Process fault detection and diagnosis of fed-batch plant using multiway principal component analysis

Rahul Dey

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

Department of Electrical Engineering

June 2016

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Rahul Dey.

(Signature)

Rahul Dey.

(Rahul Dey)

EE14MTECH11033

(Roll No.)

Approval Sheet

This Thesis entitled Process fault detection and diagnosis of fed-batch plant using multiway principal component analysis by Rahul Dey is approved for the degree of Master of Technology from IIT Hyderabad

(Dr. Phanindra Jampana) Examiner
Dept. of Chemical Eng
IITH

(Dr. Abhinav Kumar) Examiner
Dept. of Electrical Eng
IITH

(Dr. Ketan P. Detroja) Adviser
Dept. of Electrical Eng
IITH

(Dr. K. Sri Rama Murty) Chairman
Dept. of Electrical Eng
IITH

Abstract

With the advent of new technologies, process plants whether it be continuous or batch process plants are getting complex. And modelling them mathematical is a herculean task. Model based fault detection and diagnosis mainly depend on explicit mathematical model of process plant, which is the biggest problem with the model based approach. Whereas with process history based there is no need of explicit model of the plant. It only depends on the data of previous runs.

With the advancement in electronic instrumentation, we can get large amount of data electronically. But the crude data we get is not useful for taking any decision. So we need develop techniques which can convey us the information about the ongoing process. So we take the help of multivariate statistics such as Principal Component Analysis(PCA) or Partial Least Squares(PLS). These methods exploits the facts such as the process data are highly correlated and have large dimensions, due to which we can compress them to lower dimension space. By examining the data in the lower dimensional space we can monitor the plant and can detect fault.

Multivariable statistical methods can also be used for monitoring batch and semi-batch processes. The only information required for monitoring is the historical database of the past batches that were considered normal. In the first part of thesis we analyse the operational and quality control problems of the past batches, with the help of which we can segregate the faulty batches from the normal ones. The control limits for charting statistics are calculated with the help of historical reference distribution of the past successful batches. But the method in the first part is not fully accomplished in online monitoring of batch. Plus it suffers from drawback such as how to calculate the control limits when the past batches are of unequal lengths. So, in the second part of thesis we discuss a method which is capable of dealing with the drawbacks that have been suggested in the first part.

The methods have been discussed with simulation case study of batch culture of *Clostridium acetobutylicum* more commonly know as the acetone-butanol-ethanol(A-B-E) fermentation. 500 batches have been generated out of which 100 are normal operating batches and the remaining 400 batches are different types of faulty condition that can occur in A-B-E fermentations.

Contents

Chapter 1

Introduction to fault detection and diagnosis

Fault Detection and Diagnosis(FDD) is a key problem in any process plant, which requires lot of attention and is the central component of abnormal event management(AEM). The key task of AEM is to detect a fault timely, diagnose the cause of fault and the corrective action taken to bring the process back to normal operating region. Although after having a sophisticated AEM unit the vital task of managing a process plant is done by human operators. However complete dependence on human operators to deal with abnormal events and emergencies is difficult due to various reasons such as, wide scope of diagnostic activity which includes a variety of faults, large size and complexity of process plants and sometimes the variable measurements we get are inadequate, insufficient due to various problems. Due to the following reasons, sometimes human operators take wrong corrective action or indecisive in taking a particular control action due to which the condition of plant further degrades causing hazardous accidents. Example of such accidents are Chernobyl Nuclear Power Plant in USSR, Union Carbide's, Bhopal in India. So, due to all these factors automation of process FDD was required. In a wide sense we can classify fault detection methods into three groups, they are quantitative model-based methods, qualitative model-based methods and process history based methods[1].

1.1 Introduction to Fault Diagnostic System

In this section we define some of the basic terminologies and characteristics which are common in the area of FDD.

1.1.1 Definition of fault

The term *fault* is generally defined as a departure from an acceptable range of an observed variable or a calculated parameter associated with a process[1][2]. In simpler words, fault is an undesirable deviation of any of the process parameter or property of the system from the nominal condition.

The main cause(s) which cause deviation of the process parameter from the nominal trajectory is(are) called *basic event(s)* or the *root cause(s)*.Commonly basic event is also termed as *malfunction*

or a *failure* .

1.1.2 Classification of general diagnostic structure

Figure 1.1 represents the very basic structure of a general fault diagnostic structure. The figure represents the simple structure of any controlled process plant which comprises of the dynamic plant which is the heart of any process plant, the feedback controller which decides our process variable whether it has deviated from the set point, and the actuator which takes corrective action in bringing the process value back to the required value. The figure describes about the three main classification of failures or malfunction that can occur in a process plant.

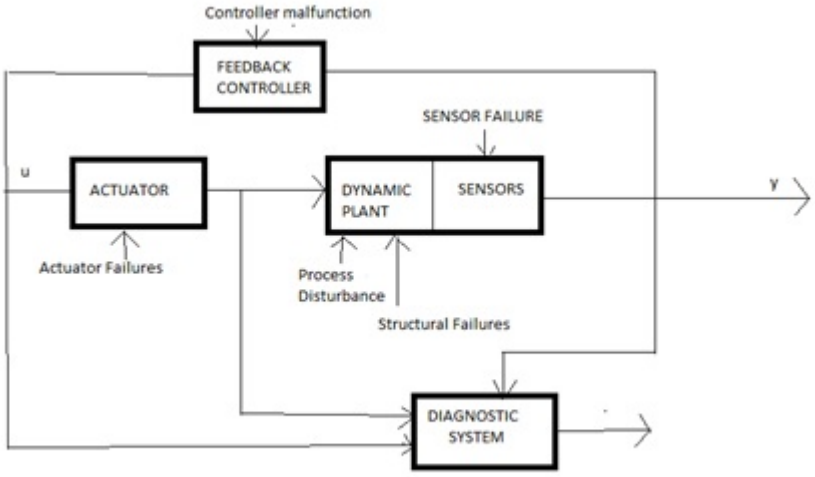

Figure 1.1: A common diagnostic structure

- ***Gross parameter changes in a model***
  In modeling any process plant, there are some processes that occur, which are not covered by the modeling equations, such processes occur below the selected level of detail of the model. These type of processes which are not modelled are lumped to a parameter. These lumped parameters also include relationship of the system boundary and environmental conditions. Failure or malfunction arises when there is a change in one or more independent variable due to environmental changes. An example of such a failure is density of crude oil entering the distillation column in petrochemical plants.
- ***Structural changes***
  Due to structural changes the process changes itself. The main cause of these changes are hard failure in components. Due to structural changes the relationships between various variables gets disturbed. An example of such failure is sensor malfunction or a stuck valve.
- ***Malfunction sensors and actuators***
  Serious errors generally occur with sensors and actuators. Some of the causes for these type of faults are, fixed failure, a constant bias which can be positive or negative,an out-of range failure or due to improper or loose wiring/contact. Some of these sensors provide feedback signals which are important in controlling the overall plant. A failure in these instruments

could cause serious problems in normal operation of the process plant.

Unstructured uncertainties, *process noise* and *measurement noise* are some of the factors that are outside the scope of FDD. By unstructured uncertainties we mean the modeling mismatch or the difference in dynamics between finite dimensional plant model and the unknown, possibly infinite dimensional actual process [3]. Process noise refers to the difference between actual value of the process variables and the values that are predicted from the model equation. Measurement noise also known as observational noise is the high frequency additive components in sensor measurements.

1.1.3 Important characteristics of a fault diagnostic system

In order to differentiate between different diagnostic methods, it is important to determine a set of desirable characteristics that a diagnostic system should have, with the help of which we can select the best approach for FDD for a particular process plant.

- ***Quick detection and diagnosis***
  The FDD system should be quick in detecting faults and to diagnose the cause of fault. However there is a trade-off between quick response to failure diagnosis and tolerable performance to disturbances and noises during normal operation. A system that can detect fault quickly will be responsive to high frequency changes such as noise. So this will lead to false alarm during normal operation.
- ***Isolability***
  By isolability we mean that the diagnostic system is able to differentiate between different type of faults. Under ideal condition i.e noise free and without any modeling uncertainties the diagnostic classifier should indicate zero outputs to the faults that have not occurred. But in this case also there is a trade-off between isolability and modeling uncertainties. Because as we go on designing classifier with high degree of isolability would perform poorly in rejecting modeling uncertainties.
- ***Robustness***
  This is an important requirement that every diagnostic classifier should posses, that the classifier should be robust to various kind of noises and uncertainties. In case of noise or any uncertainty we would want the performance of classifier to degrade gracefully rather than failing totally and abruptly.
- ***Novelty identifiability***
  The very basic requirement of every FDD classifier is to able to discriminate, whether the current ongoing process is working in normal condition or in abnormal condition. If the process is in abnormal condition, then whether the abnormality is caused due to any known fault or by an unusual new fault. This is known as novelty identifiability. But the main difficulty in designing such classifier which can differentiate novel fault is that, we have process history data of the normal region but inadequate data for modeling the abnormal region, thus it is possible that abnormal region is not properly modeled. So we would want our FDD classifier to recognize a novel fault and not group them with other faults or as normal operation.
- ***Classification error estimate***
  Another important trait which a diagnostic classifier should posses is building the user's confidence on its reliability. This can be achieved if the diagnostic classifier could provide a priori estimate of classification error that can occur.

- ***Adaptability***
  Any process in general differs and evolve with time due to the fact that various changes in external input or due to environment or due to structural changes by changing the components of the process plant. Thus the diagnostic system should adaptable to different changes. The diagnostic classifier should be able to evolve with the system as new cases and problem emerges, as more data is available.
- ***Explanation facility***
  The main task of the FDD classifier is to identify the source of fault but also be able to provide satisfactory explanation on how the fault originated and propagated to the present situation. It is an important part in on-line diagnosis. This requires an extensive ability to reason about cause and effect relationship in a process. The FDD classifier should be able to explain its suggestions so that the operator can take proper actions with the help of past experiences.
- ***Modeling requirements***
  The total amount of modeling required for developing a FDD classifier plays an important role in choosing between different approaches. Most of the time modeling efforts should be minimal as possible for fast and easy deployment of diagnostic classifier.
- ***Storage and computational requirements***
  In most of the FDD classifier they provide quick real-time solutions which requires algorithms and implementation which are computationally less complex, but might require large storage. So there should be proper balance between storage and computational requirements for an ideal FDD classifier.
- ***Multiple fault identifiability***
  The skill to detect multiple faults is an important but a tough requirement. It is difficult problem due to the fact that most of the faults are interacting in nature. Most of the system, are nonlinear in nature due to which the interactions is combined in nature due to which the FDD classifier may not be able to identify a fault by using separate fault patterns, and if we try to design fault patterns for each and every combined fault, it is a very extensive task, which is not possible for large process.

1.2 Classification of diagnostic algorithm

The two main components in any diagnostic classifier are:

i. the type of knowledge, i.e model based or process data based
ii. the type of diagnostic search strategy.

Diagnostic search strategy basically it depends strongly on the type of knowledge presentation, which is largely dependent on a priori knowledge available. Therefore the most important part for constructing a diagnostic classifier is the a priori knowledge that is available. Hence, a priori knowledge used is the main differentiating factor in diagnostic systems. The basic a priori knowledge that is required for fault detection and diagnosis is the set of failures and relationship between failures and the observations. The a priori information can be constructed from basic understanding of the process using first-principles knowledge. This type of knowledge is referred to as deep, causal or model-based knowledge[1][4]. Another way of obtaining this knowledge is from the past experience with the process. This knowledge is termed as shallow, compiled, evidential or process history-based

knowledge.

The model based method can be classified under two section, namely qualitative model based approach and quantitative model based approach. The model is usually developed by understanding the physics of the process. In quantitative model based approach we get mathematical relationships between inputs and outputs of the system with the help of which we construct the model. In qualitative model equations we get these relationships in terms of qualitative functions from different units in a process.

In process history based methods unlike model based approach we do not have any model, but availability of large amount of historical process data. With the help of different methods this data can be converted and presented as a priori to the diagnostic system. The transformation and representation of data to serve as a priori knowledge is known as feature extraction.
Different types of diagnostic algorithm are shown in figure1.2. From the above discussion it is clear

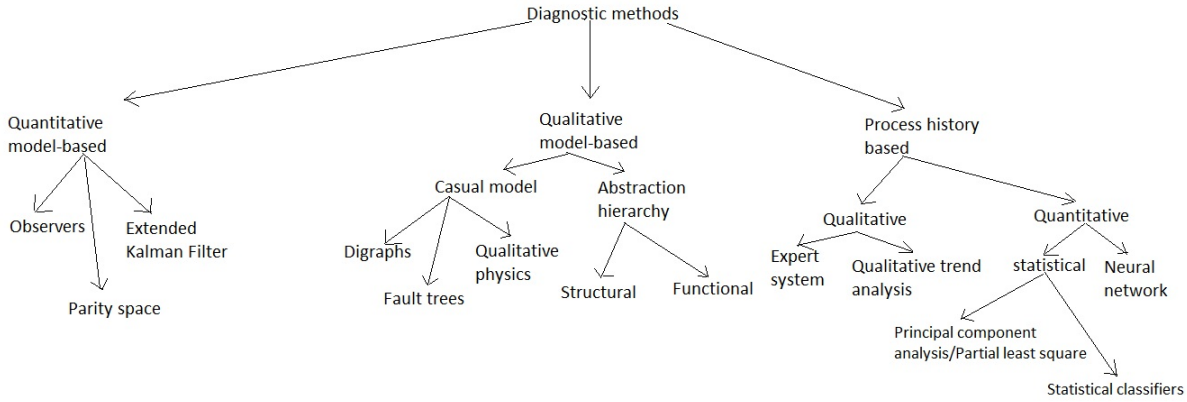

Figure 1.2: Different types of diagnostic algorithms.

that all model based method, qualitative or quantitative need data for approximating some of the parameters in the model and all the methods based on historic process data need to formulate some form of model to do fault diagnosis.
Now we will discuss some of the methods in brief one by one.

1.3 Quantitative model based approaches

In this section we will discuss in brief the quantitative model-base FDD methods. The most common types of methods which falls under this sections are diagnostic observers, Kalman filter, parity relations and parameter estimation. We will focus mainly on principles behind these methods, briefly mention their application, advantages and disadvantages in process monitoring and for fault detection. Most of the quantitative model based fault classifier rely on general input-output and state-space model which is as discussed in Appendix A. The quantitative model-based approach heavily relies on the concept of *analytical redundancy*. So, in brief we will discuss the concept of analytical redundancy.

Analytical redundancy By the term analytical redundancy we mean that, checking the actual system behavior with that of the expected behavior which is obtained from the model. If there is a fault which has occurred it will generate differences between expected and actual behavior. Such difference is called as *residuals*, which are artificial signal indicating an occurrence of faults.
So, diagnostic algorithm which take advantage of analytical redundancy comprises of two steps, first is generating the residual and the second step is choosing a decision rule for diagnosis. A single residual signal is sufficient for fault detection but for fault diagnosis we require a set of residuals.
Mainly there are two types of redundancies, hardware redundancy and analytical redundancy. Hardware redundancy requires redundant sensors. It is used in places where safety is the utmost priority. Example of such system is the hydraulic system of aircraft or in nuclear plants where some part of control system is triplicated formally termed as *triple modular redundancy*. However this method is not applicable everywhere due additional cost and space.
In analytical redundancy also known as artificial, inherent or functional redundancy it is obtained from functional dependence between the process variables and is obtained from set of algebraic relationships between the states, inputs and outputs of the system. The analytical redundancy can be sub-divided in two more parts on the basis of how the redundancy is acquired, they are direct and temporal.
In direct redundancy we establish algebraic relationships among various sensor measurements and then we can compute the value of a particular sensor from measurements of other sensors. So we can compare the actual value and the calculated value and comment on whether there is a sensor fault or not based on the difference of calculated and actual value. Whereas in temporal redundancy we obtain the difference or differential relationship between different sensor outputs and actuator inputs. With the help of input and output data of the process, temporal redundancy is useful for detecting sensors and actuator fault.

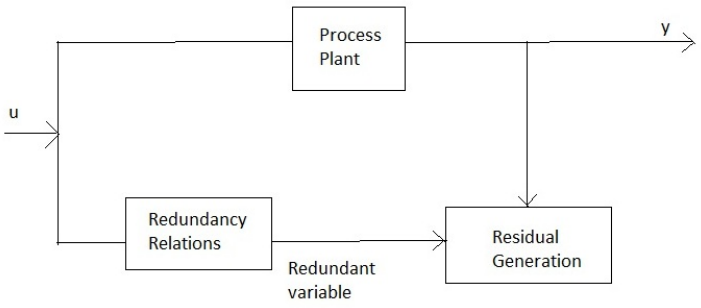

Figure 1.3: General scheme for using analytical redundancy.

A basic block diagram as in figure1.3 is shown for how analytical redundancy works in diagnostic system. The main requirement of analytical redundancy in fault diagnosis is to inspect actual system behavior with that of system model for consistency. Any inconsistencies which are expressed as the difference between actual value and measured value is used for fault detection and isolation. Under no fault condition the difference or residual should be close to zero indicating that no fault

has occurred and during faulty condition the difference should change, clearly indicating a faulty condition. In the above figure1.3 we can see that the input sequence is common to both the process plant and redundancy relation. So we can compute the expected value from redundancy relation. We can see that the output from redundancy relation are sent to residual generation where we also see that output from process that is actual value is also fed to residual generation. So the difference between actual and expected value is the residual, which is further used for fault detection and diagnosis purpose.

1.3.1 Diagnostic observer for diagnostic system

The main task in observer-based fault detection is the generation of set of residuals which can detect fault and is able to discriminate between different faults. The residuals should be robust such that the decision is not altered by unknown inputs such as unstructured uncertainties such as measurement and process noise and modeling uncertainties. The basic idea is to develop a set of observers such that each one of them is responsive to a subset of faults while unresponsive to the remaining faults and unknown inputs. During fault free condition, the observer monitors the process and the residual generated due to unknown inputs will be negligible. When a specific fault occurs the observers that is sensitive to that particular fault will generate residuals of large magnitude, whereas the observers that are insensitive to the fault will continue to show small residuals due to the unknown inputs. The set of observers are constructed in such a manner that the residuals thus generated has a individual pattern for each fault, due to which we can easily isolate the fault. The construction of observers is as follows.
Let us consider the state space model as described in AppendixA, equation A.1 with matrix $\mathbf{D} = 0$. An observer is a dynamic algorithm which estimates the states of a system with the help of the model and observed outputs and inputs. It is constructed as follows[5]

$$\begin{aligned} \hat{x}(t+1) &= \mathbf{A}\hat{x}(t) + \mathbf{B}u(t) + \mathbf{K}[y(t) - \mathbf{C}\hat{\mathbf{x}}(\mathbf{t})] \\ \hat{x}(t+1) &= (\mathbf{A} - \mathbf{KC})\hat{x}(t) + \mathbf{B}u(t) + \mathbf{K}y(t) \\ r(t) &= y(t) - \mathbf{C}\hat{x}(t) \end{aligned} \tag{1.1}$$

Here $\hat{x}(t)$ and $r(t)$ are the estimated states and the prediction error or output error respectively. Let the state estimation error denoted by $e(t)$ is given by $e(t) = x(t) - \hat{x}(t)$. Then these results can be incorporated in the state space model with fault as described in Appendix A, equationA.5 with $q(t) = 0$. Then

$$\begin{aligned} e(t+1) &= (\mathbf{A} - \mathbf{KC})e(t) + \mathbf{E}p(t) \\ r(t) &= \mathbf{C}e(t) \end{aligned} \tag{1.2}$$

From equation(1.2) we can see that $r(t)$ is a set of residuals, and it is dependent on the outputs of the system which are affected by faults. But the main point that should be taken care of is that observer in closed loop must be stable or any initial error will keep on growing with time. In designing the observer we have to choose the feedback matrix $\mathbf{K}$ properly so that the observer in closed loop is stable.

1.3.2 Parity relations

Parity equations are basically rearrangement followed with transformation of state-space and input output model. The main characteristics of the parity relations is to examine the parity(consistency) of the plant models with the output of the plant i.e sensor outputs(measurements) and known process input. Under ideal condition i.e under fault free and noise free condition, the residual or the value of the parity equation tends to zero. But in most of the process, the residuals is not zero exactly due to the measurement and process noise, gross errors in sensors and actuators, modeling mismatch, and faults in the plants. The main aim of this method is to rearrange the model relations so as to get the best fault isolation. The basic requirement here is that the residual vectors are orthogonal to each other for different faults. And redundancy gives us the flexibility in designing such equations which helps in residual generation.

The parity equations can be constructed by measurement equation[6].

$$y(t) = \mathbf{C}x(t) \tag{1.3}$$

Let y is the $n \times 1$ measurement vector; and x be $m \times 1$ vector of true values of the state variables and $\mathbf{C}$ is the $m \times n$ measurement matrix having full rank. The above equation(1.3) holds true under fault free condition. When fault occurs with any one of the measurements then equation(1.3) changes as follow

$$y(t) = \mathbf{C}x(t) + \Delta y(t) \tag{1.4}$$

We choose a projection matrix $\mathbf{V}$ having dimensions $(n-m) \times n$ such that it satisfies the following properties

$$\begin{aligned} \mathbf{VC} &= 0 \\ \mathbf{V}^T\mathbf{V} &= \mathbf{I}_n - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \end{aligned} \tag{1.5}$$

Matrix$\mathbf{V}$ is orthogonal i.e $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{n-m}$ due to the reason that $\mathbf{C}$ is null space of $\mathbf{V}$. The parity vector p can be written as

$$\begin{aligned} p(t) &= \mathbf{V}y(t) \\ p(t) &= \mathbf{VC}x(t) + \mathbf{V}\Delta y(t) \\ p(t) &= \mathbf{V}\Delta y(t) \end{aligned} \tag{1.6}$$

Therefore the residuals generated from parity equations contains the signature of each measurement fault. In case of no fault $p = 0$ as Δy is zero. For a single i^{th} sensor fault, where $i < m$.

$$\begin{aligned} \Delta y &= [0\ 0\ 0\ ...\ \Delta y_i\ ...\ 0]^T \\ \mathbf{V}\Delta y &= \Delta y_i\ .\ (the\ i^{th}\ column\ of\ \mathbf{V}) \end{aligned} \tag{1.7}$$

So, from the above equation(1.7) we can see that the columns of $\mathbf{V}$, we can determine n different combinations associated with the n sensor faults in the plants. This procedure heavily relies on direct redundancy.

The following procedure proposed in[7] provides a generalized scheme for both temporal and

direct redundancy. Let us consider the state-space model as in equation(A.1). The output at $t+1$ time is given by substitution

$$y(t+1) = \mathbf{CA}x(t) + \mathbf{CB}u(t) + \mathbf{D}u(t+1) \tag{1.8}$$

Now for any s which is $s > 0$ we can substitute recursively to get

$$Y(t+s) = \mathbf{CA}^s x(t) + \mathbf{CA}^{s-1}\mathbf{B}u(t) + ... + \mathbf{CB}u(t+s-1) + \mathbf{D}u(t+s) \tag{1.9}$$

After writing the equation for $s = 0, 1, 2, ..., n_1 \leq n$ and writing it in matrix form gives

$$\mathbf{Y}(t) = \mathbf{R}x(t-n_1) + \mathbf{QU}(t) \tag{1.10}$$

where,

$$\mathbf{Y}(t) = \begin{bmatrix} y(t-n_1) \\ y(t-n_1+1) \\ y(t-n_1+2) \\ . \\ . \\ . \\ y(t) \end{bmatrix} \quad \mathbf{U}(t) = \begin{bmatrix} u(t-n_1) \\ u(t-n_1+1) \\ u(t-n_1+2) \\ . \\ . \\ . \\ u(t) \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ . \\ . \\ . \\ \mathbf{CA}^n \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D} & \underline{\mathbf{0}} & . & . & . & \underline{\mathbf{0}} & \underline{\mathbf{0}} \\ \mathbf{CB} & \mathbf{D} & . & . & . & \underline{\mathbf{0}} & \underline{\mathbf{0}} \\ \mathbf{CAB} & \mathbf{CB} & . & . & . & \underline{\mathbf{0}} & \underline{\mathbf{0}} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \mathbf{CA}^{(n_1-1)}\mathbf{B} & \mathbf{CA}^{(n_1-2)}\mathbf{B} & . & . & . & \mathbf{CB} & \mathbf{D} \end{bmatrix}$$

For a system consisting of m inputs and k outputs, $\mathbf{Y}$ has dimensions $(n_1+1) \times m$ and $\mathbf{U}$ has dimension $(n_1+1) \times k$ and $\mathbf{Q}$ has dimension $[(n_1+1) \times m] \times [(n_1+1) \times k]$ while that of $\mathbf{R}$ is $[(n_1+1) \times m] \times n$.
Pre-multiplying a vector $\mathbf{w}^T$ to equation(1.10) having dimension $(n_1+1) \times m$ gives,

$$\mathbf{w}^T\mathbf{Y(t)} = \mathbf{w}^T\mathbf{R}x(t-n_1) + \mathbf{w}^T\mathbf{QU}(t) \tag{1.11}$$

The above equation(1.11) contains the input, output and the state variables in the mixed form. It will become a parity equation of order n_1 when it will satisfy the condition of

$$\mathbf{w}^T\mathbf{R} = 0 \tag{1.12}$$

which means that the state variables become zero. We can see that this is a case of homogeneous linear equation having m equations. for an observable system, these m equations are independent.

1.3.3 Kalman filters

One of the main problem in process plant is that the plant disturbance are random fluctuating in nature and oftentimes we only their statistical parameters. The problem of fault diagnosis in such condition can be solved by monitoring the residuals(innovations) or by monitoring prediction error. The main motto is to construct a state estimator with minimal estimation error. One such popular method is Kalman filter which is based on optimal state estimate and is designed based on system model in normal operating mode.

Kalman filter is a recursive algorithm used for estimating state. The Kalman filter implementation for state-space model is similar to an optimal predictor for a linear stochastic system constructed from input-output model. The kalman filter equations can be summarized as follows
Let us consider the equations (A.1) and (A.5) with matrix $\mathbf{D} = 0$ and introducing white noise term $w(t)$ and $v(t)$.

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{E}p(t) + w(t) \\ y(t) &= \mathbf{C}x(t) + q(t) + v(t) \end{aligned} \tag{1.13}$$

The general Kalman filter equation is designed as in[8]

$$\begin{aligned} \hat{x}(t+1|t) &= \mathbf{A}\hat{x}(t|t) + \mathbf{B}u(t) \\ \hat{x}(t|t) &= \hat{x}(t|t-1) + \mathbf{K}'r(t) \\ r(t) &= y(t) - \mathbf{C}\hat{x}(t|t-1) \end{aligned} \tag{1.14}$$

Where $\mathbf{K}'$ is known as Kalman gain. We define

$$\begin{aligned} \hat{x}(t) &= \hat{x}(t|t-1) \\ \mathbf{K} &= \mathbf{A}\mathbf{K}' \end{aligned} \tag{1.15}$$

Substituting equation(1.15) into equation(1.14), we get

$$\begin{aligned} \hat{x}(t+1) &= \mathbf{A}\hat{x}(t) + \mathbf{B}u(t) + \mathbf{K}r(t) \\ r(t) &= y(t) - \mathbf{C}\hat{x}(t) \end{aligned} \tag{1.16}$$

The recursive error equation can be written as follows

$$\begin{aligned} e(t+1) &= (\mathbf{A} - \mathbf{K}\mathbf{C})e(t) + \mathbf{F}p(t) + w(t) - \mathbf{K}q(t) - \mathbf{K}v(t) \\ r(t) &= \mathbf{C}e(t) + q(t) + v(t) \end{aligned} \tag{1.17}$$

where $e(t) = x(t) - \hat{x}(t)$ is the state estimation error. The Kalman gain $\mathbf{K}'$ in such a way that state estimates have optimum properties under the influence of random noise. It has been seen that a bank of Kalman filters constructed from all the available system models under all possible change can also be used for fault isolation. The main advantage of using Kalman filter for fault detection is that, its residual are uncorrelated in time due to which fault isolation is relatively easy.

1.3.4 Parameter estimation

Diagnosis of parameter drifts which cannot be measured directly requires some form of online parameter estimation methods for which exact parametric models of the process are needed, mostly in continuous domain in the form of partial or ordinary differential equations. The process model as in the equation(A.1) and equation(A.2) assumes that the process parameter are either constant or depends only on state variable. Basically faults which arises as time dependent parameter drifts can be managed with parameter estimation methods. The method is as follows: obtaining the process model with only the measurable inputs and outputs having the form

$$y(t) = f(u(t), \theta) \tag{1.18}$$

where,θ is the model parameter which are estimated when $y(t)$ and $u(t)$ are available.θ is related to the the physical parameters φ in the process by relation

$$\theta = g(\varphi) \tag{1.19}$$

So any change in the physical parameter $\Delta\varphi$ are computed from the above equation(1.19). With the help of methods from pattern recognition we can connect the changes in $\Delta\varphi$ to process fault. But the problem with these methods is that they require accurate dynamic models of the process plant and are computationally very difficult for a large process. Another main drawback is the complexity involved with parameter estimation in the field of fault diagnosis is that if the process model is nonlinear first principle model, then parameter estimation turns out to be problem of nonlinear optimization, which are computationally expensive to solve

1.3.5 Hardware redundancy and voting schemes

Voting schemes are used in a system where there is high degree of parallel hardware redundancy. This type of hardware redundancy is mainly implemented in safety critical plant such as aircraft, nuclear power plants. Suppose we have three identical sensors measuring the identical variable. If one of the three sensor shows a different value than other two, then the differing sensor is identified as faulty. Voting schemes are easy to apply and are swift in detecting mechanical faults in an instrument. Another approach is to search for subsets of sensor measurements with different degree of consistency, given the error bound for different sensors. The most consistent subset is used for predicting the measured quantity and the least consistent subset can be used for pin pointing the faulty sensors. The main advantage of hardware redundancy is that the faulty sensors can be removed easily from the system reducing the number of false alarms.

1.3.6 Enhanced residuals

In all the above subsections we have generated diagnostic residuals to identify there is a potential fault in the system. Our next task is to detect and isolate the fault in the system. For the isolation of fault, it is required to generate such residuals that are not only sensitive to faults but also specific to faults. So the residual generator should produce a set of residual rather than a single residual so that the residuals responds selectively to different faults. So the residuals in these cases are not only

capable of fault detection but are also capable of fault isolation. Mainly there are two methods of generating such kind residuals, structured residual generators and directional residual generators.

Directional residuals In this method, residual vectors are generated which are restricted to fault specific direction due to which we can isolate faults in the multidimensional residual space. The construction of directional residual generator is build on linear time-invariant finite dimensional systems. Let the system model be

$$h(z^{-1})y(t) = \mathbf{U}(z^{-1})u(t) + \mathbf{V}(z^{-1})p(t) + \mathbf{W}(z^{-1})\omega(t) \ \ ; t \geq 1 \tag{1.20}$$

where $\omega(t)$ and $p(t)$ represent noise and faults, respectively.
The residual generator is defined as linear dynamic operator operating on observable input $u(t)$ and output $y(t)$ and having structure as follows

$$r(t) = \mathbf{G}(z^{-1})y(t) + \mathbf{H}(z^{-1})u(t) \tag{1.21}$$

the above equation(1.21) is known as the *computational form* of the residual generator. It is designed such that $\mathbf{G}(z^{-1})$ and $\mathbf{H}(z^{-1})$ are polynomials. Constructing such a residual generator is computationally easy and it yields bounded residuals the constraint being $u(t)$ and $y(t)$ are bounded. The system input $u(t)$ should not be affecting residuals. So $\mathbf{H}(z^{-1})$ can be defined as

$$\mathbf{H}(z^{-1}) = -h^{-1}(z^{-1})\mathbf{G}(z^{-1})\mathbf{U}(z^{-1}) \tag{1.22}$$

Now combining equations(1.20),(1.21) and (1.22), we get

$$r(t) = \mathbf{F}(z^{-1})p(t) + \mathbf{L}(z^{-1})\omega(t) \tag{1.23}$$

The above equation(1.23) is known as the *internal form* of the residual generator, which describes the source of residual.
The effect on directional residual vector $r(t)$ due to the combined effects of all faults and noise is

$$r(t) = \mathbf{\Psi\Delta}(z^{-1})p(t) + \mathbf{\Pi M}(z^{-1})\omega(t) \tag{1.24}$$

where, $\mathbf{\Delta}(z^{-1})$ and $\mathbf{M}(z^{-1})$ describes the dynamics of the faults and noises, respectively. And the matrices $\mathbf{\Psi}$ and $\mathbf{\Pi}$ tells us about the direction faults and noises.

Structured residuals In this approach, the residual vector so generated, is like each residual elements, responds specifically to a subset of faults. It is required that the corresponding residual component of the specific fault is nonzero. Alternatively, the residuals corresponding to the particular fault will be restricted to the subspace of the entire residual space. Due to this, we can form binary fault signatures, or the residual structure for fault isolation as each of the residual set is orthogonal to corresponding different faults.

The structured residual is generated by structured parity equation in either Moving Average(MA) or Auto-Regressive Moving Average(ARMA) format or in state-space format. They can also be constructed by using direct eigenstructure(the set of eigenvalues of matrix) assignment of diagnostic

observer. The construction is briefly described as follow
For any linear system, the observed output $y(t)$ and input $u(t)$ are related to their true values $y^0(t)$ and $u^0(t)$ as

$$\begin{aligned} y(t) &= y^0(t) + q(t) \\ u(t) &= u^0(t) + p(t) \end{aligned} \tag{1.25}$$

where, $q(t)$ and $p(t)$ are sensor and actuator faults respectively. A residual can be defined as follow

$$o(t) = \mathbf{H}(z)y(t) - \mathbf{G}(z)u(t) \tag{1.26}$$

where, $\mathbf{G}(z)$ and $\mathbf{H}(z)$ have structure as in Appendix(A). From equation(1.25) and (1.26) we get

$$o(t) = \mathbf{H}(z)(y^0(t) + q(t)) - \mathbf{G}(z)(u^0(t) + p(t)) \tag{1.27}$$

And from equation(A.2), we can rewrite the equation(1.27) as

$$o(t) = \mathbf{H}(z)q(t) - \mathbf{G}(z)p(t) \tag{1.28}$$

By applying further transformation to the residual $r(t)$ we get,

$$r(t) = \mathbf{W}(z)o(t) \tag{1.29}$$

By properly choosing $\mathbf{W}(z)$ will affect certain properties of $r(t)$.
The residual structure are distinctive by incidence matrices, whose rows and columns are residuals and fault code respectively. For example there are three faults in a system denoted by $\mathbf{F}$, where $\mathbf{F} = [F_1\ F_2\ F_3]'$, the possible structure of incidence matrix is,

$$\begin{pmatrix} & F_1 & F_2 & F_3 \\ r_1 & 1 & 1 & 0 \\ r_2 & 0 & 1 & 1 \\ r_3 & 1 & 0 & 1 \end{pmatrix}$$

Here 1 denotes particular residual is affected by that fault and 0 denotes that it does not. So if there is a change in r_1 and r_2 we can say that fault F_1 has occurred. So we can say that columns of incidence matrix are the identification of particular fault. A fault cannot be detected if its corresponding column contains all element as 0, which implies that none of the residual is affected by that particular fault. Also two faults are indistinguishable by structure if they have same corresponding columns.

So with that we have covered some of the basic method of fault detection using quantitative model based approach. One of the major advantages of quantitative model based approach is that we have some control over the nature of the residuals. Another main problem is the simple assumption of the noise, which includes modeling mismatch. In general, we assume the disturbance matrix includes only additive noise, but in practicality severe modeling uncertainties occurs due to parameter drifts which affects in the form of multiplicative uncertainties. Moreover factors like

complex system, nonlinear process, lack of process data, high dimensionality creates hindrance to develop exact mathematical model of the process plant. Most of the quantitative model based approach for fault diagnosis is used extensively in the field of electrical, mechanical and aerospace engineering. Quantitative model based method for fault diagnosis is not much used in the field of chemical engineering.

1.4 Qualitative model based approaches

In this section, we will review qualitative model based methods and the search strategies used for fault diagnosis. Qualitative model unlike quantitative models are developed based on the fundamental understanding of the underlying physics and chemistry of the process. The qualitative models can be constructed by two methods as qualitative casual models or abstraction hierarchies. And for search strategies there are two ways for fault, topographic search and symptomatic search. In topographic search we search for fault using a template from a normal operating region, whereas, in symptomatic search strategies we look for symptoms to localize the fault.

1.4.1 Qualitative models

Qualitative model was first developed as knowledge based expert system. Basically an expert system is a computer program that imitates the human like problem solving approach in a restricted domain. Expert system consists of knowledge base which is essentially a set of if-then-else rule and an inference engine(process of deduction) which searches through the knowledge base to reach to a particular conclusion from given initial facts. But the main problem with this approach is that, the condition of if-else-then grows exponentially with the behavioral complexity o the system. And in case of a novel problem, the system will fail as it has no underlying knowledge of the process. Therefore such knowledge representation is known as *shallow* as it does not have intense understanding of the process.

In symbolic reasoning we have three different kind of reasoning, they are *abductive*, *inductive* and *default* reasoning. By abduction we mean that, it is the hypothetical description for what has been observed. It is different from simple logical deduction in the sense that, we can get more than one solution in abduction reasoning. There is no definite rule decide between different solution, so we choose the hypotheses that is having the highest probability of occurrence. So searching for a cause of a fault or abnormality is an abductive reasoning. By abduction reasoning we can also provide explanation of how the cause is related to the abnormality.

In inductive reasoning the set of observations are classified into different categories or concept. Inductive learning is applied only where we have specialized or generalized a concept so well that we have included all the observations that belongs to the particular concept and exclude do not fall in that concept. But the problem is definition of a category or concept is not simple because of variety of inputs or uncertainty due to noises.

In default reasoning, we make assumption on the various quantities that are manipulated due to specific reason we ignore the current value for other values due to some reason. Default reasoning is also known as non-monotonic reasoning. The example of such reasoning is that as the outlet is blocked, flow will be zero hence level in the tank is constant. In traditional logic, if we deduce a fact and if it is found out to be true, then it remains true for rest of the reasoning. This is known as

monotonic reasoning. But this is not possible as new cases arises, for example in the above example of tank if there is a leak in the tank definitely the level in the tank cannot remain constant so we will have to change our conclusion that level in the tank will remain constant.
So we need reasoning tool which can qualitatively model a system, which will reproduce the causal structure of the system in more wise manner than the conventional expert system and which is flexible in nature. So in the next section we will discuss different types of qualitative models.

1.4.2 Digraphs based causal models

Digraphs based causal models are basically based on cause-effect relationships. In any first principle model based approach, we begin by describing the system along with the experience from malfunctioning process. The main reasoning here is to pinpoint the functional changes that are due to the fault in the process. So by digraph we mean that, it is a type of graph with directed arcs between two nodes. And by signed digraph(SDG) is digraph with positive or negative sign on them. The directed arc starts from tail(cause) and ends at head(effect) nodes. Each of the nodes in SDG represents deviation of a state variable from steady state. Each node(vertex) represents an event or a variable and the line joining them(edges) denotes the relationships between nodes.
SDG are more condensed in nature than decision tables, truth tables or finite state models. For better understanding digraphs lets us consider an example, considering a tank where F_1 is the flow of inlet and F_2 be the outlet flow and Z is the initial height of liquid in the tank. So the equation to represent this system are

$$F_1 - F_2 = \frac{dZ}{dt}$$
$$F_2 = \frac{Z}{R}$$

So we can interpret it as change in input flow rates F_1 will cause the level of the liquid$Z$ in the tank to change, so to maintain the level of liquid outlet flow F_2 must change accordingly. A corresponding digraph for the tank system is shown in figure(1.4)

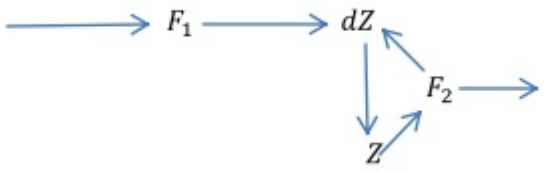

Figure 1.4: Digraph for the example

SDG is a very efficient method of depicting qualitative models graphically. There are basically three kinds of nodes in SDG for representing a chemical process

i. nodes from which has only output. They mainly represent basic or to be precise fault variables which changes independently.
ii. nodes from which has both input and output arcs, they are mainly the process variable

iii. nodes which has only input arcs, they are mainly the output variables and they do not influence any other variable.

SDGs are the most used methods which use casual knowledge for fault diagnosis.

1.4.3 Fault trees

Fault trees are mainly used in inspecting the system reliability and safety. It is a logic tree that propagates from primary events or faults to the top level event or hazard. The tree is comprised of layers of nodes. In each of the node different logical operators like OR and AND are done for propagation. Any basic fault tree analysis comprises of following four steps[9]

i. The undesired event(fault) is defined
ii. The immediate cause of the undesired event is found out.
iii. We keep on finding the cause until we find out the basic cause
iv. Using the above information we construct the fault tree or logical diagram showing the logical event relationship.

For construction of fault tree, we should have complete understanding of the process. The fault tree is established by questioning as what could have caused the top level event. As we answer, we come across other events which are connected by logic nodes. This process continues till we get to the primary event which is not expanded further. After the construction of fault tree, the next step is evaluation of fault tree. There are mainly two types of evaluation, qualitative evaluation and quantitative evaluation.

In qualitative evaluation we develop minimal cut-sets, which is defined as a collection of basic failures which are necessary and sufficient for system failure by the minimal cut-set which is constructed. By minimal cut-set we mean that it is the set of critical component failures.

In quantitative evaluation, information about the probability of primary events/faults are used to calculate the probability of top level events/faults.

The main feature of fault tree is that it combines logical analysis for computational means. In fault tree we can construct using different logical operators such as OR, AND, XOR unlike only the dominant OR logic as in digraph, due to which we can reduce erroneous solution and representing the system in compact form. Some of the advantages of using fault tree are

- The fault tree clearly describes al the different relationships that are sufficient and necessary for affecting the top event.
- In constructing a fault tree, in depth knowledge of the logic and basic causes that lead to the top event.
- We can highlight the important elements of a system which can result in failure. Due to which we can monitor the single element which can cause major faults
- Fault tree analysis also takes into account places where human error can occur.

But the main drawback with fault tree analysis is that, fault tree is constructed using the opinion of plant operators and exert judgments at various stages, so it is prone to mistakes at different steps in the process.

Fault tree can also be developed from digraphs[10][9]. Basically fault trees determine rules or casual pathways through which primary events or faults propagate through the system to cause a major malfunction. So if a top event(major malfunction) occurs, it will search for the immediate cause using the process information in the form of digraph, it will continue to search until we reach the primary

event or combination of primary events which resulted in top event. And the process information is made available in the form of unit models and the process relation gives us the relation between these different units. By unit models we mean that the ways in which different process variables are inter related during normal region and also when the process is running under faults. Fault tree analysis is mainly formulated as search in finite state space domain. If we have a given initial state, the algorithm relates it to operators(OR, AND, XOR) which transforms the initial state to the goal state. Here initial state refers to the top event, and the goal state is the algorithm connecting the initial state to all possible basic events. All the possible states are defined by digraphs. Once the fault tree is constructed, the details are stored in the form of cut sets. A cut set is defined as any set of primary events, which when they occur at the same time cause the top event. A minimal cut set is defined as the subset of cut set, which contains all the primary events that must occur together for a top event to occur. The set of minimal cut set which can cause a top event describes all the ways in which a given top event can happen.

1.4.4 Qualitative physics

Qualitative physics also known as common sense reasoning of physical system is a major research topic in artificial intelligence. There are two ways in which qualitative physics knowledge in fault diagnosis can be used

i. The first method is to obtain qualitative equation from the differential equation which is termed as confluence equation. Let us consider the tank example as in digraph section(1.4.2). The steady state confluence equation can be written as follows

$$[F_2] - [F_1] = 0$$
$$[Z] - [F_2] = 0$$

where $[F_1]$ denotes the sign of the deviating variable F_1. The above equation are solved qualitatively to get qualitative value of the variables. Suppose any of the variable has +, then it denotes that the variable can only have qualitative higher value, i.e the value of the variable will only increase. The main advantage of this approach is that, even if we do not have exact mathematical model, we can know about the qualitative behavior.

ii. The second method is called precedence ordering. In this method we order the variables according to information flow among them. The main idea is that the information flow among the model equations are not occurring at the same time, there is some kind of partial or complete precedence order among the variables. Due to the ordering we can see the information flow thus we can see the cause effect relationship among different variables.

There are also other methods in qualitative physics like derivation of qualitative behavior from ordinary differential equations. The main advantage of qualitative physics is that we can draw partial conclusions about the process from incomplete and unknown knowledge of the process. Each of the above methods described above starts with the explanation of the physical process, then construction of model, then we use an algorithm to obtain the pattern and performance of the system without exact value of system parameters and functional relationship.

1.4.5 Abstraction hierarchy of process knowledge

By abstraction hierarchy we mean that we deal with the ideas based on decomposition rather than mathematical knowledge. The main concept here is to be able to draw conclusion about the whole system from the behavior of its subsystem. Such decomposition is solely based on no-function-in-structure, which states that the rules that are specified and applied to a subsystem can in no way be applicable to the overall system function.
Another important principle which is used in abstraction hierarchy is the principle of locality which means that, laws which are defined for a part specifically is not applicable to any other part.
So, with no-function-in-structure it allows consistent and accurate behavior among different units and principle of locality allows us to predict the behavior based on local information. The two most accepted decomposition technique of process system are

i. *structural decomposition* which states that how a unit in a process are connected to other units.
ii. *functional decomposition* which relates the output of a unit to its corresponding inputs or state information.

In abstraction hierarchy the process plant are break down or decomposed into different corresponding process units. Due to the decomposition, we can represent the system in terms of its input-output function of different units. Decomposition of the system to its constituent subsystem can be done on different level of abstraction as required. Majority of fault diagnostic classifier in chemical process use functional decomposition. It is widely used in chemical process due to the fact that chemical process are complex, which cannot be explained in terms of structure.
Fault diagnosis can be regarded as top-down search from higher level abstraction which includes group of equipments and functional system to lower level abstraction which includes individual units and unit functions. Based on this property we can decompose the process into a hierarchy of functional subsystem. The nodes in the hierarchy represents the intended function of the subsystem. Then by comparing the intended function to the function of subsystem, then we can make a conclusion about whether the system is in faulty or normal operating region.
According to [11] the process plant is a set of interacting subsystem, where each subsystem is classified as a control system(closed loops) or passive system(open loops) or an external system. Each of these subsystem has a corresponding function at the level of system description. Depending on their functional properties they are further class as

i. stressed, functional, uncontrolled or saturated in case of closed loop.
ii. malfunctional or functional in case of open loop and external system.

The lower level subsystem are basically sensors, controller, actuators, process units and control elements. The main idea in [11] is that the fault in higher level subsystem arises due to to the failure in one or more of these units and to use the information of higher level subsystem to pinpoint the subsystem which is causing the fault. The main advantage of this method is that we can overlook irrelevant information in the initial stages of diagnosis and quickly focus in diagnosing the problem areas.

1.4.6 Typology of diagnostic search strategies

There are mainly two different methods to search for fault in a system, *topographic search* and *symptomatic search*. In topographic search we perform the fault analysis using a template of normal

operation and in symptomatic search we look up for symptoms to localize the fault.

Topographic search

In topographic search we can search in the malfunctioning system with reference to a template which represents the normal or desired operation. The fault is located as a mismatch and is localized by its location in the system. This type of search strategy is called topographic search.

Decomposition techniques As discussed topographic search strategies relies on searching with a template of model of normal operating region and is therefore applicable to places where the disturbance are not known beforehand or not known by the system designer. As the faults are not known beforehand, with the help of topological search we can narrow the search to a subsystem in a process. The topographic search can be understood by the figure(1.5).

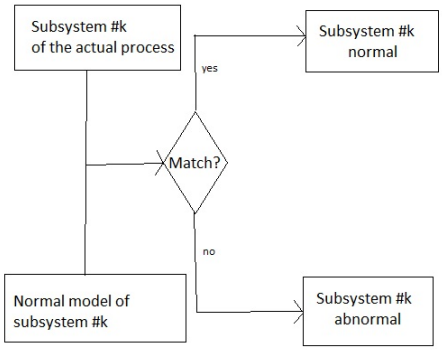

Figure 1.5: Topographic search

Topographic search can be done in two ways either structural or functional search. In structural search, we first identify the path of information flow from input to the desired unit and its output. If there is a fault in the followed path, then all the subcomponents which lies on this information flow path are included in the hypotheses set. Localization is performed by refining the search by selecting the subcomponents. Based on the search we isolate a set of good path and bad paths. The subset of bad paths that lie on the good paths are assumed to be good and the cardinality of hypotheses set is reduced.
In functional search, we use the functionality of various groups of subcomponents to search for the fault. Improvement can be made in localizing by searching from top to down through the sub models. Like structural search here also we classify the sub models as normal sub models and abnormal sub models. The basic assumption made here is that individual components in abnormal and normal set is assumed to be normal, to reduce the cardinality of hypotheses set. In most of the process plant where topographic search method is used, a combination of functional and structural search is preferred.
But the problem with topographic search is that the designer has no knowledge about the faulty modes of operation. All the assumptions that are made is only about the normal operating region, due to which searches are comparatively not sensitive to novel faults and multiple faults.

Symptomatic search

In symptomatic search we take a set of observation which represents the abnormal state of the system is used as a template to find a matching set from the library of known symptoms which are related to different faults. The main feature of symptomatic search are these methods derive onto decision from structure of the data sets and the relationship between them. The main drawback which symptomatic search suffers is that the observation of faulty condition must be available, and multiple faulty conditions and novel faults are difficult to detect.

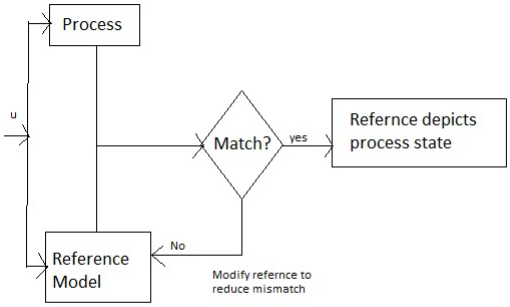

Figure 1.6: Closed loop approach

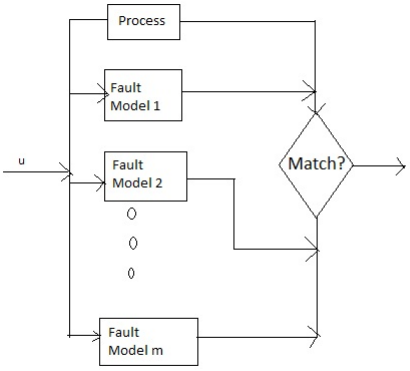

Figure 1.7: Open loop approach

In the symptomatic search methods the input that is supplied to the process is also fed to the reference model. The output of the model and the output from the process is then compared. There are mainly two approaches of symptomatic search, open loop and closed loop approach. Open loop approach is followed where there are finite states with each states corresponds to distinct plant parameters. In such a case if all the states are clearly stated then we follow open loop method. If the states are generated from an estimator and then tested for mismatch, then we follow open loop method.

As in figure(1.6) shows the basic schematic of closed loop approach symptomatic search. In the this method, a specific hypotheses is chosen and is tested if the reference model matches the process. If the reference model does not matches, then a new reference model is chosen. In closed loop approach, the data from the mismatch between the process and current reference model is used to generate new hypotheses. But unlike closed loop approach, in open loop approach many reference models are there with different hypotheses. Then we compare reference model and the process, the reference model that is closest in approximating the process is kept.

1.5 Process history based methods

Unlike model based approach(either quantitative or qualitative) where we require a priori knowledge about the process, in process history based method we require only large amount of historical process data. There are different approaches in which this large amount of historical data can be converted to a priori knowledge and provided to the diagnostic system. This process is known as feature extraction. Mainly there are two approaches of feature extraction qualitative approach or

quantitative approach.
There are two ways in which qualitative feature extraction they are trend modeling and expert system methods. Whereas in quantitative feature extraction it is classified as either statistical or non-statistical methods. Under the category of non-statistical classifier neural network is an important method. Principal component analysis(PCA)/partial least squares(PLS) and statistical classifier fall under statistical feature extraction methods.

1.5.1 Qualitative feature extraction

As discussed Qualitative feature extraction has two approaches expert system and trend modeling.

Expert system

Expert system are basically a very specialized system that can solve problem in very narrow domain of expertise. The main elements in constructing an expert system are

i. knowledge acquisition
ii. the way in which we represent the knowledge
iii. coding the knowledge into the knowledge base
iv. designing inferencing machine for diagnostic reasoning
v. lastly, developing the input output interfaces.

The advantage of expert system for solving diagnostic problems are

- its easy in constructing.
- we can get clear easy reasoning
- able to reason under uncertainty.
- the potential to explain the solution that is provided.

In the domain of Artificial Intelligence(AI), expert system is defined as a computer system that mimics the decision making ability of human expert. The expert system are constructed primarily by *if-then* rules rather than the conventional procedural codes in C or FORTAN. The main objective of expert system is divided in two steps. Firstly, the system categorizes the causes of the current problem under operator error, system disturbance or equipment failure. Secondly, the expert system takes prescriptive decisions to bring the process to normal operating condition.
In expert system the knowledge base contains the facts and the rules. Basically it stores complex structured models which are well described and unstructured models which are not well defined.
The advantage of expert system is that it is easy to construct relatively and its crisp reasoning is also another advantage. But it suffers from various drawbacks such as they are system specific, i.e rules and data of one system cannot be applied to other and they are difficult to update in case of novel cases.

Qualitative trend analysis

Trend analysis and prediction are important concept of process controlling. By trend of process we mean the course of event with time. Trend analysis is useful in explaining various key events that are happening in the process, and for detecting and diagnosing faults in the process and for predicting the future states of the process. In order to get system signal trend which is not affected by instantaneous noise, we employ filtering to clip the noise. Example of such filter is auto regressive

filter with particular filter coefficients to the required degree of smoothing. But the problem with such kind of filters is that it cannot distinguish between the transient noise and persistent fault or instability[5]. To avoid this problem, we should view the trend from different time scale or from different levels of abstraction. In most of the cases, process faults generate a distinctive trend in the sensor monitored. These distinctive trend can be suitably exploited in identifying the abnormality in the process. Thus suitable categorization and analyzing the process trend we can detect the fault earlier and control the process quickly.

The two main objective of qualitative trend analysis are

- Identifying the nature of trend represented by the series of observations.
- Identifying the fault and predicting the future values of a time series variable.

There are different types of trend that trend analysis can detect, they are

- Period trends
  - Short-term trend which is for a short duration of time caused due to noise.
  - Long-term trend which is for a long duration of time caused due to faults or malfunction in the process.
- Composite trends which have a definite pattern such as the S-curve or logistic curve.

1.5.2 Quantitative feature extraction

In this section we briefly describe fault detection methods which are based on quantitative feature extraction. In quantitative methods we construct the fault diagnosis problem as a pattern recognition problem. In pattern recognition problem we classify the data points to predetermined classes. Statistical methods like Bayes classifier use the a priori class distribution knowledge to perform the classification. Whereas methods like PCA extract the major trends in the data using less number of relevant factors. Unlike statistical methods, in neural networks we assume a function for making decisions thus we are parameterizing the classifier.

Statistical feature extraction from process data

Each and every real life processes are affected with the random noise or fluctuations. Unlike deterministic systems, we cannot comment on the future states of the stochastic system based on past and present states and future control inputs. As the system is corrupted by random noise it is necessary to construct the system in probabilistic fashion. When the process is working under normal condition, the observations have a certain probability distribution which corresponds to the normal operating region. So the probability distribution changes when the process is in faulty condition. Generally the probability distribution depends parameters such as mean and standard deviation when parametric approach is used used for monitoring, so under faulty conditions the monitored variable or data, its mean or standard deviation may differ from the mean and standard deviation of the normal data. So, it can be concluded that fault diagnosis can be classified as problem of identifying the changes in the parameter of dynamic or static system of a probabilistic system.

Principal components analysis

In this section we are going to discuss about PCA briefly. PCA is mainly based on orthogonal decomposition of the data matrix along the direction that maximizes the variance of the data[12].

The main motto of using PCA is to find the reduced space of the data which has lower dimension than the original data matrix, and which can explain the major trend of the original data matrix.

PCA is elaborately described in chapter 2. Briefly we can discuss the method. Let $\mathbf{X}$ denote the data matrix having dimensions $n \times p$. The columns of $\mathbf{X}$ are mean centered and scaled. The covariance matrix of $\mathbf{X}$ is represented by $\mathbf{\Sigma}$. Here$p$ denotes the number of process variables and n denotes the number of samples or observations. We can decompose $\mathbf{\Sigma}$ to a diagonal matrix $\mathbf{L}$ with the help of orthogonal matrix $\mathbf{U}$ having dimension $p \times p$, mathematically it can be said as singular value decomposition of covariance matrix. So, $\mathbf{\Sigma}$ can be written as

$$\mathbf{\Sigma} = \mathbf{ULU}'$$

The columns of $\mathbf{U}$ are known as *principal loading vectors*. The matrix $\mathbf{L}$ contains $\lambda_1, \lambda_2, ..., \lambda_p$ the eigenvalues of $\mathbf{\Sigma}$ in descending order. The eigenvalues explain the variance explained by the corresponding eigenvectors(principal loading vectors). Mathematically principal component transformation can be written as follows

$$\mathbf{T} = \mathbf{XU} \quad \text{or} \quad \theta_i = \mathbf{X}u_i$$

which implies that $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{TU}' = \sum_{i=1}^{p} \theta_i {u_i}' \tag{1.30}$$

Where the matrix $\mathbf{T}$ having dimension $n \times p$ are called the *principal score matrix* and θ_i represent the columns of the score matrix. If the data matrix is correlated, then the variation in the data can be explained by few principal components which is less than p. Let us choose only a principal components such that $a < p$ then the decomposition is as follows,

$$\mathbf{X} = \theta_1 {u_1}' + \theta_2 {u_2}' + \ ... \ + \theta_a {u_a}' + \mathbf{E} = \sum_{i=1}^{a} \theta_i {u_i}' + \mathbf{E} \tag{1.31}$$

where $\mathbf{E}$ is known as *residual matrix* which is having dimension same as data matrix. Mostly it has been encountered that first two or three principal components are often adequate to explain the variance in the data. So we can see that the dimensionality of the data is reduced.
For monitoring a process and quality control with the help of PCA, several other steps are there in addition to the above method, which will be elaborately discussed in chapter 2.
As in PCA we operate on a single block of matrix which is known as data matrix. But many a times we have an additional matrix, for example product quality variables matrix $\mathbf{Y}$. So, most of the time it is desirable to use the data in the matrix $\mathbf{X}$ to predict and to detect changes in matrix $\mathbf{Y}$. This task is accomplished by PLS method. Basically it enhances the relationship between two blocks of data with the help of a mathematical model and simultaneously compressing both the data blocks.

Statistical classifier

Fault diagnosis is a classification of different patterns or trends in the data. Hence it can be said that its a classical statistical pattern recognition framework. The Bayesian classifier for a two class

problem, where we have Gaussian density function for different classes is given by [13],

$$d_1 = (y - m_1)^T {\mathbf{\Sigma}_1}^{-1} (y - m_1)$$
$$d_2 = (y - m_2)^T {\mathbf{\Sigma}_2}^{-1} (y - m_2)$$

where d_1 and d_2 is the density function for the two different classes. Let h be a discriminant function, it is given mathematically as

$$h = d_1 - d_2$$
$$h < \delta, \text{ x belongs to class I}$$
$$h > \delta, \text{ x belongs to class II}$$

where δ is known as the threshold of classifier given by $\delta = \log \frac{|\Sigma_2|}{|\Sigma_1|}$. This idea can be transferred to n class problem, we can keep on classifying x into two more sub-classes. Point to note here is that Bayes classifier is an optimal classifier only when the classes are Gaussian distributed and the information regarding the distribution is also available. If $\Sigma_1 = \Sigma_2 = I$, then the Gaussian distribution just converts to $l^2 - norm$ or it is said as Euclidean distance based classifier. Such type of distance classifier calculates the metric distance of a pattern from different known classes and then classify the pattern to the class which has the least distance or most similarity.

1.5.3 Neural networks

The neural network is a part of machine learning in which we model a process by mimicking the biological neural networks of humans. Neural network basically estimate approximate function which depends on huge number of input in which most of them are unknown. In general fault diagnosis can be employed with the help of neural network in two ways,

i. on the basis of architecture of the system network such as sigmoidal, radial basis etc.
ii. on the basis of learning strategy such as unsupervised and supervised learning.

In supervised learning technique, we basically choose a structure specifically for the neural network, the network is parametrized such that the problem statement is estimating the connecting weights. The connecting weights are determined by explicitly employing the mismatch between the actual and desired values for leading the search. Due to this property supervised neural network a better option for fault classification as described in [14]. On the other hand spectrum are the type of neural network structure which employs unsupervised estimation techniques. These type of neural network are commonly known as self-organizing neural network due to the fact that the structure is adaptively decided based on the input which is supplied to the network.
The popular methodology that is used commonly in the supervised learning technique is the back propagation algorithm. This algorithm is used in field of fault diagnosis. Back propagation is mainly abbreviated for backward propagation of error. In back propagation algorithm we require a known calculated output for the corresponding input value for calculating the loss function gradient. For further review on how back propagation is employed in fault diagnosis we can refer to [15].

1.6 Comparison between various fault diagnosis methods

Quantitative model based approach such as observer-based methods and parity space methods have many desirable characteristics, if we have adequate knowledge about the inputs and outputs of the system and also if we have adequate knowledge about the interactions caused due to environmental change, then fault diagnosis is comparatively easy regardless to the number of faults. But the main drawback of these approaches is simplistic approximation of noises and disturbances which also includes modeling errors. Another major problem is that we linearize a nonlinear model for simplistic approach near the operating point which causes error. And also for a bulky process the size of filter bank increases surprisingly.
Rule based expert system such as digraphs and fault trees can used where we do not have exact knowledge or fundamental principles are inadequate but we have abundant experience so we can develop qualitative models. Qualitative model based approach are good alternative when quantitative models are unavailable but the functional dependencies are well defined. Another advantage of such approach is that it can provide explanation as to how the fault propagated in the system. However the main drawback is that they suffer from generation of erroneous solution.
Diagnostic classifier constructed from process history based methods such as neural network architectures like radial basis function and ellipsoidal units perform well under the influence of noises and disturbances, but they require huge amount of training data which is the historic process data for its classification. It cannot detect novel faults easily plus the method also suffers problem in detecting multiple faults.

From all the discussion the most widely methods from industrial point of view are process history based methods mainly the statistical methods. This is due to the reason that process history based methods require very less effort in modeling the system. And very less a priori knowledge about the system is required. Plus statistical approaches are easy to build and provide fast decision in detection of faults.

Chapter 2

Principal Component Analysis

Principal Component Analysis in short known as PCA is the most common approach used for dimensionality reduction which is used for detecting and diagnosing faults in chemical process plants. PCA is mainly very useful in fault detection rather than fault isolation and diagnosis. The main aim of PCA is to extract important information from the data matrix and to map the data in reduced space with the help of new orthogonal variables called the principal components. PCA was first derived by Pearson in the year 1901, but later it was modified by Hotelling[16] in year 1933 who presented the method of using covariance/correlation matrix of the variables for analyzing PCA.

2.1 Introduction

Large amount of data is generated by most of the modern day chemical plant. Previously most of the process plant were operated on univariate variables such as temperature, density, pH etc and single variables were monitored at a time. But with the advancement of technology, there were occasions were monitoring of more than one variable at a time was necessary for exploiting the inter-relationship between them. One such multivariable analysis is PCA which highlights the similarity and dissimilarity between the variables in a data matrix. The basic thinking is that, we have a multivariable matrix $\mathbf{X}$ and we transform it into new variables which are linear combination of the original variables and are also orthogonal to each other.
The main motivation for using PCA in process monitoring is due to following reasons

i. Firstly PCA reduces the dimension of the data, so with the help of normal operating data(training set) we can model the process plant, so that for a new set of data we don't have to tackle the entire dimension for process monitoring and control.
ii. Secondly the structure generated by PCA can be helpful in identifying the variable that is the cause of fault and can also comment about the variable that is most affected by the faults.
iii. Thirdly PCA can separate out the data matrix into two parts one containing the systematic trends of the process and second containing the noise. And most of the fault affects anyone of the parts in a data matrix.

2.2 Graphical interpretation of PCA

The concept of PCA can be illustrated with the help of figure(2.1) taken from[17]. There are three axes in the data which are the three variables. The bold line corresponds the first principal component which is the minimize Euclidean distance from all the points to the line which corresponds to the first principal component, which captures the maximum variance in the data. The second bold line corresponds to the second principal component which is orthogonal to the first principal component and captures the second maximum variance in the data. In crude sense we can say that, PCA minimizes the the error orthogonal to the principal components.

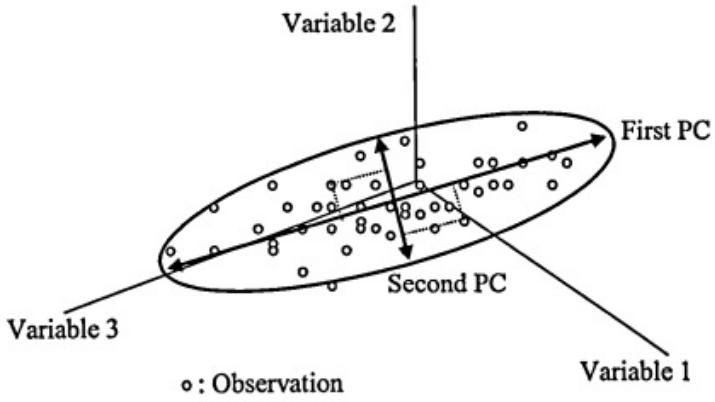

Figure 2.1: Graphical interpretation of PCA[18]

2.3 Mathematical procedure of PCA

PCA is basically a linear dimensionality reduction technique, which projects the original data to lower dimensional space with minimum loss in information. It maximizes the variance in the data. In PCA we determine a set of orthogonal vectors which are termed as *principal components*. There are various ways in which PCA can work to get the principal components like the Nonlinear Iterative Partial Least Square algorithm(NIPALS) or it can be derived from Singular Value Decomposition(SVD) or from Eigenvalue decomposition of the covariance matrix. The basic difference between methods is that, NIPALS algorithm computes the principal components recursively whereas in SVD we get the principal components at the same time by performing SVD operation on data matrix, whereas in eigen value decomposition we calculate the covariance matrix of the data matrix then calculate the principal components.

2.3.1 PCA by Eigenvalue decomposition

Let the data matrix $\mathbf{X}$ has dimensions $n \times m$, where n denotes the number of samples/observations and m denotes the number of process variables. Pre-filtering of the data matrix is done in order to separate the errors from the variables. Pre-filtering of the data matrix includes mean centering and auto-scaling of each of the column in $\mathbf{X}$. The data matrix $\mathbf{X}$ can be decomposed in the sum of series

of vector termed as *loading vectors* denoted by p_i and *score vectors* denoted by t_i and R denotes the number pf principal components which are less than or equal to the smaller dimension of the matrix $\mathbf{X}$, i.e $R = min\{n, m\}$.Generally in process plant where $n >> m$, $R \leq m$. $\mathbf{E}$ is the residual matrix. Mathematically it can be written as,

$$\mathbf{X} = t_1 {p_1}^T + t_2 {p_2}^T + ... + t_m {p_R}^T + \mathbf{E} = \sum_{i=1}^{R} t_i {p_i}^T + \mathbf{E} \tag{2.1}$$

Basically the score vector t_i represents the relationship of how the samples are related to each other and the loading vector p_i describes the relationship among different variables.
After pre-filtering of data, we calculate the covariance matrix of the pre-filtered data matrix, which is given by

$$\mathbf{S} = cov(\mathbf{X}) = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \tag{2.2}$$

The loading vectors are the eigenvectors of the covariance matrix of data matrix $\mathbf{X}$, which is calculated as follows

$$cov(X)p_i = \lambda_i p_i \ ; \ i = 1, 2, ..., R \tag{2.3}$$

The principal components are stacked horizontally in descending order according to the corresponding eigenvalues $\lambda_i (i = 1, 2, ..., R)$.$\lambda_i$ gives the measure of variance explained by its corresponding eigenvector/loading vector. The loading vector having the highest value of eigenvalue is the first principal component, the loading vector having the second largest eigenvalue is the second principal component and so on, and in matrix format let it be denoted by $\mathbf{P}$.
The corresponding score vector can be calculated with the formula

$$t_i = \mathbf{X}p_i \quad i = 1, 2, ..., R \tag{2.4}$$

We can see that the score vector for the corresponding principal components are the linear combinations of the variables in data matrix and the coefficients are given by the loading vector p_i. So, in matrix form let the score vector which is constructed by horizontally stacking t_i with the corresponding principal component is represented by $\mathbf{T}$.
So the residual matrix can be given by

$$\begin{aligned} \mathbf{E} &= \mathbf{X} - \hat{\mathbf{X}} \\ \hat{\mathbf{X}} &= \mathbf{T}\mathbf{P}^T \end{aligned} \tag{2.5}$$

where $\hat{\mathbf{X}}$ denotes the conversion of score matrix back to the original m-dimensional space. The residual matrix $\mathbf{E}$ captures the difference of the data matrix that is reconstructed from the principal scores and principal loading associated with $m - R$ eigenvalues.
Each of the eigenvalue, λ_i of the covariance matrix of X indicates the percentage of variance explained in the corresponding direction of loading vector p_i, which is given by

$$\text{Percentage of variance explained by } p_i = \frac{\lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{2.6}$$

2.3.2 PCA by Singular value decomposition

Singular value decomposition(SVD) is an algebraic technique of factorizing real or complex matrix. It can be looked from three mutually consistent view points[19] as follows

i. It is a technique for converting correlated variables in a matrix into a set of uncorrelated variables so that we can exploit the inter relationship amongst different variables.
ii. Simultaneously it is technique in which we identify and order the variables along which the data shows maximum variance.
iii. It simultaneously reduces the dimension of the data along which the variance of the data point is maximum.

So let us see how we use SVD in our PCA methodology. The pre-filtering step as discussed in above section is also same for SVD. So for a mean centered and auto-scaled data matrix $\mathbf{X}$ we can decompose it into

$$\frac{1}{\sqrt{n-1}}\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2.7}$$

where $\mathbf{U}$ is a matrix having dimension $n \times n$ and matrix $\mathbf{V}$ having dimension $m \times m$ and both of them are unitary matrix. The matrix Σ having dimension $n \times m$ is a diagonal matrix containing the non negative real singular values in the descending order such that $\sigma_1 \geq \sigma_2 \geq .... \geq \sigma_m \geq 0$. The loading vectors is the orthonormal column vectors of the matrix $\mathbf{V}$, and the variance explained by each column is given by ${\sigma_i}^2$ where $i = 1, 2, ..., m$.

Now if we take transpose of equation(2.7) and multiply it with itself then we get,

$$\begin{aligned}
&\frac{1}{\sqrt{n-1}}\mathbf{X}^T\mathbf{X}\frac{1}{\sqrt{n-1}} = \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right) \\
&\frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\
&\mathbf{U}^T\mathbf{U} = I \\
&\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T
\end{aligned} \tag{2.8}$$

where $\mathbf{\Lambda} = \mathbf{\Sigma}^T\mathbf{\Sigma}$ contains the positive real eigenvalues of the covariance matrix in descending order($\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_m \geq 0$). And for $i = 1, 2, ..., m$, $\lambda_i = {\sigma_i}^2$ that is the square of the singular value of data matrix is equal to the eigenvalue of the covariance matrix of $\mathbf{X}$.

2.3.3 PCA from NIPALS algorithm

Non-linear iterative partial least squares (NIPALS) algorithm we compute the principal components by decomposing the data matrix recursively. The schematic diagram of the process is shown in figure(2.2).

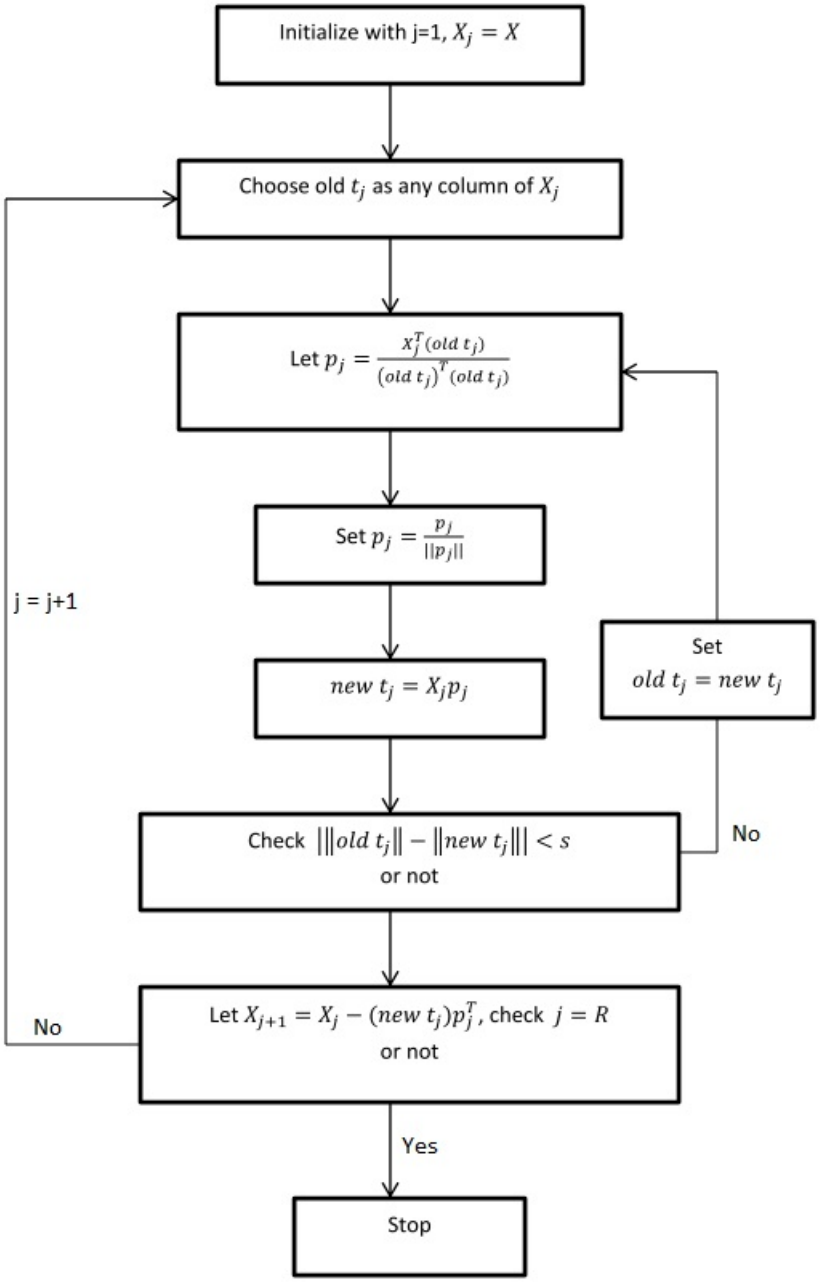

Figure 2.2: Block diagram of NIPALS algorithm

Let us consider the data matrix $\mathbf{X}$ having dimension $n \times m$. And let R denote number of principal components such that $R \leq m$. The algorithm is as follows

1. Pre-filtering of data is done such that each column of $\mathbf{X}$ has mean zero and unit variance. Initialize with $j = 1$. And let the pre-filtered data matrix be denoted by $\mathbf{X}_1$
2. We select randomly any column of $\mathbf{X}_1$ as t_j, which is referred as *old* t_j in diagram for convenience.
3. Then we calculate the corresponding loading vector p_j, with the help of the formula

$$p_j = \frac{\mathbf{X}_j{}^T (old\ t_j)}{(old\ t_j)^T (old\ t_j)} \tag{2.9}$$

4. We normalize the loading vector obtained to unit length.

$$p_j = \frac{p_j}{\|p_j\|} \tag{2.10}$$

5. Calculate the score vector t_j which is referred as *new* t_j in the diagram.

$$t_j = \mathbf{X}_j p_j \tag{2.11}$$

6. We check for convergence which is the difference of sum of squares of the vectors, where s is a small number such as 0.0001. If the *new* t_j has not converged then update *old* t_j to *new* t_j and jump to step 3 and repeating the steps till convergence is achieved.
7. If the convergence is achieved in step 6 and if $j < R$ then we update $j = j + 1$ and update the data matrix $\mathbf{X}_j$ as

$$\mathbf{X}_{j+1} = \mathbf{X}_j - (new\ t_j) p_j{}^T \tag{2.12}$$

   and jump to step 2 to calculate the next principal component.
8. We stop the loop when $j = R$, i.e when we have all the required number of principal components.

As compared to NIPALS algorithm, SVD is a more efficient method for calculating the principal components. SVD is computationally fast as compared to NIPALS algorithm.

2.4 Selecting the number of principal components

One of the important step in applying PCA is determine the suitable number of principal component which can capture the maximum variance in the data. A densely correlated data will usually require two or three principal components for describing the data in the reduced space with very less loss in the information. But in process which does not have densely correlated variables if we choose less number of principal components will incorporate more noise in the PCA model which will affect the sensitivity of the process, whilst choosing more number of principal components than required to explain the maximum variation in the data will not result in proper dimensionality reduction of the original data. Basically by improper choice of number of principal components will adversely affect the performance of the model in terms of sensitivity and process monitoring performance. There are number of methods which have been proposed for selecting the number of principal components which mainly rely on the cumulative percentage of variance explained or cross-validation of data.

2.4.1 Kaiser method

This method was first proposed by Henry Kaiser in the year 1960[20]. This is one of the extensively used method for retaining the number of principal components. According to Kaiser method we retain the principal components that have their corresponding eigenvalues greater than one. Mostly due to the simplicity of the method it is widely used rule for retaining the number of principal components. But it also has drawbacks like if we have eigenvalue having value 1.1 and 0.9 the method will not consider the principal component corresponding to 0.9 as important for retaining in the model. Secondly, the rule sometimes chooses more number of principal components or vice-versa in some cases. Mostly as discussed in [20], the number of principal components that are retained by Kaiser method is generally between one-third and one-fifth or one-sixth of the variables.

2.4.2 Scree test or Scree plot

The Scree test is a graphical method of determining the number of principal components to be retained. It was formulated by Cattell[21] in the year 1966. The methodology is that the eigenvalues are arranged in descending order and are plotted by connecting with each other. The plotting is examined to determine the *elbow point* or the point at which there is a sudden drop or the point where we encounter a flat region after a steep slope. We choose the loading vectors corresponding to all the eigenvalues that are above the elbow point. The logic is that the elbow point divides the major components from the minor components. The drawback of this method is that there is no specific definition between for the elbow point. In lot of cases it has been seen that there is no steep decrease in the value of eigenvalues which does not results in the elbow. In figure(2.3) we can

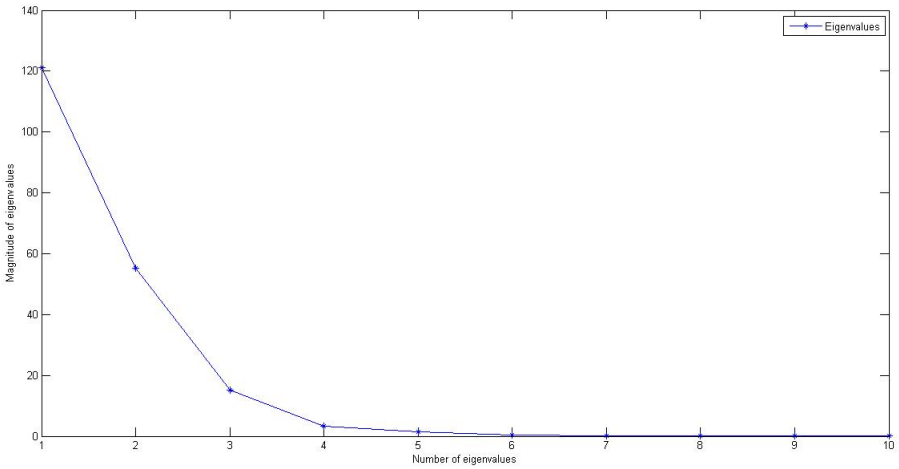

Figure 2.3: Example of Scree plot

select either two or three principal components out of ten. It has been experimentally seen that performance of Scree test is more accurate that Kaiser method.

2.4.3 Proportion of total variance

This is an ad-hoc method for selecting the number of principal components. Here we select the principal components whose cumulative sum of variance is above some predefined value. Mostly the predefined value is $85 - 95\%$ of the total variance explained.

2.4.4 Average of eigenvalues

Another simple method by which can determine the number of principal components is by average of eigenvalues method[22]. In this method we keep only the components whose corresponding eigenvalues is greater than the average of all the eigenvalues. Mathematically it can be defined as keeping the k^{th}-component if it satisfies the condition

$$\lambda_k > \frac{1}{m}\sum_{i=1}^{m}\lambda_k \tag{2.13}$$

where m denotes the number of variable. Due to simplicity of the method it is also widely used method, but the drawback is that by choosing principal components via this methods sometimes results in loss of information.

2.4.5 Parallel Analysis

This method is developed by John Horn[23] in the year 1965. This a modified method of Scree test. In this method we plot the scree plot of the given process and on the scree plot we superimpose the scree plot for a data having variables which are not correlated to each other. The point where the

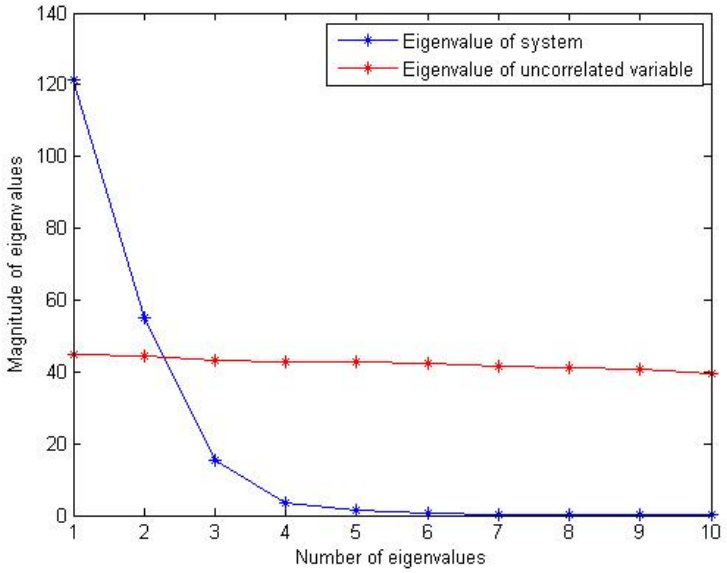

Figure 2.4: Parallel Analysis

two plot intersect, separates the significant principal components from the insignificant one. In the figure(2.4) we can see that the blue line describes the scree plot for the actual process whereas the

red line is the scree plot for the set of uncorrelated variables. So from the plot we can conclude that for the process we can take two principal components to explain the maximum variability in the data. But the main problem with this method is that if we choose the number of samples different for a given process then sometimes we may get different number of principal components for the same process.

2.4.6 Cross-validation method

The cross validation method proposed by S Wold[24][25] in the year 1978. It is the most reliable method for calculating the number of principal components. The basic methodology is that we remove every row(sample) of the data matrix and compute the PCA model with the remaining rows. The row that was removed previously is estimated using the PCA model that was computed by deleting the row and then we calculate Predicted Error Sum of Squares(PRESS) for the deleted row. The number of principal components are chosen based on the model(number of loadings) which generates the minimum PRESS. We can understand the method with a help of example[25].

- Suppose we have a matrix $\mathbf{X}$ which has dimension 100×5.
- So the first step is to divide the data in different groups($\mathbf{G}$). For the example we divide the data in four groups $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4$ each having 25 samples. The size of sample can be different.
- The next step is to remove anyone of the group of data and compute PCA model(loadings) for the remaining groups. Suppose we remove $\mathbf{G}_4$ and compute PCA for the rest of the group of data.
- The first loading vector is then used to determine the scores for the group that was removed.

$$\mathbf{T}_{G_4} = \mathbf{X}_{G_4}\mathbf{P}_{(G_1-G_3)} \tag{2.14}$$

  where $\mathbf{T}_{G_4}$ is score matrix of the removed group calculated with the help of loadings $\mathbf{P}_{(G_1-G_3)}$ of the remaining group of data.
- The calculated score matrix is then used to calculate the predicted value of data in $\mathbf{G_4}$.

$$\hat{\mathbf{X}}_{G_4} = \mathbf{T}_{G_4}(\mathbf{P}_{(G_1-G_3)})^T \tag{2.15}$$

- Then we calculate the PRESS for the $\mathbf{G}_4$

$$E_4 = \sum_{i=1}^{25} \|\hat{\mathbf{X}}_{G_4} - \mathbf{X}_{G_4}\|^2 \tag{2.16}$$

- The above steps are calculated similarly by removing the other groups of data one at a time and calculating their individual press and then adding them to get the overall PRESS using the first principle component

$$Total\ PRESS = E_1 + E_2 + E_3 + E_4 \tag{2.17}$$

- The full procedure is repeated for increasing the number of principal component and calculating the $Total\ PRESS$. For our example we will iterate the process five times as we will be having

five loading vectors. The number of loading vectors for which PRESS is minimum are the number of principal components that has to be chosen.

There are many ways in which we can group the data. The simple way is to group the data in the sequence it was generated like in example we grouped 1-25, 26-50, 51-75 and 76-100. Or we can group the data randomly having different group length the minimum length being one and repeating the samples. Another method is by randomly choosing and non repeating samples. From experimental results it has been found that by the nature of grouping the number of principal components does not gets affected.The only drawback with this method is, it consumes a lot of time as compared to the other methods but it is the most reliable method for determining the number o principal components.

2.5 Process monitoring using PCA

2.5.1 Score Plot

As we have seen in the previous section that the principal components are the eigenvectors of the covariance matrix of the data matrix. And we have also calculated the principal score matrix corresponding to the principal components. Plotting the principal component loadings and the score could reveal the underlying relationships between the variables and the observations of the data matrix.

The score plot can be represented either in univariate or bivariate format. The control limits are calculated using the statistical property of the normal data which is used to construct the PCA model. The score plot contains the common variations that affect the process, therefore the scores are assumed to be normally distributed and independent. The control limit is given by the formula

$$\pm t_{n-1,\frac{\alpha}{2}} s_{est} \sqrt{1+\frac{1}{n}} \tag{2.18}$$

where n is defined before as the number of samples, α corresponds to the significance level, generally value of α is 0.01 or 0.05 corresponding to 99% and 95% confidence value respectively, s_{est} denotes the estimated standard deviation of the scores calculated with principal component taking one at a time for all the samples, $t_{n-1,\frac{\alpha}{2}}$ denotes the t-distribution with $n-1$ degree of freedom.

2.5.2 Hotelling's T^2-statistics

This method was proposed by Hotelling in the year 1947. It is usually referred as T^2-statistics. The T^2 statistics is similar to the Mahalanobis distance which measures the distance of each point using the covariance matrix. In other words T^2 statistics can be said as the scaled(mean centered and unit variance) square of l^2-norm of each of the sample in $\mathbf{X}$ from its mean. If for the normal data the covariance matrix is known the the T^2 statistics is defined as

$$T^2 = (\mathrm{x}-\bar{\mathrm{x}})^T \sum^{-1} (\mathrm{x}-\bar{\mathrm{x}}) \tag{2.19}$$

where x denotes the vector of samples and $\bar{\mathrm{x}}$ denotes the mean vector. T^2-statistics follows chi-square(χ^2) having m degree of freedoms and m represents the number of variables. The multivariate

T^2 plot can be constructed by calculating T^2 values for each samples and plotting it versus time. The control limit for the T^2 distribution can be calculated with the help of F-distribution.

$$T_\alpha^2 = \frac{m(n-1)}{n-m} F_\alpha(m, n-m) \tag{2.20}$$

where, n and m are number of samples and variables in matrix $\mathbf{X}$ and $F_\alpha(m, n-m)$ denotes the upper $100\alpha\%$ critical point of the F-distribution having m and $n-m$ degree of freedom[26].

Employing T^2-statistics for fault detection using PCA

Let a matrix $\mathbf{X}$ having dimension $n \times m$ contains the normal operating data of a process. We perform PCA with the help of singular value decomposition as explained in the section(2.3). Suppose we select R principal components from any one of the procedures explained in the section(2.4). And we construct the PCA model by choosing only first R loading vectors and horizontally stacking them in a matrix $\mathbf{P}$ having dimension $m \times R$. Let $\mathbf{\Sigma}_R$ contains the first R rows and columns of $\mathbf{\Sigma}$. We calculate control limit using the equation(2.20).
Now for a new set of data $\mathbf{X}_{new}$ we calculate the T^2-statistics using the formula

$$T_i^2 = \mathrm{x}_i \ \mathbf{P} \ \Sigma_R^{-2} \ \mathbf{P}^T \ \mathrm{x}_i^T \quad \text{where, } i = 1, 2, .., n \tag{2.21}$$

x denotes the sample vectors of $\mathbf{X}_{new}$. We plot T_i^2 versus i. And the instant where T_i^2 violates the control limit as calculated from equation(2.20) is the point in time where fault has occurred.

2.5.3 Q-statistics or residual analysis

The Q-statistics was proposed by Jackson and Morris in the year 1957.With the help of T^2-statistics we can monitor the faults among the different variables in the reduced space generated by the principal components. Plus T^2-statistics is sensitive to error in the reduced space of PCA. Q-statistics monitors the residual space. Basically it is the Squared Prediction Error(SPE) of the actual data and the estimated data obtained from the PCA model. With the help of Q-statistics we can detect the fault in the variables. As Q-statistics does not measures the differences due to each individual principal component as in T^2-statistics, it measures the overall difference of principal components taken simultaneously so it is insensitive to the variation caused by principal components which has smaller eigenvalues compared to other principal components having large eigenvalues. In simpler language we measure the error in the observation space which has been incorporated due to excluding some of the loading vectors. The methodology is as follows.

We have a data matrix $\mathbf{X}$ having dimensions $n \times m$ and x_i represents the observation vector corresponding to $i = 1, 2, ..., n$. And $\hat{\mathbf{X}}$ represent the estimated data that has been reconstructed by using the PCA model by retaining R principal components out of m and $\hat{\mathrm{x}}_i$ represents the observation vector of $\hat{\mathbf{X}}$. The sum of the squares of error is given by,

$$\mathbf{Q}_i = (\mathrm{x}_i - \hat{\mathrm{x}}_i)(\mathrm{x}_i - \hat{\mathrm{x}}_i)^T \tag{2.22}$$

The $\mathbf{Q}_i$ represents the metric distance of the sum of squares of the difference between actual and reconstructed data in the R-dimensional space which is defined by the PCA model.

The control limit for the Q-statistics is calculated by[27],

$$\theta_1 = \sum_{j=R+1}^{m} \sigma_j^2$$
$$\theta_2 = \sum_{j=R+1}^{m} \sigma_j^4$$
$$\theta_3 = \sum_{j=R+1}^{m} \sigma_j^6$$

and

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{2.23}$$

where σ_j corresponds to the singular values corresponding to the loading vectors calculated by SVD algorithm and $j = 1, 2, .., R, .., m$.
Then we define a quantity c as,

$$c = \theta_1 \frac{\left[\left(\frac{Q}{\theta_1}\right) - \frac{\theta_2 h_0(h_0-1)}{\theta_1^2} - 1\right]}{\sqrt{2\theta_2 h_0^2}} \tag{2.24}$$

The quantity c approximately is normally distributed having zero mean and unity variance. The control limit for Q-statistics can be calculated as,

$$Q_\alpha = \theta_1 \Big[\frac{c_\alpha\sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0(h_0-1)}{\theta_1^2} + 1\Big]^{1/h_0} \tag{2.25}$$

where c_α is the normal deviation corresponding to area chopped by α under the lower tail if h_0 is negative or under the upper tail if h_0 is positive.
Another method which has been used frequently for calculating Q_α is by assuming the distribution of c_α asymptotic to χ^2-distribution. By this method Q_α can be calculated by[18]

$$Q_\alpha = f\ \chi^2_{h,\alpha} \tag{2.26}$$

where f and h is defined as,

$$f = \frac{\theta_2}{\theta_1}$$
$$h = \frac{\theta_1^2}{\theta_2} \tag{2.27}$$

Employing Q-statistics for fault detection using PCA

For a normal operating process data $\mathbf{X}$, we calculate the PCA model and also compute the control limit, Q_α. Then for a new set of data $\mathbf{X}_{new}$ we calculate the Q-statistic as in equation(2.22) where the estimated values of the observation vector $\hat{\mathrm{x}}_i$, are calculated using the PCA model of the normal operating data. Then we plot $\mathbf{Q}_i$ versus i. And for faulty region the specific values of i the corresponding $\mathbf{Q}_i$ violates the threshold Q_α.

2.6 Contribution plots

After applying PCA and using multivariate monitoring techniques such as T^2-statistics and Q-statistics, if the process is found to be in abnormal region, further analysis needs to be done on identifying those variables due to which the process is in abnormal condition. For on-line monitoring we need to identify the variables which are causing the process to go in abnormal region as early as possible to maintain the product quality and safety of the process plant. One of the main methods for identifying the fault is with the help of contribution plot. By identifying the time instance at which the T^2-statistics or Q-statistics violates its corresponding control limits, the contribution plot at that time instance is helpful in determining the variables that are affected due to the fault.
Let us consider a new set of observation vector $\mathrm{x}_{new,k}$ at time instance k having dimensions $1 \times m$. At the time instance k where the control limit is violated, the contribution plot for the T^2-statistics is given by,

$$c_j(T_k^2) = t_{new,k}\, \mathbf{S}_k^{-1}\, [\mathrm{x}_{new,kj}.p_j^T]^T \tag{2.28}$$

where, $c_j(T_k^2)$ denotes the contribution of the j^{th} variable on the T_k^2 i.e on the T^2-statistics at the k^{th} instance. $\mathrm{x}_{new,kj}$ denotes the j^{th} element in $\mathrm{x}_{new,k}$, and p_j^T corresponds to the j^{th} row of the principal component matrix $\mathbf{P}(m \times R)$.
The contribution plot for the Q-statistics can be given by,

$$c_j(Q_k) = e_{kj}^2 \tag{2.29}$$

where e_{kj} denotes the j^{th} variable of $e_{new,k} = \mathrm{x}_{new} - \hat{\mathrm{x}}_{new}$

PCA is the widely used concept for process monitoring in most of the process plant. It is easy to construct with the availability of large amount of process data. And it can capture the maximum variance in the data. There various methods by which we can incorporate PCA to monitor batch process. One such method is Multiway PCA(MPCA), which is discussed in the next chapter.

Chapter 3

Batch Process monitoring using Multiway PCA

3.1 Introduction

Batch process is a product manufacturing technique in which the product that is manufactured is processed in different stages over different stations/assembly. Usually there are three ways in which a product is manufactured:

- *Custom production* are the products manufactured according to customer's specifications. Example of such processes are furniture industry, architectural construction etc.
- *Mass production* are the products manufactured continuously without any pause. Example of such processes are power-plants, oil refineries etc.
- *Batch production* are the products manufactured in sequential steps and there is some time gap in different batches of product. Example of such processes are pharmaceutical drugs, paint and dyes etc.

According to [18] batch process is defined as a finite-duration process having three main steps:

i. Charging the batch vessel with the pre-defined ingredients of the product.
ii. Processing the ingredients under prescribed condition during which process variable such as temperature, pressure, concentration are varied timely according to the prescribed procedure.
iii. Removing the end product from the batch vessel.

Batch processes are important to monitor firstly to ensure the safety of process plant and secondly to produce good quality raw materials. If the process goes out of control it could result in degraded quality of the product which could eventually result in the rejection of the particular batch.
The main advantages of batch plant are batch production has low initial setup cost due to the fact that same production line can be used to produce different product. Another advantage is that during shortage of demand we can adjust the production rate or stop the production according to need, thus avoiding wastage. The main drawback related to batch processes is that after completion of each batch there is a time lag between two consecutive batches known as *idle time* in which we have to again reset the process from beginning which affects the productivity of the process plant.

3.2 Different approaches for monitoring batch process

Batch process monitoring is still a challenge problem for monitoring and controlling the process. The major difficulties faced in monitoring and controlling the batch processes are non-availability of on-line sensor for measuring process variables, non-linearity in the actual process, finite duration of batch processes, and constructing an exact model which takes into account all the thermodynamic relations.

For manufacturing the same product with same initial conditions, the product quality may differ from one batch to another due to the fact that some of process variables deviate from their mean trajectory during the run, or due to disturbances caused by the variation in basic ingredients in charging the batch or due to error by environmental conditions. So, the product quality can degrade drastically if the problem is not detected, which can lead to the cancellation of batch.

Most of the time it is seen that batch processes work in open loop condition due to the fact that very less on-line sensors are available for measuring the product quality variables. At the end of batch process we take random samples from the batch and check for the quality variables and if some rectification is required we do it in the next batch to improve the quality. Generally most of the industrial batch plant heavily rely on precise sequencing and documentation of the process for maintaining consistent product quality from batch to batch. Monitoring in such cases is mainly done by checking the reactor variables such as pH, temperature of the reactor is following the mean trajectory or not.

Some of the methods for monitoring batch process is briefly described below.

3.2.1 State estimation based monitoring

This method is mainly relies on mathematical model of the batch process along with on-line measurements for iteratively estimating the on-line states of the batch process. We construct the state estimator such as Kalman filter or observers with the help of mathematical model obtained during normal operating condition for estimating the states and output of the process and then monitoring the residual error for detecting the fault. The basic construction of observers and monitoring using the state estimation approach is same as described in section(1.3.1) and (1.3.3).
Another way of employing model based approach to batch monitoring is by incorporating the knowledge of stochastic disturbance such as parameter variations and noise into the state model. Due to the additional knowledge which has been added to the model, the state estimator is able to provide robust and unbiased estimation of the states. Then we can monitor the batch process by keeping a track of the important states and we can compare the estimated state trajectory with mean trajectory. The advantage and dis-advantage of mathematical model based approach is described in section(1.3).

3.2.2 Qualitative model or knowledge-based approach

In qualitative model or knowledge based approach we employ artificial intelligence and expert system for processing the batch data. As in section(1.4) we have seen that the process model is basically qualitative equations which have been derived from the understanding of the physics and chemistry of the process obtained from engineers and operators. Additional analysis can be done by combining probability theory or fuzzy logic with expert system to highlight the modeling uncertainties and

the hypothesis set of the proposed faults. This method does not requires rigorous mathematical modeling as compared to the state estimation based monitoring but the main drawback is that by combining expert system with probabilistic or fuzzy logic is complex algorithm and time consuming.

Another approach which has been studied and employed for monitoring batch process is with the help of neural networks with different range of nonlinear functions such as Gaussian, sigmoidal and wavelets which have produced results which demonstrated them as a good pattern classifier hence capable of detecting faults. But the main drawback for this approach is we require a huge amount of data so that we can classify each faulty conditions which is not available for a real process.

3.2.3 Multivariable statistical approach

Statistical Process Control(SPC) charts such as Shewhart control chart, Cumulative Sums(CUSUM) charts, Exponentially Weighted Moving Average(EWMA) charts can be used to monitor very few number of batch processes. These charts are not suitable for processes having correlated process variables. Another drawback which is encountered when employing these charts are due to the dynamic nature of batch process. Most of charting techniques uses the process variable data available at the end of the process, therefore we can monitor only the variation between different batches.

Most of the batch process generates huge amount of data during the batch run. Process variables such as temperature, pressure, concentration etc. can be easily measured. In most of the batch processes there can be roughly 20 or more process variables which are obtained every few seconds throughout the batch duration. Moreover, we have the batch data for the past batches, which contains the data for normal batch and faulty batches. From these data we try to classify the current batch if it is running in normal condition or in abnormal condition. And we try to develop a range of nominal values for which we can classify the data under normal operating region.

For such a large data it is quite difficult to monitor it using the charting techniques, moreover the variables in the data are autocorrelated in time and the variables are correlated to each other at any given instant. And the relationship among the variables for the past batches are also important for classifying. So for monitoring such large amount of data we employ PCA for monitoring and compressing the data. As we have seen in chapter(2) that PCA project the data into lower dimensional space while retaining the maximum variance in the data. So we incorporate PCA to batch process, with the help of historical batch data we can generate range of nominal trajectories for the process variables and with the help of which we can comment on the current batch whether it is in faulty condition or in normal condition.

3.3 Multiway Principal Component Analysis

In this section we will see the application of Multiway PCA(MPCA) to monitor the batch process data. The data of the batch process is arranged as shown in figure(3.1) Let the batch process data is denoted by a three-way matrix $\underline{\mathbf{X}}$ having the dimension $I \times J \times K$ where I is the number of batch, J is the number of variables and K is the number of samples. In the three-way array $I \times J \times K$, each horizontal slice($J \times K$) as shown in figure(3.1) represent the individual batch trajectories of all the variable of single batches. And the vertical slice($I \times J$) represent the value of all the variables at each time instant.

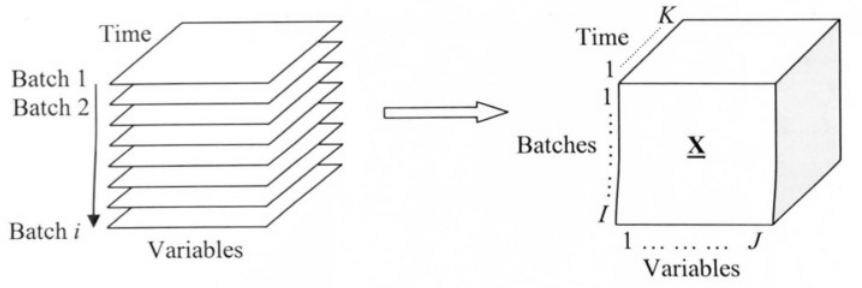

Figure 3.1: Structure and arrangement of batch data in a three-way array[28]

We can perform PCA on three-way matrix $I \times J \times K$ by unfolding each of the vertical slice and placing them horizontally one after the other to form a two dimensional matrix($I \times JK$) as shown in figure(3.2). This type of unfolding is referred as time-wise unfolding. This type of unfolding is

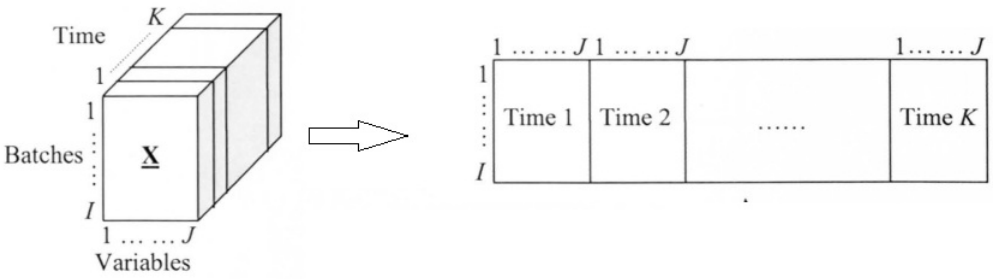

Figure 3.2: Unfolding of three-way array $\underline{\mathbf{X}}$ [28]

advantageous because we can study the simultaneous variation in the process variable with respect to each time instance.In simpler way, we can compare the process variable of each with all the other batches at that single instance in time. There are also other two types of unfolding namely, batch-wise and variable-wise unfolding.

The main aim of MPCA is to disintegrate the three-way matrix into two parts, the first part containing the series summation principal components comprising of score vectors(t_r) and and loading matrices ($\mathbf{P}_r$) and the second part containing the three dimensional residual error denoted by $\underline{\mathbf{E}}$ as shown in figure(3.3). Mathematically it can be represented as,

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} t_r \otimes \mathbf{P}_r + \underline{\mathbf{E}} \tag{3.1}$$

where, R denotes the number of principal components. The methodology for applying MPCA is same as NIPALS algorithm as seen in section(2.3). The steps for applying NIPALS algorithm to three-way array is as follow[29][30]:

i. Pre-filtering of $\underline{\mathbf{X}}$ i.e subtracting each column by its corresponding mean and diving by its

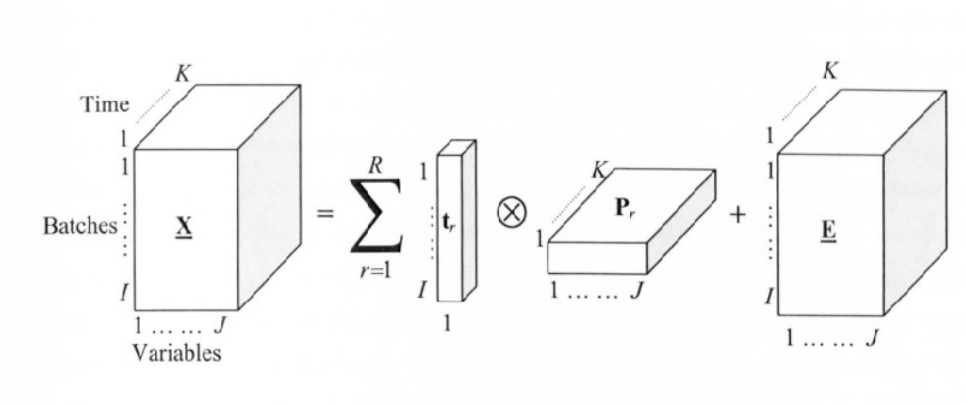

Figure 3.3: Disintegration of batch data $\underline{\mathbf{X}}$ by MPCA[28]

standard deviation.

(a) Mean centering is shown in figure(3.4). Mathematically it can be represented as,

$$\tilde{\mathrm{x}}_{ijk} = \mathrm{x}_{ijk} - \frac{\sum_{i=1}^{I} \mathrm{x}_{ijk}}{I} \quad i = 1, 2, .., I \;\; j = 1, 2, .., J \;\; k = 1, 2, .., K \tag{3.2}$$

Where, $\tilde{\mathrm{x}}_{ijk}$ is the mean centered vector. By subtracting the mean of each column essentially we are centering the variable on its mean trajectory thereby removing the nonlinearity in the data.

(b) Standardization of data, as in figure(3.4) instead of mean we have the standard deviation of each column and we divide the mean centered vector by its corresponding standard deviation. Mathematically, it can be written as:

$$\mathrm{x}^{*}_{ijk} = \frac{\tilde{\mathrm{x}}_{ijk}}{\sqrt{\frac{\sum_{i=1}^{I} \mathrm{x}_{ijk} - \bar{\mathrm{x}}_{\mathrm{jk}}}{I-1}}} \tag{3.3}$$

where, x^{*}_{ijk} is the mean centered and having unity variance. The standardization of data is done to give equal weightages to the variables at each instant in time.

ii. Unfolding the auto-scaled three-way array $\underline{\mathbf{X}}(I \times J \times K)$ into matrix $\mathbf{X}(I \times JK)$.

iii. Initialize by randomly choosing a column of $\mathbf{X}$ as t.

iv. Then computing $p = \mathbf{X}'t$.

v. Then scaling p to unit length $p = \frac{p}{|p|}$.

vi. Calculating the score vector, $t = \mathbf{X}p$.

vii. Then we check for convergence on t, if t is converged then proceeding to next step orelse go to step(iv).

viii. Updating the residual matrix as $\mathbf{E} = \mathbf{X} - tp'$

ix. Updating $\mathbf{X} = \mathbf{E}$ and go to step(iii) for calculating the next principal component.

3.3.1 Analysis of historical batch data using MPCA

The study of historical batch data which is collected from a batch process from several runs gives us an idea about how the process behaves. The first step is to build a model of the process using

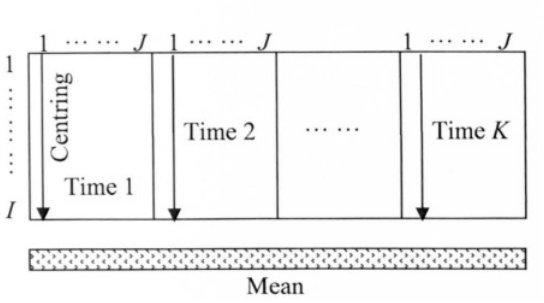

Figure 3.4: Mean centering of the unfolded data[28]

the data of historical batches which were considered good according to the product quality. So by building the model with the help of good batches defines a range of normal operating region for different process variables obtained from the batch to batch variations, which will help us in predicting the future batches whether it is good or not.

Building the model from normal batch runs

1. We collect all the normal batch run and stack them in three-way array as shown in figure(3.1).
2. We auto-scale the data as described in section(3.3).
3. Unfolding the auto-scaled three-way array $\underline{\mathbf{X}}(I \times J \times K)$ into matrix $\mathbf{X}(I \times JK)$.
4. Then we apply PCA on the unfolded matrix to obtain the normal operating data model.
5. Based on the normal operating data we obtain the control limits of the T^2-statistics and Q-statistics.

Off-line monitoring

Suppose we have data for several batch runs, and we want know which of the batches are normal and which one is abnormal, i.e we want to determine the batches in which the product quality variables are well within the nominal range. For that purpose we plot the score plot and the sum of residuals$[Q_i = \sum_{k=1}^{K} \sum_{j=1}^{J} \underline{\mathbf{E}}(i,k,j)^2]$ of each of the batch of the three-way array and using the control limits of the normal batch run for thresholding.

On-line monitoring

In the off-line monitoring we are able to monitor the batches once the batch run is complete, i.e at the end of the batch. It will be more helpful if we can monitor a batch which is ongoing and can monitor it all the way to the end of batch. The main advantage of on-line monitoring is that we can take corrective actions to improve the product quality variables and consistently produce products having similar quality. But the main problem with on-line monitoring as pointed out in([30]) is that we have we have data for the current batch from starting till the current point, so we do not have values for the future time. There are three ways in which we can fill the missing values[30],

1. **Filling with zero**

   As in figure(3.5) we will fill the missing observation in x_k with zeros. This type of filling assumes that the future values will follow the trajectory of the nominal batch operation. In this case we assume that if there is some deviation in measurement, with time it will come back nominal range.

2. **Filling with current value**

   This is the most widely accepted approach. In this method we assume that the future value will continue to deviate at the same rate as in present time. Here we will assume the missing observations in x_k with the current value. Here the assumption is that if there is a fault occurring in the process it will continue to affect the system till the batch ends.

3. **Missing data method**

   In this method we assume that observations in x_k are considered as missing values. The scores and residual for monitoring is computed by using the loading up to time k. This method exploits the ability of PCA to handle missing data. The loading matrix continues to grow as the batch proceeds with time. When the current observation is available we update it in the past measurement as shown in figure(3.5). The scores denoted by t_k is given by equation,

$$t_k = x_k \, \mathbf{P_k} (\mathbf{P}_k^T \mathbf{P}_k)^{-1} \tag{3.4}$$

   where, x_k is the observation vector containing data upto k^{th} time and $\mathbf{P}_k$ denotes the principal components containing the first k rows.

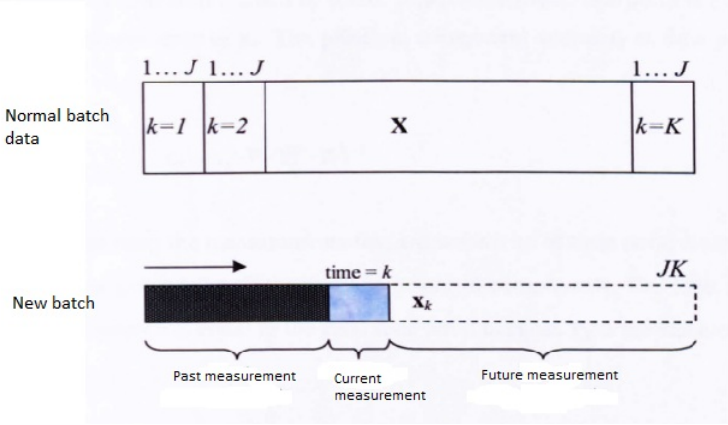

Figure 3.5: On-line monitoring of batch data[28]

Methodology for on-line monitoring

1. First step is constructing the three-way matrix $\underline{\mathbf{X}}(I \times J \times K)$ as shown in figure(3.1) which contains data of various batches considered as normal.

2. Then we unfold it time wise and horizontally place them one after another as shown in figure(3.5) into a two dimensional matrix $\mathbf{X}(I \times JK)$.
3. Then we auto-scale the data as shown in figure(3.4) and we save the mean vector and standard deviation vector for each vertical slice.
4. And calculate the PCA model the principal components denoted by $\mathbf{P}_r$ for the nominal data.
5. Then we calculate the control limits for T^2-statistics and Q-statistics which is discussed in appendixC.2.

Now for on-line monitoring of a new batch data denoted by $\mathbf{X}_{new}(K \times J)$, we calculate the score vector t_k and square prediction value(SPE_k) at each time instant k using the following procedure:

1. For a new batch of data at time k the observation vector is denoted by $\mathrm{x}_{new}(1 \times J)$.
2. Then from x_{new} we subtract the mean vector and divide by the standard deviation vector corresponding to the k^{th} time instant which is obtained from the normal data to get the vector with the current deviations.
3. Then we fill the rows of $\mathbf{X}_{new}$ from k^{th} instant to K with the current deviation vector, i.e the observation vector after auto-scaling.
4. We calculate the score matrix and and the SPE at the k^{th} instant using the equation

$$\begin{aligned} t_{r,k} &= \mathbf{X}_{new}\, \mathbf{P}_r \\ \mathbf{E} &= \mathbf{X}_{new} - \sum_{r=1}^{R} t_{r,k}\mathbf{P}_r \\ SPE_k &= \sum_{j=1}^{J} \mathbf{E}(k,j)^2 \\ T_k^2 &= t'_{r,k}\mathbf{S}^{-1}t_{r,k} \end{aligned} \tag{3.5}$$

5. Then at $k+1^{th}$ instant, we update the k^{th} row $\mathbf{X}_{new}$ with the actual observation vector obtained and return to step1, and we recursively continue the method till $k = K$.

And we plot the T_k^2 and SPE_k with the confidence limit which has been obtained from normal operating data. Where S is the covariance matrix of the principal scores which is obtained from the normal operating data.

There are two ways in which $\mathbf{X}_{new}$ can deviate from normal MPCA model. In the first case the scores calculated may violate the nominal variations which can be seen in T_k^2 or the residual generated $\mathbf{E}(k,j)$ is large due to variation from the nominal values and the batch will be placed outside, orthogonal to the reduced space. In the first the model will be valid but only but the value of variation will be large enough. But in the second case the model is invalid because of the occurrence of a new event which has not been recorded in the normal data set, and the batch data cannot be projected in reduced space so the SPE value will be large enough to violate the threshold calculated with the help of nominal data.

The main drawback of MPCA approach for monitoring batch process is that for monitoring a new batch, we have to fill the future values of the batch till the end of the process. Three different methodologies we have seen in the previous section. But the estimated values affects the dynamic relations among variables which causes false alarm. There is also another method in which the future values are selected from a library containing previous batch data. In this approach we select future values of the current batch by measuring similarity of each batch in library and filling the future

values with the batch value that is most similar to the current batch. But this approach requires extensive construction of the batch library. And then searching through the database to search for a similar batch which is very time consuming and constructing such extensive library requires rigorous analysis. Another main drawback in batch process monitoring by MPCA method is that, we require equal batch length for constructing the normal batch three-way array, which is not practical in real process.So a more efficient method is addressed in the next section.

3.4 Enhanced MPCA

This method is mainly focused on solving the drawback of MPCA methodology. It takes advantage of both batch-wise unfolding and variable-wise unfolding as shown in figure(3.5) The basic methodology

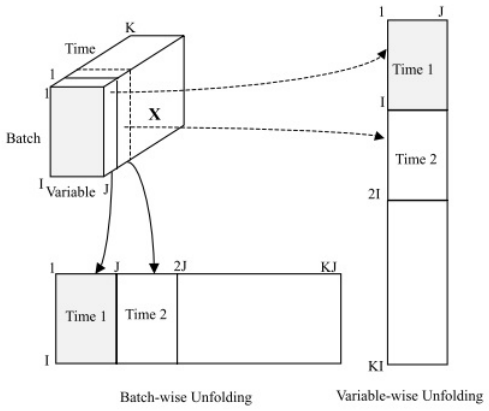

Figure 3.6: Approaches of unfolding batch data[31]

can be summed up as follows[31]:

A. Constructing the normal operating data model
   (a) Unfolding the three-way array $\underline{\mathbf{X}}(I \times J \times K)$ to a matrix $\mathbf{X}(I \times JK)$.
   (b) Pre-filtering the data $\mathbf{X}$ by subtracting each column with its corresponding column mean and dividing by column standard deviation.
   (c) Rearranging the scaled data data $\mathbf{X}(I \times JK)$ to $\mathbf{X}_{SC}(KI \times J)$.
   (d) Calculating the score matrix $\mathbf{T}(KI \times R)$ and loading matrix $\mathbf{P}(J \times R)$ by doing PCA on $\mathbf{X}_{SC}$.
   (e) We divide the score matrix $\mathbf{T}(KI \times R)$ into $\mathbf{T}_k(I \times R)$ for each time instant k.
   (f) We calculate $\mathbf{S}_k(R \times R)$ the covariance matrix of $\mathbf{T}_k$ for each time k. This is the main difference from the conventional MPCA method where we assume the covariance matrix to remain constant throughout the time due to which the T^2-statistics was not dynamic.
   (g) Calculating the control limits of T^2 and Q-statistics at each time k.Refer appendix(C.2).

B. On-line monitoring of a new batch
   (a) For a new batch at time k, the observation vector is denoted by $\mathrm{x}_{new,k}(1 \times J)$ and preprocessing the vector using the mean and standard deviation corresponding to normal batch data at k^{th} instant which was used for modeling.
   (b) Calculating the new score vector by projecting $\mathrm{x}_{new,k}$ on the loading matrix $\mathbf{P}$ calculated

from the model,

$$\mathrm{t}_{new,k} = \mathrm{x}_{new,k}\ \mathbf{P} \tag{3.6}$$

(c) Calculating the T^2 and Q-statistics at each time instant k,

$$T_k^2 = \mathrm{t}_{new,k}\ \mathbf{S}_k^{-1}\ \mathrm{t}_{new,k}^T \tag{3.7}$$

$$Q_k = \mathrm{e}_{new,k}\ \mathrm{e}_{new,k}^T \tag{3.8}$$

where $\mathrm{e}_{new,k}$ is given by

$$\mathrm{e}_{new,k} = \mathrm{x}_{new,k} - (\mathbf{P}\ \mathbf{P}^T\ \mathrm{x}_{new,k}^T)^T \tag{3.9}$$

(d) Then we monitor the T_k^2 and Q_k with respect to the control limit obtained for each time k from the normal data model.

(e) If at any point k the T_k^2 or Q_k violates the control limit then we try to identify the variables that is causing the fault in other words the variables that is most affected by the fault.

C. Contribution plots.

(a) The variables that are contributing in the fault for T_k^2,

$$c_j(T_k^2) = t_{new,k}\ \mathbf{S}_k^{-1}\ [\mathrm{x}_{new,kj}.p_j^T]^T \tag{3.10}$$

(b) The variables that are contributing in the fault for Q_k,

$$c_j(Q_k) = e_{kj}^2 \tag{3.11}$$

The following procedure has various advantage in on-line monitoring, we can cancel out the nonlinear dynamics of the data by subtracting it from the mean of the normal batch data. The loading matrix remains same fro the entire trajectory but the score matrix contains the dynamic information along different batches in different time instant. And in this method the covariance matrix is time varying so it contains the major dynamic features of the batch process. So, when a new data is projected onto the reduced space of the model, the T^2-statistics can detect the changes in the score mean as well as the change of correlation structure among variables. In this method we do not require to fill the future value.

3.5 Results and discussion

All the methodology discussed above is used to monitor the fermentation of acetone-butanol batch. We simulated the process[32][33] using mathematical model of the process plant which is discussed in appendix(B) for a defined normal and faulty conditions[34]. The model is capable of explaining the biochemical and physiological characteristics of growth and metabolic synthesis. The details of the model variable and parameters are shown in Table(3.1).

Table 3.1: Model variables and parameters for ABE batch process

Variables/Parameters	Details	Sampling time
y	Cellular RNA concentration	Not measured
X	Reactor cell concentration	30 mins
S	Reactor substrate concentration	1 min
BA	Reactor butyric acid concentration	1 min
AA	Reactor acetic acid concentration	1 min
B	Reactor butanol concentration	1 min
A	Reactor acetone concentration	1 min
E	Reactor ethanol concentration	1 min
$C0_2$	Reactor carbon dioxide concentration	1 min
H_2	Reactor hydrogen concentration	1 min

3.5.1 Historical database for normal operating model

To simulate the four abnormal condition we vary the parameter values and initial condition from batch to batch randomly. We have simulated the normal condition by varying the parameter and initial condition under a nominal range of value. The variation is shown in Table(3.2). A single batch is simulated for 30 hours. And each of the abnormal conditions along with normal condition is simulated for 100 batches. So a total of 500 batches are simulated. The data is then corrupted with Gaussian noise with signal to noise ratio of 10 for a normal batch run. The batch data for all

Table 3.2: Various mode of ABE fermentation

Mode	Details	Normal parameter value	Parameter ranges
1	Normal batch	$y(0) = 1$ $X(0) = 0.03$g/L $S(0) = 50$g/L	$0.9 \leq y(0) \leq (1.1)$ $0.01 \leq X(0) \leq 0.05$g/L $45 \leq S(0) \leq 55$g/L
2	Slow substrate utilization	$K_S = 40$g/L	$30 \leq K_S \leq 50$g/L
3	Increased sensitivity to butanol	$K_I = 0.425$g/L	$0.25 \leq K_I \leq 0.6$g/L
4	Decreased sensitivity to butanol	$K_I = 1.27$g/L	$1.11 \leq K_I \leq 1.42$g/L
5	Dead inoculum	$y(0) = 0.075$g/L $X(0) = 0.003$g/L	$0.05 \leq y(0) \leq 0.1$g/L $0.001 \leq X(0) \leq 0.005$g/L

the five condition including the normal operating condition and four faulty conditions is shown in figure(3.8). And from figure(3.7) we can see how each variables differ from the normal condition in the faulty case. We can see that every measurement is having unit g/L and the unit of time is minute. There are four faulty cases which affect the A-B-E fermentation process:

- *Decreased sensitivity to butanol*: The concentration of butanol plays a vital role in A-B-E fermentation process. The butanol mainly affects the pH concentration of the fermentation broth. With low concentration of butanol pH increases which inhibits the process.
- *Increased sensitivity to butanol*: With the increase in butanol concentration the pH level drastically drops below 6.2 which is the optimum pH level for the fermentation to take place.
- *Dead inoculum*: The dead inoculum mainly refer to the *Clostridium* bacteria which is helpful in fermentation process are dead due to pH or temperature changes of the fermentation broth.
- *Slow substrate utilization*: In slow substrate utilization, the substrate which is sugar containing

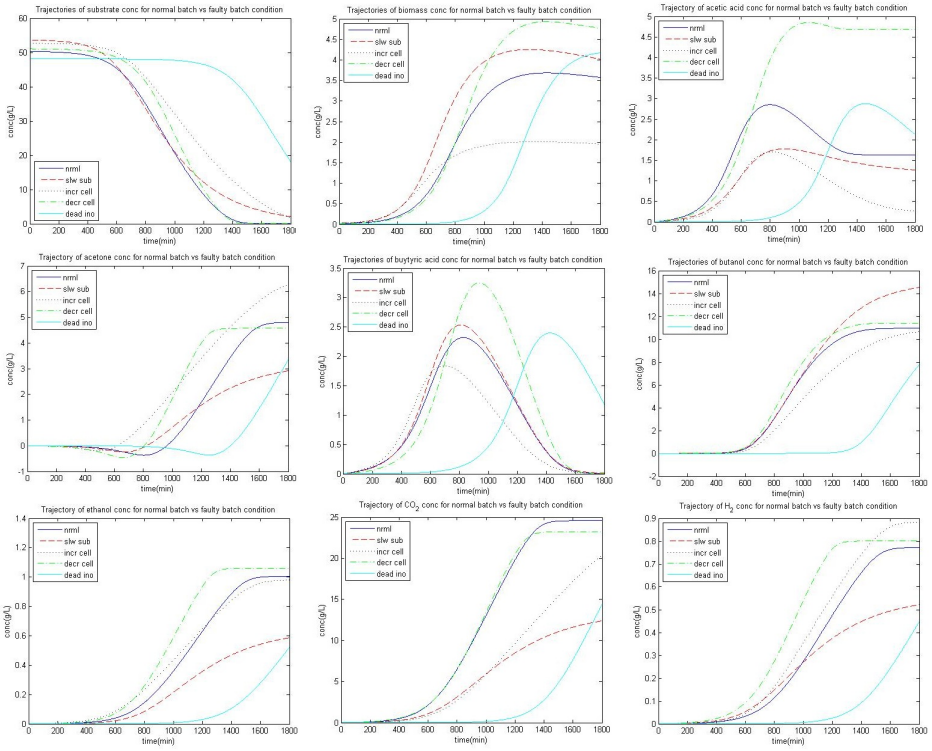

Figure 3.7: Each of the measurement variables of normal batch vs faulty batches

equal amount glucose and fructose, the glucose is consumed faster than fructose due to which in the later stage of process the sugar consumption is less.

For our case study we have simulated the normal process 100 times to construct the normal MPCA model for detecting faults for a new batch of data.

3.5.2 Analysis of historical batch data using MPCA

For post batch analysis, we have selected only 50 normal batches randomly out of the simulated 100 normal batches and randomly selected 3 batches from each of the four faulty conditions. The batch run is for 30 hours, measurement taken every 1 minute interval. The three way array constructed from 50 normal batches and 12 faulty batches, having dimension $62 \times 10 \times 1800$. The projection of these 62 batches in the reduced score space taking three principal component is shown in figure(3.9). where we can see that the 12 faulty batches do not cluster with the 50 good batches, which indicates that there was any deviation from the normal operating conditions. We can also conclude that batches having similar attributes cluster with each other as we can see in figure(3.9) the batches having same kind of fault clusters with each other. So, rom this we can conclude that MPCA is able to differentiate between normal and abnormal batches through the score plot. The SPE of each of the 62 batches is shown in the figure(3.10) which can be calculated as follows:

$$Q_i = \sum_{k=1}^{K} \sum_{j=1}^{J} \underline{\mathbf{E}}(i,k,j)^2 \tag{3.12}$$

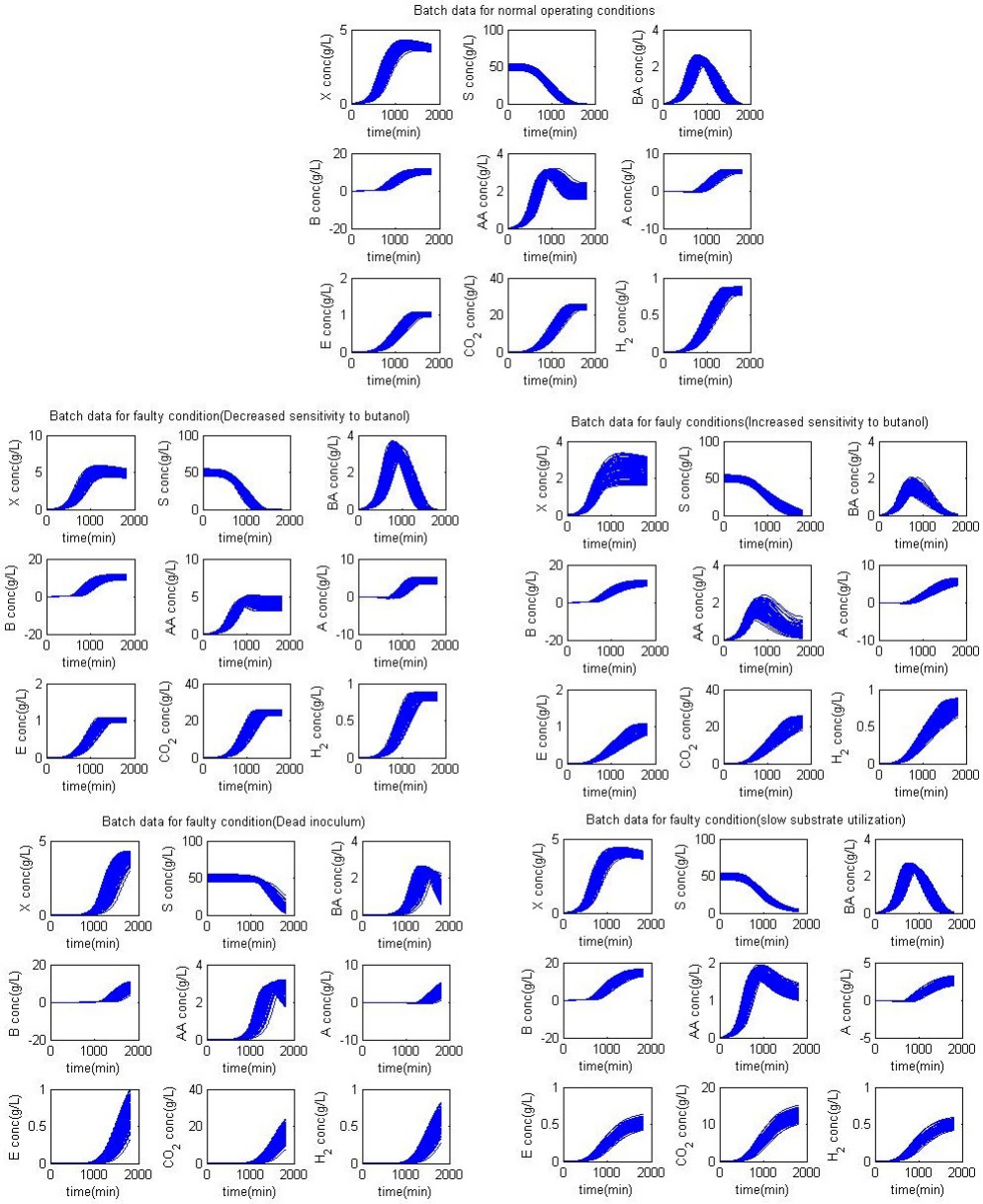

Figure 3.8: Batch profile for normal condition and four faulty conditions

Where calculation of $\underline{\mathbf{E}}$ has been discussed in the section(3.3.1). We can see that most of the faulty batches violates the control limits, some of the normal batches violate the 95% control limits but do not violate the 99% control limit. Basically the Q_i values specifies the perpendicular distance of the respective batch in the reduced space constructed from the three principal components with the help of MPCA. So from the above results we can conclude that MPCA is able to differentiate adequately between normal and abnormal batches.

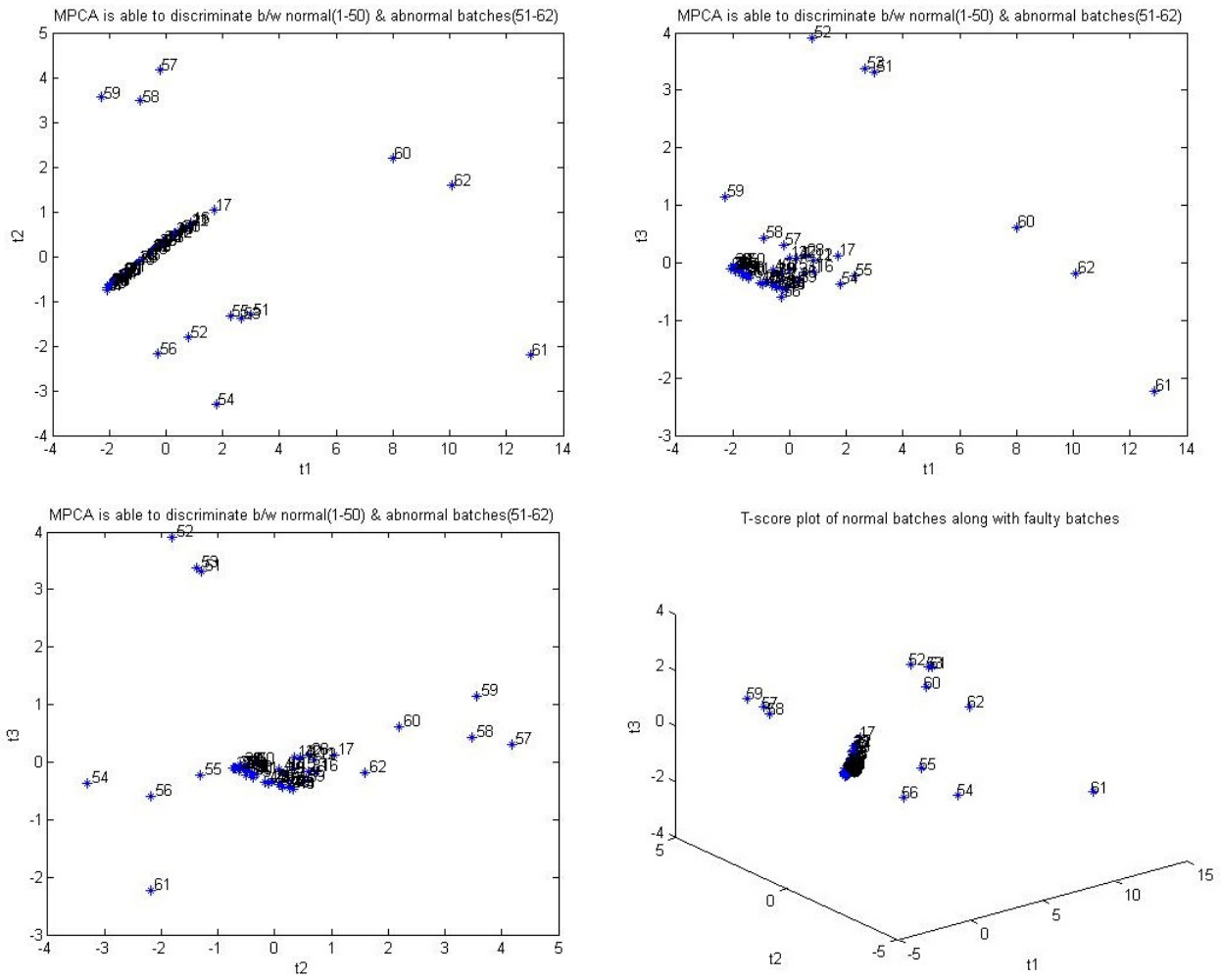

Figure 3.9: T-score plot of the 62 different batches.

3.5.3 On-line monitoring with MPCA and enhanced MPCA

In the previous section we were able to segregate the normal and abnormal batches from the historical process data available from the respective batches. In this section we will be commenting about the on-going batch whether it is faulty batch or normal batch and will be comparing the results from the two discussed methods, MPCA and Enhanced MPCA. We constructed the nominal model using the 100 normal batches and then checking the model with a normal batch data and then subsequently followed with four faulty batches. Let us first discuss the on-line monitoring of a normal batch for which the results can be seen in the figure(3.11). From the results we can see that SPE plot of MPCA method shows slight violation of the control limit from approximately 800 minutes to 1400 min whereas the T^2 plot initially violates the threshold but after that there is no violation, this due to the fact that initially we start filling the batch data using the current deviation values. For the Enhanced MPCA method we can see that the SPE chart and T^2 chart the control limit is not

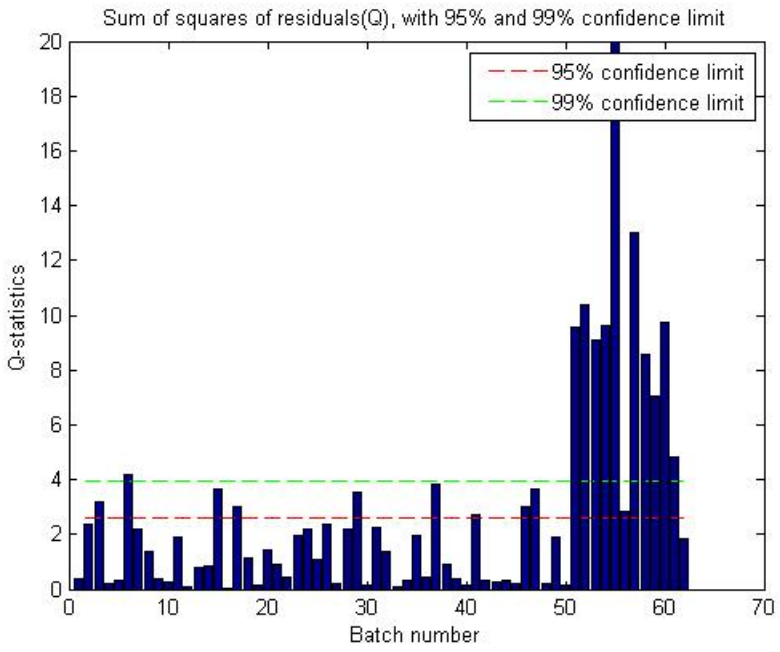

Figure 3.10: SPE(Q) of 62 batches with 95% and 99% confidence limit.

violated during the total batch run. So the Enhanced method is better from the simulation results.

Now let us take the first faulty condition of slow substrate utilization. The simulation result is shown in the figure(3.12). We can see that the SPE of MPCA is constantly indicating a fault after the 176 minutes and the values fluctuates whereas in the T^2 is able to detect fault in the batch after 1356 minutes and again returns to normal condition during the end of the batch. But in Enhanced MPCA method the SPE shows the fault starting from 789 minutes constantly violates the threshold and the T^2 violates the threshold at 1329 minute and continues to violate the threshold indicating a presence of faults. MPCA was able to detect the fault at an very early stage with the help of SPE but it was fluctuating between normal and abnormal region whereas in enhanced MPCA the detection time was more but it constantly violated the threshold. So we can conclude that both the method were advantageous in different aspects.

Now let us move to the next faulty condition of increased sensitivity to butanol. From the figure(3.13) we can see that MPCA method was able to detect fault at 789 minute with the help of T^2 statistics whereas with the help of SPE we were able to detect fault at 200 minutes with some false alarming whereas in Enhanced MPCA method the results are better than MPCA.

Now let us move to the next faulty condition decreased sensitivity to butanol. The simulation result is shown in the figure(3.14). Initially we can see that the SPE and T^2 of MPCA indicates a fault but the SPE is more reliable in indicating a fault from 729 minute where as in the enhanced MPCA method the T^2 starts to grow from 700 minute but it violates the threshold after 1200 minute same as the MPCA method whereas the SPE violates the threshold at 732 minutes and it continues to violate the threshold. So, there is a similarity in detection of T^2 limit but the SPE of both the methods differ in the sense that in MPCA method detection is fast but not reliable where in enhanced MPCA the detection is slow but more reliability is guaranteed.

Now let us move to the fault condition of dead inoculum, for which the results can be seen

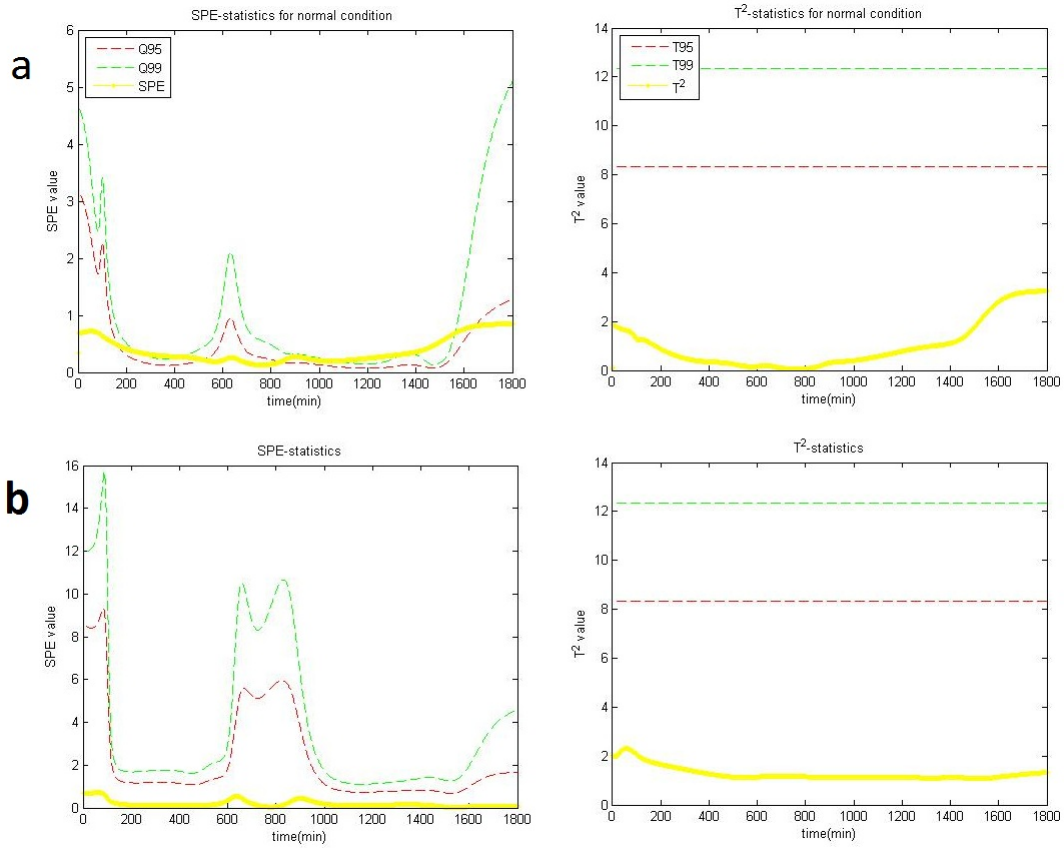

Figure 3.11: Monitoring chart of normal batch (a)MPCA (b)Enhanced MPCA

in figure(3.15). In the dead inoculum case the enhanced MPCA from the beginning of the batch indicated a presence of fault, whereas in the MPCA case the SPE detected the fault after 200 minutes and the T^2 initially indicated a presence of fault but later on it indicated no fault and again violated the threshold after 1300 minutes. For this case enhanced MPCA performed far better than PCA method.

So, from the above discussion we can see that both the methods have advantage and disadvantages, for MPCA method we need to fill the batch to the full length after each time interval and for constructing the nominal model we require same length of batches, whereas in enhanced MPCA we do not require to fill the batch to the full length after each time nor do we require same batch length. But the main disadvantage of this method is that loading matrix is assumed to be constant throughout the batch run which due to which it is not suitable in the process where the relation between variables changes with the time, but for the A-B-E process it was slow in detecting fault for most of the cases but it was reliable than the MPCA method. Another advantage of the enhanced MPCA method was that the covariance matrix captures the dynamics of each score at each time interval whereas in MPCA it remained same throughout the process.

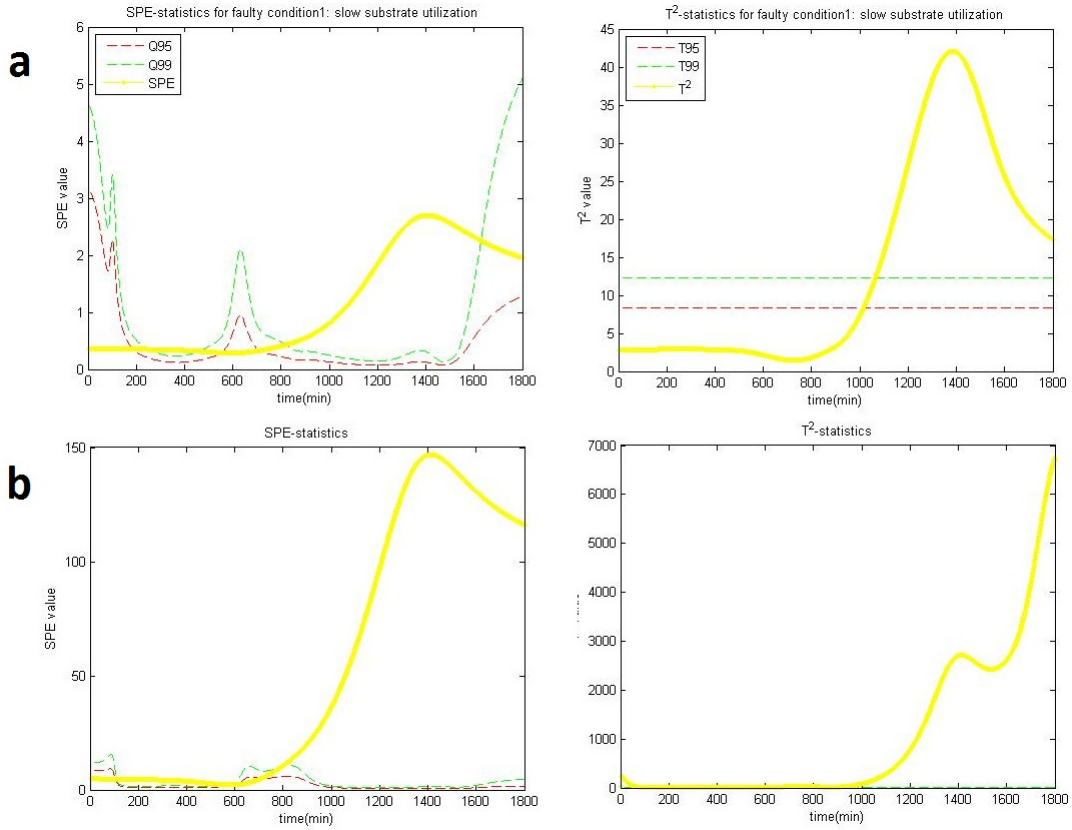

Figure 3.12: Monitoring chart of the (a)MPCA method and (b)Enhanced Method for a faulty batch:Slow substrate utilization.

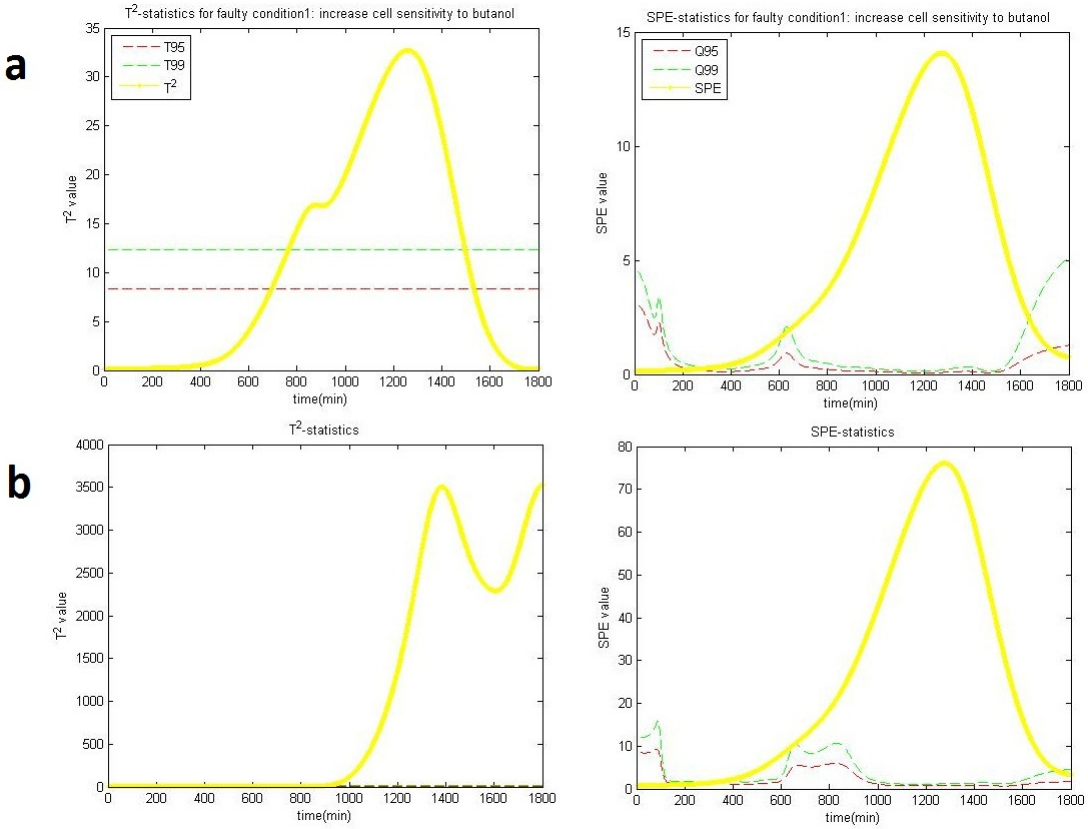

Figure 3.13: Monitoring chart of the (a)MPCA method and (b)Enhanced Method for a faulty batch:increased sensitivity to butanol.

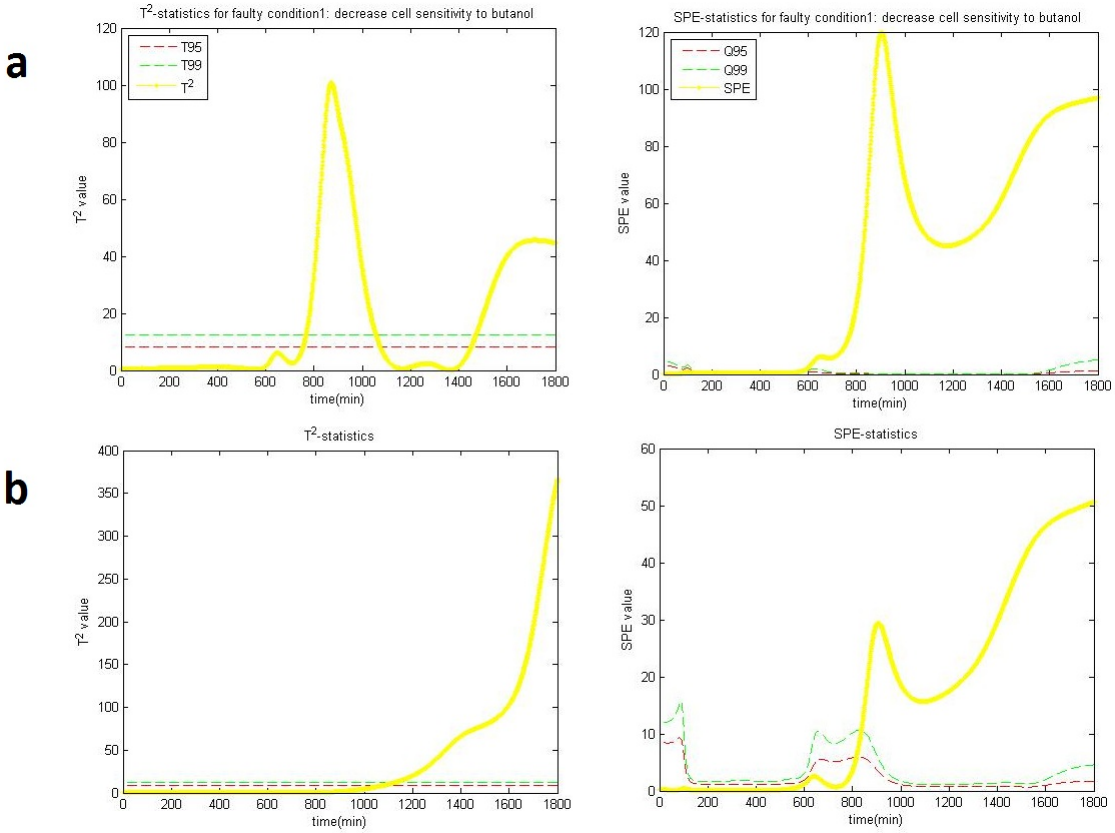

Figure 3.14: Monitoring chart of the (a)MPCA method and (b)Enhanced Method for a faulty batch:Decreased sensitivity to butanol.

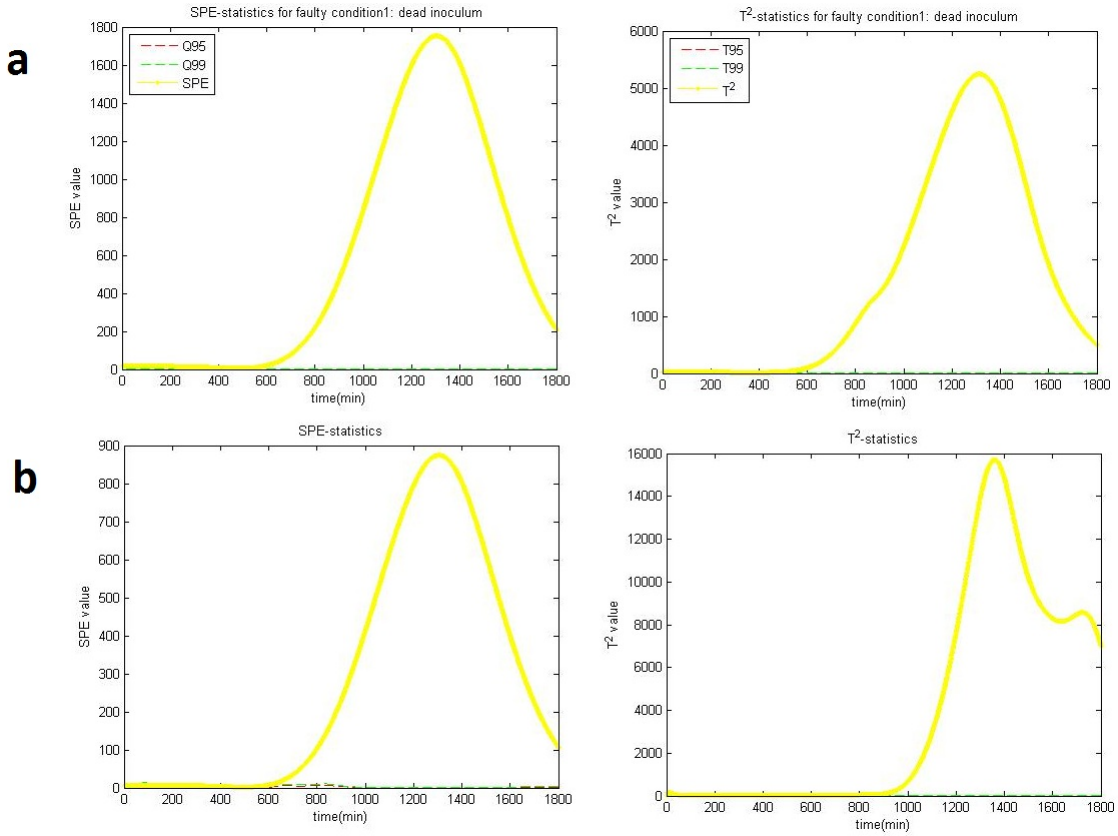

Figure 3.15: Monitoring chart of the MPCA method and Enhanced Method for a faulty batch:Dead inoculum.

Appendices

Appendix A

Different types of models

In this section we are going to discuss about various types of models such as state-space model and input-output models which are mainly used in quantitative model-based fault classifier.
The main task of fault diagnosis is to identify the current state of the process based on its behavior, and the behavior is monitored with the help actuator input and sensor outputs. In most of the cases in FDD we use discrete black box models like state-space model or input-output models and the main assumption that is made in modeling is that, the process plant is assumed to be linear, if any process plant is nonlinear in nature then, first we have to linearize the plant around the operating point. Another important point to note here is that all dynamic process are continuous but all the instruments such as sensors give sampled data, due to which we have discussed only discrete time model.

Considering a system has m inputs and k outputs. Let $u(t) = [u_1(t)...u_m(t)]^T$ be the input vector and $y(t) = [y_1(t)...y_k(t)]^T$ be the output vector, where t denotes the time in discrete domain. Then the basic state-space model is represented as[5]

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) \end{aligned} \tag{A.1}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are parameter matrices with proper dimensions and $x(t)$ denotes the state matrix having dimension $n \times 1$.
The other form in which the same system could be explained is the input-output form

$$\mathbf{H}(z)y(t) = \mathbf{G}(z)u(t) \tag{A.2}$$

where $\mathbf{H}(z)$ and $\mathbf{G}(z)$ denote the polynomial matrices in z^{-1} which is known as backward-shift operator, where $\mathbf{H}(z)$ is diagonal. The structure of $\mathbf{H}(z)$ and $\mathbf{G}(z)$ are as follows

$$\begin{aligned} \mathbf{H}(z) &= \mathbf{I} + \mathbf{H}_1 z^{-1} + \mathbf{H}_2 z^{-2} + ... + \mathbf{H}_n z^{-n} \\ \mathbf{G}(z) &= \mathbf{G}_0 + \mathbf{G}_1 z^{-1} + \mathbf{G}_2 z^{-2} + ... + \mathbf{G}_n z^{-n} \end{aligned}$$

The input-output model is related to the state space model as

$$\mathbf{S}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{A.3}$$

which implies,

$$\begin{aligned} \mathbf{G}(z) &= \mathbf{C}adj(\mathbf{I} - z^{-1}\mathbf{A})z^{-1}\mathbf{B} + \mathbf{D} \\ \mathbf{H}(z) &= [det(\mathbf{I} - z^{-1}\mathbf{A})]\mathbf{I} \end{aligned} \tag{A.4}$$

Most of the time input-output model is directly obtained from physical consideration and process identification rather than obtaining it from state-space model.
Both the process model (A.1) and (A.2) have defined the ideal situation, i.e fault free and noise/disturbance free environment. For incorporating fault in the above models, they are modeled as[5]

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{E}p(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + \mathbf{E}'p(t) + q(t) \end{aligned} \tag{A.5}$$

where we assume that our input sequence $u(t)$ and output sequence $y(t)$ are both observable, $p(t)$ denotes actuator faults, disturbances, plant faults and input sensor faults. $q(t)$ contains output sensor faults. $\mathbf{E}$ is the wieght given to different fault in other words its known as fault entry matrix. The input-output model with noise is as follow

$$\mathbf{H}(z)y(t) = \mathbf{G}(z)u(t) + \mathbf{H}(z)q(t) + \mathbf{F}(z)p(t) \tag{A.6}$$

where $p(t)$ and $q(t)$ is defined same as above.

There are two different kinds of faults when constructing model for a real process, additive faults and multiplicative faults. Both of them changes the model equation (A.2). With multiplicative fault, model equation (A.2) changes as

$$(\mathbf{H}(z) + \Delta\mathbf{H}(z))y(t) = (\mathbf{G}(z) + \Delta\mathbf{G}(z))u(t) \tag{A.7}$$

Multiplicative fault basically changes the plant parameters. The change can be seen in matrices $\mathbf{G}(z)$ and $\mathbf{H}(z)$. These kinds of faults are difficult to handle due to the fact that the coefficient of the plant model are not constant but contains plant variable which changes with time.

On the other hand, with additive fault there are some extra terms which are added in model equation A.2 which do not depend on the observed variables. The model can be represented as

$$\mathbf{H}(z)y(t) = \mathbf{G}(z)u(t) + \mathbf{H}(z)q(t) + \mathbf{F}(z)p(t) + \mathbf{K}(z)\omega(t) \tag{A.8}$$

where $p(t)$ represents actuator faults and $\omega(t)$ denotes unknown disturbances, which includes *unstructured uncertainties* such as parts in plants that are not modeled, unknown faults and measurement noise. All these have been added to the process model as additive faults.

Appendix B

Mathematical Model of a Batch Acetone-Butanol Fermentation

Batch Acetone-Butanol fermentation also known as batch *Clostridium acetobutylicum*. The mathematical model was constructed using experimental data for anaerobic solvent production. The model mainly outlines biochemical as well as physiological feature of growth and metabolite synthesis by production strain. The acetone-butanol-ethanol (A-B-E) fermentation process has been been well known for a long time, it was used for production of solvents. *Clostridium* culture has an intriguing metabolism which includes accumulations of important solvents(acetone, butanol, ethanol), acids(acetic and butyric) accompanied with the release of gases(carbon dioxide(CO_2) and hydrogen(H_2)).With the help of the model we can get help in further explorations of fermentation by

i. With the help of computer simulation we can aid the experimental work more efficiently.
ii. With the help o model we can have a dynamic control over the fermentation process.

The A-B-E fermentation model is based on the following assumption, which are made from knowledge of the fermentation process[33]

- The only limiting substrate in the batch cultivation is glucose.
- Nitrogen source does not inhibit the process.
- The batch culture is mainly hindered by the collection of metabolic product such as butanol, butyric and acetic acid. Their corresponding level production depends on the culture growth.
- Only undissociated for of butyric and acetic acid can re-enter into the cell for conversion to butanol and acetone respectively. Dissociated form of acids collect in the fermentation broth as extracellular products.
- Acetoin, Ethanol and lactic acid are mainly synthesized in the *Clostridium* substrate only.

Formulation of the model

The process-orientated mathematical model of the A-B-E model should be able to depict the biochemical kinetics of the process and the physiological aspects of the culture.
The relationship between specific growth rate μ, and the *metabolic activity functional* denoted by

Q which takes into account the culture history and a variable environment is given by

$$\mu = Y_{X/S}\, Q\, g(S) \tag{B.1}$$

where, $Y_{X/S}$ denotes theoretical, thermodynamically maximum macroscopic yield coefficient and $g(S)$ is a function which depends on the environment. In most of the cases $g(S)$ can be considered as unity when then there is presence of substrate in culture medium and is zero when there is no substrate present in the culture medium. For A-B-E process the value of $g(S)$ can be considered as unity.

The metabolic activity function $Q(t)$ is a complex time dependent function whose value depends on the culture history and the variable substrate consumption rate which changes with different stages as the experiment proceeds. It can be defined as

$$Q(t) = \int_0^\infty f(\xi)\; q\; [S(t-\xi)]\; d\xi \tag{B.2}$$

where $f(\xi)$ denotes distribution function distinguishing the structure of population according to age and q denotes specific substrate consumption rate for a certain age category[32]. Even though $Q(t)$ can be obtained from equation(B.2), it can be also considered similar to variation of concentration of some cell component that is linked to growth rate. For example the intracellular RNA(ribonuclei acid) concentration can be used, as because RNA concentration shows a linear relationship with cell growth rate.

The relation between concentration of intracellular marker i.e RNA and the growth rate μ can be given by equation

$$\mu = \text{const}(RNA - RNA_{min})\; g(S) \tag{B.3}$$

where RNA_{min} denotes the concentration of RNA in the cell at $\mu = 0$.

For the A-B-E process, it is known that the culture growth rate is directly proportional to the substrate(sugar) concentration which is multiplied by term which distinguish the inhibition of culture growth by butanol

$$\frac{d(RNA\ \ X)}{dt} = k_1 S \frac{K_I}{K_I + B}(RNA\ \ X) \tag{B.4}$$

Where the value of k_1 and K_I are specified in table(B.1).

The marker used in the experiment was the dimensionless quantity denoted by y which is given as

$$y = \frac{RNA}{RNA_{min}} \tag{B.5}$$

Therefore, the specific growth which is related to the cellular growth content can be defined as

$$\mu = ay - b \tag{B.6}$$

where a and b are constant values. Therefore we define a parameter λ which can be defined as

$$\lambda = \text{const}\frac{g(S)}{RNA_{min}} = 0.56 \tag{B.7}$$

which characterizes different bacterial culture.

We can express the differential mass for biomass, the cell decay and lysis which is directly proportional to the concentration of butanol(B)

$$\frac{dX}{dt} = \lambda(y-1) - k_2 XB \tag{B.8}$$

$$\frac{dy}{dt} = \left[k_1 S \frac{K_I}{K_I + B} - \lambda(y-1)\right] y \tag{B.9}$$

where all the constant and symbols are mentioned in table(B.1) and table(B.2). The differential mass balance depicting the substrate(sugar) consumption can be written as

$$\frac{dS}{dt} = -k_3 SX - k_4 \frac{S}{K_S + S} X \tag{B.10}$$

The butyrate mass balance for the system can be written as

$$\frac{dBA}{dt} = k_5 S \frac{K_I}{K_I + B} X - k_6 \frac{BA}{K_{BA} + BA} X \tag{B.11}$$

in this differential equation the first term on the right hand side describe sugar substrate is formed from the biosynthesis of of butyrate which is inhibited by butanol. The second term describes how butyrate is converted to butanol.
The differential equation for butanol in the system can be written as

$$\frac{dB}{dt} = k_7 SX - 0.841 \frac{dBA}{dt} \tag{B.12}$$

The constant 0.841 is obtained from the stoichiometric conversion as the ratio of molecular weight of butanol and butyric acid.
The differential equation for acetic acid is very similar to butyric acid, and the equation is given by

$$\frac{dAA}{dt} = k_8 \frac{S}{K_S + S} \frac{K_I}{K_I + B} X - k_9 \frac{AA}{K_{AA} + AA} \frac{S}{K_S + S} X \tag{B.13}$$

Here also like butyrate production, the first term on the right side represents the rate of acetate bio-synthesis and the second term represent the rate of acetate converting to acetone.
The acetone production can be described similar to that for mass balance equation given as

$$\frac{dA}{dt} = k_{10} \frac{S}{K_S + S} X - 0.484 \frac{dAA}{dt} \tag{B.14}$$

The ethanol production rate can be described the equation

$$\frac{dE}{dt} = k_{11} \frac{S}{K_S + S} X \tag{B.15}$$

The rate of production of the fermentation gases H_2 and CO_2 are given by

$$\frac{dCO_2}{dt} = k_{12}\frac{S}{K_S + S}X \tag{B.16}$$

$$\frac{dH_2}{dt} = k_{13}\frac{S}{K_S + S}X + k_{14}SX \tag{B.17}$$

The system of differential equation from equations (B.8)-(B.16) represnts the mathematical model which is efficient enough to describe the dynamic of the acetone-butanol-ethanol process batch fermentation.

Table B.1: Values of different parameter in equations used above and their parametric sensitivity for the A-B-E fermentation

Parameters	**Absolute parametric sensitivities**	**Relative parametric sensitivities**	**Units**
$k_1 = 0.0090$	202827.0	11.54	unit less
$k_2 = 0.0008$	2780.9	0.014	hour^{-1}
$k_3 = 0.0255$	11114.0	1.79	unit less
$k_4 = 0.6764$	277.49	1.187	hour^{-1}
$k_5 = 0.0136$	129034.0	11.097	unit less
$k_6 = 0.1170$	-943.83	0.698	unit less
$k_7 = 0.0113$	-99973.1	7.144	hour^{-1}
$k_8 = 0.7150$	30.349	0.137	unit less
$k_9 = 0.1350$	-63.25	0.054	unit less
$k_{10} = 0.1558$	-32.34	0.0318	unit less
$k_{11} = 0.0258$	32.17	0.000525	unit less
$k_{12} = 0.6139$	33.01	0.128	hour^{-1}
$k_{13} = 0.0185$	7.92	0.00093	hour^{-1}
$k_{14} = 0.00013$	140.85	0.00011	hour^{-1}
$K_I = 0.833$	272.43	1.1435	g/L
$K_S = 2.0$	0.0358	0.000453	g/L
$K_{BA} = 0.5$	102.95	0.3255	g/L
$K_{AA} = 0.5$	3.34	0.0106	g/L

Table B.2: Fermentation variables with their corresponding rates of change

X	biomass concentration[g/L]	specific growth rate	$\mu = \frac{1}{X}\frac{dX}{dt}$
S	substrate concentration[g/L]	specific substrate utilization rate	$q_S = \frac{-1}{X}\frac{dS}{dt}$
B	butanol concentration[g/L]	specific rate of butanol accumulation	$q_B = \frac{1}{X}\frac{dB}{dt}$
BA	butyrate concentration[g/L]	specific rate of butyrate accumulation	$q_{BA} = \frac{1}{X}\frac{dBA}{dt}$
A	acetone concentration[g/L]	specific rate of acetone accumulation	$q_A = \frac{1}{X}\frac{dA}{dt}$
AA	acetate concentration[g/L]	specific rate of acetate accumulation	$q_{AA} = \frac{1}{X}\frac{dAA}{dt}$
E	ethanol concentration[g/L]	specific rate of ethanol accumulation	$q_E = \frac{1}{X}\frac{dE}{dt}$
CO_2	carbon dioxide concentration[g/L]	specific rate of carbon dioxide accumulation	$q_{CO_2} = \frac{1}{X}\frac{dCO_2}{dt}$
H_2	hydrogen concentration[g/L]	specific rate of hydrogen accumulation	$q_{H_2} = \frac{1}{X}\frac{dH_2}{dt}$

Appendix C

T^2 and Q-statistics for batch process monitoring

C.1 T^2-statistics for batch process

The T^2-statistics for a new batch of data can be calculated with the formula:

$$T_k^2 = (t_{new,k} - \bar{t}_k)' {\mathbf{S}_k}^{-1} (t_{new,k} - \bar{t}_k) \tag{C.1}$$

where, $t_{new,k}$ is the scores of the new batch of data at the k^{th} time instant and $\bar{t}_k$ corresponds to the column mean of the score matrix $\mathbf{T}_k$ from the normal batch data at the k^{th} instant and $\mathbf{S}_k$ is the covariance matrix of $\mathbf{T}_k$.
The control limit for the T^2-statistics can be calculated with the help of F-distribution:

$$T_\alpha^2 = \frac{R(I^2-1)}{I(I-R)} F_\alpha(R, I-R) \tag{C.2}$$

where, α is the significance level. Mostly α is taken as 0.99 and 0.95 for 99% and 95% critical points for F-distribution.

C.2 Q-statistics

The Q-statistics is mainly used for residual analysis, also know as squared prediction error(SPE). The SPE at each point k, for a new batch can be calculated as:

$$\begin{aligned} SPE_k &= \sum_{j=1}^{J} \mathrm{e}(k,j)^2 \\ \mathrm{e}(k,j) &= \mathrm{x}_{new} - \sum_{r=1}^{r} t_{new,k}\, \mathbf{P}_r \end{aligned} \tag{C.3}$$

The control limit of SPE can be calculated as[18]:

$$Q_{\alpha,k} = f.\chi^2_{h,\alpha} \tag{C.4}$$

We can roughly find out the values f and h by equating the mean and variance of the $f.\chi^2_{h,\alpha}$ distribution to the mean($\bar{x}$) and variance(σ^2) of SPE of the normal batch data at each time instant k. So, f and h are calculated as,

$$\begin{aligned} f &= \frac{\sigma^2}{2\bar{x}} \\ h &= \frac{2\bar{x}^2}{\sigma^2} \end{aligned} \tag{C.5}$$

So, the control limit can be re-written as,

$$Q_{alpha,k} = \frac{\sigma^2}{2\bar{x}}\ \chi^2_{\frac{2\bar{x}^2}{\sigma^2},\alpha} \tag{C.6}$$

where, $\chi^2_{\frac{2\bar{x}^2}{\sigma^2},\alpha}$ denotes the critical value of χ^2 distribution having $\frac{2\bar{x}^2}{\sigma^2}$ degrees of freedom with α level of significance. This method has been proved[18] and seen that it produced desirable results.

Bibliography

[1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & chemical engineering* 27, (2003) 293–311.

[2] D. Himmelblau. Fault detection and diagnosis in chemical and petrochemical processes. Chemical engineering monographs. Elsevier Scientific Pub. Co., 1978.

[3] D.-W. Gu, P. H. Petkov, and M. M. Konstantinov. Robust Control Design with MATLAB®, chapter Modeling of Uncertain Systems, 13–21. Springer London, London, 2013.

[4] R. Milne. Strategies for diagnosis. *IEEE Transactions on Systems, Man, & Cybernetics* .

[5] J. Gertler. Analytical redundancy methods in fault detection and isolation. In Preprints of IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'91. 1991 9–21.

[6] M. Desai and A. Ray. A fault detection and isolation methodology. In Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on. 1981 1363–1369.

[7] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control* 29, (1984) 603–614.

[8] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica* 12, (1976) 601 – 611.

[9] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. A review of process fault detection and diagnosis: Part II: Qualitative Models and Search Strategies. *Computers & Chemical Engineering* 27, (2003) 313–326.

[10] S. A. Lapp and G. J. Powers. Computer-aided Synthesis of Fault-trees. *IEEE Transactions on Reliability* R-26, (1977) 2–13.

[11] F. E. Finch and M. A. Kramer. Narrowing diagnostic focus using functional decomposition. *AIChE Journal* 34, (1988) 25–36.

[12] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & chemical engineering* 27, (2003) 327–346.

[13] K. Fukunaga. Introduction to statistical pattern recognition. Academic press, 2013.

[14] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2, (1989) 303–314.

[15] V. Venkatasubramanian and K. Chan. A neural network methodology for process fault diagnosis. *AIChE Journal* 35, (1989) 1993–2002.

[16] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology* 24, (1933) 417.

[17] L. H. Chiang, R. D. Braatz, and E. L. Russell. Fault detection and diagnosis in industrial systems. Springer Science & Business Media, 2001.

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

[19] K. Baker. Singular value decomposition tutorial. *The Ohio State University* 24.

[20] H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and psychological measurement* .

[21] R. B. Cattell. The scree test for the number of factors. *Multivariate behavioral research* 1, (1966) 245–276.

[22] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, (2010) 433–459.

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

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

[25] D. M. Himes, R. H. Storer, and C. Georgakis. Determination of the number of principal components for disturbance detection and isolation. In American Control Conference, 1994, volume 2. 1994 1279–1283 vol.2.

[26] L. H. Chiang, E. L. Russell, and R. D. Braatz. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and intelligent laboratory systems* 50, (2000) 243–252.

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

[28] C. W. L. Wong. Data Integration for the Monitoring of Batch Processes in the Pharmaceutical Industry. PhD dissertation, Newcastle University 2007.

[29] P. Nomikos. Statistical process control of batch process. PhD dissertation, MacMaster University 1995.

[30] P. Nomikos and J. F. MacGregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal* 40, (1994) 1361–1375.

[31] J.-M. Lee, C. K. Yoo, and I.-B. Lee. Enhanced process monitoring of fed-batch penicillin cultivation using time-varying and multivariate statistical analysis. *Journal of Biotechnology* 110, (2004) 119 – 136.

[32] J. Votruba, B. Volesky, and L. Yerushalmi. Mathematical model of a batch acetone–butanol fermentation. *Biotechnology and bioengineering* 28, (1986) 247–255.

[33] A. K. Srivastava and B. Volesky. Updated model of the batch acetone-butanol fermentation. *Biotechnology Letters* 12, (1990) 693–698.

[34] A. Singhal and D. Seborg. Pattern Matching In Historical Batch Data Using Pca. *IEEE Control Systems Magazine* 22, (2002) 53–63.