat{P}$ and augmented data matrix, Y_{aug} conditional expectations of the measurements are calculated by the following equation:

$$\hat{X} = Y_{aug}\hat{P}\hat{P}^T \tag{3.15}$$

The differences between the observed measurements and the corresponding estimated values of X give the residuals:

$$r_{ij} = y_{ij}^{obs} - \hat{x}_{ij}^{obs} \tag{3.16}$$

These residuals are collected in a matrix to form the residual matrix r. A residual term sampled from the residual matrix is added with each expected value of the missing data points. The imputation values for the missing data points are given by the following expression:

$$\hat{y}_{ij}^{mis} = \hat{x}_{ij}^{mis} + r_{kj} \tag{3.17}$$

where k is a random integer number between 1 and N and r_{kj} is a residual term sampled randomly from the j-th column of the residual matrix r. These estimated values are used to impute the missing values and the augmented data matrix is given by, $Y_{aug} = [Y_{obs}, \hat{Y}_{mis}]$.

Posterior Step

Model parameters from their posterior distributions are sampled at this stage. A 'Bootstrap' re-sampling technique is used to create the posterior distributions of the model parameters. The parameters in this case are the elements of the loading matrix P.

Let us assume that after imputing the missing values, the completed data matrix is Y_{aug}. Using the 'Bootstrap' re-sampling method, J Bootstrap data sets $Y^* = [Y^{(1)}, Y^{(2)}, \ldots\ldots Y^{(J)}]$ are created from the augmented data matrix. Singular Value Decomposition (SVD) is performed on each of these data sets, which results in a series of model parameters (i.e., loadings matrix, $P^* = [P^{(1)}\ P^{(2)}\ \ldots\ldots P^{(J)}]$). Averages of the estimated model parameters are given by:

$$\hat{P} = \frac{1}{J}\Sigma_{i=1}^{J} P^i \tag{3.18}$$

In the next iterative step the estimated loading matrix, $\hat{P}$ is used to calculate the conditional expectation of the missing values. The Imputation step and the Posterior step are repeated alternatively until convergence. Convergence is monitored by observing sum of squared errors between the observed values and corresponding predicted values.

$$SSE_{obs} = \sum_{i=1}^{N}\sum_{j=1}^{n}(y_{ij} - \hat{x}_{ij})_{obs}^2$$

3.3.2 Iterative Principal Component Analysis

Iterative Principal Component Analysis (IPCA) is developed for solving the following functional latent variable problem. Suppose that we have a data matrix Y which can be decomposed as follows:

$$Y = X + \varepsilon \tag{3.19}$$

where the noise free signal X is deterministic, measurement error is multivariate normal $\varepsilon_i \sim N(0, \Omega_\varepsilon)$ and the constraints between the noise-free measurements is expressed in the following principal relations form:

$$XA = 0$$

The objective of IPCA is to simultaneously estimate the error covariance matrix Ω_ε and identify the constrained model A. The algorithm is motivated from the fact that PCA minimizes a least squares objective function and therefore is not optimal in the maximum likelihood sense for non-isotropic measurement errors (Narasimhan and Shah 2004). Maximum likelihood estimation is often a desired optimality criteria because it provides minimum variance estimates of parameters as sample size tends to infinity. However, it has been proved that even in the case of univariate regression for the functional model problem, the maximum likelihood procedure for simultaneously estimating the model and the error variances gives unbounded solution (Anderson 1984).

Thus, several modified maximum likelihood methods have been developed to solve this problem (Chan and Mak 1985). However, none of these methods can achieve maximum likelihood estimates if there is no *a priori* knowledge of the error covariance matrix available. For known error covariance matrix Wentzell *et al.* (1997) developed a maximum likelihood estimation method in the PCA framework, called MLPCA to estimate the noise-free signal X. The algorithm is based on an alternative regression strategy in combination with singular value decomposition. MLPCA minimizes the following objective function:

$$\mathbf{S} = \Sigma_{i=1}^{N}(y_i - x_i)\Omega_\varepsilon^{-1}(y_i - x_i)' \tag{3.20}$$

For known Ω_ε minimization of the above objective function maximizes the log likelihood function. The method has been used in different chemometrics application. In chemometric data often replicates of the measurements are available and the variances of measurement errors can be independently estimated. On the other hand, in process industries such information is rarely available. In this context IPCA

is very relevant since it simultaneously estimates the model and the error covariance matrix. The method minimizes a weighted least squares criteria and it has been shown that for functional model with non-isotropic measurement errors IPCA gives better estimates than PCA.

IPCA combines PCA with an optimization procedure for estimating the error covariance matrix. IPCA has two main steps: (i) optimal scaling of measurements and (ii) estimation of the error covariance matrix. The method iteratively alternates between these two steps until convergence. An optimal scaling strategy is used in IPCA. It was shown that PCA is scaling invariant with this scaling scheme. Let scaling factor L be defined by,

$$LL^T = \Omega_\varepsilon$$

After scaling the measurements with the scaling matrix, the transformed measurements are given as follows:

$$y_i^s = x_i L^{-1} + \varepsilon_i L^{-1}$$

With the assumption that measurement errors are uncorrelated with underlying signals the covariance of the scaled matrix is given by,

$$\Sigma_{y_s} = S_{x_s} + I$$

This is an important result as it provides a convenient way to select the order of the model. According to the *eigenvalue* shift theorem, eigenvectors of the covariance matrix of noise corrupted scaled data, Σ_{y_s} are equal to the eigenvectors of the covariance matrix of scaled noise-free data, S_{x_s}. There is no distortion in the eigenvectors due to the presence of noise in the signals. In addition to that, eigenvalues of the covariance matrix of noise corrupted data are shifted by unity from the eigenvalues of the covariance matrix of noise-free data. This property is conveniently used for selecting the order of the model. For example, if the rank of the data matrix X_s is m, the last $(n\text{-}m)$ eigenvalues of S_{x_s} will be exactly zero and the last $(n\text{-}m)$ eigenvalues of Σ_{y_s} will be unity. The eigenvectors corresponding to these unity eigenvalues define the basis vectors of the residual space, A_s which in this case is the constraint model in scaled domain. Therefore, it provides a definitive way of selecting the model order. The constraint model in the original domain is simply given by,

$$A = LA_s$$

However, in most of the cases the error covariance matrix Ω_ε is not available and the following iterative technique is used for simultaneous estimation of A and Ω_ε.

An initial estimate of the constraint model, A^0 is obtained by ordinary PCA on the unscaled data matrix Y. This initial estimated model is used to calculate the residuals, $r_i = y_i A^0$. If the estimated model is exact, the residuals will be independent in the sample direction and may be assumed normally distributed with zero mean and covariance matrix $\Omega_r = \hat{A}^0 \Omega_\varepsilon (\hat{A}^0)^T$. Thus, the joint density function of $[r_1^T \; r_2^T \; r_3^T \; \cdots \; r_N^T]^T$ is easily obtained, and Ω_ε is estimated by maximizing the log likelihood function, which is equivalent to minimizing the following objective function:

$$\min_{\Omega_\varepsilon} N log \left| \left(\hat{A}^0\right)^T \Omega_\varepsilon \hat{A}^0 \right| + \sum_{i=1}^{N} \left(r_i \left(\left(\hat{A}^0\right)^T \Omega_\varepsilon \hat{A}^0 \right)^{-1} r_i^T \right) \tag{3.21}$$

where r_i^T, the residuals of $i-th$ observation and $i=1,\ldots.N$.

In the next step, data is scaled using the estimated error covariance. The constraint model is estimated by applying Singular Value Decomposition (SVD) on the scaled data matrix. The estimated model will be closer to the true model in each successive step. The maximum number of elements in Ω_ε that can be estimated is restricted by the number of constraints or the rank of A. If rank of A is m then a maximum of $m(m+1)/2$ elements can be estimated.

Missing data handling in IPCA

IPCA is combined with the Bootstrap and Data Augmentation(DA) techniques for dealing with missing data in the data matrix. The main objective of the proposed algorithm is to take measurement errors in missing values into consideration during the imputation of the missing values. An intuitive way is to add an error term scaled to the variances of the measurement error with the conditional expected values and use them for imputation. However, addition of the errors causes divergence in the iterative algorithm. Extension in Data Augmentation (DA) framework gives the iterative algorithm good convergence property. The algorithm is termed as IPCA Data Augmentation (IPCADA). The basic idea is to implement the imputation step and posterior step described by Equation (2.26) and (2.27).

Imputation Step (I-Step)

The Imputation step requires that the missing values be sampled from the distribution conditioned on the observed values and the current estimates of the parameters. This can be conveniently done in IPCA since the error covariance is also simultaneously estimated in the procedure. Similar to the stochastic regression a scaled error is added with the conditional expected values. However, in this case the regressors are also corrupted with measurement noise. Therefore, the ordinary least squares method is not suitable for estimation. Instead a total least squares method (i.e., IPCA) is

used to calculate the conditional expectation of the missing values. The procedure is explained below:

Using the previous notations, at any time step t an estimate of the loading matrix, $\hat{P}$ and error covariance, $\hat{\Omega}_\varepsilon$ and augmented data matrix Y_{aug} are available. The conditional expected value is given by,

$$\hat{X} = Y_{aug}\hat{P}\hat{P}^T \tag{3.22}$$

A scaled error term is added to the expected values:

$$\hat{Y} = \hat{X} + L\nu \tag{3.23}$$

where $\nu_i \sim N(0, I)$ and $LL^T = \hat{\Omega}_\varepsilon$. This is equivalent to drawing samples from the predictive distribution. The factorization of the error covariance matrix can be carried out by LU decomposition. If the error covariance matrix is diagonal, then the scaling factor is given simply by the square root of the matrix. Missing values are filled using the corresponding values of $\hat{Y}$ and the augmented data matrix is given as, $Y_{aug} = \left[Y_{obs}, \hat{Y}_{mis}\right]$

Posterior Step

At this step model parameters are sampled from their posterior distribution. The 'Bootstrap' re-sampling technique is used to create the posterior distribution of the model parameters. The parameters in this case are the elements of the loading matrix P. Let us assume that the scaled data matrix is Y_s. Using 'Bootstrap' m re-sampled data matrix $Y_s^{(1)}, Y_s^{(2)}, \ldots\ldots Y_s^{(m)}$ are created, and SVD is performed on each of these data sets, which results in a series of model parameters (i.e., loadings matrix, $P_s^{(1)}P_s^{(2)}\ldots\ldots P_s^{(m)}$). Average estimates of the model parameters are given by,

$$\hat{P}_s = \frac{1}{m}\Sigma_{i=1}^{m} P_s^i \tag{3.24}$$

The estimated average loading matrix in the scaled domain, $\hat{P}_s$ is used to calculate the conditional expected values of the missing data in the imputation step. The steps of the proposed algorithm for dealing with missing data are described below:

1. Initially the missing values of the data matrix are filled with the unconditional mean of the variables. For example, missing values are filled by the column averages of Y_{obs} and augmented data matrix, $Y_{aug} = [Y_{obs}, mean(Y_{obs})]$.

2. The filled data matrix Y_{aug} is supplied to the IPCA algorithm. IPCA automatically determines the number of significant principal components and gives

an estimate of the constraint model A and scaling matrix L, where $LL^T = \Omega_\varepsilon$. Data is scaled using the scaling matrix.

3. Using Bootstrap several re-sampled data matrices are created. SVD is performed on each of the re-sampled data matrices to calculate the loading matrix. The variability of the data gets transmitted to the loading matrices, therefore the calculated loadings are the sampled parameters from the distribution. The average of the parameters are used to estimate the missing values. The noise-free variables in scaled domain $\hat{X}_s$ is given by,

$$\hat{X}_s = \hat{P}_s \hat{P}_s^T Y_{aug}^s$$

4. The scaled noise free variables $\hat{X}_s$ are converted to $\hat{X}$ in the original domain, $\hat{X} = \hat{X}_s L$ and a scaled noise is added with the predicted $\hat{X}$.

$$\hat{Y} = \hat{X} + L\nu$$

where $\nu_i \sim N(0, I)$. Missing values in the data matrix are filled with these predicted $\hat{Y}$ values.

5. Convergence is monitored by observing sum of squared errors between the observed values and corresponding predicted values from step (4).

$$SSE_{obs} = \sum_{i=1}^{N} \sum_{j=1}^{n} (y_{ij} - \hat{x}_{ij})_{obs}^2$$

6. Steps (2) to (5) are repeated until convergence.

3.4 Structural Latent Variable Model

3.4.1 Maximum Likelihood Factor Analysis

Structural modelling has been extensively studied in the area of statistics, psychology and econometrics; and Factor Analysis (FA) is used to model such processes. We feel that these methods have not been used to their full potential in the area of chemical engineering. In this section we describe the steps of the algorithm, demonstrate its potential in process fault detection and finally extend the method in the Data Augmentation frame work for handling missing data. The general structure of the Factor Analysis model is:

$$y_i = \xi_i \boldsymbol{B}' + \varepsilon_i$$

where $\boldsymbol{B}$ is a matrix of order $n \times k$ assumed to be full row rank, called the factor loadings, $\xi_i \sim \Re^{1\times k}$ is the score or factor, and assumed to be normal with mean zero and positive definite covariance matrix Φ. The observations y_i are independent and identically distributed,

$$y_i \sim N(0, \Sigma)$$

with $\Sigma = \boldsymbol{B}\Phi\boldsymbol{B}' + \Omega_\varepsilon$. This general structure is used for two different types of factor analysis, Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis(CFA). Given a data set Y, the essence of EFA is to investigate, whether the observed covariance of the data can be approximated by such a structure with a small number of factors. On the other hand, CFA implies structure in MLFA model with restrictions on $\boldsymbol{B}$ and Φ. These restrictions are implied to relate the underlying factors to some hypothesized causes. Hypothesis testing is done on the restricted values to confirm the structure. In process industries the analysis is more exploratory in nature. Therefore in this section we will restrict our discussion to Exploratory Factor Analysis.

Maximum Likelihood Factor Analysis, based on iterative switching between eigenvalue problem and updating of Ω_ε, has been in use for a long time (Lawley and Maxwell 1963). But because of limited computational power, eigenvalue computations were difficult and it was used only for small problems. The breakthrough in the application of Maximum Likelihood (ML) principles to Factor Analysis (FA) came after the application of fast optimization methods to optimize the log likelihood function (Jöreskog 1970). The work of Jöreskog later led to the well known LISERL program. An Expectation Maximization (EM) version of MLFA is also available (Rubin and Thayer 1982). The iterative steps of EM algorithm are very similar to the steps described by Lawley and Maxwell (1963). Lawley's version of MLFA will be described here because of its close resemblance to PCA and easy implementation (Wansbeek and Meijer 2000).

MLFA Algorithm

1. The initial values of the error variances (diagonal elements of Ω_ε) are selected. This can be taken as 10% of the variances of the observed signals.

2. The estimated covariance of the observed data, S is scaled with the error covariance matrix. Eigenvalue decomposition is performed on the scaled covariance matrix $S\,\Omega_\varepsilon^{-1}$.

3. Λ_k is a diagonal matrix with largest k eigenvalues. $\tilde{B}$ are the corresponding eigenvectors in the scaled domain. The loadings in the original domain are given by, $B = \tilde{B}\Delta$,

$$\Delta = ((\Lambda_k - I_k)(\tilde{B}'\Omega_\varepsilon^{-1}\tilde{B}^{-1})^{-1})^{1/2} \tag{3.25}$$

4. The diagonal elements of the error covariance matrix are calculated as follows:

$$\Omega_\varepsilon(i,i) = S(i,i) - \sum_{i=j=1}^{k} \hat{B}_{ij}^2$$

Step 2 to step 4 are repeated until convergence.

Once the error covariance and the loadings have been estimated the scores are estimated by regressing ξ_i on y_i.

$$\hat{\xi}_i' \equiv E(\xi_i|y_i) = \Lambda^{-1}\boldsymbol{B}'\Omega_\varepsilon^{-1}y_i'$$

Application to Process Monitoring

The application of Maximum Likelihood Factor Analysis in process monitoring is similar to the application of PCA. The breakdown of correlation between the variables can be easily tested using *Squared Prediction Error (SPE)*. Since the data is assumed multivariate normal, *Hotteling T-square plot* can also be used to detect the deviation of the process from normal operating region. Once fault has been detected, SPE contribution charts can be used to isolate the root cause. In addition to using SPE and T-square chart, *Squared Weighted Residuals (SWR)* can be used for fault detection and isolation, since estimates of error variances are also available. SWR gives sharp detection and it has superior fault isolation capability over SPE. Under normal condition SWR has a *chi-squared* distribution (Oxby and Shah 2000).

Connection with Modified PCA Methods

Maximum Likelihood Factor Analysis (MLFA) is closely related to two modified form of PCA, Probabilistic PCA (PPCA) and Bayesian PCA (BPCA). Unlike PCA these two methods assume that the underlying noise-free variables X are multivariate normal which is exactly the assumption in MLFA. Probabilistic PCA is actually a restrictive case of MLFA. It further assumes that the measurement noise is isotropic-iid ($\Omega_\varepsilon = \sigma^2 I$) (Tipping and Bishop 1999). On the other hand, the main objective of Bayesian PCA is to use the prior information about the loadings in performing

Principal Component Analysis. The method can deal with small data samples very effectively. However, the method is optimal in the least squares sense, does not incorporate the noise covariance in estimation and falls short in achieving maximum likelihood estimates (Nounou and Shen 2002).

Missing Data Handling in MLFA

The EM algorithm for factor analysis gives a natural way to handle missing values in MLFA. However, the method does not use eigenvalue solution and the steps of the algorithm are not intuitive. The PCA-like MLFA algorithm described in Section 3.4.1 will be extended for handling missing data. Since the error variances are also calculated, an intuitive way is to perform error correction while imputing for the missing values, and extend the method in Data Augmentation framework.

The Imputation step (I-step) of the Data Augmentation algorithm can be implemented very easily using the iterative algorithm. Using the same description of Section 3.4.1, at any step t the estimate of the noise free variable X is given by,

$$\hat{x}_n^{(t)} = B^{(t)}\hat{\xi}_n^{(t)}$$

$$\hat{y}_{mis}^{(t)} = \hat{x}_{mis}^{(t)} + \nu_i L$$

where $\nu_i \sim N(0, I)$. So, $\hat{y}_{mis}^{(t)}$ is a random draw from the distribution conditioned on Y_{obs} and $\Sigma^{(t)}$.

The Posterior step (P-step) is to sample the parameters from their respective distributions. Here the parameters are the eigenvectors of the covariance matrix. Instead of sampling each element of the eigenvectors from its distribution the covariance matrix is sampled from the distribution, and eigenvalue decomposition is performed on each sampled covariance matrix. For implementation purpose, the covariance matrix is sampled from the Inverse-Wishart distribution, as the posterior distribution of the covariance matrix of a multivariate normally distributed data has an Inverse-Wishart distribution (Gelman and Rubin 2004). The covariance matrix estimated from the augmented data matrix $Y_{aug} = \left(Y_{obs}, Y_{mis}^{(t)}\right)$ is used to scale the Wishart distribution. The uncertainty of the covariance matrix gets transferred to the model parameters which in this case are the eigenvectors of the covariance matrix. So essentially the eigenvectors are sampled from their distribution. The distribution of the posterior density of the covariance matrix of multivariate Gaussian samples is derived in the Appendix.

3.5 Results and Discussions

3.5.1 Flownetwork Example

The flow-network process, shown in Figure 3.2, will be used to compare the relative advantages and disadvantages of different methods. This is a benchmark example used by Narasimhan and Shah (2004) and a similar example was used by Nounou and Shen (2002) to evaluate different properties of Bayesian PCA. It is assumed that the fluid flowing through the network is incompressible and there is no time delay in the process. The constraint model A, of the process can be obtained easily from the mass balance equation at the junctions. The following four flow balance equations can be written for this flow-network system:

$$\begin{aligned} x_1 + x_2 - x_3 &= 0 \\ x_3 - x_4 &= 0 \\ x_4 - x_5 - x_2 &= 0 \\ x_5 - x_6 &= 0 \end{aligned}$$

where x_i to x_6 are flowrates at different points of the system. Thus the constraint model is:

$$A = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

The rank of the constraint matrix is four, which is also known as the order of the constraint model. In the above example x_1 and x_2 were chosen as independent variables. These are two deterministic signals and output of auto regressive (AR) processes given in Table 3.2. The rest of the flow rates, x_3 to x_6 were calculated from the mass balance equations. These variables are noise free and satisfy the constraint,

$$AX^T = 0$$

where $X = [X_1 \;\; X_2 \;\; X_3 \;\; X_4 \;\; X_5 \;\; X_6]$ and X_1 to X_6 are vectors containing the actual flowrates at each sampling point. However, in process industries the actual values of the variables are generally not available, only the noise corrupted measurements Y are available,

$$Y = X + \varepsilon$$

where ε is a matrix containing the measurement noise. Measurement noises are assumed Gaussian, independent and identically distributed and also uncorrelated in the variable direction $\left(i.e.,\ \varepsilon_i \sim N\left(0,\ \sigma_j^2 I\right),\ j = 1, 2 \ldots, 6\right)$.

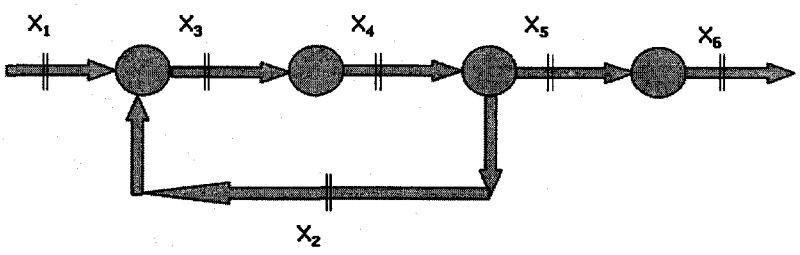


Figure 3.2: *Schematic Diagram of the Flow-Network*

Table 3.2: Transfer Functions of the Deterministic Signals

X_1	X_2
$\frac{1}{1-az^{-1}}$	$\frac{1}{1-bz^{-1}}$

3.5.2 Performance measures

The following two measures were used to quantify the performance of the proposed algorithms.

Subspace angle

Principal Component Analysis (PCA) is carried out by applying Singular Value Decomposition (SVD) on the covariance matrix where the loadings of the PCs are given by the eigenvectors. In a multidimensional problem the eigenvectors can be multiplied by any non-singular matrix to define the same hyperplane. The exact value of each element of the matrix depends on how the basis vectors are selected. Therefore, direct comparison of the elements of the eigenvectors with actual model parameters is not feasible. Instead one should examine if the hyperplane defined by the estimated model is in agreement with the actual model hyperplane. In this study the subspace angle, θ is used to measure such agreement.

Let F and G be given subspaces of real space $\Re^m$, $u \in F$, $v \in G$, and assume for convenience that $p = dim(F) \geq dim(G) = q \geq 1$. The smallest angle $\theta_1(F,G) = \theta_1 \in [0, \pi/2]$ between F and G is defined by

$$\cos(\theta_1) = max_{u \in F} max_{v \in G} u^T v$$

Assume that the maximum is attained for $u = u_1$ and $v = v_1$. Continuing in this way until one of the subspaces is empty, we are led to the following definition. The principal angles $\theta_k \in [0, \pi/2]$ between F and G are recursively defined for $k = 1, 2, \cdots, q$ by,

$$\cos(\theta_k) = max_{u \in F} max_{v \in G} u^T v = u_k^T v_k, \|u\|_2 = 1, \|v\|_2 = 1$$

subject to the constraints

$$u_j^T u = 0, v_j^T v = 0$$

where σ_k is an eigenvalue of $F^T G$. Therefore subspace angle or principal angle is the minimum angle between the subspaces (Bjorck and Golub 1973). On the other hand, similarity index is a combined index defined by,

$$\theta_0^2 = \frac{1}{q} \sum_{i=1}^{q} \cos^2(\theta_i) = \frac{1}{q} \sum_{i=1}^{q} \lambda_i$$

where λ_i is the eigenvalue of $F^T G G^T F$. The value of the similarity index is between 0 and 1, where 1 means that the two subspaces are linearly dependent (Krzanowski 1979). Clearly these two indicators have the same origin but differs in the way the result is reported. In the current study the subspace angle is used to quantify the model quality. The built-in function 'subspace.m' from Matlab's 'Data analysis and Fourier transforms' toolbox is used to calculate the subspace angle. The details of the algorithm can be found in Knyazev and Argentati (2002).

Total Sum of Squared Error (TSE)

The main objective of process monitoring is to estimate the noise-free values of the signals. In order to evaluate the performance of the proposed algorithms we also calculated the total sum of squared errors (TSE) between the noise-free signal and predicted signal. Total sum of squared error is given by,

$$TSE = \sum_{i=1}^{N} \sum_{j=1}^{m} (x_{ij} - \hat{x}_{ij})^2$$

In addition to the prediction trend plots, TSE gives a quantitative way of comparing the performances of the algorithms.

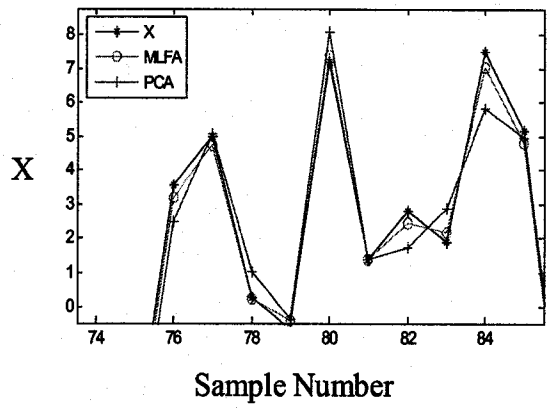


Figure 3.3: *Trend Plot of X and predicted $\hat{X}$ using MLFA and PCA*

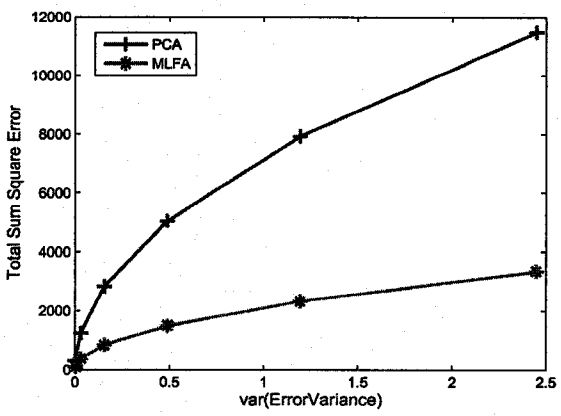


Figure 3.4: *Total Sum Square Error between X and predicted $\hat{X}$ using MLFA and PCA*

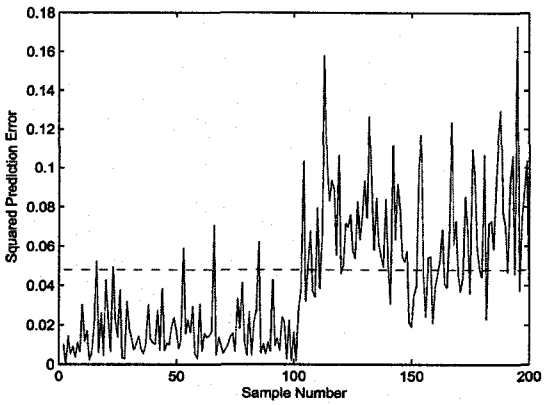


Figure 3.5: *Squared prediction error calculated using PCA*

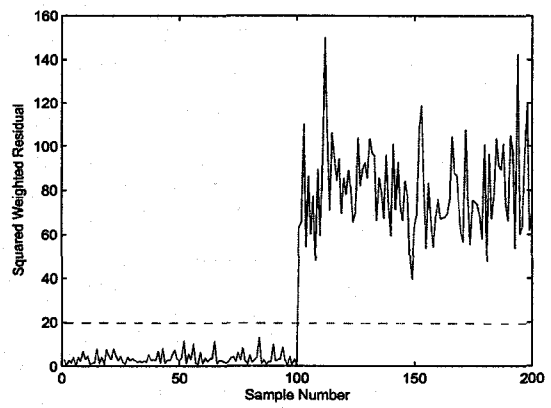


Figure 3.6: *Squared weighted residuals calculated using MLFA*

3.5.3 Monitoring of Structural Process

Simulation studies have been carried out using the flownetwork system to check the performance of the MLFA and PCA in monitoring structural processes. In order to simulate structural process data, it is assumed that the independent variables X_1 and X_2 are sampled from the Gaussian distributions $N(10, 4)$ and $N(20, 5)$ respectively. Since the other variables are linear combinations of these two variables, data matrix X will have a multivariate normal distribution.

One of the primary objectives of process monitoring is to estimate the noise-free measurements, X as defined in Equation 3.1. In this study the estimated $\hat{X}$ from MLFA and PCA have been compared with the true X. Figure 3.3 shows that the estimated $\hat{X}$ from MLFA follows the true X more closely than the estimated values from PCA. To compare the models in a more quantitative manner total sum of squared errors (TSE) between the estimated values and the true values have been plotted in Figure 3.4. The *var(error variances)* is plotted along x-axis, which is an approximate measure of non-isotropic (i.e., variances of each of measurement noise is unequal) nature of the errors. Higher values signify that the noise is more non-isotropic. It is evident from the figure that MLFA minimizes higher prediction error than PCA at all non-isotropic error conditions. The two methods are equivalent at the limit when the errors are isotropic. In the process of estimating latent variables using MLFA, variances of the measurement errors are also estimated. Therefore, squared weighted residuals (SWR) can be calculated and used for fault detection purpose. In Figure 3.5 and 3.6 we compare the fault delectability of the two algorithms. A sensor fault has been simulated by adding a constant bias to one of the sensors. Though both PCA and MLFA were able to detect the fault, the squared weighted residuals obtained from MLFA are more sensitive to the fault and shows abnormal behavior very distinctly on the SWR plot. Also there is no false alarms during the normal operation. On the other hand the SPE-plot obtained from PCA have many false alarms during the normal operation. The better fault detection performance of SWR over SPE observed in this case can be attributed to two factors (i) SWR is more sensitive to faults than SPE for non-isotropic measurement noise (ii) the minimization criteria of MLFA is better suited for non-isotropic noise. However, it is arguable which has the dominant effect. Even if the better performance is due to the sensitivity of the SWR, it is not possible to calculate SWR in PCA based monitoring since the covariance of measurement noise is not estimated.

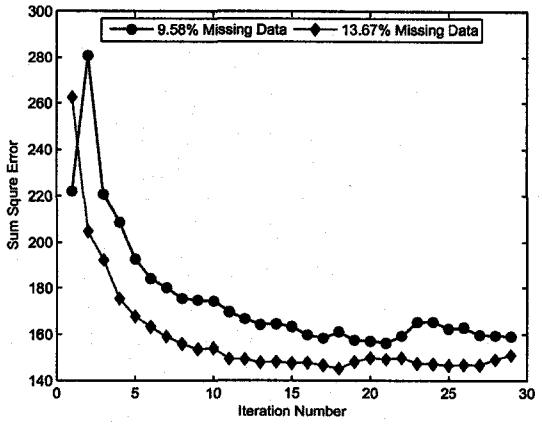
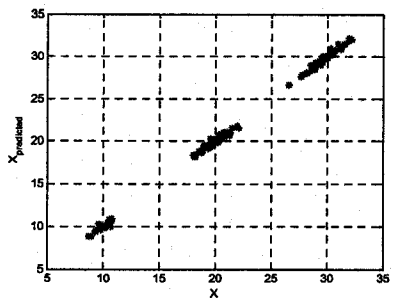


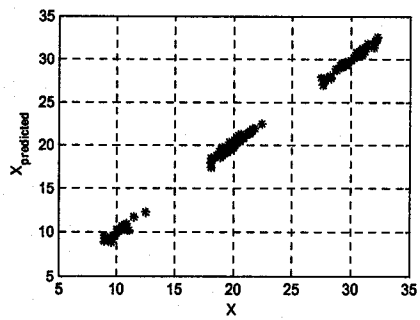
Figure 3.7: *Sum of Squared Errors between the observed measurements and the predicted noise free values showing the convergence of MLFA-Data Augmentation algorithm*

3.5.4 Results on missing data handling in Structural Model

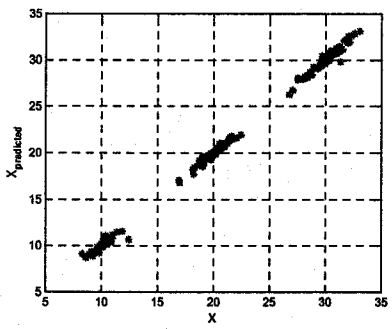
MLFA was extended to the Data Augmentation framework for dealing with missing data. The convergence property of the proposed algorithm in the presence of missing data is shown in Figure 3.7. The algorithm converged up to a very high percentage of missing data. However, the data requirement for the algorithm is quite substantial. For the flownetwork system with six variables, at least 2000 samples were required to estimate good estimates. The estimated values of the missing measurements by MLFA-DA are plotted against the true noise-free values in Figure 3.8. The plot shows good agreement between the estimates and the true values. However, as the missing percentage goes above 20% the performance of the algorithm deteriorates. At this high percentage of missing values it becomes a non-identifiable problem. In that case too many parameters are being estimated from too few constraints. In other words, when the percentage of missing value is high there is more probability that on some rows we will not have sufficient number of observed values for calculating all the missing values using the constraints. Therefore, some the estimates of the missing values will not change from the initially assigned values during the iterative process. In addition to that, the estimates of error variances also get affected. Since the variables are scaled with the standard deviation of the errors, a poor estimate of error variances may lead to poor estimates of the missing values.



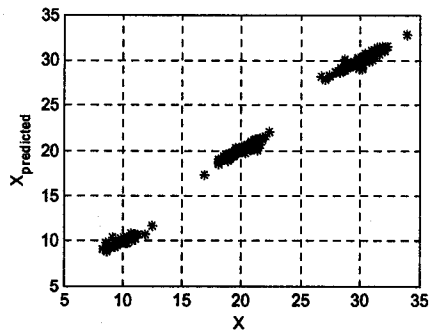
(a) Plot of predicted missing X vs. actual X for 9.58% missing data.



(b) Plot of predicted missing X vs. actual X for 13.68% missing data



(c) Plot of predicted missing X vs. actual X for 18.5% missing data.



(d) Plot of predicted missing X vs. actual X for 21.83% missing data.

Figure 3.8: *Plot of predicted noise free values of the missing measurements by MLFA-DA against the true values*

3.5.5 Results on the Functional Latent Variable Problem

The functional process behavior is simulated using the transfer functions given in Table 3.2 with $a = 0.9$ and $b = 0.8$. Utility of Missing data handling technique is the most when the sample size is small. In this case the sample size is 200.

Results on Model order Selection

Figure 3.9 shows that with more data missing in the data matrix, total variance explained by the first two principal components increases. So, methods which use variance information for selecting model order, such as, SCREE-plot, Broken Stick method are affected because of missing data. Figure 3.10 shows the model order selection in the presence of 15% missing data using cross validation. Though the calculated values of prediction error sum of squares (PRESS) change due to missing values, the slope of the curves clearly indicate that the first two PCs are sufficient to express the variability of the data. However, PRESS calculated using PCADA are closer to the actual values than calculated using PCAIA.

The improved model order selection criteria of IPCADA in the presence of missing data are evident from Figure 3.11. Because of the scaling strategy, in IPCA the eigenvalues corresponding to the null space becomes one. This property is very useful in selecting the order of the model. The estimated eigenvalues by IPCADA at different percentages of missing data are shown in Figure 3.11. The last four eigenvalues do not deviate significantly from unity. This is primarily because at each iterative step residual corrections are carried out on the estimated missing values. This helps to restore overall covariance and error covariance structures of the original measurements to some extent. Therefore, the model order selection is precise for IPCADA in the presence of missing data.

Convergence Property

Convergence of PCA-Data Augmentation (PCA-DA) and IPCA-Data Augmentation (IPCA-DA) were monitored using the calculated sum of squared errors between the observed values and corresponding predicted noise-free values. In the flownetwork example, the actual constraint model is exactly known, the changes of the subspace angles with iteration steps were also calculated to reaffirm the convergence properties. Both the sum of squared errors and the subspace angles decrease with each additional iteration step and reaches to their minimum values at convergence. It is also evident that both indices have similar trends and point towards the convergence around the same iteration steps (Figure 3.12, Figure 3.15). Therefore, when

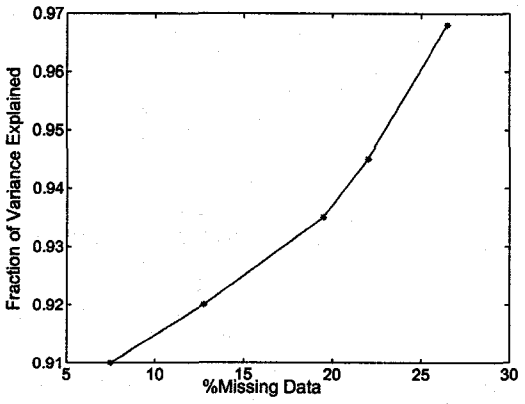


Figure 3.9: *Fraction of Variance Explained by first two PCs in Principal Component Analysis vs. %Missing Value*

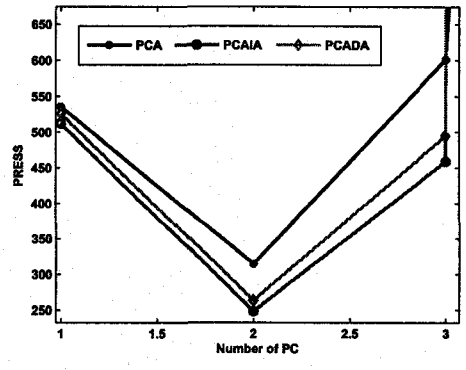


Figure 3.10: *Model order selection using cross validation*

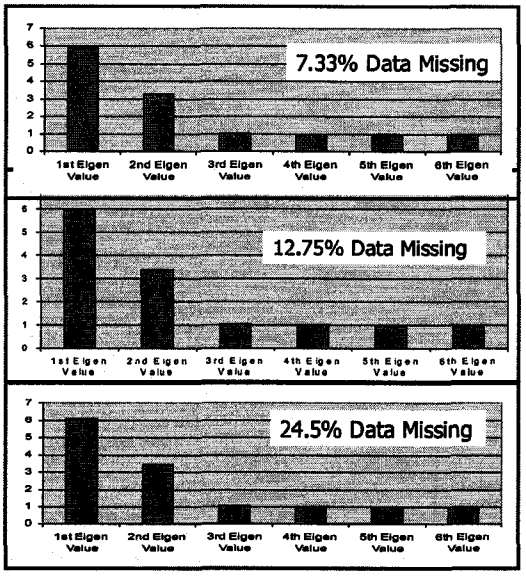


Figure 3.11: *Estimated Eigenvalues from IPCADA; the last four eigenvalues are unity*

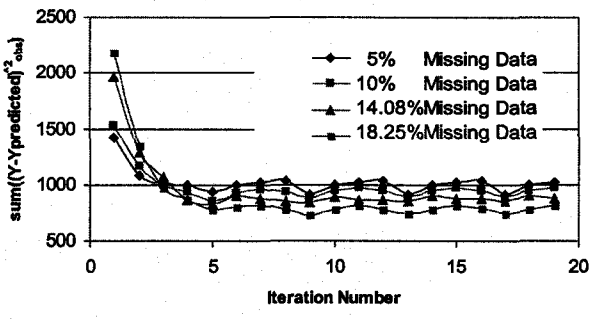


Figure 3.12: *Convergence monitoring of IPCADA using the sum of squared error between the observed and the predicted values*

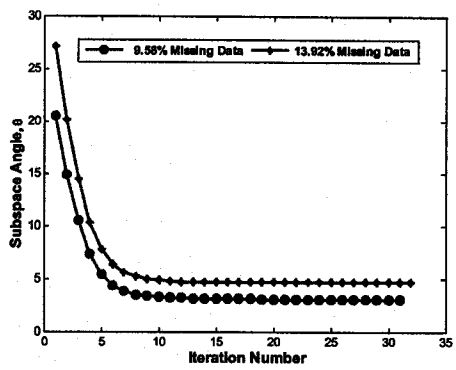


Figure 3.13: *Convergence monitoring of PCA-Iterative Algorithm using the Subspace Angle*

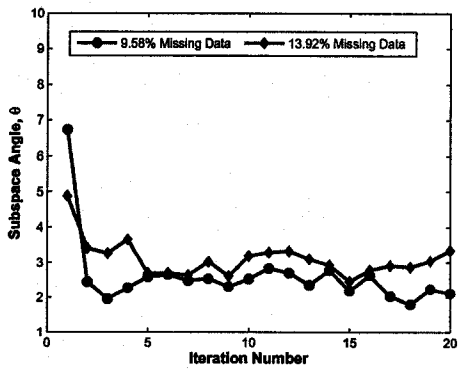


Figure 3.14: *Convergence monitoring of PCA-Data Augmentation using the Subspace Angle*

the true model is not known sum of squared errors between the observed values and the corresponding predicted values can be used to monitor the convergence of the algorithms. However, the convergence of PCA-Data Augmentation (PCADA) and Iterative-PCA-Data Augmentation (IPCADA) are not smooth like PCAIA as shown in Figure 3.13 because at each iterative step residual error corrections are carried out by adding random noise with the conditional expected values. This randomness is also reflected on the convergence plot. Once the algorithm converges the indices vary around the minimum values. Therefore one should check for a bounded value rather than a constant term to determine the convergence.

PCADA converges even for very low signal to noise ratio and high percentage (i.e.

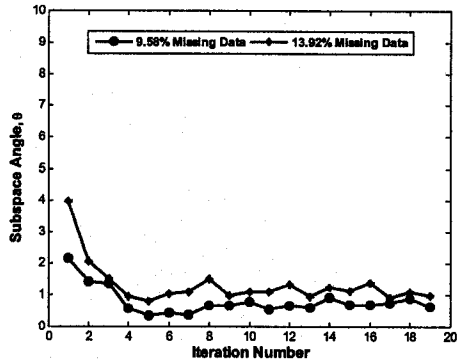


Figure 3.15: *Convergence monitoring of IPCA-Data Augmentation using the Subspace Angle*

True Error Variances	Estimated Error Variances (without Error Correction)	Estimated Error Variances (with Error Correction)	Estimated Error Variances (using IPCADA)
1.7	1.094	1.325	1.85
0.16	0.109	0.175	0.13
0.49	0.323	0.623	0.494
0.36	0.282	0.31	0.331
0.04	0.0	0.0	0.059
1.21	0.8741	1.13	1.37

Table 3.3: *Comparison of Estimated Error Variances with True Values*

25%) of missing data. IPCADA also enjoys good convergence properties except at very low signal to noise ratio with very high percentage of missing data. In those cases the estimation of error variances is poor and also the variance corrections are of the same order of magnitude as the contributions from the underlying true signal. Therefore it interacts with the underlying signals and the model quality deteriorates. However, the sum of squared errors between the observed values and the predicted values also shows the divergence and the iterations can be terminated at that point.

Estimation of error variances is an integral part of IPCADA algorithm. Due to the presence of missing values in the data matrix these are the most vulnerable parameters. In Table 3.3 the estimates of error variances have been compared with the true error variances. True error variances are given in column 1. Reported values in column 2 are estimated by using only the IPCA algorithm iteratively and missing

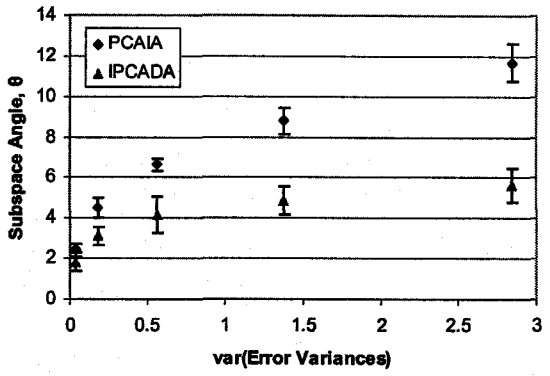


Figure 3.16: *Model Quality Comparison. Subspace Angle between the estimated model and actual model vs. variance of error variances*

values are filled by directly substituting the conditional expected values without any error correction. The values in column 3 are obtained by IPCA where the conditional expected values were corrected to account for the measurement errors before being used for imputation, but 'Bootstrap' is not used to sample the parameters from the distributions. Parameters in column 4 are estimated using the complete IPCADA algorithm. The error variances of the signals estimated using IPCADA are the closest to the true values. For the other two methods the estimated values of the smaller error variances are poor and the smallest error variance estimate always tends to zero.

Comparison of Model Quality

It was claimed in the previous sections that the IPCADA algorithm is advantageous over PCAIA when the errors are non-isotropic (*i.e.*, $\Omega \neq \sigma^2 I$). To demonstrate the point, the Subspace Angle (θ) between the models estimated by PCAIA and IPCADA are plotted in Figure 3.16. The deviations of the noise variance from isotropic noise has been quantified by *var(error variances)*. In both cases 10% of the total values are missing. When the noise variances are nearly isotropic two methods are practically indistinguishable. But as error variances become more non-isotropic, the performance of the PCA based algorithm deteriorates sharply.

The quality of the models estimated from the algorithms at different percentage of missing values are compared in Figure 3.17. In this case the error variances deviates from isotropic noise only moderately; *var(error variances)* is 0.05. Estimated models from IPCADA algorithm have better quality for missing values up to 20%. For missing

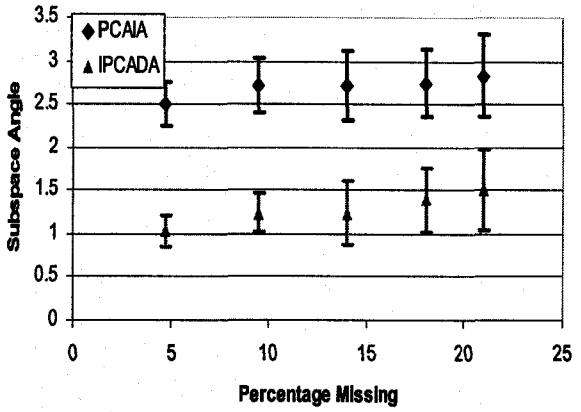
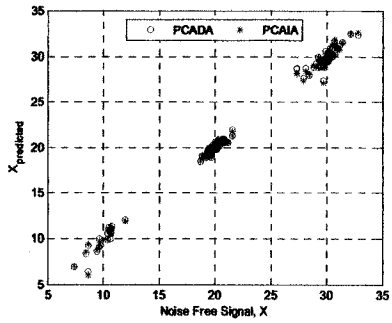


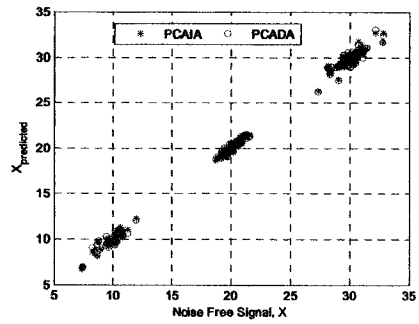
Figure 3.17: *Flownetwork Example: Comparison of Model Quality estimated by IPCDA and PCAIA at different percentage of missing values*

values beyond that range the use of IPCADA is not recommended, as the algorithm runs into convergence problem and the quality of the model may deteriorate sharply. At high percentage of missing data, similar to MLFA there may arise identifiability problems as discussed in Section 3.5.4.

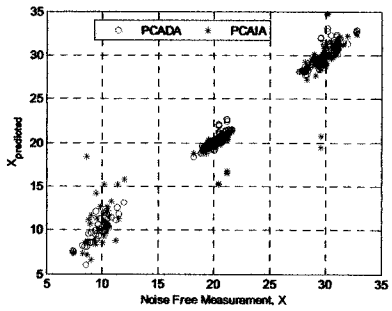
In Figure 3.18 the estimated values of the missing measurements have been plotted against the noise-free true values. The plot clearly shows that PCADA gives better prediction than PCAIA. In order to get a quantitative feel, we also calculated the sum of squared errors between the predicted values and the true values as shown in Figure 3.19 for different percentages of missing values. These values are an average of twenty Monte Carlo simulations and the error bars indicate the standard deviations of the estimates. It is evident from the plot that the proposed PCADA algorithm gives better estimates of the missing values, as well as, the estimated models are of better quality than that of estimated by PCAIA. However, the computational load of PCADA is substantially higher than PCAIA. Until now we have shown the application of the iterative techniques for solving problems which are directly related to missing data. Besides these a wide range of problems can be formulated as missing data problem and the iterative techniques may be used effectively to solve such problems. In the following section we demonstrate one such example where PCAIA is used to synchronize uneven length batch process data. PCAIA is used mainly from the consideration to limit the computational load of the analysis, as the method is already computationally intensive.



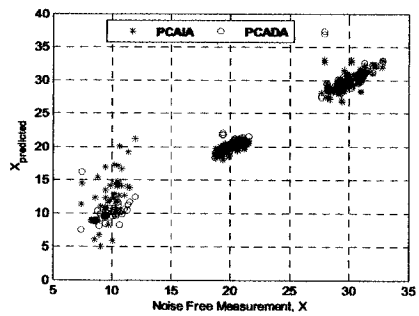
(a) Plot of predicted X vs. actual X at 9.41% missing data



(b) Plot of predicted X vs. actual X at 13.71% missing data



(c) Plot of predicted X vs. actual X at 17.75% missing data



(d) Plot of predicted X vs. actual X at 21.58% missing data

Figure 3.18: *Plot of predicted noise free values of the missing measurements by PCAIA and PCADA against the true values*

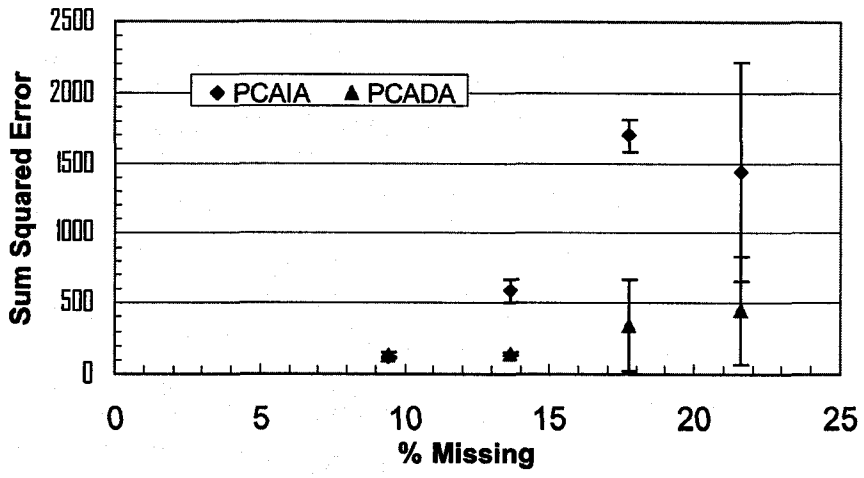


Figure 3.19: *Sum of Squared Errors between the true values and the values predicted by PCAIA and PCADA at different percentage of missing values*

3.6 Synchronizing Uneven Length Batch Process Data

Batch processes are used for producing highly value added products such as, pharmaceutical products, speciality polymers, biomedical products etc. Monitoring of batch process is important because early detection of the anomaly can facilitate taking corrective actions or altogether discarding further costly processing of the batch. Multiway PCA (MPCA) based methods are commonly used for monitoring of batch processes (Nomikos and MacGregor 1994). Batch process monitoring is essentially monitoring the trajectories. The allowable band for the trajectories are calculated from the trajectories of different batches. Therefore to take this variation into account data from different batches have to be included for building the model. So typical batch process data will have a three-way matrix structure as shown in Figure 3.20(a).

For applying PCA data need to be unfolded to a two-way structure. Data can be unfolded in three distinct ways to a two-way data matrix (Westerhuis *et al.* 1999). However, the unfolding proposed by Nomikos and MacGregor (1994) is the most meaningful in this context as it provides a way to include the normal batch to batch variations in the model. This is illustrated in Figure 3.20 in steps (d) and (e). In a typical batch run ($j = 1, 2, \cdots, J$) variables are measured at $k = 1, \cdots, K(i)$ time intervals. Here the end point of the measurements is a variable and varies from batch

to batch. The variation of batch length may be due to a wide variety of reasons including feed quality variation, poor control etc. Similar data will be generated from several $i = 1, \cdots, I$ similar batch runs. This vast amount of data can be organized in a three-way data matrix. The unfolding proposed by Nomikos and MacGregor (1994) slices the matrix in vertical direction and arrange the time slices side by side. In the unfolded matrix each batch appears as an object. The data are then mean centered and scaled prior to applying PCA. This unfolding is particularly meaningful because by subtracting the means of the columns of the unfolded matrix the main non-linear and dynamic component of the data are removed. A PCA performed on these mean-centered data is therefore a study of the variation in the time trajectories of all the variables in all batches about their mean trajectory. However, if the batch lengths are uneven, before carrying out the mean centering operation different batches need to be synchronized. In this section we will directly move into the various issues related to synchronization of batch data. The details of the application of MPCA techniques in batch process monitoring can be found in Nomikos and MacGregor (1995*a*) and Nomikos and MacGregor (1995*b*). Review and comparative study on different techniques of batch process MPCA was done by Eric N. M. van Sprang (2002) and Westerhuis *et al.* (1999).

Several methods have been used to synchronize the uneven length data from different batches. Nomikos and MacGregor (1994) proposed the use of an indicator variable instead of time. Data from different batches are synchronized with respect to the indicator variable. The variable has to have some specific properties, such as, same starting and ending point in different batches, monotonicity that is increasing or decreasing trend in time and free from noise. A constant increment is selected along this indicator variable and rest of the variables are synchronized with respect to this indicator variable. This method has also been used by Kourti *et al.* (1996) to synchronize a semi-batch polymerization process. The main critique for the method is, in many cases it is difficult to find a variable which can meet all these criteria. A simpler practical solution was used by Lakshminarayanan *et al.* (1996) where they padded the shorter batches with the last measurements and made all of the batch lengths equal. This essentially implies that, all the time differences are at the last stage of the batch process. Therefore it is not suitable for batch processes which have multiple stages. However, the method works well in many situations when the batch lengths are not substantially different. Another option is to consider data from all batches only up to the shortest batch length. Thus data collected during the later stage of the longer batches are not included in the model. Unfortunately the data collected towards the end is of great interest as these measurements provide infor-

mation whether or not the reaction is complete or the cycle is finished. To estimate the end point of the batch process a two-stage method is proposed by Marjanovic *et al.* (2006). The method is particularly useful for predicting the batch completion time before hand. However, until now the most general and elegant solution for synchronizing batch trajectories is via Dynamic Time Warping (DTW). DTW is widely used in speech recognition, particularly in isolated word recognition (Myers *et al.* 1980, O'Shaughnessy 1986, Silverman and Morgan 1990). In chemical processes DTW was introduced by Gollmer and Postens (1995) to detect the onset of different growth phases and failure in batch fermentation process. Nomikos and MacGregor (1994) later used DTW for synchronizing un-even length batch trajectories. Dynamic Time Warping is a flexible, deterministic pattern matching technique. It is able to translate, compress and expand the patterns locally, which are very attractive features for multi-stage batch process data. The method uses the theory of dynamic programming, hence the name Dynamic Time Warping. There are many versions of Dynamic Time Warping. However, for batch data synchronization asymmetric dynamic time warping is commonly used and in this study we will focus on asymmetric DTW only.

Let T and R denote the 'test' and 'reference' multivariate trajectories with dimensions of $t \times m$ and $r \times m$ respectively, where t and r are the number of observations, m is the number of variables and i, j denote the time indices of the respective trajectories. DTW will find a sequence F^* of K points on a $t \times r$ grid.

$$F^* = c(1), c(2), \cdots, c(k), \cdots, c(K) \tag{3.26}$$

$$max(t, r) \leq K \leq t + r \tag{3.27}$$

where $c(k) = [i(k), j(k)]$ and each point $c(k)$ is an ordered pair indicating a position on the grid. In the most common asymmetric DTW algorithm the test trajectories will match the time index of the reference trajectory. Therefore, the common time index k is in fact the time index j of the reference trajectory R, and contains exactly r points.

$$F^* = c(1), c(2), \cdots, c(k), \cdots, c(r)$$

and $c(j) = (i(j), j)$. This implies that the path will go through each vector of R but it may skip vectors of T.

Asymmetric DTW is a two step procedure. In the first step, the test batch data points are aligned along the time indices of the reference batch. Some total distance measured between the two trajectories are minimized in order to find the best indices

for the test batch. The most commonly used local distance is the weighted quadratic distance.

$$d(i(k), j(k)) = (T(i(k),:) - R(j(k),:))W(T(i(k),:) - R(j(k),:))^T \quad (3.28)$$

$$D(t,r) = \frac{\sum_{k=1}^{K} d[i(k), j(k)]}{w(k)} \quad (3.29)$$

where W is a positive definite weight matrix that reflects the relative importance of each measured variable and $w(k)$ is a nonnegative weighting function for $d(i(k), j(k))$. The optimal path is found as the solution to the following optimization problem:

$$D^*(t,r) = min_F[D(t,r)] \quad (3.30)$$

$$F^* = argmin_F[D(t,r)] \quad (3.31)$$

Several local and global constraints are also imposed in the algorithm. End point bounds are the most common and useful when the end points of both trajectories are known with certainty. It implies that the first c(1) and the last c(K) path points are as follows:

$$c(1) = (1,1) \quad (3.32)$$

$$c(K) = (t,r) \quad (3.33)$$

Monotonicity constraints are imposed (e.g., $i(k+1) \geq i(k)$ *and* $j(k+1) \geq j(k)$) to preserve natural order of the trajectories in time. In order to prevent excessive compression or stretching, slope constraints are included into the algorithm. This is imposed by specifying a set of allowable predecessors for each point in the grid.

After aligning the trajectories the next step is to deal with the excess or inadequate data points in the synchronized data matrix. If the length of the test batch is longer than the reference batch, some points of the test batch need to be discarded. For example, if two points from the test batch correspond to one point in the reference batch then based on the distance measure one point is discarded or an average of these two points are assigned to that position. On the other hand, if the test batches are shorter than the reference batch, after alignment there would be many gaps in the test data set (i.e., some of the rows of the test data set will be empty). Typically a 'zero order hold' or 'first order hold' is used to fill these gaps. However, none of these methods are deemed appropriate considering the fact that batch process data

Steps of Synchronizing the Uneven Length Batch Process Data
Step 1: Collect data in a three way data matrix. Select a reference batch from the collection of data.
for i=1... I
Step 2: Align data set from each batch with the reference data set using dynamic time warping.
Step 3: Create a dynamic data matrix by including the lagged variables.
Step 4: Use PCAIA to fill the missing values of the shorter data sets.
end
Step 5: Unfold the data set in two dimensional data matrix and build model using PCA.

Table 3.4: *Algorithm describing the steps of synchronizing data from batch processes with different completion time using a combined DTW and missing data technique*

are dynamic and multivariate. By applying 'zero order hold', though the spatial correlation between the variables are preserved, the temporal trends of the variables get distorted. On the other hand, 'first order hold' or 'linear interpolation' takes care of the temporal correlation to some extent but destroys the spatial correlation between the variables. Therefore a method is needed which will preserve both temporal trend and spatial correlation between the variables. In the current study we propose a method based on missing data handling technique which attempts to conserve both temporal and spatial correlation of the batch data set. The method takes advantage of the data matching capability of DTW and multivariate nature of the missing data handling technique.

3.6.1 Combined DTW and Missing Data Technique

The basic idea is to align the test data set along the time stamp of the reference data set using DTW. If the test data set is shorter than the reference data set, some gaps will be created in the test data set. Subsequently these gaps are filled by applying missing data handling technique. The overall methodology of the technique

is described in Table 3.4. We have discussed the basic principle of DTW in the previous section. In this section we will explain how missing data technique is used to fill the gaps in the shorter batches. The pattern of data from a single batch after synchronization is shown in Figure 3.20(b). Apparently in this pattern there is no way of using multivariate methods to fill the missing values as the rows that contain missing values are completely empty. Therefore, the model cannot be used to predict unique values for the missing data points. However, batch process variables are not only spatially correlated with each other at any given time, but also correlated in the temporal direction. So it is reasonable to include lagged variables in the data matrix and the pattern of the data matrix with lagged variables is shown in Figure 3.20(c). From this pattern it is evident that iterative missing data techniques can be used to fill the missing values, as well as build PCA model. Once the lagged data matrix has been created, PCAIA is applied to fill these missing values. PCAIA is a pseudo-EM algorithm which iterates between the model parameter estimation and missing value estimation steps. The algorithm is described in Section 3.3.1. Because of the time shifted values in the rows containing missing values, we will get some unique prediction of missing values by using the model. As the model is multivariate the predicted values are also consistent with the correlation structure of the data. The procedure is repeated for each set of batch data which has length shorter than the reference batch. Once all batches have the same length they can be unfolded by any of the three unfolding techniques. In the current study, we unfolded the matrix in the variable direction as proposed by Nomikos and MacGregor (1994). Once the data has been arranged in a two-dimensional rectangular structure, the ordinary PCA can be applied. Notice that, in this example PCAIA has been applied to dynamic data contrary to the steady state data dealt in the previous sections. In the following section we demonstrate the technique using a batch polymer reactor.

3.6.2 Batch Polymer Reactor

The proposed methodology is applied to a feed batch polymer reactor process (Chen and Liu 2002). The reaction system involves two consecutive first-order reactions:

$$A \rightarrow B \rightarrow C$$

The schematic diagram of the reactor with the different measurement locations are shown in Figure 3.21. The reactor is operated in closed loop under an on/off control strategy. The reaction has three distinct stages. In the *start-up* stage, the steam in the jacket initially heats up the reactor content until the temperature reaches desired

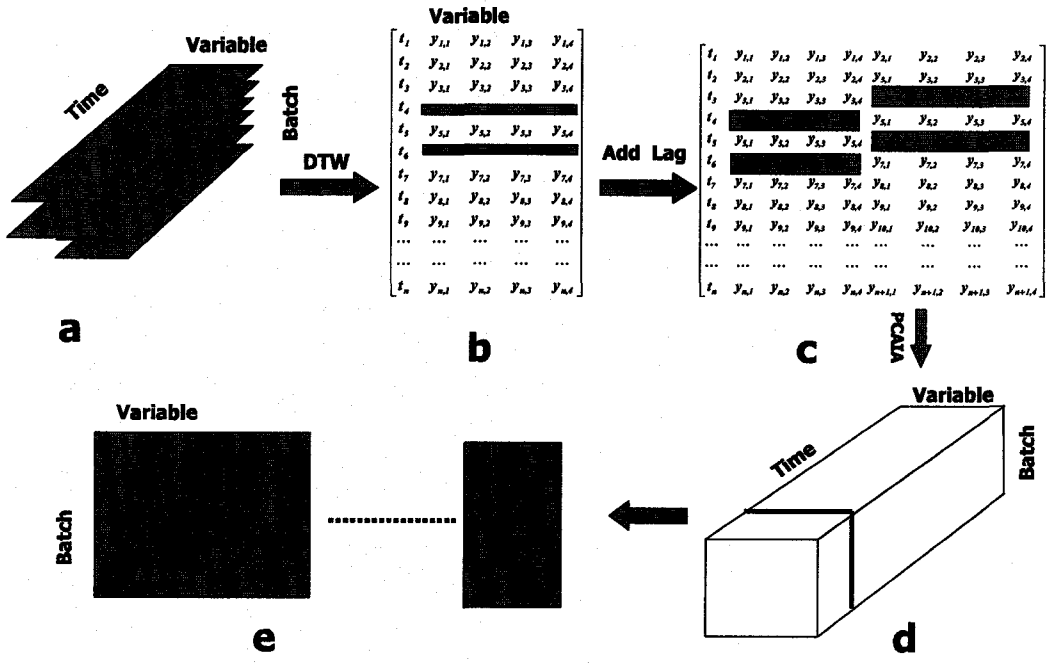


Figure 3.20: *Pictorial representation of the proposed technique for synchronizing the uneven length batch process data using the combined DTW and PCAIA, and unfolding to a two way data matrix*

operation level. In the second *cooling* stage, the cooling water in the jacket is used to remove the exothermic heats of reaction. The third stage is more of a *maintenance* stage, the reaction temperature is self sustained at this stage. The cooling water valve is turned on occasionally to take out the excessive heat generated from the reaction. The jacket temperature, the temperature of the metal wall between the reactor and the jacket, the reactor temperature and the cooling water flow rate are the four measured process variables. Two quality variables, concentrations C_B and C_C are measured at the end of each batch run. The simulation conditions and relevant parameters remains same as that of Luyben (1990), except the initial concentration C_A.

Variation in initial feed concentration is common, as feed may be obtained from different sources or because of the presence of impurities in the feed. We consider a range of initial feed concentrations which affect the batch completion time. Data are collected from a total of sixteen batches and the batch completion time varies from 73 minutes to 100 minutes. We select the batch with 95 min completion time as our reference data set since the distribution of the batch lengths has a peak around that point. This will minimize the time shifting of the rows of the batches which have completion time on both ends to some extent. In order to build the monitoring scheme the four process variables are included in the model.

The typical trend of the variables are shown in Figure 3.22. Since the process has three different stages and there is no monotonically increasing or decreasing variable, DTW is the most appropriate method for synchronizing the data. The data are synchronized using two different techniques. First, we use DTW for aligning all the batches along the reference batch and apply 'first order hold' to fill the gaps of the shorter batches. Second, we use DTW for aligning the data with the reference batch and use the proposed methodology based on missing data handling technique to fill the missing values. Subsequently PCA is carried out on the complete-synchronized data sets.

PCA is carried out on the data matrices synchronized by the two different methods and the results are illustrated in Figures 3.23 through 3.26. Figure 3.23 gives the cumulative percentage of variances explained by the PCs. We are able to get a very compact model from the synchronized data where the proposed missing data handling technique is used to fill the missing values. The first principal component, extracted from the data set which is synchronized using the proposed methodology, explains 85% of total variance compared to 40% of total variance explained by PC_1 obtained from the data synchronized using 'DTW-first order hold'. In the proposed method only two PCs explain 90% cumulative variance, whereas eight PCs are required to

explain the same amount of variance of the data set synchronized using 'DTW-first order hold'. In MPCA compact model has a special meaning as we are looking at the variation between the batches. In the ideal case if all batches have normal behavior and the variations are due to random measurement noise then a single PC should be sufficient to capture most of covariance information of the data. One possible explanation for the compact structure of the model from the proposed methodology is, PCAIA is a multivariate technique therefore the predicted values commensurate with the overall correlation structure of the data matrix. On the other hand, 'first order hold' creates points which may have introduced extra variation in the data matrix. Therefore more PCs are required to explain this additional variation in the data.

The SPE and the T-square plots for the two methods are given in Figures 3.24 and 3.25. The SPE plot from the traditional 'DTW-first order hold' marginally detects batch number four as an abnormal batch compared to the rest of the batches. The T-square plot of the proposed missing data based method detects the sixteenth batch as abnormal. The fourth batch has a completion time of 85 mins and completion time of the sixteenth batch is 73 mins. Compared to the reference batch completion time of 95 mins the sixteenth batch feed is further off from the normal batch completion time and it has the most impurities therefore it appears more justified to single out sixteenth batch as the abnormal batch.

Though the proposed method shows good promise in off-line analysis there are several limitations of DTW and as such the proposed methodology. The biggest critique against DTW is that, on-line application of DTW is not straight forward. Our view point on this is that the monitoring does not have to be on-line throughout the processing. Rather in systems like this where the processing is going through multiple stages, after completion of each stage, PCA can be applied to find out the process status. For example, in this case three models can be built using the normal data sets. First one, using data from the heating phase only, second one using data of both the heating and cooling phase and the third one using data from all three stages. As soon we detect the completion of a stage, the data can be synchronized using the proposed methodology and the respective model may be used to detect any abnormality in the batch. This is consistent with the overall objectives of batch process monitoring. Without the measurement of the quality variable this non-invasive method may help to detect any abnormality in the processing. In the event an abnormality is detected the subsequent processing may be abandoned to avoid any additional cost.

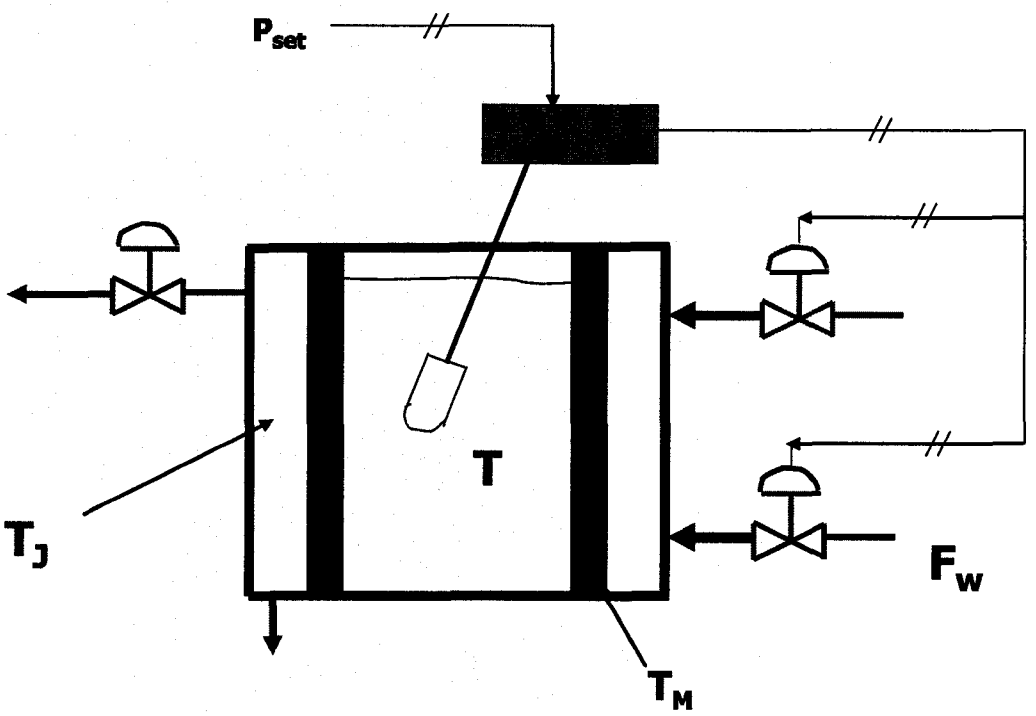


Figure 3.21: *Schematic diagram of a batch reactor*

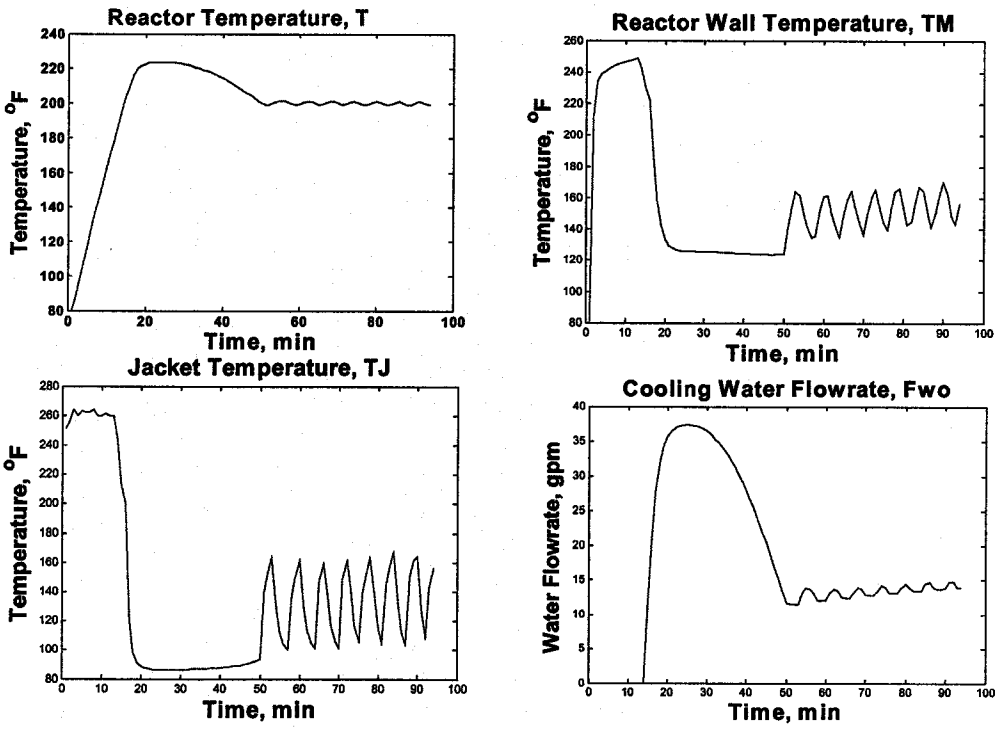


Figure 3.22: *Trend plot of the measured variables of the batch reactor*

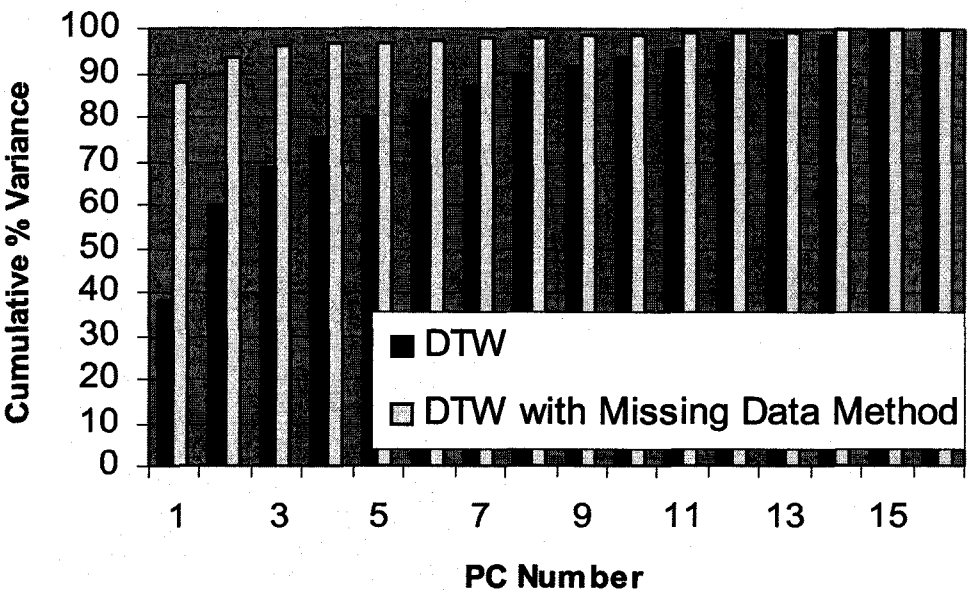


Figure 3.23: *Cumulative variance explained by the principal components calculated from un-even length batch data.*

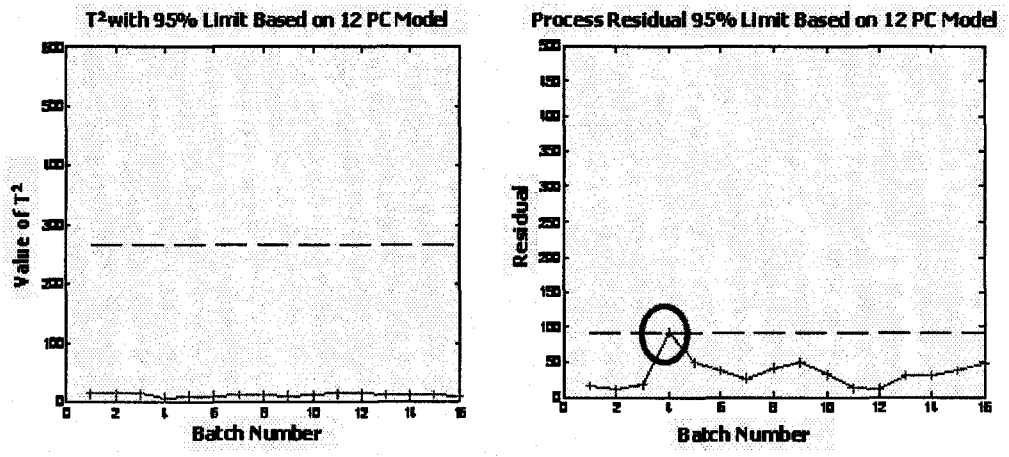


Figure 3.24: *T-Square and SPE plot obtained from the data matrix which has been synchronized using traditional DTW*

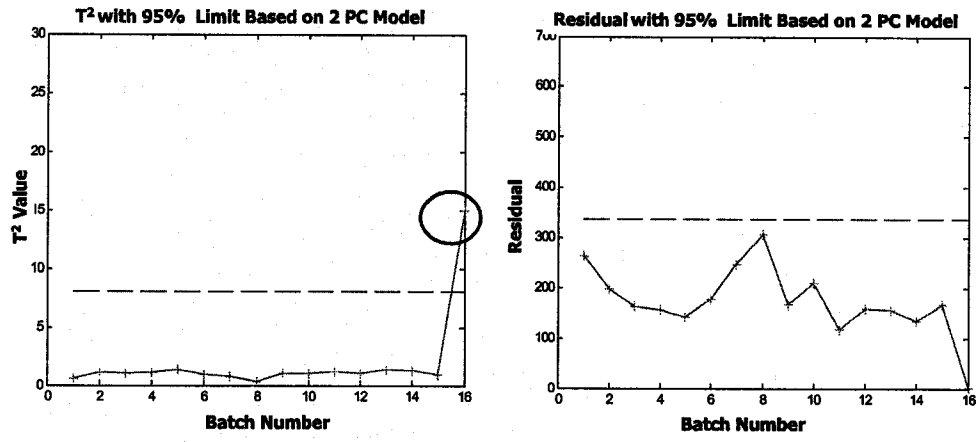


Figure 3.25: *T-Square and SPE plot obtained from the data matrix which have been synchronized using a combined DTW and missing data handling technique*

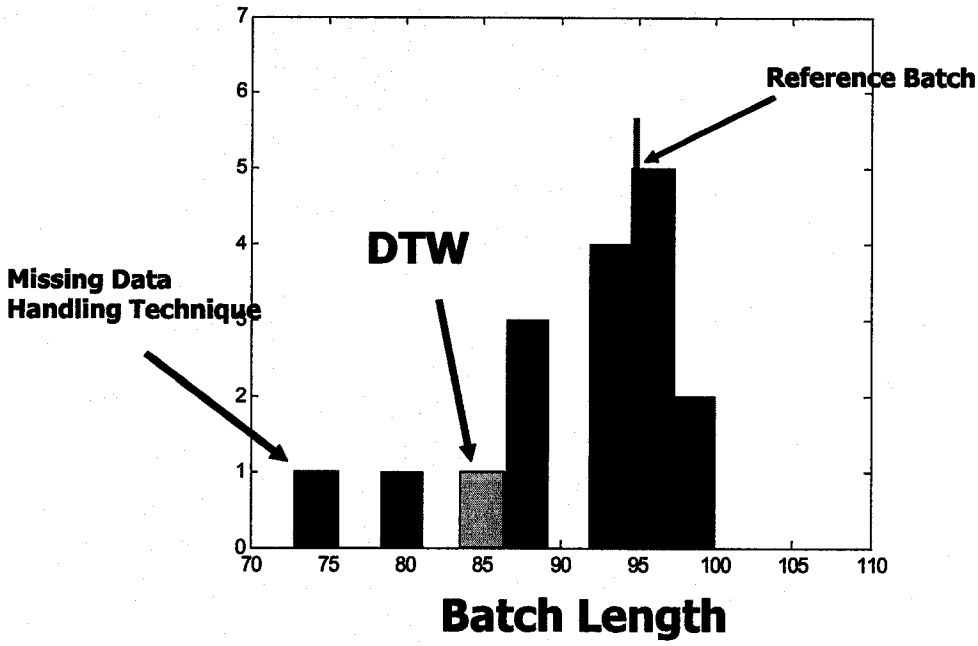


Figure 3.26: *Distribution of batch completion time: the position of the abnormal batches and the reference batch are also indicated in the plot*

3.7 Concluding Remarks

A general framework for classification of the latent variable models is proposed in this chapter. This classification can serve as a guideline to select the latent variable models for process monitoring according to the characteristics of the process. Further, the classification helps in making any modifications in the analytical techniques. In this study we extended PCA, IPCA and MLFA to the Data Augmentation framework for handling missing values in the data matrix. We applied 'Bootstrap' resampling for implementing the Data Augmentation technique. The proposed methods demonstrate better performance in preserving the correlation structure between the variables. The proposed iterative methods have good convergence properties and the estimated models have better quality compared to the models estimated using the traditional methods.

Missing data handling technique is used in combination with Dynamic Time Warping (DTW) to synchronize uneven length batch data. The proposed method conserves the correlation between the variables and thereby leads to a compact latent structure of the model.

Chapter 4

Data Compression as a Missing Data Problem

4.1 Introduction

Data compression is a widely used practice in the process industries. The current industrial practice in data archiving is to archive or store compressed data using the vendor supplied compression algorithms. As the name suggests the main objective of compression is to compress a data file to reduce the size of data file so that storage space is minimized or reduced. Compression is now redundant since storage is relatively inexpensive yet industrial practitioners continue to compress data as a default practice. However, if the main purpose of data compression is to facilitate transmission of data through telecommunication or satellites then data compression can be justified.

Whenever possible, we suggest using uncompressed data for any analysis. However, in many situations when historical data has to be analyzed for investigative purposes such as post-mortem of faults, one may have no choice other than using compressed data from the data historian. In many other situations we may be required to use compressed data for analysis for reasons such as:(1) data analyst may be located at a remote place and it may not be possible to reset the compression factor and collect uncompressed data for analysis; (2) sometimes it may be of interest to compare current performance index of a control loop with the historical performance index of the loop when the controller was originally tuned. For calculating the past performance index one has to rely on compressed data obtained from the process historian.

Although compressed data is regularly used for different analysis, it is also well known that analysis of compressed data can lead to erroneous results in data based

analysis. The effect of data compression on various univariate statistics, such as, mean, standard deviation, as well as various loop performance indicators are well studied (Thornhill *et al.* 2004). The effect of data compression on pattern matching was studied by (Singhal and Seborg 2005). In their study, the data compression algorithms were assessed on the basis of not only how accurately they represent process data but also how they affect the identification of similar patterns from historical data. However, to the best knowledge of the authors, the effect of compression on multivariate data analysis and model building has not been studied so far.

The data historian currently used in industries mostly use direct methods (for example, Swinging Door data compression) for compressing data. Such compressed data are usually reconstructed using univariate methods, such as, linear interpolation. These reconstruction methods do not take into account the changes that take place in other variables, and as such linear interpolation-based data reconstruction algorithms may destroy the correlation between different signals. So the reconstruction may not be reliable depending on the end use of the data. In particular such techniques may be potentially detrimental if the reconstructed data is used for multivariate analysis since such analysis makes use of the correlation between different variables. The main objective of this study is to investigate the impact of data compression on multivariate data analysis, specifically Principal Components Analysis (PCA). In this paper we investigate two compression algorithms, the conventional Swinging Door compression algorithm and the more recent Wavelet Compression Algorithm from a multivariate data analysis and modelling perspective. We also investigate the impact of compression on processes of different dynamical behavior, e.g., fast and slow dynamics, stochastic process. Finally, we analyze compression from a missing data point of view. A novel method, based on a missing data handling technique, is proposed to restore the correlation of the data which have been compressed using the Swinging Door compression algorithm.

4.2 Overview of Data Compression Methods

In process industries the measurements from all on-line sensors are first transmitted to the DCS systems. Most DCS systems are a repository of raw data for a short period. However, for long term storage data are first compressed and stored in the data historian. Data in its compressed form are stored as a sparse matrix of raw values or coefficients in the transformed space. Since most data analysis techniques can only deal with a complete data matrix and time domain data, it is necessary to reconstruct the compressed data to a complete data matrix in the original time domain.

Therefore each data compression algorithm also has an accompanying reconstruction algorithm. The combined compression and reconstruction is referred to as 'compression algorithm'.There is a wide variety of compression algorithms described in the literature especially in the context of image compression. Compression algorithms can be divided into two main groups:1) Direct method and 2)Transform method.

Direct methods are rule based methods which store data by looking at its deviation from the trend of the signal. Some of the popular direct methods are, piecewise linear compression (Hale and Sellars 1981), Box-Car, Backward Slope, a combination of these two methods called Box-Car-Backward-Slope (BCBS) and the Swinging Door algorithm (Bristol 1990). Direct methods make the archiving decision in real time as the data are recorded from the process. Therefore Direct methods have been the methods of choice for most industrial data archiving systems, for example, *AspenTech*® uses an adaptive method based on Box-Car-Backward-Slope(BCBS) in their data historian and *OSI*® uses a variant of Swinging Door algorithm in their PI historian (Misra *et al.* 2000, Matthew *et al.* 1998).

Transform methods perform an integral transform of the original data set and then transform it to a set of coefficients in the new space. Compression is performed on these transformed coefficients. Examples of some commonly used transforms are, Laplace transform, Fourier transform, and Wavelet transform. However, Wavelet transform is most suitable from a data compression perspective and most of the transformed compression algorithms are based on Wavelet transformation. Superior performance of Wavelet compression has been demonstrated in different context including, compression and subsequent reconstruction of process data from a paper making machine (Nesic *et al.* 1997), on-line feature extraction and noise removal from non-stationary signals (Bakshi and Stephanopoulos 1996) and pattern matching in historical data (Singhal and Seborg 2005). All these applications are off-line in the sense that compression is applied on that data set after a batch of data has been collected. An online data compression strategy using Wavelets have also been developed by (Misra *et al.* 2000). This algorithm works sequentially, i.e., with the arrival of each new point the algorithm computes all approximation coefficients and updates the multi-resolution tree. An efficient bookkeeping methodology has also been proposed, which improves compression ratios significantly over the batch or off-line version of Wavelet compression.

In this study, the Swinging Door compression and the Wavelet compression algorithms are taken as two representative algorithms from the direct and transform methods, respectively. These two algorithms are described in more detail in the following subsections.

4.2.1 Swinging Door Compression and Reconstruction

Swinging Door Compression

Swinging Door compression algorithm is based on the idea that within a signal trend it may be possible to identify many linear segments. Therefore storing only the end points of these linear segments may be sufficient to capture the main dynamics of the system. Swinging Door compression acts sequentially on each data point and therefore it can be applied in an on-line fashion to compress the data. The working principle of Swinging Door compression method is explained in Figure 4.1. Let us assume in Figure 4.1 that point c is the last recorded point. Based on the desired level of compression a distance referred to as 'compression deviation' is specified by the user. The actual amount of compression or the compression factor cannot be specified directly in the Swinging Door compression. Deviation thresholds are specified in engineering units which indirectly dictates the compression ratio. 'Compression Deviation' is used to calculate the two limiting points, a and b from point c by adding and subtracting the deviation distance respectively. Points a and b are called pivot points. As each new spot value arrives, lines are drawn from the pivot points to form a triangular envelope that tries to enclose all the spot values since the last recorded point, c. The sides of the triangle are the 'doors'. For instance, in Figure 4.1, aa' and bb' are two doors that envelope all points up to point d in a triangle. However, the next sample e cannot be enclosed in the triangular envelope. The deviation of sample e from the linear trend of the signal is such that, to encompass it within the 'doors', the 'doors' have to be rotated wider than as shown by aa' and bb''. Therefore the 'doors' will not form a triangle encompassing sample e, which signifies that a new trend has started after point d. The first and the last samples from the previous trend, i.e., point a and d are archived. New upper and lower bounds are calculated at point d and the procedure is repeated from point d onwards. In the compressed form, only the end points of linear trends and the corresponding sample numbers are stored in the process historian.

Swinging Door Reconstruction

The Swinging Door algorithm uses a linear interpolation method to reconstruct the signal. Therefore the reconstructed signals will have many linear segments in between the raw data points. For example in Figure 4.1, point c and point d are the only two originally stored spot values. Linear interpolation will create intermittent points at regular time interval as specified by the user. The interpolated points will lie on the straight line connecting points c and d. Clearly the reconstruction criteria is

to minimize the deviation of the reconstructed signal from the actual signal and not aimed towards preserving the variance of the signal or the correlation between different variables in the reconstructed signals. Since multivariate analysis makes use of the correlation between the variables, linear interpolation type reconstruction is clearly unsatisfactory for such analysis.

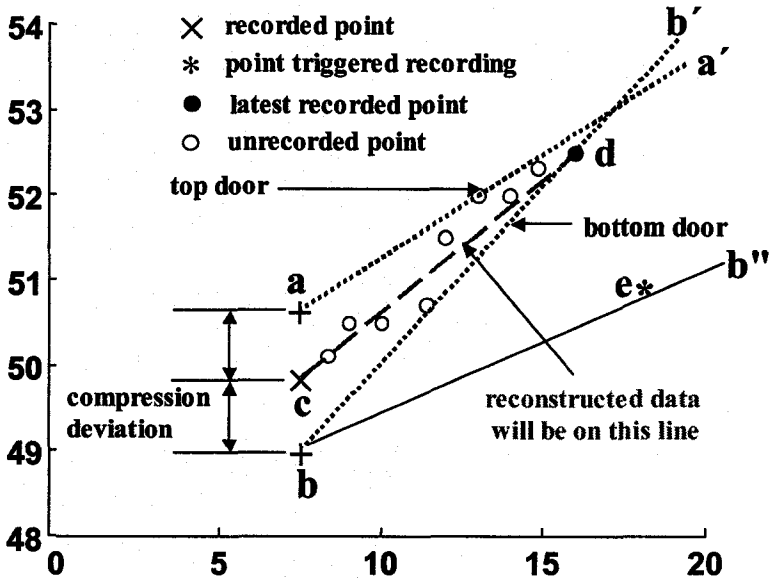


Figure 4.1: *Schematic representation of the Swinging Door algorithm for data compression*

4.2.2 Wavelet Compression and Reconstruction

Wavelet compression and reconstruction is based on Wavelet Transform and Inverse Wavelet Transform respectively. The main objective of Wavelet Transform is to locate a frequency component as well as the exact time of occurrence. In this sense it is very similar to Short Time Fourier Transform (STFT). However, a Wavelet transform does it more efficiently by dividing data, functions, or operators into different frequency components and then processing each component with a resolution matched to its scale. For example, a high time resolution (narrow window) is used for high frequency signals and low time resolution (wide window)is used for low frequency signals. During data compression only the high frequency information is lost. This is commensurate with the needs of the process and control engineers since most of the high frequency signals come from disturbances, are short lived and not of interest. On the other

hand, process dynamics are mostly in low frequency region and persist throughout the duration. Wavelet compression reconstruction is implemented in three main steps: (i) Wavelet Transform (ii) Thresholding and (iii)Inverse Wavelet Transform. The transformation of a signal during these steps is shown in Figure 4.2.

Wavelet Transform

The raw signal *x(t)* is represented in terms of Wavelet coefficients $\Psi_x^{\psi_{j,k}}$. This is done by taking the Wavelet transform of the signal using a particular mother Wavelet and a level of decomposition (Equation 4.1). The number of coefficients generated during this transformation is the same as the number of data points.

$$\Psi_x^{\psi_{j,k}} = \int x(t)\psi_{j,k}(t)\,dt \tag{4.1}$$

where the mother Wavelet $\psi_{j,k}$ is given by,

$$\psi_{j,k}(t) = s_0^{j/2}\psi\left(s_0^j t - k\right) \tag{4.2}$$

integer j represents the scale at which the signal is decomposed and k represents its position in time.

Mother wavelets are short lived functions which enhance the time localization. The most common mother Wavelet functions used in the literature are the Haar function and the Daubechies family of orthogonal Wavelet functions. The Haar Wavelet function has been used in this paper for Wavelet data compression (Polikar 2005). It is a square wave defined as follows:

$$\begin{aligned} \psi_{j,k}(t) = &\quad 1 \quad t \in \left[0,\ \frac{1}{2}\right] \\ = &\quad -1 \quad t \in \left[\frac{1}{2},\ 1\right] \\ = &\quad 0 \quad t \notin [0,\ 1] \end{aligned} \tag{4.3}$$

At this stage still there is no loss of information. Using Inverse Wavelet the original signal can be reconstructed from the coefficients. However, the disk space occupied by the coefficients is also the same as the original signal.

Thresholding

Compression or reduction of file size is achieved during the thresholding stage. Coefficients whose magnitudes are less than a predefined threshold ϕ are set to zero. The threshold limit ϕ is calculated iteratively based on the desired level of compression.

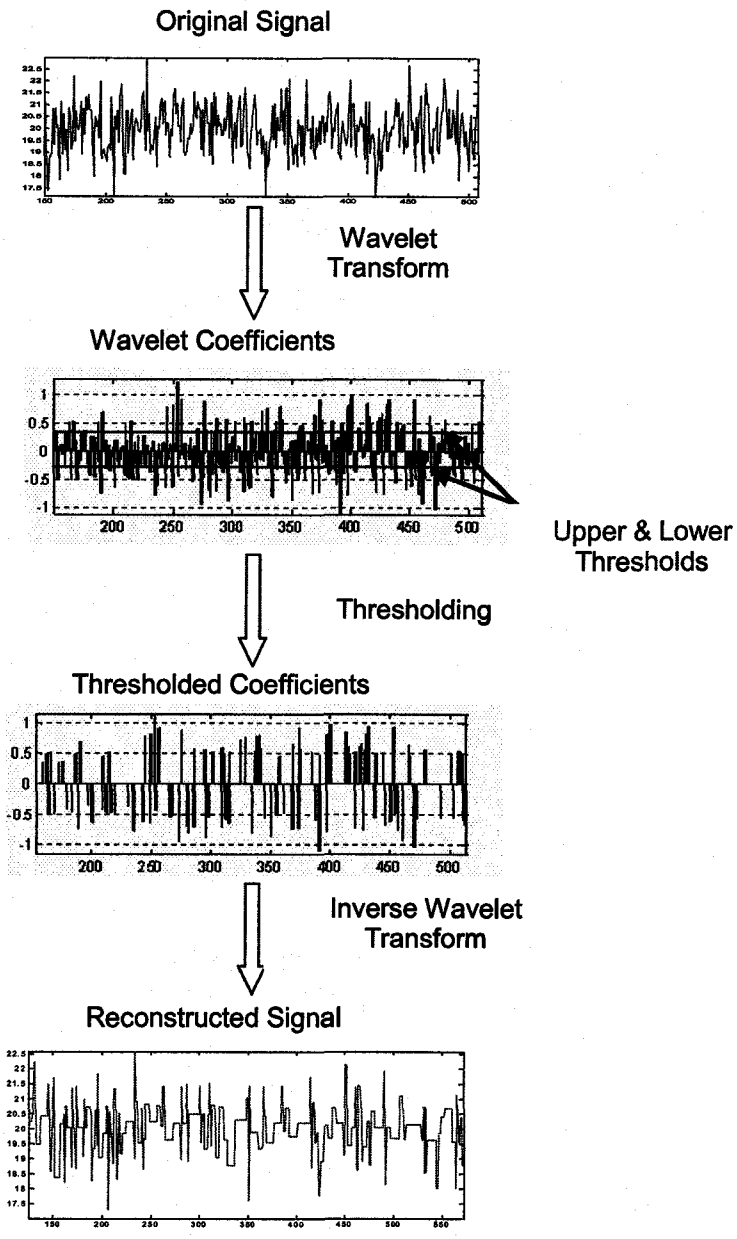


Figure 4.2: *Schematic representation of Wavelet Compression and Reconstruction Algorithm*

The threshold limit can also be calculated by comparing the variance information of the original signal and the reconstructed signal. Thresholding will lead to many zeros in the data matrix. These zeroes are neglected and only the non-zero coefficients are stored. However, it is also important to record the locations or index numbers of the coefficients for reconstructing the signal. The non-zero coefficients are stored in the form $sign(w_k)(k + (|w_k|/||w_k||_\infty))$ where k is the index number and w_k is the Wavelet coefficient at location k. Together with the scaled non-zero coefficients in this form, the length of the original vector and the maximum value of the coefficients, $||w_k||_\infty$ are stored, and Compression Ratio is given by,

$$compression\ ratio = \frac{length\ of\ original\ vector}{number\ of\ nonzero\ coefficients + 2} \tag{4.4}$$

These thresholded coefficients are the compressed form of the data and are stored in the data historian. However, when the coefficients are neglected, the transformation is no longer lossless and the reconstructed signal from these coefficients will be different from the original signal.

Wavelet Reconstruction

The first step of reconstruction is transformation of the coefficients to their absolute values. The integer parts of the scaled coefficients give the index numbers and the decimal parts are essentially the scaled coefficients. The decimal parts are multiplied by $||w_k||_\infty$ and the appropriate signs are placed in front of the coefficients. The coefficients are placed at their respective positions in the data vector as indicated by the index numbers. All points in between the coefficients are filled with zeros. Inverse Wavelet transform (Equation 4.5) is applied on this coefficient vector with the same mother Wavelet and the same level of decomposition originally used for compressing the data set (Singhal and Seborg 2005). Data vector $x(t)$ is the decompressed signal in the time domain.

$$x(t) = c_\psi \sum_j \sum_k \Psi_x^{\psi_{j,k}} \psi_{j,k}(t) \tag{4.5}$$

4.3 Formulation of Compression as a Missing Data Problem

Data historians used in process industries almost exclusively use direct methods for compressing data. In this section we will formulate compressed data from direct methods as a missing data problem. Process historians use decompression algorithms

to provide a data matrix with the specified sampling rate. These decompression methods mostly use linear interpolations to fill the points in-between the originally stored spot values. In order to cast the problem in a missing data formulation, the first step is to take out the interpolated points. Only the spot points are retained and subsequently used for building the model using multivariate missing data handling techniques.

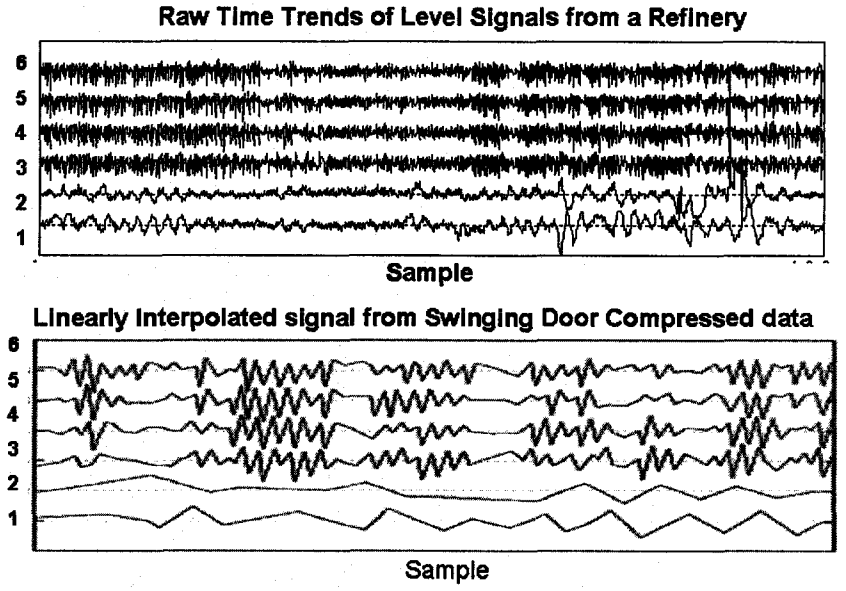


Figure 4.3: *Data from several loops of a refinery process archived using a Swinging Door compression algorithm to a factor of 10 and subsequently reconstructed using the built-in reconstruction algorithm. The reconstructed signals show many linear segments.*

For example, the reconstructed signals from Swinging Door compression algorithm are shown in Figure 4.3. The measurements from the level loops of a distillation column were compressed by a factor of three using Swinging Door compression algorithm. The signals were reconstructed using linear interpolation. Data matrix of the corresponding signals are shown in Figure 4.4(a), where the linearly interpolated points are replaced with 'NaN'. This shows the distribution of the originally stored spot values. Some of the rows do not contain a single spot value in the row. These rows have been shaded in the data matrix. Since these rows do not contain any information they were removed and the new data matrix is shown in Figure 4.4(b). This is the missing data formulation of the compressed data, where the missing values exist all over the data matrix. Multivariate missing data handling techniques may be used to predict

(a)

18.915	10.199	29.486	30.427	19.639	19.639
20.241	9.0755	NaN	NaN	20.288	20.288
16.505	7.7958	26.58	26	18.539	18.539
NaN	NaN	NaN	NaN	NaN	NaN
20.931	11.231	32.214	31.923	20.941	20.941
22.26	10.549	32.853	32.312	21.679	21.679
NaN	13.139	32.418	32.736	NaN	NaN
18.484	NaN	NaN	NaN	18.877	18.877
21.029	10.2	31.665	NaN	20.87	20.87
NaN	NaN	NaN	30.56	NaN	NaN
19.052	12.804	NaN	NaN	NaN	NaN
NaN	NaN	32.119	32.276	19.494	19.494
NaN	9.8168	NaN	NaN	NaN	NaN
20.95	NaN	31.356	31.186	21.305	21.305
NaN	8.4052	28.349	27.593	NaN	NaN
18.455	NaN	NaN	NaN	18.735	18.735
NaN	NaN	29.101	NaN	NaN	NaN
NaN	10.087	NaN	30.08	NaN	NaN
21.213	NaN	33.521	NaN	20.886	20.886
NaN	13.015	NaN	33.423	NaN	NaN
NaN	8.5436	28.526	28.927	NaN	NaN
20.217	NaN	NaN	NaN	19.898	19.898
NaN	9.5583	NaN	30.567	NaN	NaN
NaN	NaN	NaN	NaN	21.527	21.527
NaN	10.053	31.034	30.56	NaN	NaN
20.579	NaN	NaN	NaN	NaN	NaN
NaN	9.539	29.484	29.418	NaN	NaN
NaN	13.166	33.806	33.689	20.463	20.463
19.614	NaN	NaN	NaN	NaN	NaN
NaN	8.2582	27.895	27.337	NaN	NaN
NaN	NaN	NaN	30.397	NaN	NaN
19.559	11.375	30.664	31.456	19.307	19.307

(b)

18.915	10.199	29.486	30.427	19.639	19.639
20.241	9.0755	NaN	NaN	20.288	20.288
16.505	7.7958	26.58	26	18.539	18.539
NaN	NaN	NaN	NaN	NaN	NaN
20.931	11.231	32.214	31.923	20.941	20.941
22.26	10.549	32.853	32.312	21.679	21.679
NaN	13.139	32.418	32.736	NaN	NaN
18.484	NaN	NaN	NaN	18.877	18.877
21.029	10.2	31.665	NaN	20.87	20.87
NaN	NaN	NaN	30.56	NaN	NaN
19.052	12.804	NaN	NaN	NaN	NaN
NaN	NaN	32.119	32.276	19.494	19.494
NaN	9.8168	NaN	NaN	NaN	NaN
20.95	NaN	31.356	31.186	21.305	21.305
NaN	8.4052	28.349	27.593	NaN	NaN
18.455	NaN	NaN	NaN	18.735	18.735
NaN	NaN	29.101	NaN	NaN	NaN
NaN	10.087	NaN	30.08	NaN	NaN
21.213	NaN	33.521	NaN	20.886	20.886
NaN	13.015	NaN	33.423	NaN	NaN
NaN	8.5436	28.526	28.927	NaN	NaN
20.217	NaN	NaN	NaN	19.898	19.898
NaN	9.5583	NaN	30.567	NaN	NaN
NaN	NaN	NaN	NaN	21.527	21.527
NaN	10.053	31.034	30.56	NaN	NaN
20.579	NaN	NaN	NaN	NaN	NaN
NaN	9.539	29.484	29.418	NaN	NaN
NaN	13.166	33.806	33.689	20.463	20.463
19.614	NaN	NaN	NaN	NaN	NaN
NaN	8.2582	27.895	27.337	NaN	NaN
NaN	NaN	NaN	30.397	NaN	NaN
19.559	11.375	30.664	31.456	19.307	19.307

Figure 4.4: *(a)Data matrix after the linearly interpolated points have been replaced by 'NaN'(b)Data matrix after removing the shaded rows from data matrix (a)*

missing values in such a data matrix.

4.3.1 Characterization of Compression Mechanism

In order to reconstruct the missing values, it is important to characterize the mechanism that generated the missing values. In the missing data literature, mechanisms are classified in three categories: i)Missing Completely At Random (MCAR),ii) Missing At Random (MAR) and iii)Non Ignorable (NI) mechanism. Details of the definitions can be found in (Rubin 1977, Little and Rubin 2002). These classifications provide a guideline for reconstruction and possible implications of any assumption. Here we give a brief description of these mechanisms to classify compression in light of these definitions. As shown in Figure 4.4(b), the original spot values and missing values are spread all over the data matrix. Any data set, $Y = (y_{ij})$, containing an observed part and a missing part is represented as $Y = (Y_{obs}, Y_{mis})$. This notation will also be used in this paper. A matrix $M = (m_{ij})$, referred as the missingness matrix, is used for indexing the missing and the observed part. Each element of M is a single binary item indicating whether y_{ij} is observed ($m_{ij} = 1$) or missing ($m_{ij} = 0$). In

the statistics literature, missingness is treated as a random phenomena. The distribution of M, called the missingness mechanism, is characterized by $f(M|Y,\phi)$, the conditional distribution of M given Y, where ϕ denotes parameters unrelated with Y. Classification of missingness mechanism is based on the conditionalities:

1. Missing Completely At Random(MCAR)

   In this case missingness does not depend on any part of the data Y either missing or observed.

   $$f(M|Y,\phi) = f(M|\phi)$$

2. Missing At Random(MAR)

   Missingness depends only on the observed component Y_{obs} and not on the missing component Y_{mis} of the data matrix.

   $$f(M|Y,\phi) = f(M|Y_{obs},\phi)$$

3. Non Ignorable Mechanism (NI)

   If the mechanism of missingness is dependent on both the observed and the missing part of the data then the mechanism is Non Ignorable.

Under MCAR and MAR conditions the mechanism that led to missing data can be ignored in the reconstruction process. Any model based on the observed data will give reasonable reconstruction. For Non-Ignorable cases the missing mechanism has to be taken into account in the reconstruction of missing data. In many cases it is not possible to include the mechanism in the reconstruction process. To circumvent this, it is customary to assume data as MAR and build the model only based on the observed part of the data at the expense of some accuracy.

Now let us analyze where the compression mechanism stands according to the above definitions. While storing a value using Swinging Door, the deviation of the value from the linear trend is calculated. If the point is outside the desired bound only then is it archived, otherwise it is discarded. So essentially the distribution of M will be dependent on both Y_{obs} and Y_{mis}. Therefore, from the view point of missing data, compression is a 'Non-Ignorable(NI)' mechanism. This indicated that a method that is inverse to the compression algorithm should be used for exact reconstruction of the signal. However, the mechanism used for compressing the data is an irreversible one and it is not possible to include it in any form in the multivariate reconstruction process. In the absence of any such mechanism, we will proceed with

methods developed based on 'Missing at Random (MAR)' and use the model based on the observed data to reconstruct the missing part of the data matrix. Since there is a probability that the observed part of the data may be systematically different than the missing part, it may introduce some error in the model or the reconstructed signal. However, this is the best practice in this situation.

Due to compression the percentage of missing data is very high, for example, for a compression factor of 3 approximately 66% of the data is missing and at a compression factor of 10 only one out of ten points is recorded which means 90% of the data is missing. So, from a missing data view-point compression can be seen as Non-Ignorable mechanism with a very high percentage of missing data.

4.4 Reconstruction of Swinging Door Compressed Data using PCAIA

PCA based missing data handling techniques have been used to reconstruct small amount of missing data and perform PCA based process monitoring in the presence of missing values in the data matrix. The details of the methods can be found in (Nelson and MacGregor 1996, Grung and Manne 1998). However, in the current study we show the application of the methods in a completely new context. Missing data handling method is used for restoring the correlation structure and building a multivariate model from compressed data. First, data compression is cast as a missing data problem and subsequently the Principal Component Analysis Iterative Algorithm (PCAIA) (Grung and Manne 1998) is used for building the model. Reconstructing signals from compressed data using missing data handling techniques is challenging because most of the techniques are not suitable for dealing with such high percentage of missing data. However, if used judiciously, missing data handling techniques can be useful in extracting the true correlation between the variables. The implementation steps of the algorithm are shown via a flow diagram in Figure 4.5. The method is suitable for working with compressed data from any direct method. The retrieved data matrix from the process historian contains some originally stored spot values and linearly interpolated points in between them. The first part of the reconstruction algorithm is to find the original stored points. To find these data points a compression detection algorithm was used (Thornhill *et al.* 2004). This algorithm can find the spot values from signals which were reconstructed using linear interpolation. Since the reconstructed signal is piecewise linear, it will have discontinuity only at the locations of the spot values. Therefore, the locations of the spot values are given by the locations of the non zero double derivatives. Second derivatives are calculated

at each point of the signal using the difference relationship shown in Equation 4.6.

$$\begin{aligned} \Delta\left(\Delta\hat{y}\right)_i &= \frac{\left(\hat{y}_{i+1}-\hat{y}_i\right)/_h - \left(\hat{y}_i-\hat{y}_{i-1}\right)/_h}{h} \\ &= \frac{\hat{y}_{i+1}-2\hat{y}_i+\hat{y}_{i-1}}{h^2} \end{aligned} \tag{4.6}$$

where $\hat{y}_i$ is the reconstructed signal and h is the sampling interval. If N is the total length of the signal, index i ranges from 2 to $(N\text{-}1)$. Only the spot values are retained, and the rest of the points in the data matrix are considered as missing. This is illustrated in Figure 4.4(a) where the missing values have been indicated by 'NaNs'. However, at this stage the 'percentage of missing data' in the data matrix would be high since in many situations we may encounter highly compressed data, e.g., for a compression factor of five, 80% of the data would be missing. This poses difficulty in reconstruction as most iterative missing data handling techniques do not converge for more than 20% missing data in the data matrix. Therefore a multistage procedure is applied to bring down the percentage of missing data in the data matrix. In the first step all rows which do not have any original points are taken out of the data matrix. Clearly, these rows do not contain any information and it will not have any impact on process models as only steady-state models are of interest. This is illustrated in Figure 4.4(a) where all the rows which do not contain a single spot value are shaded. This data matrix was obtained from a data set which was compressed by a factor of three. Therefore 66% of the data are missing at this stage. After removing the rows which do not contain a single spot value, i.e., the shaded rows in Figure 4.4(a), the new data matrix takes the form shown in Figure 4.4(b). The ratio of spot values to missing values improved at this stage and the 'percentage missing data' in the new data matrix(Figure 4.4(b)) reduced to '50%'. In the next phase, rows which contain only one spot value are taken out of the data matrix. This will help to further reduce the percentage of missing values in the data matrix. The procedure is repeated until the percentage of missing values in the data matrix comes down to 30%, e.g., removing rows with two original values in the next step. The PCA based missing data handling technique gave good estimates of model and the iterative algorithm converged well up to 30% of missing values in the data matrix. After doing extensive simulation studies we arrived at this number. However, it is not possible to take out all the missing values and create a complete data matrix, because the original spot values of different variables are not aligned with each other. If only complete rows are retained it will drastically reduce the sample size. After the percentage of missing data is within 30%, Principal Component Analysis Iterative Algorithm (PCAIA) is used to

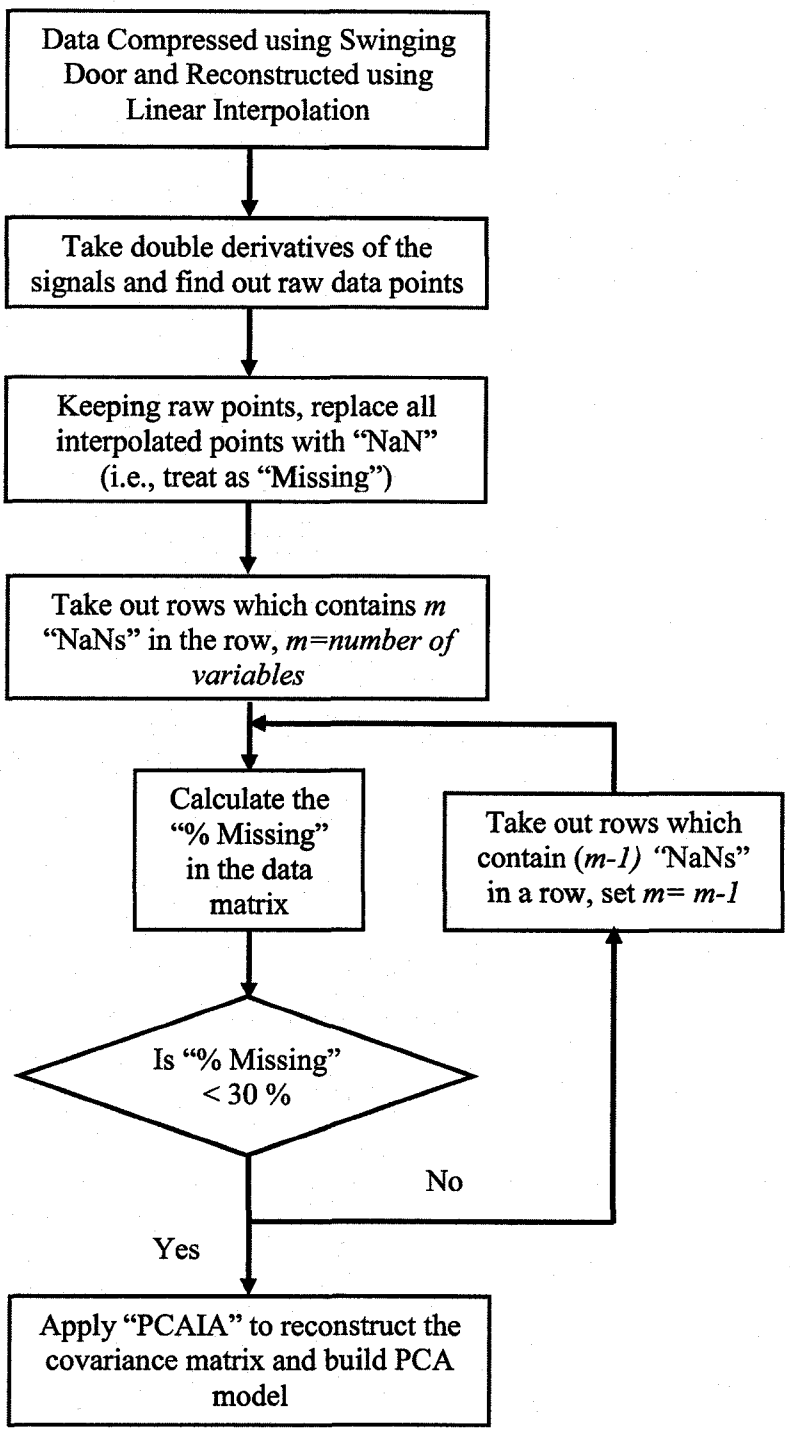


Figure 4.5: *Proposed algorithm for building PCA model from Swinging Door Compressed and Linearly Reconstructed Data*

restore the correlation structure and build PCA model from the data matrix. The implementation of PCAIA is carried out as follows:

1. The missing values of the data matrix are filled with the unconditional mean of the variables. For example, the missing values of the data matrix are filled by the column averages of $\boldsymbol{Y}_{obs}$ which gives the augmented data matrix $\boldsymbol{Y}_{aug} = [\boldsymbol{Y}_{obs}, \boldsymbol{Y}_{mis}]$ where $\boldsymbol{Y}_{mis} = mean(\boldsymbol{Y}_{obs})$ and $\boldsymbol{Y}_{aug} \epsilon \mathbb{R}^{N \times n}$.

2. Singular Value Decomposition (SVD) is performed on the augmented data matrix. The loading matrix $\boldsymbol{P}$ is used to predict the noise free values $\hat{\boldsymbol{X}} = \boldsymbol{Y}\boldsymbol{P}\boldsymbol{P}^T$.

3. Missing values are filled with predicted values, $\hat{\boldsymbol{X}}$ and the augmented data matrix will be $\boldsymbol{Y}_{aug} = [\boldsymbol{Y}_{obs}, \hat{\boldsymbol{X}}_{mis}]$, where $\hat{\boldsymbol{X}}_{mis}$ are predicted values in the previous step.

4. Convergence is monitored by observing the sum of squared errors between the observed values and corresponding predicted values from step (2).

$$SSE_{obs} = \sum_{i=1}^{N} \sum_{j=1}^{n} \left(Y_{ij} - \hat{X}_{ij} \right)^2_{obs}$$

   Step (2) and step(3) are repeated until convergence.

In the current study we assumed that the model order or the dimensions of the loading matrix $\boldsymbol{P}$ are known. However, in many real applications the model order may not be known exactly. Because of missing data, the percentage variance explained by the PCs becomes a function of missing data and model order selection gets complicated. In the presence of missing values a cross-validation based detailed method is incorporated into the algorithm to find out the model order (Walczak and Massart 2001).

Remarks

PCAIA is a pseudo version of the more general Expectation Maximization (EM) algorithm (Dempster and Rubin 1977). Here it may be interesting to explore the link with EM. Similar to EM we can identify the two major iterative steps of the algorithm.

Parameter Estimation step is similar to the Maximization (M-Step) of the EM algorithm. From the augmented data matrix, where missing values are filled with conditional expected values, the loadings of the PCs are calculated. These are the parameters in this case. However, the method is optimal in the least squares sense contrary to the Maximum Likelihood Estimates obtained in EM.

Missing Value Estimation resembles the Expectation step of the EM algorithm. Using the estimated parameters, missing values are estimated in this step. These values are used to fill the missing values and get a better augmented data matrix. In the Expectation step of the EM algorithm missing values are not directly estimated, rather the expectation of the sufficient statistics of the log-likelihood function are calculated. Therefore, the two methods will be only equivalent when the log-likelihood is linear in data or in other words the sufficient statistics of the log-likelihood equation are function of the data values only.

4.5 Results and Discussions

The results of the analysis are demonstrated using two examples, a simulated flow-network system and an industrial case study. The industrial data is taken from a petroleum refining process. The description of the Flow-network system and the refinery data are given below:

4.5.1 Simulation Example

Data generated from this simulated system were compressed using both Swinging Door and Wavelet compression algorithms and subsequently decompressed using the commonly used built-in reconstruction methods, and also the proposed PCAIA. To investigate the effect of compression on model quality, PCA models were built from the decompressed data sets and the estimated models were compared with the true model. The total data length for current study is 2000 samples.

4.5.2 Industrial Case Study

The industrial data used in this analysis were obtained from a petroleum refining process. All six variables are level measurements at different locations of a distillation column. The sampling time interval for the data is 60 sec and the total length of the data set is 20000 samples. The data was obtained in uncompressed form. For investigative purpose it was compressed to different compression levels. Due to the proprietary nature of the refining process, no process information is provided here.

4.5.3 Performance measure for model quality

Principal Component Analysis (PCA) is obtained by the Singular Value Decomposition (SVD) of the covariance matrix where the loadings of the PCs are given by the eigenvectors. In a multidimensional problem the eigenvectors can be multiplied using

any non-singular matrix to define the same hyperplane. The exact value of each of the element depends on how the basis vectors are selected. So a direct comparison of the parameter values with actual model parameters is not feasible. Instead one should examine it the hyperplane defined by the estimated model is in agreement with the actual model hyperplane. In this study the subspace angle, θ is used to measure such agreement. The calculation details of subspace angle is given in Section 3.5.2. In reality the exact value of A is seldom known so subspace angle cannot be used for monitoring convergence. Convergence of PCAIA was monitored using the calculated sum squared errors of the observed values and corresponding predicted values. In addition to that, subspace angle was used to reaffirm the claims made about the performance of the algorithm.

4.5.4 Effect of Compression on Correlation Structure

Almost all multivariate statistical data analysis methods, for example, pattern matching of historical data, fault detection and isolation using PCA, make use of the correlation between the variables. It is important to understand how compression affects the correlation structure of the data. A variety of industrial data has been used to visualize the effect of compression on correlation structure. The petroleum refining example described earlier will be stated here. The correlation matrix of the raw uncompressed data set is mapped in the color coded plot in Figure 4.6(a). The colors in the color-map indicate the magnitude of the correlation. This data set was compressed using Swinging Door and Wavelet compression algorithms to a compression factor of 10 and subsequently reconstructed using linear interpolation and Inverse Wavelet Transform respectively. The correlation color-map of the reconstructed data are shown in Figures 4.6(b) and (c) respectively. It is evident from the correlation color map that in the process of compression via the Swinging Door algorithm and linear reconstruction, the correlation between the variables has been severely distorted at this level of compression and the structure is significantly different from the true correlation structure shown in Figure 4.6(a). On the other hand, reconstructed data from Wavelet compression retains the true correlation structure in most parts. Although Wavelet compression is able to retain most of the significant correlation structure of the data, none of the current commercially available data historians use Wavelet Compression Algorithms. Swinging Door Compression or similar direct methods are used almost exclusively by commercial process historians. Therefore, in order to use the Swinging Door compressed data, especially for multivariate analysis, alternative methods should be used to reconstruct the compressed data so that it retains the true correla-

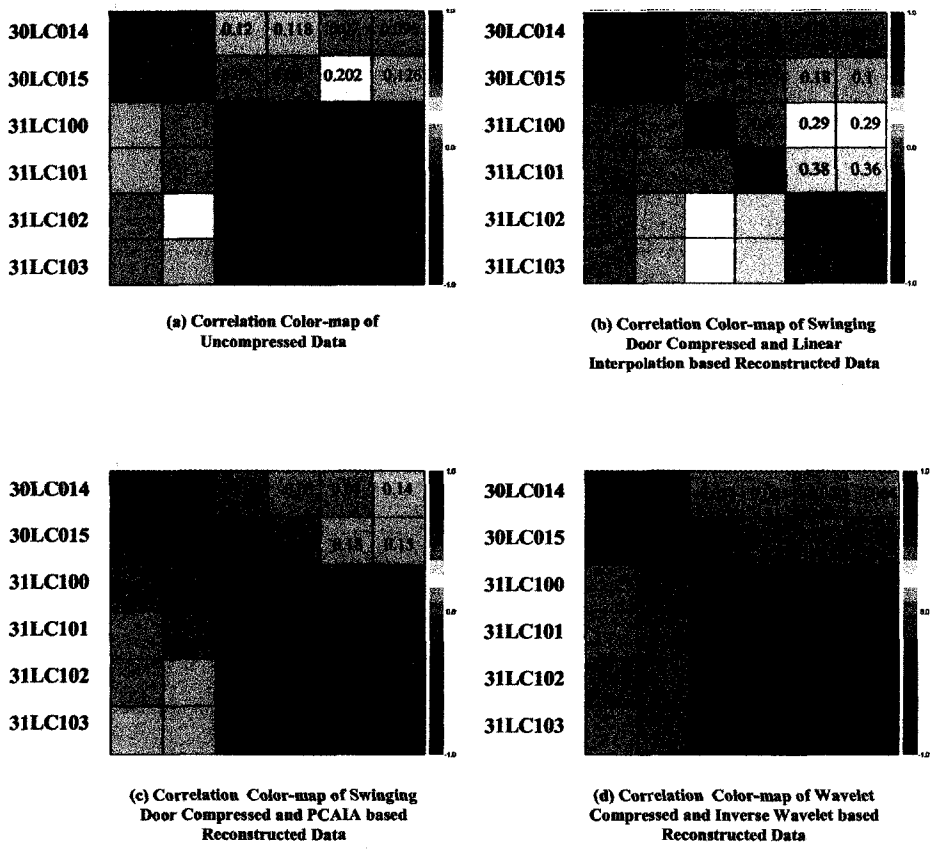


Figure 4.6: *Correlation color map of variables from a petroleum refining process. The intensity of the color shows the level of correlation between the variables.(It is recommended that this figure be viewed in color*

tion structure between the variables. Instead of linear interpolation based methods, it is recommended that PCAIA be used to reconstruct the Swinging Door compressed data set. The correlation structure of the reconstructed data using PCAIA is shown in Figure 4.6(c). A comparison of Figures 4.6(a) and 4.6(c) shows that the PCAIA based reconstruction significantly restores the true correlation between the variables.

4.5.5 Compression and Process Dynamics

In order to get a quantitative measure of the interaction between compression and process dynamics, a parametric study was conducted using the Flow-network system. The independent flow-rates were generated using transfer functions given in

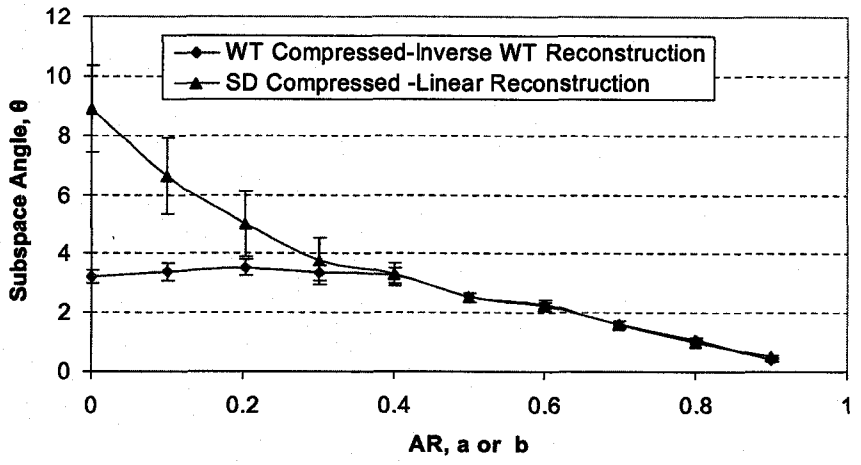


Figure 4.7: *Variation of subspace angle with the change of the dynamic behavior of the flow-network system. Swinging Door Compressed data was reconstructed using Linear Interpolation and Wavelet Compressed data was reconstructed using Inverse Wavelet Transformation and subsequently used for a building model*

Table 3.5.1. The input to the transfer function is a Gaussian random signal. Different dynamic behavior of the process were simulated by varying coefficients a and b from 0 to 0.9. As the coefficients vary from 0 to 0.9, the process gradually moves from a completely stochastic system to a slowly moving autoregressive process. All the signals were compressed by a factor of 3 using both Swinging Door and Wavelet compression algorithms. The signals were then decompressed using linear interpolation and Inverse Wavelet Transform respectively, and subsequently models were built from these decompressed data sets. Figure 4.7 shows the deviation of the estimated models (i.e. subspace angle) from true model with the change of the process dynamics. It is evident from the results that, the effect of compression is more severe on the multivariate model when the process exhibits faster dynamic behavior. However, as the individual signals become more predictable the effect in multivariate model building also gets minimal. The estimated model from 'Wavelet Compressed and Inverse Wavelet reconstructed' data has a smaller subspace angle than 'Swinging Door compressed and Linearly Interpolated' data in this region. However, as the coefficients of the AR models increase beyond 0.3, the subspace angles for the estimated models from both methods become equal. So the quality of the models are similar for processes with slow dynamics.

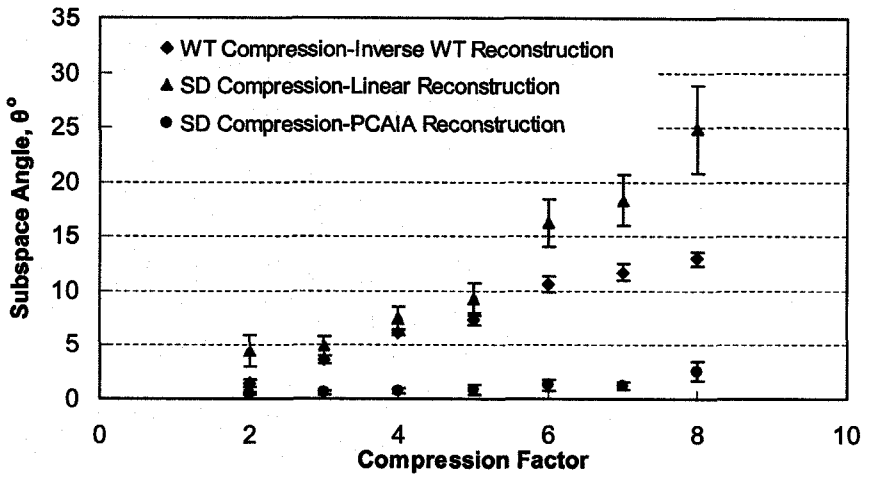


Figure 4.8: *Variation of subspace angle with compression ratio. Compressed data from flow-network system was reconstructed using the three reconstruction methods, and subsequently used for building model*

4.5.6 Improving model quality using Missing Data Handling Technique

In the previous section it was observed in the Flow-network system that, compression severely affects the model quality if the auto regressive coefficients are below 0.3. In this section we compare the performance of the proposed PCAIA method with the linear interpolation method in building a PCA model from Swinging Door compressed data. We also plot the subspace angle of the models obtained from data which were compressed using Wavelet Transform and reconstructed using Inverse Wavelet Transform.

Flow-network Example

The flow-rates x_1 and x_2 of the flow-network system are the output of the transfer functions given in Table 3.5.1 with coefficients $a = b = 0.3$. The methodology of building a PCA model from compressed data using missing data handling technique has been outlined in Section 4.4. Results of the analysis are presented in Figure 4.8. It is evident from the figure that estimated model from 'Wavelet Compressed and Inverse Wavelet Reconstructed' and 'Swinging Door Compressed and Linearly Reconstructed' data have poor quality as the data is compressed beyond a compression factor of 3. On the other hand, PCAIA based modeling provides minimum sub-

space angle, i.e., the best model quality among the three methods. Models estimated using PCAIA has better quality up to compression ratio as high as 8. It clearly demonstrates that instead of using linear interpolation to reconstruct Swinging Door compressed data, use of PCAIA can be significantly beneficial in terms of a model that preserves the multivariate relationships between the variables. The main reason for the improvement is that, in PCAIA the missing values due to compression were reconstructed in a multivariate framework. As a result, the method accounted for the changes that took place to other variables as well. On the other hand, in linear interpolation a signal is reconstructed in a univariate framework, i.e., using only that particular variable, thus the reconstruction is not reliable if changes occur in other correlated variables at those instants. In those cases linear interpolation will miss the excitations and capture only the average behavior of the signal.

Since compression leads to high percentage of missing data the convergence of the iterative algorithm is an important concern. For the Flow-network example, the true model was available, so the change of model quality (i.e. subspace angle) at each iterative step was tracked. The subspace angle, as a measure of model accuracy, has been plotted against iteration number in Figure 4.9. The plot shows a monotonic convergence of the subspace angle at each successive step. The algorithm converges in less than 10 iterations even for highly compressed data. However, when the percentage of missing data is more than 40% (not shown in the figure) or the process does not have sufficient excitation in those cases some divergent behavior was also observed. In those cases more stringent criteria has to be set and only rows with very few missing values should be retained, so that the percentage of missing data is within the manageable range. Similar to any data driven modelling the current methodology also assumes that the process is sufficiently excited.

Industrial Case Study: Refinery Data

The refinery data set used for correlation structure analysis is also used to investigate the performance of different compression algorithms and PCAIA, in a multivariate modelling context. Since this is an industrial data set the actual model of the process is unknown. In order to get a performance metric of model accuracy, first a benchmark model is built from the uncompressed raw data set. Subsequently models estimated from the reconstructed data sets are compared with this benchmark model. The percentage of total variance explained by the PCs calculated from the original uncompressed data with that from various reconstructed data are plotted in Figure 4.10. The eigenvalue distribution of the Swinging Door Compressed and Linearly reconstructed data set is quite different from the uncompressed data set.

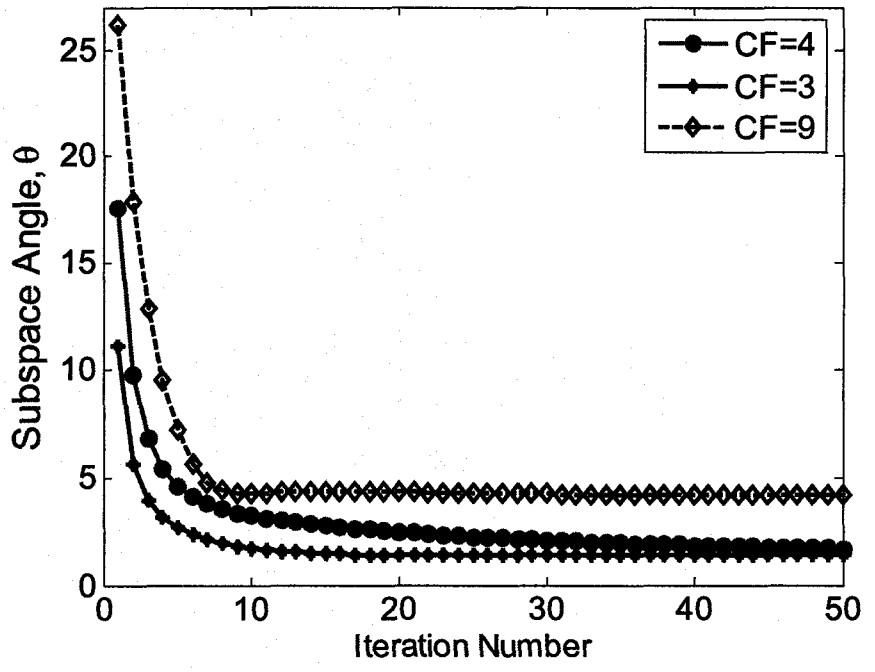


Figure 4.9: *Convergence of PCAIA at different compression ratio*

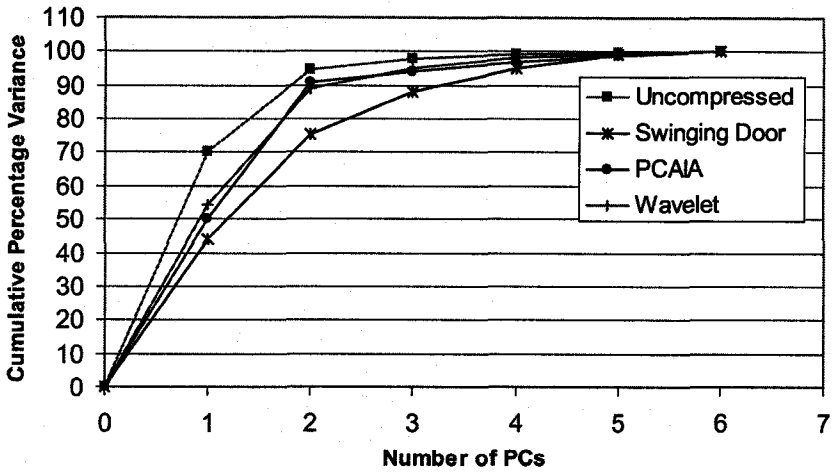


Figure 4.10: *Cumulative percentage of total variance explained by principal components from reconstructed data using different methods*

For example, for the uncompressed data 90% of the total variance is explained by the first two PCs whereas it would require took four PCs to capture 90% variance for the linearly interpolated data. This poses a serious problem in selecting the order of a PCA model as most of the model order selection criteria are based on the analysis of variance. The calculated eigenvalues from Swinging Door Compressed and PCAIA reconstructed data set, and Wavelet Compressed and Inverse Wavelet Reconstructed data set are closer to the eigenvalues calculated from the original data set and the percentage variance explained by the major PCs are also very similar to the uncompressed data set. Thus the model order selection will be more precise for these two cases.

Figure 4.11 compares the quality of the models obtained using reconstructed data from three different reconstruction techniques. It may be noted here that 'Linear Interpolation' and 'PCAIA' reconstructed the compressed data from Swinging Door Compression algorithm while 'Inverse Wavelet Transform' reconstructed data which has been compressed using 'Wavelet Transform'. Since the true model of the process is unknown, the model obtained from the uncompressed data was taken as the benchmark. Subspace Angles of all models obtained from the reconstructed data were calculated relative to this benchmark model. The model built from the 'Swinging Door Compressed and Linearly Reconstructed' data has very poor quality at moderate to high compression ratios, as linear interpolation destroys the correlation structure.

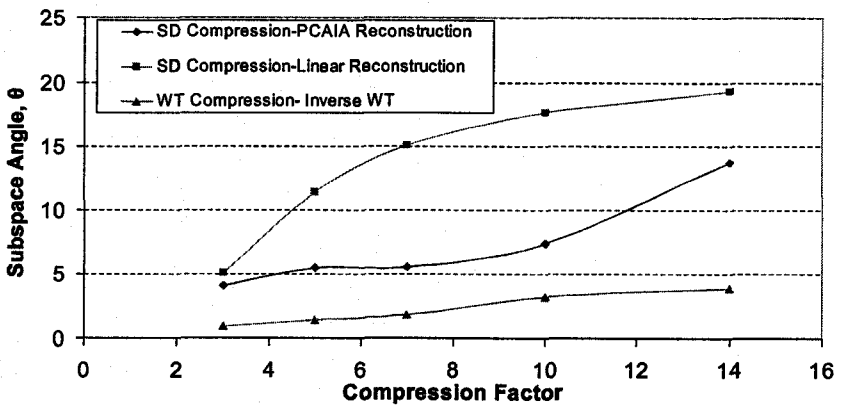


Figure 4.11: *Comparison of estimated model quality from reconstructed data using different methods*

By using PCAIA, instead of linear interpolation based reconstruction, significant improvement could be achieved in model quality. The estimated model from 'Wavelet Compressed and Inverse Wavelet reconstructed' data has the best quality. This is in contrast to the observation in the simulated flow-network system, where PCAIA had the best performance. Such result is not unexpected since the true dynamic nature of the process is not known, and the effect of compression depends on the dynamic behavior of the process. Moreover, the process may be non-stationary and nonlinear to some extent and after discarding the rows which do not contain any original values the sample size became quite small and PCAIA was applied only on that smaller sample size. As a result such small samples may not be completely representative of the process and the method may have suffered from small sample limitations.

It was also observed in the analysis that Wavelet compression preserves the correlation structures between the variables better than the Swinging Door algorithm. The primary reason for this behavior is, the correlation matrix captures the low and medium frequency information. The high frequency excitations in the signals are usually due to random noise and average out while calculating the correlation matrix. Though Wavelet compression is univariate, during the compression and reconstruction it only chops the high frequency information. On the other hand, during the compression and reconstruction of Swinging Door algorithm part of the low and medium frequency information is lost. This is illustrated in Figure 4.12, where the spectral density plot of a signal and the reconstructed signals from Wavelet compression and Swinging Door compression are shown. It clearly shows that, in Wavelet

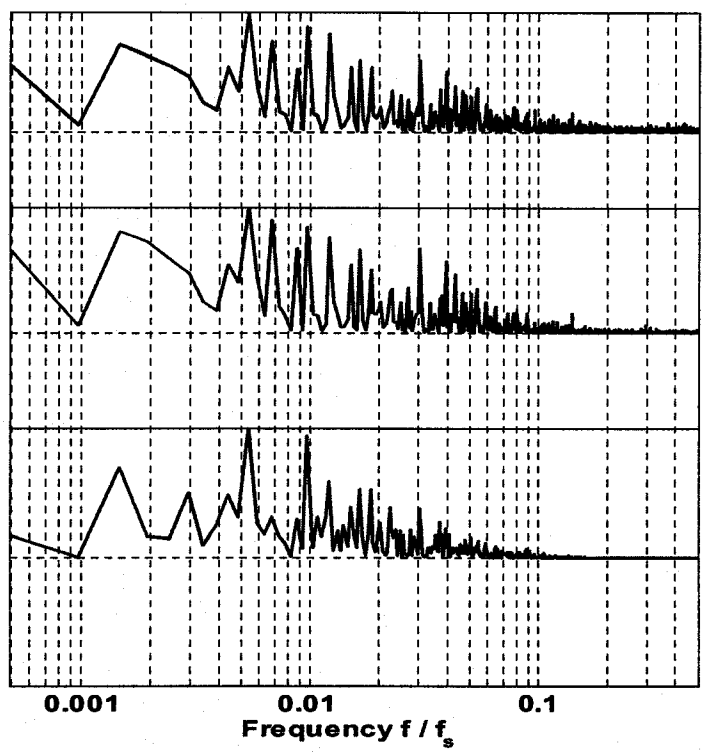


Figure 4.12: *Power Spectra Plot: Original Signal(Top) Reconstructed Signal from Wavelet Compression(Middle) Reconstructed Signal from Swinging Door Compression(Bottom)*

reconstructed signal all of the low and medium frequency information remained intake while in the Swinging Door reconstructed signal part of the low and medium frequency information got lost. This will ultimately have an effect on the correlation structure between the variables.

We recommend the use of 'Wavelet Compression and Inverse Wavelet based Reconstruction' algorithms for process historian. However process industries almost exclusively use 'Swinging Door type Compression and Linear Interpolation' based reconstruction methods in their data historian and this trend will continue to exist for some time. Instead of linear interpolation based reconstruction, missing data techniques based on PCAIA can be used to recover the correlation structure for building multivariate models. The 'Inverse Wavelet based Reconstruction' can only reconstruct data which have been compressed using 'Wavelet Compression'. Therefore 'Inverse Wavelet based Reconstruction' is not an alternative to PCAIA in reconstructing Swinging Door compressed data.

4.6 Concluding Remarks

A detailed study on the effect of compression on multivariate analysis, especially PCA-based modelling has been performed. Compression has been formulated and characterized as a missing data problem. A missing data handling technique (PCAIA) has been used successfully to build model from compressed data. The following conclusions can be drawn from this study:

- Linear interpolation methods to reconstruct compressed data from direct compression methods (i.e., Swinging Door) are not suitable for multivariate analysis. Estimated models from such data can be of poor quality and unreliable.
- A significant improvement in model quality can be achieved by using missing data handling technique to build multivariate models from compressed data.
- The impact of compression on model building increases with the increasing stochastic and dynamic nature of the processes.
- Transform compression methods (i.e., Wavelet Compression) are better in retaining the correlation structure of the signals, and as such decompressed data from transform compression algorithms are suitable for multivariate analysis. However, the performance may deteriorate if the signals have excitation only in the high frequency range.

Chapter 5

Industrial Case Study

The issue of sheet-break prevention is of considerable interest to pulp and paper manufacturers. Sheet-break is a significant contributor to lost market opportunity as well as increased downtime and greater operating expense. Sheet-breaks can occur at different stages of the processing. From three month production history of a major paper mill in Canada it was observed that 62% of the breaks took place in the dryer section, 20% in the press section, 15% were stack breakS and 3% reel breaks. Some of the causes of sheet-break build up slowly ultimately leading to sheet-breaks while other causes are quite abrupt. The abrupt faults generally come without any prior indication and are often difficult to detect in advance. Most of the faults in the press section and part of the faults in the dryer section occur because of slow changes in the process and are believed to be predictable in advance. In a recent study it was reported that the wet-end breaks result in 1.6 hr loss of production time per day, which amounts to $6-8 Million per year for each production line (Bonissone and Goebel 2002). The detection and diagnosis of the root causes of many of these sheet-breaks would bring substantial value to the industry.

The advantages of multivariate monitoring compared to univariate methods are well documented in literature. In pulp and paper mills a large set of variables are monitored. Monitoring these huge number of variables individually is a difficult task. In this context multivariate statistical monitoring of pulp and paper process has been advocated by many researchers (Bissessur *et al.* 1999, Teppola *et al.* 1998). Large scale application of multivariate methods in detecting sheet-breaks is a challenging task for several reasons: (i) numerous process variables; Skoglund *et al.* (2004) reported that over 800 process variables are tracked in a cardboard mill. The paper mill that we investigated logs around 1100 variables online (ii) the presence of many operational regimes due to different grades (iii) frequent downtime and missing data etc.

Different methods and techniques have been used to deal with these problems,

especially the first two problems. Skoglund *et al.* (2004) built PCA models with hourly averages of 177 variables recorded during the manufacture of two main products. Variables were selected based on the engineering judgement of the project team consisting of the operation's personnel. Clearly use of hourly average limits the prediction capability of the model as the time scale between a cause and sheet-break is in the order of few minutes. The other alternative is to divide the variables into conceptually meaningful blocks and then apply hierarchical multi-block PLS (or PC) models. This blocking leads to two model levels: the relationships between blocks are modeled in the upper level and the lower level shows the details of each block. On each level, standard PLS or PC scores and loading plots are available for model interpretation. This allows an interpretation focused on pertinent blocks and their dominant variables (Wold and Tjessem 1996).

Champagne and Ivanov (2002) developed multi-grade model in order to make the models consistent for the numerous operational regimes. They grouped similar grades into a family and built models for each family. In this way they could limit the number of models to a manageable size. The variability within the family was captured using PLS-Discriminant Analysis (PLS-DA). This is a two step procedure where at the first stage the atypical behavior within the family is modeled. This variability is projected on the original data space and subtracted from the data matrix. The calculated residues explain what is similar between the grades within a family. Multivariate model is built using this residual matrix and subsequently used for fault detection.

Though these methods showed success in fault detection there are several problems from an industrial implementation perspective. These methods are complex, therefore regular maintenance and updating of the monitoring schemes are challenging tasks in process industries. Inclusion of large number of variables also comes with risk of having missing data in some of the variables. Often monitoring schemes are implemented without the capability of handling missing data. Because of missing values in few variables the monitoring system may be turned off. Also with large number of variables multivariate PLS (partial least squares projection to latent structures) and PC (principal components) models plots and lists of loadings, coefficients, etc. become messy and results are difficult to interpret. Therefore, there is a need to develop monitoring schemes which are simple in application, easily maintained and of manageable size.

In this study, we describe development and implementation of a PCA-based sheet-break monitoring scheme that was carried out for a major paper mill. The main objective is to detect and isolate the root cause of sheet-breaks well ahead of the breaks, so that corrective actions can be taken to prevent the sheet-breaks. An

important objective of this application is to develop a monitoring scheme that is fairly sensitive to detecting sheet-break faults and yet has as few false alarms as possible. Though the theory of PCA is well known for fault detection and isolation, the successful implementation of the PCA based monitoring partly depends on pre-processing of the data. In this paper we describe several novel techniques that have been developed and used for selection of the good data segments, important tags, data scaling and model order selection.

In addition to a multivariate monitoring scheme an extensive root cause analysis was carried out by applying data mining techniques combined with process knowledge and engineering judgement. The root causes of sheet-breaks were identified and recommendations were made. Implementation of the changes made on the basis of these recommendations reduced the frequency of sheet-breaks significantly. The economic impact of the changes were evaluated. Savings of more than $1 Million in terms of fewer sheet-breaks and reduced downtime per year have been realized as a result of implementing the suggested changes. In this paper we also report the key performance indices before and after the changes were made.

5.1 Principal Components Analysis

Principal Components Analysis (PCA) is a dimensionality reduction technique introduced by Pearson (1901), and later developed by Hotelling (1933). PCA is being used as a multivariate Statistical Process Control (SPC) tool for monitoring a wide range of processes (Kresta *et al.* 1991, Bakshi and Stephanopoulos 1996, Qin 2003). PCA has also been developed to monitor dynamic processes (Li and Qin 2001). An excellent review on the theory and application of PCA can be found in Jackson (1991).

Principal Components Analysis (PCA) is an effective tool in multivariate data analysis. It projects the data set, which typically may have correlated variables, onto a new coordinate system where the transformed data is uncorrelated. The main coordinates of this new subspace, also known as the principal subspace, are known as the principal components (PC). Each PC is a linear combination of original variables. For example, given a data matrix or measurement matrix $X \epsilon \Re^{N \times m}$, where N represents the number of samples and m is the number of process variables, the Principal Components (PCs) are a set of score variables given by linear combination of the original variables:

$$t_i = Xp_i \quad [i = 1, \cdots, m], p_i \epsilon \Re^{m \times 1} \tag{5.1}$$

p_i is the basis vector of the principal subspace also known as loadings vector.

The coefficients of each linear combination are obtained from an eigenvector of the covariance matrix of the original variables. In general, the principal subspace has a lower dimension than the original Euclidean basis space and yet is able to capture or explain significant portion of the information content (or the variance) in the original data set. Each of the Principal Components or score vectors, $[t_1,\ t_2, \cdots\ t_r, \cdots t_m]$ collected in the matrix $T \epsilon \Re^{N \times m}$, capture as much variation as possible which has not been explained by the former PCs, i.e. the first PC maximizes the covariance in the original data and the subsequent PCs maximize the covariance in the residual matrices which are left after extracting the former PCs. The maximum number of principal components are equal to the total number of the variables. However, most of the systematic information of the data can be explained by the first few or 'r' principal components where $(r < m)$. The reduced dimensional latent variables or principal components represent a consolidated set of the most important measurements of the process. In this respect PCA offers an elegant parsimonious representation of the process, presumably along the direction of the most dominant variance of the process. In a high-dimensional or multiple unit-operation process such as, pulp and paper process, typically hundreds of measurements are available. Such processes thus offer an ideal opportunity for extracting the most important information from a myriad of sensors in a compact way. This exercise is analogous to obtaining a compact lumped parameter and reduced order model of an otherwise high dimensional process. This can be done by finding an optimum set of weights or loadings from a nominal data set. In this respect the nominal loadings and scores represent a model of the process in the principal or latent variable subspace.

Accordingly the scores and loadings matrix are partitioned in two parts: $T = [T_r\ T_e]$ and $P = [P_r\ P_e]$ explaining the systematic and random variation in the data respectively. The data matrix X can be expressed in the following way:

$$\begin{aligned} X &= [\underbrace{t_1,\ t_2,\ \cdots\ t_r}_{T_r}, \underbrace{t_{r+1}\ \cdots\ t_m}_{T_e}][\underbrace{p_1,\ p_2,\ \cdots\ p_r}_{P_r{}^T}, \underbrace{p_{r+1}\ \cdots\ p_m}_{P_e{}^T}]^T \\ &= T_r P_r^T + T_e P_e^T = \hat{X} + E \\ E &= T_e P_e^T \end{aligned} \tag{5.2}$$

Ideally r is chosen such that there is no significant process information left in the covariance matrix and E contains only the random error. Thus, the retained loading vectors, $P_r = [p_1 \cdots p_r]$ are called Principal Components 'model' that describes the systematic variation in the data. Addition of extra loading vectors to the PCA model would only fit the random error and lower the prediction capability of the model.

5.1.1 Fault Detection

Two collective test statistics have been defined for fault detection using Principal Components Analysis.

Hotteling T^2-Statistics

The original form of T^2 is:

$$T_i^2 = (x_i - \overline{x})S^{-1}(x_i - \overline{x})^T \tag{5.3}$$

Hotelling T^2 statistic is used as a measure of the variation within the PCA model. T^2 is the sum of normalized squared scores. Assuming the data to be mean centered, T^2-statistics are given by,

$$T_i^2 = t_i \Lambda_r^{-1} t_i^T \tag{5.4}$$

where Λ_r is a diagonal matrix containing the r largest eigenvalues, λ_i and t_i refers to the i-th row of $T_r \epsilon \Re^{N \times r}$, the matrix of r score vectors from the PCA model. T^2-statistics are therefore not affected by the inaccuracies of the smaller eigenvalues and in this respect are better suited to represent the normal behavior of the process. Statistical confidence limits for T^2 are directly calculated from the F distribution (Wise and Gallagher 1996).

$$T_{UCL}^2(\alpha) = \frac{r\,(N-1)}{(N-r)} F_\alpha\,(r, N-r) \tag{5.5}$$

where $F_\alpha\,(r, N-r)$ is the $100(1-\alpha)\%$ critical point of the F distribution with r and *(N-r)* degrees of freedom.

Squared Prediction Error(SPE) or Q-Statistics

This collective test, also known as Rao-statistics employs the portion of the observation space corresponding to the *(m-r)* smallest singular values. The collective test is defined by,

$$Q_i \; or \; SPE_i = (x_i - \hat{x})\,(x_i - \hat{x})^T \tag{5.6}$$

The distribution of the Q-statistic has been approximated by Jackson and Mudholkar (1979):

$$Q_\alpha \; or \; SPE_\alpha = \left[\frac{h_o c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_o\,(h_o - 1)}{\theta_1^2}\right]^{1/h_o} \tag{5.7}$$

where $\theta_i = \sum_{i=a+1}^{n} \sigma_j^{2i}$, $h_o = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$, and c_α is the normal deviate corresponding to the $1-\alpha$ percentile. Given a level of significance, the threshold for the SPE can

be calculated by Equation 5.7 and used to detect the fault. Geometrically SPE is the projection distance from a point in R^m to the hyper-plane defined by the PCs. It is a measure of the degree of agreement with the correlation structure identified by the set of PCs. When the process is 'in-control' SPE is the random variations of the process, for example, the measurement noise, that cannot be accounted by the PCA model. A violation of the threshold would indicate that the random noise has significantly changed or the correlation of the data no longer holds. The T^2 and SPE along with their appropriate thresholds detect different types of faults, and the advantage of both statistics can be utilized by employing the two measures together. To simplify the fault detection task Raich and Cinar (1996) suggested the following combined statistics

$$CI_i = k\frac{SPE_i}{SPE_\alpha} + (1-k)\frac{T_i^2}{T_{UCL}^2} \tag{5.8}$$

where $k \in (0,1)$ is a constant. Therefore, if the value of combined index (CI) is less than 1 the process is considered normal. In the current study we used such a combined index (CI) as sheet-break predictor. For detecting the faults we mainly relied on the SPE plots. However, the pulp and paper process is subject to many external disturbances and the SPE plots contained several false alarms even during the normal operation. On the other hand, the T^2 plot was below the threshold and indicated the normal operation very distinctly. But at the same time it was not as sensitive to the sheet-break faults as the SPE. Therefore it was evident that the T^2 and SPE complement each other in fault detection. Combining T^2 with SPE helped to suppress the false positives during normal operations, at the same time it was also sensitive to faults. The fault sensitivity of the combined index (CI) can be further increased or decreased by changing the tuning parameter k.

5.2 Data Description

All the data analyzed during this study were obtained from Paper Machine 7 (PM7) of ABITIBI Consolidated. A block diagram showing the different units of the process is given in Figure 5.1. The plant is located at Fort Frances, Ontario, Canada and the analysis was done at an off site location, Matrikon Inc. located in Edmonton, Canada. An online data transfer link was established between the plant historian and the desktop at Matrikon Inc. This gave direct access to all variables which are measured online. In addition to this, the downtime report and the grade change log were supplied by the plant personnel. The downtime report contained the operational state of the plant as well as the comments and findings of the operators for different

fault conditions. This information base was used to select the training data set, separating the normal data from abnormal data and subsequently for tuning of the models against known faults. The data collection started from July 2004. The initially collected data was analyzed to find out different types of sensor problems (e. g., poor resolution, broken sensor), data compression etc. These findings were reported to the plant personnel. They fixed many of the sensor problems and set the compression thresholds to very low values so that no compression was applied while archiving the data in the data historian. All the variables are continuous except a few binary variables indicating the plant status. Later the training data set was collected during the time period of October, 1 2004 to December, 31 2004.

5.3 Pre-processing of Data

Pre-processing of data is important for any data based analysis to ensure that data truly represent the different events of the process. In this analysis we used different qualitative and quantitative measures to investigate the data quality. For example, a quantization factor was used to detect any sensor resolution problem, compression factor to detect loss of information during the archiving etc. Based on the data quality analysis some of the variables were deemed unsuitable for inclusion in the multivariate models. In this way this also helped in variable selection for the models. The methodology and the rationale for using these data analysis tools are described below.

5.3.1 Quantization Factor

This is a quantitative measure of the resolution of the measuring instrument. If the variability in the measurement and the resolution of the instrument are not of the same order then much of the excitation appearing in the data may be due to the low resolution of the instrument. For example, if the measured values of a variable are varying between 0.04 and 0.06 and the resolution of the instrument is 0.1 then it will show values in quantum jumps of 0.1, e.g. from 0.0 and 0.1. This type of artificial variation may deteriorate the model quality. The mathematical definition of quantization factor is:

$$QF = \frac{min.\ difference\ between\ consecutive\ points}{standard\ deviation\ of\ the\ signal} \tag{5.9}$$

Quantization factor of all the variables were calculated to find out whether the instruments' resolutions are commensurate with the variation of the respective variable.

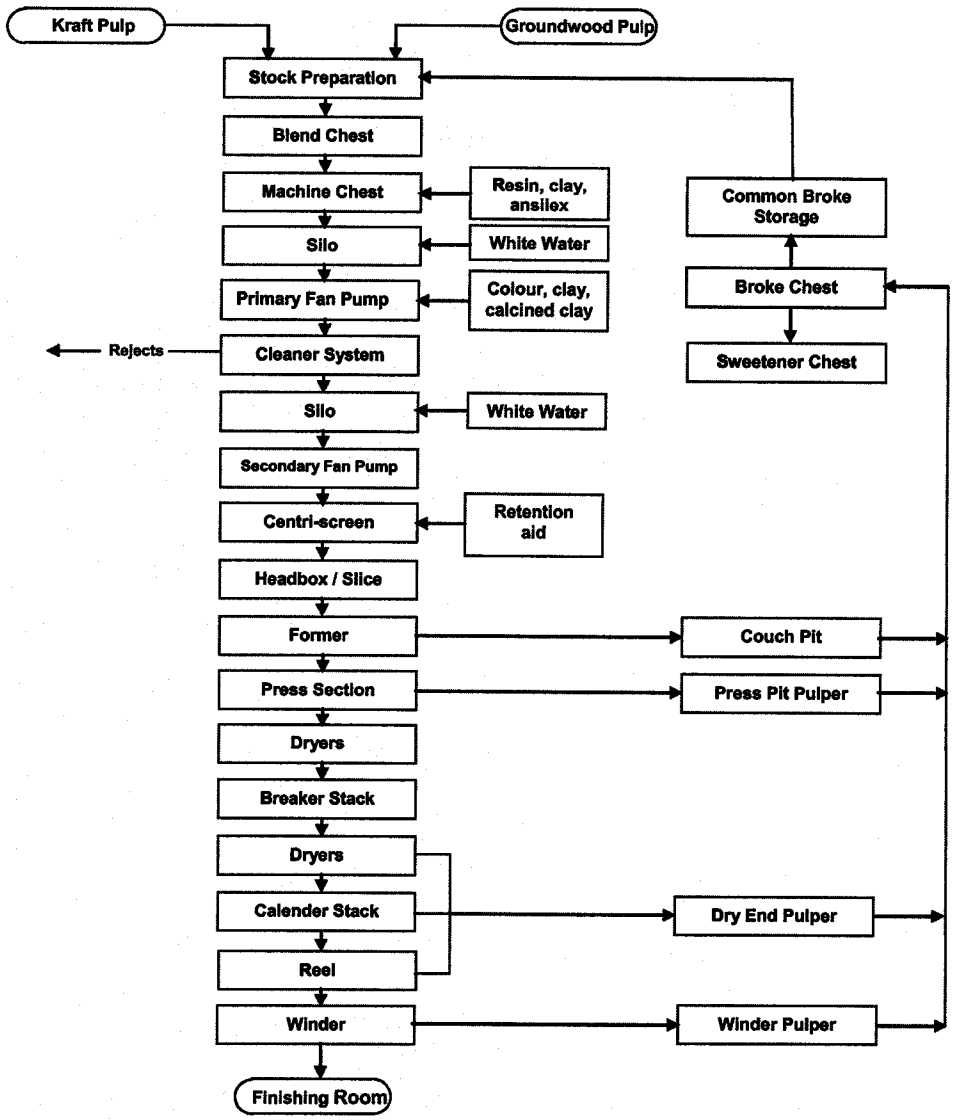


Figure 5.1: *Block diagram of different processing units of the pulp and paper process*

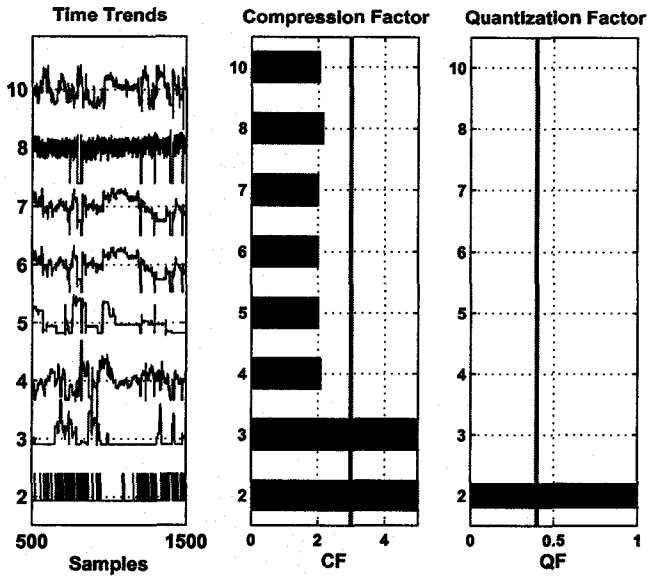


Figure 5.2: *Trend plots, and calculated compression factor and quantization factor of signals from the pulp and paper process*

Variables which have quantization factor greater than 0.4 were eliminated from the data matrix, for example, in Figure 5.2 variable 2 is highly quantized and it was eliminated from the data set.

5.3.2 Compression Factor

Process data are often compressed before storage in the data historian. Therefore, it is important to check the compression factor of the retrieved data from the process historian. Compression Factor (CF) is defined as follows:

$$CF = \frac{number\ of\ original\ measurements}{number\ of\ recorded\ measurements} \tag{5.10}$$

When data is retrieved from a process data historian, it uses a decompression algorithm to provide a data matrix with the specified sampling rate. These decompression methods mostly use linear interpolation to fill the in-between points of the originally stored spot values. The compression detection algorithm takes advantage of this linear structure. The algorithm calculates the double derivative of the signal, therefore at the linearly interpolated points, the double derivative would be '0' (Thornhill *et al.* 2004). Severe compression destroys the correlation between different variables and therefore is not suitable for multivariate statistical modelling (Imtiaz *et al.* 2005).

Variables with compression factors greater than 3 are normally not suited for model building. We identified the variables with high compression factor and asked the plant personnel to set the threshold to small values so that no compression is applied during the archiving. Therefore, the data included in the final models are uncompressed raw values.

Apart from data compression there may be other reasons for linear segments in the data, e.g. sensor failure for a period of time or no sensor activity over a long period. Compression factors will be high for these data sets as well. Therefore, the compression detection algorithm can also be used to check the excitation or the information content in the data. In this current study we used a compression factor mainly for accessing the information content of the signals. Though we are building steady state model, in reality the process is never at one steady state. We observed that the process is time varying and even within the same grade the operating conditions and the correlation between the variables change. In building the model we would like to capture the average correlation that will cover most of the natural variations of the process. Inclusion of the segments which remain constant for a long period the model may get biased towards these operating conditions. The compression factor can easily detect those segments with long constant periods or variables which have little sensor activity, and those variables may not be included in the model. The trend plots of some of the variables and corresponding compression factors are given in Figure 5.2. For the sheet-break industrial case study, variables with compression factors greater than 3 were deemed to be detrimental for data quality and as such multivariate statistical modelling.

5.3.3 Spectral Density Plot

Spectral density plot shows the signal strength at different frequency. Except for the speed signals, most of the variables in the pulp and paper process have slow dynamics. Therefore, we expect that signals with good excitation should have high strength in the low and medium frequency range. In the models we mainly included variables with spectral strength in the low and medium frequency range for two reasons: (i) we are mainly interested in detection and isolation of faults which build up slowly (ii) variables which oscillates in the same frequency are strongly correlated with each other and the correlation arises from process constraints. Variables which have strength only at high frequency regions are not good candidates for building a model as the excitation may be mainly due to measurement noise and short lived disturbances. Spectral plots together with the trend plots give complete information

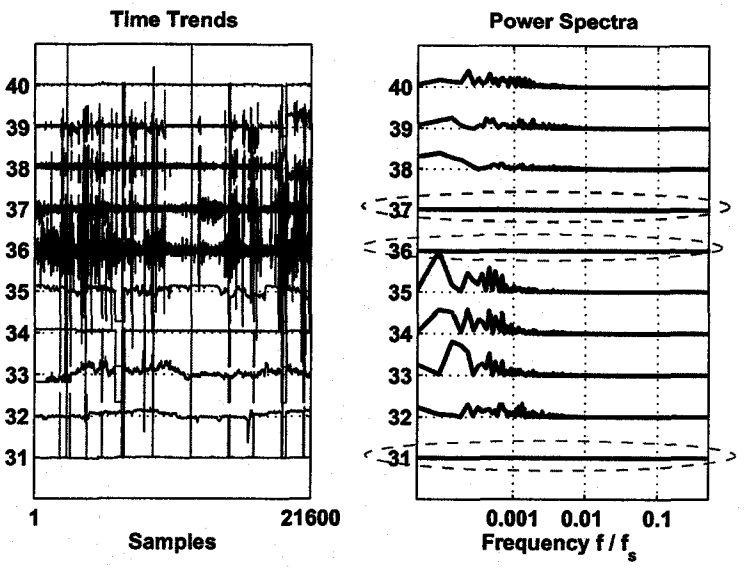


Figure 5.3: *Time trend and spectral density of signals from the wet end of the pulp and paper process*

about a signal. The frequency domain feature extraction capabilities of the spectral density plots are shown in Figure 5.3. The plots depict the normalized spectral strength of few typical variables from the pulp and paper process, at normalized frequencies. Variables that have very flat power spectra are highlighted in the figure.

5.4 PCA model for fault prediction

There are many challenges in building a PCA model to predict the sheet-break faults consistently. Primarily the challenges come from the fact that pulp and paper data are subject to many external disturbances, high number of variables, production of many different paper grades and frequent changes in grades. In this section, the techniques used to surmount these challenges and the step-by-step method of building the PCA model for sheet-break prediction are described.

5.4.1 Selection of Tags

The number of variables which are measured on-line in a pulp and paper mill is quite substantial. Initially 1024 tags were downloaded from the process data historian. Instead of building models with all these variables we chose to build parsimonious models which are easy to manage yet sensitive to most of the faults.

A detailed study of the flow sheet suggested that the stock preparation, head box and initial portion of the dryer sections may be the best places to look for slowly developing or impending faults. In the PCA model, variables from these sections only were included. The more downstream sections of the paper making process have very fast dynamics. Causes developing in these sections will lead to abrupt faults and therefore are difficult to predict. The sections from where the variables were selected are close to each other. Therefore, transportation delay is minimal and hence we did not choose to lag the variables in the data matrix. The limitation to fewer variables comes with other advantages also, for example, in case of a successful detection, isolation becomes easier as the number of variables are limited. In combination with data pre-processing other criteria were also used to eliminate variables from the list. Initially we built several models with a large set of variables for each of the grades. Those grade specific models were used locally for fault detection and isolation. We went through several trials and selected a set of variables which were identified as the causes to different kinds of faults. Finally, 39 variables were selected and used to build the PCA model. The importance of the right tag selection is illustrated in Figure 5.4. It shows the T^2 and SPE plot of a validation data set obtained using two different models. There is a confirmed sheet-break at the end of the data segment. Initially a model was built using 164 variables. Sixty six PCs were selected, which explained 80% of the total variance. The T^2 and SPE plot for the model are shown in Figures 5.4 (a) and (b). Both statistics are prone to false positives and the plots show many false alarms during the normal operation. Subsequently 39 variables were selected using engineering judgement and the data pre-processing techniques as discussed in the previous section. The T^2 and SPE values were calculated for the same validation data set using the 39-variable model and are shown in Figures 5.4(c) and (d). In this case both T^2 and SPE plots have fewer false positives compared to the previous case. However, the model needs further refinement in order to give consistent results.

5.4.2 Grade Specific Model

In order to meet market demand, pulp and paper mills have to produce paper of different grades. Some of the distinguishing features of the grades are basis weight, caliper and shade. Papers of 7 different basis weights,13 different calipers and 3 different shades are produced in the plant. With the combination of these three qualities the total number of grades becomes very large. For the current plant, in a particular month 42 grades were produced and during that month alone the change over between

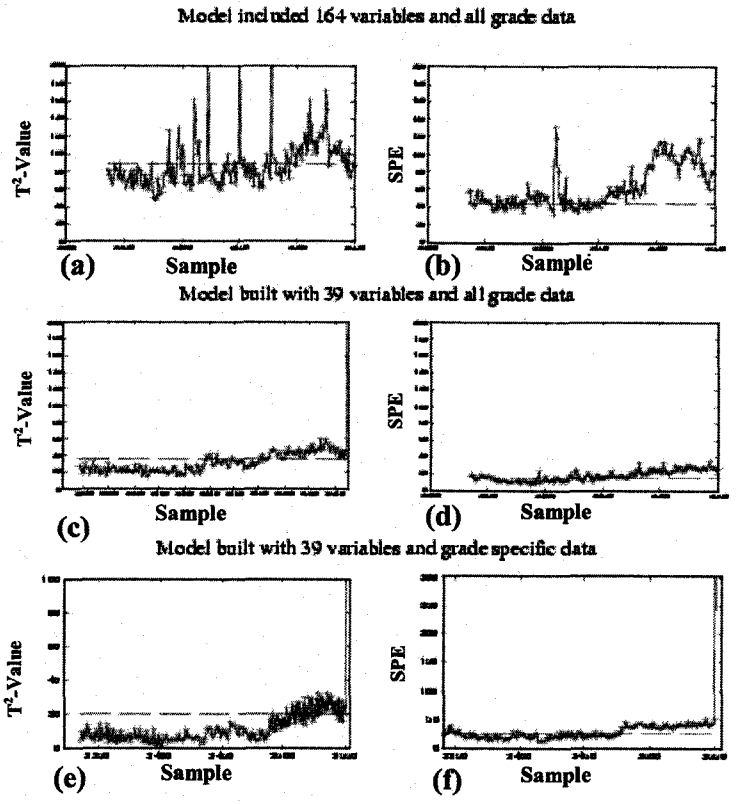


Figure 5.4: *Performance of different type of models in fault prediction*

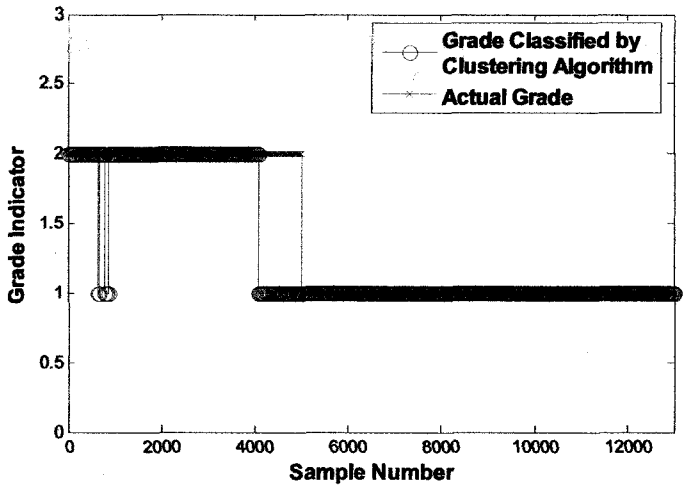


Figure 5.5: *Clustering algorithm identifying the data from two basis weights as two clusters*

these grades took place 204 times. If the model is not trained to accommodate such changes, grade changes can be wrongly detected as faults and conversely if the model is trained with such changes, the abnormalities may be perceived as grade change events. One way to circumvent this difficulty is to build a separate model for each grade. However, given the large number of grades and frequency of grade changes, this is not a feasible option. In an effort to reduce the number of grades, basis weight was taken as the main criterion for classifying data into different grades. There are several reasons that justify this criterion. Firstly, changes for adjusting basis weight is carried out in the stock preparation region while changes for adjusting caliper occur further down stream, mainly in the dryer section. The causes that develop in the dryer section usually lead to abrupt sheet-breaks hence undetectable by the monitoring scheme. Secondly, a change of shade does not affect the strength of the paper. In addition to this, a classification algorithm also verified that basis weight is a good criteria to divide the data in different grades. Data with two different basis weights were put together and the classification algorithm was used to search for two clusters. The result is shown in Figure 5.5, where the actual grade based on basis weight and the grade identified by the clustering algorithm are plotted. Except for very few data points, the classification algorithm was very successful in identifying the two basis weights as two different grades.

Subsequently using 'basis weight' as the criterion for grade classification, data

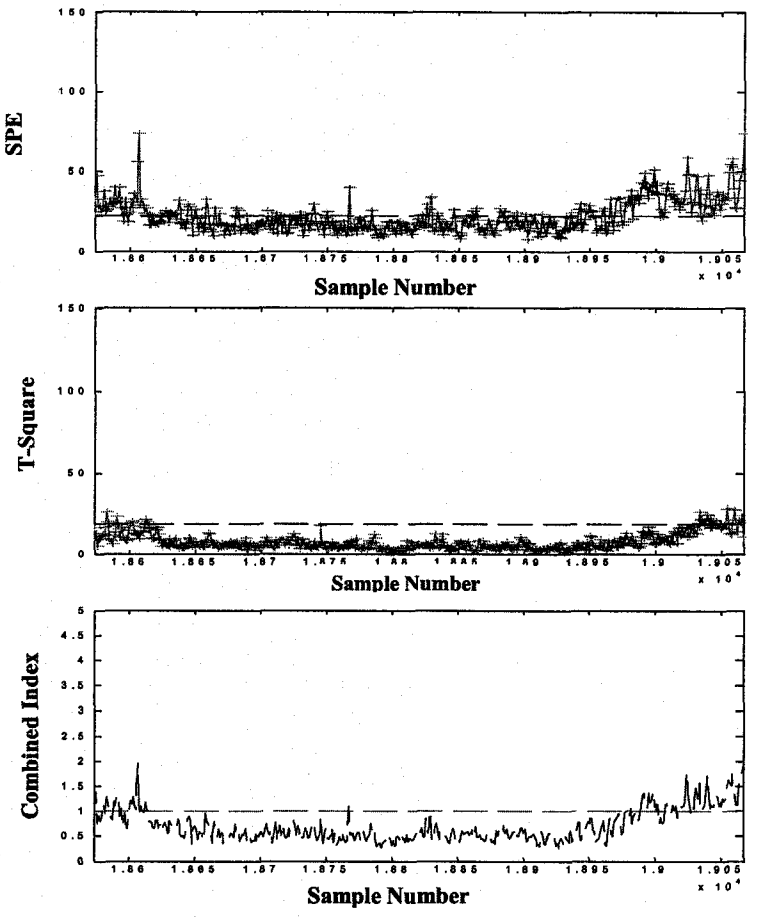


Figure 5.6: T^2, *SPE and Combined Index(CI) plot showing the utility of CI to suppress false alarms*

was divided into seven different grades and a separate model was built for each grade. These grade specific models showed superior performance in explaining the normal behavior of the plant and also in detecting any abnormalities in the process. This is illustrated in Figures 5.4 (e) and (f) with the T^2 and SPE plot respectively. During the normal operation the T^2 statistics is well below the threshold, yet it predicted the fault 40 minutes before the sheet-break occurred. The SPE plot has few false positives and lies just below the threshold. This is a typical pattern observed in other validation data sets as well. False positives in the SPE plot are not desirable as this will significantly lower the reliability of the model. However, the pattern suggests that instead of using T^2 and SPE separately if they are combined together, this will suppress the false positives of the SPE plot. Equation 5.8 was used to calculate the combined index. Warnings were generated only if the combined index goes outside the threshold limit. Figure 5.6 demonstrates the improved performance of the combined index in fault prediction. Though there were few false positives in the SPE plot, the combined index is within the limit during the normal operation. However, there are few false positives initially. This is primarily because the process was resumed after an upset and it is still in a transition phase, where the correlation structure is much different from the training data set.

5.4.3 Building a Training Data Bank

Pulp and paper processes are interrupted frequently both intentionally and unintentionally. From the grade run report of the plant it appeared that the process operation was interrupted more than 100 times during a particular month. Once the process operation is interrupted it takes approximately 30 minutes to bring the process back to its normal state. In order to make a large data bank, data from different segments of normal operation were concatenated. For 'seamless' 'patching' or fusion of data segments in the temporal domain, transient data from the initial region of operation after any event has taken place, and abnormal data before any event were eliminated before 'patching' the data, i.e., 50 samples (1 minute interval) were discarded from either side of each data segment and subsequently concatenated. However, concatenation of data has also some disadvantages, as different segments may have different means and covariances. This gives rise to false alarms or 'blips' at the joining points in the T^2 and SPE plot of the normal model. Some of these issues can be taken care of by doing the scaling judiciously as discussed in the next section.

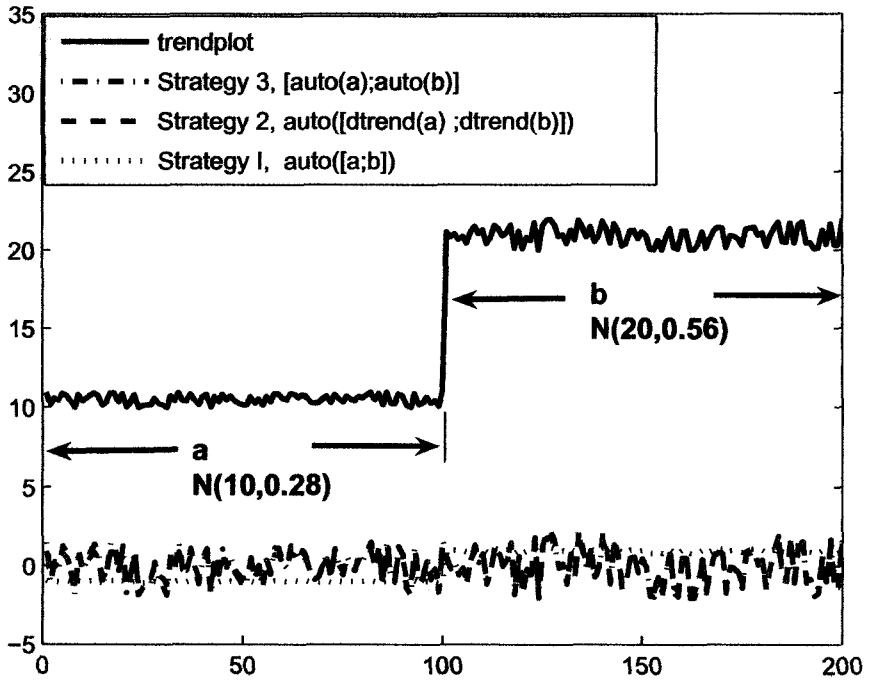


Figure 5.7: *Simulation study showing the effect of different type of mean centering and scaling*

5.4.4 Scaling and Centering of Data

In a data set where the variance of the measurement noise are unequal, PCA will be scaling invariant only if each variable is scaled with the standard deviation of its measurement error. However, estimation of error variances for these large number of variables is not very accurate and hence the general practice is to auto scale the data. PCA is not scale invariant with regard to auto scaling. In order to enrich the data bank, data segments which were lying further apart in time were stitched. Though these segments belong to the same grade, yet significant changes were observed in the mean value. To visualize the effect of scaling on concatenated data we carried out a simple simulation study. The results of the study are shown in Figure 5.7. Two segments of data segment (a) and segment (b) with different means and variances, were concatenated and scaled in three different ways.

1. *Stitching the segments together and subsequently auto-scaling the whole data set:*

This completely destroys the variance structure of the signal. The standard deviation of the stitched signal is 5.23, which is much higher than the original standard deviation of either segment (a) or segment (b). When this high standard deviation was used to scale the signal, the excitation of the original signal got attenuated.

2. *Mean Centering of Each Segment, Stitching and Auto Scaling*: Each segment of data was first mean centered to its local mean and stitched together. The resulting signal was auto scaled. The excitation pattern of original signal was retained after scaling and the standard deviation of the concatenated signal is a value in between the standard deviation of the two segments.

3. *Auto Scaling Each Segment and then Stitching*: The resulting signal retains the excitation pattern but there is a problem from an application point of view. If any segment of the signal is near constant, estimated standard deviation will be very low and the signal can get artificially inflated.

In the current application the second strategy was used where each segment was first mean centered, then stitched and auto scaled. The primary purpose of mean centering is to express the variables in deviation form or in regression terminology subtract the 'intercept' term. However, in this application, instead of mean centering all variables were median centered, since median is a more robust measure of the 'intercept' in the presence of large unwanted deviations such as outliers.

5.4.5 Model Order Selection

Model order selection is very crucial for the satisfactory performance of PCA. There are many different methods for selecting model order, e. g. broken stick method, 'SCREE plot', cross validation etc. Of these methods from a prediction point of view cross validation is regarded as the most reliable method. It selects the number of latent variables which give the lowest prediction error sum of square (PRESS). However, cross validation is a computationally intensive method and does not offer an easy technique to tune the model against known faults. There are also other methods, e.g., Backward Q_c^2um (BQ), Variable Importance in Projection (VIP) which can be used to simultaneously select the model order and the important variables for prediction (Lazraqa *et al.* 2003).

In the current study we adopted a different approach in selecting the model order or number of retained PCs in the model. In combination with the eigenvector plot or variance plot a trial and error method was used to select the number of PCs. The

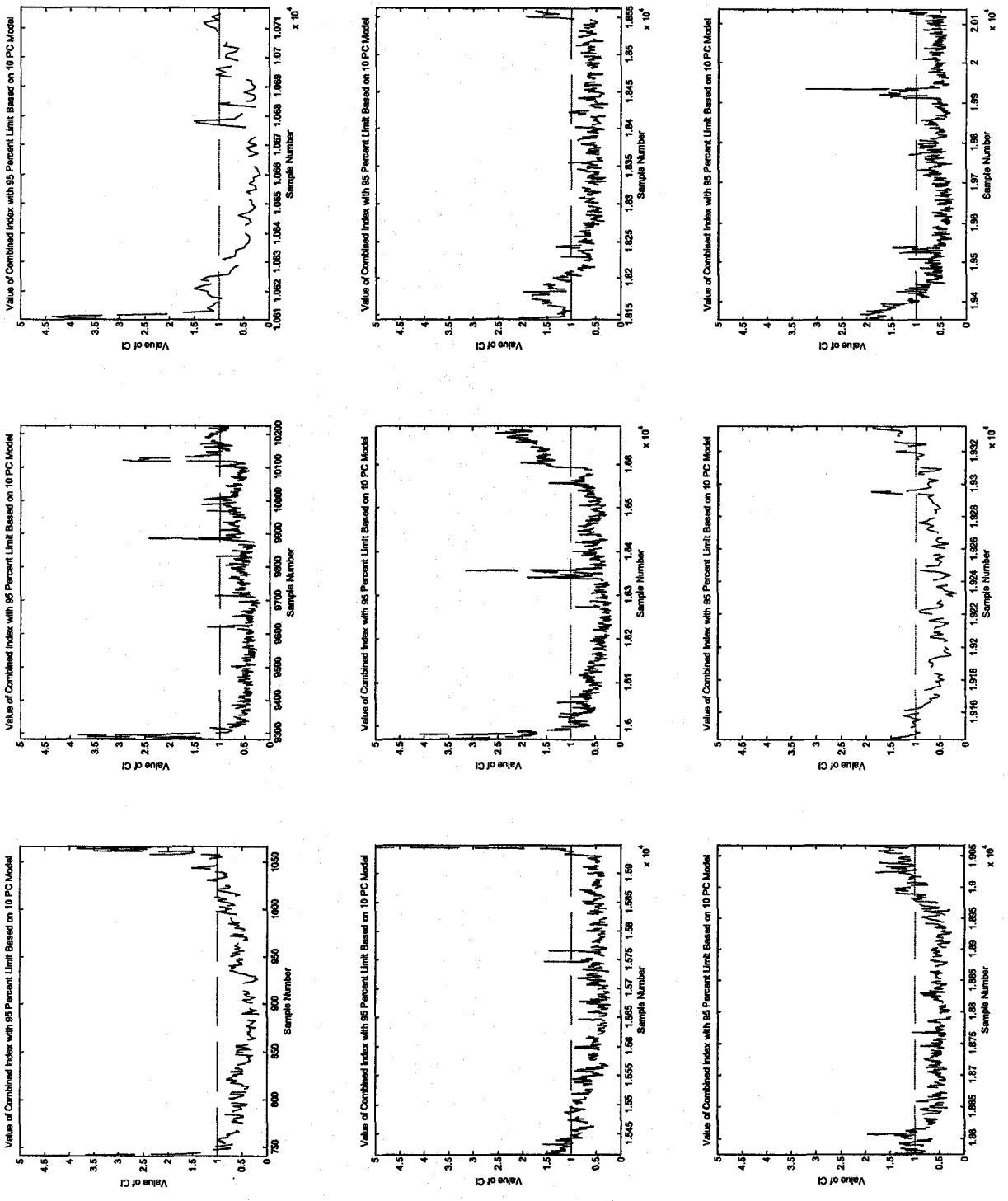


Figure 5.8: Fault prediction of a particular grade using Combined Index in one month

main objective is to get a model, which gives the fewest number of false alarms during the normal operating condition and yet has enough power to detect the fault. We had a large repository of validation data sets with known faults. Therefore we used the validation data sets with faults to tune the models. Models were built by varying number of PCs and fault detection performance was evaluated for each of the cases. After several trials 11 principal components were included in the model. These 11 PCs cover approximately 65% of the total variance. The notable fact here is that, the model explains much lower amount of variance of the data then usually suggested. An increase in the number of PCs reduced the SPE threshold and gave rise to false alarms in the SPE plot whereas, a decrease in number of PCs gave rise to false alarms in the T^2 plot during the normal operation.

5.5 Fault Detection and Isolation

5.5.1 Combined Index

For detection of a fault there are two indicators, the T^2 plot and the SPE plot. A violation of the limit in T^2 plot indicates that the process is drifting away from its normal operation region. On the other hand, violation in SPE plot indicates that the correlation structure has broken down. A violation of both SPE and T^2 limits is a strong indication of an impending fault. So, in this analysis we relied on a combined index which combines both of these indicators. A warning is issued whenever several consecutive points of the combined index go outside the threshold. The model was tested using three months of data. The fault prediction of a particular grade in one month is shown in Figure 5.8. Following any shutdown the combined index always goes outside the threshold immediately after the start up. This is because during this time period the process is in a transition phase and the model was not trained with the transient data. It takes approximately 30 minutes for the process to reach its normal states. To avoid false warnings it is recommended that the alarms be turned off during this initial period. The models were tested against some faulty test data sets selected by the plant personnel. If the CI gave a consistent warning 15 minutes prior to the sheet-break event then it was considered to be a good detection. The models detected 63% of the faults successfully from these selected test cases with very few or no false alarms. This was considered quite satisfactory by the plant personnel considering that a small number of variables were included in the model and among the test cases many of the faults are abrupt and hence undetectable until the very end. SPE contribution plots were mainly used to isolate the faulty variables. The score's

contribution plots did not give any additional information. This may be due to the fact that with the mean centering of each segment of data some systematic variation also got removed from the data matrix. Subsequently, a 'process trouble-shooting' exercise was carried out to diagnose the root causes.

5.5.2 Dynamic SPE Chart

Break down of the correlation structure of the data are reflected in the SPE chart. Therefore, SPE-contribution plot points towards the variables which are responsible for the deviation in the correlation structure. Typical SPE-contribution plots give the contribution of the variables at a specific instant. In this current study we used color coded Dynamic SPE-contribution plot to isolate the faulty variables. Dynamic SPE-contribution plot gives the contribution of the faulty variables at each time instant. Therefore, the exact time at which the variables started deviating is clearly indicated in the figure. This is very helpful in differentiating the causes from the effect. Furthermore, since it is color coded the faulty variables stand out in the plot and can be isolated very easily. A representative Dynamic SPE-contribution plot is shown in Figure 5.9. The plot shows the exact instant at which the variables started behaving abnormally, and color code shows the relative contribution of the variables in the SPE. It is evident from the figure that, in this case the 'Speed' started behaving abnormally prior to the 'Dryer Pressure'. The change in the 'Speed' may be the main cause for the sheet-break while change in 'Dryer Pressure' may have been due to the speed change. The trend plots of the two variables are shown in Figure 5.9 which also suggest that change in the speed occurred prior to the change in the dryer pressure.

Dynamic SPE contribution plot was used to analyze all the faults of November 2004 and December 2004 data. Variables which were most frequently detected as probable causes of sheet-breaks are listed in Figure 5.10. The information from the list combined with process engineering knowledge led to the conclusion that the stock proportioning and the dryer pressure systems are two most likely areas where the causes of sheet-breaks originated. Therefore, the detected variables from these two areas were further investigated for confirming the root causes of sheet-breaks.

5.6 Diagnosis Results

In order to establish connection between the isolated variables from the multivariate analysis and the root causes, further investigation was done using process flow-sheet, the downtime report and data visualization tool. For visualization of the data,

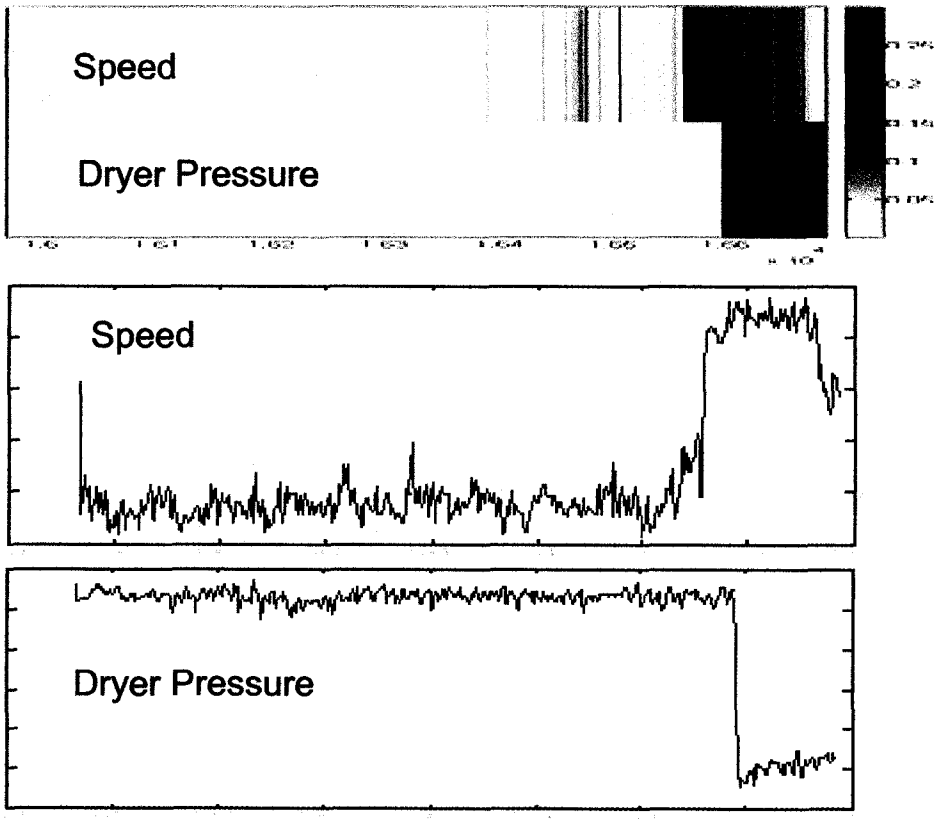


Figure 5.9: *Color Coded Dynamic SPE contribution plot showing the relative contribution of the variables in the SPE chart. The trend plots show the exact changes in the corresponding variables*

Serial	Tag Description
1	PM7 BROKE FLOW
2	PM7 S/A CLOUDY DILUENT FLOW
3	PM7 2 DRYER PRESSURE DIFFERENTIAL
4	PM7 1 DRYER PRESSURE DIFFERENTIAL
5	NO2 BROKE STORAGE TO PM7 FLOW
6	PM7 ANSILEX FLOW
7	PM7 JET TO WIRE RATIO
8	PM7 KRAFT REFINER POWER
9	PM7 KRAFT CONSISTENCY
10	PM7 BROKE CONSISTENCY
11	PM7 S/A REPULP CONSISTENCY

*PM7= Paper Machine 7

Figure 5.10: *List of variables which were detected most frequently by dynamic SPE contribution plot as the probable causes of sheet-break*

ProcessDoctor® was used extensively. It has many useful features such as, stacking several plots, conveniently importing the tag names and time stamps etc. In this section we discuss some of the identified root causes.

5.6.1 Stock Proportioning

The flow diagram of the stock proportioning system is shown in Figure 5.11. The stock in paper machine seven (PM7) consists of mainly three types of stock solutions: (i) groundwood stock, (ii) kraft stock and (iii) the broke solution. Of these the kraft stock has the best fiber quality while the broke stock has poor fiber quality and high variability in its composition. Broke consists of pulp obtained from the paper making process drains (waste) and from the off quality products because of non-conformity to the specifications. The waste pulp is collected at different locations of the paper machine such as couch pit, press pit, dryers, calendar, reel drum and winder areas. The collected refuges are then re-pulped and sent back to Broke storage tanks. The Broke system helps recover wasted fibers and is an essential step in loss prevention, i.e. it helps in maintaining high fiber yield and thus reduces the cost of paper since less fiber that has been treated (pulped and bleached) is lost to the secondary facility. Recycling 'broke' also minimizes the environmental impact. Broke from two other paper machines are also fed to the PM7 production line and is marked as 'make-up' broke in Figure 5.11, while broke from PM7 is labeled as 'machine broke'. Broke is later blended with the kraft pulp and the groundwood pulp. The mixed stock is used as a feed to the machine chest. Broke addition rate is largely dictated by the broke inventory. The storage tanks are connected with parallel piping

connections. Typically broke is fed from one or two of the tanks and the other tanks are kept as stand by. Once the level of the feeding tank is below a certain level the flow is switched to the stand-by tanks.

The trend plots of different broke flowrates and the sheet-break indicator are shown in Figure 5.12. The break indicator is a binary signal,'0' indicates normal operation while '1' is for abnormal conditions. Our preliminary study showed significant broke flow variation before several sheet-breaks. To further investigate this issue and confirm consistency of results, a comprehensive analysis was performed on the data with a window of 1 hour before the sheet-breaks on three months of data. The result of the study showed that as many as 30% of the sheet-breaks occurred soon after large changes in broke flowrate or when switching between the tanks took place.

Based on this observation we put together the following hypothesis as the probable mechanism how changes in broke flow can lead to sheet-breaks. As the broke is stored in the tanks for a significant amount of time the fibers get settled at the bottom of the tank. This creates a large concentration gradient within the tank. At the beginning of feeding from a new tank, the fiber concentration will be high in the feed and it will diminish gradually. So whenever there is a change over of the feed tank the consistency of the broke will increase drastically. Therefore, the portion of recycled fibers coming from the broke will be much higher in the sheet. Since recycled fibers have poor bonding capacity the tensile strength of the sheet is expected to be poor, and ultimately this can lead to sheet-breaks.

This hypothesis is also supported by the studies on the effect of re-pulping on fiber bonding by other investigators (McKee 1971, Jahan 2003, Garg and Singh 2006). They concluded that the properties of the fibers are altered by the recycling processes particularly because the fibers have been completely dried, which causes shrinkage in the open structure, and internal bonding takes place within the fiber itself. With fresh pulp, the beating process normally works on internal delamination and opening up of the structure of the fiber walls so that they bond between each other. However, with the dried fiber it can only restore the bonding property partially. In addition to the internal fiber property, there may be reduction of fiber length and some fragmentation of fibers, resulting from the mechanical stresses which are incidental during re-pulping of the dried paper. The conspicuous change in fiber properties upon drying and recycling results in an increase in stiffness and decrease in the flexibility and conformability. In terms of paper properties, this is manifested in decreased bursting and tensile strength, decreased density, increased tearing strength, increased opacity and increased stiffness (Robinson 1980).

Based on the hypothesis, the piping of the broke system has since been changed

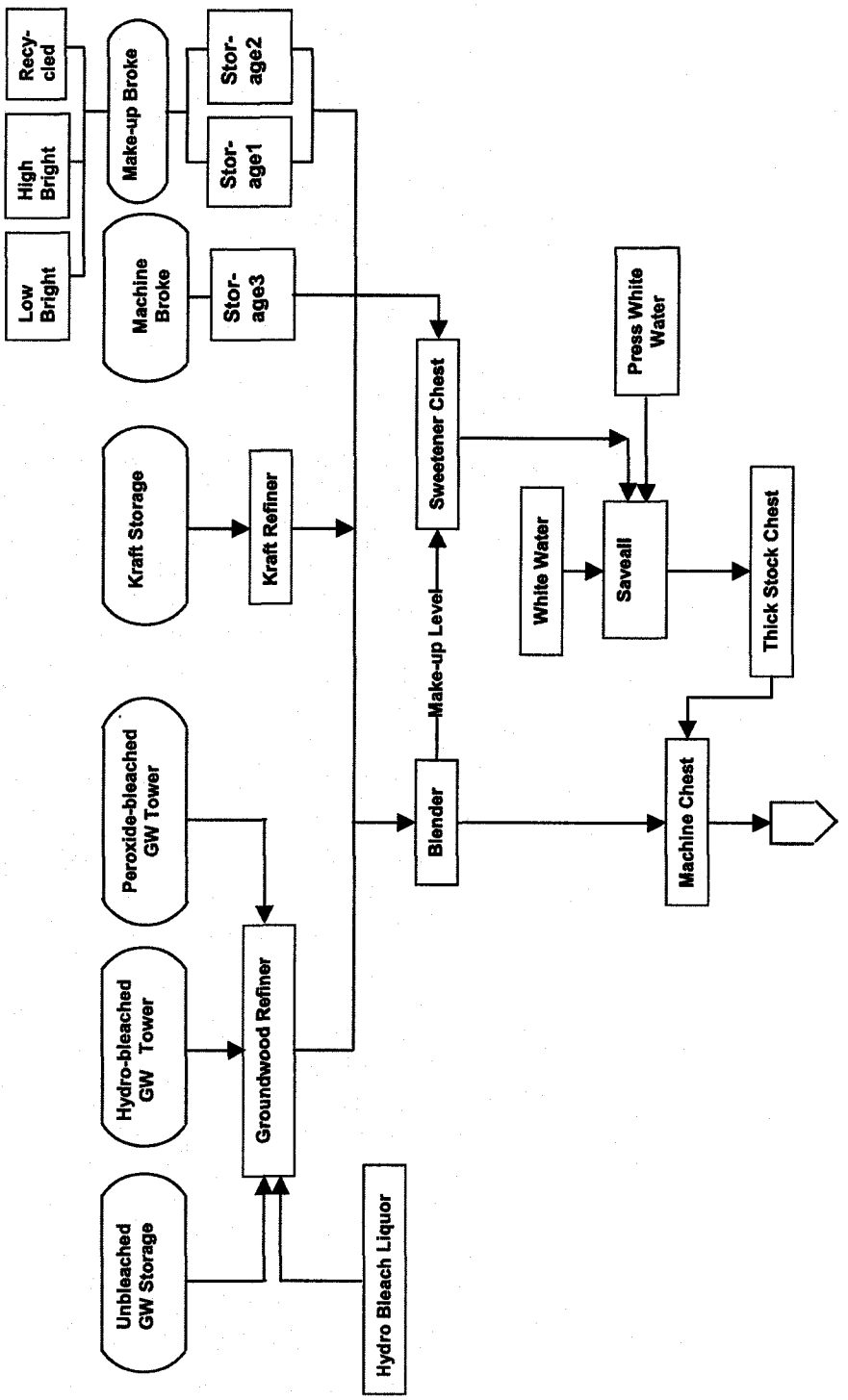


Figure 5.11: Stock approach flow diagram of Paper Machine 7 before making any changes in the process

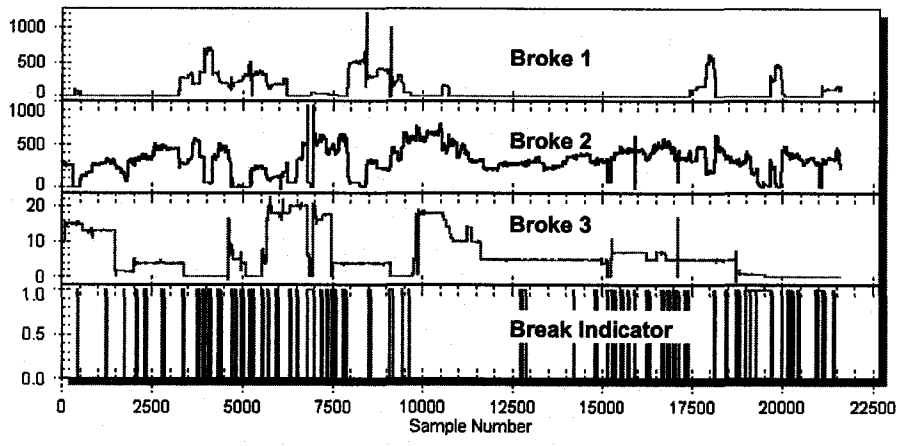


Figure 5.12: *Variation in the broke flowrate and the break indicator showing the close correlation*

from the parallel configuration to a series piping connection, i.e. in Figure 5.11 storage tank 1, 2 and 3 were connected in series and the out flow from storage tank 3 was blended with fresh stock. In this new arrangement the age of the broke remains constant and there is less variation in the consistency and composition (e.g., fiber mix, ash content) of the broke. Therefore, the re-pulped fiber amount will be proportional to the combined flowrate of the broke, and the proportioning of the 'fresh fiber' to 'recycled fiber' can be achieved simply by manipulating the flowrates. In addition to piping changes the furnish control logic was also changed. In contrast to the earlier arrangement where broke replaced all virgin stock, in the new arrangement broke was used as a make up only for the virgin groundwood stock. Therefore, the kraft stock portion which contains the fibers with best bonding capacity, remains constant in the combined stock.

5.6.2 Dryer Pressure System

The sheet leaving the press section contains 60% moisture which is removed in the dryer section. The paper moisture content is controlled by measuring paper moisture at the dry end and manipulating steam inlet pressure and differential pressure across the dryer cylinders. The paper drying process and proper moisture control has major impact on sheet-break. According to the downtime report 60% of sheet breaks took place in the dryer section. Following is a summary of observations from dryer data

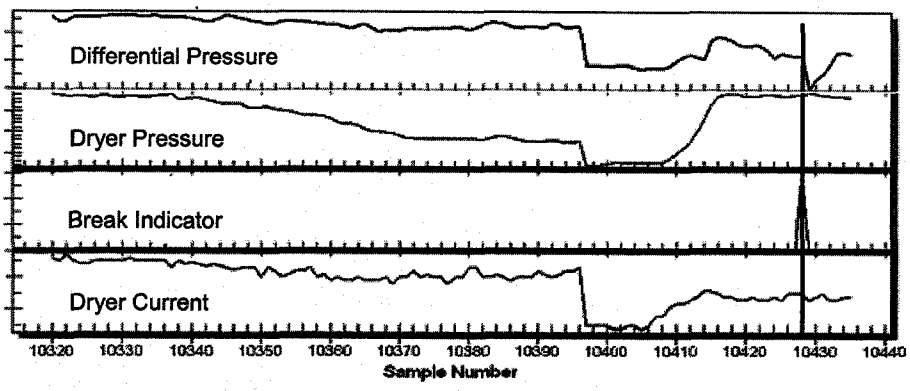


Figure 5.13: *Changes in Dryer Variables before sheet-break*

analysis :

- Several sheet-breaks are preceded by a large decrease in differential pressure of dryer 3 and 4. This is an indication of possible condensate build up in the dryers' can. A representative case of changes in dryer 4 before sheet break is shown in Figure 5.13. Large decrease in pressure and differential pressure is observed from the data. Also a decrease in dryer current is noticed.

- Differential pressures of dryer 1 and 2 were not controlled adequately. The set point and the measured differential pressure had significant bias and very little tracking. A segment of the differential pressure and corresponding set point of dryer 2 is shown in Figure 5.14. This lack of control causes inefficient thermal efficiency in dryer 1 and 2, which has to be compensated in 3rd and 4th dryers.

- The steam temperature in the dryer is almost 100 degrees above the saturation temperature, this decreases the efficiency of drying process.

Based on the observations several changes were made in the dryer control system. The feed pressure valve of the 1st dryer was not working properly. Some repair work was done on the valve. Pressure set points for the dryers were adjusted to keep them within the control range and also the controllers were tuned so that it tracks the set point. After making these changes the drying profile of the paper, i.e. how much water should be removed in each dryer section, improved significantly and a more 'text book' like profile could be achieved.

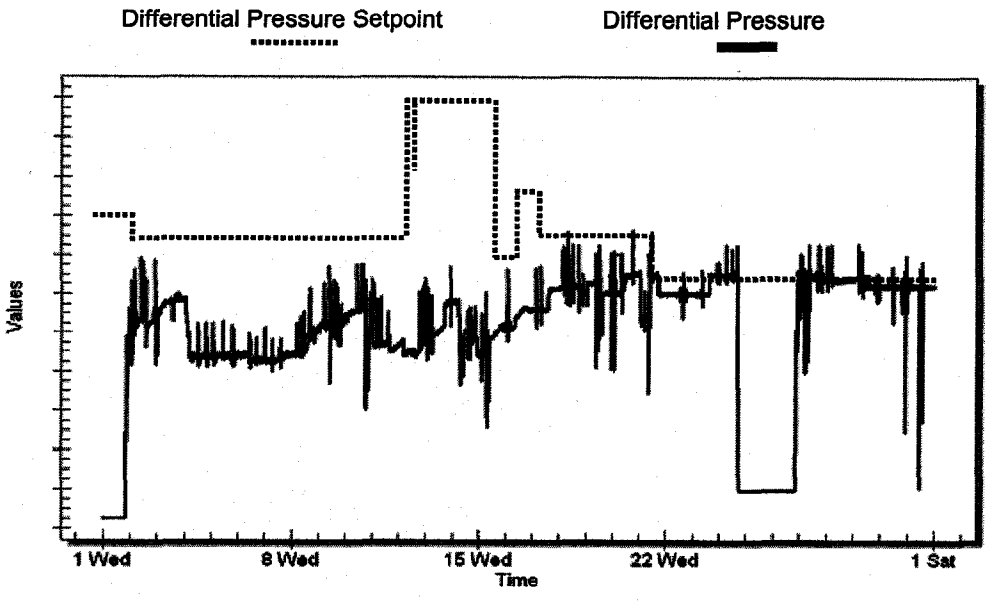


Figure 5.14: *Dryer pressure and set point showing poor tracking performance*

5.7 Key Performance Indicator

A data analysis project has many tangible and in-tangible benefits. Some of the benefits result from the mere fact that more attention is given to the process during such an exercise. As a result it is not easy to attribute the quantifiable and non-quantifiable benefits to the recommendations reported in this paper. However it is clear that, some of the bottlenecks in the process operation were removed due to the additional insight gained in the data analysis. The principal benefits that were gained from this analysis are:

- Detection and diagnosis of cause of some of the sheet-breaks.
- Additional insight in process operation.
- Isolation of the faulty sensors.

In order to quantify the benefits, several key performance indices were calculated from the historical data before and after the changes were made. The performance of the plant prior to the changes are an average of 13 months production history, and performance indices after the changes are calculated from 9 months of data. These two phases have been termed as 'Pre-Project' and 'Post-Project' respectively. The

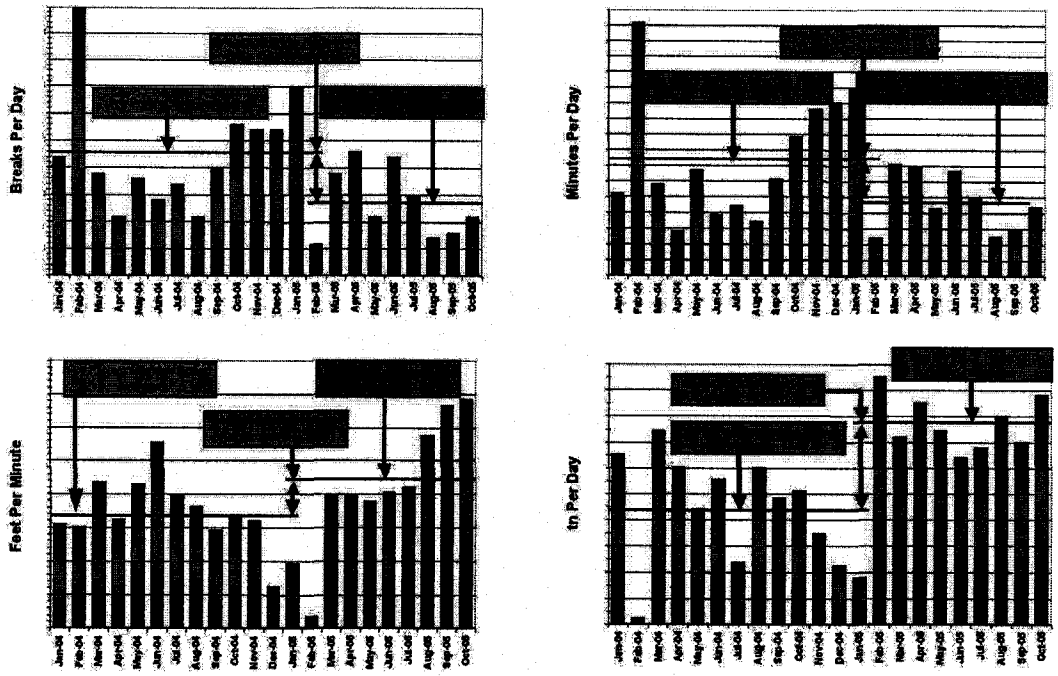


Figure 5.15: *Key Performance Indicators of the process before and after the implementation of the corrective measures*

different Key Performance Indicators (KPI) are shown in Figure 5.15. The y-axis values of the plots have been masked in order to retain confidentiality. However, the relative changes have been stated. A direct impact of the corrective measures is the reduced number of sheet-break occurrences. The number of breaks were reduced almost by one per day. As a result the downtime of the plant was reduced by half an hour per day. The better control of the dryer and lower number of sheet-break gave confidence in the operation. So the machines were operated at significantly higher speeds. This together with the increase in operation time gave a boost in the production. Average production rate of the plant has since increased by 35.1 tons/day.

5.8 Concluding Remarks

In this chapter we describe the successful application of a PCA based monitoring scheme for sheet-break detection and the root cause analysis in a pulp and paper

mill. Several conclusions can be drawn from this study. Some of these apply to general process data analysis while others are specific to pulp and paper process. These conclusions are summarized below:

- Preprocessing of data is important for the success of PCA. Engineering judgement as well as the excitation quality or informative content of the data should be taken into account before including a variable in the data matrix.
- Basis weight was found to be a good criteria for classifying the data into different grades. Grade specific model built for each grades showed better performance in fault detection than a unified model based on data from all grades.
- In the paper industry much of the excitation observed in the variables are due to random variation. Therefore, the PCA model should be used to explain much lower variability than conventionally used.
- The Combined T^2 and SPE Index is a better predictor of sheet-break and gives fewer false alarms.
- Data based analysis can be used for root cause analysis of different operational problems. Corrective measures can significantly improve the machine operation performance and has a direct impact on the total revenue.

Chapter 6

State Estimation of Nonlinear Processes

The estimation of unmeasured states is an important problem in process industries, primarily because knowledge of such states lead to better control. Processes (e.g., power generation, nuclear, chemical etc.) are generally nonlinear. However, often within the region of operation they can be approximated by linear models. The Kalman Filter is by far the most popular linear state estimator and gives unbiased minimum variance estimate under linear and Gaussian circumstances. When the Gaussian assumption of noise is violated, the Kalman Filter is still optimal in a mean squared error sense, but the estimate does not produce the conditional mean (i.e., it is biased) (Chen 2005). However, in many real applications the process dynamics and the measurement equation are non-linear. Therefore a non-linear extension of the Kalman Filter has been developed by many researchers. The Extended Kalman Filter (EKF) is a natural extension of the Kalman Filter to deal with non-linearity and is used widely as a non-linear estimator. The EKF linearizes the nonlinear model using a Taylor series expansion and applies the linear Kalman filter locally. However, in practice EKF has two well-known drawbacks: (i) linearization can produce a highly unstable filter if the assumptions of local linearity is violated (ii) the derivation of the Jacobian matrices often lead to significant implementation difficulties. The Unscented Kalman Filter (UKF) is an elegant way of dealing with the nonlinearity. UKF is based on the idea that it is easier to approximate a Gaussian distribution than to approximate an arbitrary nonlinear function (Julier *et al.* 2004). The UKF, like the Kalman Filter still is only optimal for a Gaussian posterior which can fail in certain non-Gaussian problems with multi-mode and heavy tailed posterior distributions (van der Merwe 2004).

In this context, Sequential Monte Carlo (SMC) methods are very relevant, as

these methods can deal with the nonlinearities, as well as, non-Gaussianity in the posterior density of the states and measurements (Gordon *et al.* 1993). The improved performance of the filters have been demonstrated by many researchers (Doucet 1998, Doucet *et al.* 2000, Arulampalam *et al.* 2002, Chen *et al.* 2004*a*, Chen 2005). In these applications the filtering of the noisy signals has been the main focus and the estimation of unmeasured states received little attention. Therefore, the application of SMC filters in unmeasured state estimation is not yet popular as the implementation details are missing in the literature. Many of the application challenges arise from the fact that obtaining a non-linear model is time consuming and not easy, as it requires fairly accurate physio-chemical or mechanistic description of the process. Therefore, it is not uncommon to see significant mismatch between the process and the model, and unless due attention is given to tuning the filter, the performance of the filter may be very poor. In addition, there also may arise challenges due to the imperfections in the measurements. In process industries all measurements may not be always available at every sampling event. Often measurements may be missing due to lack of sensors, multi-rate sampling strategies or asynchronous sampling, process upset etc. So there is a need to adapt the estimation filter for missing data. In this chapter we investigate these implementation issues and propose novel techniques for tuning the Particle filter and dealing with multi-rate data.

The objectives of the current study are as follows:

- Provide an engineering insight into the SMC filter. Explain different steps of SMC filter in the light of Kalman filter.
- Investigate the effect of measurement noise, process noise and plant model mismatch on state estimation and provide guidelines for the tuning of the SMC filter.
- Develop techniques to implement SMC and Unscented Kalman filters for missing or multi-rate data.
- Compare the performance of SMC filter with Unscented Kalman Filter (UKF).

This chapter is organized as follows: Section 6.1, formulates the problem with the notation details. The theory of recursive Bayesian filter is briefly covered in Section 6.2. Section 6.3 is a tutorial on sampling. Section 6.4 describes the SMC filter, the selection of the 'importance function' and the classification of the filters. The relationship of the particle filter or SIR filter to the broader family of the Bayesian filters is also explored here. An important step in the implementation of the particle filter is

Re-sampling, which is often viewed as a fixing step in the filtering operation. In this section we explain re-sampling from a different perspective and show that re-sampling is an integral part of the particle filter. Different implementation issues of SIR filters are discussed in Section 6.5. The proposed techniques for tuning the particle filter in the presence of significant process-model mismatch and large measurement noise are described in this section. We also present the strategy for handling multi-rate data in a particle filter. Section 6.6 briefly describes the Unscented Kalman Filter (UKF). Section 6.7 contains implementation results of the SIR filter to demonstrate better performance of the proposed techniques on a simulated non-linear CSTR and a laboratory scale four tank system. The results of the comparative study between the SIR and Unscented Kalman filter are reported in this section followed by concluding remarks in Section 6.8.

6.1 Problem Formulation and Notations

The following non-linear form of the state-space model is used for the present filtering and estimation problem.

$$\begin{aligned} x_k &= f\left(x_{k-1}, u_{k-1}\right) + \omega_k \\ y_k &= g\left(x_k\right) + \nu_k \end{aligned} \tag{6.1}$$

where x_k is the state vector, $f(.)$ is the non-linear system equation, $g(.)$ is measurement equation and y_k is the measurement vector. The process system noise term, ω_k, represents disturbances and all unmodelled dynamics; and the measurement noise term, ν_k, captures the inaccuracy and the noise in the measuring devices. Unlike most other state estimation methods, the process noise or measurement noise terms do not necessarily have to be Gaussian, rather depending on the behavior of the process noise or the measurement noise any standard or non-standard distribution can be attributed to the noise characteristics.

Similar to other stochastic filtering problems, the Sequential-Monte-Carlo filtering is an inverse problem. Given the process model $f(.)$ and measurement model $g(.)$, collected measurements y_k at discrete time steps up to the current time $y_{1:n} = [y_1\ y_2\ y_3 \cdots y_n]$ and knowledge about the distribution of the process noise, ω_k and measurement noise, ν_k, the objective is to find optimal $\hat{x}_k$ or $E_{p(x_k|y_{1:k})}[x_k]$ where $k = 1, 2 \cdots, n$.

6.2 Recursive Bayesian Filter

The non-linear filtering problem described in the previous section can also be formulated as a Bayesian estimation problem. The objective of the filter is to obtain the posterior conditional density of the states, $p(x_n|y_{1:n})$. It is also assumed that states are first order Markovian $p(x_n|x_{0:n-1}) = p(x_n|x_{n-1})$

From Bayes rule we have

$$\begin{aligned} p(x_n|y_{1:n}) &= \frac{p(y_{1:n}|x_n)p(x_n)}{p(y_{1:n})} \qquad (6.2) \\ &= \frac{p(y_n, y_{1:n-1}|x_n)p(x_n)}{p(y_n, y_{1:n-1})} \\ &= \frac{p(y_n|y_{1:n-1}, x_n)p(y_{1:n-1}|x_n)p(x_n)}{p(y_n|y_{1:n-1})p(y_{1:n-1})} \\ &= \frac{p(y_n|y_{1:n-1}, x_n)p(x_n|y_{1:n-1})p(y_{1:n-1})p(x_n)}{p(y_n|y_{1:n-1})p(y_{1:n-1})p(x_n)} \\ &= \frac{p(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} \qquad (6.3) \end{aligned}$$

In Equation 6.3 the different terms can be identified as follows:

- $p(x_n|y_{1:n-1})$ is the prior term, which can be further expressed in the following form
$$p(x_n|y_{1:n-1}) = \int p(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1} \qquad (6.4)$$
where $p(x_{n-1}|y_{1:n-1})$ is the estimate of the conditional density of the state from the previous time step, $p(x_n|x_{n-1})$ is the transition density between $(n-1)-th$ to $n-th$ time step, equivalent to the state equation $f(.)$.

- $p(y_n|x_n)$ is the likelihood function and similar to the measurement equation $g(.)$.

- The denominator $p(y_n|y_{1:n-1})$ is a constant, called evidence and often seen as a normalization factor.

The recursive relation of Equation 6.3 does not have a general analytical solution, except for some very restrictive cases, such as, the Kalman Filter for linear systems Gaussian noise models. The Sequential Monte Carlo Filter is also an approximation of the general Recursive Bayesian Filter where the densities are approximated by samples and instead of the density function, samples are propagated in time and the measurements are used to weight the samples. Thereby, the priors are transformed in to the posterior density. The samples truly represent the density function when the

number of samples, $N_p \rightarrow \infty$. Since the densities are represented by samples, one of the challenges is how to sample from a desired distribution. In the following section we discuss the most commonly used methods of sampling from a distribution.

6.3 Sampling from a Distribution

Sampling from a distribution is a fundamental numerical problem that arises in all statistical inference or estimation problems. For example, given conditional pdf $p(x_n|y_{1:n})$ our filtering objective is to obtain the expected value of x_n.

$$E_{p(x_n|y_{1:n})}[x_n] = \int_x x_n p(x_n|y_{1:n}) dx \tag{6.5}$$

For many density functions it is difficult to evaluate the integration analytically and therefore one has to adopt a Monte Carlo method of integration. Monte Carlo integration requires x_n to be sampled from $p(x_n|y_{1:n})$. However it is often not possible to sample directly from the desired distribution, as the probability density function may not be integrable analytically. Therefore, the indirect methods of sampling such as, Rejection Sampling or Importance Sampling are used to sample from the desired distribution. These two methods are described in the following sections.

6.3.1 Rejection Sampling

Rejection sampling is a method to draw independent samples from a probability distribution. Let us consider a case where we would like to sample from the distribution $p(x)$. First we choose a distribution $Q(x)$ from which we can easily draw independent samples and evaluate $Q^*(x)$ at any x. Since the area under each density function is 1, a constant c is chosen such that $cQ(x)$ will cover the total $p(x)$ or equivalently $cQ^* \geq p^*(x) \vee x$. Now samples are drawn from the tractable distribution $Q(x)$ instead of target distribution $p(x)$. At each location $x_i \sim Q(x)$ and the function $cQ^*(x_i)$ is evaluated. A random height is drawn such that $h_i \sim uniform[0 \quad cQ^*(x_i)]$. If $h_i < p^*(x_i)$ then x_i is accepted as a random sample from $p(x)$ otherwise both x_i and h_i are rejected. This process is repeated until sufficient samples have been collected. The scheme is depicted in Figure 6.1 (Murray 2004). There is an optimal value for the coefficient c_{opt} when the envelope density curve touches the target density curve. The method is most efficient for $c = c_{opt}$ as this minimizes the number of rejected samples and algorithm will break down if $c < c_{opt}$.

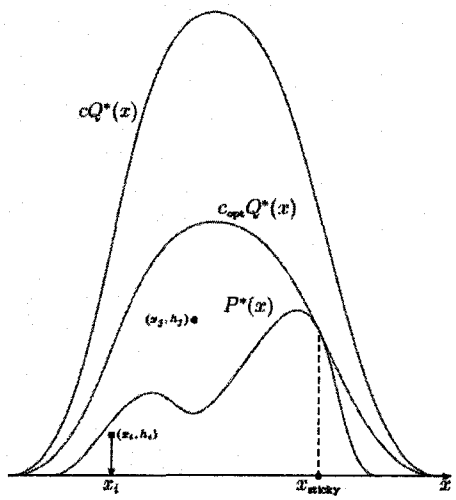


Figure 6.1: *Rejection sampling from a bi-modal distribution using a Gaussian Distribution as Proposal Distribution (Murray 2004)*

6.3.2 Importance Sampling

The idea of importance sampling is to choose a proposal distribution in place of the target probability distribution. The support of the proposal distribution has to cover the support of the target distribution. Instead of sampling from the target probability distribution, each sample is drawn from the proposal distribution, and a weight, equal to the ratio of the two density functions evaluated at the sampled point, is assigned to the sample. The idea is explained using the integration problem stated in Equation 6.5. In order to evaluate the integration we need to sample from the distribution $p(x_n|y_{1:n})$. In case it is difficult to sample directly from the distribution importance sampling can be used to carry out the integration. In order to use importance sampling we cast the problem in the following form:

$$E_{p(x_n|y_{1:n})}[x_n] = \int_{x_n} x_n \frac{p(x_n|y_{1:n})}{q(x_n)} q(x_n) dx_n \tag{6.6}$$

Using direct sampling method independent samples x_n^i are drawn from $q(x)$ and weight $W(x_n^i) = \frac{p(x_n^i|y_{1:n})}{q(x_n^i)}$ where $i = 1 \cdots N_p$ are assigned to the samples. The weighted sum of these samples approximate the expectation.

$$\hat{x}_n = \sum_{i=1}^{N_p} W(x_n^i) x_n^i \tag{6.7}$$

Remarks

The objective of both Importance Sampling (IS) and Rejection Sampling (RS) is to overcome the difficulty of sampling from a distribution which is not integrable analytically. Therefore, more direct methods of sampling, i.e. inverse method cannot be used to sample from the distribution. However, from a implementation and computational point of view these two algorithms are quite different.

- The proposal distribution of Importance Sampling needs to have the same support as the target distribution, however it is not necessary to cover the target distribution. On the other hand, in Rejection Sampling the proposal needs to cover the whole target distribution. This is ensured by selecting a large enough multiplier 'c'. Typically a Gaussian or a mixture of Gaussian probability density function will be used as proposal distribution. If target distribution is non-standard with large peaks in those cases a larger 'c' has to be used.

- In Rejection Sampling, the total number of samples drawn from the proposal distribution will be always higher than the required number (i.e., particle size, N_p) from the target distribution. The number of samples from the proposal distribution will depend on 'c'. For large value of 'c' more samples will be rejected, therefore the computational load will be higher. In Rejection Sampling, no extra sampling is required, the number of samples generated from the 'proposal distribution' will be equal to the required number of samples as a result computational load is lower.

- Samples generated using 'Rejection Sampling' have equal weights ($\frac{1}{N_p}$), on the other hand, a weight equal to the ratio of the target density to proposal density, evaluated at the sampled point ($\frac{p(x_i)}{q(x_i)}$) is assigned to the samples generated using 'Importance Sampling'.

6.4 Sequential Monte Carlo Methods

One of the important aspects of the SMC methods is the recursive update of the posterior. Recursion in time translates the information from one time step to the next time step. Recursion is important mainly for three reasons: i) it minimizes the calculation cost, ii) the effect of the initial condition dies out as more and more measurements become available and iii) the variance of the posterior decreases with time.

Importance Sampling (IS) discussed in the previous section can be used to sample from probability distributions which are difficult to sample from. However, the

formulation as shown in Equation 6.6 is not suitable for recursive estimation. Therefore, an importance function $q(x_{1:n}|y_{1:n})$ is proposed instead of $q(x_n)$, and instead of $p(x_n|y_{1:n})$ we will assume that we would like to sample from $p(x_{1:n}|y_{1:n})$ though we are interested only on the latest state, x_n. Such importance functions allow recursive evaluation of the importance weights in time.

$$E_{p(x_{1:n}|y_{1:n})}[x_n] = \int_{x_n} x_n \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} q(x_{1:n}|y_{1:n}) dx_n \tag{6.8}$$

In Equation 6.8 the recursive weight, $W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})}$ where the proposal distribution $q(x_{1:n}|y_{1:n})$ can be factorized in the following form:

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{1:n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{6.9}$$

Since the states are Markovian $q(x_n|x_{1:n-1}) = q(x_n|x_{n-1})$. Therefore, Equation 6.9 can be further simplified and written as

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{6.10}$$

The target distribution $p(x_{1:n}|y_{1:n})$ can be factorized in the following form:

$$\begin{aligned} p(x_{1:n}|y_{1:n}) &= p(x_n|x_{1:n-1}, y_{1:n}) p(x_{1:n-1}|y_{1:n}) && (6.11)\\ &= p(x_n|y_{1:n}) p(x_{1:n-1}|y_{1:n}) && (6.12)\\ &= \frac{p(y_n|x_n) p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} p(x_{1:n-1}|y_{1:n-1}) && (6.13) \end{aligned}$$

Using the factored form of Equation 6.10 and 6.13 the importance weight, W_n can be updated recursively,

$$\begin{aligned} W_n &= \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} && (6.14)\\ &\propto \frac{p(y_n|x_n) p(x_n|x_{n-1}) p(x_{1:n-1}|y_{1:n-1})}{q(x_n|x_{1:n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1})} && (6.15)\\ &\propto \frac{p(y_n|x_n) p(x_n|x_{n-1})}{q(x_n|x_{1:n-1}, y_{1:n})} W_{n-1} && (6.16) \end{aligned}$$

The recursive form of Equation 6.16 essentially transfers the state and measurement information from all previous time steps to the current time.

6.4.1 Selection of Importance Function

One of the difficult steps in the implementation of the SMC filter is selection of the importance function. Selection of an arbitrary density function $q(x_{1:n}|x_{1:n-1}, y_{1:n})$

can lead to increase in the variance over time (Doucet *et al.* 2000). Thus after a few iterations all the particles will collapse to one point, meaning one normalized weight will be close to '1' and all other weights will become close to zero. This is also known as degeneracy. One way to avoid degeneracy is to select importance function optimally. The concept of optimal importance function was introduced by Zaritskii *et al.* (1975). The optimal importance function is the one which minimizes the variance of the weights to zero and is given by $p(x_k|x_{k-1}^i, y_k)$ (Doucet *et al.* 2000). However, it may not be easy to sample directly from the optimal importance density function (Arulampalam *et al.* 2002). Analytical evaluation of optimal importance function may be possible for some special classes of models. Doucet *et al.* (2000) has derived the optimal importance function for a useful class of models when the state transition equation is nonlinear but measurement equation is linear and the noise is Gaussian. Such a system is given by,

$$x_k = f(x_{k-1}, u_{k-1}) + \omega_k \tag{6.17}$$

$$y_k = Hx_k + \nu_k \tag{6.18}$$

f is a nonlinear function, H is an observation matrix, and ω_k and ν_k are mutually independent and each of them are i.i.d.-Gaussian sequences with $\Sigma_\omega > 0$ and $\Sigma_\nu > 0$. Defining

$$\Sigma^{-1} = \Sigma_\omega^{-1} + H^t \Sigma_\nu^{-1} H \tag{6.19}$$

$$m_k = \Sigma(\Sigma_\nu f(x_{k-1}) + H^t {\Sigma_\nu}^{-1} y_k) \tag{6.20}$$

Therefore, the importance density and the likelihood are given by the following equations respectively,

$$p(x_k|x_{k-1}, y_k) = N(x_k; m_k, \Sigma) \tag{6.21}$$

$$p(y_k|x_k) = N(y_k; Hf(x_{k-1}), \Sigma_\omega + H\Sigma_\nu H^t) \tag{6.22}$$

For nonlinear measurement equation an approximate optimal importance function can be obtained by linearizing the measurement equation. Once linearized, Equations 6.20 through 6.22 may be used to calculate the optimal importance function (Doucet *et al.* 2000). However, this method requires the calculation of the Jacobian of the measurement equation at each time step. Therefore, the method is computationally intensive and may not be easy to implement. Several Monte Carlo methods have been proposed to approximate the importance function and the associated importance weight based on importance sampling (Doucet 1998), Rejection

Sampling and Markov Chain Monte Carlo methods (Berzuini *et al.* 1997, Liu and Chen 1998). However, these methods are iterative in nature, computationally expensive and they lack theoretical convergence results. A simple choice is to use the prior as an importance function which was originally proposed by Handschin and Mayne (1969). This is closely related to the method developed by Gordon *et al.* (1993). The method essentially uses the transition density $p(x_{k+1}|x_k)$ as the importance function and applies re-sampling or Bootstrap to shape the importance function to the posterior. Therefore the method is known as Bootstrap Filter or Particle Filter. In the current study we mainly restrict ourselves to the class of filters which use transition density as the importance function.

6.4.2 SMC with Transition Density

SMC filters have essentially two steps, (i)Prediction and (ii) Update. If the transition density is used as the importance function these two reduce to the implementation of the following actions:

Prediction Each sample is passed through the system model to obtain samples from the prior at time step k: $x_k^{(i)*} = f(x_{k-1}^{(i)*}) + w_{k-1}$ where w_{k-1} is a sample from $p(w_{k-1})$.

Update The update step is essentially an implementation of Bayes rule $p(x_k|x_{k-1}, y_k) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$. However, instead of performing the calculation on density functions, the operation is carried out on the discrete samples. On receipt of the measurement y_k, the likelihood is used to calculate the weights. With the transition density as the importance function, the weight is given by:

$$W_n = \frac{p(y_n|x_n)p(x_n|x_{n-1})p(x_{1:n-1}|y_{1:n-1})}{q(x_n|x_{1:n-1}, y_{1:n})q(x_{1:n-1}|y_{1:n-1})} \tag{6.23}$$

$$\propto \frac{p(y_n|x_n)p(x_n|x_{n-1})}{q(x_n|x_{1:n-1}, y_{1:n})} W_{n-1} \tag{6.24}$$

$$\propto p(y_n|x_n)W_{n-1} \tag{6.25}$$

The generated weights are used to shape the samples from the prior $p(x_k|x_{k-1})$ to the samples from posterior $p(x_k|x_{k-1}, y_k)$.

The above recursive form of the weights can be used directly to implement the update-step or a re-sampling strategy may be applied to shape the priors as proposed by Gordon *et al.* (1993). Based on the implementation strategy of the update-step,

'the SMC algorithm with transition density as importance function' can be classified into two major classes: (i) Sequential Importance Sampling (SIS) (ii) Sequential Importance Re-sampling (SIR) filter commonly known as Particle filter or Bootstrap filter.

6.4.3 Sequential Importance Sampling

In Sequential Importance Sampling (SIS) algorithm the transition density function, $p(x_k|x_{k-1})$ is used as the proposal, therefore $q(x_n|x_{1:n-1}, y_{1:n}) = p(x_k|x_{k-1})$ and the weight update equation reduces to the following form:

$$W_n \propto \frac{p(y_n|x_n)p(x_n|x_{n-1})}{q(x_n|x_{1:n-1}, y_{1:n})} W_{n-1} \tag{6.26}$$

$$\propto p(y_n|x_n)W_{n-1} \tag{6.27}$$

The implementation of SIS algorithm is explained using a flow diagram in Figure 6.2. In the flow diagram the superscript '-' sign is used with the states to emphasize that the sampled states have not been updated, for example, x_k^{i-} denotes a sample at $k-th$ step before measurement update. Clearly the transition of the samples from one time step to the next time step take place through the state transition equation. On the other hand, the measurement information flow through the recursion of the weights. The implementation steps of the SIS algorithm are given in Table 6.1

6.4.4 Sequential Importance Re-sampling

The SIR filter differs from SIS filter in the update step. Measurement update is essentially implementation of Bayes rule. The primary objective of the update step is to shape the prior, $p(x_k|x_{k-1}, y_{1:k-1})$ based on the newly available measurement y_k. In Sequential Importance Re-sampling (SIR) the update step is executed using a re-sampling strategy. After re-sampling the weights of the particles are reset to $\frac{1}{N_p}$. Therefore, the weight update (Equation 6.16) takes the following form:

$$W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} \tag{6.28}$$

$$\propto \frac{p(y_n|x_n)p(x_{1:n-1}|y_{1:n-1})}{q(x_{1:n-1}|y_{1:n-1})} \tag{6.29}$$

$$\propto p(y_n|x_n)W_{n-1} \tag{6.30}$$

$$\propto p(y_n|x_n)\frac{1}{N_p} \tag{6.31}$$

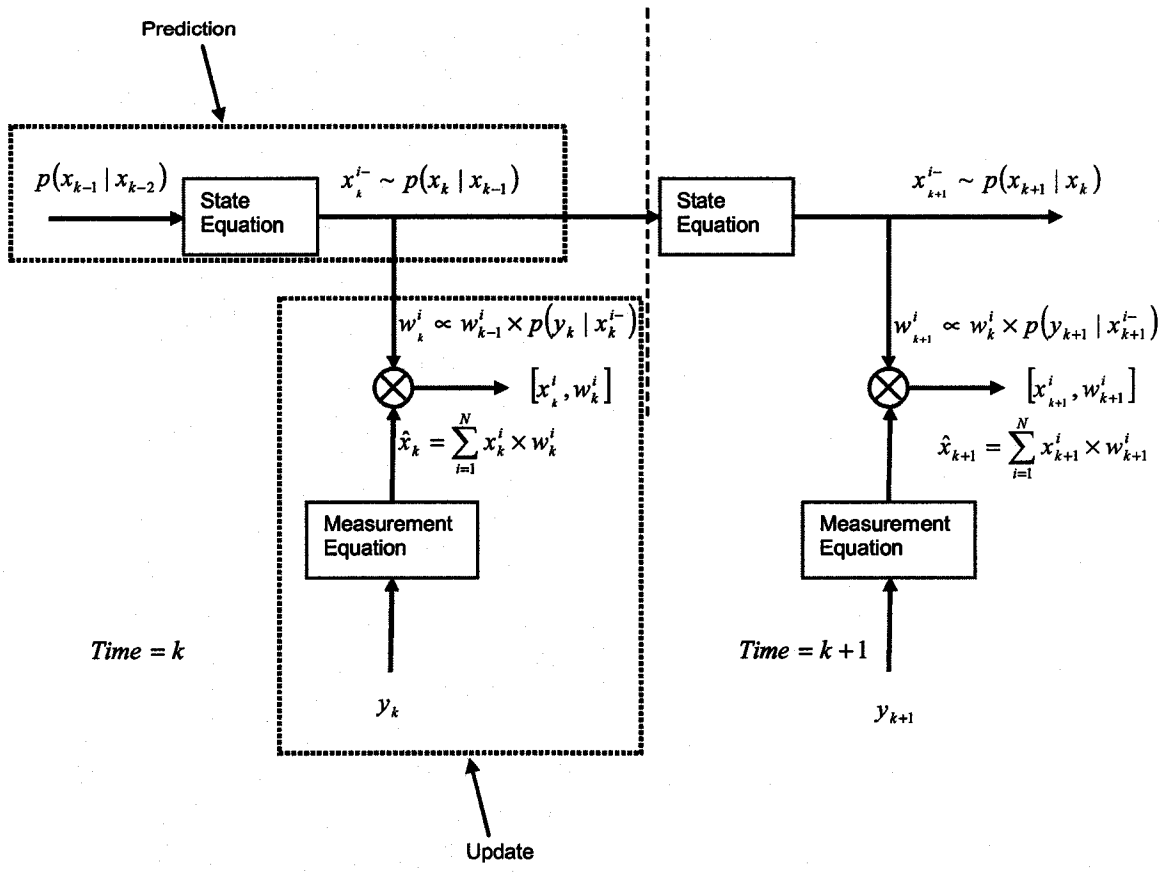


Figure 6.2: *Schematic diagram explaining the implementation steps of the SIS algorithm*

At any given time step, the weight is given by the likelihood function $p(y_k|x_k)$. The justification of using bootstrap or re-sampling is based on the result from Smith and Gelfand (1992), where it is shown that Bayes theorem can be implemented as a weighted bootstrap. The theorem states, suppose samples $[x_i{}^*, i = 1, \cdots, N]$ are available from a continuous density function $G(x)$ and it is required to sample from $L(x)G(x)$, where $L(x)$ is a known function. In that case the discrete distribution $[x_i{}^*, i = 1, \cdots, N]$ with probability mass $\frac{L(x_i^*)}{\sum_{i=1}^{N} L(x_i{}^*)}$ assigned to the samples x_i* tends in distribution to the required distribution as N tends to infinity. Here $G(x)$ can be identified with $p(x_k|x_{k-1}, y_{1:k-1})$ and $L(x)$ with $p(y_k|x_k)$. The other aspect of the update stage is the progression of the measurement information in time direction, so at any time instant the estimated state is based on all the previous measurements as well. In SIR the weight information are transferred to the samples by applying a

Sequential Importance Sampling
Initialization: $x_0^i \sim p(x_0)$; $i = 1, \cdots, N_p$ with $W_0^i = \frac{1}{N_p}$
For t=1,.... , k,...., n
Importance Sampling: Use the state transition equation to translate samples to the next time step, draw N_p samples, $x_k^{i-} \sim p(x_k \mid x_{k-1})$ $i = 1, \cdots, N_p$
Weight Update: Calculate importance weights, $W_k^i \sim p(y_k \mid x_k^{i-}) \times W_{k-1}^i$; $i = 1, \cdots, N_p$
Normalize Importance Weights: $\overline{W}_k^i = \frac{W_k^i}{\sum_{i=1}^{N_p} W_k^i}$; $i = 1, \cdots, N_p$
State Estimation: Expected value of state is given by, $\hat{x}_k = \sum_{i=1}^{N_p} x_k^{i-} \times \overline{W}_k^i$; $i = 1, \cdots, N_p$

Table 6.1: *Sequential Importance Sampling Algorithm using transition prior as proposal and without any re-sampling*

re-sampling or bootstrap strategy. This can be explained using a simple dartboard (e.g., Figure 6.3) analogy.

The numbers or points on the dartboard may be assumed to be the samples $x_k{}^i$ and the '% area' assigned to each of the numbers is the corresponding weight, $w_k{}^i$ of the samples. Now if someone throws a dart on the board the probability of hitting a number will be proportional to the corresponding '% area'. For large number of throws we will have a pool of samples which are distributed according to the weights. Weighted Re-sampling or Bootstrap essentially performs the same operation. However, to make the algorithm computationally efficient and the samples truly random many different re-sampling algorithms have been developed, for example, multinomial re-sampling, residual re-sampling, stratified re-sampling (Douc *et al.* 2005). The re-sampled samples are translated through the transition equation to the next time step and used as priors in that time step. Re-sampling shapes the priors and thereby provides a mechanism of transferring the measurement information from one time step to the next time step. The implementation of SIR algorithm is shown schematically in Figure 6.4.

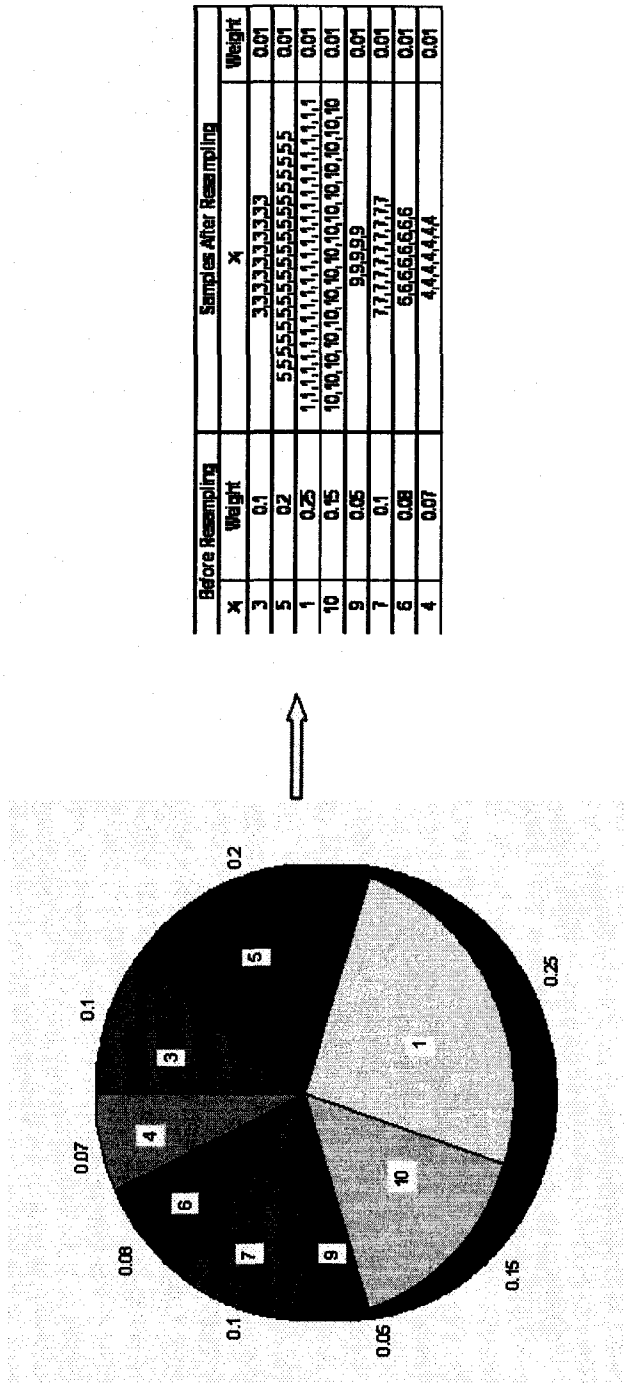


Figure 6.3: *A simple dartboard analogy showing how the weight information is transformed to the samples*

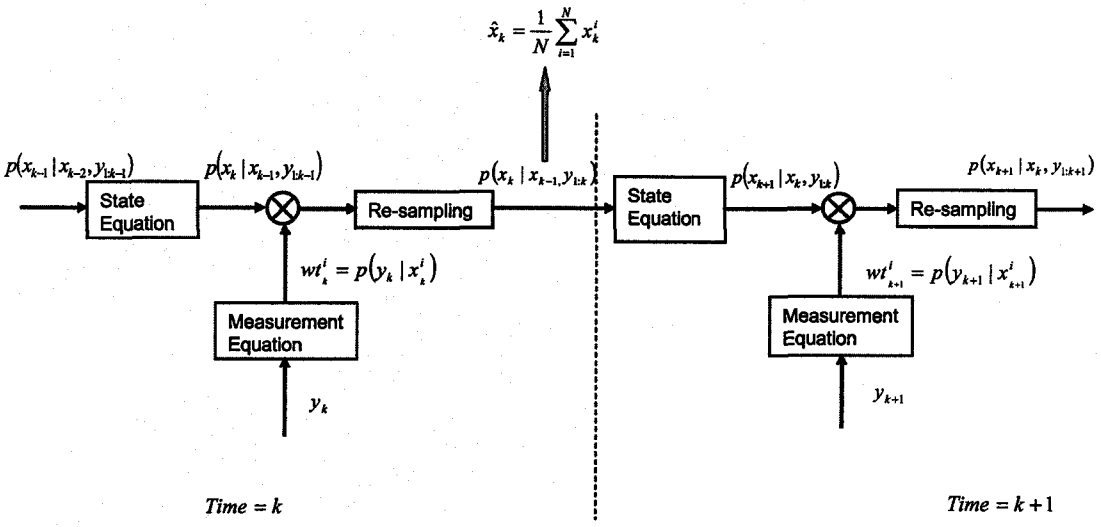


Figure 6.4: *Schematic diagram explaining the implementation steps of the SIR algorithm*

6.5 Implementation Issues in SIR Filter

Several implementation issues have been pointed out by Gordon *et al.* (1993) in their original paper. The problems arise from the fact that a limited sample size, N_p is used for approximating the probability density function. Because of the limited samples, it is important that there is good overlap between the prior and the likelihood. The prior should be proposed in a way such that there will exist many samples from the prior $p(x_k|x_{k-1}, y_{1:k-1})$ in the region where the likelihood $p(y_k|x_k)$ takes significant values. This will ensure that many samples from the prior will receive large weights. If the prior space and the likelihood space are very different, most of the samples will receive very small weights and thereby get wasted. Gordon *et al.* (1993) suggested use of a larger process noise than the actual process noise to expand the proposal distribution. This was called 'roughening' of the prior. Results have been shown that the tracking performance was improved by roughening the priors. However if the SIR filter is used for estimating the unmeasured states, an arbitrary increase of the process noise will introduce mismatch between the process and the model. Therefore, the estimated states may be very oscillatory and in the extreme cases diverge from the true values. Also there is no direct way of tuning the parameters if the filter is being used for estimation of unmeasured states. In this chapter we investigate tuning issues further from a state estimation perspective and propose the following practical ways of tuning the filter.

6.5.1 Weights vs. a-priori state plot

SIR filter uses the transition density as the importance function therefore, the structure of the prior and the likelihood are fixed. The only two parameters that can be used for tuning are the process noise and the measurement noise. The process noise essentially governs the spread of the prior samples and the measurement noise controls the variance of the weights. The increase in the measurement noise makes the weights more uniform. The relative magnitude of the process noise and the measurement noise is also important as it decides whether more weight will be given on the model or the measurement in the estimation process. Therefore, it is important to maintain a balance between these two quantities. However, in the estimation problem the states are not readily available to tune the parameters. We propose a qualitative way of finding the range of the tuning parameters by visualizing the 'weights vs. a priori states' plot. Both the process noise and the measurement noise appear in the likelihood expression. Depending on the relative magnitude of the tuning parameters the shape of the plot will change. A schematic diagram of the weights vs. a priori states plot is shown in Figure 6.5. The region within the dotted lines is the desired shape for the weights. The top curve with large variance in weights may lead to excessive re-sampling and ultimately select only very few samples from the priors. On the other hand, a flat distribution of weights signifies that the update step is not taking a part in shaping the priors. Though according to the discussion of optimal prior we would like to have the variance in the weights close to zero, but from a practical point of view this is not feasible. Primarily because the importance function is not optimal and in this case we would like the re-sampling to shape the priors. From a process model mismatch point of view, since there is mismatch between the process and the model we would like the measurements to play a role in updating the states. Therefore, it is more justified to use a distribution of weights which is slightly curved. The exact shape will depend on a case by case basis. The strategy would be to first select a process noise large enough to show some jitters in the estimated states but such that it does not make it too oscillatory. Then the measurement noise can be increased until we reach the weights' plot which looks similar to that shown in Figure 6.5. The measurement noise can be further tuned to get rid of jitters of the estimated weights. Finally, more tuning can be done to match the measurements with the predicted values.

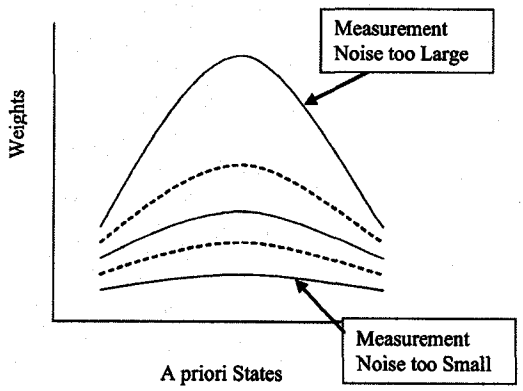


Figure 6.5: *A schematic diagram of 'weight vs. a priori state' plot depicting the optimum region for tuning purpose*

6.5.2 Annealing of the Weights

An increase in process noise produces jitters in the estimated states and increases the variance of the weights. According to the optimal weight estimation strategy, decrease in the variance of the weights leads to selecting diverse samples during the re-sampling and gives better estimates. Therefore, an intuitive way is to use an annealing parameter with the likelihood in order to calculate the weights,

$$wt_k{}^i = p(y_k|x_k{}^i)^\alpha \; where \; \alpha \sim [0 \;\; 1] \tag{6.32}$$

$$Normalized\ weights,\ \bar{wt}_k{}^i = \frac{wt_k{}^i}{\sum_{i=1}^{N} wt_k{}^i} \tag{6.33}$$

where α is the annealing parameter. Decrease of variance of the weights has a direct effect on the re-sampling step. Since the values of weights will be close, the re-sampled values will be a progeny of many values from the original samples. Therefore, the estimate of the state will be an average of a set of diverse samples and any unusual spiking will die out.

The implementation of these two strategies will be further explained using simulation examples in Section 6.7.

6.5.3 Multi-rate Data Handling in SIR

In the process industry often quality variables (e.g., composition) are less frequently measured compared to the condition variables (e.g., temperature) which gives rise to multi-rate data. Since the filters are developed for regularly sampled data, it needs to be adapted for multi-rate data so that it can make use of all available information. Let us consider the system described in Section 6.1. In a multi-rate sampling setting, at any time instant k, the measurements can be divided into two distinct parts $y_k = [{y_k}^{fast} {y_k}^{slow}]$. ${y_k}^{fast}$ is from sensors with faster sampling rate and ${y_k}^{slow}$ is from sensors with a slower sampling rate and only available at some intermittent sampling points, we call these sampling points 'Major Sampling Events' while the instants where only fast sampled measurements are available are called 'Minor Sampling Events'. The prediction step remains unchanged, only the update step needs to be adapted for multi-rate data. The following two modifications of the update step are proposed for dealing with multi-rate data in the SIR filter.

Strategy I The weights at the update step of a SIR filter are given by the likelihood equation:

$$wt^i = p(y_k|{x_k}^i)$$

A two step update procedure can be used to calculate the weights of the SIR filter.

Minor Sampling Event: At the minor sampling event $y_k = {y_k}^{fast}$ and the weights are given by,

$$wt^i = p({y_k}^{fast}|{x_k}^{i,fast}) \tag{6.34}$$

$$R = [R^{fast}] \tag{6.35}$$

Major Sampling Event: The slow sampled measurements are available at the major sampling event and the measurements become $y_k = [{y_k}^{fast} {y_k}^{slow}]$. Weights can be calculated by the following conditional density function:

$$wt_k^i = p([{y_k}^{fast} {y_k}^{slow}]|[{x_k}^{i,fast} {x_k}^{i,slow}]) \tag{6.36}$$

$$R = \begin{bmatrix} R^{fast} & 0 \\ 0 & R^{slow} \end{bmatrix}$$

Also R^{slow} is set to a value much smaller than R^{fast} to give more weight to slow sampled measurements at the major sampling events. The main disadvantage of the method is that, at the minor sampling events the method mainly relies on the prediction equation for estimating the states associated with the slow sampled measurements. This is disadvantageous because in most of the cases there will be a

mismatch between the process and the model. Therefore, the states estimated by the prediction equation should not be trusted absolutely.

Strategy II The major sampling events contain information about the mismatch between the model predicted value and the slow sampled measurement. This information can be used to correct the model predicted value. However, in order to make the estimation robust, a Multiple Imputation (Rubin 1987) strategy can be adopted in the update stage. The main idea of Multiple Imputation is to fill the missing measurements with all possible estimates of the measurements. This will result in several complete data sets. Parameters are estimated for each of the imputed data sets and an average can be used as the estimate. The method provides a way of evaluating the sensitivity of the estimated parameters to the imputed missing values. The details of the method can be found in (Rubin 1987). In the MI framework the update step of SIR filter can be implemented as follows:

Major Sampling Event: At the major sampling event both fast and slow measurements are available $y_k = [{y_k}^{fast} {y_k}^{slow}]$ and the weights are calculated using Equation 6.36. In addition to that, the residual between the slow sampled measurement and the predicted state, $r_k = y_k - \bar{x}_k$ is calculated. These residuals are stored in vector r and used for predicting the unmeasured measurements at minor sampling events.

Minor Sampling Event: All possible values of the missing measurements, ${y_k}^{slow,(d)}$ are calculated by adding the residuals calculated at the Major Sampling Events,

$$\hat{y}_k^{slow,(d)} = \bar{x}_k + r_k^{(d)}$$

The estimated values are imputed in the data set and the complete data sets are given by: ${y_k}^{(d)} = [{y_k}^{fast} {y_k}^{slow,(d)}]$ where $d = [1 \cdots D]$. Weights are calculated for each data set:

$$wt_k{}^{i,d} = p([{y_k}^{fast} {y_k}^{slow,d}] | [{x_k}^{i,fast} {x_k}^{i,slow}])$$

These d vectors of weights are used to re-sample the predicted states. This will result in d estimates of states. Finally expected values and the variances of the states are calculated as follows:

$$E_{p(x_k|y_k)}(x_k) \approx \frac{1}{D}\sum_{d=1}^{D} \bar{x}_k^d = \bar{x}_k \tag{6.37}$$

$$var(x_k) \approx \frac{1}{(N_p - 1)(D-1)} \sum_{i=1}^{N_p}\sum_{i=1}^{D} ({x_k}^{i,d} - \bar{x}_k)^2 \tag{6.38}$$

It has an effect similar to prior boosting and therefore helps in minimizing sample impoverishment. However, the method is computationally intensive.

6.6 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a recursive minimum mean squared error estimator, that addresses some of the issues of EKF. Since EKF only uses the first order Taylor series expansion to linearize the nonlinear function, it often introduces large error in the estimated statistics of the posterior distribution of states. This is specially evident if the model is highly nonlinear and the assumption of local linearity breaks down. In order to deal with the nonlinearity UKF takes a different route and Unscented Transform is at the center of the method. We will first explain Unscented Transform and describe the algorithm.

6.6.1 Unscented Transformation

The basic idea of Unscented Transform is that, it is easier to approximate the probability density function instead of the arbitrary nonlinear function and it introduces less approximation error. A set of sample points called sigma points, χ_i are chosen so that their sample mean and sample covariance are $\bar{x}$ and P_{xx}. These points are selected deterministically so that they capture the mean and covariance information. The n-dimensional random variable x with mean $\bar{x}$ and covariance P_{xx} is approximated by $2n+1$ weighted points given by,

$$\chi_0 = \bar{x} \qquad U_0 = \kappa/(n+\kappa) \tag{6.39}$$

$$\chi_1 = \bar{x} + (\sqrt{(n+\kappa)P_{xx_i}} \qquad U_i = 1/2(n+\kappa) \tag{6.40}$$

$$\chi_{(i+n)} = \bar{x} - (\sqrt{(n+\kappa)P_{xx_i}} \qquad U_{(i+n)} = 1/2(n+\kappa) \tag{6.41}$$

where κ is a tuning parameter that gives an extra degree of freedom to 'fine tune' the higher order moments of approximation. When $x(k)$ is assumed Gaussian, a useful heuristic is to select $n+\kappa = 3$. The associated weights with the sigma points are essentially the probability density function evaluated at the respective points. Each of these points are translated through the nonlinear state transition equation and the measurement equation, which gives a cloud of points and $\bar{y}$ and P_{yy} are the statistics of the transformed points. In order to make the estimated statistics unbiased the transformed points are also weighted accordingly. The implementation steps of the algorithm are given in Table 6.2.

Steps of the Unscented Kalman Filter Algorithm

In a recursion from *k-th* time step to *(k+1)-th* time step

Step 1: The set of sigma points are created by applying Equation 2.39 to 2.41.

Step 2: The transformed set is given by translating each point through the process model,

$$\chi^i(k+1|k) = f\left(\chi^i(k|k), u(k)\right)$$

Step 3: The predicted mean and covariance are calculated,

$$\bar{x}(k+1|k) = \sum_{i=0}^{2n} U_i \chi^i(k+1|k)$$

$$P_{xx}(k+1|k) = \sum_{i=0}^{2n} U_i \left[\chi^i(k+1|k) - \hat{x}(k+1|k)\right]\left[\chi^i(k+1|k) - \hat{x}(k+1|k)\right]^T$$

Step 4: Translate each sigma point through the measurement equation. Predicted observation and mean of predicted observation are given by,

$$y^i(k+1|k) = g\left(\chi^i(k+1|k), u(k)\right)$$

$$\hat{y}(k+1|k) = \sum_{i=0}^{2n} U_i y^i(k+1|k)$$

Step 5: The innovation covariance is given by,

$$P_{yy}(k+1|k) = R(k+1) + \sum_{i=0}^{2n} U^i \left[y^i(k+1|k) - \hat{x}(k+1|k)\right]\left[y^i(k+1|k) - \hat{x}(k+1|k)\right]^T$$

Step 6: Finally the cross correlation matrix is calculated,

$$P_{xy}(k+1|k) = \sum_{i=0}^{2n} U^i \left[\chi^i(k+1|k) - \hat{x}(k+1|k)\right]\left[y^i(k+1|k) - \hat{x}(k+1|k)\right]^T$$

Table 6.2: *Implementation steps of Unscented Kalman Filter*

The Unscented Kalman Filter also needs to be adapted for multi-rate or missing data. However, UKF can be easily modified in a similar fashion as the Extend Kalman Filter (EKF) and these issues have been extensively investigated by Gudi *et al.* (1995) and Prasad (2001). Therefore in this thesis we do not further explore this area. We restrict ourselves in investigating the computational efficiency of UKF. In this study we compared the computation time of UKF with SIR filter for similar type of systems and implemented UKF on a laboratory scale experimental system.

Remarks

- Sequential Monte Carlo Filter (SMC) and Unscented Kalman Filter (UKF) both use the same idea of approximating the probability density by samples. However, in UKF the sampled points are selected deterministically whereas in SMC filters the samples are completely random.
- Since UKF represents the probability density function with only a few strategic points (e.g., in a correlated multivariable Gaussian distribution sigma points will be located on the eigenvector directions) it has more success in representing symmetric Gaussian distributions. For representing skewed Gaussian distribution more points will be required and the weights also need to be adjusted accordingly. Derivation of weights may not be possible for non standard distributions.

6.7 Results

In this section we illustrate the proposed methodology to overcome different implementation difficulties of Particle Filter using a simulated non-linear adiabatic CSTR (Henson and Seborg 1997) and a laboratory scale Four Tank setup.

6.7.1 Non-linear CSTR

A non-linear CSTR is simulated where the operating conditions have been taken from (Henson and Seborg 1997). This system was also studied by (Chen *et al.* 2004*b*) to demonstrate the different aspects of SIR Filters. The governing equations of the system are as follows:

$$\begin{aligned} \frac{dC}{dt} &= \frac{q}{V}(C_0 - C) - kCe^{-E_A/T} \\ \frac{dT}{dt} &= \frac{q}{V}(T_0 - T) - \frac{\Delta H}{\rho C_p}kCe^{-E_A/T} - \frac{UA}{\rho C_p V}(T - T_c) \end{aligned} \tag{6.42}$$

where C is the concentration, T is the temperature, q is the flowrate, V is the volume of the reactor, C_0 and T_0 are inflow concentration and temperature, $kCe^{-E_A/T}$ is the reaction rate, ΔH is the heat of reaction, ρ is the density, C_p is the specific heat, U and A are the effective overall heat-transfer coefficient and area of heat transfer, T_c is the temperature of the cooling fluid. The continuous differential equations are discretized via finite difference, resulting in the following discrete model:

$$\begin{aligned} C_k &= C_{k-1} + (\frac{q}{V}(C_0 - C_{k-1}) - kC_{k-1}e^{-E_A/T_{k-1}})\Delta t + \omega_k^I \\ T_k &= T_{k-1} + (\frac{q}{V}(T_0 - T_{k-1}) - \frac{\Delta H}{\rho C_p}kCe^{-E_A/T_{k-1}} - \frac{UA}{\rho C_p V}(T_{k-1} - T_c)) + \omega_k^{II}\Delta t \end{aligned}$$

where $\omega_k = [\omega_k^I \ \omega_k^{II}]^T$ is the process noise and Δt is the time interval between two consecutive time steps.

We consider two measurement strategies for the system. In a single rate sampling setup only the temperature sensor is available and measured at 1 sec sampling interval. The measurement equation has the following form:

$$y_k = 0.5T_k + \nu_k^I \tag{6.43}$$

A multi-rate sampling strategy was also considered to demonstrate the performance of the proposed multi-rate SIR filter. In a multi-rate sampling setup the temperature measurements are available at every second, while concentration is measured at every 10 sec interval. The measurement equation at the major sampling instant can be written as,

$$y_k^{major} = 0.5 \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} T_k \\ C_k \end{bmatrix} + \nu_k^{major} \tag{6.44}$$

The operating conditions of the CSTR are given in Table 6.3. The system was also studied for these operating conditions by Henson and Seborg (1997) and Bakshi and Stephanopoulos (1996). The normalization factors used for this operating condition are $C_r = 1 \ mol/L$ and $T_r = 100K$.

The normalized initial condition of the system is, $C_{initial} = 0.5$ and $T_{initial} = 3.5$. One of the main objectives of this exercise is to study the effect of system noise and process noise on the estimation performance of the SIR filter. Therefore, the system was simulated for a wide range of process noise and measurement noise. We also consider poor guesses as initial condition for the system which is common in many practical situations, to demonstrate the convergence of the algorithm.

parameter	value	units	parameter	value	units
q	100	L/min	C_p	0.239	$J/g/K$
V	100	L	U	5.0×10^3	$J/cm^2/min/K$
C_0	1.0	mol/L	E_A	8750	K
k	7.2×10^{10}	$1/min$	T_0	350	K
δH	-5.0×10^4	J/mol	A	10	cm^2
r	1000	g/L	T_c	305	K

Table 6.3: *Parameters of the non-linear CSTR*

Tuning of SIR filter

The graphical method described in Section 6.5.1 has been used to tune the SIR filter for the CSTR system. The process noise and the measurement noise for the system on a normalized scale are,

$$p(\omega_k) \sim N\left(\begin{matrix}0\\0\end{matrix}, \begin{matrix}0.0001 & 0\\0 & 0.0004\end{matrix}\right) \tag{6.45}$$

$$p(\nu_k) \sim N(0, 0.2) \tag{6.46}$$

In the filter we used a process noise one order higher than the actual process noise. Setting the process noise to a higher value makes the prior wide and helps the prior to cover the actual state of the system. However, the variance should not be increased more than one order of magnitude as it gives rise to large oscillations in the estimated states of the unmeasured variables. Keeping the process noise to the above value we changed the measurement noise of the process and at the same time observed the shape of the 'weight vs. a-priori state' plots. The shape of the weights for four different cases are shown in Figures 6.6(a) to (d). In each of these figures we also plotted the distribution of the states before and after re-sampling. Clearly if the variance of the weights is very large, the distribution of the state changes significantly after re-sampling. The sum of squared errors between the estimated concentration and actual concentration for these four cases are plotted in Figure 6.7 in the same order. The smallest sum squared error is obtained for Case (c). From the 'weight vs. a-priori state plot' it is also evident that the variance of the weights is moderate for Case (c), and due to re-sampling the distribution of the weights did not change significantly. Therefore under this condition both the model and the measurements played equal roles in estimating the states.

The 'Weights vs. a-priori plot' gives a qualitative idea about the ratio of the

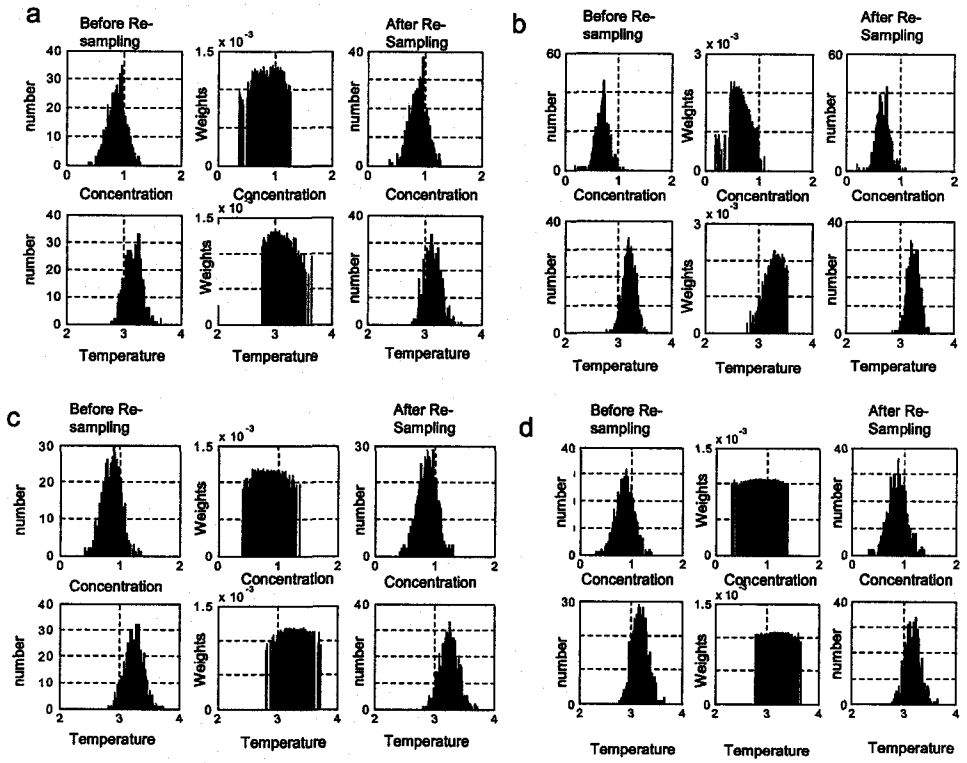


Figure 6.6: *Weights vs. a priori state plot depicting the tuning methodology of the non-linear CSTR*

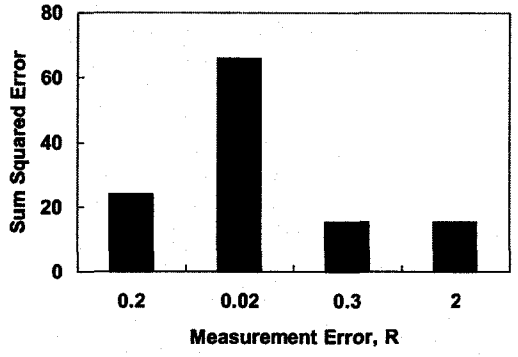


Figure 6.7: *Effect of tuning measurement noise on the sum squared error between the concentration and estimated concentration*

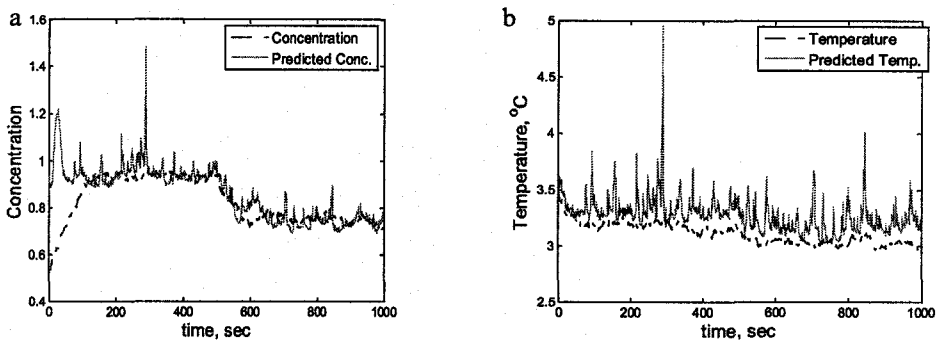


Figure 6.8: *Trend plots of the actual and predicted concentration and temperature showing jitters in the estimates due to poor tuning*

process noise and the measurement noise where the filter may work best. The method works well if an order of magnitude information about the variance of the process noise is available. However, in addition to the 'Weights vs. a-priori plot' we suggest visual inspection of the trend plots of the measured states and the estimated states.

Annealing of the weights is another way of tuning the particle filter. The effect of annealing of the weights is demonstrated in Figures 6.8 and 6.9. In both cases we used the same process noise and measurement noise as tuning parameter. However Figure 6.9 shows the estimates of the states when an annealing parameter $\alpha = 0.2$ was used. Clearly the jitters have disappeared and the trend of the estimated states are very similar to the expected behavior of the states.

Application to Multi-rate Data

The Sum Squared Error between the actual concentrations and the estimated concentrations are plotted in Figure 6.10. The SSE are calculated from 20 Monte Carlo simulations. The additional information made the system states more observable and improved the estimates of the states if the filter is well tuned. For example, in the multi-rate case there are two tuning parameters. Figure 6.10 shows the SSE values for different level of errors in concentration measurements. Cases where only temperatures measurements are used for updating do not change with this tuning parameter as in the weight calculation concentration error term do not appear in the likelihood equation. For the well tuned case, the Multiple Imputation strategy gives the lowest SSE, however computationally by far it is the most expensive method. Computa-

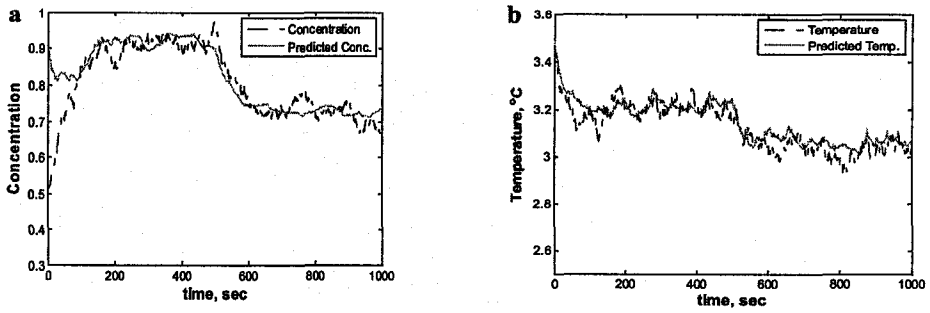


Figure 6.9: *Trend plots of the actual and predicted concentration and temperature showing smooth behavior of the predicted states due to annealing of weights*

tional load is approximately one order higher than that of Strategy-I. In addition to this, performance of the Multiple Imputation strategy depends on the number of imputations.

6.7.2 Experimental Four Tank System

Experiment on the Four Tank system was carried out at the process control laboratory at the University of Alberta. A mechanistic model of the system was developed using the measured dimensions of the tank. The schematic diagram of the process is shown in Figure 6.11. A comprehensive analysis of the process can be found in Johansson (2000). Mass balance and Bernoulli's law yield the following set of equations:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{f_1}{A_1} \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_3} + \frac{a_4}{A_4}\sqrt{2gh_4} + \frac{f_2}{A_2} \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{f_3}{A_3} \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{f_4}{A_4}
\end{aligned} \tag{6.47}$$

where A_i is the cross-sectional area of the tank, a_i cross-section of the outlet hole and h_i water level. The process inputs are water flowrates, $[f_1, f_2, f_3, f_4]$ and outputs are measurements of the bottom two levels $[y_1,\ y_2]$. The dimensions of the laboratory process are given in Table 6.4.

The process has four states(i.e., heights of the tanks are states of the system). While the measurements of all four heights are available, in applying the filter we

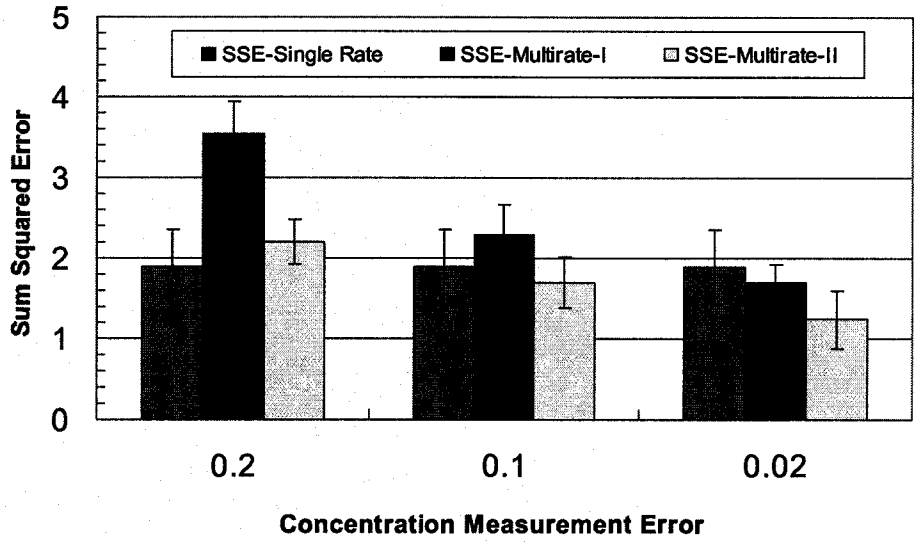


Figure 6.10: *Mean Squared Error comparing the estimation performance of the multi-rate strategy with the single rate strategy*

parameter	unit	value
A_1 , A_2	cm^2	392.7
A_3, A_4	cm^2	392.7
a_1, a_2	cm^2	1.54
a_3, a_4	cm^2	0.75

Table 6.4: *Parameters of the laboratory scale four tank system*

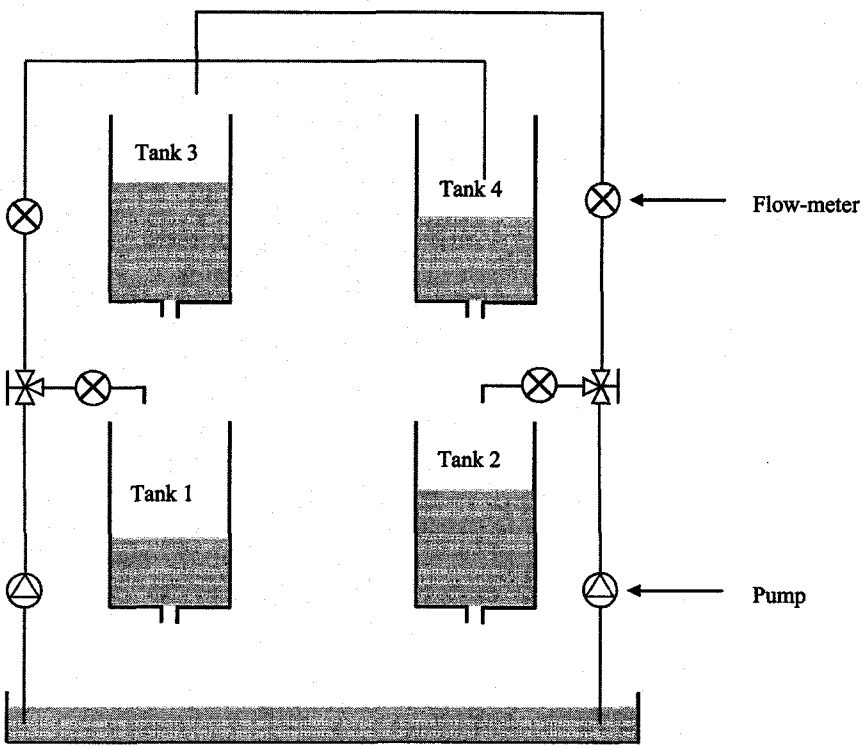


Figure 6.11: *Schematic diagram of the laboratory scale four tank system*

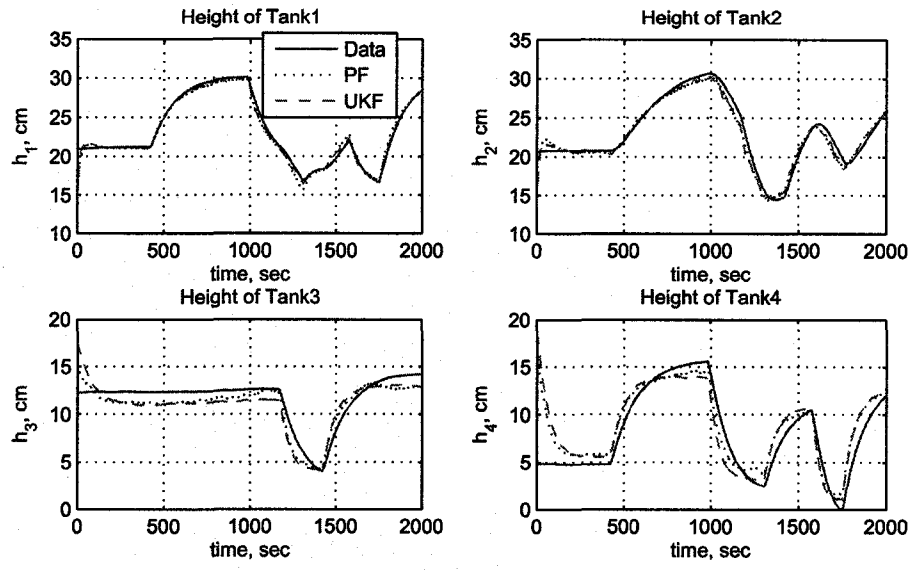


Figure 6.12: *Trend plots of the heights of the Experimental Four tank system and the predicted values by Particle Filter and Unscented Kalman Filter*

assume that only the heights of tank 1 and tank 2 are measured and heights of tank 3 and tank 4 are unmeasured. Thereafter both SIR filter and UKF are applied to estimate these two unobserved states. The SIR filter is tuned using the tuning rule discussed in the previous sections. The trend plots in Figure 6.12 shows good agreement between the true states and the estimated states by the SIR filter. The tuning of UKF was done by trial and error and the trend plots of the measured and unmeasured states were used to see the effect. The UKF also gives a similar estimate of the states. In terms of estimation quality the two methods are comparable for this system. We also recorded the execution time for both filters. The execution time for this system is shown in Figure 6.13. This is an average of twenty monte carlo simulations. It is clearly evident that the computational load of the Unscented Kalman Filter is much lower than SIR filter. Therefore the UKF is suitable for extremely fast sampling rates, for example, filtering of vibration data. However, we have to remember that the UKF has limited applicability and is not optimal for estimating skewed and heavy tailed states.

6.8 Concluding Remarks

The following conclusions can be drawn from the study.

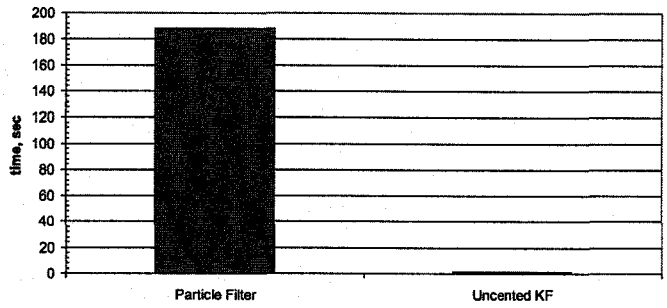


Figure 6.13: *Execution time for the SIR filter and the UKF on the experimental four-tank system*

- The proposed tuning of Particle Filter using 'Weights vs. a priori state' plot was tested on non-linear processes with a wide range of process noise and measurement noise. Here we reported partial results from one simulated and one experimental case study. This is an easy to use effective visualization tool which can help the practicing engineers to tune the Particle Filter.

- The Particle Filter is also extended to the Multiple Imputation framework for dealing with multi-rate data. In the presence of large process and measurement noise, the additional information from slow sampled measurements improve the estimates of the states.

- The Unscented Kalman Filter (UKF) is very fast compared to the SIR or Particle Filter. Therefore, the UKF may be suitable for state estimation of fast non-linear processes. However, the tuning of the UKF for state estimation remains an open problem.

Chapter 7

Concluding Remarks and Recommendations for Future Work

This chapter lists the contributions of this thesis and the directions identified for future research.

7.1 Contributions of This Thesis

The main contribution of this thesis can be summarized as follows:

- This thesis acts as a bridge between the vast literature on missing data in the area of statistics and analysis of process data with missing values. It explores the similarities and dissimilarities of missing value problem in process data analysis and related subjects (e.g., statistical surveys).
- A tutorial introduction of different concepts and methods related to missing data problem is presented in Chapter 2. These concepts have been explained with examples related to process data analysis.
- A general framework for classifying the symmetrical latent variable models is proposed. Commonly used latent variable models are classified according to this framework. This classification can serve as a guideline to select latent variable models for different process characteristics.
- Principal Component Analysis (PCA), Iterative Principal Component Analysis (IPCA) and Maximum Likelihood Factor Analysis (MLFA) have been extended to Data Augmentation framework for building model from data with missing values. The estimated models using proposed methods are of better quality compared to the models from use of the conventional methods.

- Multivariate missing data handling technique is combined with Dynamic Time Warping (DTW) to synchronize the un-even length batch process data. The proposed method conserves the correlation between the variables and leads to a compact latent variable model.
- A detailed study is carried out on the impact of data compression on multivariate modeling. The analysis shows that, transform methods preserves the correlation between the variables better compared to direct methods.
- Compressed data have been formulated as a missing data problem. A multivariate missing data handling technique has been used to restore the correlation between the variables.
- The importance of data pre-processing is highlighted in Chapter 5. Besides missing values, process data suffer from many different types of flaws (e.g., frequent downtime, grade change etc.). These issues have been dealt with in connection with an industrial case study. A PCA based monitoring scheme is developed to detect and diagnose the root cause of sheet-breaks in a pulp and paper mill. The pre-processing of the data played a significant role in the success of the monitoring scheme.
- A tutorial on the Sequential Monte Carlo (SMC) methods is presented in Chapter 6. In general chemical engineers are not familiar with sampling theory. This tutorial will serve as an introduction to this area.
- Several tuning methods have been developed to implement SIR or Particle filter in the presence of large measurement noise and process model mismatch.
- The Particle filter has been extended to Multiple Imputation (MI) framework so that the filter can be updated using multi-rate data.

7.2 Recommendations for Future Work

Treatment of missing data in process industries remains a challenging area for further research. Some areas of further research that were identified during the course of our research are listed below:

- In this thesis we mainly focused on building latent variable models from historical data. On-line application of latent variable models have been investigated by Nelson and MacGregor (1996). However, calculation of T^2 and SPE test statistics and their confidence interval still remains an open problem.

- The treatment of missing data in dynamic process monitoring models (e.g., Dynamic PCA) have been outlined in this thesis. Extension of IPCA for dynamic data and dealing with missing data is still an open problem.
- The EM algorithm has been used for building time series models from data with missing values (Shumway and Stoffer 1982). However, the algorithm can be difficult to implement if the closed form solution of the parameters are not available. Instead of using EM, Data Augmentation may be used to estimate time series models from data containing missing values.

Bibliography

Afifi, A. A. and R. M. Elashoff (1966). Missing observation in multivariate statistics. *Journal of the American Statistical Association* **61**, 595–604.

Allison, P. D. (2002). Inference and missing data. *Sociological Methods and Research* **28**(3), 301–309.

Anderson, T.W. (1984). *An Introduction to Multivariate Statistical Analysis.* 2 ed.. John Willy and Sons.

Artega, R. and A. Ferrer (2002). Dealing with missing data in MSPCS: several methods, different interpretations, some examples. *Journal of Chemometrics* **16**, 408–418.

Arulampalam, M. S., S. Maskell, N. Gordon and T. Clapp (2002). A tutorial on particle filter for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* **50**(2), 197–188.

Azen, S. and M. Van Guilder (1981). Conclusions regarding algorithms for handling missing data. *Proceeding of Statistical Computation Section, American Statical Association* pp. 53–56.

Bakshi, B. and G. Stephanopoulos (1996). Compression of chemical process data through functional approximation and feature extraction. *AIChE Journal* **42**, 477–492.

Baum, L. E., Petrie T. Soules G. and N. Weiss (1970). A maximization technique occuring in the statistical analysis of probilistic function of markov chains. *Annals of Mathematical Statistics* **41**, 164–171.

Beale, E. M. L. and R. J. A. Little (1975). A method of estimation of missing values suitable for use with electronic computer. *Journal of Royal Statistical Society B* **37**, 129–145.

Berzuini, C., N. Best, W. Gilks and C. Larizza (1997). Dynamic conditional independence models and markov chain monte carlo methods. *Journal of American Statistical Association* **92**, 1403–1412.

Bissessur, Y., E. B. Martin and A. J. Morris (1999). Monitoring the performance of the paper making process. *Control Engineering Practice* **7**, 1357–1368.

Bjorck, A. and G. Golub (1973). Numerical methods for computing angles between linear subspaces. *Mathematical Computations* **27**, 579–594.

Bonissone, P. and K. Goebel (2002). When will it break? a hybrid soft computing model to predict time-to-break in paper machine. General Electric Global Research Centre. Report.

Bristol, E. H. (1990). Swinging door trending: adaptive trend recording. *ISA National Conference Proceedings* pp. 749–753.

Buck, S. F. (1960). A method of estimation of missing values suitable for use with electronic computer. *Journal of Royal Statistical Society B* **22**, 302–306.

Champagne, M. and I. Ivanov (2002). Multigrade modelling - paperboard quality modeling. *Proceedings of the American Control Conference* **11**(6), 8–10. Anchorage. AK.

Chan, N. N. and T. K. Mak (1985). Estimation in multivariate errors-in-variable models. *Linear Algebra and Its Applications* **70**(426), 197–207.

Chen, J. and K. C. Liu (2002). On-line batch process monitoring using dynamic pca and dynamic pls models. *Chemical Engineering Science* **57**, 6375.

Chen, W. S., B. R. Bakshi, P. K. Goel and S. Ungarala (2004*a*). Bayesian estimation via sequential monte carlo sampling: Unconstained nonlinear dynamic systems. *Industrial Engineering and Chemistry Research* **43**, 4012–4025.

Chen, W. S., B. R. Bakshi, P. K. Goel and S. Ungarala (2004*b*). Bayesian estimation via sequential monte carlo sampling: Unconstained nonlinear dynamic systems. *Industrial Engineering and Chemistry Research* **43**, 4012–4025.

Chen, Z. (2005). *Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond.* Manuscript. McMaster University, Hamilton, Ontario, Canada.

Cochran, W. G. and D. B. Rubin (1973). Controlleing biases in observational studies: A review. *Sankya A* **35**, 417–446.

Colledge, M. J., Johnson J. H. Pare R. and I. G. Sande (1978). Large scale imputation of survey data. *Proceeding of Survey Research Methodology, American Statistical Association* pp. 431–436.

Dempster, A. P., Laird N. M. and D. B. Rubin (1977). Maximum likelihood from incomplete data via *EM* algorithm. *Journal of the Royal Statistical Society. Series B(Methodological)* **39**(1), 1–38.

Douc, R., O. Cappe and E. Moulines (2005). Comparison of resampling schemes for particle filtering. personal communication.

Doucet, A. (1998). On sequential simulation-based methods forbayesian filtering. *Technical Report. University of Cambridge, Dept. of Engineering, CUED-F-ENG-TR310. Available on the MCMC preprint service at http://www.stats.bris.ac.uk/MCMC.*

Doucet, A., S. Godsill and C. Andrieu (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing* **12**, 197–208.

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

Fuller, W.A. (1987). *Measurement Error Model.* John Willy and Sons.

Garg, M. and S. P. Singh (2006). Reasons of strength loss in recycled pulp. *APPITA JOURNAL* **59**(4), 274–279.

Gelman, A., Carlin J. B. Stern H. S. and D. B. Rubin (2004). *Bayesian Data Analysis.* 2 ed.. Chapman and Hall/CRC.

Gollmer, K. and C. Postens (1995). Detection of distorted pattern using dynamic time warping algorithm and application for supervision of bioprocess. *IFAC Workshop on On-Line Fault Detection and Supervision in Chemical Process Industries.* Newcastle.

Gordon, N. J., D. J. Salmond and A. F. M. Smith (1993). Novel approach to nonlinear or non-gaussian bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing* **140**, 107–113.

Grung, B. and R. Manne (1998). Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **42**, 125–139.

Gudi, R. D., S. L. Shah and M. R. Gray (1995). Adaptive multirate state and parameter estimation strategies with application to a bioreactor. *AIChE Journal* **41**(11), 2451–2463.

Haitovsky, Y. (1968). Missing data in regression analysis. *Journal of Royal Statistical Society B* **30**, 67–81.

Hale, J. H. and H. L. Sellars (1981). Historic data recording for process computers. *Chemical Engineering Progress* **77**, 38–43.

Handschin, J. E. and D. Q. Mayne (1969). Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control* **9**, 547–559.

Hartley, H. O. (1958). Maximum likelihood estimation from incomplete data. *Biometrices* **14**(2), 174–194.

Henson, M. A. and D. E. Seborg (1997). *Nonlinear Process Control.* Prentice Hall PTR. Upper Saddle River, NJ.

Imtiaz, S. A., M. A. A. S. Choudhury and S. L. Shah (2005). Building multivariate model using compressed data. *Submitted to Industrial and Engineering Chemistry Research.*

Jackson, J. E. (1991). *A user's guide to principal components.* Wiley-Interscience publication.

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

Jahan, M. S. (2003). Changes of paper properties of nonwood pulp on recycling. *TAPPI JOURNAL* **2**(7), 9–12.

Jöreskog, K. G. (1970). A general method for analysis of covariance structures. *Biometrika* **57**(2), 239–251.

Jennrich, R. I. and P. F. Sampson (1966). Rotation for simple loadings. *Psychometrika* **31**(3), 313–323.

Johansson, K. H. (2000). The quadruple-tank process: A multivariable laboratort process with an adjustable zero. *IEEE Transactions on Control Systems Technology* **8**(3), 456–465.

Julier, S. J., J. K. Uhlmann and H. F. (2004). Unscented filtering and nonlinear estimation. *Proceedings of IEEE* **92**(3), 401–422.

Kim, J. O. and J. Curry (1977). The treatment ofmissing data in multivariate analysis. *Sociological Methods and Research* **6**, 215–240.

Knyazev, A. W. and M. E. Argentati (2002). Principal angles between subspaces in an a-based scalar product: Algorithms and perturbation estimates data. *SIAM Journal of Scientific Compuations* **23**(6), 2008–2040.

Kourti, T. P., J. Lee and J. F. MacGregor (1996). Experience with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering* **20**(Suppl. A), S745–S750.

Kresta, J., J. F. MacGregor and T. E. Marlin (1991). Multivariate statistical monitoring of process operating performance. *CJChE* **69**, 35–47.

Krzanowski, W. J. (1979). Between-groups comparison of principal components. *Journal of American Statistical Association* **74**(367), 703–707.

Laird, N. M. and J. H. Ware (1982). Random-effects models for longitudinal data. *Biometrics* **38**, 963–974.

Lakshminarayanan, S. R., R. D. Gudi, S. L. Shah and K. Nandakumar (1996). Monitoring batch processes using multivariate statistical tools: Extensions and practical issues. *IFAC Triennial World Congress.* San Francisco.

Lawley, D. N. and A. E. Maxwell (1963). *Factor Analysis as a Statistical Method.* London: Butterworths.

Lazraqa, A., R. Clroux and J. P. Gauchi (2003). Selecting both latent and explanatory variables in the pls1 regression model. *Chemometrics and Intelligent Laboratory Systems* **66**, 117–126.

Li, K. H. (1985). *Hypothesis testing in multiple imputation with emphasis on mixed-up frequencies in contingency tables.* Unpublished PhD dissertation, University of Chicago, Statistics Department.

Li, W. H. and S. J. Qin (2001). Consistent dynamic pca based on errors-in-variables subspace identification. *JOURNAL OF PROCESS CONTROL* **11**(6), 661–678.

Little, R. J. A. (1988). Small sample inference about means from bivariate normal data with missing values. *Computational Statistics and Data Analysis* **7**, 161–178.

Little, R.J.A. and D.B. Rubin (2002). *Statistical Analysis with Missing Data.* Vol. 2. John Willy and Sons.

Liu, C. and D. B. Rubin (1994). The ecme algorithm: A simple extension of em and ecm with faster monotone convergence. *Biometrika* **81**(4), 633–648.

Liu, S. J. and R. Chen (1998). Sequential monte carlo methods for dynamic systems. *Journal of American Statistical Association* **93**(443), 1032–1044.

Luyben, W. L. (1990). *Process modeling, simulation and control for chemical engineers.* McGraw-Hill. New York.

Marjanovic, O., B. Lennox, D. Sandoz, K. Smith and M. Crofts (2006). Real-time monitoring of an industrial batch process. *Computers and Chemical Engineering* **30**, 1476–1481.

Matthew, J. W., A. Liakopoulos, B. Dragana and C. Georgakis (1998). A practical assessment of process data compression techniques. *Industrial Engineering Chemistry Research* **37**, 267–274.

McKee, R. C. (1971). Forming and recycling of paper making fibers. *Proceedings of the 15th Annual Pulp and Paper Conference* pp. 185–226.

Meng, X. L. and D. A. van Dyk (1997). The em algorithm - an old folk song sung to a fast new tune(with discussion). *Journal of Royal Statistical Society B* **59**, 511–567.

Meng, X. L. and D. B. Rubin (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika* **80**, 267–278.

Misra, M., S. J. Qin, S. Kumar and D. Seemann (2000). On-line data compression and error analysis using wavelet technology. *AIChE Journal* **46**(1), 119–132.

Murray, I. (2004). Note on rejection sampling and exact sampling with the metropolised independence sampler. *http://www.gatsby.ucl.ac.uk.*

Myers, C., L. R. Rabiner and A. E. Rosenberg (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acuostics, Speech and Signal Processing* **ASSP-28(6)**(6), 623.

Narasimhan, S. and S.L. Shah (2004). Model identification and error covariance matrix estimation from noisy data using PCA. ADCHEM.

Nelson, P.R.C., Taylor P.A. and J.F. MacGregor (1996). Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems* **35**, 45–65.

Nesic, Z., G. Davis and G. Dumont (1997). Paper machine compression using wavelets. *Tappi Journal* **80**, 191–203.

Nielsen, S. F. (2003). Proper and improper multiple imputation. *International Statistical Review* **71**(3), 593–627.

Nomikos, P. and J. F. MacGregor (1994). Monitoring batch processes using multiway principal component analysis. *AIChE Journal* **40**(8), 1361–1375.

Nomikos, P. and J. F. MacGregor (1995*a*). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems* **30**, 97–108.

Nomikos, P. and J. F. MacGregor (1995*b*). Multivariate spc charts for monitoring batch process. *Technometrics* **37**(1), 41–59.

Nounou, M. N., Bakshi B. R. Goel P. K. and X. Shen (2002). Bayesian principal component analysis. *Journal of Chemometrics* **16**, 576–595.

O'Shaughnessy, D. (1986). Speaker recognition. *IEEE ASSP Mag.*

Oxby, P. and S. L. Shah (2000). A critique of the use of pca for fault detection and diagnosis. *Techical Report, University of Alberta, Department of Chem. and Mat. Engineering.*

Polikar, R. (2005). The wavelet tutorial. Technical report. http://www.public.iastate.edu/ rpolikar/WAVELETS/WTutorial.html/.

Prasad, V. (2001). *Model based identification, estimation and control of chemical processes.* Rensselaer Polytechnic Institute. PhD Dissertation.

Qin, S. J. (2003). Statistical process monitoring: basics and beyond. *JOURNAL OF CHEMOMETRICS* **17**(8-9), 480–502.

Raich, A. and A. Cinar (1996). Statistical process monitoring and disturbance diagnosis in multivariate continuous processes. *AIChE J.* **42**, 995–1009.

Robinson, J. V. (1980). *Pulp and Paper Chemistry and Chemical Technology.* Vol. 2. 3 ed.. John Willy and Sons, Inc.

Rubin, D. B. (1973*a*). Matching to remove biases in observation studies. *Biometrics* **29**, 159–183.

Rubin, D. B. (1973*b*). The use of matched sampling and regression adjustment to remove bias in observational studies. *Biometrics* **29**, 185–2033.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63**, 581–592.

Rubin, D. B. (1977). Formalizing subjective notions about the effect of nonrespondents in sample surveys. *Journal of American Statistical Association* **72**, 538–543.

Rubin, D. B. (1978a). Bayesian inference for causal effects: the role of randomization. *Annals of Statistics* **7**, 34–58.

Rubin, D. B. (1987). *Multiple Imputations for nonresponse in surveys.* John Wiley and Sons Inc.

Rubin, D. B. (2003). Discussion on multiple imputation. *International Statistical Review* **71**(3), 619–625.

Rubin, D. B. and D. T. Thayer (1982). Em algorithms for ml factor analysis. *Psychometrika* **47**(1), 69–76.

Schafer, J. L. (1983). *Incomplete Data in Sample Surveys.* Vol. 2. Academic Press.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data.* Vol. 2. CRC Press.

Schafer, J. L. and J. W. Graham (2002). Missing data: Our view of the state of the art. *Psychological Methods* **7**(2), 147–177.

Schafer, J. L. and N. Schenker (2000). Inference with imputed conditional mean. *Journal of the American Statistical Association* **95**, 144–154.

Schafer, J. L. and N. Schenker (2002). Missing data: Our view of the state of the art. *Psychological Methods* **7**(2), 147–177.

Shumway, R. H. and D. S. Stoffer (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis* **3**, 253264.

Silverman, H. F. and D. P. Morgan (1990). The application of dynamic programming to connected speech recognition. *IEEE ASSP Mag.*

Singhal, A. and D. E. Seborg (2005). Effect of data compression on pattern matching in historical data. *Industrial Engineering Chemistry Research* pp. 267–274.

Skoglund, A., A. Brundinb and C. Mandenius (2004). Monitoring a paperboard machine using multivariate statistical process control. *Chemometrics and Intelligent Laboratory Systems* **73**, 3 6.

Smith, A. F. M. and A. E. Gelfand (1992). Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician* **46**(2), 84–88.

Swendsen, R. H. and J. S. Wang (1987). Nonuniversal critical dynamics in monte carlo simulation. *Physics Review Letters* **58**, 86–88.

Tanner, M. A.Wei and W. H. Wong (1990). The calculation of posterior distribution by data augmentation. *Journal of the Ameriacan Statistical Association* **82**(398), 528–540.

Teppola, P., S. P. Mujunen, P. Minkkinen, T. Pujjola and P. Pursiheimo (1998). Principal component analysis, contribution plots and feature weights in the monitoring of sequential process data from a paper machine's wet end. *Chemometrics and Intelligent Laboratory Systems* **44**, 307–317.

Thornhill, N. F., M. A. A. S. Choudhury and S. L. Shah (2004). The impact of compression on data-driven process analysis. *Journal of Process Control* **14**, 389–398.

Tipping, M. E. and C. M. Bishop (1999). Probabilistic principal component analysis. *Journal of Royal Statistical Society B* **61**(3), 611–622.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R.B. Altman (2001). Missing value estimation methods for DNA microarrays. *Bioinfomatics* **17**(6), 520–525.

van der Merwe, R. (2004). *Sigma Point Kalman Filters for Probabilistic Inference in Dynamic State Space Models.* Oregon Health and Science University. PhD Dissertation.

VanDyk, D. A. and Xiao-Li Meng (2001). The art of data augmentaion. *Journal of Computational and Graphical Statistics* **10**(1), 1–50.

Walczak, B. and D.L. Massart (2001). Dealing with missing data. *Chemometrics and Intelligent Laboratory Systems* **58**, 15–27.

Wansbeek, T. and E. Meijer (2000). *Measurement Error and Latent Variables in Econometrics.*

Wentzell, P.D., E.T. Andrews, A.D. Hamilton, K. Faber and B.R. Kowalski (1997). Maximum likelihood principal component analysis. *Journal of Chemometrics* **11**, 339–366.

Westerhuis, J. A., T. Kourti and J. F. MacGregor (1999). Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics* **13**, 397–413.

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

Wold, S. N. and K. Tjessem (1996). Hierarchical multi-block pls and pc models, for easier interpretation as an alternative to variable selection. *Journal of Chemometrics* **10**, 463–482.

Zaritskii, V. S., V. B. Svetnik and L. I. Shimelevich (1975). Monte carlo technique in problems of optimal data processing. *Automation and Remote Control* **12**, 95–103.

Zhang, P. (2003). Multiple imputation: Theory and method. *International Statistical Review* **71**(3), 581–592.

Appendix A

Derivation of the Distribution of a Covariance Matrix

The conditional posterior density of the covariance matrix for multivariate normal data is derived below. The complete data $Y = (Y_{obs}, Y_{mis})$ where Y_{obs} and Y_{mis} are the observed and missing portion of the matrix respectively. The i-th row of the data matrix is $y_i = (y_{i1}, y_{i2}, \ldots, y_{in})$. We assume that $Y = (Y_1, Y_2, \ldots, Y_n)$ has a multivariate normal distribution with mean vector μ and covariance matrix Σ. There is no prior restrictions on Σ other than positive definiteness. Since Y is normally distributed the complete data likelihood is:

$$L(\theta|Y) \propto |\Sigma|^{-\frac{N}{2}} \exp\{-\frac{1}{2}\sum_{i=1}^{N}(y_i - \mu)\Sigma^{-1}(y_i - \mu)^T\} \tag{A.1}$$

$$L(\theta|Y) \propto |\Sigma|^{-\frac{N}{2}} \exp\{-\frac{1}{2}\sum_{i=1}^{N}(y_i - \overline{y} + \overline{y} + \mu)\Sigma^{-1}(y_i - \overline{y} + \overline{y} + \mu)^T\} \tag{A.2}$$

Following algebraic manipulation, the complete-data likelihood can be written as,

$$L(\theta|Y) \propto |\Sigma|^{-\frac{N}{2}} \exp\{-\frac{N}{2} tr\Sigma^{-1}S\} \times \exp\{-\frac{\tau}{2}(\overline{y} - \mu)\Sigma^{-1}(\overline{y} - \mu)^T\} \tag{A.3}$$

Now let us apply the following prior distribution, given Σ, μ is assumed to be conditionally multivariate normal

$$\mu|\Sigma \sim N(\mu_0, \tau^{-1}\Sigma) \tag{A.4}$$

where the hyperparameters $\mu_0 \epsilon R^n$ and τ are fixed and known. Moreover Σ is inverted -Wishart,

$$\Sigma \sim W^{-1}(N, \Lambda) \tag{A.5}$$

for fixed hyperparameters $N \geq n$ and $\Lambda > 0$. Then the prior density of (μ, Σ) is:

$$\pi(\mu,\Sigma) \propto |\Sigma|^{-\frac{(N+n+2)}{2}} exp\{-\frac{1}{2} tr\Lambda^{-1}\Sigma^{-1}\} \times exp\{-\frac{\tau}{2}(\mu-\mu_0)\Sigma^{-1}(\mu-\mu_0)^T\} \quad (A.6)$$

Multiplying equation(A.3) and (A.6), it follows that the posterior distribution of (μ, Σ) has the same form as the prior distribution but with different hyperparameters. The complete-data posterior is inverted-Wishart,

$$\mu|\Sigma, Y \sim N\left(\mu_0'\right),$$

$$\Sigma|Y \sim W^{-1}\left(N', \Lambda'\right),$$

where the updated hyperparameters are

$$\tau' = \tau + n$$
$$N' = N + n$$
$$\mu_0' = \left(\frac{N}{N+\tau}\right)\overline{y} + \left(\frac{\tau}{\tau+N}\right)\mu_0$$

and

$$\Lambda' = [\Lambda^{-1} + NS + \left(\frac{\tau N}{\tau+N}\right)(\overline{y}-\mu_0)^T(\overline{y}-\mu_0)]^{-1}$$

In the absence of any prior information on the distribution of the covariance matrix $|\Lambda| \to \infty$. If we also assume that mean is exactly known then, $\overline{y} = \mu_0$. Under these conditions, the posterior distribution of the covariance matrix is:

$$P(\Sigma|Y_{mis}, Y_{obs}) \sim W^{-1}\left(N+n, (NS)^{-1}\right)$$