Structural Fault Detection Using Dynamic Principal Component Analysis (DPCA)

By

Mohana Rooparn A/L Kalaichelvan
15338

Dissertation submitted in partial fulfilment of
the requirements for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL)

SEPTEMBER 2014

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Structural Fault Detection Using Dynamic Principal Component Analysis (DPCA)

by

Mohana Rooparn A/L Kalaichelvan
15338

A project dissertation submitted to the

Chemical Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirements for the

BACHELOR OF ENGINEERING (Hons)

(CHEMICAL)

Approved by,

____

(Ir. Dr. Abd. Halim Shah Bin Maulud,)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

SEPTEMBER 2014

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MOHANA ROOPARN A/L KALAICHELVAN

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The author gratefully acknowledges the supervisor for this project, Ir. Dr. Abd Halim Shah Bin Maulud, for his continuous support and assistance provided throughout this project and Universiti Teknologi PETRONAS for providing the necessary equipment and facilities to conduct this project.

ABSTRACT

Principal component analysis (PCA) is a well-known data dimensionality reduction technique that has been used to detect faults during the operation of industrial processes. A modification to this is the Dynamic Principal Component Analysis (DPCA) which takes into account serial correlations for rapid sampling and detection of faults. This method, although being studied for its fault detection capabilities, has not yet been widely tested and proved to detect a particular type of faults called structural faults. In this paper, a dynamic model of a Continuous Stirred Tank Reactor (CSTR) is built using the MATLAB software and used to generate a sample of base data and another sample of data with structural faults present. Further on in this project, the two data sets would be compared and tested using T^2-statistics and Q-statistics method to identify the faults occurring in the system and study the difference in performances of these methods. Q-statistics which quantifies variations in the residual space is more sensitive but less robust to the faults than the T^2-statistics quantifying the variations in the score or state space. Faster fault detection via DPCA is also achieved in T^2-statistics whereas results from Q-statistics are inconclusive.

CHAPTER 1
INTRODUCTION

This chapter provides a basic introduction of the project by giving a brief abstract of the project, its problem statement and finally highlighting the objectives outlined for the project.

1.1. BACKGROUND

In the recent years, there has been an increase in awareness towards to quality of manufactured goods as to producing higher quality products and maintaining a low rejection rate while under compliance with stricter safety and environmental regulations. To cope with this, process monitoring and control becomes of utmost importance. Standard process controllers can only detect and correct certain disturbances occurring in the process but however, are unable to handle other changes to the process. These changes, known as faults, are unpermitted deviations to the characteristic properties of the process. Therefore, detecting and subsequently correcting these faults are of major importance.

Fault detection, along with fault identification, fault diagnosis and process recovery form the basis of process monitoring. The goal of process monitoring is to ensure the monitored process stays within desired conditions by recognising anomalies in the process behaviour and subsequently correcting it (2). Fault detection has been the object of study for a long time with many scholarly articles on fault detection and diagnosis. Its importance is seen when detecting a fault and correcting it in time while the system is operating avoids abnormal events and losses, thus keeping the process in desired conditions and avoiding any major system breakdowns. There are two main methods for fault detection, namely, model based and data driven. In model based fault detections one has to know the exact process model or the mathematical model of the system. However for certain industrial processes contain many process variables and input parameters, it would be very difficult to know the exact process model of such systems, and in such cases data driven or statistical techniques can be used to detect and diagnose the fault.

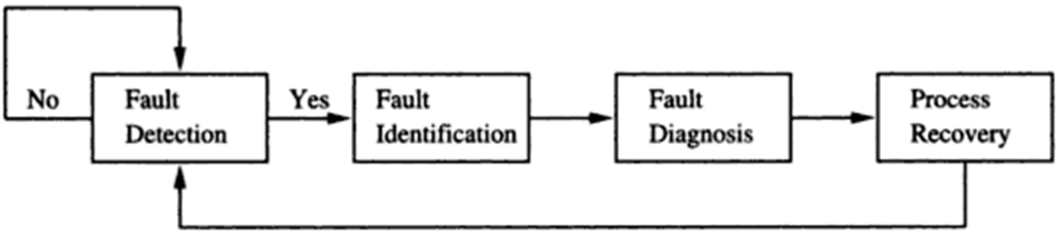

Figure 1: Process monitoring loop

This project studies such data driven fault detection techniques occurring in industrial process systems. The particular type of fault being studied are structural faults. In large industrial process systems, a wide array of sensors and transmitters collect real-time data of the process. This data is monitored to identify and determine any abnormalities in the process behaviour known as faults, which may arise from factors such as equipment wear and tear, process parameter changes and equipment failure (4). In this project, the author applies several multivariate statistical methods to compress and reduce the dimensionality of the data and study the accuracies of fault detection.

1.2. PROBLEM STATEMENT

In recent years, there has been concerted effort throughout the industry improve upon production efficiency and quality. At the heart of this are process monitoring and diagnosis procedures such as fault detection (1). Early detection of these faults may provide invaluable warning on emerging problems, with appropriate actions taken to avoid serious process upsets (2). Commonly used methods for detecting faults are multivariate statistical methods with the most popular one being Principal Component Analysis (PCA), which is a data dimensionality reduction technique. An expansion of PCA to handle process dynamics is the Dynamic Principal Component Analysis (DPCA). Accuracies of fault detection using PCA and

DPCA methods were studied in various literatures, however, majority of it address only on the accuracy of the variable fault detection. Variable faults can be easily detected via   Due to this; further research is required to study the accuracies of structural fault detection.

1.3. OBJECTIVES

The two main objectives the author has identified for carrying out this project are such as the following:

1. To develop a Continuous Stirred Tank Reactor (CSTR) computer simulation model and generate structural faults in the simulation.

2. To investigate the performance of fault detection accuracies using the DPCA as compared to PCA with T^2 statistics and Q statistics techniques.

1.4. SCOPE OF STUDY

The scopes of studies covered by the author in this particular project pertaining to the objectives mentioned previously are as per following:

1. Studying and modelling a dynamic computer simulation of a CSTR to generate faults.

2. Studying the application of DPCA and PCA fault detection methods with differing statistical techniques.

CHAPTER 2
LITERATURE REVIEW

In this chapter, the author discusses the themes addressed in this project such as the types of faults studied and the fault detection methods applied, with reference to available literature and past researches. The final study will be based upon the literature discussed in this chapter.

2.1. TYPES OF PROCESS FAULTS

Disturbances or faults in a process being monitored can be broken down into two main types, namely variable faults and structural faults. A variable fault or change is a basic form of disturbance observed as step changes or exponential variations usually occurring in the variables governing the process itself. Such variable faults include, temperatures, pressures and feed flowrates. Variable faults are rather easily detectable via univariate process monitoring methods.

Structural faults occur in a process when the main characteristics governing the process changes or deviates from the ideal operating conditions. Examples of structural faults in a process include drift in reaction kinetics due to catalyst deactivation or a loss in heat transfer efficiency caused by fouling within the heat exchanger. Structural faults are rather difficult to detect compared to variable faults and this is the type of faults being studied in this project.

2.2. STATISTICAL PROCESS CONTROL

Statistical process control (SPC) employs statistical methods to study the data collected from processes for any changes. SPC can be broken down into univariate methods and multivariate methods.

Univariate methods generally involve the monitoring of a single process variable at a given time to compare its value against lower and upper control limits. Univariate monitoring methods can be used to determine the threshold for each observation variable where these thresholds define the boundary for in-controlled operations and a violation of these limit with on-line data would indicate a fault has occurred (2).

The application of univariate process monitoring methods breaks down when applied to larger multivariable process systems (8). Multivariate process monitoring methods address this limitation by considering all the data simultaneously and extracting information on the directionality of the variations of one variable relative to the other. This technique allows the efficient handling of the massive amounts of data collected in the plant (6).

2.3. PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) is an optimal data dimensionality reduction technique which is able to capture the maximum variance present in the data set (4). It is able to do this by determining a set of orthogonal vectors called loading vectors. These loading vectors are ordered by the amount of variance present in the loading vector directions. The use of PCA as a fault detection method is motivated by these factors:

1. PCA produces lower dimensional representations of the data set, transforming it into a new set of generalized independent variables called principle components and therefore, improve the proficiency of detecting and diagnosing faults.

2. The structure derived from PCA aids in identifying the variables causing the fault and/or the variables most affected by the fault.

3. PCA can separate the observation space into a subspace capturing the systematic trends of the process and a subspace containing essentially the random noise

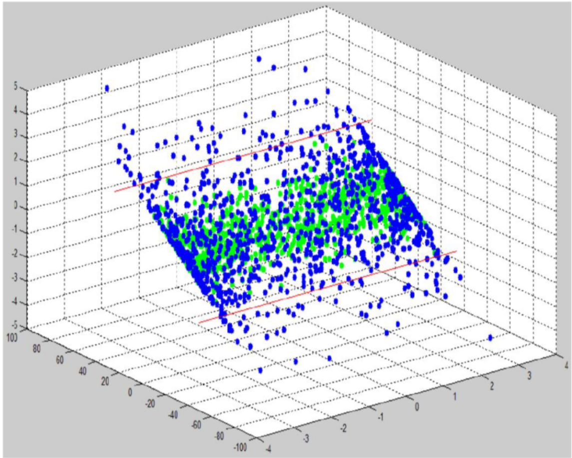

Figure 2: PCA analysis of a sample data (normal (GREEN) and faulty (BLUE) operating conditions)

2.2.1. Loading Vectors

Given for *n* observations of *m* measurement variables organised into data matrix X, the loading vectors are obtained by computing the singularities of the optimisation problem (4).

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

Eq. (2.1)

where $v \in R^m$. The stationary points of Eq. (1) can be computed using Singular Value Decomposition (SVD)

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

Eq. (2.2)

where $U \in R^{n \times n}$ and $V \in R^{m \times m}$ are unitary matrices and $\Sigma \in R^{n \times m}$ is the matrix containing the non-negative singular values. The loading vectors are the orthonormal column vectors in the matrix *V*, and the variance of the training set projected along the *i* th column of *V* is equal to σ_i^2.

This equation is equivalent to solving an eigenvalue decomposition of the covariance matrix *S* of the data set:

$$S = \frac{1}{n-1} X^T X = V \Lambda V^T$$

Eq. (2.3)

where *V* is the orthogonal and Λ is the diagonal and the loading matrix *P* is then built from the columns of *V*.

The data set is then projected into a lower dimensional score matrix T:

$$T = XP$$

Eq. (2.4)

2.2.2. Fault Detection

T^2 statistic is normally used to detect faults for multivariate process system. For an observation vector $\boldsymbol{x}$ and assuming that $\Lambda = \Sigma^T \Sigma$ is invertible, the T^2 statistic can be obtained from the PCA representation as such (4):

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

Eq. (2.5)

By including the matrix P the loading vectors associated with the a largest singular values, the T^2 statistic is then computed as:

$$T^2 = x^T V P \Sigma_a^{-2} P^T x$$

Eq. (2.6)

where, Σ_a contains the first a rows and columns of Σ, and x is an observation vector of dimension m.

The T^2 statistic threshold is:

$$T_\alpha^2 = \chi_\alpha^2(a)$$

Eq. (2.7)

and for a normalised data set, the T^2 statistic threshold is derived as:

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

Eq. (2.8)

where $F\alpha(a, n - a)$ is the upper $100\alpha\%$ critical point of the *F-distribution* with a and $n - a$ degrees of freedom. The calculated threshold would represent an elliptical confidence region. Any observation vector outside this region would register as a fault. The T^2 statistic can be overly sensitive to inaccuracies in the PCA due to smaller singular values.

For robust monitoring of the portion of the measurement space corresponding to the $m - a$ smallest singular values, Q statistic would be of better use:

$$Q = r^T r, \qquad r = (1 - PP^T)x$$

Eq. (2.9)

where r is the residual vector, a projection of observation x into residual space. The Q statistic is not overly sensitive as the T^2 statistic for it measures the total sum of variations in the residual space and not directly along each loading vector (4).

The threshold for Q statistic can be approximated as:

$$Q_a = \theta_1 \left[\frac{h_0 C_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{1/h_0}$$

Eq. (2.10)

where

$$\theta_i = \sum_{j=a+1}^{n} \sigma_j^{2i}$$

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

and c_α is the normal deviate which corresponds to *(1- α)* percentile. The Q statistic is able to measure the random variations within a process such as noise and a violation of the threshold would indicate a significant change in the random noise.

The T^2 statistic and the Q statistic along with their appropriate thresholds detect different types of faults and advantages of both these methods can be utilized by employing the two measurements together.

2.4. DYNAMIC PRINCIPAL COMPONENT ANALYSIS (DPCA)

PCA monitoring methods assume that observations at any time instant are independent to observations of previous time instances. This is only valid for chemical process systems of long sampling intervals. For shorter sampling intervals and faster fault detection, the PCA methods are extended to take into account the serial correlations, by augmenting each observation vector with the previous l observations and stacking the data matrix as following (4):

$$X(l) = \begin{bmatrix} X_t^T & X_{t-1}^T & \cdots & X_{t-l}^T \\ X_{t-1}^T & X_{t-2}^T & \cdots & X_{t-l-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ X_{t+l-n}^T & X_{t+l-n-1}^T & \cdots & X_{t-n}^T \end{bmatrix}$$

Eq. (2.11)

where $x^T{}_t$ is the m-dimensional observation vector in the training set at time instance t. By performing PCA on the data matrix in Eq. (2.11), a multivariate autoregressive (AR) model is extracted from the data. The Q statistic is then the squared prediction error of the model. If enough lags l are included in the data matrix, the Q statistic becomes statistically independent between time instances, and the threshold Eq. (2.10) is would be justified. This approach is known as dynamic PCA (DPCA).

To put this in perspective, consider the following case where an AR process is defined as (11):

$$z(k) = 0.8 \cdot z(k-1) + u \cdot (k-1)$$

If using static PCA method,

$$\mathbf{x}_1 = [z(k) \quad u(k)]^T$$

where u is a coloured noise sample with 1000 samples generated as follows:

$$u(k) = 0.7 \cdot u(k-1) + w \cdot (k-1)$$

and w is a white noise with variance 1. The SVD of the covariance matrix of the data matrix $\mathbf{X}_1(k)$ without any lagged variables is:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 21.47 & 0 \\ 0 & 1.352 \end{bmatrix}, \qquad \mathbf{V}_1 = \begin{bmatrix} 0.985 & -0.175 \\ 0.175 & 0.985 \end{bmatrix}$$

After autoscaling,

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.542 & 0 \\ 0 & 0.458 \end{bmatrix}$$

It can be seen that there is no relation between the variables due to the absence of singular values equal to zero. To apply dynamic PCA to this case, the data vector is modified to:

$$\mathbf{x}_2 = [z(k) \quad u(k) \quad z(k-1) \quad u(k-1)]^T$$

By applying SVD,

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 42.37 & & & \\ & 2.599 & & \\ & & 0.709 & \\ & & & 0 \end{bmatrix}$$

$$\mathbf{V}_2 = \begin{bmatrix} 0.699 & -0.225 & 0.287 & -0.615 \\ 0.110 & -0.633 & -0.766 & 0 \\ 0.692 & 0.452 & -0.274 & 0.492 \\ 0.145 & -0.587 & 0.506 & 0.615 \end{bmatrix}$$

If the data is auto-scaled,

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 2.967 & & & \\ & 0.751 & & \\ & & 0.285 & \\ & & & 0 \end{bmatrix}$$

The zero singular value present in the matrix suggests that there is a dynamic relation between the variables revealed by the fourth singular vector:

$$-0.615 \cdot z(k) + 0.492 \cdot z(k-1) + 0 \cdot u(k) + 0.615 \cdot u(k-1)$$

Rearranging this yields

$$z(k) = 0.8 \cdot z(k-1) + u \cdot (k-1)$$

CHAPTER 3
METHODOLOGY

This chapter discusses the methods and procedures suggested to carry out this project. A Gantt chart is attached at the end of the chapter, highlighting the progress of the project.

3.1. CSTR SIMULATION MODEL

The continuous stirred tank reactor (CSTR) is a common reactor unit used in industrial process. It is a tank type reactor in which the contents are well stirred and it runs with continuous flow of reactants as well as products (5). The CSTR normally runs at a steady state condition with a uniform distribution of concentration and temperature throughout the reactor, thus making it easier to model as compared to other forms of reactors.

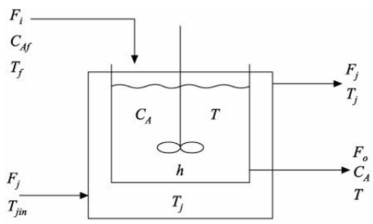

Figure 3: Schematic representation of CSTR

To develop a mathematical model of the CSTR in MATLAB, several assumptions were made. The assumptions are as following (5):

1. The heat losses from the process are negligible (well-insulated).
2. The mixture density and heat capacity are assumed constant.
3. There are no variations in concentration, temperature, or reaction rate throughout the reactor as it is perfectly mixed.
4. The exit stream has the same concentration and temperature as the entire reactor liquid.
5. The overall heat transfer coefficient is assumed constant.

6. No energy balance around the jacket is considered. Indeed, the jacket temperature can directly be manipulated in order to control the desired reactor temperature.
7. The reactor is a flat-bottomed vertical cylinder and the jacket is around the outside and the bottom.

The CSTR simulation model in MATLAB will be built using these predefined parameters and operating conditions (5):

Table 1: Default operating parameters

Operating Parameter	Symbol	Value
Cross-sectional area of the reactor, ft^2	A_c	10.36
Concentration of reactant A in the exit stream, lb-mol/ft^3	C_A	0.05
Concentration of A in the feed stream, lb-mol/ft^3	$C_{A,f}$	0.9
Diameter of the cylindrical reactor, ft	d	3.6319
Activation energy, BTU/ lb-mol	E	30000
Volumetric feed flow rate, ft^3/h	F_i	20
Height of the reactor liquid, ft	h	3.8610
Heat of reaction, BTU/ lb-mol	$-\Delta H_r$	-30000
Universal gas constant, BTU/ (lb-mol)(R)	R	1.987
Frequency factor, h^{-1}	α	7.08×10^{10}
Multiplication of mixture density and heat capacity, BTU/(ft^3)(R)	ρC_P	37.5
Reactor temperature, R	T	650
Feed temperature, R	T_f	600
Jacket temperature, R	T_j	70.0
Overall heat transfer coefficient, BTU/(ft^2)(R)(h)	U_i	150

3.1.1. Model Development

Total Continuity Equation:

Mass inflow rate = F_i

Mass outflow rate = F_o

Rate of mass accumulation within reactor = $\frac{d(\rho V)}{dt} = \frac{d(\rho A_c h)}{dt}$

Eq. (3.1)

A_c is cross-sectional area of reactor and h is the height of the reactor liquid.

$$\frac{d(\rho V)}{dt} = (F_i - F_o)\rho$$

$$\frac{dV}{dt} = F_i - F_o$$

Eq. (3.2)

The reactor holdup, V and the exit flow rate F_o can be related as:

$$F_o \propto \sqrt{V}$$

For this CSTR, $F_o = \sqrt{10 A_c h}$ Eq. (3.3)

Combining equations 3.2 and 3.3:

$$\frac{dh}{dt} = \frac{F_i}{A_c} - \sqrt{\frac{10h}{A_c}}$$ Eq. (3.3)

<u>Component Continuity Equation:</u>

Mass inflow rate component A = $F_i C_{Af}$,

Mass outflow rate component A = $F_o C_A$,

Rate of generation of component A = – $(-r_A)V$

Rate of accumulation of component A within the reactor = $\frac{d(VC_A)}{dt}$

where $-r_A$ is the rate of consumption of chemical species A. The basic balance equation then becomes,

$$\frac{d(VC_A)}{dt} = F_i C_{Af} - F_o C_A - (-r_A)V$$

$$C_A \frac{dV}{dt} + V\frac{dC_A}{dt} = F_i C_{Af} - F_o C_A - (-r_A)V$$

Eq. (3.4)

Substituting equation 3.2 into 3.4 and simplifying,

$$\frac{dC_A}{dt} = \frac{F_i}{A_c h}(C_{Af} - C_A) - (-r_A)$$ Eq. (3.5)

For the given first-order reaction,

$$-r_A = kC_A$$
$$= \alpha \exp\left(\frac{-E}{RT}\right)C_A$$

Eq. (3.6)

Combining equations 3.5 and 3.6,

$$\frac{dC_A}{dt} = \frac{F_i}{A_c h}\left(C_{Af} - C_A\right) - \alpha \exp\left(\frac{-E}{RT}\right)C_A \qquad \text{Eq. (3.7)}$$

<u>Energy Balance Equation:</u>

Energy input rate = $F_i C_p T_f$

Energy output rate = $F_o C_p T + U_i A_h (T - T_j)$

Energy added by exothermic reaction $= (-\Delta H)V\alpha \exp\left(\frac{-E}{RT}\right)C_A$

Energy accumulation rate:

$$\frac{d(V\rho C_p T)}{dt} = F_i \rho C_p T_f - F_o \rho C_p T - U_i A_h (T - T_j) + (-\Delta H)V\alpha \exp\left(\frac{-E}{RT}\right)C_A \qquad \text{Eq. (3.8)}$$

Using equation 3.2 and further simplifying:

$$\frac{dT}{dt} = \frac{F_i}{A_c h}\left(T_f - T\right) + \left(\frac{-\Delta H}{\rho C_p}\right)\alpha \exp\left(\frac{-E}{RT}\right)C_A - \frac{U_i A_h}{\rho C_p A_c h}\left(T - T_j\right) \qquad \text{Eq. (3.9)}$$

And therefore, this is the final form of the energy balance equation.

3.2. SIMULATION OF FAULTS

The developed CSTR simulation model will be used to generate faults to test the detection accuracies of the PCA, DPCA and SPE methods. These faults, mentioned earlier are structural and not variable faults. Structural faults are the ones occurring in a process due to alteration of the main characteristics of the process. The structural faults simulated in this project would include:

1. Drift in reaction kinetics.
    - e.g. Activation energy
    - Drift ranges, 1%, 5% and 20%

A data set would then be generated using the developed model which is subjected to these faults and the data studied using the PCA, DPCA and SPE to detect the presence of any such faults. The results from each fault detection method would then be compared to investigate the accuracies of the respective methods.

3.3. PROJECT TOOLS

The primary tool used for this project is the MATLAB software, specifically the SIMULINK simulation environment. MATLAB, (MATrix LABoratory), is a high level computing software for numerical computations and graphics developed by MathWorks. MATLAB is primarily designed for matrix computations such as solving linear systems, computing eigenvalues and eigenvectors and etc. It even has its own programming language which can be used to program specific functions to expand the capabilities and interface with other programmes written in other programming languages. The SIMULINK simulation environment is a separate feature of MATLAB which is a data graphical programming tool. It is widely used in control theory as it allows the user to model, simulate and analyse dynamic systems such as the CSTR model as mentioned above.

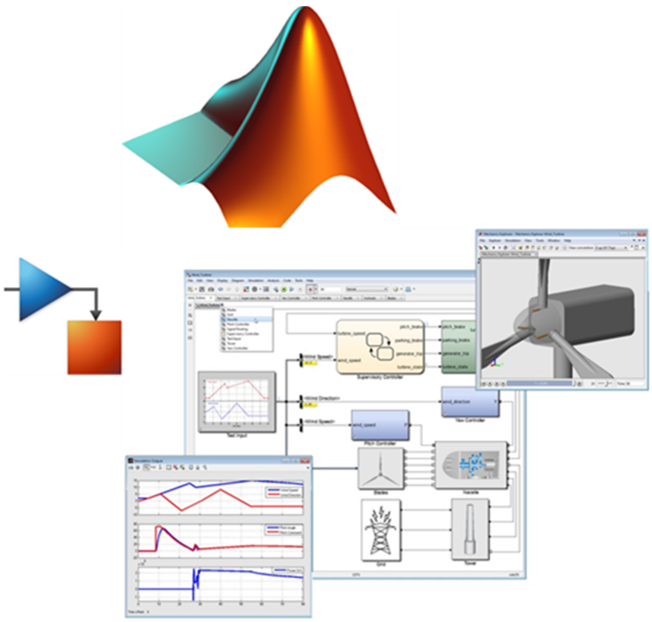

Figure 4: MATLAB logo (top), SIMULINK logo (left), wind turbine simulation in SIMULINK (right)

3.4. FYP II GANTT CHART

Table 2: Gantt chart

No.	Activities	Weeks														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Project Work Continuation															
2	Progress Report Submission							✓								
3	Project Work Continuation															
4	Pre-SEDEX										✓					
5	Draft Final Report Submission											✓				
6	Dissertation Submission (Soft Bound)												✓			
7	Technical Paper Submission												✓			
8	Viva													✓		
9	Dissertation Submission (Hard Bound)															✓

Milestones: ✓

3.5. PROJECT FLOWCHART

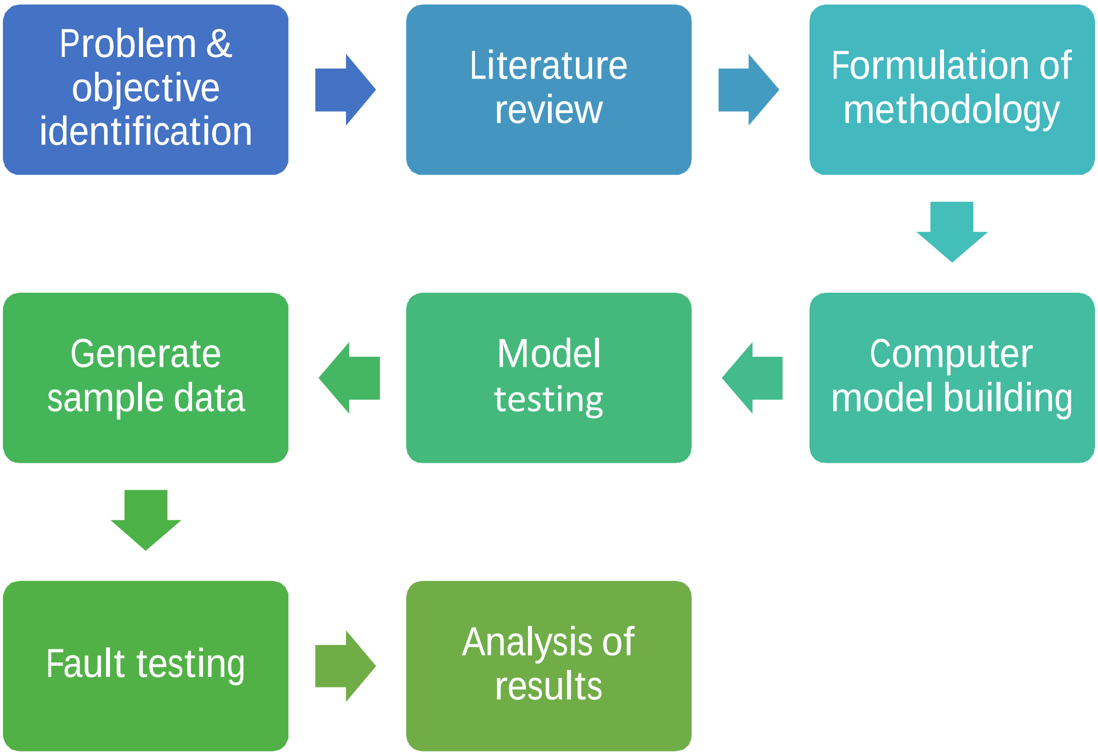

Figure 5: Project Flowchart

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter compiles the data collected up to this date from the simulations conducted and the subsequent analysis of the data. This is followed by the findings regarding the project based on the engineering and technical review done on the second chapter.

4.1. SIMULINK MODEL

The diagram shows the computer model built using Simulink for the dynamic simulation of a CSTR using the given parameters. This model is then used to generate a sample set of baseline data (without faults) to be tested and used as benchmark later on.

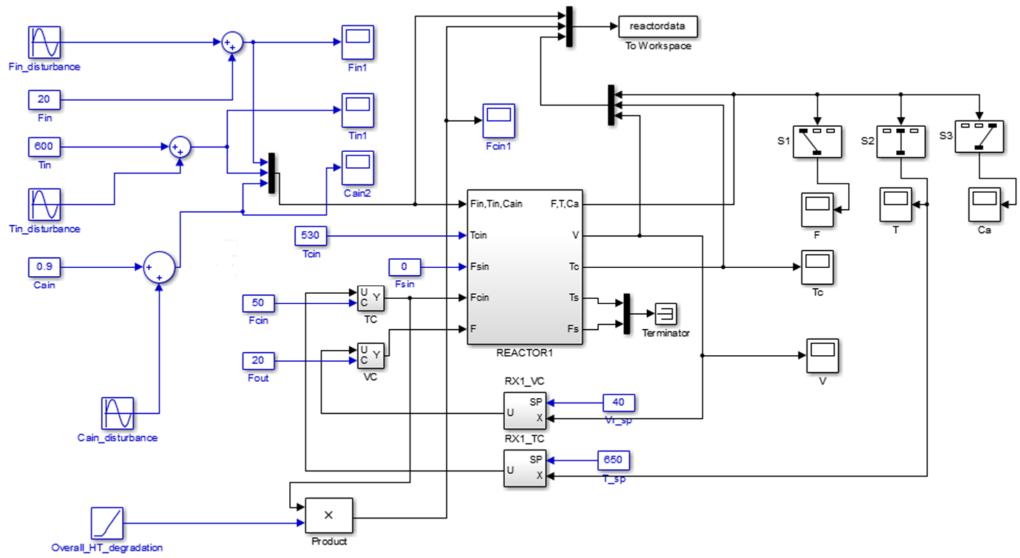

Figure 6: Simulink Model

Based on the model, there are 3 main inputs, namely feed flowrate, temperature and concentration denoted as F_{in}, T_{in} and $C_{a,in}$. To simulate a non-ideal operating condition, disturbances are added such as the sine wave function and random number function for measurement noise. The 3 main outputs which are recorded are the product flowrate, temperature and concentration denoted as F, T and C_a.

Initially, a base data set without any faults present is generated. It is done by running the simulation using exactly the default operating parameters shown in Table 1. This base data set is the one used to compute the loading vectors for both PCA and DPCA methods. The loading vector will then be incorporated with the fault data set. However, for the fault data set, the Simulink model is slightly modified. A ramp tool is added to the model as an input to simulate the drifting increase in activation energy at 1%, 5% and 20%. Therefore, 3 fault data sets corresponding to the different drift levels are generated. Both the base and fault data set consist of n = 5000 observations.

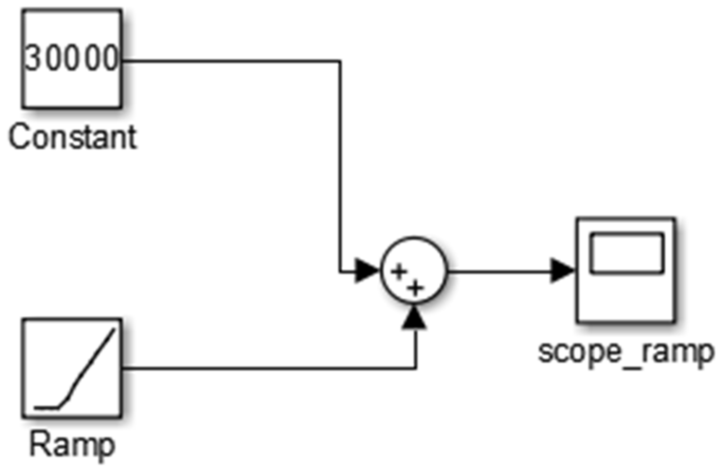

Figure 7: Ramp input

4.2. BASE DATA

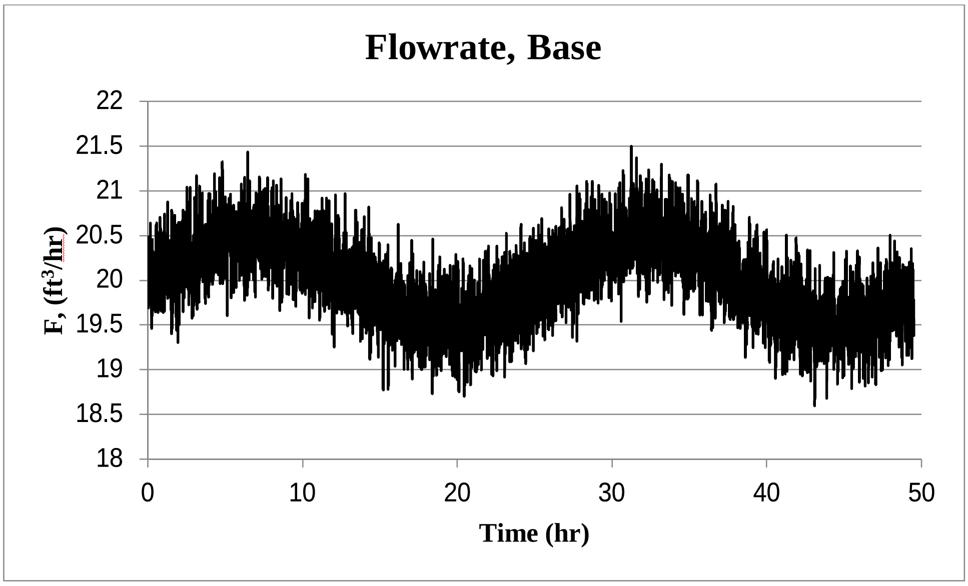

Figure 8: Product flowrate base data

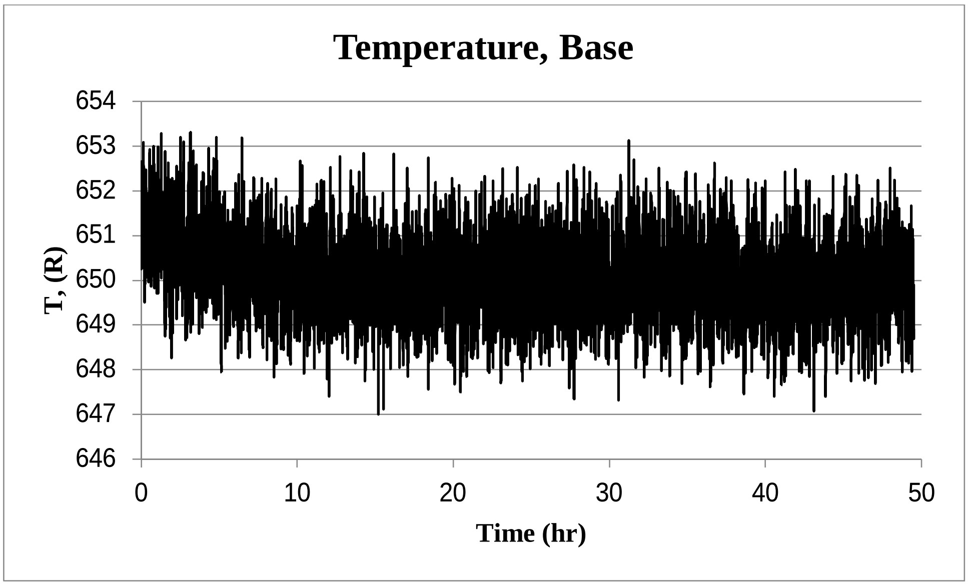

Figure 9: Product temperature base data

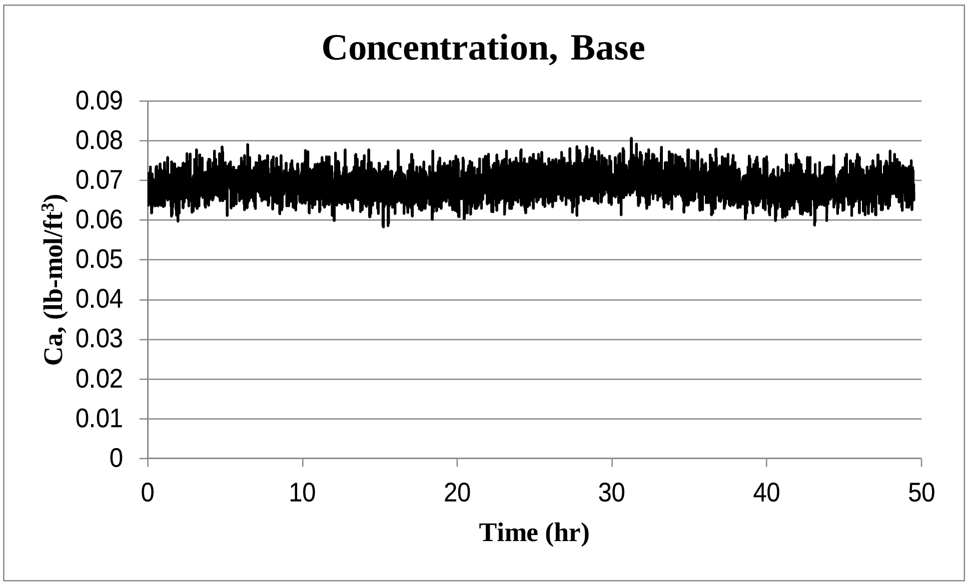

Figure 10: Product concentration base data

The data obtained in the graphs shown above are the sample base data set generated without the structural faults. The effects of noise is evident in all the graphs with fluctuating values along the plot but the sine wave disturbances can only be clearly seen in the product flowrate graph. The noise present in the data simulates measurement noise.

This base data set then normalized to 0 mean and unit variance and becomes the input for the MATLAB function *princomp*, which calculates the loading matrix (V), eigenvalues (Λ) and the score matrix (T). The obtained values are:

$$V = \begin{bmatrix} 0.5460 & 0.8323 & -0.0956 \\ 0.5875 & -0.4618 & -0.6646 \\ 0.5973 & -0.3067 & 0.7411 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 2.6155 & 0.3170 & 0.0675 \end{bmatrix}$$

To optimally capture the variations of the data while minimizing the effects of noise, only the loading vectors corresponding to the a largest singular values are retained in the loading matrix, in this case, 2. Therefore, the new loading matrix, P is:

$$P = \begin{bmatrix} 0.5460 & 0.8323 \\ 0.5875 & -0.4618 \\ 0.5973 & -0.3067 \end{bmatrix}$$

4.3. FAULT DATA

The graphs below show the fault data generated from the Simulink model at 1%, 5% and 20% drifts.

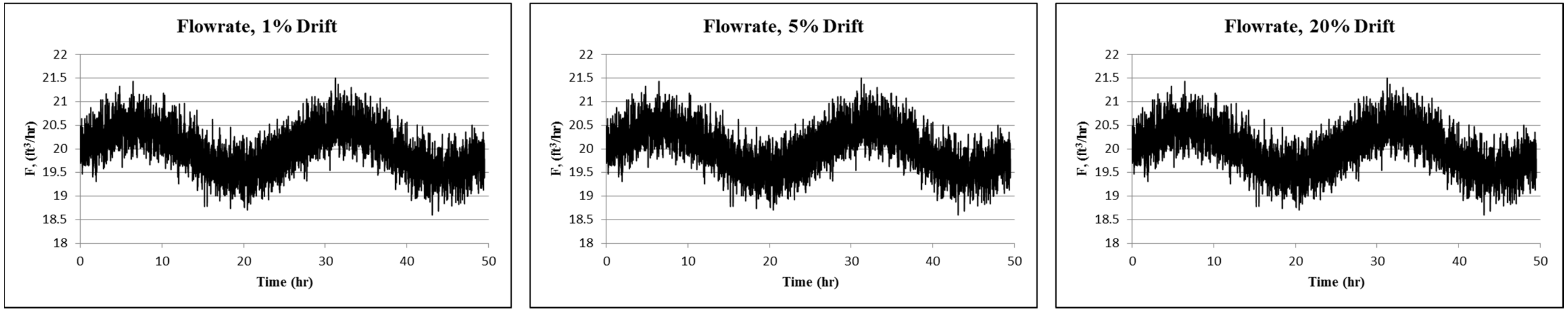

Figure 11: Product flowrate fault data

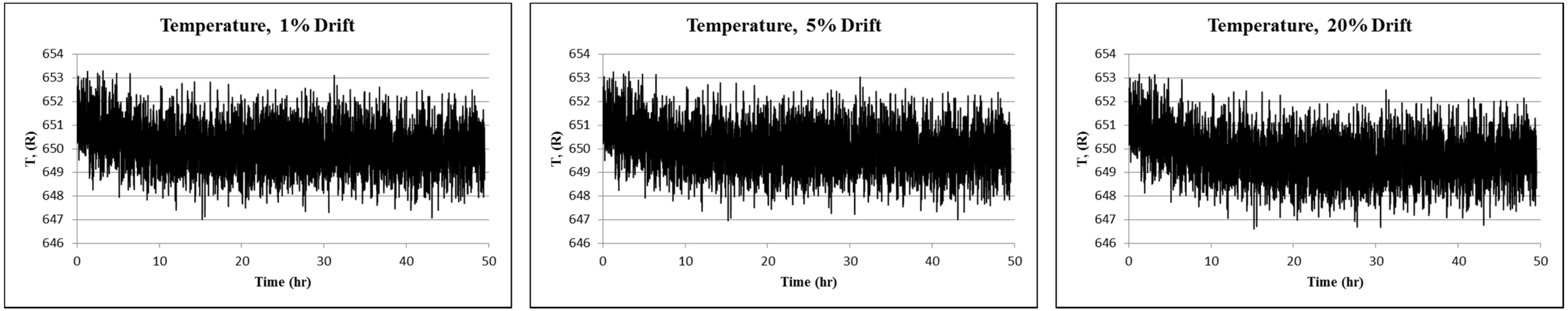

Figure 12: Product temperature fault data

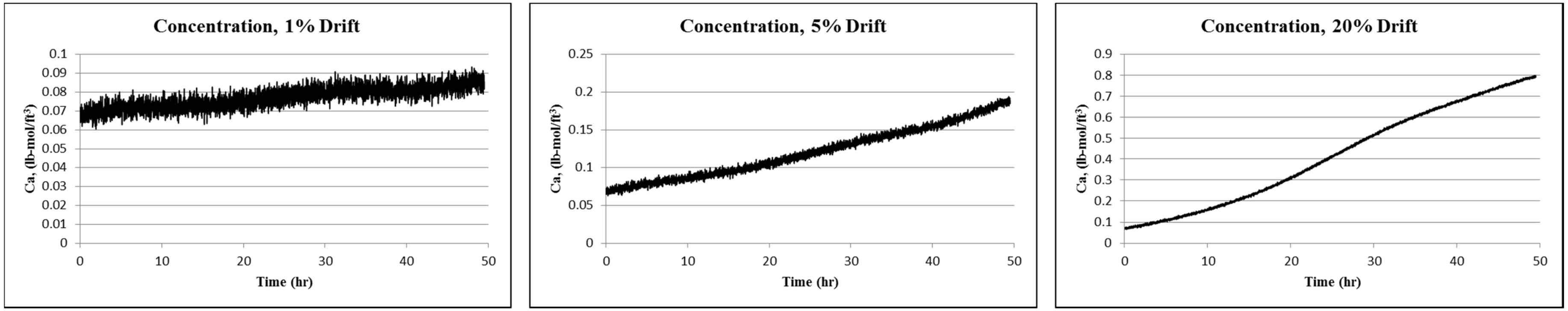

Figure 13: Product concentration fault data

As can be seen in the fault data, the variation among flowrate and temperature for different fault levels are almost non-existent but stark differences can be seen among concentration data. This could be due to the fact that at higher activation energies, reaction would not proceed fast and thereby consuming lesser reactants.

4.4. STATISTICAL ANALYSIS

4.4.1. PCA and DPCA

With the fault data generated, 3 new data arrays for the 3 levels of drift are constructed in the following configuration:

Table 3: PCA configuration

Column 1	Column 2	Column 3
F	T	C_a

This array is then autoscaled according to the mean and variances of the base data. PCA score of the autoscaled array now is calculated using Eq.(2.4) and the loading matrix *P*. For the DPCA scores, the process is repeated but this time with a lag of 2 units as shown below:

Table 4: DPCA configuration

C1	C2	C3	C4	C5	C6	C7	C8	C9
$F_{(1x1)}$	$F_{(1x2)}$	$F_{(1x3)}$	$T_{(1x1)}$	$T_{(1x2)}$	$T_{(1x3)}$	$C_{a(1x1)}$	$C_{a(1x2)}$	$C_{a(1x3)}$

4.4.2. T^2-statistics and Q-statistics

The PCA and DPCA data are first tested using T^2-statistics. The thresholds calculated using Eq. (2.8) is shown in the table below:

Parameter	PCA	DPCA
a	2	3
n	5000	5000
T_α^2	9.22	11.36

The graphs in the following page show the PCA and DPCA T^2-statistics for fault detection.

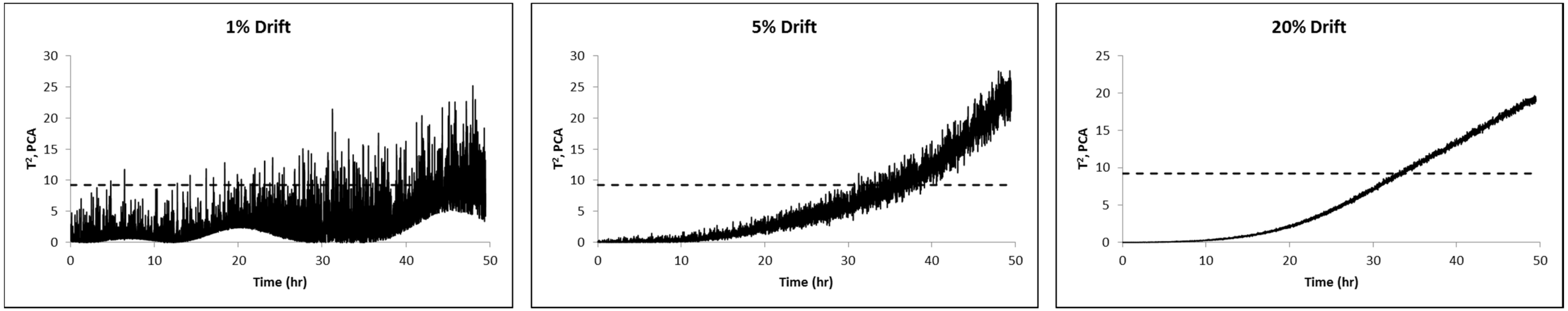

Figure 14: PCA T^2-statistics for fault detection

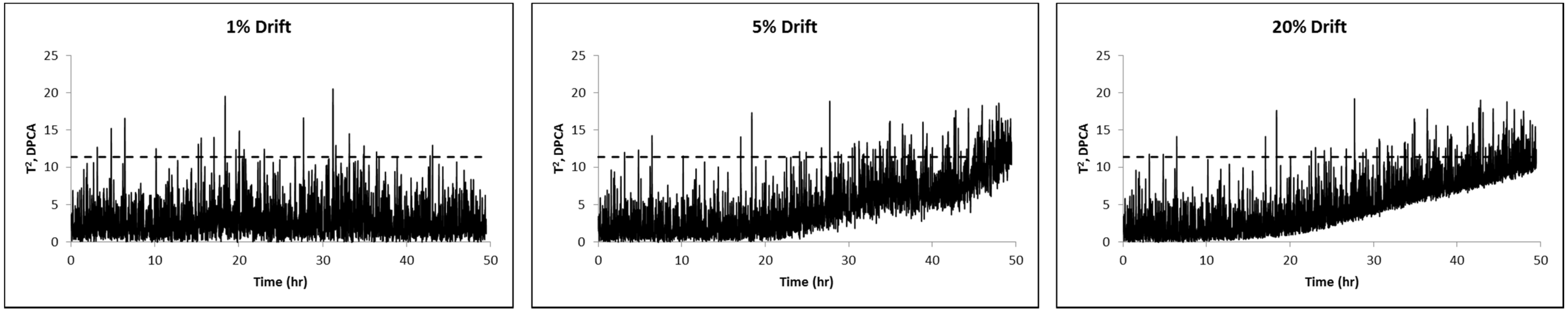

Figure 15: DPCA T^2-statistics for fault detection

To model using Q-statistics, the residual matrix of the data is required. This residual matrix captures the variations beyond the a loading vectors. The residual matrix for PCA and DPCA is obtained from Eq.(2.9). The threshold for Q-statistics is obtained from Eq.(2.10).

Case	**Thresholds, Q_α**	
	PCA	**DPCA**
1%	10.78	16.33
5%	440.48	499.24
20%	20144.28	22181.29

The graphs in the following page show the PCA and DPCA Q-statistics for fault detection.

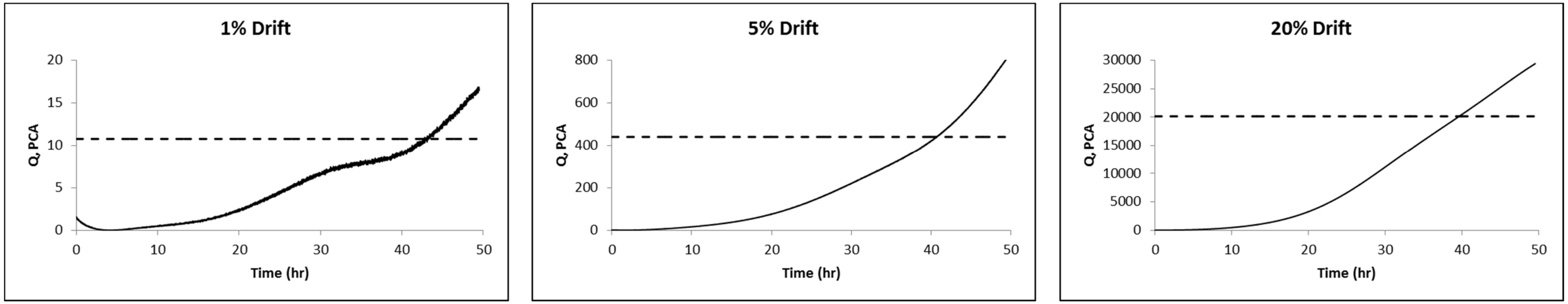

Figure 16: PCA *Q*-Statistics for fault detection

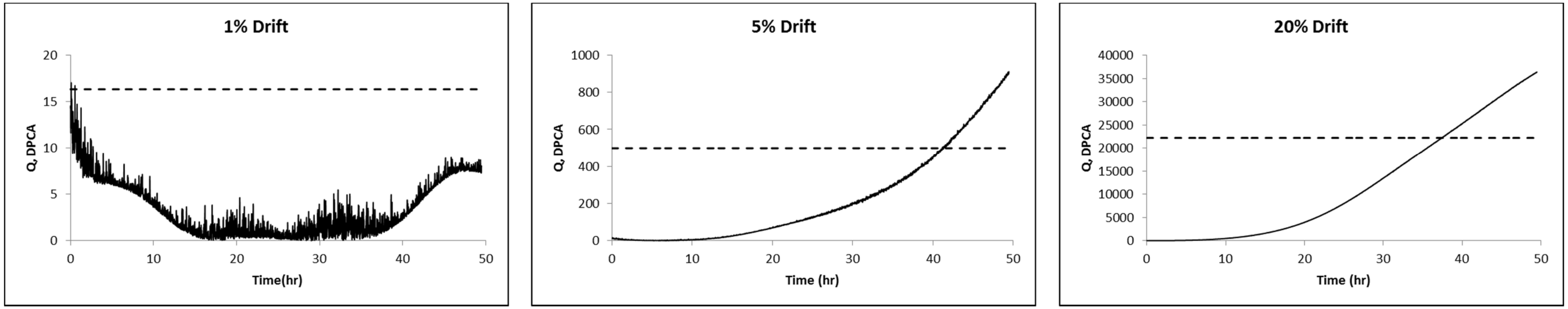

Figure 17: DPCA *Q*-Statistics for fault detection

From the T^2-statistics graphs shown above, it can be seen that there are observations above the threshold occurring earlier in DPCA compared to PCA. This occurrence indicates fault detection and this fault detection happening earlier in DPCA shows that DPCA detects the fault earlier compared to PCA.

However, for the Q-statistics graphs, it is difficult to notice the difference in first detection times for PCA and DPCA. Further analysis of both sets of the results is shown in the tables below.

Table 5: First detection times (hours) from T^2-statistics

Case	PCA	DPCA
1%	4.8	3.2
5%	30.7	17.1
20%	32.9	20.1

Table 6: First detection times (hours) from Q-statistics

Case	PCA	DPCA
1%	42.63	0.09
5%	40.65	41.19
20%	39.6	37.42

CHAPTER 5
CONCLUSION AND RECOMMENDATION

Based on the simulations conducted and the statistical methods utilised, it can be concluded that DPCA is more effective and faster fault detecting technique compared to PCA but only marginally.

This can be seen in the T^2-statistics where DPCA has an earlier first detection time compared to PCA for all three tested cases. From the Q-statistic method, which was used for the quantifying variations in the residual space, earlier detection times for DPCA are seen for cases of 1% and 20% drift but not in 5% drift leading to inconclusive result and therefore will require further study.

In addition to this, the two main objectives highlighted at the beginning of this project, which are:

- To develop a Continuous Stirred Tank Reactor (CSTR) computer simulation model and generate structural faults in the simulation.
- To investigate the performance of fault detection accuracies using the DPCA as compared to PCA with T^2- statistics and Q-statistics techniques.

have been successfully accomplished.

REFERENCES

1. Chen, J., & Liao, C.-M. (2002). Dynamic process fault monitoring based on neural network and PCA. *Journal of Process Control, 12,* 277-289.

2. Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault Detection and Diagnosis in Industrial Systems.* London: Springer.

3. Jiang, Q., & Yan, X. (2012). Chemical processes monitoring based on weighted principal component analysis and its application. *Chemometrics and Intelligent Laboratory Systems, 119,* 11-20.

4. Russell, E. L., Chiang, L. H., & Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 51,* 81-93.

5. Jana, A. K. (2011). Chemical Process Modelling and Computer Simulation

   (2nd ed.). Delhi: PHI Learning Private Limited.

6. Matrin, E. B., Morris, A. J., & Zhang, J. (1996). Process performance monitoring using multivariate statistical process control. *IEE Proceedings - Control Theory Application, 143*(2), 132-144.

7. Bankó, Z., Dobos, L., & Abony, J. (2011). Dynamic principal component analysis in multivariate time-series segmentation. *Conservation, Information, Evolution,* 11-24.

8. Tatara, E., & Cinar, A. (2002). An intelligent system for multivariate statistical process monitoring and diagnosis. *ISA Transactions,* 255-270.

9. MacGregor, J. F., & Kourti, T. (1995). Statistical process control of multivariate process. *Control Engineering Practice, 3*(3), 403-414.

10. Mina, J., & Verde, C. (2005). Fault detection using dynamic principal component analysis by average estimation. *International Conference on Electrical and Electronics Engineering* (pp. 374-377). Mexico City: ICEEE.

11. Ku, W., Storer, R., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems,* 179-196.