Department of Mechanical Engineering

Vibration-Based Multi-Fault Diagnosis for Centrifugal Pumps

Berli Paripurna Kamiel

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

February 2015

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: ………………………………………….

Date: …………………………

Abstract

Centrifugal pumps perform an important role in many industries. They are classified as one of the most critical rotating machines which ensures the continuation of many production processes. During their operation, the centrifugal pumps may fail which will subsequently lead to the interruption of the production line. The capability to detect premature failure of the pump not only ensures the continuation of the production line but also prevents more severe damage to the pump. Therefore, monitoring the health status of centrifugal pumps is essential in order to avoid unwanted stoppage of the pump which may further lead to the breakdown of the whole production process.

A reliable and low-cost maintenance system for the centrifugal pump is an important requirement for industries. This reason becomes the motivation for extensive research in searching for improved methods for the centrifugal pump fault diagnosis. There is a trend to utilise combinations of vibration signal processing and classification techniques to produce better and more reliable centrifugal pump fault diagnosis. The use of dimensionality reduction methods has also gained attention in past decades due to the increase in the number of monitored maintenance variables. The application of vibration signal processing, dimensionality reduction method and classification techniques is an open research area for further exploration in order to develop improved methods for centrifugal pump fault diagnosis.

A literature review of vibration signal processing, statistical features extraction method, dimensionality reduction technique, wavelet transform and machine learning for classification technique in fault diagnosis is presented in this thesis. Based on the findings from the literature review, a new combined method is proposed for centrifugal pump fault diagnosis. The proposed method consists of the use of statistical features, Symlet wavelet transform, Principal Component Analysis (PCA) and k-Nearest Neighbors.

Six statistical features (i.e., energy level, standard deviation, RMS, kurtosis, variance and crest factor) were extracted from the time domain vibration signal which were previously decomposed using Symlet wavelet transform. Three types of Symlet wavelet: sym4, sym8 and sym12 were investigated. The decomposition process was up to 5 levels using multi resolution analysis (MRA) which produced approximation coefficient (*cA*) and detailed coefficient (*cD*) components. For the purpose of the statistical features extraction, only the low frequency part from the decomposition results were undertaken, thus the *cA* parts were used for further analysis. The six statistical features were extracted from the *cA* parts of each sym-n wavelet transform (up to 5 levels). The resulting 30 features obtained from the feature extraction process, after normalizing, were used to develop the PCA models.

Once the PCA models were developed, they can then be used for dimensionality reduction and fault detection. The fault detection based on PCA models were performed by utilising T^2 and Q statistics and subsequently fault classification and identification were carried out using score matrices and k-Nearest Neighbors respectively.

Four accelerometers were mounted onto different locations of the centrifugal pump including the pump's inlet, volute, outlet and bearing housing. The accelerometers were used to collect the vibration data from seven different pump fault conditions including cavitation, impeller fault, bearing fault, blockage, impeller fault-cavitation, impeller fault-blockage and bearing fault-cavitation.

The results demonstrated that the proposed wavelet-PCA based method can be used for multi-fault diagnosis for centrifugal pumps with high performance where the lowest misdetection rate was 0.3% and the highest identification accuracy was 99.2% and visible separation of faults was evident. Although, different PCA models have to be employed in order to achieve the best performance.

Acknowledgements

I would like to express my deepest appreciation to many people for their support and valuable contribution for the completion of this doctoral thesis.

I would like to thank my supervisor Associate Professor Ian Howard for his excellent support and supervision in all phases of research and in the thesis writing stage. He always provided me both his time and academic expertise for directing me during my study.

I am also thankful to my Co-Supervisor Dr Kristoffer McKee for his extensive suggestion and valuable advice especially during the writing process and to my Associate-Supervisors Dr Rodney Entwistle and Dr Ilyas Mazhar for their advice and valuable discussion during the test rig preparation. Thanks also go to Associate-Professor Ramesh Narayanaswamy for chairing my thesis committee.

My sincere thanks is also extended to: Mr. David Collier from Mechanical Engineering Department workshop at Curtin University for his technical support on preparing the bearing test for the fault simulator test rig. Ms. Kim Yap, Ms. Margaret Brown, Ms. Bee Theng Lim, and Mr. Frankie Sia from the Mechanical Engineering Department office for their great administration support.

Grateful acknowledgement is also made for the DIKTI scholarship given by Indonesian Government through the Ministry of Education and Culture of the Republic of Indonesia which gave financial support during my doctoral study at Curtin University. I would also like to acknowledge my home university, Universitas Muhammadiyah Yogyakarta (UMY), for giving me support and the opportunity to pursue my doctoral degree.

Lastly but most importantly, I would like to thank my father and my mother for all the sacrifices that they have made on my behalf and my wife *Lely Lusmilasari*, and my children *Darin Zahra Salsabila* and *Ghaisan Rifqi Kamiel* for their love, support, encouragement and understanding during my study.

Table of Contents

Declaration ii

Abstract iii

Acknowledgements v

Table of Contents vi

List of Figures x

List of Tables xvi

1 Introduction 1

1.1 Maintenance Techniques 2

1.1.1 Breakdown Maintenance 2

1.1.2 Preventive Maintenance 2

1.1.3 Condition Based Maintenance 3

1.2 Condition Monitoring 5

1.2.1 Condition Monitoring Techniques 5

1.3 Thesis 6

1.4 Scientific Contribution 7

1.5 Thesis Outline 7

2 A Review of Centrifugal Pumps and Their Failure Modes 11

2.1 Centrifugal Pump 12

2.1.1 Radial Flow Centrifugal Pumps 13

2.1.2 Axial Flow Pumps 14

2.1.3 Mixed Flow Pumps 15

2.2 The Construction of Centrifugal Pumps 15

2.2.1 Pump Impeller 16

2.2.2 Shaft and bearings 17

2.2.3 Volute 18

2.3 Centrifugal pumps performance characteristics 20

2.4 Common failure modes in centrifugal pumps 23

2.4.1 Cavitation 24

2.4.2 Faulty Impeller 27

2.4.3 Faulty Bearing 28

2.4.4 Blockage 31

3 An Overview of Vibration Signal Analysis Methods....................................32

3.1 Time Domain Analysis 34

3.1.1 Standard Deviation 34

3.1.2 Kurtosis.. 34

3.1.3 RMS 35

3.1.4 Variance... 35

3.1.5 Skewness.. 36

3.2 Time-Synchronous Averaging (TSA)..36

3.3 Autoregressive Moving Average (ARMA)..37

3.4 Filter-Based Method 38

3.5 Stochastic Method and Blind Source Separation...39

3.6 Implementation of Time-domain Analysis of Vibration Signals for Fault Diagnosis of Machinery 39

3.7 Frequency Domain and Time-Frequency Domain Analysis 43

3.7.1 Fast Fourier Transform (FFT) 45

3.7.2 Envelope Analysis.. 48

3.7.3 The Use of Fast Fourier Transform for Fault Diagnosis 50

3.8 Wavelet Transforms...52

3.8.1 Continuous Wavelet Transform (CWT).. 57

3.8.2 Discrete Wavelet Transform (DWT) 57

3.8.3 Decomposing and Reconstructing a Signal Using the Wavelet Transform................ 59

3.8.4 Multi Resolution Analysis (MRA) Using Discrete Wavelet Transform 62

3.9 The Application of Wavelet Transform in Fault Analysis 66

3.10 Concluding Remarks...68

4 A Review of Principal Component Analysis in Fault Diagnosis: Techniques and Literature...69

4.1 Principal Component Analysis 70

4.1.1 The Mechanics of PCA... 71

4.1.2 Dimension Reduction 78

4.1.3 PCA Analysis for Fault Detection 80

4.2 Application of PCA in Feature Extraction and Fault Diagnosis..............................84

4.2.1 PCA-Based Extraction Techniques in Fault Diagnosis.................................... 89

4.3 Concluding Remarks..94

5 Principal Component Analysis and Wavelet-Based Framework for Fault Diagnosis 96

5.1 The Proposed Integrated Framework for the Centrifugal Pump Fault Diagnosis 98

5.2 kNN Classifier 102
5.3 The Wavelet Multi-Resolution Analysis 104
5.4 Proposed Feature Extraction 106
5.4.1 Energy Level (1st feature) 106
5.4.2 Standard Deviation (2nd feature) 107
5.4.3 RMS (3rd feature) 107
5.4.4 Kurtosis (4th feature) 107
5.4.5 Variance (5th feature) 108
5.4.6 Crest Factor (6th feature) 108
5.5 Example of Numerical Calculation of the Features 108
5.6 Integrated Framework for Fault Diagnosis Algorithm 114
5.6.1 Pre-Processing Training Data for PCA Modelling 116
5.6.2 PCA Modelling Process 117
5.6.3 Pre-Processing Testing Data 119
5.7 Fault Diagnosis Algorithm 120
5.8 Constructing the k-Nearest Neighbors (kNN) Rule 125
5.9 Evaluation of the Proposed Method's Performance 127
5.10 Test of the Proposed Method Using External Vibration Data 129

6 The Centrifugal Pump Test Rig and Vibration Data Acquisition 135

6.1 Centrifugal Pump Test Rig 136
6.2 Artificial Fault Conditions 138
6.3 Accelerometers and Data Acquisition Device 141
6.4 Data Structure 141

7 Results and Discussion of the Proposed Method 143

7.1 PCA Model Developed Using Vibration Signals Collected from Pump's Inlet 143
7.2 PCA Model Developed Using Vibration Signals Collected from Pump's Volute ..148
7.3 PCA Model Developed Using Vibration Signals Collected from Pump's Outlet ...152
7.4 PCA Model Developed Using Vibration Signals Collected from Bearing Housing 155
7.5 Summary of PCA Models Obtained from Channel 1 to 4 158
7.6 Fault Detection and Diagnosis: Evaluation and Results 159
7.6.1 Fault Diagnosis Results Using PCA Model 1 161
7.6.2 Fault Diagnosis Results Using PCA Model 2 169
7.6.3 Fault Diagnosis Results Using PCA Model 3 177
7.6.4 Fault Diagnosis Results Using PCA Model 4 185
7.7 Performance Comparison of the PCA models 192

8 Conclusions and Suggestions for Future Work 198

8.1 Conclusions 198
8.2 Suggestions for Future Work 202

References 205

Appendix A − MATLAB Script Code for Data Acquisition Using NI-9234.....215

Appendix B − Visualisation of Correlation Matrices for All Channels Using sym4, sym8 and sym12 218

Appendix C – Integrated Framework for Fault Diagnosis Code 225

List of Figures

Figure 2.1 Major parts of centrifugal pump [17]..12

Figure 2.2 The major centrifugal pump classifications [19] 13

Figure 2.3 Radial flow pump [20] 14

Figure 2.4 Axial flow pump [22]..15

Figure 2.5 Open impeller [23]...16

Figure 2.6 Semi open impeller [24] 17

Figure 2.7 Closed-Impeller [24] 17

Figure 2.8 Integral shaft bearing [26] 19

Figure 2.9 The cut-water in a volute [27]..19

Figure 2.10 Common type of volutes in centrifugal pumps [28] 20

Figure 2.11 Total static head, static discharge head (hd), and static suction head (hs) [30]..22

Figure 2.12 Performance curves for a centrifugal pump [31]. 23

Figure 2.13 Cavitation in centrifugal pump: no cavitation (left), cavitation (right) [34]..25

Figure 2.14 Worn impeller due to cavitation [42] 27

Figure 2.15 Inner race damage of ball bearing [45] 29

Figure 2.16 Build-up of rags in a screw centrifugal pump [46] 31

Figure 3.1 Signal processing techniques commonly used in CM [50] 33

Figure 3.2 Number of calculation required for computing DFT using the FFT vs direct calculation [105] 47

Figure 3.3 Procedure for the envelope analysis technique [107] 49

Figure 3.4 The fundamental principle of wavelet transform [119] 53

Figure 3.5 Basis function and time-frequency resolution plane of (a) the Fourier transform and (b) the Daubechies wavelet [120] 55

Figure 3.6 Symlet wavelet function 58

Figure 3.7 Symlet wavelet scaling function 59

Figure 3.8 Decomposition and reconstruction process 60

Figure 3.9 Two levels decomposition using symlet 8 wavelet transform..................61

Figure 3.10 Five levels decomposition using symlet 8 wavelet transform 62

Figure 3.11 Principle of multi resolution analysis...63

Figure 3.12 Approximated parts, *a1-a5*, of a noisy sinusoidal signal 65

Figure 3.13 Detailed parts, *d1-d5*, of a noisy sinusoidal signal 66

Figure 4.1 Two dimensions artificial data, (a) uncorrelated, (b) correlated. 76

Figure 4.2 Two principal components plotted with the original data........................77

Figure 4.3 The transformed data projected on the two principal components. 77

Figure 4.4 Graphical representation of the experimental data as summation of approximation and error using PCA model 79

Figure 4.5 T^2 and Q-statistic in PCA model [138] 83

Figure 5.1 Stage 1 of the proposed integrated framework (feature extraction) 99

Figure 5.2 Stage 2 of the proposed integrated framework (PCA modelling) 100

Figure 5.3 Stage 3, fault detection process...101

Figure 5.4 Stage 4, fault identification and classification process 102

Figure 5.5 Example of kNN classification...103

Figure 5.6 Wavelet MRA decomposition up to 5 levels..105

Figure 5.7 Separation of frequency band up to 5 levels 105

Figure 5.8 The six features obtained from each cA parts 106

Figure 5.9 Wavelet decomposition results for normal condition, s (original signal), a_1 – a_5 (approximate parts level 1- 5)...109

Figure 5.10 Wavelet decomposition results for impeller fault, s (original signal), a_1 – a_5 (approximate parts level 1- 5) 110

Figure 5.11 Plot of example feature value for normal condition and impeller fault (part 1)...111

Figure 5.12 Plot of example feature value for normal condition and impeller fault (part 2)...111

Figure 5.13 Numerical values of feature number 1 to 30 for normal condition, single-fault, and multi-faults (part 1) 112

Figure 5.14 Numerical values of feature number 1 to 30 for normal condition, single-fault, and multi-faults (part 2) 113

Figure 5.15 Integrated framework algorithm (Part A)...115

Figure 5.16 PCA model building process 119

Figure 5.17 Integrated framework algorithm (Part B) 121

Figure 5.18 Example of T^2 and Q-statistic plot 123

Figure 5.19 Example plot of scores (a) on PC1 and PC2, (b) on PC1, PC2, and PC3 124

Figure 5.20 k-Nearest Neighbors construction..125

Figure 5.21 Combined score matrix in kNN rule 127

Figure 5.22 Evaluation of fault identification performance...................................128

Figure 5.23 Time waveform of vibration signal of bearing 1, data set # 10 (at the beginning region of test), data set #450 (at the middle region of test), data set #650 (sign of fault observed) and data set #890 (at the end region of test) 130

Figure 5.24 Fault detection (outer race fault) using T^2 and Q-statistic of external vibration data...132

Figure 5.25 Plot of scores on PC1 and PC2 133

Figure 5.26 Performance of k-Nearest Neighbors (kNN) rule...............................134

Figure 6.1 Data acquisition process...136

Figure 6.2 Centrifugal pump test rig set up...137

Figure 6.3 Detail of the accelerometer (channel) locations 137

Figure 6.4 Cavitation condition observed in the centrifugal pump 138

Figure 6.5 Faulty impeller..139

Figure 6.6 Shaft bearing...140

Figure 6.7 Faulty shaft bearing with small localized spalling................................140

Figure 7.1 Time waveform of normal (no fault) and faulty components acquired from channel 1 (data set #15) 144

Figure 7.2 Number of principal components retained in the model based on 95% of the variance. (a) 7 PCs retained in the model by using sym4, (b) 8 PCs retained using sym8 and sym12 146

Figure 7.3 Three PCA models obtained from channel 1..147

Figure 7.4 Correlation between principal components (PCs) and the features of channel 1 (using sym8 decomposition) 148

Figure 7.5 Time waveform of normal (no fault) and faulty components acquired from channel 2 (data set #15) 149

Figure 7.6 Seven PCs retained in the model by using sym4, sym8, and sym12 (based on 95% of the variance)...150

Figure 7.7 PCA models of channel 2 151

Figure 7.8 Correlation between principal components (PCs) and the features of channel 2 (using sym8 decomposition) 151

Figure 7.9 Time waveform of normal (no fault) and faulty components acquired from channel 3 (data set #15) 152

Figure 7.10 Seven PCs retained in the model by using sym4, sym8, and sym12 (based on 95% of the variance) 153

Figure 7.11 PCA models of channel 3 154

Figure 7.12 Correlation between principal components (PCs) and the features of channel 3 (using sym8 decomposition) 154

Figure 7.13 Time waveform of normal (no fault) and faulty components acquired from channel 4 (data set #15) 155

Figure 7.14 Number of principal components retained in the model based on 95% of the variance. (a) 5 PCs retained in the model by using sym4, (b) 6 PCs retained using sym8 and sym12 157

Figure 7.15 PCA models obtained from channel 4 157

Figure 7.16 Correlation between principal components (PCs) and the features of channel 4 (using sym8 decomposition) 158

Figure 7.17 PCA models evaluation process 160

Figure 7.18 Fault classification and fault identification evaluation process 161

Figure 7.19 T^2 chart for normal and all faulty conditions obtained using PCA model 1a, model 1b, and model 1c 162

Figure 7.20 Q chart for normal and all faulty conditions obtained using PCA model 1a, model 1b, and model 1c 163

Figure 7.21 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1a. Note that exhibit (b) zooms into the green ellipse area in exhibit (a) 165

Figure 7.22 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1b. Note that exhibit (b) zooms into the red ellipse area in exhibit (a) 165

Figure 7.23 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a). 166

Figure 7.24 Identification accuracy comparison of the PCA model 1a using 1 to 7 PCs. 167

Figure 7.25 Identification accuracy comparison of the PCA model 1b using 1 to 8 PCs. 168

Figure 7.26 Identification accuracy comparison of the PCA model 1c using 1 to 8 PCs. 169

Figure 7.27 T^2 chart for normal and all faulty conditions obtained using PCA model 2a, model 2b, and model 2c 170

Figure 7.28 Q chart for normal and all faulty conditions obtained using PCA model 2a, model 2b, and model 2c 171

Figure 7.29 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2a. Note that exhibit (b) zooms into the green ellipse area in exhibit (a) 172

Figure 7.30 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2b. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a) 173

Figure 7.31 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a) 173

Figure 7.32 Identification accuracy comparison of the PCA model 2a using 1 to 7 PCs. 174

Figure 7.33 Identification accuracy comparison of the PCA model 2b using 1 to 7 PCs. 175

Figure 7.34 Identification accuracy comparison of the PCA model 2c using 1 to 7 PCs. 176

Figure 7.35 T^2 chart for normal and all faulty conditions obtained using PCA model 3a, model 3b, and model 3c 178

Figure 7.36 Q chart for normal and all faulty conditions obtained using PCA model 3a, model 3b, and model 3c 179

Figure 7.37 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3a. Note that exhibit (b) zooms into the red ellipse area in exhibit (a) 180

Figure 7.38 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3b. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a) 181

Figure 7.39 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a) 181

Figure 7.40 Identification accuracy comparison of the PCA model 3a using 1 to 7 PCs. 182

Figure 7.41 Identification accuracy comparison of the PCA model 3b using 1 to 7 PCs. 183

Figure 7.42 Identification accuracy comparison of the PCA model 3c using 1 to 7 PCs. 184

Figure 7.43 T^2 chart for normal and all faulty conditions obtained using PCA model 4a, model 4b, and model 4c 186

Figure 7.44 Q chart for normal and all faulty conditions obtained using PCA model 4a, model 4b, and model 4c 187

Figure 7.45 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4a. Note that exhibit (b) zooms into the red ellipse area in exhibit (a) 188

Figure 7.46 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4b. Note that exhibit (b) zooms into the red ellipse area in exhibit (a) 188

Figure 7.47 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4c. Note that exhibit (b) zooms into the red ellipse area in exhibit (a) 189

Figure 7.48 Identification accuracy comparison of the PCA model 4a using 1 to 5 PCs. 190

Figure 7.49 Identification accuracy comparison of the PCA model 4b using 1 to 6 PCs. 191

Figure 7.50 Identification accuracy comparison of the PCA model 4c using 1 to 6 PCs. 191

Figure 7.51 Misdetection rate of all PCA models from channel 1 to 4 193

Figure 7.52 Average misdetection rate from channel 1 to 4 194

Figure 7.53 Identification accuracy of the best PCA models in each channel 195

List of Tables

Table 2.1 Stages of degradation of rolling element bearings [29]............................30

Table 3.1 Overview of frequency-domain techniques [92]......................................44

Table 3.2 Comparison of computational cost between “direct” and FFT [104] 47

Table 3.3 Frequency sub-bands of five decomposition levels using DWT MRA......64

Table 4.1 Summary of the use of feature extraction for PCA applications 85

Table 5.1 Example of kNN parameters..126

Table 7.1 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 1)...145

Table 7.2 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 2)...150

Table 7.3 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 3)...153

Table 7.4 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 4)...156

Table 7.5 PCA models obtained from channel 1 to 4 159

Table 7.6 Comparison of fault detection accuracy of PCA model in channel1 164

Table 7.7 Comparison of fault detection accuracy of PCA model in channel 2 172

Table 7.8 Comparison of fault detection accuracy of PCA model in channel 3 179

Table 7.9 Comparison of fault detection accuracy of PCA model in channel 4 187

Table 7.10 Confusion matrix of PCA model 1b employing 20 neighbors k............196

Table 7.11 Confusion matrix of PCA model 2b employing 20 neighbors k............196

Table 7.12 Confusion matrix of PCA model 3c employing 14 neighbors k............197

Table 7.13 Confusion matrix of PCA model 4c employing 19 neighbors k............197

CHAPTER ONE

1 Introduction

A centrifugal pump is a type of rotating machinery that is commonly used in many industries such as chemical plants, wastewater treatment, power generation, sugar refining, food industries, oil and gas and many more. It is often located on the critical path of the production line which, during its operation, may experience failures that can potentially cause disruption to the production processes. The capability to detect those failures at an early stage not only assures the continuity of production processes but also promptly avoids more severe damage of the machines. A reliable maintenance system therefore plays a critical role to keep such industrial machinery in a good operational condition.

Along with the increasing complexity of the industrial equipment due to technological development, the associated machinery requires a more advanced and reliable maintenance strategy in managing the production process. A high performance maintenance system is then required in order to ensure the reliability of both the production machines and the production process while keeping low maintenance cost, which eventually may increase the profitability and competitiveness of the companies.

The need to maintain high level of operational safety, providing the reliability of production equipment, improving quality of product, and increasing productivity leads the industries to improve their existing maintenance system [1]. In addition, the maintenance system also ensures modern industries operate at low-risk impact to the environment while achieving maximum productivity.

However, a considerable amount of money must be spent on the maintenance tasks in particular industries which have modern-intensive equipment. Furthermore, the application of automation and computerization in many modern plants, utilization of artificial intelligence and unmanned equipment has increased the maintenance cost significantly. As a consequence, the implementation of a reliable and efficient maintenance system is crucial so that the maintenance cost may be set at an optimum level.

Fault monitoring and diagnosis is an important part of the maintenance processes. This is needed to monitor and diagnose the operational condition of machines with rotating components and aims to prevent unscheduled breakdown of machines caused by the failure of rotating components. An effective and efficient fault diagnosis method which is able to detect a failure at an early stage would be very useful in designing a maintenance strategy.

In conjunction with the importance of fault diagnosis in the maintenance scheme, the research in the fault diagnosis area becomes more attractive and challenging. Many diagnosis approaches have been proposed which aims to establish a more accurate and efficient fault diagnosis scheme.

In the following section, a brief discussion of maintenance techniques is presented to put the thesis in context.

1.1 Maintenance Techniques

In general, maintenance techniques may be divided into three major categories: Breakdown Maintenance, Preventive Maintenance, and Condition Based Maintenance (CBM) [2].

1.1.1 Breakdown Maintenance

The earliest maintenance technique is breakdown maintenance, where maintenance action will be taken only after equipment failure. With this technique, unscheduled maintenance will often occur which can cause high maintenance cost and unpredictable breakdown of machinery [3]. As a result, a scheduled interruption of production is not possible.

1.1.2 Preventive Maintenance

The concept of preventive maintenance consists of maintenance activities prior to the failure of the components [4]. Preventive maintenance is time-based maintenance where periodic time intervals are used to perform maintenance regardless of condition of the components being monitored. Periodic machine inspection and maintenance may include oil replacement, lubrication, component replacement at regular time intervals, and calibration, without considering the health status of machinery.

Preventive maintenance aims to reduce the frequency of machine downtime. This technique decreases the failure cost and production loss, and increases product quality [5].

In the industrial application, preventive maintenance can be applied either based on experience or original equipment manufacturer (OEM) recommendations. Preventive maintenance based on experience is a traditional practice which is usually conducted in a regular time interval [6]. With this approach, no standard procedures are followed. Appropriate maintenance actions are taken based on the previous experience of technicians and engineers. The main drawback of this approach is its high dependency on the experienced person.

Preventive maintenance based on OEM recommendation is performed at a fixed time by following manual instructions. This approach, however, is not suitable if one wants to minimise the operation cost and maximise machine performance [7].

Moreover, the preventive maintenance technique potentially does unnecessary maintenance actions to the components that makes the maintenance cost higher. Eventually, it becomes an extensive expense for many industries [2].

1.1.3 Condition Based Maintenance

Condition based maintenance (CBM) is considered a more efficient approach than the two previous techniques. CBM is a maintenance system based on condition monitoring. This technique avoids unnecessary maintenance actions by only undertaking maintenance works if there are indications of abnormal behaviour of the components being monitored. If implemented properly, CBM can significantly reduce maintenance costs since it prevents unnecessary scheduled preventive maintenance tasks.

The implementation of an effective CBM consists of three stages [2]:

1. Data acquisition; collect and store health condition data from system being monitored

2. Data processing; include pre-processing, filtering and features extraction of data collected from stage 1
3. Maintenance decision-making; provide efficient maintenance recommendation based on the machine health condition assessment.

Data acquisition is an essential step in CBM implementation. This step acquires useful data related to the system health. In a CBM approach, there are two categories of data: condition monitoring data and event data. Condition monitoring data is a collection of the information about the health status of the system, while event data provides information surrounding events such as a minor repair, breakdown, spare part change, oil change, and installation.

Condition monitoring data are available in various forms such as electric current, acoustic emission, vibration, temperature, pressure, and oil analysis data. Various sensors, such as acoustic emission sensors, accelerometers, temperature transducers, and pressure transducers have been developed to acquire different types of data [8].

There are two essential steps in data processing namely data cleaning and data analysis. For the event data category, data cleaning is a crucial step since this step ensures the event data is error-free for further analysis. Many factors cause data errors including human factor and sensor faults.

Data analysis is the next step of data processing. Various techniques, models, and algorithms are available to interpret the data. The techniques, models and algorithms being chosen for a particular case depend on the characteristics of the data.

Maintenance decision making is the final step of CBM. Diagnostic and prognostic aspects are two categories in the maintenance decision. Diagnostics is associated with detection, isolation and identification of a fault [2]. Fault detection is a scheme to identify the presence of a fault in the system, fault isolation is a method to pinpoint the location of a fault, and fault identification is a procedure to decide a fault type. Meanwhile prognostics is a method for fault prediction before the fault actually occurs within the system being monitored. It uses systematic steps to predict impending failure and the remaining useful life of components.

1.2 Condition Monitoring

Condition monitoring (CM) is the core of CBM. Generally speaking, condition monitoring is a process of gathering or collecting information/signals related to the machine's health status. Information/signals can be continuously or periodically monitored using appropriate sensors or indicators [9]. Thus, maintenance actions such as repairs or replacements are immediately taken before failure occurs.

The CM process may be carried out in two methods: on-line and off-line [5]. On-line monitoring is implemented simultaneously with the data collection, while off-line monitoring is performed after the data collection process. As mentioned earlier, CM process may be executed either continuously or periodically. Continuous monitoring is operated automatically and continuously using acquisition sensors, such as acoustic emission sensors and accelerometers, whilst periodical monitoring is performed in particular time ranges such as weekly using portable data acquisition devices like a vibrometer, acoustic emission meter, and vibration pens.

1.2.1 Condition Monitoring Techniques

Most equipment gives certain signs, conditions or indications before they fail [10]. Many CM techniques have been developed to monitor equipment conditions. Those techniques utilize interdisciplinary fields such as vibration and noise, dynamics, tribology and non-destructive testing (NDT).

Vibration monitoring is the most popular CM technique used especially for rotating equipment [11]. This technique is associated with non-destructive testing and monitoring the operational characteristics of the equipment. The equipment health status is determined by data sensor devices such as accelerometers, to detect changes in the vibration signature that may indicate damage or deterioration of components. In addition, vibration monitoring may be carried out either continuously or periodically.

Another CM technique is sound or acoustic monitoring. This technique has a similarity with the vibration monitoring. The difference between those methods is based on the data acquisition technique, where vibration sensors record local displacements while acoustic sensors record sound of the equipment.

Oil-analysis is another CM technique. As its name suggests, the technique assesses the quality of the oil to decide the wear condition of the internal parts such as journal bearings and gears. Several other CM techniques available in the literature include temperature, pressure and electric current condition monitoring.

This thesis focuses on vibration monitoring. The analysis of the health of the equipment is based on the vibration information collected when a machine is in operation. While in running condition, a machine generates vibration waveforms with unique signatures and the signatures change with operational condition.

1.3 Thesis

The thesis discusses the development of a fault diagnosis method based on vibration signatures in condition monitoring. The proposed method consists of three stages namely fault detection, fault classification, and fault identification. This is carried out by employing a combination of statistical parameters, Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and k-Nearest Neighbors (kNN).

The main objective of the thesis is to develop a vibration-based multi-fault diagnosis method for a centrifugal pump. The method was based on the statistical features extracted from the decomposition of time-domain vibration signals where the decomposition was employed through the use of discrete wavelet transform. In this research, the mother wavelet Symlet 4, 8, and 12 (sym4, sym8, sym12) were used and six statistical features were extracted from the decomposed signals. Subsequently, the principal component analysis was used to extract the most prominent features by re-expressing the original dimension into a new subspace with a lower dimension. The fault detection stage was performed using T^2-statistic and Q-statistic while fault classification was carried out by plotting the first three principal components in three dimensional space. The k-Nearest Neighbors method was then used for fault identification.

The thesis reports on the generation of the PCA models from four channels corresponding to four mounting locations of accelerometers on a centrifugal pump. The accelerometers were mounted on the pump's inlet (channel 1), pump's volute (channel 2), pump's outlet (channel 3), and pump's bearing house (channel 4). The vibration data collected from each of the channels were decomposed using a wavelet transform and subsequently six statistical features were extracted. The generated features were then used to build the PCA models for all channels. The purpose of building the PCA models was for the detection, classification and identification of faults in the centrifugal pump. The types of faults considered were cavitation, impeller fault, bearing fault, blockage and some of their combinations. The performance of each Symlet wavelet was evaluated in term of its accuracy to detect, classify and identify the fault.

1.4 Scientific Contribution

There are many works in the literature relating to finding better methods for fault detection in rotating machinery. A centrifugal pump is a type of rotating machinery which performs an important role in industries; therefore there is a need to find an effective and reliable method to detect its failure.

In order to find a superior method, many techniques have been developed for fault diagnosis in a centrifugal pump. Principal component analysis (PCA) is one of the popular methods for fault diagnosis in rotating machinery. However, the use of PCA has not been fully investigated for fault diagnosis in a centrifugal pump. In this thesis, the use of PCA is combined with the wavelet transform using Symlet wavelet family and statistical feature extraction.

The study aims to contribute a new method for fault diagnosis in a centrifugal pump by using a combination of the statistical parameters, wavelet transform and the PCA model. It also contributes to add references for the selection of the Symlet wavelet type for decomposing the time-domain vibration signal for a centrifugal pump fault diagnosis. Furthermore, the extended use of the T^2 and Q-statistics for fault detection, scores of principal components (PCs) for fault classification, and k-Nearest Neighbors (kNN) for fault identification also provides additional contributions to knowledge.

1.5 Thesis Outline

The outline of the thesis is as follow,

Chapter One

This chapter describes a general introduction of the importance of maintenance systems in the industries. It gives an overview of the need of maintenance systems implemented in the industries in order to achieve a high level of reliability and availability of production equipment while keeping the maintenance cost at a reasonable level. It describes the division of the maintenance techniques, defines the fundamental concepts, and explains the advantages and disadvantages of each maintenance technique. It also describes the motivation and objectives of the thesis. The final part of the chapter outlines the structure and content and the scientific contribution of the thesis.

Chapter Two

The chapter describes the division of centrifugal pumps commonly found in the literature and also reviews their construction, characteristics, and their failure modes. The mechanical construction includes the description of the impeller, pump volute and bearing. It then considers the common failure modes in centrifugal pumps such as cavitation, impeller fault, bearing fault and blockage.

Chapter Three

This chapter discuss an overview of the vibration signal analysis techniques commonly applied in fault analysis. These techniques include time-domain, frequency-domain, and time-frequency-domain analysis. Several popular techniques in time-domain analysis area and a thorough review of the implementation of time-domain analysis in fault diagnosis are presented. The use of the Fast Fourier Transform (FFT) in frequency-domain analysis is elaborated with an extensive review of its application for fault diagnosis. Finally, the chapter describes the wavelet transform technique and its application in fault analysis. Some remarks corresponding to important findings from the literature review are also presented.

Chapter Four

This chapter explains the fundamental concept of Principal Component Analysis (PCA) and its use in fault diagnosis. The first part of the chapter thoroughly explains the mechanics of PCA, dimension reduction using PCA, and the use of PCA for fault detection. The second part reviews the application of PCA in feature extraction and fault diagnosis. An extensive review of the use of a combination of PCA with other techniques for fault analysis is also presented. This chapter concludes with important findings related to the PCA-based fault diagnosis in the centrifugal pump.

Chapter Five

This chapter proposes a new integrated framework based on the Wavelet-PCA for fault diagnosis of the centrifugal pump. The algorithms of the proposed method are presented in detail and explained thoroughly. It describes the pre-processing training data, PCA modelling process, and pre-processing of the testing data. It explains the construction of the k-Nearest Neighbors rule and evaluates the proposed method’s performance. The final part of the chapter presents a test of the proposed method using external vibration data.

Chapter Six

This chapter describes the centrifugal test rig and vibration data acquisition process. It explains the configuration of a Spectra Quest Machinery Fault Simulator which was set up with a centrifugal pump and depicts the mounting locations of the accelerometers. It also describes the artificial component fault, the accelerometers and the acquisition device used in the experiment. Finally, the chapter concludes with the data structure of the vibration signal.

Chapter Seven

This chapter reports the analysis of the experimental vibration data collected from the centrifugal pump test rig using the proposed method. The PCA models generated from all channels are evaluated and compared in order to conclude the best PCA model in each channel. The accuracy performance is also examined to select the PCA model with the highest identification accuracy.

Chapter Eight

This chapter presents the conclusions and suggestions for future work. It presents the major research findings from the fault diagnosis of a centrifugal pump using the proposed method which combines the statistical parameter, wavelet transform, and PCA. It also reviews the objectives and describes the achievements of the study. The second part of the chapter presents the recommendations for future work.

The next chapter describes mechanical construction and characteristic of the centrifugal pumps. It also reviews failure modes such as cavitation, impeller fault, bearing fault and blockage.

CHAPTER TWO

2 A Review of Centrifugal Pumps and Their Failure Modes

The use of pumps is essential in most industrial plants like power generation, oil and gas, water treatment, petrochemical, pharmacy, agriculture and fertilizers. A pump is a mechanical device used to transport fluids (varying from clean water to hazardous chemicals) from one place to another. Pumps operate by mechanical action and consume energy to perform mechanical work.

Various types of pumps are available in the market and widely used in many areas of application. All pumps can be categorized into [12]: *dynamics pumps*, where the fluid velocity inside the pump is increased by adding the energy continuously, and *displacement pumps*, where the addition of energy to moveable fluid boundaries is performed by applying the force periodically. The subdivision of the dynamics pumps may consist of some of centrifugal pump types and other special-effect pumps. Meanwhile, the displacement pumps include rotary and reciprocating pump types.

Centrifugal pumps are a type of rotordynamic pump where the flow velocity is increased by adding kinetic energy. A centrifugal pump is classified into velocity pumps which increases the flow rate and pressure of fluid using a rotating impeller. The use of centrifugal pumps in the new pump market reaches 64% [13]. The high demand on centrifugal pumps are caused by their high efficiency, simple design, continuous flow rate, vast array of capacity, and ease of maintenance and operation [14]. Centrifugal pumps have relatively fewer moving parts which tend to be relatively small with less weight than other pumps. The superiority of centrifugal pumps is also because of their capability to handle liquids containing dirt, abrasives, solids, etc [15].

Centrifugal pumps can fail during their service due to problems that arise mainly from hydraulic and mechanical failure [16]. Hydraulic failures include cavitation, pressure pulsation, radial thrust, axial thrust and suction and discharge recirculation.

Meanwhile, mechanical failures consist of bearing failure, seal failure, lubrication failure, excessive vibration, and fatigue.

The following sections discuss some important aspects on mechanical construction, characteristics, and failure modes of centrifugal pumps.

2.1 Centrifugal Pump

A centrifugal pump is a mechanical device which uses a rotating impeller to accelerate fluid by converting electrical energy into kinetic energy of the fluid. The volute pump as depicted in Figure 2.1 is the most common centrifugal pump type which is widely implemented in industries. The stationary volute (or diffuser) converts the kinetic energy of the fluid into fluid pressure. The inlet pump (or suction nozzle) delivers the fluid into the pump's impeller eye due to low-pressure area at the suction eye. The low-pressure area is created because the rotation of the impeller pushes the fluid sitting between vanes outward into the volute or diffuser. This outward movement creates a vacuum at the impellers eye that continuously draws fluid into the pump.

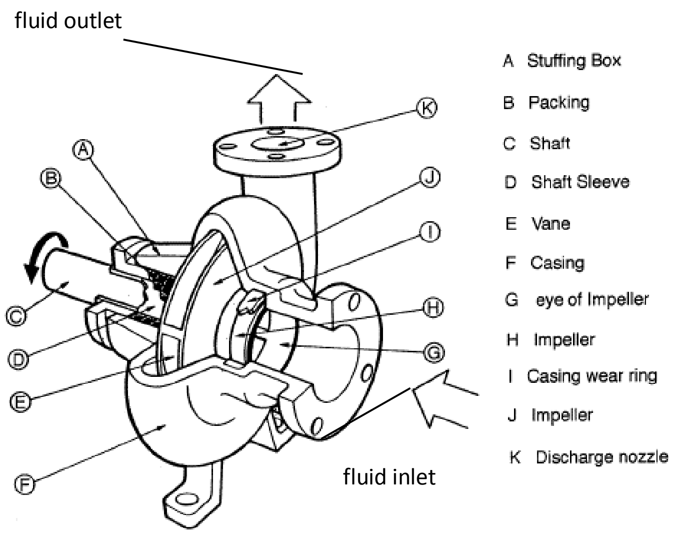


Figure 2.1 Major parts of centrifugal pump [17]

A centrifugal pump consist of several components such as the rotating element which consist of the shaft and the impeller and the stationary element, which consists of the casing, casing box, and bearing housing. In general, centrifugal pumps can be grouped based on their design. Bachus and Custodio [18] suggests classifying centrifugal pumps as either axial-flow pumps, radial-flow pumps, mixed-flow pumps and turbines as shown in Figure 2.2. Other classifications, however, may be present in the literatures such groupings based on single-stage, double-stage, or multi-stage; single-suction or double suction.

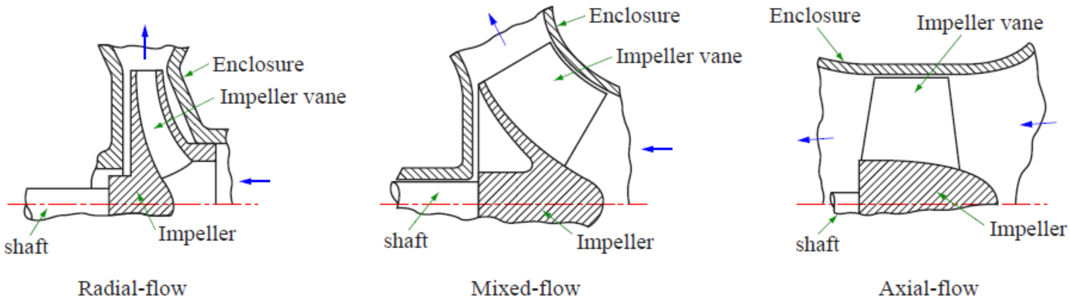


Figure 2.2 The major centrifugal pump classifications [19]

2.1.1 Radial Flow Centrifugal Pumps

Radial flow centrifugal pumps are the most frequently used centrifugal pump types in many application areas [13]. The fluid in the radial flow centrifugal pump enters along the axial plane and exits the pump radially with respect to the impeller shaft, as shown in Figure 2.2 and Figure 2.3. As opposed to axial pumps, in which fluid exits the pump axially, radial flow pumps have higher centrifugal force due to flow deflections in the impeller. This makes a radial flow pump have higher head pressure, but with smaller capacity flow.

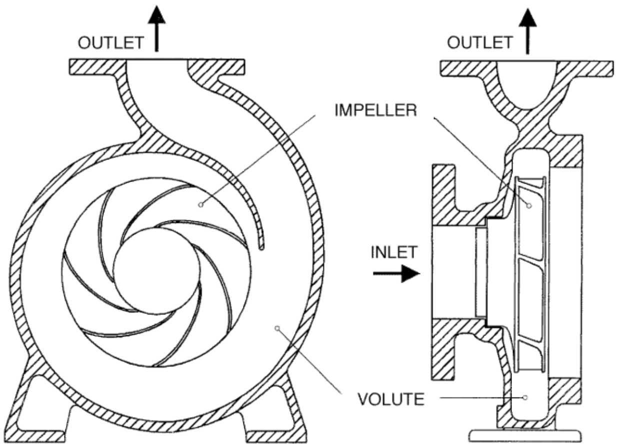


Figure 2.3 Radial flow pump [20]

The radial flow pump, due to its design, can be used for applications where it requires pumping of raw wastes. The materials such as rags and trash can be allowed to be present in the flow and do not clog the pump [21].

2.1.2 Axial Flow Pumps

The axial flow pump as shown in Figure 2.4, also known as a propeller pump, is another popular type of pump which is constructed by a propeller inside a pipe. In this type of pump, the blades of the propeller develop the pressure by passing the fluid on them. The fluid enters the pump in the axial direction along the shaft of the propeller, so that each fluid particle does not change radial direction during the flow through the pump. The fluid then exits the impeller nearly axially.

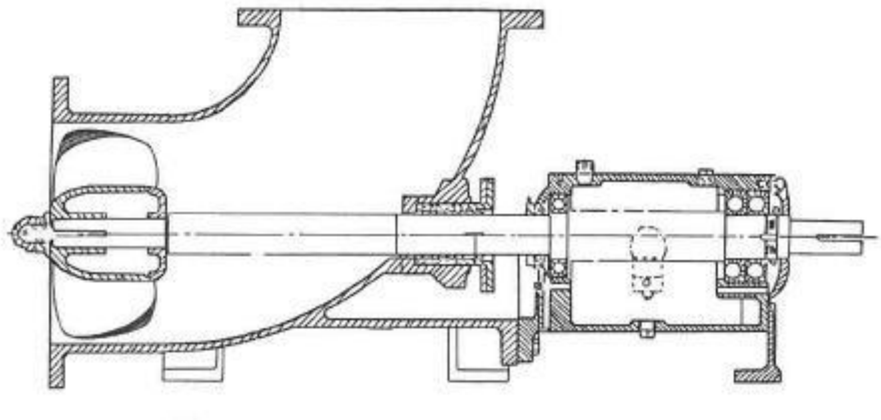


Figure 2.4 Axial flow pump [22]

An axial pump has a high relative flow rate with low head at the inlet end. In some models, the pitch of the propeller is adjustable to allow the pump to achieve peak efficiency. The most common applications are in handling sewage from industrial plants and commercial sites.

2.1.3 Mixed Flow Pumps

Mixed flow pumps have a unique design that is between a radial flow and axial flow pumps which gives the operating characteristics a combination of both. The fluid encounters both axial force and radial acceleration from the impeller. The fluid exits the impeller in the direction of 0 to 90 degrees with respect to the axial direction. The construction of the mixed flow pump gives several mechanical advantages such as higher pressure and higher discharge compared to axial flow and radial flow pumps respectively.

2.2 The Construction of Centrifugal Pumps

There are a large number of centrifugal pump designs available for any given application. In this section, a general description of the mechanical components of centrifugal pumps is discussed.

2.2.1 Pump Impeller

The pump impeller is a rotating part in a centrifugal pump which converts electrical energy from the motor into kinetic energy in the fluid by accelerating the fluid radially with respect to impeller shaft. The impeller is usually made of steel, bronze, aluminium, brass or plastic. The shape, size and speed of the impeller are the influential factors that determine pump performance. As a consequence, any kind of impeller faults could cause poor performance and a decrease in efficiency of the pump. Generally, impellers can be classified into three categories i.e., *open impellers*, *semi-open impellers*, and *closed* (or *enclosed*) *impellers*.

An open impeller, as shown in Figure 2.5, consists only of blades attached directly to a shaft. The blades are usually short and structurally weaker than either semi-closed or closed impellers. This type of impeller has a low efficiency and generally is used only in small and low energy pumps. The advantage of this impeller is that it is suitable for applications where clog resistance is required.

Figure 2.5 Open impeller [23]

Figure 2.6 shows a semi-open impeller which has a circular plate (shroud) attached to one side of the blades. The shroud is used to stiffen the blades and adds structural strength. Semi-open impellers are commonly used in medium-diameter pumps and with fluids containing small amounts of suspended solids. This type of impeller has higher efficiency than the open impeller.

Figure 2.6 Semi open impeller [24]

The closed-impeller, as shown in Figure 2.7, has a circular plate attached to both side of the blades for maximum strength. They are used in large pumps and can be operated with liquids containing suspended-solids for service without clogging. This type of impeller is widely used for centrifugal pumps handling clear fluids. The pumps with closed-impeller rely on close-clearance wear rings on the casing and on the impeller. The wear rings are used to separate the inlet pressure from the pressure within the pump, reduce axial loads, and maintain pump efficiency.

Figure 2.7 Closed-Impeller [24]

2.2.2 Shaft and bearings

A shaft is a major component in a centrifugal pump which delivers torque from the motor to the impeller mounted on the shaft. The pump shafts are commonly made of carbon steel and stainless steel. It is subjected to several stresses such as torsional, shear, flexural, tensile, etc. Among these stresses, torsional stress is generally most dominant and is used as a basic factor to determine the shaft diameter. Another important consideration in determining the pump's shaft diameter is the operating speed. If the operating speed is at its critical speed, it can result in excessive and destructive rotor vibration. One way to avoid the vibration resonance is to change the shaft size in order to change the rotor natural frequency.

Ball bearings are the most commonly used type of bearings in small and medium sized centrifugal pumps because of their high speed capability and low friction. Ball bearings have many configurations, such as single and double row with various contact angles which can handle radial loads, combined radial and axial loads, and purely axial loads. Ball bearings are considered to have a relatively low load rating because the small contact area results in high contact stress for a given load [25].

Integral shaft bearings (or water pump bearings) are usually used in water pump applications. They are double row bearings with a simplified structure and, in contrast to conventional double row bearings, do not have inner rings for the two supporting bearings. The grooves for the inner ring are machined directly into the surface of the shaft and the outer rings are made to a unity. The two sides of the bearing are closed by rubber seals. Figure 2.8 shows typical integral shaft bearing. Compared to conventional ball bearings, in the same loading capacity, the radial dimensions are usually smaller than those of the same kind. Meanwhile, in the same radial dimensions, the loading capacity of the integral shaft bearing is usually bigger.

2.2.3 Volute

The volute in a centrifugal pump is the casing where the impeller is housed. The fluid being pumped by the impeller enters the volute and decreases its rate of flow. A volute has a unique shape, called a curved funnel, where its cross-sectional area gradually increases as it approaches the discharge.

As a consequence of increasing the cross-sectional area, the speed of the fluid is decreased and its pressure is increased. The volute also helps to balance the hydraulic pressure on the pump shaft. The wall separating the curved funnel and the discharge nozzle portion is called the tongue of the volute or the cut-water, as depicted in Figure 2.9.

Figure 2.8   Integral shaft bearing [26]

Generally, there are four types of volute commonly used in centrifugal pumps, i.e., single volute, double volute, volute diffuser, and circular volute as illustrated in Figure 2.10.

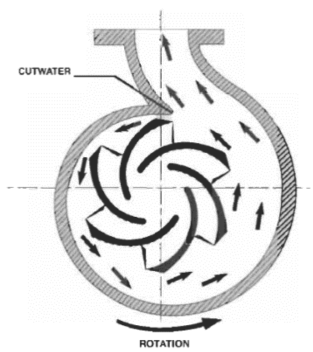


Figure 2.9   The cut-water in a volute [27]

Small centrifugal pumps usually have a single volute, volute diffuser and circular volutes. The use of the diffuser vanes makes a uniform distribution of velocity around the impeller resulting in lower radial impeller loads. The radial load is also a minimum in a circular volute at pump shut-off (or zero flow), and is maximum near the BEP.

Double volutes are commonly used in larger centrifugal pumps. This type of volute has two cutwaters that radially balance the two resulting and opposing hydraulic forces. The presence of two cutwaters significantly reduces the hydraulic radial load on the impeller.

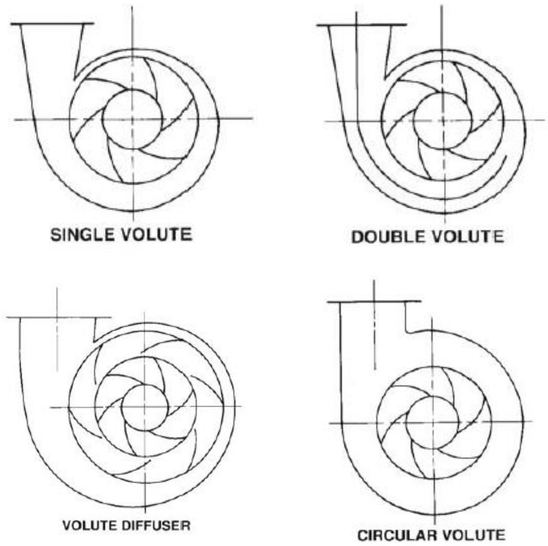


Figure 2.10 Common type of volutes in centrifugal pumps [28]

2.3 Centrifugal pumps performance characteristics

There are four basic quantities for measuring pump performance, i.e., *head*, *power*, *efficiency*, and *flow* [29]. The pump performance is normally described by a set of curves. Head, power, and flow are usually measured and efficiency is calculated by using the equation,

$$\eta = \frac{Q\rho g H}{P}\ , \tag{2.1}$$

where Q is the volumetric flow rate, ρ is density, g is standard gravity, H is total head, and P is power consumption.

Volumetric flow rate (Q) is defined as the volume of fluid which passes a particular cross-sectional area per unit time. It usually has units of either cubic metres per hour (m^3/hr) or litres per second (l/s). The flow rate is not constant during pump operation. It usually changes as the operation conditions are altered. It also depends on various factors such as fluid properties, pump size and its inlet and outlet condition, impeller size, pump speed, pump geometry, pump suction, discharge temperature and pressure conditions [13]. The volumetric flow rate Q can be calculated by,

$$Q = VA \, , \tag{2.2}$$

where A represents a cross-sectional, and V is the mean velocity of the fluid flowing in the pipe.

Head (H) is a measure of the total energy imparted to the fluid at a certain operating condition and capacity. The total head of a system in which a pump must operate consists of the static head, friction head, and velocity head [15].

Static head is the difference in elevation of the fluid surface. Thus, total static head of a system refers to a height measured from the suction fluid level to the discharge fluid level. The static discharge head is the height from the centreline of the pump to the discharge fluid level while the static suction head is measured from suction fluid level to the centreline of the pump. Figure 2.11 describes the relation of total static head, static discharge head, and static suction head.

Friction head, expressed in unit length, is the equivalent head which is used to overcome the friction losses due to the flow of the fluid through the piping system. Friction head varies with the diameter and the length of the pipe, the number and type of fittings, and the flow rate of the fluid.

Velocity head refers to the kinetic energy in the fluid at any point. Velocity head is expressed in joules per kilogram of liquid, that is, in metres of the fluid under consideration. The velocity head can be calculated by the following equation,

$$h_v = \frac{V^2}{2g}\,, \qquad 2.3$$

where h_v is the velocity head, V is the fluid velocity, and g is the standard gravity.

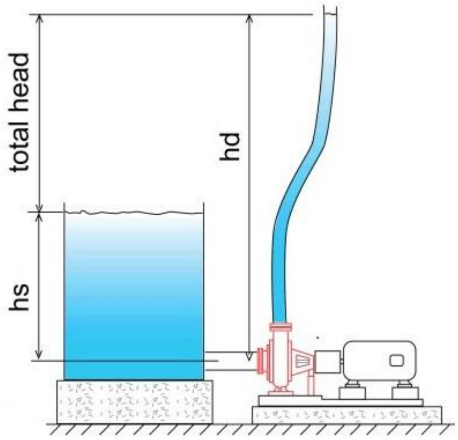


Figure 2.11 Total static head, static discharge head (hd), and static suction head (hs) [30]

The typical performance curves for centrifugal pumps are normally plotting head, power, and efficiency against flow as illustrated in Figure 2.12. All pump manufacturers usually provide performance curves together with pump power and operating efficiency.

The intersection point on the head-flow curve indicates the BEP—the *Best Efficiency Point*. The BEP graphically represents the point on a pump performance curve which yields the maximum efficiency of pump operation. Operating away from the BEP causes the pump's performance and operational life to significantly decrease, since the pump is subjected to an increase in wear.

Net positive suction head (NPSH) is the difference between the actual pressure of the fluid in a piping system and the fluid's vapour pressure at a given temperature. The NPSH value is an important parameter in piping system design and indicates that when the fluid pressure drops below the vapour pressure, the fluid starts boiling and cavitation potentially occurs.

NPSH may refer to one of two quantities for the cavitation analysis. The available NPSH (NPSHA) is a measure of the level of pressure at a given point. Meanwhile, the required NPSH (NPSHR) is the head at a specific point, i.e., the suction opening of a pump required to keep the fluid away from cavitation.

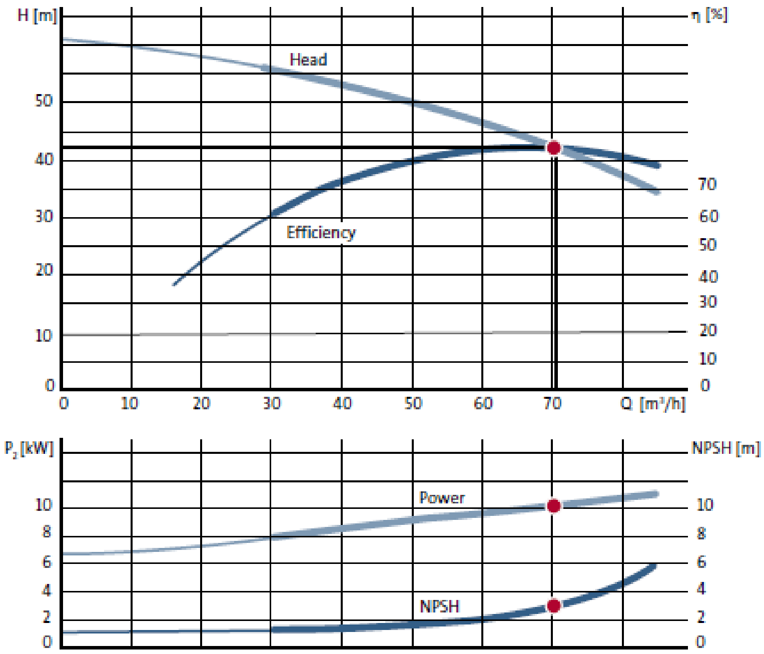


Figure 2.12 Performance curves for a centrifugal pump [31].

Pump manufacturers usually provide NPSH data on the pump inlet pressure at which serious cavitation is likely to occur. Cavitation is often observed at about a 3% drop in head from the BEP. However, this value varies with the physical properties of the fluid and the roughness of the surface of the piping and hydraulic equipment [32].

2.4 Common failure modes in centrifugal pumps

Centrifugal pumps which are a popular type of pump in most industries, can fail during their service as a result of problems that arise within the fluid such as cavitation and mechanical faults such as impeller faults and bearing faults.

In the following section, a brief overview of the failure modes which commonly occurs in the centrifugal pumps is discussed.

2.4.1 Cavitation

Cavitation is considered as the most common problem that occurs at the suction side of centrifugal pumps. Cavitation arises when the pressure of the fluid inside the pump drops below the fluid's vapour pressure. During cavitation, the vapour bubbles appear in the moving fluid where the pressure of the fluid is lower than its vapour pressure. The formation of vapour bubbles can have two harmful effects. Firstly, it can be sufficient to block the piping system, resulting in a significant reduction in the hydraulic performance. Secondly, the collapse of the vapour bubbles as they move to a higher pressure region may produce noise and erosion of the waterway surfaces.

The vapour bubbles normally contract to a particular size, then spring back, and contract again, repeating a series of growths and collapses until they eventually disappear, however in most cases, the vapour bubbles experience one cycle only [13]. Figure 2.13 shows the volute of pump with a transparent cover when the cavitation is non-existent and when cavitation is occurring.

The collapse of vapour bubbles is very fast resulting in very intense local pressure which leads to pitting and serious material erosion of the impeller blades, shrouds and volute; high level of noise and vibration; and reduced pump hydraulic performance [33]. It is clear that cavitation is an unacceptable condition which must be avoided.

The common way to solve the cavitation condition is by modifying the system design or operation to increase the NPSHA. If the NPSHA increases sufficiently above NPSHR then cavitation disappears. One method to increase the NPSHA is to increase the inlet pressure by raising the feed tank or the fluid level in the feed tank (open to atmosphere) or by increasing the pressure in the space above the fluid (open tank). Another way is by reducing NPSHR. This is conducted by lowering the temperature of the fluid so its vapour pressure decreases. Another alternative method to decrease NPSHR is by reducing friction head loss.

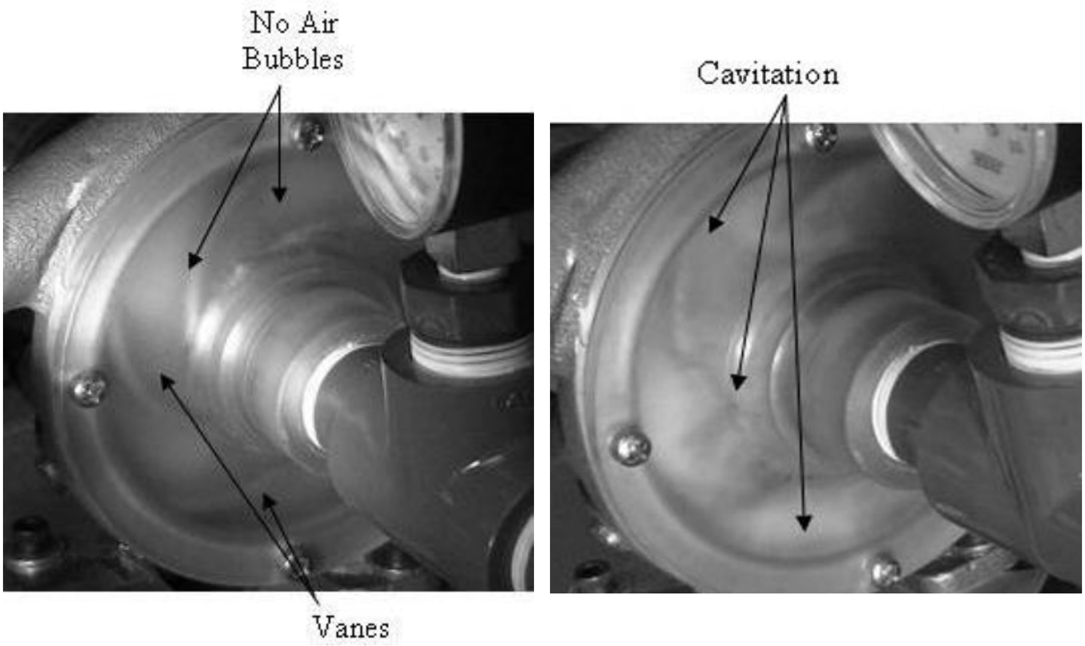


Figure 2.13 Cavitation in centrifugal pump: no cavitation (left), cavitation (right) [34]

2.4.1.1 Types of cavitation

Incipient cavitation describes the beginning stage of cavitation where it is only just detectable. There is a difference in the condition between the appearance and disappearance of cavitation. Generally, an increase in pressure above that at which cavitation appears is required to cause cavitation to disappear.

Traveling cavitation represents individual bubbles which appear in the fluid and move together with the fluid until they disappear. Usually, the bubbles appear at the low pressure point on solid boundaries or at the low pressure point in turbulent flow [35].

Fixed cavitation describes a cavity attached to the boundary of an immersed rigid body in the flowing fluid. It occurs when the flow detaches from the rigid body. This type of cavitation is defined as stable in a quasi-steady sense [36].

Vortex cavitation occurs when the cores of turbulence-generated vortices have the high shear area. This cavitation may exist as traveling or fixed [35].

Vibratory cavitation occurs at the very low flow velocity so that recirculation occurs in the pump. This type of cavitation is a special case since any elements of the fluid may undergo many cavitation cycles. Whereas, in all other types of cavitation, any fluid elements passes the cavitation area only once [37].

2.4.1.2 Cavitation damage

The collapse of vapour bubbles may produce very intense local pressure and shock waves which can cause severe damage to the metal surface due to erosion and destroy any surface film protecting the metal from corrosion. There are no materials which are completely resistant to damage due to cavitation. The signs of erosion appear as pitting, caused by the water-hammering action of the collapsing vapour bubbles. The damage of the metal surface occurs because when the cavities collapse, the jet of fluid that is released hits the surface of the metal at the local speed of sound, which creates a local high surface stress that can be higher than the ultimate strength of the metal [38]. The damage of the metal due to cavitation may be reduced by redesigning appropriate details of the pump such as smoothing the surface of the metal, coating the metal, using corrosion-resistant materials, minimizing pressure differences in the cycle, and using cathode protection [39]. Generally, the more brittle the material, the more severe they are damaged by cavitation, because brittle materials are more vulnerable to fatigue.

Pitting of the impeller surface may cause a serious deterioration of hydraulic performance which leads to a high level of damaging structural vibration together with a high level of noise. The noise generated by collapsing cavities is a sharp crackling sound. The level of noise resulting from cavitation may indicate the severity of the cavitation. The noise can be observed in and around the pump suction. If the crackling noise seems to be random with high intensity knocks, then it indicates cavitation in the suction recirculation.

Pump vibrations due to cavitation can cause reduction in pumping efficiency. The vapour bubbles created during cavitation in the suction area around the impeller impede the fluid, thus resulting in a reduction in output. An efficiency drop is considered a more accurate sign of cavitation occurrence, since noise is not clearly observable until cavitation has progressed to the stage where the efficiency of the pump becomes very low [16].

Generally, pump vibration due to cavitation occurs within the high frequency spectrum region, which is often overlapped with the blade pass frequency (BPF) harmonics [40, 41].

2.4.2 Faulty Impeller

The impeller is one of the most critical rotating elements in a centrifugal pump which converts electrical energy from the motor into kinetic energy of the impeller to accelerate the fluid outwards from the centre of rotation. The impeller is usually made of iron, steel, aluminium, brass, or bronze.

During its service, the impeller may become worn out which alters the operating condition of the pump such as reducing the hydraulic performance, reducing the pumping pressure, and the flow rate. The main cause of the impeller wear is cavitation. The high pressure resulting from imploding vapour bubbles makes the impeller erode. Figure 2.14 shows an example of a worn impeller caused by cavitation. The worn impeller may also produce imbalance of the impeller resulting in increased vibration to the system.

Figure 2.14 Worn impeller due to cavitation [42]

The source of vibration in the pump may come from mechanical and hydrodynamic motion. The unbalanced masses and friction in the bearing is the most frequent cause of mechanical vibration, while fluid flow disturbance is the common cause of hydrodynamic vibration.

A worn impeller may cause a loss in discharge pressure resulting in reduced pump efficiency which in turn may increase the power consumption as wear occurs. The decrease in discharge pressure may result in increasing vibration levels as choking the flow of the fluid induces cavitation. If vibrations occur at shut-off, a physical imbalance of the impeller is most likely exist [43].

An unbalanced impeller usually appears in the vibration spectrum as a pump shaft speed vibration. Generally, the unbalanced impeller is not inspected until heavy pitting is found on the impeller. The severity level of pitting is usually used as the indicator of the need for balancing an impeller [16].

2.4.3 Faulty Bearing

In general, bearings fail due to contamination of the bearing lubrication by water, foreign particles, other liquid or because of overheating caused by an overload on the bearing [44].

If the pump is running under normal conditions i.e., BEP, the bearing loads are caused by the weight of rotating elements only. However, due to other operating conditions, BEP is not always achieved which eventually leads to overloading conditions on the bearings. Some other conditions which result in bearing overloading are imbalanced shaft, cavitation, bent shaft, blocked impeller, and extreme radial and axial thrust [15, 25].

Excessive loads can produce a large amount of heat which can cause damage on bearings. The appearance of damage caused by an excessive load is similar in appearance to inadequate lubrication damage. Figure 2.15 shows typical inner race damage due to overloading a ball bearing.

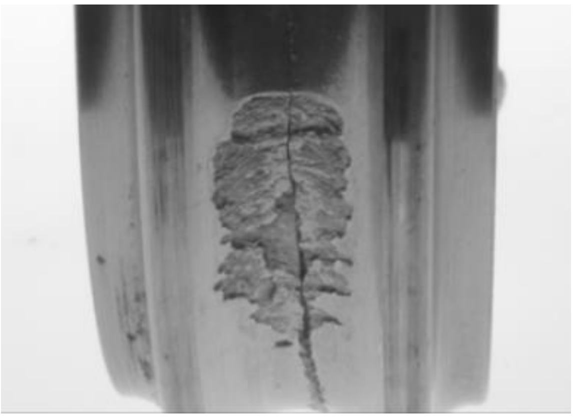


Figure 2.15 Inner race damage of ball bearing [45]

Overloading of bearings within a short time period may cause deterioration of bearing performance leading to an increase in vibration level. The vibration signal from a bearing contains spectral components that are associated with the geometry of the bearing, the rotation speed, the number of rolling elements, and the location of the defect.

When a bearing rotates, any defects in the rolling elements may produce several vibration frequencies which correspond to fundamental defect frequencies (or characteristic frequencies) such as ball spin frequency (BSF), fundamental train frequency (FTF), ball pass frequency of the inner race (BPFI), and ball pass frequency of the outer race (BPFO). These frequency components are small in amplitude so that they are often hidden among high level vibration components.

The equation to calculate the fundamental defect frequencies are given as,

$$FTF = \frac{S}{2}\left(1 - \frac{d}{D}cos\phi\right) , \tag{2.4}$$

$$BSF = \frac{D}{2d}\left(1 - \left(\frac{d}{D}cos\phi\right)^2\right), \tag{2.5}$$

$$BPFO = \frac{nS}{2}\left(1 - \frac{d}{D}cos\phi\right), \tag{2.6}$$

and

$$BPFI = \frac{nS}{2}\left(1 + \frac{d}{D}cos\phi\right), \qquad 2.7$$

where S is the shaft speed, n is the number of rolling elements, d is ball diameter, D is pitch diameter, and ϕ is the contact angle.

Deterioration of a bearing usually occurs in a gradual stage from light to severe damage. Table 2.1 as taken from Beebe in Ref. [29] depicts degradation stages of rolling element bearings.

Table 2.1 Stages of degradation of rolling element bearings [29]

Stage of bearing wear	Noise Level	Vibration level
Stage 1	Normal	Normal
Stage 2 (less than 20% bearing life left)	Slight change	Slight increase in acceleration. Resonance of bearing components show. At end of this stage, sidebands appears on these resonances
Stage 3 (less than 5% bearing life left)	Audible to a trained ear, but repeatability is poor	Large increase in acceleration and velocity. Bearing defect frequencies and harmonics are now detectable and growing. Sidebands are evident on these and also on component resonances
Stage 4 (about 1% of bearing life left)	Change in pitch clearly audible	Significant increase in displacement and velocity, 1/rev and harmonics show.

2.4.4 Blockage

Blockages in pumps refer to the condition where the pumped fluid contains materials such as rags, fibres, pieces of wood, abrasive grit, sand or bricks that can wrap around the impeller to prevent or even stop impeller rotation or can choke the area between impeller blades to reduce the flow rate. The accumulation of rags in a screw centrifugal impeller is depicted in Figure 2.16.

Problems with the blockages commonly occur in pumps used in the sewage industry. Pump blockages may result in decreasing hydraulic efficiency, increasing power consumption, and also consume unnecessary operational time and resources.

Blockages in pumps may generate vibrations that are either caused by residual mechanical unbalance due to the accumulating foreign materials in the rotating parts or caused by change of fluid flow behaviour. The magnitude of the vibration of blocked pumps normally increases at the frequency of the shaft rotational speed times the number of impeller blades, (blade pass frequency).

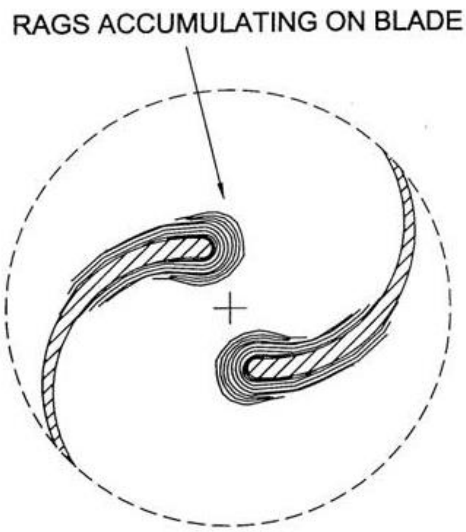


Figure 2.16 Build-up of rags in a screw centrifugal pump [46]

In the next chapter a review of current vibration signal analysis methods in fault diagnosis is introduced. It also presents the application of time-domain analysis, frequency-domain analysis, and time-frequency analysis in fault diagnosis of rotating machinery.

CHAPTER THREE

3 An Overview of Vibration Signal Analysis Methods

Condition-based maintenance (CBM) is considered as the most modern and popular maintenance technique discussed in the literature and widely implemented in industries [1, 5]. In order to recommend maintenance actions, CBM collects information of the health condition of equipment through condition monitoring (CM) processes, so that there is a close relationship between the performance of CBM and the performance of the associated condition monitoring. In turn, the performance of condition monitoring depends on the quality of the fault diagnosis. This indicates that fault diagnosis plays critical roles within CBM processes during the assessment of the machines condition. Fault diagnosis is related to the determination of the fault location, type, cause and severity level of component faults. Hence, the skill required for fault diagnosis for maintenance personnel is substantial to correctly utilise the condition monitoring method.

Many condition maintenance techniques have been developed and discussed in the literature [47]. Various indicators of faulty components have been implemented either on a laboratory scale or within industries such as vibration, noise, electric current, tribology, pressure and temperature. According to Deng and Zhao [48], vibration-based methods are the most common techniques used in condition monitoring.

Vibration monitoring relates to the vibration measurement of components to detect changes of a vibration signature which could indicate component failure. Vibration measurement is commonly used due to its effectiveness and versatility. Feng et al., [49] argued that the level of vibration severity of a component and its pattern have a strong correlation with the health condition of the machine. Many researchers have investigated the correlation between a vibration signal and machine health condition and have proposed various analysis for particular vibration signals.

In general, vibration signals are collected *in situ* (or in place) using a transducer like an accelerometer, mounted to the component. Physical information (e.g., mechanical vibration) is collected and stored at certain time intervals which form time series or time waveforms. The data is then analysed further for the purpose of diagnosis and prognosis in condition monitoring.

Signal processing involves data processing of the data series and many signal processing methods are available in the literature to analyse and interpret it. Generally, signal processing may be divided into three groups, i.e., *time-domain*, *frequency-domain* and *time-frequency domain*. Figure 3.1 shows a schematic diagram of signal processing techniques commonly used in CM.

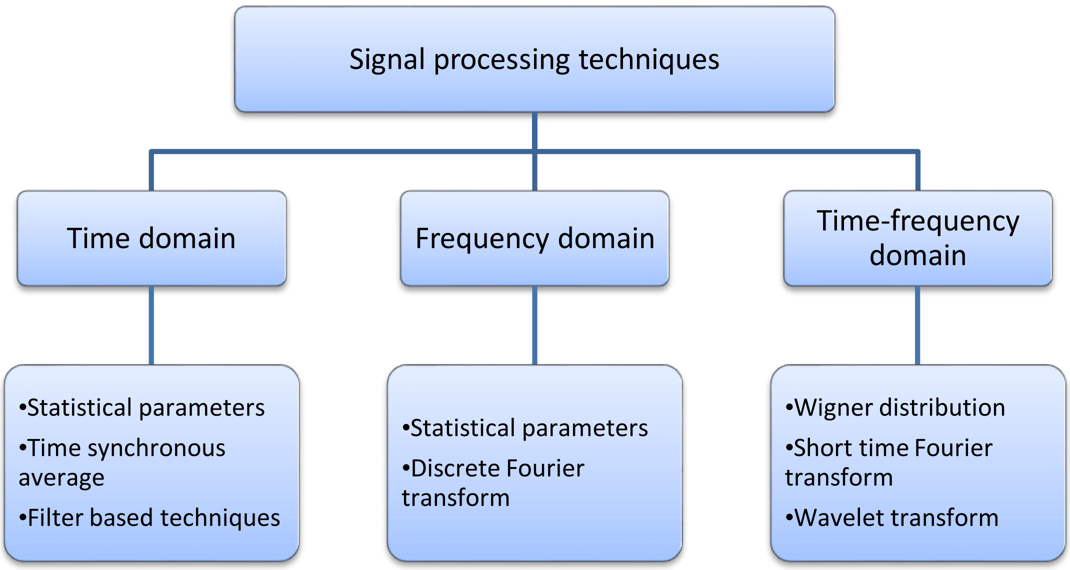


Figure 3.1 Signal processing techniques commonly used in CM [50]

The main objective of signal processing in fault diagnosis is to extract the most relevant features which may be hidden in the original time waveform. Feature extraction techniques consist of either domain transformation approaches or dimensional reduction approaches. In the case of a domain transformation approach, signal processing converts the original time waveform into another domain in order to reveal the hidden information, whilst the dimensional reduction approach reduces the amount of dimensionality but still retains the most important information. In addition, with the reduced amount of dimensionality of data, the computation cost can be significantly reduced.

In the following sections, some important aspects of the three categories of signal processing are discussed.

3.1 Time Domain Analysis

In general, vibration data are obtained in the time domain as data series indicating displacement, velocity, or acceleration. Typically, the vibration signals are acquired using a particular transducer, such as an accelerometer. Time-domain vibration features can be extracted using descriptive statistics like standard deviation, kurtosis, root mean square (RMS), variance, skewness, and others.

3.1.1 Standard Deviation

Standard deviation (σ) indicates how much dispersion the set of data (samples) has from its mean. A low standard deviation represents that the samples are near to the mean, while a high standard deviation denotes that the samples are scattered from the mean. Standard deviation can also be considered as a measure of the power content of the signal [51]. Standard deviation is defined as,

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \tag{3.1}$$

Supposing the signal $x = x_1, x_2, \ldots, x_N$, then x_i is an element of x, $\bar{x}$ represents the mean of the x and N is the number of elements. Ahmed et al., [52] used this feature as one of the time waveform vibration features for fault classification in compressors using a genetic algorithm and a probabilistic neural network.

3.1.2 Kurtosis

Kurtosis shows the shape of data/signals whether they are flat or spiky. A normal component (no fault) often gives a very low kurtosis, while a faulty component has a high kurtosis which is caused by the spikiness of the signal. It is given mathematically as,

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\sigma^4}\ , \qquad 3.2$$

where x_i is an element of x, N is the number of elements, $\bar{x}$ represents the mean of x and σ denotes the standard deviation. The use of kurtosis as a feature in fault diagnosis can be found in many literatures. Some of them can be found in Lin and Zuo [53], Antoni and Randall [54] and Sawalhi et al., [55].

3.1.3 RMS

The root mean square (RMS) indicates the energy level or power level of a signal. It is defined as,

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\ , \qquad 3.3$$

where N is the number of elements, $\bar{x}$ represent the mean of the elements and x_i is the element of x. An example of implementation of RMS as a feature in fault detection of centrifugal pump can be found in Sakthivel et.al [51].

3.1.4 Variance

Variance is also known as the second moment statistical measure and the formula is given as,

$$Variance = \frac{\sum(x_i - \bar{x})^2}{N-1}\ , \qquad 3.4$$

where $\bar{x}$ represents the mean of the signal, x_i denotes the element of x and N represents the number of elements. A few examples which show the implementation of variance for fault detection can be found in Samanta and Al-Balushi [56] and Rafiee et al., [57]. It should be noted that it is given by the square of the RMS.

3.1.5 Skewness

Skewness indicates the degree of asymmetry of a distribution around its mean and it is defined as:

$$Skewness = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(N-1)\sigma^3}, \qquad 3.5$$

where $\bar{x}$ represents the mean of the elements, x_i represents the element of x, σ denotes the standard deviation and N represents the number of elements. An example of the use of skewness in fault detection can be found in Ahmed et al. [52].

A thorough discussion about implementation of the above mentioned statistical features in pump fault diagnosis is presented in section 3.6.

Other popular methods of time domain analysis found in literatures include Time-Synchronous Averaging (TSA), Autoregressive Moving Average (ARMA), Filter Based Method, and Stochastic Method and Blind Source Separation which discuss briefly in the following section.

3.2 Time-Synchronous Averaging (TSA)

Time-synchronous averaging (TSA) is considered as one of the most powerful algorithms for vibration analysis especially in gear fault detection [58]. TSA is able to effectively separate the vibration signature of a particular gear from other gears' vibration and noise in a gearbox which are not synchronous with the gear under analysis. TSA is a technique that refines periodic time waveforms from the sources corrupted by noise. This is achieved by averaging the vibration signals over a number of shaft revolutions, so as to improve the desired vibration signal components [2].

TSA is implemented by taking the average of the series of signal segments each of which corresponds to one period of a synchronised signal, triggered by a once-per-revolution trigger of a known phase (key-phasor) [59]. The following TSA's formula is suggested by [2, 60],

$$s(t) = \frac{1}{N}\sum_{n=0}^{N-1} x(t+nT), \qquad 0 \le t \le T \ , \qquad 3.6$$

where $x(t)$ represents the signal, T represents period of averaging and N represents the number of samples. Details on discussion of TSA can be found in literatures of [58, 60, 61].

3.3 Autoregressive Moving Average (ARMA)

Autoregressive moving average (ARMA) is another advanced method in time-domain analysis. The method converts time-domain into time series models. In principle, ARMA fits the time-domain to a parametric time series model and extracts features from the proposed parametric model. An example of an ARMA model is suggested by Jardine et.al. [2] as,

$$x_t = a_1 x_{t-1} + \cdots + a_p x_{t-p} + \varepsilon_t - b_1 \varepsilon_{t-1} - \cdots - b_q \varepsilon_{t-q} \ , \qquad 3.7$$

where p and q is the order of ARMA model, x denotes the time-domain signal, ε's represent normal distribution with a zero mean and variance σ^2, and a_i and b_i represent coefficients.

An example of implementation of ARMA can be found in Baillie and Mathew [62] which explored the performance of AR models by comparing three AR modelling approaches. Other examples are from Garga et al., [63] which proposed an AR model used for dimension reduction and from Zhan et al., [64] which combined the use of state space model and an AR model for vibration signal analysis.

There are many other time-domain techniques available for vibration signal analysis for fault diagnosis. Some of them are: filter-based method, prony model technique, adaptive noise cancellation (ANC), stochastic methods, and blind source separation methods. The following sections briefly discuss these techniques.

3.4 Filter-Based Method

Filter-based methods, as suggested by its name, use filters to remove noise and isolate signals to extract features from the time waveform. This technique includes demodulation, prony model, and adaptive noise cancellation (ANC).

Modulation is the process of varying properties of a signal (e.g. amplitude or phase). If the varied property is amplitude then it becomes amplitude modulation, whilst if the frequency is varied then it becomes frequency or phase modulation. Demodulation is the reverse of the modulation process. Demodulation also consists of amplitude demodulation and phase demodulation. The amplitude demodulation is widely used as vibration signal analysis particularly for bearings and gears. It is also known as resonance demodulation, envelope analysis, or high frequency resonance [65]. This technique separates low frequency signals from high frequencies noise which makes them easy to be analysed. With this technique, the signal envelope can be extracted by using amplitude demodulation and then its frequency is analysed to reveal the repetitive frequencies that are related to the faults [66].

Pan et al., [67] demonstrated the use of envelope analysis for multi-fault detection of ball bearings. Other investigation on fault detection in bearings can be found in [68-70].

Prony-model technique is commonly used in fault detection of low speed bearing elements. The technique analyses transient vibration signals and estimates the frequency, magnitude, damping, and phase directly from the modal components. [71]. The results of a Prony-model include spectral plots, trending parameters, and Prony parameters. The method has been shown to be able to analyse the transient signal and to determine fault severity [72].

Adaptive noise cancellation (ANC) is a technique to analyse time waveforms by filtering signals corrupted by additive noise. The technique uses two input signals as main input and a reference input. The main input contains the corrupted signals, whilst the reference input contains noise associated with the main input. These two input signals usually are acquired simultaneously. In practice, Shao and Nezu [73] proposed to mount the sensor for the reference signal acquisition at a position in the noise area where the signal is weak.

3.5 Stochastic Method and Blind Source Separation

Stochastic parameter techniques such as chaos and blind deconvolution are considered as advanced methods to analyse vibration in the time domain waveform. Chaos and the correlation dimension have been implemented to identify faults in a rolling element bearing with various severity [74]. Mevel et al., [75] argued that the correlation dimension can provide true information associated with the dynamical system, thus it can be used to determine different faults.

Another advanced analysis method in the time domain is Blind Source Separation (BSS). This technique attempts to restore the source signals from a set of mixed observed signals without knowing information about mixing process or source signals. [76]. The different combination mixture of observed signals is generally taken from a set of sensors. BSS assumes the mutual independence of the sources, and then the source signals and information about mixture are not necessarily needed.

BSS has been shown to be a very attractive signal processing method for fault diagnosis of rotating machinery [77]. Gelle et al., [76] implemented BSS to rotating machinery by acoustical and vibration parameters. The results showed that BSS can be used for fault detection of rotating machinery by using the Nguyen-Jutten algorithm within a temporal context. It also concluded that vibration based parameters had better performance to monitor a whole system than the acoustical parameters. The BSS method generally assumes that the signal is either free from noise or has spatially distinct white noise. However, the real signals from rotating machinery may contain spatially correlated noises. Serviere and Fabry [78] proposed a technique called 'robust-to-noise' to separate signals with spatially correlated noise from rotating machinery. The results indicated that the technique is efficient for analysis of a signal with low signal-to-noise ratio.

3.6 Implementation of Time-domain Analysis of Vibration Signals for Fault Diagnosis of Machinery

Generally, for the purpose of fault diagnosis, time-domain analysis is applied by measuring the level of statistical features such as standard deviation, kurtosis, RMS, variance, skewness, etc. There are many statistical parameters suggested in the literature that have been successfully applied for fault diagnosis of rotating machinery like RMS, peak factor, histogram lower bound (HLB), histogram upper bound (HUB), entropy, variance, skewness, kurtosis [79], crest factor, absolute value, shape factor, clearance factor, normal negative log-likelihood value (Nnl) and Weibull negative log-likelihood value (Wnl) [52], range, minimum value, maximum value and sum [51].

Statistical parameters may indicate the distribution of amplitude of the vibration signal acquired from rotating machinery. In addition, the advantage of using statistical parameters for fault diagnosis is that the variation and speed condition of a machine has a lower effect on their values [80].

McCormick and Nandi [81] investigated the use of artificial neural networks (ANN) for condition monitoring of the rotating machinery from the vibration time series. The method proposed automatic classification of the machine condition using features extracted from several methods as neural network inputs. The extraction methods were based on the zero lag higher-order statistic which were applied either to horizontal and vertical vibration time series. The backpropagation with adaptive learning and momentum technique was used to train the ANN. The backpropagation method was tested in a test rig which was set up to introduce unbalanced shaft and rub faults. The results showed that the method using the combination of moments of the complex time series with the moments of its derivative achieved more than 90% fault detection success rate.

Huaqing and Peng [82] proposed a time-domain based method, called “partially-linearized neural network (PNN)”, for condition diagnosis of a centrifugal pump. This technique developed a sequential diagnosis method using PNN which can distinguish types of failure at an early stage for rotating machinery effectively. A new indicator, called non-dimensional symptom parameters (NSP) was defined to reflect the time-domain features for the purpose of fault diagnosis. In order to measure the sensitivity of NSP for detecting faults, the synthetic detection index (SDI) was also developed. The method was tested for identifying fault types of a centrifugal pump, such as impeller fault, cavitation, and unbalance. The result showed that the method could identify the type of faults effectively.

Lei and Zuo [83] performed an investigation on the usage of feature parameters in the time-domain and frequency-domain for gear damage detection. They proposed 25 feature parameters which were extracted to characterize the gear conditions. The features consisted of 10 time-domain features, 11 statistical features specially formulated for gear diagnosis and 4 statistical features extracted from the frequency-domain. Before the features was inputted into a classifier, the selection of relevant features was carried out through a two-stage feature selection and weighting technique (TFSWT) based on the Euclidean distance evaluation technique (EDET). The weighted k-Nearest Neighbors (W-kNN) was proposed to identify the gear crack levels. The proposed method was tested using the vibration signals from the gears with varying speeds and loads. The result showed that the proposed method could identify the gear crack levels with a very good accuracy.

Sakthivel et al., [51] investigated the use of a set of statistical features (e.g. standard deviation , standard error, kurtosis, variance, range, skewness, sum minimum value and maximum value) for fault diagnosis of centrifugal pumps. This method applied a decision tree algorithm by extracting statistical features from vibration signals that were normal and contained faulty conditions. The faulty conditions included bearing fault, seal fault, impeller fault, seal and impeller faults, and cavitation. The result concluded that the approach was a promising method for practical application of fault diagnosis of a monoblock centrifugal pump.

In addition, Sakthivel et al., [84] compared the use of rough set-fuzzy and decision tree-fuzzy methods for fault classification. These two methods extracted statistical features from normal and faulty condition of vibration signals of a centrifugal pump. The study concluded that the decision tree-fuzzy method has better accuracy than the rough set-fuzzy system.

Al Thobiani et al., [85] investigated a cavitation measurement method in centrifugal pumps using vibro-acoustic techniques. They evaluated vibro-acoustic techniques with conventional statistical parameters like peak factors and kurtosis from both time domain and frequency domain. The result showed that peak factor and kurtosis were inefficient for indicating cavitation. However, they found spectral entropy to be more accurate for detecting cavitation. It was also revealed that spectral entropy from airborne acoustic signals gave a better diagnostic performance than the surface vibration.

Al-Hashmi [86] proposed the use of probability density function (PDF) and standard deviation (SD) for cavitation detection of centrifugal pumps. PDF and SD were extracted from acoustical time domain signals which were taken from the discharge port. These two statistical parameters were then analysed and compared with the flow rate. It was concluded that both PDF and SD together with pump flow-rate can be used to identify the occurrence of cavitation from a centrifugal pump.

Nasiri et al., [87] used two statistical features (kurtosis and crest factor) and neural networks to detect cavitation in centrifugal pumps. The purpose of the method was to apply two statistical features as the input of a specially built neural network for cavitation detection. Three conditions were introduced to test the method i.e., normal, developed and fully developed cavitation. The feed forward back propagation was then used to train the network. The result showed that the method was able to identify cavitation in centrifugal pumps.

Ahmed et al., [52] explored the use of a combination of principal component analysis (PCA) with statistical parameters for different fault diagnosis from a reciprocating compressor. PCA was applied to select the effective diagnosis features from 14 statistical parameters. Fault detection based on PCA was subsequently developed by using Hotelling's T^2 and Q statistic to detect various faults. The result showed that the method could detect single and multi-faults introduced in a reciprocating compressor. The existence of faults was detected by comparing T^2 and Q statistics values of the features with a defined threshold line.

Sun et al., [88] used statistical features like kurtosis, crest factor, and RMS as a generic method for analysing faults from rolling element bearings. The features were extracted from the time domain vibration signals of normal and faulty bearings (inner race fault, outer race fault, and roller fault). These features were used for the purpose of fault pattern recognition by mapping them to create feature integration, linear classification and diagnosis. In order to generate a classifier for bearing fault diagnosis, an artificial neural network (ANN) was used and combined with a mapping strategy. The study concluded that the technique had advantages for monitoring complicated vibration signals which commonly occur in rolling element bearings.

The use of new statistical parameters for the early detection of rolling element bearing fault was proposed by Niu et al., [89]. The modified statistical parameters, called unified description of normalized statistical parameters, contained skewness and kurtosis for rolling element bearing fault detection. The results from both experimental test and computer simulation showed that the new statistical parameters had the same performance with the original parameters for early bearing fault detection.

The combination of statistical parameters extracted from the time-domain, frequency-domain, and band frequency was developed by Hope and Wang [90] for detecting various types of bearing faults in centrifugal pumps. A statistical parameter of RMS was calculated from the time-domain, frequency-domain, and band frequency for each of the six types of bearing defects. Artificial neural networks (ANNs) then were applied in the classification of six different bearing defects. It was found that the method could classify all of the faults successfully using 2 hidden layers in cascade-forward back-propagation neural networks. It was also shown that the classification accuracy was 100%.

The use of statistical parameters like standard deviation as a feature for detecting bearing faults was investigated by Rafiee et al., [91]. Standard deviation was extracted using the wavelet transform approach. The standard deviation values together with energy levels obtained from the wavelet transform were then used to train the neural network. It was found that there was a significant improvement in training convergence and network performance, in terms of accuracy, in detecting gear and bearing faults.

3.7 Frequency Domain and Time-Frequency Domain Analysis

The features extracted from frequency domain and time-frequency domain generally can indicate machinery faults better than the use of purely time domain vibration features. This is due to the characteristics of frequency components like resonance frequency components which are easily observed and associated with the faults. A summary of developed frequency-domain analysis techniques is given in Table 3.1 as taken from Yang et al., in Ref. [92]

Table 3.1 Overview of frequency-domain techniques [92].

First order	Second order	Third order	Fourth order
Spectrum (FFT)	Power spectrum, Power cepstrum (logarithm of Power Spectrum), Cyclostationarity	Biocoherence spectrum, Bi-linearity	
Correlation of spectrum, Signal averaging, Short Time Fourier Transform (STFT)	Spectrogram, Wigner distribution	Wigner bi-spectra	Wigner tri-spectra

The FFT [93] is considered to be the most popular signal analysis technique and has been commonly applied to identify the desired frequency components. The FFT may be used either for raw vibration signals or processed vibration signals. Envelope analysis is an instance by which the processed vibration signal is used before applying the FFT.

In principle, the power spectrum is the square of the magnitude of the amplitude spectrum and can be used to diagnose faulty components [94]. Bispectrum is the higher order spectrum and can also be used to identify failure in bearings [95]. The third-order spectrum, called bicoherence spectrum, measures phase coherence among spectral components and has been applied for monitoring of bearing condition [96]. The power cepstrum is defined as a logarithm of the power spectrum and can be used to diagnose faulty components in machinery [97].

Another second order spectrum method is cyclostationarity. This is a second order frequency domain statistical analysis method. This technique attempts to extract the spectral correlation function from cyclostationarity which is a powerful feature for fault analysis in gears [98] and bearings [99].

Short time Fourier transform (STFT) is a time-frequency analysis technique that can represent signals in both time and frequency domain [100]. Due to this characteristic, STFT is suitable for analysing vibrations signals which are non-stationary.

Examples of the well-known quadratic time-frequency techniques are the Wigner distribution and the spectrogram which are commonly applied in gear faults. By its nature, the vibration signals from gear systems cause interfering cross-terms, which makes the analysis of the energy distribution not straightforward. The third-order and fourth-order Wigner spectra, called Wigner bi- and tri-spectra respectively have been studied and developed for analysing signals of rotating machinery [101].

The continuous wavelet transform (CWT) is an improvement of STFT in order to obtain better time-frequency resolution. The scalogram is the squared modulus of the CWT. Both of these techniques have been applied for faults diagnosis. The discrete wavelet transform (DWT), discrete wavelet packet analysis (DWPA), and time-frequency-scale domain (TFS) are other variant of the wavelet transform which have also been applied for faults diagnosis of machinery [92].

3.7.1 Fast Fourier Transform (FFT)

The FFT transforms time-domain into frequency-domain which produces a complex spectrum of the sampled signal. It calculates spectra power level and phases of the signal from frequency range of zero to half of the sampling frequency.

One of the advantages of the FFT over other frequency-domain techniques is that it retains phase information of the signal which makes the inverse transformation possible and relatively simple. Another advantage is that it can evaluate multi-channel measurements and system analysis such as frequency response function, coherence, and correlation [102].

The FFT is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) and its inverse. The algorithm calculates a series of N data points in approximately $N \log_2 N$ operations instead of using N^2 operation [103]. This makes a significant improvement of the calculation speed. As a result, this method is very popular and widely applied in engineering and science in the frequency-domain area [97].

The Fourier transform and its inverse for continuous signals is defined mathematically as,

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \,, \tag{3.8}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} dt \,, \tag{3.9}$$

where $x(t)$ is the time-domain function and $X(f)$ is the frequency-domain function. The analogous DFT and its inverse are given by [103]:

$$X_d(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi k n}{N}} \,, \tag{3.10}$$

$$x(n) = \sum_{k=0}^{N-1} X_d(k) e^{j\frac{2\pi k n}{N}} \,, \tag{3.11}$$

where both $X_d(k)$ and $x(n)$ are, in general, complex series.

An important property of the DFT is the convolution relationship. That is, the periodic mean convolution of the two time series of the DFTs is the DFT of the product of two DFTs. This property is powerful and useful for computing the filtered output of an input waveform.

As abovementioned, FFT is an algorithm which makes the calculation of the DFT of a time series significantly faster than other algorithms. Table 3.2 as taken from Cochran et al., in Ref. [104] shows a comparison of computational cost which can be achieved by the using the FFT.

Table 3.2 Comparison of computational cost between “direct” and FFT [104]

Operation	Approximate number of multiplications	
	Direct	FFT
DFT	N^2	$2N\log_2 N$
Filtering (Convolution)	N^2	$3N\log_2 N$
Autocorrelation Function	$\frac{N}{4}\left(\frac{N}{2}+3\right)$	$3N\log_2 N$
Two-Dimensional Fourier Transform (Pattern Analysis)	N^4	$4N^2\log_2 N$
Two-Dimensional Filtering	N^4	$3N^2\log_2 N$

Comparison of the number of multiplication required for calculating the DFT using the FFT algorithm and the number of calculation required using direct calculation of the DFT is depicted in Figure 3.2.

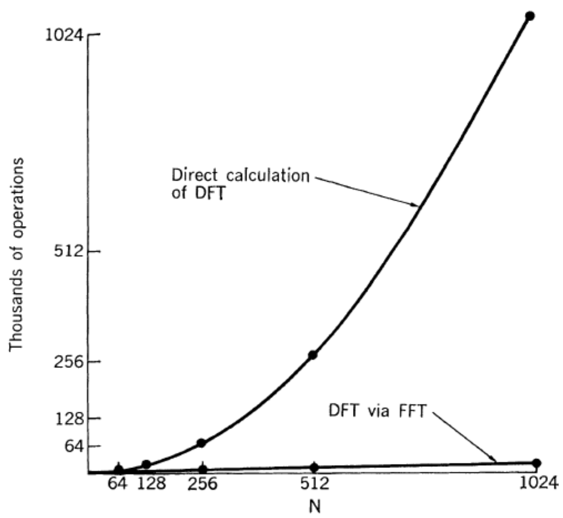


Figure 3.2 Number of calculation required for computing DFT using the FFT vs direct calculation [105]

The FFT is a very efficient calculation technique which combines larger weighted sums of samples sequentially to produce the DFT coefficient [104]. More detailed explanation about the FFT algorithm can be found in [104-106].

Despite the advantages, there are three problems most often encountered in using the FFT: leakage, aliasing, and the picket-fence effect.

Leakage is a characteristic attribute of Fourier transform. Leakage usually refers to the *windowing* effect, which is a product of $x(t)$ with the window function. Since the basic assumption in Fourier transform requires the signal to be periodic so that the sampled signal in the sample window must be also periodic, otherwise leakage will occur. The effect of leakage in the spectrum is that the energy from the true frequency spreads into adjacent frequencies. Leakage also reduces the magnitude of the signal amplitude so that it will be less than the original amplitude of the signal. The usual approach to reduce the leakage is by applying special sample windows to the time series, which has a special characteristic of having lower side lobes than the rectangular sample window.

Aliasing is a condition where the high-frequency components can impersonate the low frequencies. It also refers to the effect that causes different signals to become indistinguishable of one another when sampled. The aliasing effect occurs when the sampling rate is too low. The aliasing effect can be removed by using a sampling frequency at least twice the maximum component frequency of the function being sampled.

The picket-fence effect, also called resolution bias error, means there may be peaks in the original spectrum that are between the lines of the FFT analysis. That is, the peaks in an FFT spectrum will be measured too low in level, and the valley will be measured too high. The cure to this problem involves applying data windows to the time series and performing complex interpolation on the complex Fourier coefficients.

3.7.2 Envelope Analysis

Envelope analysis, also known as high frequency resonance technique (HFRT), is a signal processing technique which generally refers to the sequence of processes whereby a raw vibration signal from an accelerometer is band-pass filtered, then band-pass enveloped (or rectified), and finally transformed using the FFT, to create the calculated enveloped signal [107], as depicted in Figure 3.3

Figure 3.3 Procedure for the envelope analysis technique [107]

The procedure begins with filtering time-domain vibration signal using a band pass filter which is chosen to include high frequency range. The band pass filter is used to eliminate the low frequency range which is generally found in rotating machinery such as unbalance and misalignment. The filtered signal is then rectified by taking the time waveform and folding the bottom part of the signal onto the top half. The rectifying process is usually performed by using the Hilbert Transform. After the signal is being rectified, the next process is applying FFT to the enveloped signal to reveal the envelope spectrum. The result is an envelope spectrum showing obvious amplitude peaks that are not visible in the FFT.

Envelope analysis has been a proven useful method in cavitation detection of a centrifugal pump [107]. At an operating condition without cavitation, the peak at the 1x rpm was observed in the envelope spectrum. In this operating condition, the peaks are relatively low and the harmonics are not clearly visible. Meanwhile, under the cavitation condition, the strong peak was observed at ½ blade pass frequency (BPF). The peaks in the envelope spectrum under the cavitation condition are consistent at different measurements location such as the pump’s inlet, volute, outlet, and bearing housing.

The strong peak at ½ BPF in the envelope spectrum gives some evidence of the cavitation condition. The reason why a strong peak occurs at the sub-harmonic ½ BPF during cavitation condition is due to the early collapse of the bubbles before they reach one single cycle. The sub-harmonic ½ BPF may be observed in the envelope spectrum as short bursts spaced by long intervals [107].

Envelope analysis is a widely used technique applied to fault diagnosis of rolling element bearing [108]. This technique may extract periodic impact produced by a faulty rolling element bearing even though the energy signal generated by the rolling element bearing is very low. This makes it possible to identify a fault bearing at the early stage of failure.

A faulty rolling element bearing generates impact or impulse in very short duration. The impact occurs when the rolling element passes over the fault zone. This low energy impact is then distributed over a broad range of high frequency. Hence, the identification of bearing faults is difficult due to the presence of relatively high level energy of vibration from other machine components. However, the natural frequency of the bearing element is excited by the impact which generates much higher frequency components than that produced from other machine elements. The envelope spectrum can show the repetitive impact of rolling element bearings as a strong peak with several harmonics at the frequency component representing faulty bearing elements. The characteristic defect frequency of each bearing element may also be calculated theoretically [109].

3.7.3 The Use of Fast Fourier Transform for Fault Diagnosis

The early study of FFT application for faults diagnosis was investigated by McFadden and Smith [110]. They developed the procedures to obtain the spectrum of the envelope signals for bearing monitoring by using the high-frequency resonance technique.

The basic FFT technique for fault diagnosis involved presenting and analysing the vibration data only from the vibration spectrum. Birajdar et al., [40] described sources of vibrations in centrifugal pumps e.g., mechanical causes of vibrations, hydraulic causes of vibrations, and peripheral causes of vibrations. In their study, the FFT reading showing vibration peak due to several causes such as unbalance, eccentricity, bent shaft, misalignment, blade pass and vane pass vibration, flow turbulence, and cavitation were explained.

The use of power spectra density (PSD) for fault diagnosis of hydraulic pump was demonstrated by Mollazade et al., [111]. In principle, PSD is a plot of power spectrum from a signal’s power which is given in the frequency bins. The power spectrum is commonly produced using the FFT. The vibration signal was taken from various pump conditions such as normal, inner race defect, outer race defect, and gear tooth defect. It was found that between frequency ranges 70-120 Hz, the strong peak in the PSD was observed for all pump conditions. The fault diagnosis of the pump was then performed by comparing the area under PSD versus frequency.

Ho and Randall [112] investigated the envelope's spectrum performance to separate the signals from background noise by using the squared envelope. In this study, the bearing faults were digitally simulated which included random fluctuations of an excitation pulse for the bearing and a varying load angle of a rolling element. The results concluded that if the random fluctuation of an excitation pulse for the bearing was less than 1% then the use of the squared envelope gained an improvement if the ratio of MSR to SNR was greater than a factor of 0.2.

Rai and Mohanty [113] proposed a fault diagnosis technique for bearings using the FFT based Hilbert-Huang Transform (HHT). The HHT is most suitable for the signal that is nonlinear and nonstationary. It is a method to decompose a signal into intrinsic mode functions (IMF) in order to obtain the instantaneous frequency. The result showed that HHT with the FFT intrinsic mode functions (IMF) clearly indicated the effectiveness of HHT in detecting bearing faults.

The use of the FFT by means of envelope analysis for cavitation detection in a centrifugal pump was investigated by Tan and Leong [107]. The experimental testing considered 3 conditions of the centrifugal pump, e.g., at the condition of best efficiency point (BEP), 90% BEP, and 80% BEP. Vibration signals were acquired at several locations such as from the pump inlet, bearing, pump outlet and casing. The envelope spectra was compared between cavitation and non-cavitation conditions. The research concluded that envelope analysis was able to detect cavitation over 3 testing conditions.

Farokhzad et al., [114] proposed a method which used statistical parameters extracted from frequency domain and artificial neural networks (ANN) for fault diagnosis of a centrifugal pump. A set of potential statistical features such as standard deviation, mean, skewness, kurtosis and variance were extracted from the frequency domain of vibration signals. Those features were then inputted into an ANN. In this study, an ANN model was developed based on a multi-layer perceptron neural network and back propagation algorithm. Several centrifugal pump conditions were introduced to test the method such as normal condition (no fault), impeller fault, cavitation, and seal fault. It was found that the proposed method was capable of classifying the pump condition with 100% accuracy. This accuracy was achieved using a neural network model with one hidden layer.

In addition, Farokhzad [115] applied the FFT technique to extract features from vibration signals for faults diagnosis of centrifugal pumps and used those features as input vectors to an adaptive neuro-fuzzy inference system (ANFIS). The study considered various condition of the centrifugal pump like healthy, broken impeller, worn impeller, leakage, and cavitation. A set of statistical features from the frequency domain (standard deviation, kurtosis, mean, skewness, RMS, and sample variance) were extracted to reflect different types of faults. The performance of the method was validated by applying the testing data set into the trained ANFIS model. The result demonstrated an accurate and automatic classification technique, by which the total classification accuracy was 90.67%.

Moosavian et al., [116] investigated a method for fault diagnosis of main journal-bearings of an internal combustion (IC) engine based on combinations of PSD technique and two classifiers, namely k-Nearest Neighbors (kNN) and artificial neural network (ANN). The study aimed to compare the role of PSD, kNN, and ANN in the IC engine fault diagnosis. PSD was used to analyse the vibration signals collected from three different conditions of journal-bearings: normal, with oil starvation condition and extreme wear fault. Thirty frequency-domain feature parameters were extracted from PSD values and were then used as inputs to the classifier. The kNN and ANN were trained using training data sets and their performance was computed by using the testing data sets. The variable *k* value and hidden neuron (N) were varied from 1 to 20 to gain the best classification performance. The result showed that both classifiers could reliably separate different fault conditions of journal-bearings of an IC engine. However, the performance of ANN was better than kNN.

3.8 Wavelet Transforms

Wavelet transform (WT) is another signal processing tool based on a time-scale representation of a signal. The first study of the wavelet transform was introduced by Alfred Haar in 1909. However, Jean Morlet and Alex Grossman were the ones who proposed the concept of the wavelet and invented the term wavelet. The first invented and the simplest orthogonal wavelet is named the Haar wavelet [117].

WT is a tool that transforms a time domain signal into a different form such as various wavelet coefficients in the time-frequency domain [118]. In principle, the wavelet transform utilises different sized scaling factors (window or scale) for viewing and analysing signals. A large scale corresponds to a big frame for analysing a signal, whilst a small scale corresponds to a small frame for viewing the details of a signal. In other words, the wavelet transform has the capability for zooming in and zooming out of the signals [59].

The wavelet transform uses a small wave, called wavelet function, which has wavelike features and energy that is compressed in a short time. The wavelet function acts as a basic function to produce localised features of the original in a scaled domain [118].

A wavelet function has a property called convolution where a wavelet can be reversed, shifted, integrated, and multiplied with a particular signal in order to extract information. The wavelet function also has a scalable and modulated characteristic which allows it to be shifted over the signal as depicted in Figure 3.4. The process can be repeated by using different scales of the window (slightly shorter or longer) and the result will be a representation of a signal in time-frequency with different resolutions.[59]. This unique characteristic is very useful for solving signal cutting problems and for an analysis of non-stationary signals.

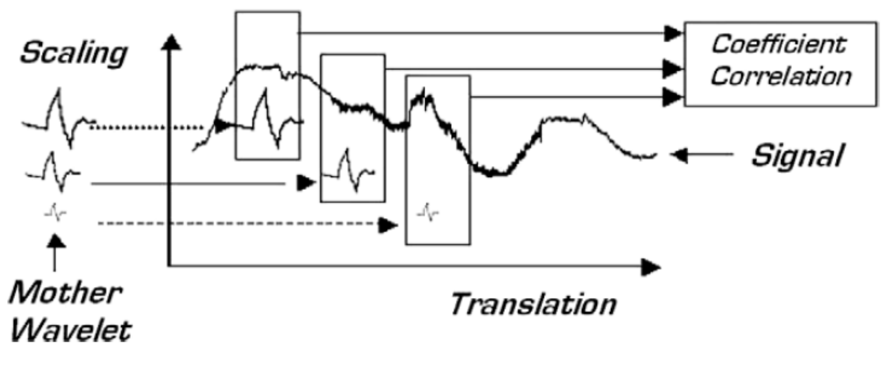


Figure 3.4 The fundamental principle of wavelet transform [119]

In the Fourier transform, a signal is decomposed into various sine waves with different frequency. Meanwhile, with the wavelet transform, a signal is decomposed into shifted and scaled versions of the original (or *mother*) wavelet. During the transformation process, the scaled mother wavelet is translated from the beginning to the end of the analysed signal and the process is repeated with a new determined scale of the wavelet function.

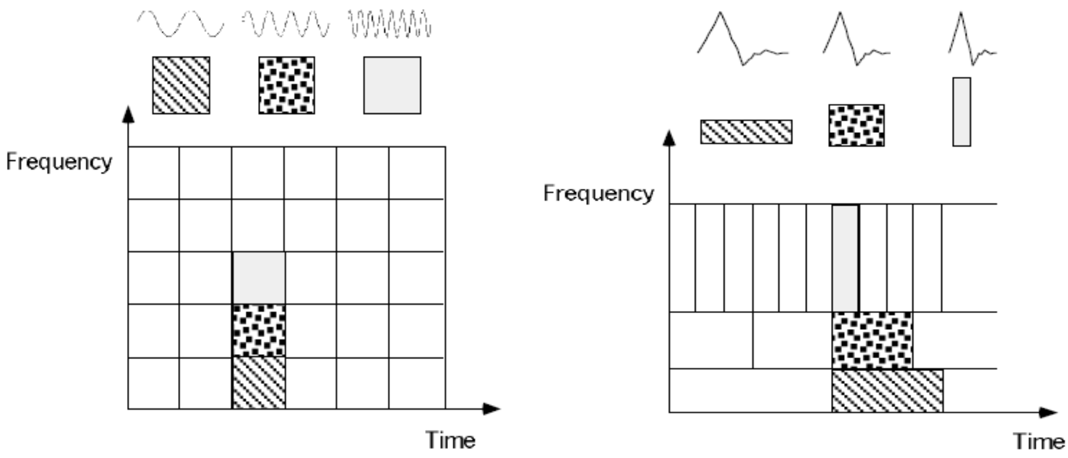
The result of the wavelet transform are segments of signals which consist of approximated versions, labelled by ‘a’ and detailed version, labelled by ‘d’. The characteristic of the approximated version is low frequency content which is used to approximate the original processed signal, whilst the detailed version contains high frequency information of the processed signal.

The wavelet transform was developed to overcome the limitation of the Fourier transform. In the FT, the original signal is decomposed into continuous sine and cosine functions which may not always be suitable to particular needs; this is because sine and cosine functions are not concentrated in space. For instance, if sine and cosine functions are used in the approximation of non-stationary signals then it will end with unsatisfactory results.

The advantage of the wavelet transform includes its capability to analyse signals in different frequency bands with a resolution based on the wavelet scaling factor. Graps [120] stated that if a longer scale of wavelet is used, then the transformation results in low frequency information of the signals, meanwhile if a shorter scale of wavelet is used, then the high frequency information of the signal is obtained.

The wavelet transform differs from the Fourier transform since the wavelet function is in the time-frequency plane [121]. Figure 3.5(a) depicts a Fourier transform with a simple square wave window. The sine and cosine function are trimmed with a single size of square wave window. Because only one window size is used along the whole signal, the result will give the same resolution in the time-frequency plane.

On the other hand, the wavelet transform uses windows with varying width. For the purpose of isolating signal discontinuities, the WT applies very short wavelet functions. Meanwhile, in order to obtain high frequency information, WT uses a long wavelet function. This can be achieved by applying long low-frequency functions and short high-frequency basis ones. Figure 3.5(b) illustrates the basis function coverage of Daubechies wavelet in the time-frequency plane.



(a) (b)

Figure 3.5 Basis function and time-frequency resolution plane of (a) the Fourier transform and (b) the Daubechies wavelet [120]

There are many types of wavelet families which are commonly used in signal analysis application e.g., Daubechies wavelets, Coiflets, Biorthogonal wavelets, Symlets, Mexican Hat, Morlet wavelet, etc.

The Daubechies *N* wavelet (*dbN*) was defined by Ingrid Daubechies [122] which is commonly used in various applications. Latuny [59] applied the *db4* wavelet type to extract features for the purpose of training an ANFIS for a bearing fault classifier. Muralidharan and Sugumaran [123] used *db1 – db10* wavelets for feature extraction from raw signals for fault diagnosis of a centrifugal pump.

The Symlet wavelet family was used by Muralidharan and Sugumaran [124] for feature extraction in the application of the centrifugal pump fault diagnosis. One of the properties of the Symlet wavelet is that it is orthogonal. An orthogonal wavelet is a discrete wavelet transform in which the adjoint of the wavelet transform is equal to the inverse of wavelet transform. Figure 3.6 and Figure 3.7 show the Symlet wavelet function of *sym2* to *sym17* and Symlet scaling functions of *sym2* to *sym17*.

A wavelet, or mother wavelet or analysing wavelet $\psi(t)$, is a waveform which has a finite duration. Wavelets consist of dilations and translations of a function $\psi(t) \in L^2(C)$, where $L^2(C)$ is a complex function. Dilation means a scaling of the argument, so that a given function $\psi(t)$ and a parameter $s > 0$, $\frac{1}{\sqrt{s}}\psi\left(\frac{t}{s}\right)$ gives a dilation of $\psi(t)$. The dilation of a function gives either a spreading out or contraction to the function. The factor $\frac{1}{\sqrt{s}}$ provides normalization needed to have an orthonormal wavelet basis. A translation corresponds to a shift of the argument along the real axis, so that for a given parameter τ, the translation of the function $\psi(t)$ by τ is $\psi(t-\tau)$.

A mother wavelet $\psi(t)$ can be dilated by scale s and time-translated by factor τ as follows [125]:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right), \quad s,\tau \in R, \ \ s > 0 \qquad 3.12$$

For any time-variable signal $x(t)$, the wavelet transform of $x(t)$ which has a mother wavelet $\psi(t)$ at scale s and translation τ is defined by the convolution of $x(t)$ with a scaled and conjugated $\psi(t)$:

$$Wx(s,\tau) = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{s}}\psi^*\left(\frac{t-\tau}{s}\right)dt\,, \qquad 3.13$$

where Wx is the wavelet transform of $x(t)$ and $\psi(t)^*$ is the complex conjugate of $\psi(t)$.

The result of the wavelet transform $Wx(s,\tau)$ is defined in the $s-\tau$ plane, where s determines frequency and τ relates wavelet time location. In principle, the changes in the value of s relates to the alteration of time and frequency of the signal. At the same time, as the value of τ increases, the analysing wavelet shifts along the length of the analysed signal [119].

Generally, the wavelet transform may be divided into the continuous wavelet transform (CWT) and discrete wavelet transform (DWT). In the following section, a brief explanation on each of them is introduced.

3.8.1 Continuous Wavelet Transform (CWT)

The CWT of a signal $x(t)$ is carried out through convolution between the signal $x(t)$ and complex conjugate mother wavelet $\psi(t)$ [118]:

$$cwt(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-\tau}{s}\right) dt\,, \tag{3.14}$$

where s represents scale and τ denotes translation factors. $\psi^*(t,s,\tau)$ indicates the complex conjugate of mother wavelet $\psi(t)$ which is scaled and shifted with s and τ factor respectively. Since the CWT has two parameters, s and τ, a transformation of a signal using the wavelet transform produces a two-dimensional projection in time-scale plane.

3.8.2 Discrete Wavelet Transform (DWT)

The DWT is the discrete form of CWT by discretising the mother wavelet $\psi_{s,\tau}(t)$. This may be achieved using a dyadic discretisation which is well known as a wavelet discretisation technique [126] and is given mathematically by,

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j k}{2^j}\right), \tag{3.15}$$

where s is replaced by 2^j and τ by $2^j k$. The discrete form of a wavelet may allow the implementation of a computational wavelet transform.

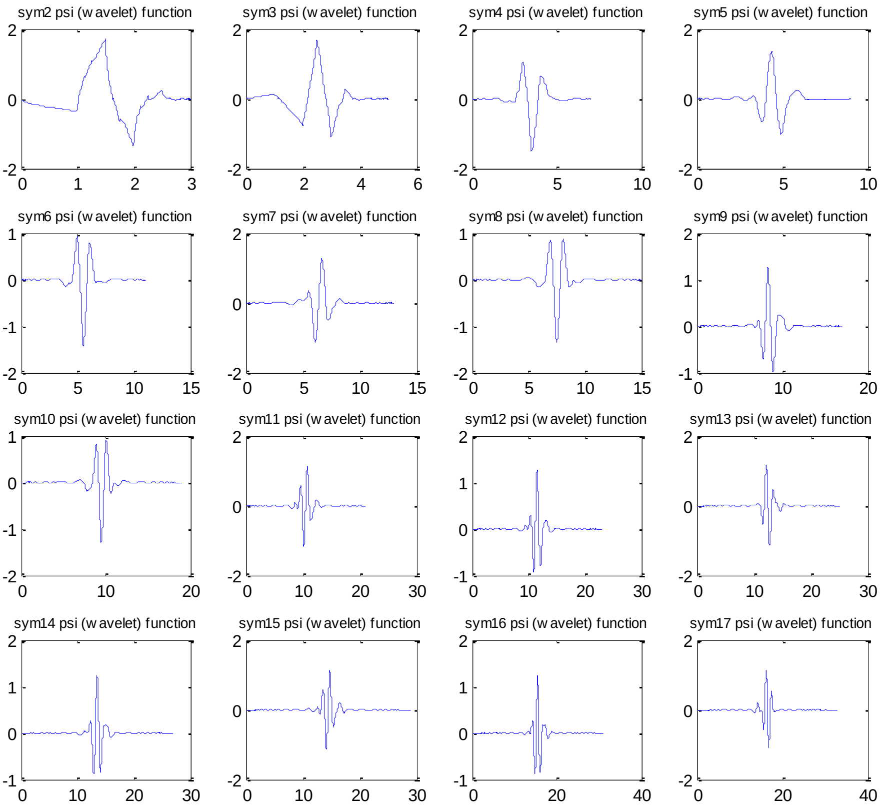


Figure 3.6 Symlet wavelet function

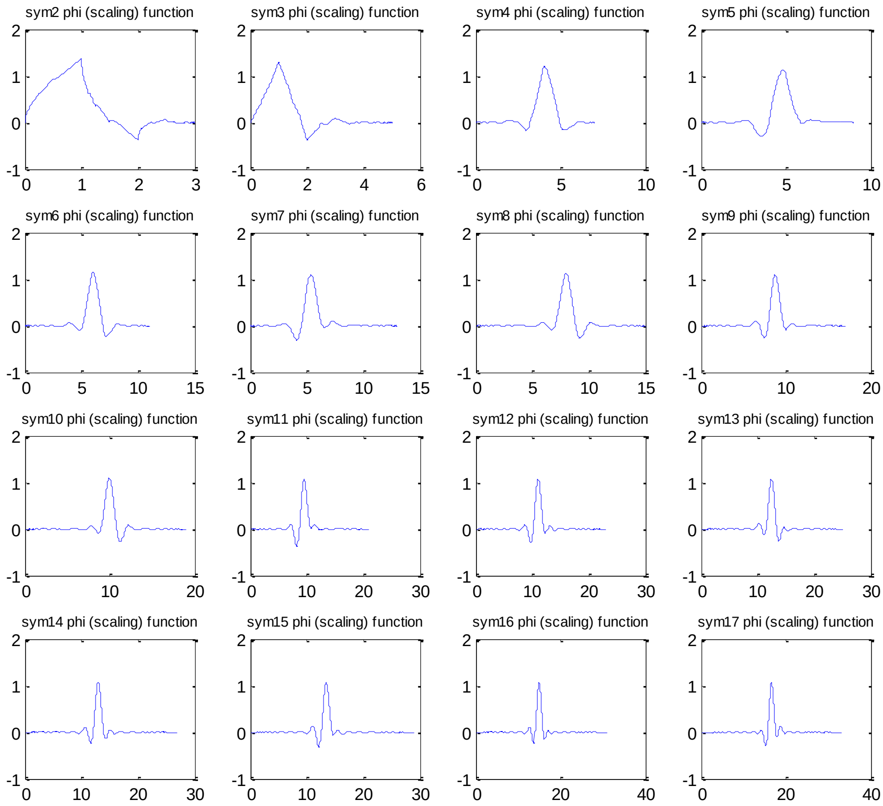


Figure 3.7 Symlet wavelet scaling function

3.8.3 Decomposing and Reconstructing a Signal Using the Wavelet Transform

The wavelet transform has an important property which allows it to decompose a signal into a high frequency and low frequency part by applying a specific filtering process [127]. The filtering process may consist of a low and a high-pass wavelet filtering analysis which is used in the filtering of a discrete signal [128]. Figure 3.8 shows a schematic diagram of the concept of the wavelet transform filtering process.

Figure 3.8 shows a wavelet implementation in decomposing and reconstructing a discrete signal using a set of wavelet filters. The fundamental idea of a signal decomposition is represented in Figure 3.8a where a convolution is achieved through the use of a low pass filter F_L, and a high pass filter F_H. The discrete signal s(t) is passed to filter F_L and F_H which produces *Approximate Coefficients, cA* and *Detail*

Coefficients, *cD*. The symbol (↓2) represents down-sampling in the decomposition process. The process can be achieved by discarding the odd coefficients of the filtered signal which then results in a number of coefficients in *cA* and *cD*. The total number of coefficients in *cA* and *cD* is approximately equal to the number of the filtered signal.

(a) Decomposition

(b) Reconstruction

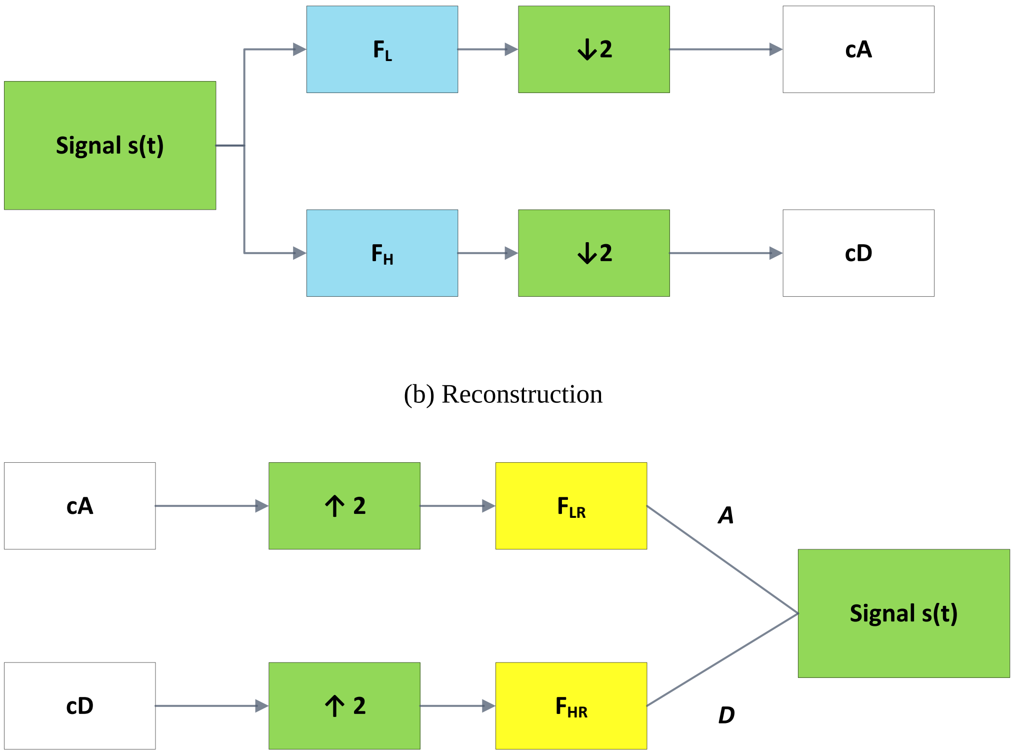


Figure 3.8 Decomposition and reconstruction process

The reconstruction process, as can be seen in Figure 3.8b, is performed by convolving both vectors *cA* and *cD* using a pair of low-pass (F_{LR}) and high pass (F_{HR}) reconstruction filters. The process produces two reconstructed signal namely *A* and *D* which are known as the reconstructed *Approximate Coefficient* and the reconstructed *Detail Coefficient* respectively.

An up-sampling process (↑2) is performed during the reconstruction process which it involves the addition of zeros between the coefficients of the vectors *cA* and *cD*. The reconstruction process is subsequently carried out by adding coefficients *A* and *D* which produce a complete form of the original signal *s*.

Figure 3.9 and Figure 3.10 show the application of a wavelet transform for filtering a noisy signal by using a low and a high pass filter. A noisy sinusoidal signal was processed using a symlet 8 (sym8) wavelet up to five levels of decomposition. Figure 3.9a shows the original signal and Figure 3.9b show the filtered signal using two levels of decomposition. It is shown that the noise level is significantly reduced after the original signal is filtered using two levels of decomposition. By increasing number of levels of the decomposition process, the noise level even can be more reduced as can be seen in Figure 3.10.

Figure 3.10b shows a clean signal which is achieved by decomposing the original signal up to five levels of decomposition.

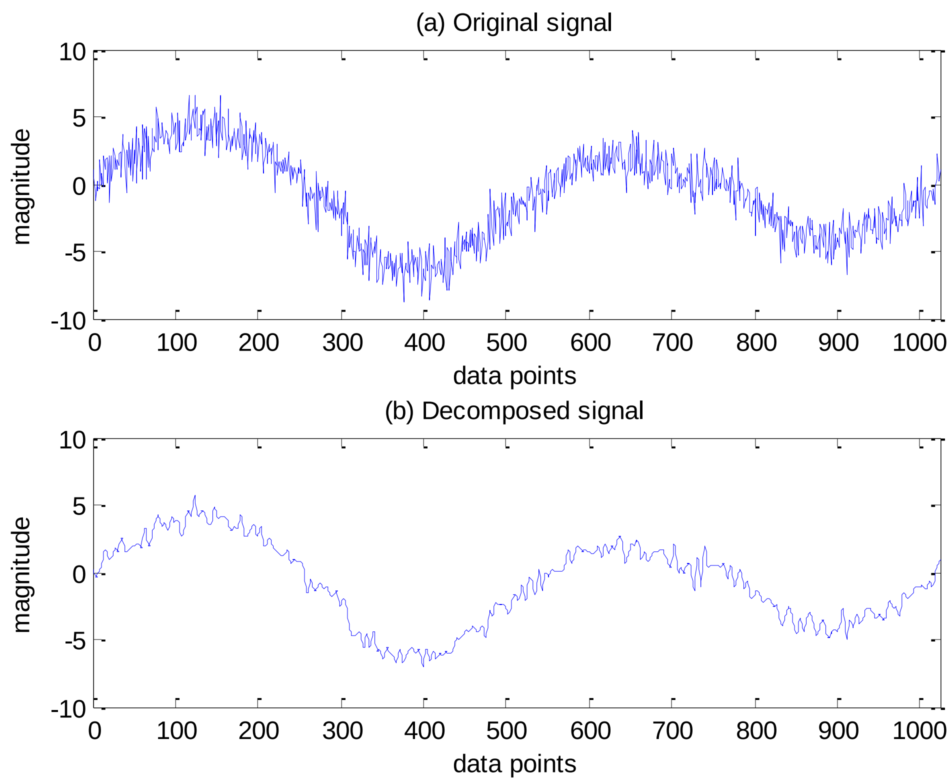


Figure 3.9 Two levels decomposition using symlet 8 wavelet transform

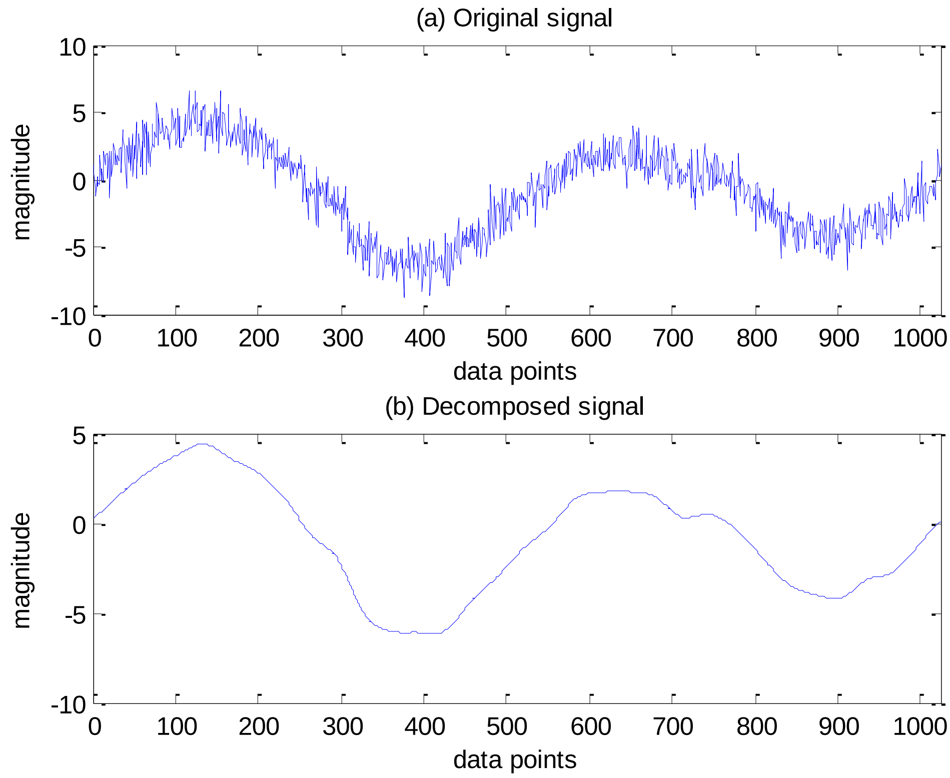


Figure 3.10 Five levels decomposition using symlet 8 wavelet transform

3.8.4 Multi Resolution Analysis (MRA) Using Discrete Wavelet Transform

Discrete wavelet transform (DWT) can be used to transform a signal using Multi Resolution Analysis (MRA) [129]. Figure 3.11 shows a schematic diagram of the principle of multi resolution analysis [130] applied to a signal up to 5 levels of decomposition. The MRA decomposes the original signal using DWT which produces two parts in the signal, Approximated (a) and Detailed (b). The index in each approximate and detailed part, *a1* to *a5* and *d1* to *d5*, corresponds to the decomposition level at the related level.

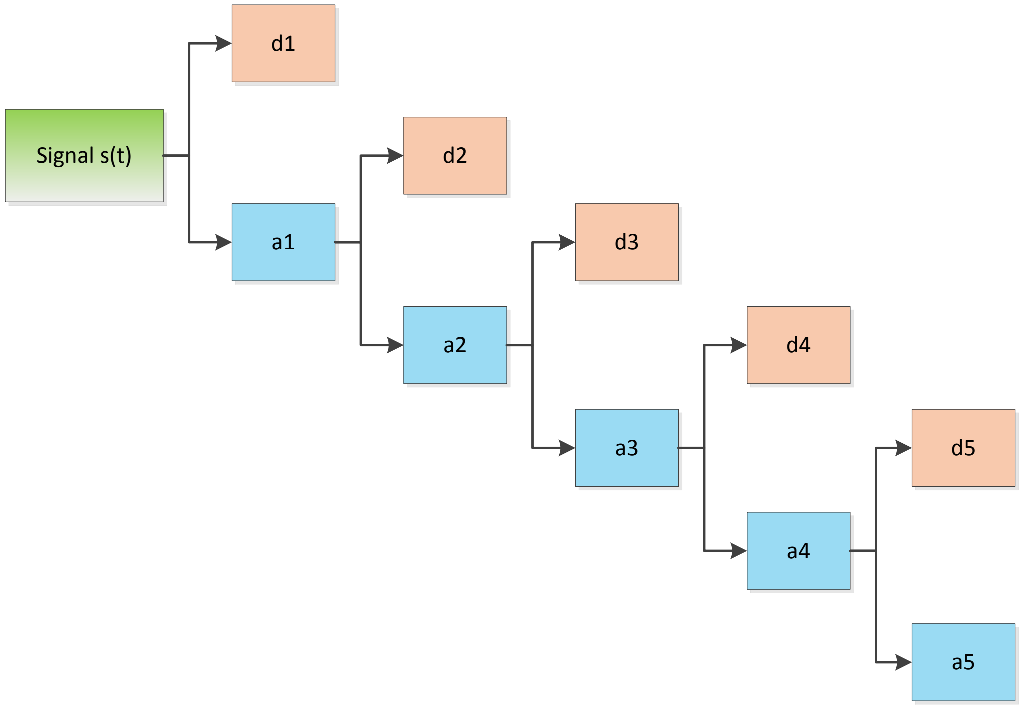


Figure 3.11 Principle of multi resolution analysis

The division of frequency sub-bands of a signal processed using DWT MRA follows a standard rule. The rule is applied for each approximation (a) and detailed (d) part at each level of decomposition [131].

Figure 3.11 shows an example of the decomposition process using DWT MRA of a signal with the maximum frequency up to 10 kHz. The signal is decomposed up to 5 levels and the frequency sub-band divisions of approximate (a) and detailed (d) parts from level 1 to 5 are given in Table 3.3. The decomposition process starts at the first level of DWT MRA which transforms the signal into two parts, *a1* and *d1*. The frequency sub-bands for the first level of approximate part, *a1*, ranges from 0−5000 Hz and for the first level of detailed part, *d1*, ranges from 5000−10000 Hz. The remaining frequency sub-band divisions corresponding to each level of decomposition can be seen in Table 3.3.

Table 3.3 Frequency sub-bands of five decomposition levels using DWT MRA

Approximation parts	Sub-bands (Hz)	Detailed parts	Sub-bands (Hz)
a1	0 − 5000	d1	5000 − 10000
a2	0 − 2500	d2	2500 − 5000
a3	0 − 1250	d3	1250 − 2500
a4	0 − 625	d4	625 − 1250
a5	0 − 312.5	d5	312.5 − 625

Figure 3.12 and Figure 3.13 show an example of a DWT MRA implementation to a noisy sinusoidal signal processed using 5 levels of wavelet transforms (sym8 wavelet). The results show an obvious gradually reduced noise from level 1 to level 5.

The result of approximated (a) parts after 5 levels decomposition is depicted in Figure 3.12 while the detailed (d) parts is shown in Figure 3.13. The detailed parts contain high frequency components of the original signal which are related to the noise.

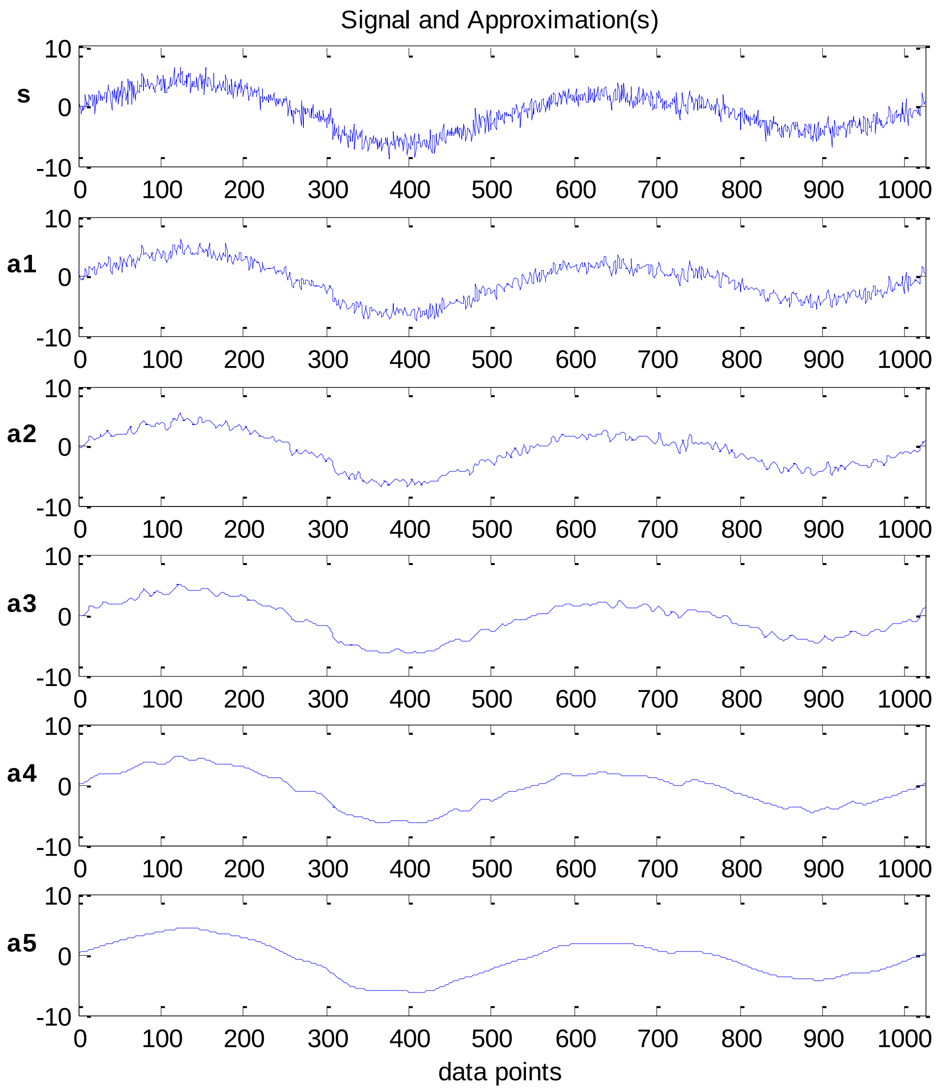


Figure 3.12 Approximated parts, *a1*-*a5*, of a noisy sinusoidal signal

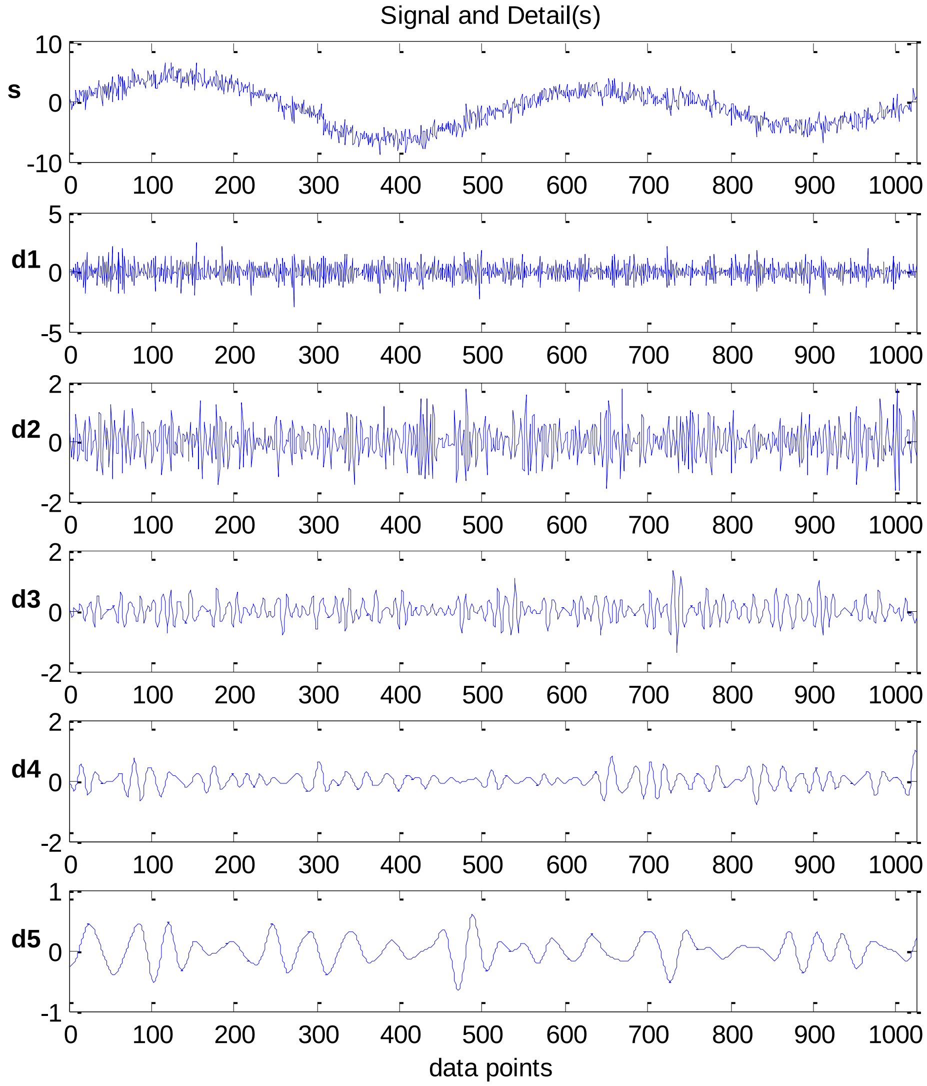


Figure 3.13 Detailed parts, *d1-d5*, of a noisy sinusoidal signal

3.9 The Application of Wavelet Transform in Fault Analysis

The wavelet transform is a popular technique applied for feature extraction of vibration signal for machine fault diagnosis. Many literatures related to machine fault diagnosis which is based on the wavelet transform have been written. In this section, a brief discussion of wavelet application in machine fault diagnosis is presented.

The early application of wavelet analysis for fault diagnosis was proposed by Leducq in 1990 [132]. The study investigated the use of the wavelet transform for hydraulic noise detection of a centrifugal pump. It was considered as one of the first publications to use wavelets in machine fault diagnosis.

Prabhakar et al., [133] investigated the application of the discrete wavelet transform for single and multiple point defects of bearings. The proposed method was tested using vibration signals collected from several types of ball bearing faults such as outer race failure, inner race failure and combination faults. The result showed that wavelet decomposition can detect the impulse due to either single or multiple faults in the ball bearing effectively.

Lou and Loparo [134] developed a scheme based on the combination of wavelet transform and neuro-fuzzy classification for bearing fault diagnosis. The scheme proposed a technique to extract features from an accelerometer signal using a wavelet transform. Three conditions of the bearing were included in the experiment i.e., normal bearings, inner race faults and ball faults. As a fault classifier, an adaptive neuro-fuzzy inference system (ANFIS) was trained and proposed. Test results showed that the proposed fault diagnosis scheme can effectively distinguish different fault types under varying load conditions.

Gao et al., [135] undertook a comparative experimental study on performance monitoring of a hydraulic pump using the conventional FFT based technique and a wavelet based multi-resolution analysis method. In the study, outlet pressure data from the pump was selected as the signal. The pressure signal was then analysed using FFT and wavelet packet analysis. The performances of both methods were tested based on simulation and experimental results. It was noted that the wavelet transform based fault diagnosis gave a more sensitive and robust detection result for all three conditions of the pump (normal, control plate wear, and loose ball-socket joints).

Muralidharan and Sugumaran [124] compared the performance of the combination of a wavelet transform with two classifiers namely Naïve Bayes and Bayes net classifier for fault diagnosis of a centrifugal pump. In their study, three analysis steps were proposed: feature extraction, classification, and comparison of classification performance. The wavelet transform was used to extract features from the vibration signal acquired from the normal and faulty centrifugal pump components. Six wavelet families were considered in the study, namely Symlets, Daubechies, Meyer, Coiflet, Bi-orthogonal, and Reversed Bi-orthogonal. It was found that a combination of feature extraction using the wavelet and the Bayes classifier was a promising candidate for fault diagnosis of centrifugal pumps.

Muralidharan and Sugumaran [123] extended their investigation to wavelet transform for fault diagnosis of centrifugal pumps. In this study, the discrete wavelet transform was used to extract wavelet features from the vibration signals. The rules were then generated using rough set theory and the classifier developed using fuzzy logic. Various different faults were introduced in this study namely normal condition (without fault), bearing fault, impeller fault, cavitation, and the combination of bearing fault and impeller fault. The result showed that the wavelet features with a rough set and fuzzy logic was able to identify the presence of faults in the centrifugal pump.

3.10 Concluding Remarks

The vibration-based technique is an approach commonly used in many fault diagnosis applications. The popularity is due to its effectiveness and versatility, in which the health condition of the machine has a strong correlation with the level of vibration severity of a component. There are three categories of the vibration-based technique that are commonly used, namely time-domain analysis, frequency-domain analysis, and time-frequency domain analysis.

The use of the statistical parameters extracted from time- and/or frequency-domain becomes a popular method and has a wide range of applications in fault diagnosis. It has been found that statistical parameters have an advantage where the variation and speed condition of a machine has a lower effect on their values. In addition, there is a trend in fault diagnosis method to process the features extracted from time- and/or frequency-domain using the methods like the PCA, ANN, kNN, fuzzy method, etc. It is shown that those methods achieve a reliable performance in monitoring the health condition of machinery.

The next chapter describes techniques and literature reviews of Principal Component Analysis (PCA) in fault analysis. It also presents the application of PCA in dimension reduction, fault detection, and feature extraction.

CHAPTER FOUR

4 A Review of Principal Component Analysis in Fault Diagnosis: Techniques and Literature

In many engineering fields, it is often that one engages with complex systems which are a result of sophisticated developments with machinery, control systems and industrial processes. As a consequence of increased complexity, there is an increased risk of failure of components which requires an immediate maintenance action.

The complexity of modern machinery may cause the number of variables that must be monitored to sharply increase, causing difficulties in analysis. However, these variables can often be found to be correlated to each other. That is, there can be a substantial redundancy among variables which leads to high correlation and multicollinearity.

Principal component analysis (PCA) is a multivariate statistic technique which linearly transforms original data into a new space and may reduce a number of original data dimensions down to only a couple of dimensions without losing a significant amount of information. PCA attempts to re-orient the original data in such a way that only the first several dimensions may retain as much of the available details in the original data as possible [136]. PCA may avoid redundancy in the original data by holding most of the details in the original data using a very few dimensions.

In essence, PCA projects the original data into the first principal component (PC) so that the projected data has a maximum variance and into the second principal component (orthogonal to the first PC) so that the variance of the projected data is as large as possible. This principle continues to the remaining PCs which in turn produces a set of projected data in the PCs.

The smaller number of dimensions makes interpretation of the data easy and simple and the following analysis becomes more convenient. It may also significantly reduce the computational cost.

The principal components are orthogonal each other, that is, each component is uncorrelated with all the others. This gives an advantage, by means of eliminating multicollinearity in the data set.

PCA may be used not only for dimensional reduction of the original data but can also be used for fault detection and monitoring through the use of PCA being combined with Hotelling's T^2 and Q-statistic [52, 137, 138]. T^2-statistic indicates the variation of each sample inside the PCA model whereas the Q-statistic represents the residual between the original data and projected data in the PCs retained within the model.

The following sections discuss details of the PCA technique and its application in monitoring and fault diagnosis.

4.1 Principal Component Analysis

PCA has been applied successfully for monitoring and fault diagnosis in many scientific fields [139]. Many statistical techniques have been developed to draw essential information out from large data sets and to analyse the information. PCA has been implemented for those tasks more frequently in fault detection and in a diagnosis scheme [140].

It is generally accepted that Pearson in 1902 and Hotelling in 1933 proposed the first technique to reduce multivariate data dimension which is recently known as PCA [141].

The main idea of PCA is to reduce the original data set dimension which most likely has considerable amount of multicollinearity between variables while at the same time attempts to hold as much of the variation in the model as possible. PCA transforms the original data set to new orientations namely principal components in such a way that they are uncorrelated and the first PCs hold most of the inherent variance of the original data set.

The motivation underlying the use of PCA to reduce multidimensional original data to a smaller number of dimensions is based on the possibility that the implicit relationship between variables in a complex system can often be quite simple. PCA may give a procedure to reduce the original data set dimension and seek some essential patterns which frequently influence it [138].

PCA constructs principal components through linearly transforming the original variables using eigenvalue decomposition of the original variables covariance matrix [142]. PCA also may be used to represent the similarity of patterns in the observations and variables by plotting them as points on a graph [143].

4.1.1 The Mechanics of PCA

Consider the original data set to be a matrix of $\mathbf{X} \in \mathbb{R}^{m \times n}$ where m represents observations and n represents variables and is given mathematically as,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mj} & \dots & x_{mn} \end{pmatrix}, \qquad 4.1$$

or in column form as,

$$\mathbf{X} = (\mathbf{w}_1\ \mathbf{w}_2 \dots \mathbf{w}_j \dots \mathbf{w}_n),$$

Row vector $\mathbf{x}_i$ represents all variable measurements at the specific time instants, meanwhile column vector $\mathbf{w}_j$ represents one variable measurements over the whole experimental time.

In general, due to different magnitude and scales of the physical variables, the matrix $\mathbf{X}$ needs to be scaled before further processing. One of the scaling techniques is auto scaling which re-scales the original data to have a mean of 0 and a variance of 1 by transforming column vector $\mathbf{w}_j$ as given by,

$$\mu_{\mathbf{w}_j} = \frac{1}{m}\sum_{i=1}^{m} x_{ij}, \qquad 4.2$$

$$\sigma_{\mathbf{w}_j} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(x_{ij}-\mu_{\mathbf{w}_j}\right)^2}, \qquad 4.3$$

$$\bar{x}_{ij} = \frac{x_{ij} - \mu_{\mathbf{w}_j}}{\sigma_{\mathbf{w}_j}} , \qquad 4.4$$

where $\mu_{\mathbf{w}_j}$ represents mean and $\sigma_{\mathbf{w}_j}$ denotes standard deviation of variable $\mathbf{w}_j$ whereas $\overline{x}_{ij}$ is the data point which re-scaled to $\mu_{\mathbf{w}_j} = 0$ and $\sigma_{\mathbf{w}_j} = 1$. Subsequently, the re-scaled data is written without bar notation for convenience.

The covariance matrix **X** is then defined as,

$$\mathbf{C}_{\mathbf{X}} = \frac{1}{m-1}\mathbf{X}^{\mathrm{T}}\mathbf{X} ,$$

which can be written as,

$$\mathbf{C}_{\mathbf{X}} = \frac{1}{m-1}\begin{pmatrix} \mathbf{w}_1^{\mathrm{T}}\mathbf{w}_1 & \mathbf{w}_1^{\mathrm{T}}\mathbf{w}_2 & \cdots & \mathbf{w}_1^{\mathrm{T}}\mathbf{w}_j & \cdots & \mathbf{w}_1^{\mathrm{T}}\mathbf{w}_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_1 & \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_2 & \cdots & \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j & \cdots & \mathbf{w}_j^{\mathrm{T}}\mathbf{w}_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{w}_n^{\mathrm{T}}\mathbf{w}_1 & \mathbf{w}_n^{\mathrm{T}}\mathbf{w}_2 & \cdots & \mathbf{w}_n^{\mathrm{T}}\mathbf{w}_j & \cdots & \mathbf{w}_n^{\mathrm{T}}\mathbf{w}_n \end{pmatrix} . \qquad 4.5$$

The covariance matrix **X** is a square and symmetric matrix with size of $n \times n$. It quantifies the amount of linear relationship between all possible combinations of two variables within the data set. The terms in the main diagonal of matrix **X** are variances of associated variables, meanwhile the covariance between all combinations of two variables are in the off-diagonal terms. The variance is defined as,

$$\sigma_{\mathbf{w}_j}^2 = \frac{1}{m-1}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j , \qquad 4.6$$

which can be expressed as,

$$\sigma_{\mathbf{w}_j}^2 = \frac{1}{m-1}\sum_{i=1}^{m} x_{ij}^2 ,$$

while the covariance can be expressed as:

$$\sigma^2_{\mathbf{w}_j,\mathbf{w}_k} = \frac{1}{m-1}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_k \quad , \mathrm{j} \neq \mathrm{k}\,, \tag{4.7}$$

which can be written,

$$\sigma^2_{\mathbf{w}_j,\mathbf{w}_k} = \frac{1}{m-1}\sum_{i=1}^{m} x_{ij}x_{ik}$$

In general, larger variance values indicate more essential information contained in the variables, while large covariance values represent high redundancy between any two of variables in the data set [136]. Matrix **X** consists of n dimensional space with n orthonormal basis vectors. PCA attempts to transform the vectors $\mathbf{x}_i$ into a different orthonormal basis which may reveal hidden features of interest.

PCA decomposes matrix **X** into a score matrix **T** and a loading matrix **P** through singular value decomposition (SVD) as [144],

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{t}_2\mathbf{p}_2^{\mathrm{T}} + \cdots + \mathbf{t}_n\mathbf{p}_n^{\mathrm{T}}\,, \tag{4.8}$$

which can be written as,

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}}\,,$$

where $\mathbf{p}_i \in \mathbb{R}^{n\times 1}$ represents the eigenvectors of matrix $\mathbf{C}_\mathbf{X}$ and **P** denotes loading matrix of the principal component. The $\mathbf{t}_i \in \mathbb{R}^{m\times 1}$ vectors are a projection of the original data onto the vectors $\mathbf{p}_i$ and **T** represents the score matrix of the principal components. Each eigenvector $\mathbf{p}_i$ corresponds to the eigenvalue λ_i which represents the variance of the vector $\mathbf{t}_i$.

Three important properties of PCs are uncorrelated, ordered from the largest variance, and the first several PCs have a minimal mean-squared approximation error [141]. Matrix **P** is the transformation matrix which has the eigenvectors in its columns, that is,

$$\mathbf{P} = \left(\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_j \cdots \mathbf{p}_n\right), \qquad 4.9$$

and satisfies the eigenvalue-eigenvector property [138],

$$\mathbf{C_X P} = \mathbf{P}\Lambda, \qquad 4.10$$

where $\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$ $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0)$ is a diagonal matrix which has positive real eigenvalues and ordered from the largest to the smallest magnitude.

The transformation matrix can be used to transform the original data into a new space which is mathematically defined as,

$$\mathbf{T} = \mathbf{XP} \qquad 4.11$$

In more detail, this becomes,

$$\left(\mathbf{t}_1\ \mathbf{t}_2 \cdots \mathbf{t}_j \cdots \mathbf{t}_n\right) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mj} & \dots & x_{mn} \end{pmatrix} \times \left(\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_j \cdots \mathbf{p}_n\right) \qquad 4.12$$

Each column of matrix **T** then can be expressed as,

$$\mathbf{t}_j = \mathbf{X}\mathbf{p}_j, \qquad 4.13$$

and the variances of vectors $\mathbf{t}_j$ can be calculated as,

$$\sigma_{\mathbf{t}_j}^2 = \frac{1}{m-1}\mathbf{t}_j^{\mathrm{T}}\mathbf{t}_j = \frac{1}{m-1}(\mathbf{X}\mathbf{p}_j)^{\mathrm{T}}(\mathbf{X}\mathbf{p}_j)\,, \tag{4.14}$$

which can be written as

$$\sigma_{\mathbf{t}_j}^2 = \mathbf{p}_j^{\mathrm{T}}\mathbf{C}_{\mathbf{X}}\mathbf{p}_j = \lambda_j\,,$$

while the covariances are zero, hence,

$$\sigma_{\mathbf{t}_j\mathbf{t}_k}^2 = \frac{1}{m-1}\mathbf{t}_j^{\mathrm{T}}\mathbf{t}_k = \frac{1}{m-1}(\mathbf{X}\mathbf{p}_j)^{\mathrm{T}}(\mathbf{X}\mathbf{p}_k)\,, \tag{4.15}$$

which becomes

$$\sigma_{\mathbf{t}_j\mathbf{t}_k}^2 = \mathbf{p}_j^{\mathrm{T}}\mathbf{C}_{\mathbf{X}}\mathbf{p}_k = \lambda_j\mathbf{p}_j^{\mathrm{T}}\mathbf{p}_k = 0$$

The projection of the original data onto principal components produce the score matrix $\mathbf{T}$ which has uncorrelated column vectors. Each column vector describes variables in a new set of orientation which their variances corresponds to the eigenvalues of the matrix $\mathbf{C}_{\mathbf{X}}$.

The column vectors $\mathbf{p}_j$ in loading matrix $\mathbf{P}$ correspond to the eigenvector of matrix $\mathbf{C}_{\mathbf{X}}$ which are sorted in descending order based on their eigenvalues. Vector $\mathbf{p}_j$ with the highest eigenvalues conveys the largest amount of information within the data set. Geometrically, the data matrix $\mathbf{X}$ is projected over the eigenvector $\mathbf{p}_j$ and produces the vector $\mathbf{t}_j$ of the score matrix $\mathbf{T}$.

The transformed data matrix $\mathbf{T}$ represents a new representation of the original data in a new set orientation. The first few uncorrelated principal components have the most important variation thus hold the most important information within the data.

An example of a PCA application is described here using artificial data that has significant correlation. The two-dimensional scatter plot (X and Y axes) of the artificial data is shown in Figure 4.1. In the Figure 4.1 (a), the data set is uncorrelated, but in the second figure (b), it is significantly correlated.

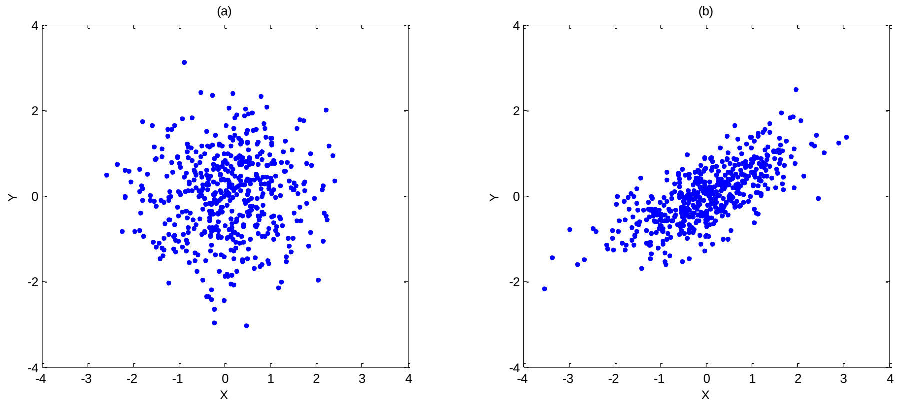


Figure 4.1 Two dimensions artificial data, (a) uncorrelated, (b) correlated.

These plots show a positive correlation between the pair of variables in Figure 4.1(b) which has correlation coefficient r = 0.7, while the pair of variables in Figure 4.1(a) has correlation coefficient of nearly zero, indicating it is uncorrelated. Since the pair of variables in Figure 4.1(b) is significantly correlated, it is possible to transform the original data into a new orientation corresponding to the principal components. In this case, the new transformed data will be uncorrelated and will have maximal variance. Intuitively, the first principal component direction should lie on the longitudinal axis of the data cloud. Analytically, the principal components are determined by calculating the eigenvectors and eigenvalues of the covariance matrix of the original data.

The principal components can then be plotted into the original data cloud as shown in Figure 4.2. The first principal component (PC1) and the second principal component (PC2) are now reassigning values to all of the original data that is achieved by calculating the score matrix **T** using equation 4.11. The transformed data can then be represented with respect to PC1 and PC2 by rotating these axes as shown in Figure 4.3.

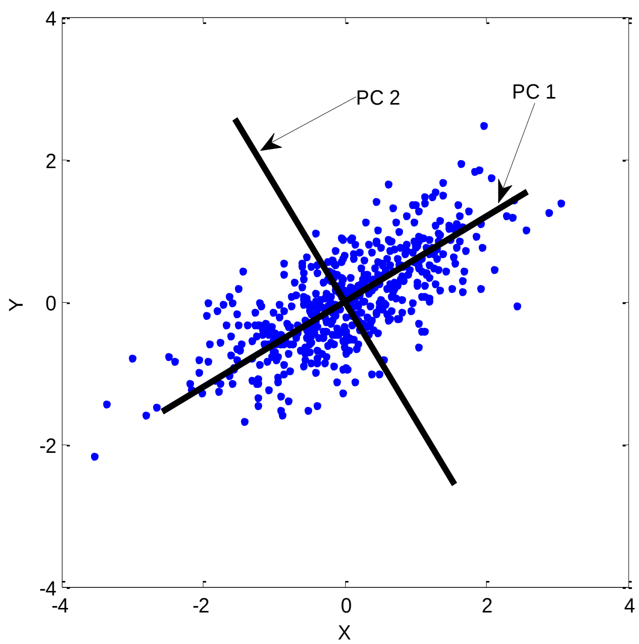


Figure 4.2 Two principal components plotted with the original data

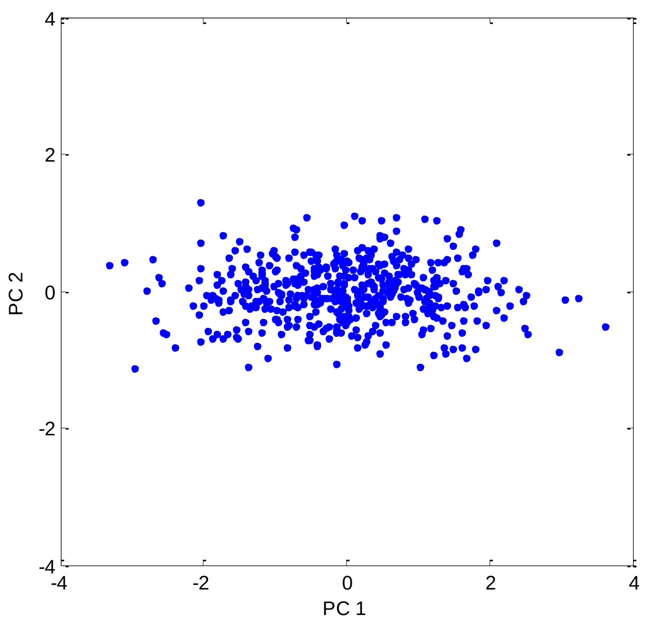


Figure 4.3 The transformed data projected on the two principal components.

4.1.2 Dimension Reduction

The eigenvectors $\mathbf{p}_j$ are sorted in descending order based on the quantity of information. Since most of the information within the data is included in the first few principal components, the dimension of the original data may be reduced down to several r principal components without losing substantial amount of details. PCA assumes the original data is adequately correlated; therefore it needs only a few principal components to cover the most important variation in the data set. The reduced principal component loading matrix **P** is then given by,

$$\widehat{\mathbf{P}} = (\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3 \cdots \mathbf{p}_r)\,, \qquad 4.16$$

and equation 4.8 can be expressed by,

$$\mathbf{X} = \overbrace{\mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{t}_2\mathbf{p}_2^{\mathrm{T}} + \cdots + \mathbf{t}_r\mathbf{p}_r^{\mathrm{T}}}^{\widehat{\mathbf{X}}} + \widetilde{\mathbf{X}}\,, \qquad 4.17$$

or,

$$\mathbf{X} = \widehat{\mathbf{X}} + \widetilde{\mathbf{X}}\,, \qquad 4.18$$

where,

$$\widehat{\mathbf{X}} = \widehat{\mathbf{T}}\widehat{\mathbf{P}}^{\mathrm{T}} = \mathbf{X}\big(\widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\mathrm{T}}\big)\,,$$

and

$$\widetilde{\mathbf{X}} = \widetilde{\mathbf{T}}\widetilde{\mathbf{P}}^{\mathrm{T}} = \mathbf{X}\big(\mathbf{I}_n - \widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\mathrm{T}}\big)$$

Matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{\mathrm{m}\times\mathrm{r}}$ is a reduced matrix established by projecting matrix **X** onto the reduced loading matrix **P** having r selected principal components. Matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{\mathrm{m}\times(\mathrm{n}-\mathrm{r})}$ is a residual matrix created by projecting matrix **X** onto the residual subspace. Expanding equation 4.18 yields,

$$\mathbf{X} = \widehat{\mathbf{T}}\widehat{\mathbf{P}}^{\mathrm{T}} + \widetilde{\mathbf{T}}\widetilde{\mathbf{P}}^{\mathrm{T}} \qquad 4.19$$

which can be written as,

$$\mathbf{X} = \mathbf{X}\widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\mathrm{T}} + \mathbf{X}\left(\mathbf{I}_n - \widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\mathrm{T}}\right), \tag{4.20}$$

where matrices $\widehat{\mathbf{T}} \in \mathbb{R}^{\mathrm{m}\times\mathrm{r}}$ and $\widetilde{\mathbf{T}} \in \mathbb{R}^{\mathrm{m}\times(\mathrm{n}-\mathrm{r})}$ are reduced score matrices which have *r* and $(n - r)$ column vectors respectively. Meanwhile, $\widehat{\mathbf{P}} \in \mathbb{R}^{\mathrm{n}\times\mathrm{r}}$ is a reduced loading matrix retaining *r* principal components and $\widetilde{\mathbf{P}} \in \mathbb{R}^{\mathrm{n}\times(\mathrm{n}-\mathrm{r})}$ represents a reduced matrix in residual subspace with $(n - r)$ ignored principal components.

The matrix $\widehat{\mathbf{X}}$ represents variations of $\mathbf{X}$ calculated using r selected principal components and matrix $\widetilde{\mathbf{X}}$ shows variations due to noise during observation. The calculation of matrices $\widehat{\mathbf{X}}$ and $\widetilde{\mathbf{X}}$ is schematically depicted in Figure 4.4. As an example, if only the first three principal components are sufficient to hold most of the variation, then $\widetilde{\mathbf{X}}$ can be computed by,

$$\widetilde{\mathbf{X}} = \mathbf{X} - [\mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{t}_2\mathbf{p}_2^{\mathrm{T}} + \mathbf{t}_3\mathbf{p}_3^{\mathrm{T}}] \tag{4.21}$$

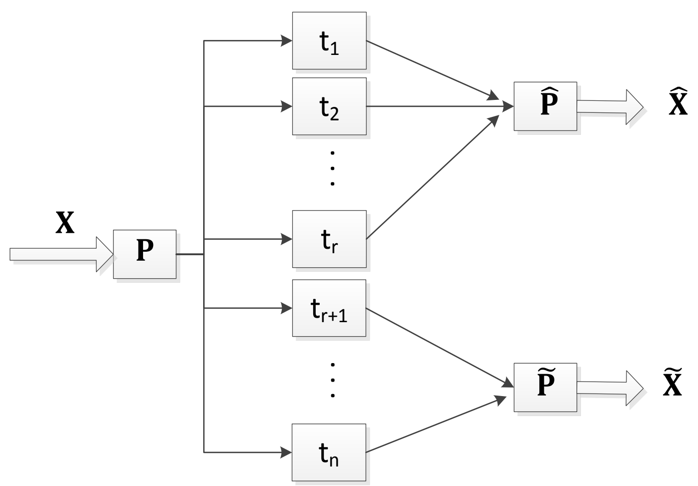


Figure 4.4 Graphical representation of the experimental data as summation of approximation and error using PCA model

The cumulative percent variance (CPV) of the selected *r* principal components can be given by [140],

$$CPV = \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \times 100\%, \qquad 4.22$$

where λ_j represents the *j*th eigenvalue of the covariant matrix $\mathbf{C_X}$. The first *r* largest eigenvalues are retained within the PCA model.

For the example shown previously in Figure 4.1 and Figure 4.2, the eigenvalues for the first PC and for the second PC were 1.2 and 0.2 respectively. Consequently, by just retaining the first principal component, the PCA model still has 86% (as calculated using equation 4.22) of the variation in the data. This provides an advantage of dimension reduction, since the model only lost 14% of the variation by compressing the data by 50%.

4.1.3 PCA Analysis for Fault Detection

The applications of PCA in fault diagnosis commonly use the well-known Hotteling T^2-statistic and the Q-statistic (or Squared Prediction Error (SPE) statistic) [145]. The T^2-statistic measures the variability of the score matrix $\mathbf{T}$. It may detect an abnormal behaviour of new data by comparing variation in the variables to that defined by baseline condition [52]. Meanwhile the Q-statistic evaluates the variability of the matrix $\widetilde{\mathbf{X}}$ which is projection of original data onto residual subspace.

4.1.3.1 T^2-statistic

In essence, the Hotteling's T^2-statistic is a general form of the Student's *t*-statistic which is a popular method for hypothesis testing of multivariate data. It evaluates the variation of the samples inside the model. The T^2-statistic of the *i*th sample or experiment can be expressed by [138, 140],

$$T_i^2 = \sum_{j=1}^{r} \frac{\hat{t}_{\sigma ij}^2}{\hat{\lambda}_j} = \hat{\mathbf{t}}_{\sigma i} \widehat{\Lambda}^{-1} \hat{\mathbf{t}}_{\sigma i}^{\mathrm{T}} = \mathbf{x}_i \widehat{\mathbf{P}} \widehat{\Lambda}^{-1} \widehat{\mathbf{P}}^{\mathrm{T}} \mathbf{x}_i^{\mathrm{T}}\ , \qquad 4.23$$

where $\hat{\mathbf{t}}_{\sigma i}$ represents $1 \times r$ row vector which indicates ith row of matrix $\mathbf{T}$. Meanwhile $\mathbf{x}_i$ is the $1 \times n$ row vector which represents the ith observation. They are related by the expression, $\hat{\mathbf{t}}_{\sigma i} = \mathbf{x}_i \widehat{\mathbf{P}}$.

When abnormal behaviour in the system occurs, such as the presence of a fault, the mapping relation of the data is collapsed which may fluctuate the corresponding T^2 and/or Q-statistic. The threshold of T^2 is based on F-distribution and mathematically given by [146],

$$T^2_{r,n;\alpha} = \frac{r(n-1)}{n-r} F_{r,n-r;\alpha} \ , \qquad 4.24$$

where r is retained principal components inside the PCA model, n is measured variables used to build the PCA model, $F_{r,n-r;\alpha}$ is an F-distribution, and α is the level of significance.

The new observation is normal if it satisfies the following condition,

$$T_i^2 \le T_\alpha^2 \qquad 4.25$$

When the T_i^2 of the new observation exceeds the threshold, T_α^2, it indicates a fault has occurred within the new data set. The contribution of the individual variables of the new observation to the T_i^2 can also be identified. The variables which considerably contribute to the T_i^2 are pointed to be likely the source of fault [52].

4.1.3.2 Q-statistic

The Q-statistic or SPE measures the variability of the observation data projected onto the residual space. It indicates the change of behaviour of the observation data which are not accounted for by the principal component subspace. The Q-statistic of the ith observation, $\mathbf{x}_i$, can be expressed as [147],

$$Q_i = \|\tilde{\mathbf{x}}_i\|^2 = \left\|\mathbf{x}_i\left(\mathrm{I} - \widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\mathrm{T}}\right)\right\|^2, \qquad 4.26$$

where $\tilde{\mathbf{x}}_i$ is the observation data projected onto the residual subspace.

A new experimental trial is considered normal if,

$$Q_i \leq SPE_\alpha \ , \qquad 4.27$$

where SPE_α represents the upper control limit with significance level α. The threshold for SPE is determined by its approximate distribution as given by [148],

$$SPE_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_o^2}}{\theta_1} + 1 + \frac{\theta_2 h_o (h_o - 1)}{\theta_1^2} \right]^{\frac{1}{h_o}}, \qquad 4.28$$

where,

$$\theta_i = \sum_{j=r+1}^{n} \lambda_j^i \ , \quad i = 1, 2, 3 \ , \qquad 4.29$$

and,

$$h_o = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \qquad 4.30$$

where *r* is retained principal components inside the model and c_α is the standard normal deviation with the upper 1-α percentile.

When the Q_i of the new experimental trial violates the SPE threshold, a fault is deemed to have occurred. The threshold value is defined with the assumption that the observation data is multivariate normally distributed and time-independent [147].

The *Q*-statistic has a very small value and consequently is more sensitive than the T^2-statistic. This characteristic makes the *Q*-statistic able to detect any small change in the system behaviour. On the contrary, the T^2-statistic needs substantial change in the system behaviour to be measureable [138].

4.1.3.3 Comparison between the T^2-statistic and Q-statistic

The idea of using the T^2 and Q-statistic for fault detection is illustrated graphically in Figure 4.5. A fault which is illustrated using a red circle can be detected by both T^2 and Q-statistic because it lies outside the thresholds. A T^2-statistic provides a measure of the deviation in the principal component subspace that most importantly corresponds to the normal condition. The normal range defined by the T^2 threshold is usually larger than that defined by the Q threshold. Accordingly, small faults can easily exceed the Q threshold, but not the T^2 threshold. This indicates that the T^2 -statistic usually is less sensitive than the Q-statistic for the small fault detection.

The score matrices from the principal component subspace can also be used to obtain information about particular events. The scores from desired principal components can be plotted in two or three dimensional space which may show a different pattern between a fault condition and a normal (without fault) condition. The T^2-statistic, however, has included this information since it is obtained from the scores. In addition, the Q-statistic provides supplementary information obtained from the residual subspace because it calculates the variation of the data that lay outside the model.

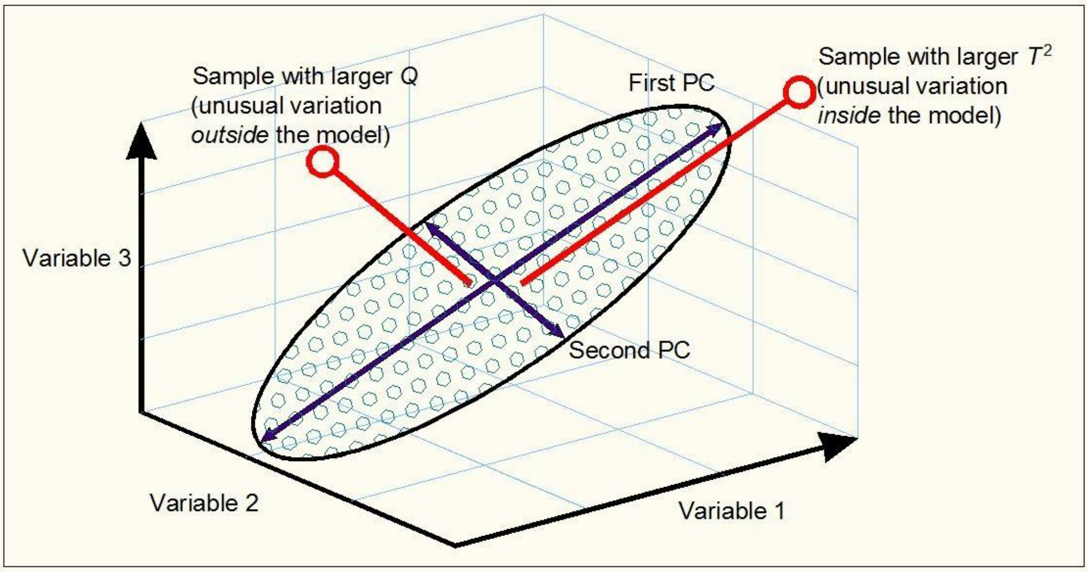


Figure 4.5 T^2 and Q-statistic in PCA model [138]

4.2 Application of PCA in Feature Extraction and Fault Diagnosis

Feature extraction from vibration data is a challenging field of research in many engineering applications, especially in the area of fault diagnosis of rotating machinery. It is one of the most extensively studied issues in condition monitoring [57]. Feature extraction is usually applied before undertaking fault diagnosis or fault classification. It aims to reduce the data set dimension and to perform a transformation, such that the critical information in the original data are extracted [84]. In this stage, pre-processing of the raw data is normally performed to obtain appropriate parameters that indicate whether the important pattern from normal machine condition is obtained.

In the recent decades, a wide range of new feature extraction techniques have been proposed. Each technique has different theoretical backgrounds and gives different results. As a consequence, the implementation of a particular extraction technique depends on the operational condition of the system and the type of faults that are required to be detected [59].

The use of a combination of feature extraction techniques seems to be more extensively studied than those using only a single feature extraction technique. The motivation in using the combined techniques is the importance of selecting the most effective and suitable techniques, thus producing more reliable diagnostic results. This motivation accelerates the development of new feature extraction techniques aimed at finding better and more suited feature extraction techniques for specific fault diagnosis.

The use of the PCA in vibration signal analysis as a feature extraction method in fault detection has been widely proposed in many literatures. An example of feature extraction techniques published within the last ten years is presented in Table 4.1. It shows that many of the PCA-based extraction techniques are combined with other methods in order to obtain better and more appropriate features for fault diagnosis.

Table 4.1 Summary of the use of feature extraction for PCA applications

References	Objects	Type of Fault	Techniques	Features	Extraction method	Results
Li et.al (2003) [149]	Gearbox	Broken tooth	Time-domain	Standard deviation, skewness, kurtosis, the max peak, absolute mean, RMS, crest factor, impulse factor, clearance factor	PCA	The proposed method is sensitive to different condition of gearbox
Malhi and Gao (2004) [150]	Ball bearing	defect on inner race, outer race, both inner and outer race	Time-domain, frequency domain, wavelet domain	Thirteen features	PCA, neural network	Smaller number of features performs better for defect classification than using all relevant features
Sun et.al. (2007) [151]	Rotor fault	Oil whirl, shaft crack unbalance, rotor radial rub, unbalance and radial rub	Time-domain, frequency-domain	Peak-peak value, peak index, waveform index, tolerance index,` impulsion index, kurtosis index, skewness index, power spectrum, and amplitude spectrum	PCA, C4.5 decision tree	A method based on combination of PCA and C4.5 decision tree shows higher accuracy than neural network approach
Shuang and Meng (2007) [152]	Rolling bearing	Faulty bearing on outer race, inner race	Time-domain	Eigenvectors	PCA, SVM	Combination of PCA-SVM-eigenvectors provides a new technique for intelligent bearing fault diagnosis
Trendafilova et.al. (2008) [153]	A scaled aircraft wing	Crack	FRF	Frequencies	Modified PCA, simple pattern recognition (PR)	The improved PCA and PR approach provides a promising technique for structural damage detection

References	Objects	Type of Fault	Techniques	Features	Extraction method	Results
He et.al. (2009) [154]	Internal-combustion engine; and automobile transmission gearbox	Worn connecting road bearing; worn bearing	Time and frequency-domain	Sixteen statistical features	PCA technique and PC representation	A low dimensional PC representation is proven to be effective for representing and classifying machine condition
Sakthivel et.al. (2010) [84]	Centrifugal pump	Impeller fault, Bearing fault, cavitation, and seal fault	Time-domain analysis	Mean, median, standard deviation, standard error, variance, skewness, kurtosis, sum, range, minimum, and maximum	PCA, C4.5 decision tree algorithm, and rough set methods	PCA based decision tree-fuzzy achieves 96.67% accuracy
Mujica et.al. (2010) [138]	Steel sheet and gas turbine blade	Crack	Frequency response function (FRF)		PCA, Hotelling T^2 statistic and Q-statistic (SPE)	T2 statistic and Q-statistic successfully detect damages in the structures
Trendafilova (2010) [127]	Roller bearings	Outer race, Inner race, and rolling element faults	Discrete wavelet transform (DWT)	Daubechies2 (db2), high frequency part	Modified PCA, simple pattern recognition (PR)	Combination of modified PCA and simple PR successfully detect and recognise bearing fault categories
Li et.al (2010) [155]	Gearbox	Single wear, Single crack, single tooth broken, combination fault of crack and tooth broken and combination fault of wear and spalling	Discrete wavelet transform (DWT)	Daubechies20 (db20) at decomposed level 5	Wavelet-AR Model and PCA	Combination of wavelet-AR-PCA is able to classify all of experimental gears operating conditions. The result with PCA performs better than that without PCA
Pirra et.al. (2011) [156]	Roll bearing	Inner race, rolling element fault	Time-domain	Absolute mean, RMS, maximum peak value	PCA based model	The result allows accurate damage recognition and reduces the number of False Alarm.

References	Objects	Type of Fault	Techniques	Features	Extraction method	Results
Zimroz and Bartkowiak (2011) [157]	Planetary gearboxes	Gears damage	Power spectral density	A number of amplitudes from spectral components correspond to fundamental frequency and its harmonics	PCA, analysis of eigenvalues	The proposed method can find earlier damage in the gearbox. The damage evolution causes increased total energy dissipation
Ahmed et.al. (2012) [52]	Recipro-cating compressor	Several combination faults of discharge valve, suction valve, intercooler leakage, and a loose drive belt	Time-domain analysis	RMS, Peak factor, histogram upper bound, histogram lower bound, variance, crest factor, entropy, absolute value, normal negative log-likelihood , shape factor, clearance factor, kurtosis, skewness, , and Weibull negative log-likelihood value	PCA, Hotelling T^2 statistic and Q-statistic (SPE)	PCA with T^2 statistic and Q-statistic method allows the detection of single and multiple faults. Q-statistic gives better detection ability than T^2 statistic
Abouhnik et.al. (2012) [158]	wind turbine blade, gearbox, induction motor	Cracks in turbine blade, gear tooth breakage, phase imbalance in induction motor current	Time-domain analysis	Kurtosis, RMS and crest factor	PCA and residual matrix	Introduce a new approach to detect faults by using combination of PCA and residual matrix. In addition, crest factor are also applied to the residual matrix to detect faults
Zhao et.al. (2012) [159]	Centrifugal slurry pump	Impeller vane leading edge damage	Frequency-domain	Amplitude at 1X, 2X, 5X, 10X, RMS between 0-1X, 0-2X, 0-5X, 0-10X	PCA, half and full spectra, fuzzy preference-based rough sets	Propose a method to generate a monotonically indicator for impeller damage propagation

References	**Objects**	**Type of Fault**	**Techniques**	**Features**	**Extraction method**	**Results**
Liying et.al. (2012) [160]	Ball bearing	defect on inner race, ball, outer race	Time-domain	Amplitude	PCA, T^2-statisctic, Q-statistic	Achieved fault detection rate higher than 92.5%
Dong and Luo (2013) [161]	Bearings	Bearing degradation	Time-domain, frequency-domain, and time-frequency domain analysis	Sixteen features from time-domain and 12 features from frequency-domain	PCA, LS-SVM	propose combined approach (PCA and LS-SVM) for the bearing degradation process prediction
Gharavian et.al. (2013) [162]	Gear box	Chipped and worn gear	Continuous wavelet transform	Morlet mother wavelet	Features extraction: FDA, PCA, and classifier: KNN and GMM	PCA-based features achieves recognition rate 84.6%
Xi et.al. (2013) [163]	Ball bearing	defect on ball, inner race, and outer race	Time-domain and time-frequency domain	Mean value, kurtosis, wavelet packet energy spectrum	PCA, back propagation neural network	Investigate the performance PCA-based feature selection technique
Sakthivel et.al (2014) [164]	Centrifugal pump	Bearing fault, impeller fault, seal fault, and cavitation	Time-domain analysis	Standard deviation, standard error, Mean, median, variance, skewness, kurtosis, range, minimum, maximum, and sum	PCA, C4.5 decision tree, Bayes Net and Naïve Bayes classifier	Combination of PCA and decision tree performs better than all other combination of dimension reduction technique-classifier
Shao et.al. (2014) [165]	Gears	Tooth root crack, pitch crack, tooth wear, and multi-fault	db4 Wavelet packet transform	Signal energy of 16 frequency bands	PCA and Kernel-PCA	Nonlinear KPCA obtains better performance than PCA for the nonlinear relationship data

Several important studies that use PCA technique for fault diagnosis of rotating machinery, as seen in Table 4.1, are explained in chronological order in the next section.

4.2.1 PCA-Based Extraction Techniques in Fault Diagnosis

The use of PCA analysis to reduce feature dimension and to obtain an effective subspace for fault diagnosis was proposed by Li et al., [149]. The features used to build the PCA model were ten statistical parameters from the time-domain, including standard deviation, skewness, kurtosis, maximum peak value, RMS, absolute mean value, crest factor, impulse factor, shape factor, and clearance factor. The proposed method was tested on an automobile gearbox under three different operating conditions, i.e., normal, cracked tooth, and broken tooth were introduced. The raw time-domain signals from three different conditions were collected and ten features from each raw signal were computed. It showed that none of the features was able to tell the difference between the tooth damage and normal (no fault) tooth condition. By re-expressing the original features into principal components (PCs) subspace, the result showed that the first three PCs (containing 85% variation of the original features) may be used as a classifier for gear fault detection and can also be used for isolating particular types of faults from another. The result also showed a good sensitivity of the method for different gearbox working conditions and may identify natural gear defects.

The application of PCA-based method for feature selection schemes of rolling bearing fault diagnosis was investigated by Malhi and Gao [150]. The study aimed to recognize the damage level of bearings without knowing previous information about the damage condition. The study proposed a new scheme for bearing fault classification using features extracted from time, frequency, and wavelet-domain. Three cases with several combinations of ball bearing faults were carried out. Initially, a set of 13 features was compiled. They included the rectified skew, RMS, kurtosis, crest factor, peak value (time-domain), amplitude of BPFO, BPFI, BSF and power in the defect frequency range (frequency-domain), wavelet & Fourier BPFO, BPFI, BSF amplitude, and power in the defect frequency range (wavelet domain). The PCA was built to reduce the input feature dimension which subsequently would be used for both supervised and unsupervised classification process. The feed forward neural network (FFNN) was then applied to verify the performances of the features in the principal components subspace. The result showed that the proposed method attained higher classification accuracy than the method without PCA. It also confirmed the suitability of the proposed method for machine health assessment.

Sun et al., [151] explored a method based on the combination of C4.5 decision tree algorithm and PCA for rotor fault diagnosis. The experiment was carried out for six types of operating condition namely normal, rotor radial rub, unbalance, shaft crack, oil whirl, and combination unbalance and radial rub. In the investigation, 7 features from the time-domain and 11 features from the frequency-domain were used as input for the PCA model. The PCA then reduced the dimensionality of the features space down into a few dimensions. The first 6 PCs which held 98% of the variation of the data was selected and the original data set were subsequently projected onto those 6 PCs. The projected data was divided into a training data set which was used for classifier training and a testing data set for testing the classifier validity. The result showed that reducing a large number of PCs did not decrease the diagnosis accuracy. Even for particular conditions, the accuracy of the proposed method with reduced PCs was higher than that without PCs reduction.

Sakthivel et al., [84] investigated the accuracy of PCA analysis based decision tree-fuzzy logic for the centrifugal pump fault diagnosis. The study aimed to monitor the health condition of centrifugal pumps. Several different fault conditions were investigated such as: impeller fault, bearing fault, cavitation, seal fault, combination of bearing and impeller fault. In the investigation, a wide set of the statistical features from the time-domain were extracted which included mean, median, standard error, variance, standard deviation, skewness, kurtosis, range, sum, minimum, and maximum. Those statistical features were then used as input for the feature selection scheme which consisted of combination of C4.5 decision tree algorithm, rough set, and PCA. The output of the scheme were the prominent features that were important for rule generation. The selected features were then used to build the fuzzy inference engine. The results showed that the accuracy of the combination of the decision tree-fuzzy method was 99.3%, the rough set-fuzzy was 97.50%, and the PCA-fuzzy was 96.67%. Even though the accuracy of PCA-fuzzy method was smallest, it provided the advantage of using a reduced number of uncorrelated features.

The application of PCA and T^2-statistic and Q-statistic fault diagnosis in the structures was explored by Mujica et.al., [138]. In the study, two kinds of structures were used to verify performance of the method: a steel sheet and a turbine blade of an aircraft engine. Two experimental approaches were performed to test the structures. For the first approach, a low frequency vibration from a shaker was applied to the structure while for the second one, a single piezoelectric (PZT) element was used as an actuator to excite high frequency vibration to the structure. In both approaches, several PZT sensors were also attached on the surface of the structure. A known vibration signal was applied to the structure and its dynamic response was recorded and analysed. The vibration data obtained from the undamaged structure (used for baseline) and from the damaged structure were projected onto principal components which subsequently produced PCA models with 20 PCs. For diagnosis purposes, the first two PCs were retained in the model and T^2-statistic and Q-statistic indices were analysed. The results showed that for the steel-sheet-low-frequency scenario, the score matrices provided sufficient features to identify damage. In some cases, the T^2-statistic was found to be more sensitive to changes of the structure while in other cases (blade-low-frequency), the Q-statistic was better.

The use of a combined PCA analysis and pattern recognition (PR) of vibration signals for roller bearing fault diagnosis was studied by Trendafilova [127]. Four types of vibration signals were considered, namely normal bearing, outer race fault, inner race fault, and rolling element fault signals. In the investigation, a wavelet transform was applied to pre-process a raw vibration signal and to extract desired high frequency components (detailed coefficient) which were appropriate for bearing fault diagnosis. The Daubechies2 (db2) wavelet was used to calculate feature vectors where only its high frequency parts (**cH**) were considered for subsequent analysis. For training purposes, one hundred signals from each fault condition were collected. The PCA was then applied to reduce the number of coefficients **cH** and to extract relevant features. Six PCs were retained in the model which provided more than 90% of the variance. The proposed method showed that even with the first two PCs, it succeeded to identify and classify the bearing fault. It was also found that the PR procedure achieved an excellent separation rate which was 94% to 96%.

The investigation of combined a PCA-based feature extraction has continued with more methods for the purpose of finding better feature representations for fault diagnosis. For instance, Li et al., [155] suggested a method for gear multi-fault diagnosis based on the wavelet-Autoregressive (AR) model and PCA. The test was performed for several gear conditions, namely normal, single wear, single broken tooth, single crack, compound fault of wear and spalling, and 6 combination faults with cracked and broken teeth. The Daubechies20 (db20) wavelet pre-processed raw vibration signals at 5 levels decomposition and the AR coefficients then extracted the prominent fault type features. To avoid redundancy and to improve fault diagnosis performance, the PCA was then applied to combine the AR coefficients into one unique parameter used as the classification criteria. It was found that the proposed wavelet-AR-PCA method could successfully classify all types of faults.

An approach in the use of statistical features for fault detection of reciprocating compressors was carried out by Ahmed et al. [52]. Several types of faults were proposed such as intercooler fault, suction valve fault, loose drive belt, and a combination of suction valve fault with intercooler fault, discharge valve fault with suction valve fault, and discharge valve fault with suction intercooler fault. In this study, fourteen statistical features were extracted from the raw vibration signals and then selected as features for the PCA model. By re-expressing the features into the PC subspace, it was shown that seven of these variables accounted for 90% of the variance. This was considered to be acceptable for the compressor fault diagnosis. The T^2-statistic and Q-statistic were subsequently developed to detect the faults. The result showed that the PCA based method succeeded to detect single and multiple faults in the compressor. The fault occurrences were detected by comparing T^2 and Q-statistic of the features with the defined threshold.

Zhao et al., [159] used a combination of half and whole spectra, fuzzy preference-based rough sets and PCA to produce the criteria that vary linearly with fault propagation for impeller fault detection in a centrifugal slurry pump. The half and full spectra were used to extract potential features associated with the pump condition. The potential features were amplitude at 1X, 2X, 5X 10X, RMS between 0-1X, 0-2X, 0-5X, and 0-10X. The fuzzy preference-based rough sets were used to select the prominent features associated with the fault propagation linearly. The PCA model was employed to re-orient the features and generate the fault propagation criteria. In the investigation, four different scenarios were tested i.e., PCA applied directly to all features, PCA applied for feature extraction and feature evaluation using fuzzy rough set, PCA for feature extraction and feature evaluation using dominance rough set, and PCA for feature extraction and feature evaluation using fuzzy preference-based rough set (proposed method). It was found that the whole spectrum was an appropriate instrument to extract features from the faulty pump. The results also showed that the proposed method successfully produced the criteria that may effectively differentiate the health status of the pump impeller.

Liying et al., [160] investigated the use of one dimensional time-domain feature extraction for bearing fault detection. The method defined the vibration signal into a new space with higher dimensions and subsequently the PCA model was built based on this new space. Three types of faults were carried out to test the proposed method, namely good bearing, faulty outer raceway, faulty inner raceway, and ball fault under different load conditions. The T^2 and Q-statistic were then applied for fault detection. The results demonstrate that the accuracy of the proposed method for bearing fault detection was above 95%.

Gharavian et al., [162] studied a comparison of the Fisher discriminant analysis (FDA)-based and PCA-based feature for automobile gearboxes fault diagnosis. In their study, a continuous wavelet transform (CWT) was applied to a vibration signal collected from several conditions of the gears and the continuous wavelet coefficient (CWC) were then evaluated for some different scales. This produced a feature vector in which the number of dimensions was the same with the number of the scales. The FDA-based and PCA-based methods were applied to reduce the dimensionality of the original features. As the classifier, the researchers examined the Gaussian mixture model (GMM) and k-Nearest Neighbor (kNN). The experimental results showed that the FDA-based method achieved a higher recognition rate than the PCA-based method, however the recognition rate of the PCA-based method was still considered high at 84.6%. Both of the classifiers gave a great performance in fault sorting for condensed fault condition.

The application of PCA to obtain the prominent features from combinations of energy spectrum and statistical features for the purpose of bearing fault diagnosis was explored by Xi et al., [163]. In the investigation, the energy spectrum from the time-frequency domain of vibration signals were selected using a wavelet packet transform from different frequency bands and the statistical features such as kurtosis and mean were extracted from the time-domain. Four types of bearing conditions were introduced to test the proposed method such as normal bearing, ball fault, inner race fault, and outer race fault. The mother wavelet Daubechies 12 (db12) was used at level 3 to calculate the wavelet packet energy spectrum. The ten features (eight energy spectrum and two statistical features) were used as input to the PCA model. Based on 85% of the cumulative percent variance, the four most prominent features were selected from the ten original features set. These features were then trained using a back propagation neural network for bearing fault classification. The results demonstrated that the PCA-based feature selection scheme was effective for bearing fault diagnosis with correct detection rate of 87.5%-100%.

4.3 Concluding Remarks

A review of the application of PCA for feature extraction and fault diagnosis discussed in this chapter has shown the trends of combining the PCA approach with other methods to obtain the most prominent features. Several important findings from the literature review are presented in this section.

- One of the most widely used applications of the PCA method is for feature extraction in order to find better feature representation for fault diagnosis by re-expressing the original space into the new lower subspace. In spite of the large number of previous research studies using PCA, the selection of input for the PCA model, which is a critical step for building a robust PCA-based fault diagnosis framework, is often arbitrary and still has further potential to be explored [149]. Hence, there is an open area in the development of methodologies for the selection of better input vectors for the PCA model.
- One of the important issues in the application of PCA in fault diagnosis is the trends that combine the PCA with other signal processing methods in order to find more effective features for fault diagnosis. Even though many PCA-based combination methods have been investigated, there are no clear standard rules which can be used as guidance for its application to a particular fault diagnosis. This means that there is no standardised methodology for selecting the combination of PCA-based framework for finding and generating reliable

features. Hence, the selection of the combination of PCA-based framework with other signal processing methods is still open for investigation.

- The use of the wavelet transform for vibration signal pre-processing prior to the application of PCA methods has been explored by several researchers [127, 150, 155]. Even though the use of the wavelet transform to pre-process signals is common, the selection of a suitable mother wavelet and decomposition level for a particular fault diagnosis is an unstandardized process. Hence, it is an area which is wide open for investigation.
- The use of combinations of simple algorithm classifiers such as the kNN with the PCA method for fault diagnosis of rotating machinery is rarely found in the literature. It is wide open for further investigation.
- The research of PCA-based fault diagnosis for the centrifugal pump health monitoring is still rare. Particularly in the case of applications that combine the wavelet transform and statistical parameters as input features to the PCA model for fault diagnosis.

The next chapter describes the proposed wavelet-based PCA method for fault diagnosis. It presents the statistical parameters extracted from decomposed time signals which used to build PCA model. The chapter also explains the proposed fault diagnosis algorithm.

CHAPTER FIVE

5 Principal Component Analysis and Wavelet-Based Framework for Fault Diagnosis

In this study a new fault diagnosis framework for a centrifugal pump is proposed. The technique is based on Principal Component Analysis (PCA) combined with the wavelet-based feature extraction.

The proposed method combines the use of statistical parameters obtained from wavelet decomposition using multi resolution analysis (MRA) and the PCA to develop a fault diagnosis framework for a centrifugal pump. The application of PCA combined with several other signal processing techniques, which is extensively discussed in previous chapters, motivates this research for further development of PCA-based techniques for fault diagnosis. In this research, six statistical features namely energy level, standard deviation, RMS, kurtosis, variance, and crest factor were extracted from the wavelet transform of the vibration signals using the MRA technique. The statistical parameters obtained were used as input vectors to build the PCA model.

The use of the Symlet wavelet family in the feature extraction process in this research was based on several findings in the literature where the Symlet wavelet family was effective in the application of fault diagnosis of centrifugal pumps [124]. These findings led to the use of the three Symlet wavelets, i.e., symlet4 (sym4), symlet8 (sym8), and symlet12 (sym12).

Several applications of the T^2-statistic and the Q-statistic based on the PCA-model, as presented in Chapter 4 (Table 4.1), motivated further investigation. The research extended the use of the T^2-statistic and the Q-statistic in developing improved fault detection schemes for a centrifugal pump.

In this research the application of scores of principal components (PCs) to identify fault location was extended. The results from several researchers motivated further investigation of the technique to identify faults in a centrifugal pump.

Examples of the use of scores of PCs for fault identification of sensor measurements from laboratory wastewater treatment process may be found in Tao et al., [140], Gharavian et al., [162] for fault diagnosis of automobile gearboxes, Widodo and Yang [166] for induction motor fault diagnosis using transient current signal, and Sakthivel et al., [164] for centrifugal pump fault diagnosis using vibration signals.

The supervised machine learning, i.e., k-Nearest Neighbors (kNN) based on scores of PCs was used in this research for fault classification and identification. kNN was investigated since it was considered as a simple algorithm with the high level accuracy [167]. In addition, it was chosen because it has been widely used in many applications in the fault classification and identification area. Several examples can be found in Pandya et al., [168] for bearing fault diagnosis, Lei and Zuo [83] for gear crack level identification, and Bouguerne et al., [169] for classification of induction machine faults.

This research aims to contribute additional references related to the use of the Symlet wavelet family for centrifugal pump fault diagnosis. This study can be used to add information to the selection of the most suitable Symlet type for feature extraction in fault diagnosis.

Furthermore, this study aims to investigate the application of statistical parameters calculated from the wavelet decomposition of the raw vibration signal and used as an input vector for building the PCA model. The use of combinations of statistical parameters, wavelet transforms and PCA in the context of fault diagnosis have been proposed in many literatures [127, 155] and this research aims to explore different schemes of the combination of statistical parameters, wavelet transforms, and PCA.

The proposed method also intends to expand the use of T^2-statistic and the Q-statistic for fault detection and scores of PCs combined with kNN for fault classification and diagnosis of the centrifugal pump.

The feature extraction method (combination of statistical parameters and wavelets), PCA model, fault detection scheme (T^2-statistic and the Q-statistic), and fault classifier (scores of PCs combined with kNN) form an integrated framework for fault diagnosis of the centrifugal pump. The details of the proposed method are discussed in the following section.

5.1 The Proposed Integrated Framework for the Centrifugal Pump Fault Diagnosis

The proposed method commences with the feature extraction stage which utilises the Symlet wavelet to pre-process the raw time-domain vibration signals collected from the centrifugal pump test rig. The multi-resolution analysis (MRA) of the wavelet transform was used to decompose the time-domain vibration signals at up to 5 levels.

The MRA of the discrete wavelet transform (DWT) decomposes the time-domain vibration signals which results in Approximation parts (A) and Detailed parts (D). In this research, the Symlet wavelet family was used. There were three types of Symlet (sym-n) wavelet investigated, namely Symlet4 (sym4), Symlet8 (sym8), and Symlet12 (sym12). Figure 5.1 to Figure 5.4 depicts a schematic diagram of the algorithm of the proposed method.

The first stage of the proposed method is the feature extraction framework as shown in Figure 5.1. The algorithm was initiated by collecting the raw time-domain vibration signal through the data acquisition process from the centrifugal pump test rig. The collected raw time-domain vibration signal was then decomposed using Symlet (sym-n) wavelet transform. The decomposition process was up to 5 levels using MRA. The decomposition process yielded two parts, namely Approximation coefficients (cA) and Detailed coefficients (cD). In this research, only the cA parts were selected for the next process [170]. Hence, the features were extracted only from the cA parts of each sym-n transform.

The six features i.e., Energy level, Standard Deviation, RMS, Kurtosis, Variance, and Crest Factor were then extracted from the cA parts of each sym-n wavelet transform (up to 5 levels). Therefore, there were 30 features (6 features x 5 levels) obtained from the wavelet transform process for each of the sym4, sym8, and sym12 wavelets.

The resulting 30 features for each sym-n transform were stored in matrix form. Hence, there were three feature matrices i.e., one each for sym4, sym8, and sym12. All of the feature matrices were normalized using a mean $\bar{x}$, and a standard deviation σ. The scale parameter vector $\bar{x}$ and σ were obtained, and were then stored into files for later use in the next stage.

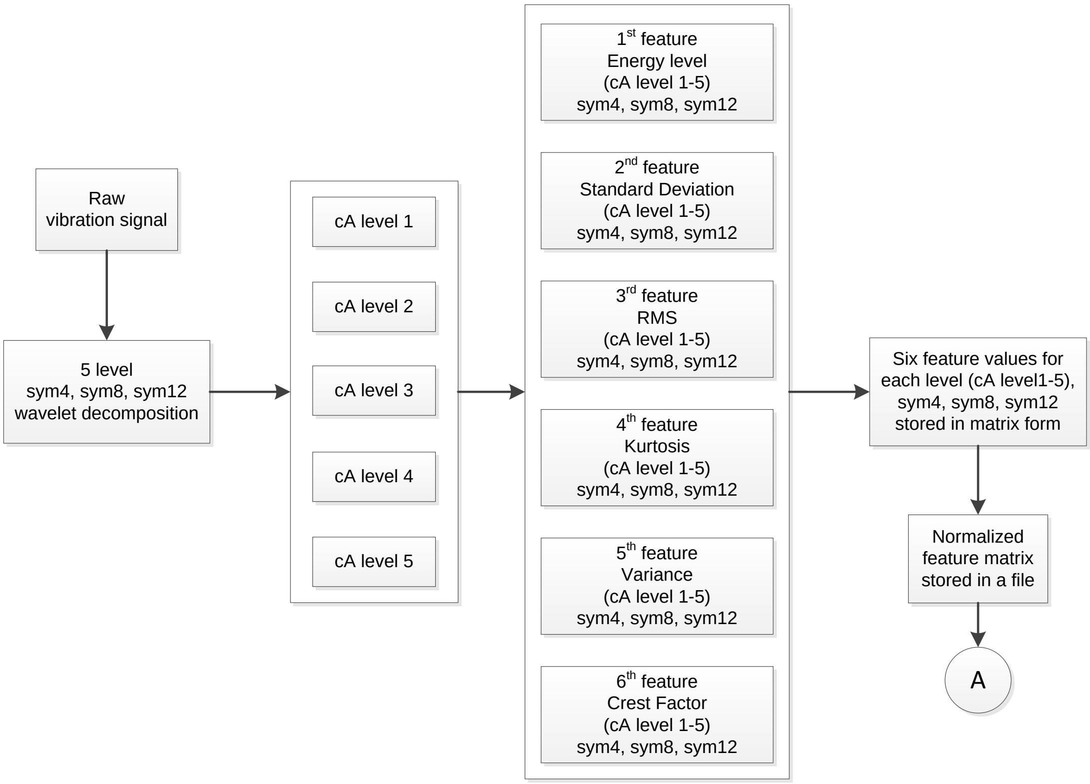


Figure 5.1 Stage 1 of the proposed integrated framework (feature extraction)

Figure 5.2 shows the second stage of the proposed method; the PCA modelling stage. In this stage, the feature matrices from normal (without fault) time-domain vibration signals obtained from the first stage were used to calculate the loading **P** and eigenvalue **Λ** matrices. There were three **P** and three **Λ** matrices obtained for each of the sym4, sym8, and sym12 wavelets decompositions. The next step was to determine the number of PCs to be retained in the PCA model. The cumulative percent variance (CPV) was used to select the number of the PCs based on the first *r* largest eigenvalues. The PCA model was then built based on the first *r* of the PCs. Three PCA models were built, each for sym4, sym8, and sym12 wavelets. All of the PCA models (i.e., reduced loading **P** and **Λ** matrices) were then stored into files for later use in subsequent stages. The last step of the stage was the calculation of upper limits i.e., thresholds for fault detection. This was carried out for T_α^2 and SPE_α which was calculated by means of an F-distribution and approximate distributions respectively as mentioned in Chapter 4.

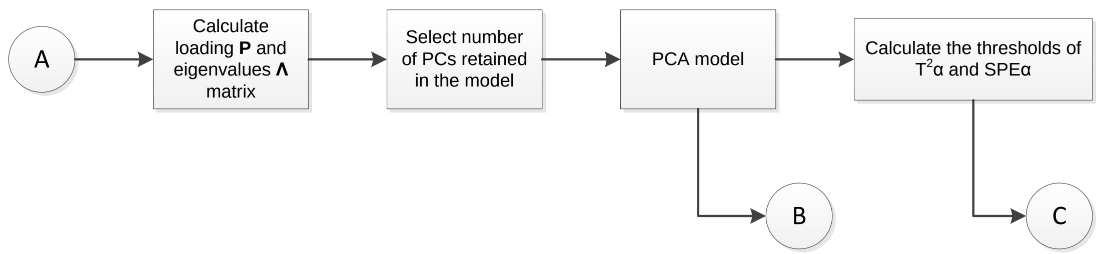


Figure 5.2 Stage 2 of the proposed integrated framework (PCA modelling)

The third stage was the fault detection stage as depicted in Figure 5.3. The process was commenced by calculating the $T^2$and Q-statistics of the feature matrix generated from the faulty time-domain vibration signals (testing data). The testing data was pre-processed in stage 1 prior to being processed in this stage. The calculation of the T^2 and Q-statistics was based on equation 4.23 and 4.26 respectively using the reduced loading **P** and **Λ** matrices obtained from stage 2. This step was carried out for each of the feature matrices calculated using sym4, sym8, and sym12 wavelets. The T^2 and Q-statistics obtained were then compared with the T_α^2 and SPE_α calculated from stage 2. If either $T^2 > T_\alpha^2$ or $Q > SPE_\alpha$ then a fault was detected.

The last stage of the proposed method was fault identification and classification as shown in Figure 5.4. This stage depicts the process to classify and identify the fault based on the scores on the PCs. In this stage, prior to calculating the scores on PCs, the testing data (might include normal and faulty vibration signals) was pre-processed in stage 1. The process was then continued by projecting the testing data into the PCA model. This process produced scores of PCs associated with the normal and faulty vibration signals.

For the purpose of fault classification, the scores were then plotted either in two dimensions (PC1 versus PC2) or three dimensions (PC1 versus PC2 versus PC3). The results were used to analyse the accuracy of scores of PCs to separate fault types. The fault identification was performed based on the use of the k-Nearest Neighbors (kNN) scheme. Several combinations for *k* and number of PCs included in the scheme were analysed for each testing data. The results were used to determine the optimal number of *k* and PCs included in the scheme to identify the fault. The details of the kNN scheme were discussed in section 5.8.

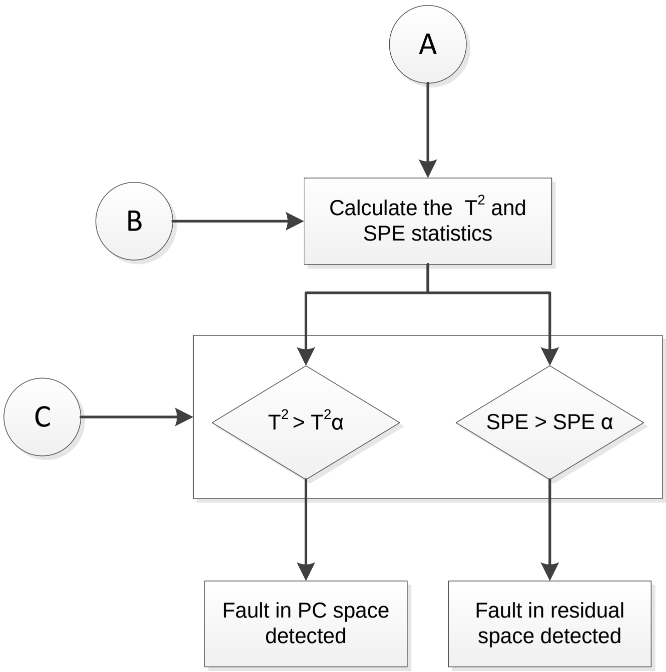


Figure 5.3 Stage 3, fault detection process

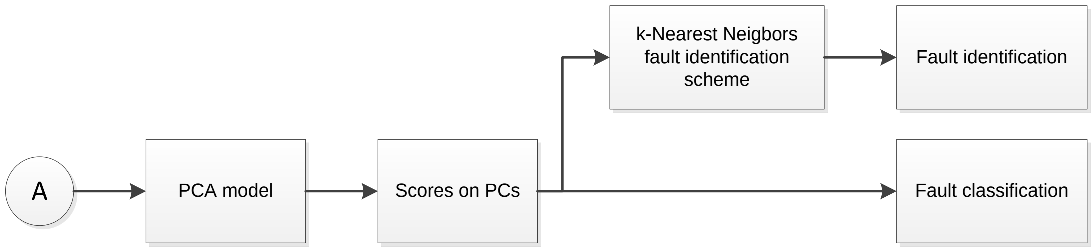


Figure 5.4 Stage 4, fault identification and classification process

5.2 kNN Classifier

A kNN is a simple non-parametric method commonly implemented for classification processes. Even though it has a simple algorithm, it performs very well and has a high level of accuracy [116]. The kNN classifier requires a metric distance *d* and a positive integer of *k* neighbors for the classification process. The principle of the kNN method is to place new observations in the class that belongs to the majority of its *k* nearest neighbors. For the large number of training data set, the kNN method proves to be very effective which provides a low misclassification error [168]. The classification accuracy mainly depends on *k* and the type of metric distance *d* used to calculate nearest distance.

In the kNN method, the classification is based on the number of training samples categorized nearby the new observations. For instance, the new observation represented by the red circle has to be classified either to the class of green triangles or to the class of blue squares.

Figure 5.5 shows that if the number of neighbors is 3, $k = 3$, which is represented by solid line perimeter then the new observation is classified into class of green triangles, since there are more number of green triangles compared to the number of blue squares inside the solid line perimeter. If $k = 5$ then inside the dashed line perimeter there are 3 blue squares and 2 green triangles, therefore the new observation is classified into the class of blue squares.

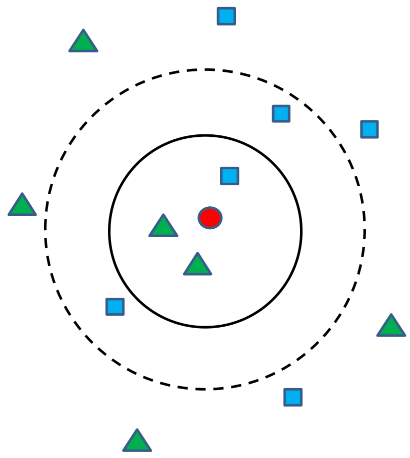


Figure 5.5 Example of kNN classification

The kNN method depends on the value of *k* and in general, the algorithm of kNN may be described in the following steps [171],

- Choose the *k* value,
- Calculate the metric distance. There are many distance calculation methods that may be used for this step. The popular distance measurements include Euclidean and Mahalanobis distance,
- Sort the distance calculation results in ascending order,
- Identify the *k* class values,
- Find the dominant class. In this step, a new observation is classified into the class having the most number of members among its *k* nearest neighbors.

The accuracy of the kNN method must always be close to 100% if the test set is a subset of the training samples since the position of training samples and their class are constant during the classification process [116].

The Euclidean metric distance is the most commonly used and easy to be implemented method for computing the distance in a multidimensional input space. The Euclidean distance between any two points in a space is defined as the length of a line between those points. In the Cartesian coordinate system, the Euclidean distance between point *a* and *b* is mathematically formulated by,

$$d_E = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \qquad 5.1$$

where a_i and b_i are two points in Euclidean *n*-space.

5.3 The Wavelet Multi-Resolution Analysis

In this research, the original time-domain vibration signal was decomposed by filtering it into a low and a high frequency part using discrete wavelet transform multi-resolution analysis (MRA). This technique has been studied previously by several researchers such as Latuny [59], Trendafilova [127], Li et al., [155], and Widodo [166]. However, in the proposed method it was modified by using the approximate parts (A) i.e, low frequency parts for up to five decomposition levels. The reason for using up to five decomposition levels was to ensure it provided a sufficient input for the purpose of PCA model building and to ensure that no essential information was lost due to inadequate data.

The basic idea of decomposing a signal into low and high frequency parts using wavelet MRA is represented in Figure 5.6. The diagram shows five levels of the decomposition process resulting in five approximate coefficient (cA) parts and five detailed coefficient (cD) parts. In this research, the cA parts were used to generate six features which were then used as inputs to build the PCA model.

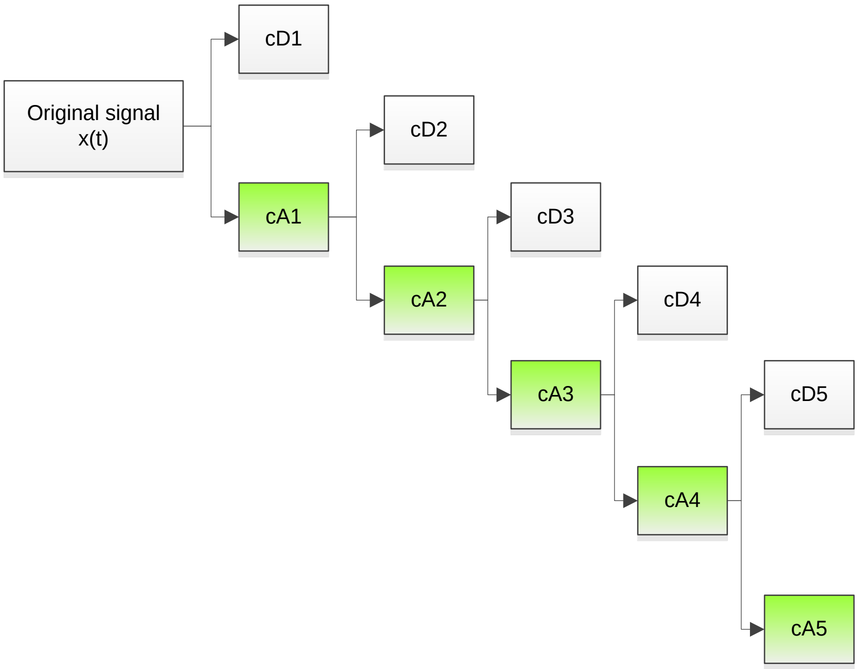


Figure 5.6 Wavelet MRA decomposition up to 5 levels

Figure 5.7 depicts the separation of the frequency band for up to five levels of the original signal using wavelet MRA.

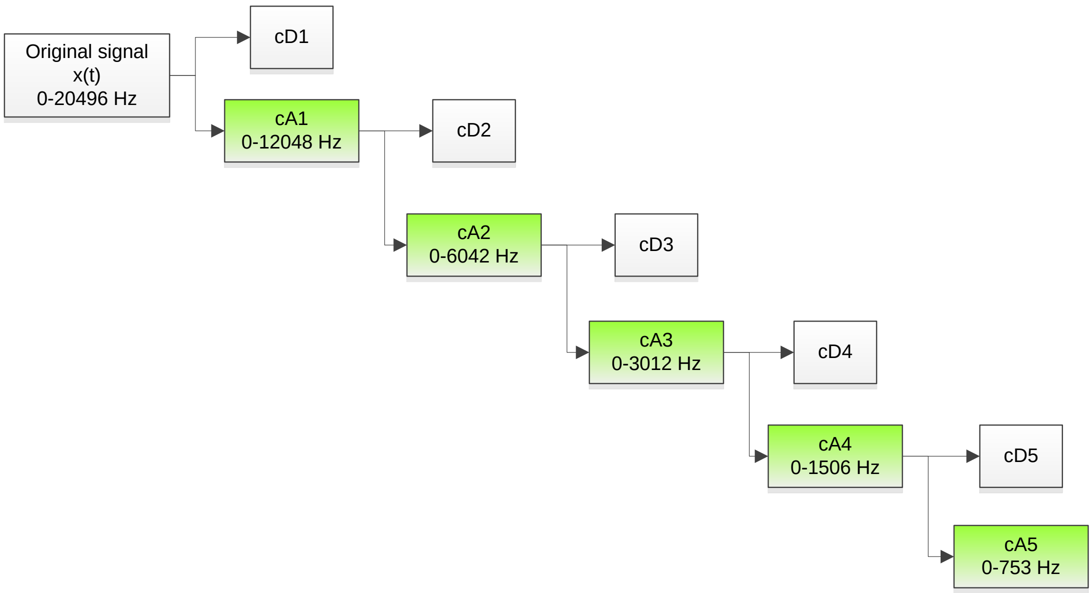


Figure 5.7 Separation of frequency band up to 5 levels

The calculation of the discrete wavelet transform was conducted using MATLAB's Wavelet Toolbox. The built-in MATLAB's function *dwt* was used to perform the single-level one-dimensional wavelet decomposition where the multi-level decompositions were carried out through the iteration process within the scripts.

5.4 Proposed Feature Extraction

There were six features used to extract essential information from each of the cA parts obtained from the decomposition process as illustrated in Figure 5.8. The six features represented the characteristics of the vibration signal and were used to build the PCA model. Each of the six features was calculated from each of the cA parts, therefore there were 30 features available to be used as inputs to the PCA modelling process. The annotation for the features were carried out as follow, E1-E5 for energy level calculated from cA1-cA5 parts respectively, and similarly S1-S5 (for standard deviation), R1-R5 (for RMS), K1-K5 (for kurtosis), V1-V5 (for variance), and C1-C5 (for crest factor). The features were then sorted as features of numbers 1 through 30 (feat1-feat30).

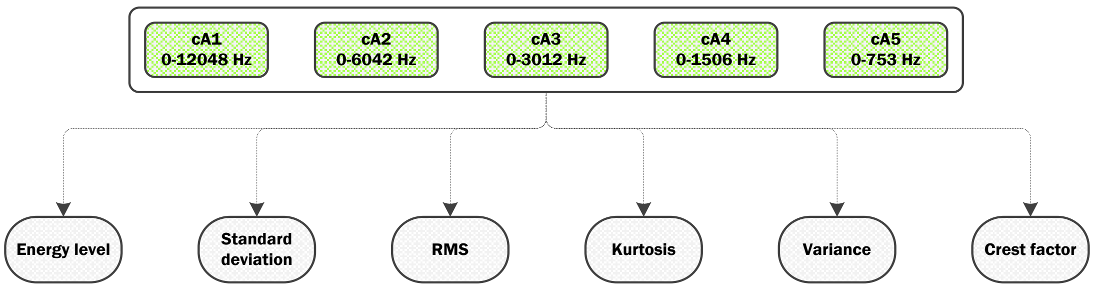


Figure 5.8 The six features obtained from each cA parts

5.4.1 Energy Level (1st feature)

The energy levels were calculated for each of the cA parts of the wavelet decomposition results using the formula suggested by Latuny and Entwistle [170],

$$E(A_i) = \left(\tfrac{1}{2}\right)^{p-1} \sum (A_i)^2, \tag{5.2}$$

where p represented the next power of two of data length, i was the decomposition level (i = 1,2, …, n), and A_i was the approximate coefficient (cA) result of the wavelet transform at the i^{th} level.

5.4.2 Standard Deviation (2nd feature)

Standard deviation was calculated for each of the cA parts using the MATLAB's built-in function and the formula was defined as,

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N - 1}} \ , \tag{5.3}$$

where x_i was an element of signal x, $\bar{x}$ was the mean of x and N was the number of data points.

5.4.3 RMS (3rd feature)

The root mean square (RMS) of each cA level was calculated using Equation 5.4 as follow,

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2} \ , \tag{5.4}$$

where N was the number of data points, $\bar{x}$ was the mean value of data set, and x_i was the element of the data set.

5.4.4 Kurtosis (4th feature)

Kurtosis value of each of the cA parts was calculated using the built-in function in MATLAB and mathematically was given as,

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\sigma^4}\,, \qquad 5.5$$

where N was the number of data points, x_i was an element of data set, $\bar{x}$ was the mean value of the data set, and σ was the standard deviation.

5.4.5 Variance (5th feature)

Variance was defined as in Equation 5.6 and calculated for each of the cA level using the MATLAB's built-in function,

$$Variance = \frac{\sum(x_i - \bar{x})^2}{N-1}\,, \qquad 5.6$$

where x_i was the i^{th} element of the data set, $\bar{x}$ was the mean of data set, and N was the number of points in the data set.

5.4.6 Crest Factor (6th feature)

Crest factor of each cA parts was calculated using the following formula,

$$Crest\ Factor = \frac{x_{max}}{x_{rms}}\,, \qquad 5.7$$

where x_{max} and x_{rms} were the absolute maximum value and RMS value of the data set respectively.

5.5 Example of Numerical Calculation of the Features

The example of decomposition results for the normalized vibration signal of a normal condition (no fault) and an impeller fault up to level 5 using a symlet8 (sym8) are depicted in Figure 5.9 and Figure 5.10. The first 1377 samples were plotted corresponding to 1x shaft rotation. In this example, the data set used was #ch123_0412_67 and #ch123_0412_18 for the normal condition and impeller fault respectively. The shaft rotation speed was 35 Hz (2100 rpm) and the vibration signal was taken from the accelerometer mounted on pump’s inlet with 48192 Hz sampling rate.

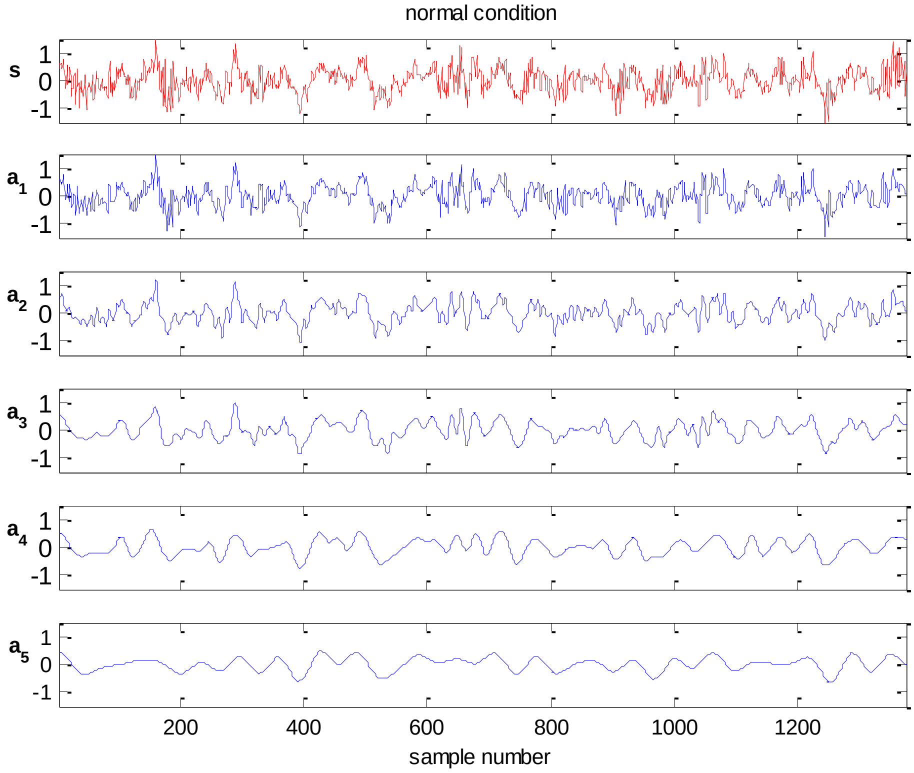


Figure 5.9 Wavelet decomposition results for normal condition, s (original signal), a_1 – a_5 (approximate parts level 1- 5)

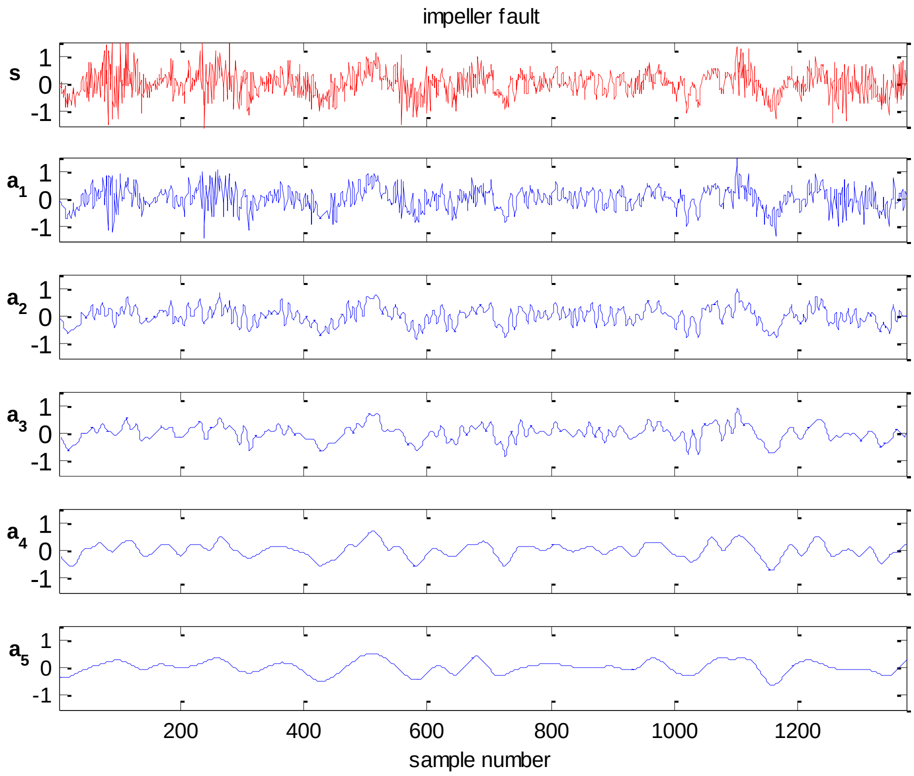


Figure 5.10 Wavelet decomposition results for impeller fault, s (original signal), a_1 – a_5 (approximate parts level 1- 5)

The visualisation of the numerical values of features calculated from the decomposition results of level 1-5 for the normal condition and impeller fault is shown in Figure 5.11 and Figure 5.12 respectively. It is obvious that all features give different values between the normal and the impeller fault. It is important to note that the results coming out from Standard Deviation (StdDev) and RMS looks very similar. It seems redundant to include both of them in the model. However, even though the redundancy will be eliminated in the PCA modelling, they are both deliberately kept in the analysis at this stage.

In this research, the normal condition (no faults) of a centrifugal pump and seven types of faults were investigated. There were four types of single faults i.e., cavitation (fault1), impeller fault (fault2), bearing fault (fault3) and blockage condition (fault4) and three types of multi-faults namely, impeller fault with cavitation (fault5), impeller fault with blockage condition (fault6), and bearing fault with cavitation (fault7).

Figure 5.13 and Figure 5.14 shows the numerical value of the six features calculated from the wavelet decomposition result of level 1 to 5 for the normal condition and for all types of fault scenario. Since there were 6 features for each decomposition level (level 1 to 5), there are 30 features in total. Feature number 1 to 30 (feat1-feat30) corresponds to E1-E5, S1-S5, R1-R5, K1-K5, V1-V5, and C1-C5.

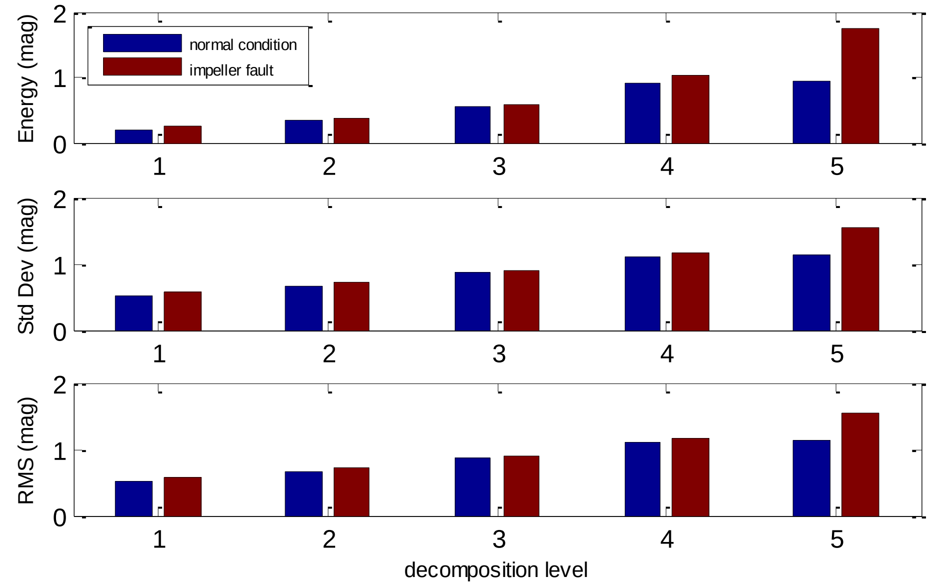


Figure 5.11 Plot of example feature value for normal condition and impeller fault (part 1)

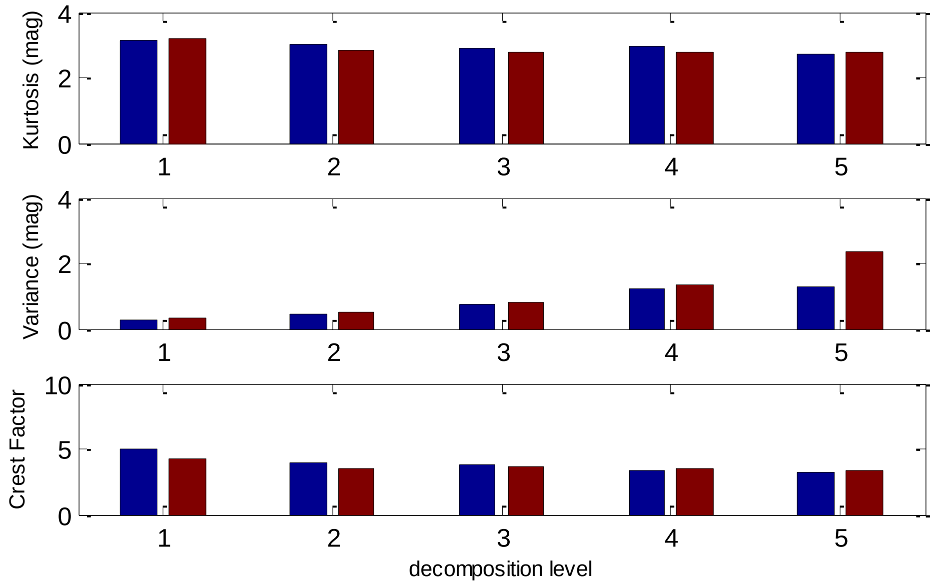


Figure 5.12 Plot of example feature value for normal condition and impeller fault (part 2)

The numerical value of the features for normal and faulty conditions as depicted in Figure 5.13 and Figure 5.14 shows the characteristics of the features for each case. It can be inferred that there is no specific pattern from the feature's characteristics which can be used to identify any fault. For example, the pattern of feature's value for normal conditions, blockage and cavitation are quite similar. Hence it is difficult to decide whether the blockage or cavitation is present or not by just analyzing the features.

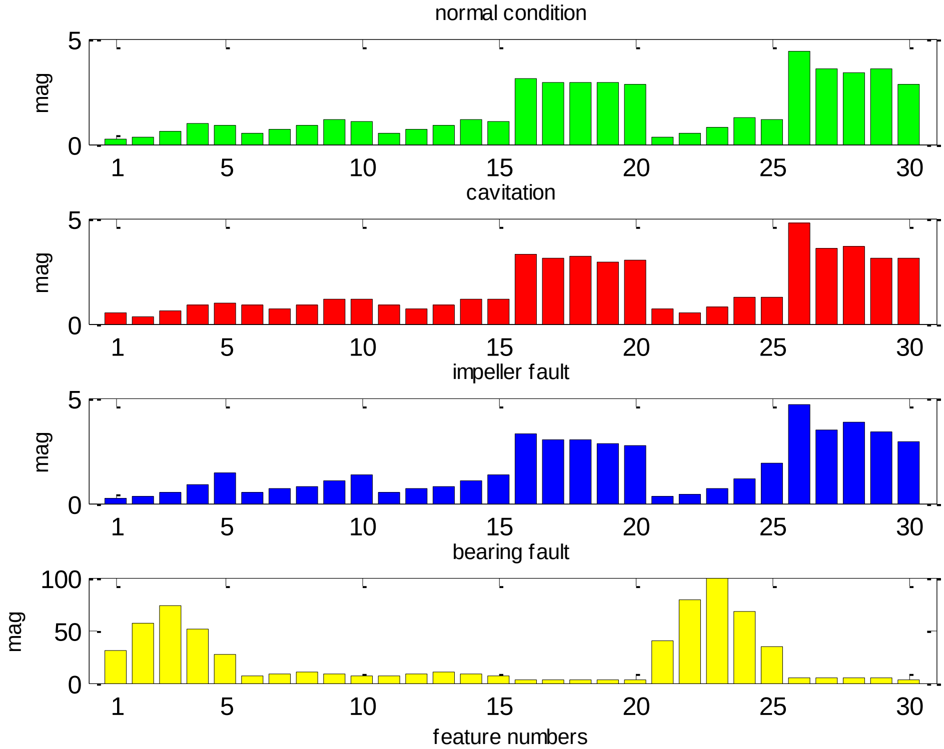


Figure 5.13 Numerical values of feature number 1 to 30 for normal condition, single-fault, and multi-faults (part 1)

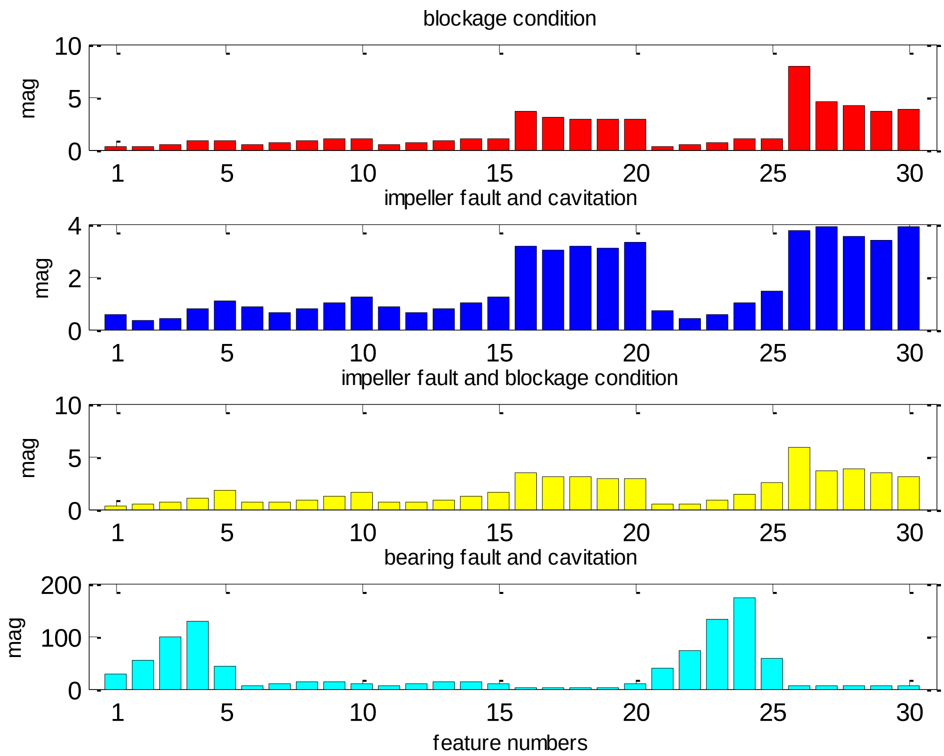


Figure 5.14 Numerical values of feature number 1 to 30 for normal condition, single-fault, and multi-faults (part 2)

The visualisation of the feature values of the normal and faulty condition also indicates that the six features generated from cA parts of level 1 to 5 are not sufficient to distinguish the unique characteristic of each case.

In this research, the PCA model was proposed to extract the most prominent features obtained from the cA parts of level 1 to 5. The scores matrix which was the projection of original data into principal components subspace was used to classify types of faults while the T^2 and Q-statistic were calculated to detect the faults. The k-Nearest Neighbors (kNN) scheme was constructed based on the scores on the principal components subspace and then used to learn and obtain conclusions regarding the fault identification.

The proposed integrated framework for fault diagnosis of centrifugal pumps is discussed in the following section. There are two main parts of the algorithm. The first part (Part A) is depicted in Figure 5.15, and the second part (part B) is depicted in Figure 5.17.

5.6 Integrated Framework for Fault Diagnosis Algorithm

In general, the proposed method has three major steps namely the detection step, classification step, and identification step. In the detection step, the method employs T^2 and Q-statistic to detect whether fault condition occurs or not; the classification step provides a separation among normal and fault conditions by plotting PCs; and the identification step identifies the type of faults by applying kNN rule. The algorithm of proposed method consists of two parts, part A and part B which are shown respectively in Figure 5.15 and Figure 5.17. In this section, part A of the algorithm will be discussed in detail.

The process was commenced by inputting the type of Symlet family (sym-n) used to decompose the vibration signal. There were three types of Symlet investigated in this study (sym4, sym8, and sym12). The number of decomposition levels was then chosen as 5 as required. The process continued with choosing the testing data code which corresponded to the file name of the stored raw vibration signal collected from the test rig. Error checking was carried out in this step to ensure the data code entered corresponded to the existing data file. This sequence is depicted in Figure 5.15.

The step proceeded with checking if the PCA model exists in the workspace. In the case the PCA model did not exist in the workspace, the routine then required the PCA model to be built prior to continuing.

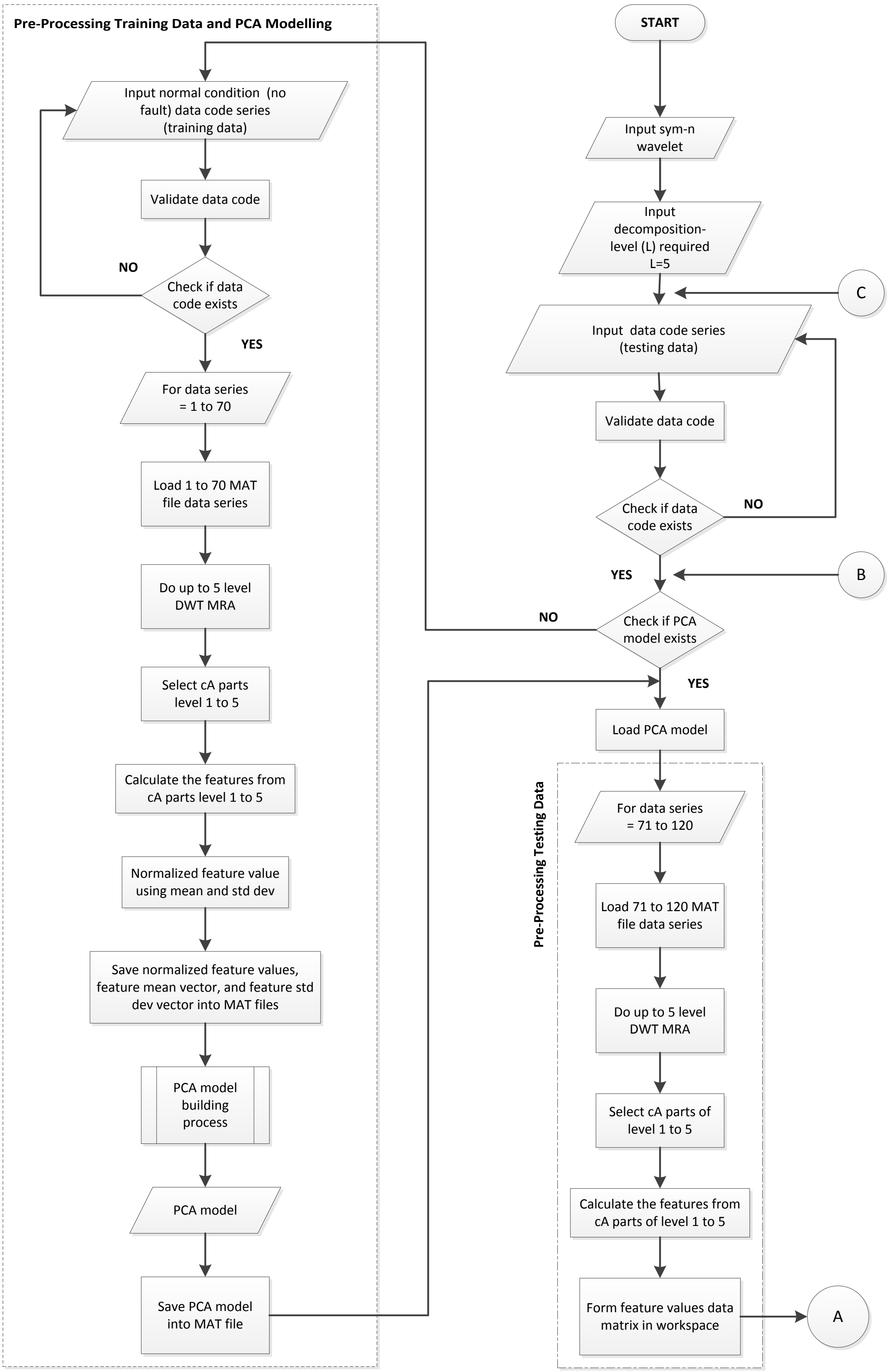


Figure 5.15 Integrated framework algorithm (Part A)

5.6.1 Pre-Processing Training Data for PCA Modelling

The cycle of this step is shown in Figure 5.15 in the pre-processing training data and PCA modelling module. The process of PCA model building started by selecting the training data code series corresponding to the files from the normal condition (no fault) raw vibration signal. The normal condition was chosen to build the PCA model since it would become a baseline for further fault diagnosis. A routine check was also performed in this step to validate the inputted data code. An error message would appear in case the data code entered did not match with the existing stored data file. The process continued by loading each single mat data file for each of the 70 mat data files. The data set used to build the PCA model was from series 1 to 70.

The script program loaded each single mat data file and processed the raw vibration signals into MATLAB's workspace environment. This was carried out for all 70 mat data files. In this loop cycle, each of the loaded data files series was transformed using the sym-n wavelet transform with up to 5 levels of decomposition. The MATLAB's built-in function, '*dwt*' was used to perform this transformation. Since the *dwt* function was a single level decomposition, the decomposition process was carried out in five cycles (cycle 1 to 5). The result produced the cA and cD parts vectors from level 1 to 5 in the workspace for all of the vibration signals from series 1 to 70. However, in this research, only the cA parts were considered for further processing.

The algorithm proceeded with calculating the six features from each cA part of the wavelet transform. For each single mat data file, the calculation of the six features produced a feature vector of 1 row × 30 columns. There were 30 features obtained from this step (6 features × 5 levels), therefore 30 columns were needed to store the feature values. These 30 features were sorted in the sequence of decomposition level and feature number. That is, feature number 1 (feat1) to 5 (feat5) represented energy magnitude calculated from cA level 1 to 5, feat 6 to feat10 represented standard deviation magnitude calculated from cA level 1 to 5, feat11 to feat15 represented RMS magnitude calculated from cA level 1 to 5, and this pattern continued for the rest of the features.

After 70 cycles of processing the single mat data files, extracting and calculating the six features, the algorithm produced a feature matrix of 70 rows × 30 columns, where 70 rows represented the number of data file series (1 to 70) and 30 columns represented the number of features obtained from cA parts 1 to 5.

The process continued with normalizing the feature matrix using the mean and standard deviation. The normalization process produced a feature matrix which had a mean of zero and variance of 1 for each feature (column). This step also created the mean feature vector of 1 row × 30 columns which contained the mean value for each column, and standard deviation feature vector of 1 row × 30 which contained the standard deviation value for each column. The normalized feature matrix, feature mean vector, and standard deviation vector were then saved as a mat file into the hard drive for later use.

The PCA model building process shown in Figure 5.15 with the block symbol ▭ is explained in the following section.

5.6.2 PCA Modelling Process

The routine was commenced by loading the normalized feature matrix mat file into MATLAB's workspace. For the purpose of PCA modelling, the normal condition (no fault) normalized feature matrix was used. The built-in MATLAB function for Singular Value Decomposition (SVD), '*princomp*', was then applied to calculate the loading matrix **P** and eigenvalue vectors. This step produced a loading matrix **P** of 30 rows × 30 columns and eigenvalues vector of 1 row × 30 columns. The eigenvalues were sorted in descending order and transformed into a diagonal matrix of 30 × 30 with eigenvalues on the main diagonal. This matrix represents the eigenvalue matrix **Λ**. The process flow of the PCA modelling approach is depicted in Figure 5.16.

In order to determine the number of principal components (PCs) retained in the model, the algorithm calculated an index which would be used as a criteria to select the most prominent PCs. The cumulative percent variance (CPV) as an index criteria was then calculated and the total variance required in the model was entered to the program.

The process continued with choosing the number of eigenvalues retained in the model based on the total variance required. For instance, if the total variance required in the model was 95%, then the algorithm would select the first r eigenvalues based on the CPV index of 0.95.

The next process was to reduce the original loading matrix $\mathbf{P}$ and eigenvalue matrix $\mathbf{\Lambda}$. In this step, the algorithm truncated the column of matrix $\mathbf{P}$ into r columns. This meant that the original size of matrix $\mathbf{P}$ (30 rows × 30 columns), was reduced into 30 rows × r columns, where $r < 30$. The retained r columns represented the retained principal components (PCs) in the PCA model.

In a similar way, the algorithm also truncated the eigenvalue matrix $\mathbf{\Lambda}$. In this case, the original matrix $\mathbf{\Lambda}$ of size 30 rows × 30 columns was reduced to r rows × r columns instead of 30 rows × r columns as previous. The matrix $\mathbf{\Lambda}$ had to be a square matrix since it was a diagonal matrix; therefore the reduced matrix had to have the same number of rows and columns. The process ended after the reduced matrix $\widehat{\boldsymbol{P}}$ and reduced matrix $\widehat{\mathbf{\Lambda}}$ (PCA model) were constructed.

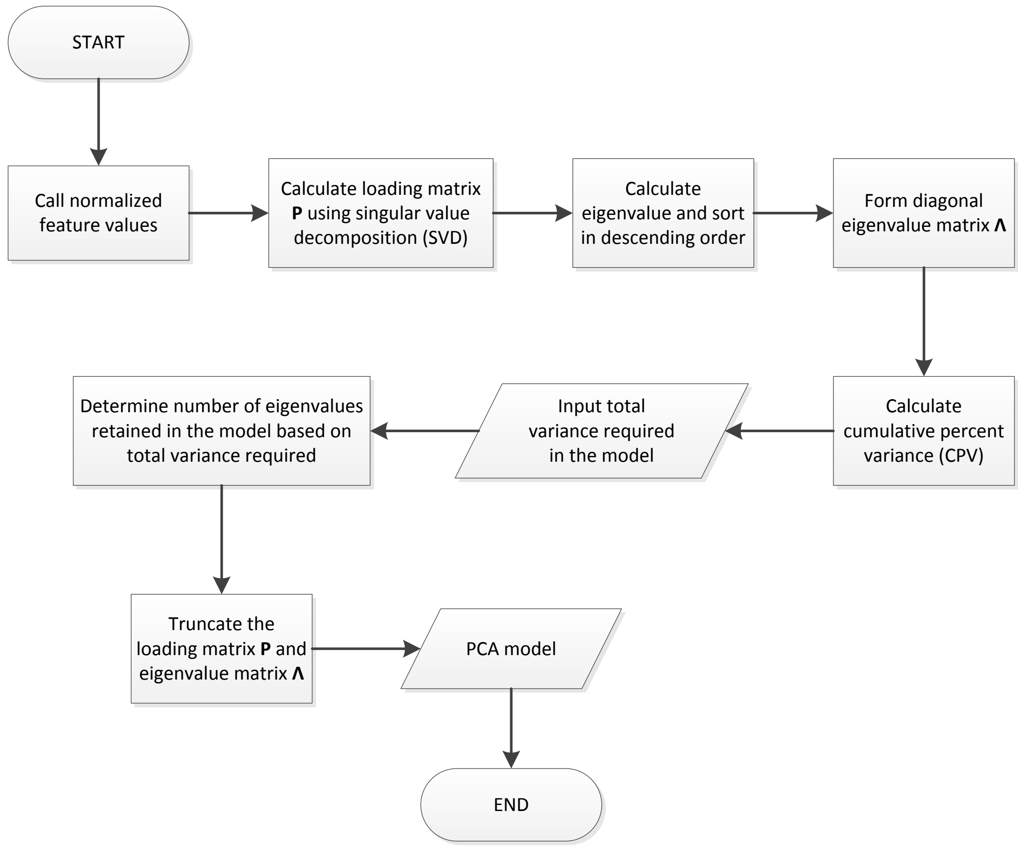


Figure 5.16 PCA model building process

5.6.3 Pre-Processing Testing Data

The routine of this step is depicted in Figure 5.15 in the module of pre-processing testing data. Prior to pre-processing the testing data, the algorithm loaded the PCA model into MATLAB's workspace. The process then continued by loading each single mat data from series 71 to 120, hence there were 50 data files/segments used for testing data.

The procedure of pre-processing the testing data was similar with that of pre-processing the training data. After loading each single mat data, the algorithm decomposed each loaded data set using the sym-n wavelet transform up to 5 levels and then selected only the cA part vectors from levels 1 to 5. The process continued with extracting the six features from each of the cA parts of the wavelet transforms. This produced a feature vector of 1 row × 30 columns for each single mat data file, therefore a feature matrix of 50 rows × 30 columns was needed to store the results for the 50 test data files. The resulting feature matrix from the testing data was then saved into the hard drive for further processing.

The next step of the algorithm was part B which was the fault diagnosis framework. The detail explanation of part B is discussed in the following section.

5.7 Fault Diagnosis Algorithm

Part B of integrated framework is the fault diagnosis algorithm. The flowchart of the algorithm is shown in Figure 5.17.

The fault diagnosis process continued from part A. The process was initiated by loading the feature mean vector and feature standard deviation vector obtained from the training data set. These two vectors were used to normalize the feature matrix of the testing data.

The process continued with checking if the reduced loading matrix $\widehat{\boldsymbol{P}}$ and reduced eigenvalue matrix $\widehat{\boldsymbol{\Lambda}}$ existed in the MATLAB's workspace. In case the matrices were not found in the workspace, the algorithm resumed by returning to part A.

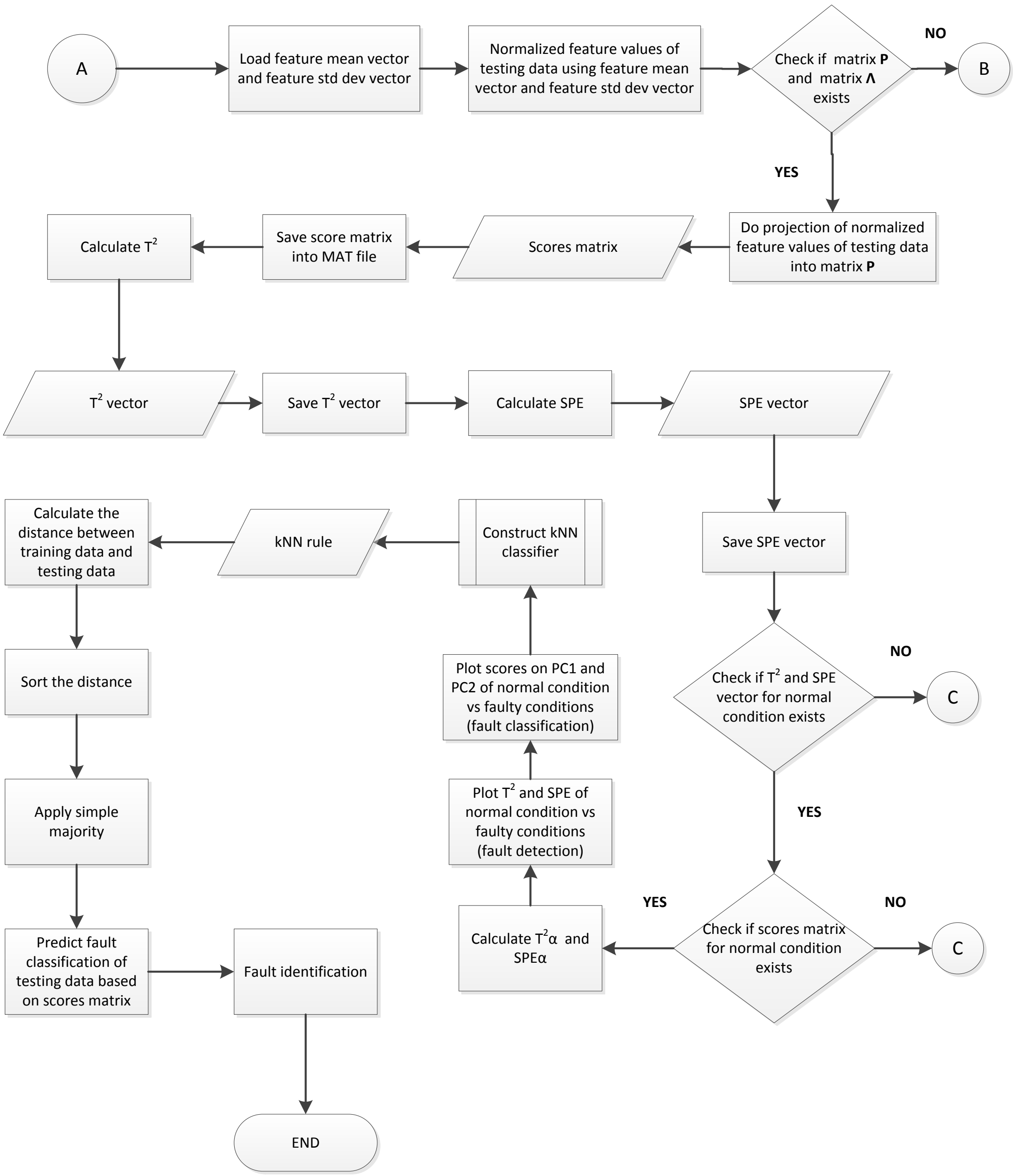


Figure 5.17 Integrated framework algorithm (Part B)

In the case where the reduced loading matrix $\widehat{\boldsymbol{P}}$ and reduced eigenvector matrix $\widehat{\boldsymbol{\Lambda}}$ exist in the workspace, the process continued with projecting the feature matrix of testing data $\boldsymbol{X}$ (50 rows × 30 columns) into the reduced matrix $\widehat{\boldsymbol{P}}$ (30 rows × r columns). The matrix multiplication of $\boldsymbol{X}$ and $\widehat{\boldsymbol{P}}$ would result in the scores matrix $\widehat{\boldsymbol{T}}$

of the testing data with dimension of 50 rows × *r* columns. At this point, the original data set (feature matrix of testing data) had been transformed to a new set of variables i.e., *r* principal components (PCs). The resulting scores matrix was then saved into a mat file for later use.

The next process was to calculate the T^2 and Q-statistic (or SPE) of the feature matrix from the testing data. This was carried out for each row (i.e., observation) of the matrix. The calculation of T^2 and SPE used Equation 4.23 and Equation 4.26 respectively. The results were stored in a vector of 1 row × 50 columns for each of the T^2 and Q-statistic calculations. The 50 columns contain values for T^2 or Q-statistic for the 50 observations. The vectors were then saved into the hard drive for the next processes.

Prior to plotting values of the T^2, Q-statistic and scores of the testing data, the algorithm proceeded with checking whether the values of T^2, Q-statistic and scores matrix for the training data (normal condition) existed in the workspace. The process resumed by returning to part A if those variables were not found in the workspace. If the variables were found in the workspace, the algorithm calculated the values of T_α^2 and Q_α (or SPE_α).

For the purpose of fault detection, the plots of T^2 and Q-statistic values for normal condition (no fault) and testing data were drawn up and analysed. A plot of T^2 consisted of T^2 values for the normal condition, T^2 values of the testing data, and a threshold value T_α^2 while a plot of the Q-statistic consisted of Q-statistic values for the normal condition, Q-statistic values of the testing data, and a threshold value Q_α. An example plot of T^2 and Q-statistic is shown in Figure 5.18 part (a) and (b) respectively. The first 50 samples were T^2 or Q-statistic values obtained from the feature matrix of normal condition (no fault), while the rest of the samples were the values of T^2 or Q-statistic calculated from feature matrix of faulty condition.

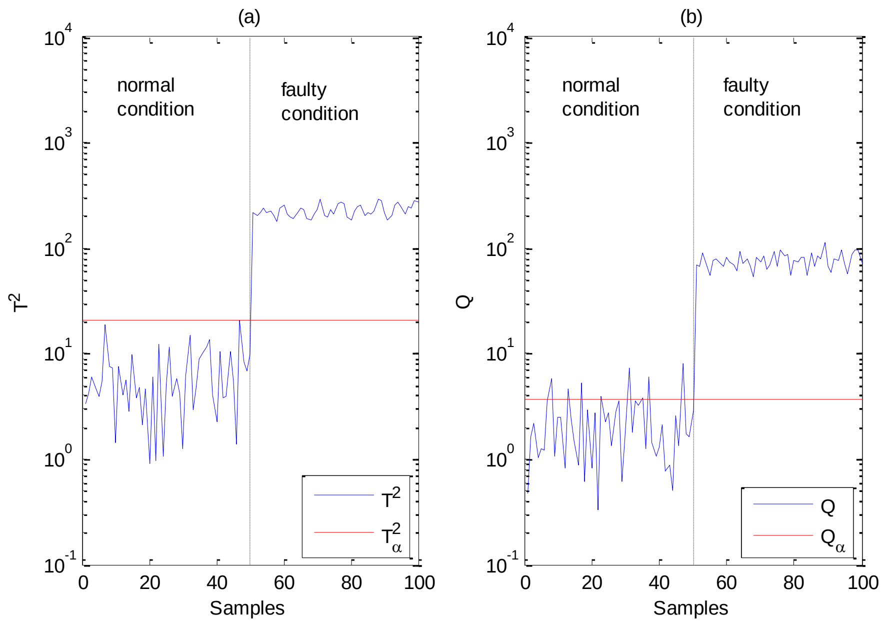


Figure 5.18 Example of T^2 and Q-statistic plot

The process continued with the fault classification using the scores matrix of the normal condition and faulty conditions. From the previous step, the scores matrix had dimensions of 50 rows × r columns. This means that there were r principal components (PCs) in the PCA model which were used to transform the original data set.

The example of the plot of scores on PC1 and PC2 was depicted in Figure 5.19 (a) and the plot of scores on PC1, PC2, and PC3 was shown in Figure 5.19 (b). These plots consisted of score values of the normal condition (no fault), fault1, fault2, and fault3. It was shown clearly that each type of fault was well separated in Figure 5.19 (b) although there was little overlap between normal condition (no fault) and fault2 in Figure 5.19 (a). The overlap could be reduced by adding more PCs as indicated in Figure 5.19 (b).

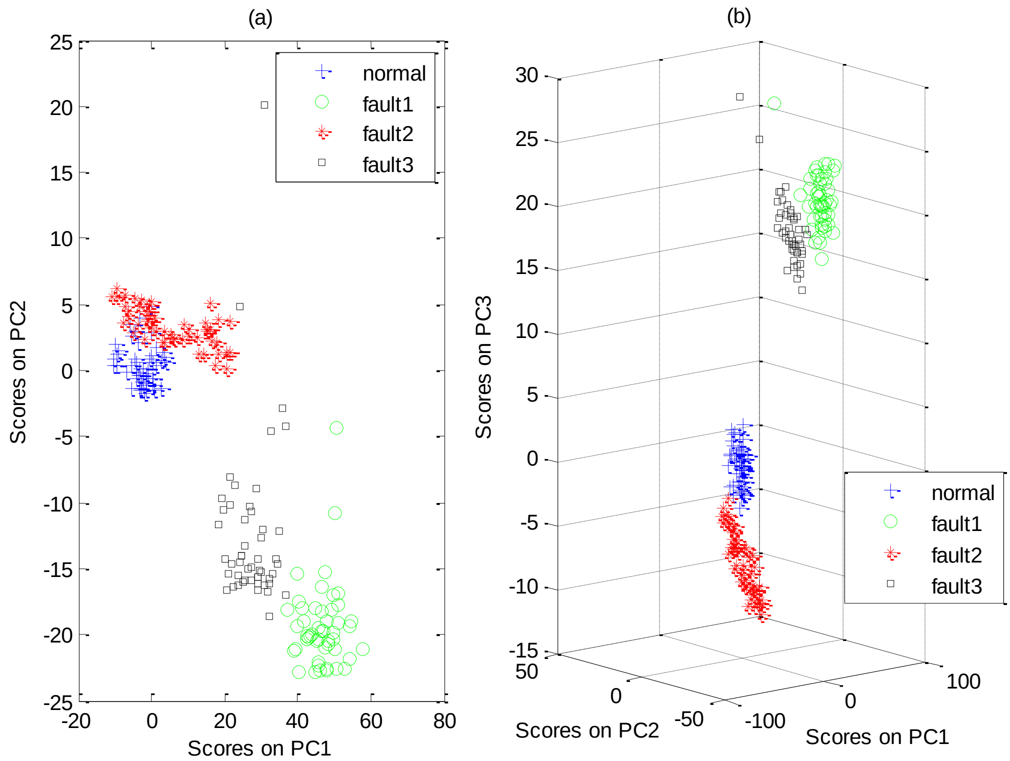


Figure 5.19 Example plot of scores (a) on PC1 and PC2, (b) on PC1, PC2, and PC3

After drawing up the plots for the purpose of fault detection and fault classification, the process continued with the construction of the classifier. In this study, the k-Nearest Neighbors (kNN) was used for fault classification and the detail of constructing the kNN rule will be presented in section 5.8.

A kNN rule held the position of training data and their class (type of fault). The fault classification process began by calculating the distance between a testing data point and the training data set. The type of distance used was defined in the kNN rule. The result of the calculation of the distance was then sorted in ascending order. Based on the defined number of *k* in the kNN rule, the algorithm chose *k* samples with the least distances. The class with more samples inbound was then assigned to the class of testing data points.

At this stage, the algorithm had performed fault detection, classification, and identification of the testing data for a particular type of fault. The whole process was then resumed by returning to Part A and starting again to evaluate the other types of faults. The whole procedure was also iterated again for the other Symlet types (sym4, sym8, and sym12). Once all of the faults and Symlet types had been performed, the algorithm ended.

5.8 Constructing the k-Nearest Neighbors (kNN) Rule

The process of constructing the kNN rule was shown in Figure 5.20. It began by recalling the scores matrix of normal condition (no fault) and scores matrix of all types of faulty conditions. For the purpose of constructing the kNN rule, the data series 1-70 were used as the training data set. All of the scores matrices were combined into one matrix where each row represented observations (samples) and each column represented scores. Since there was one score matrix of the normal condition and seven score matrices of faulty conditions, therefore the combined score matrix would have a size of 560 rows × 7 columns. Each row of this matrix was given a class name (i.e., type of fault) that corresponded to its type of fault. For instance, if the combined score matrix was formed from the original score matrix in the order of the normal condition, fault1, fault2,…, fault7, hence the order of the class name was exactly the same. The example of the combined matrix and the class name is depicted in Figure 5.21.

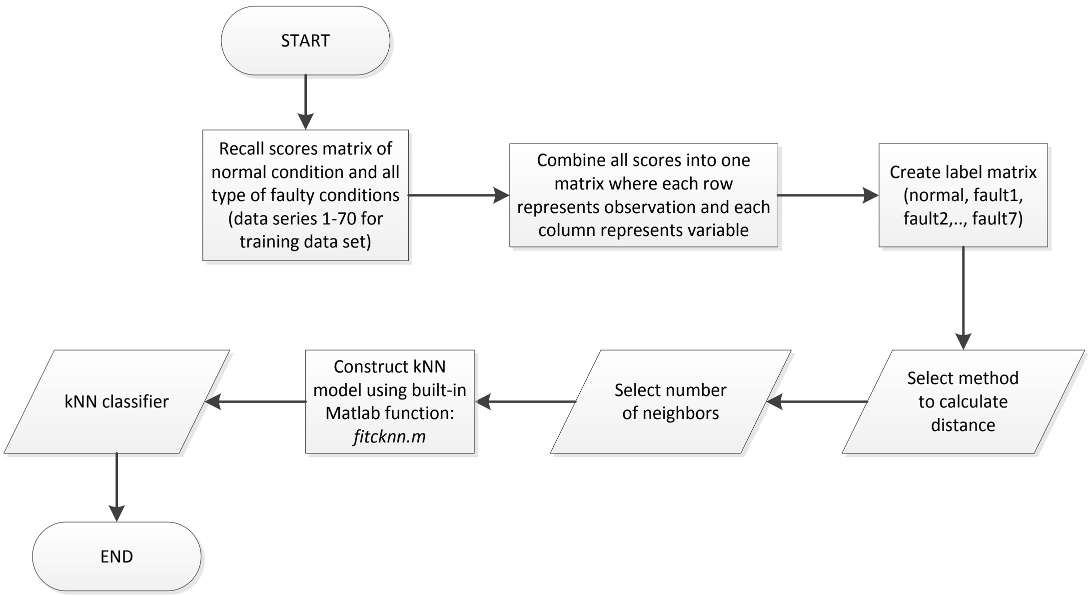


Figure 5.20 k-Nearest Neighbors construction

The process continued with selecting a method to calculate the distance between the testing data point and the training data set. In this kNN classifier, the Euclidean distance metric was used and the number of neighbors was varied from 1 to 10. The kNN rule was then constructed by using the MATLAB's built-in function '*fitcknn*' which required the combined score matrix as input arguments. An example of the kNN rule produced by the function '*fitcknn*' with the parameters: Euclidean distance, number of neighbors = 4, number of classes = 4, and number of observations = 280 is depicted in Table 5.1. The resulting kNN rule was then saved into the hard drive for later use in the classification process.

Table 5.1 Example of kNN parameters

Parameters	**Values**
Predictor Name	'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7'
Response Name	'Y'
Class Name	'normal', 'fault1', 'fault2', 'fault3'
Score Transform	'none'
Number of Observation	280
Distance	'euclidean'
Number of Neighbors	4

	Scores					Class (type of fault)
	X1	X2	X3	…	X7	Y
row 1 - 70						normal
						normal
						normal
						…
row 71 -140						fault1
						fault1
						fault1
						…
row 141 - 210						fault2
						fault2
						fault2
						…
row 211 - 280						fault3
						fault3
						fault3
						…
row 281 - 490						…
						…
						…
row 491 - 560						fault7
						fault7
						fault7

Figure 5.21 Combined score matrix in kNN rule

5.9 Evaluation of the Proposed Method's Performance

There were many parameters used and produced during the fault diagnosis processes. For instance, three types of Symlet (sym4, sym8, and sym12) were used to decompose the original signals, *r* principal components were produced from the PCA modelling and a number of *k* from 1 to 10 was evaluated with the kNN rule. All of these parameters produced combinations which affected the performance of the proposed method.

The performance of the proposed method, by means of the accuracy of fault identification, was evaluated by considering all combination parameters as given in the flowchart in Figure 5.22.

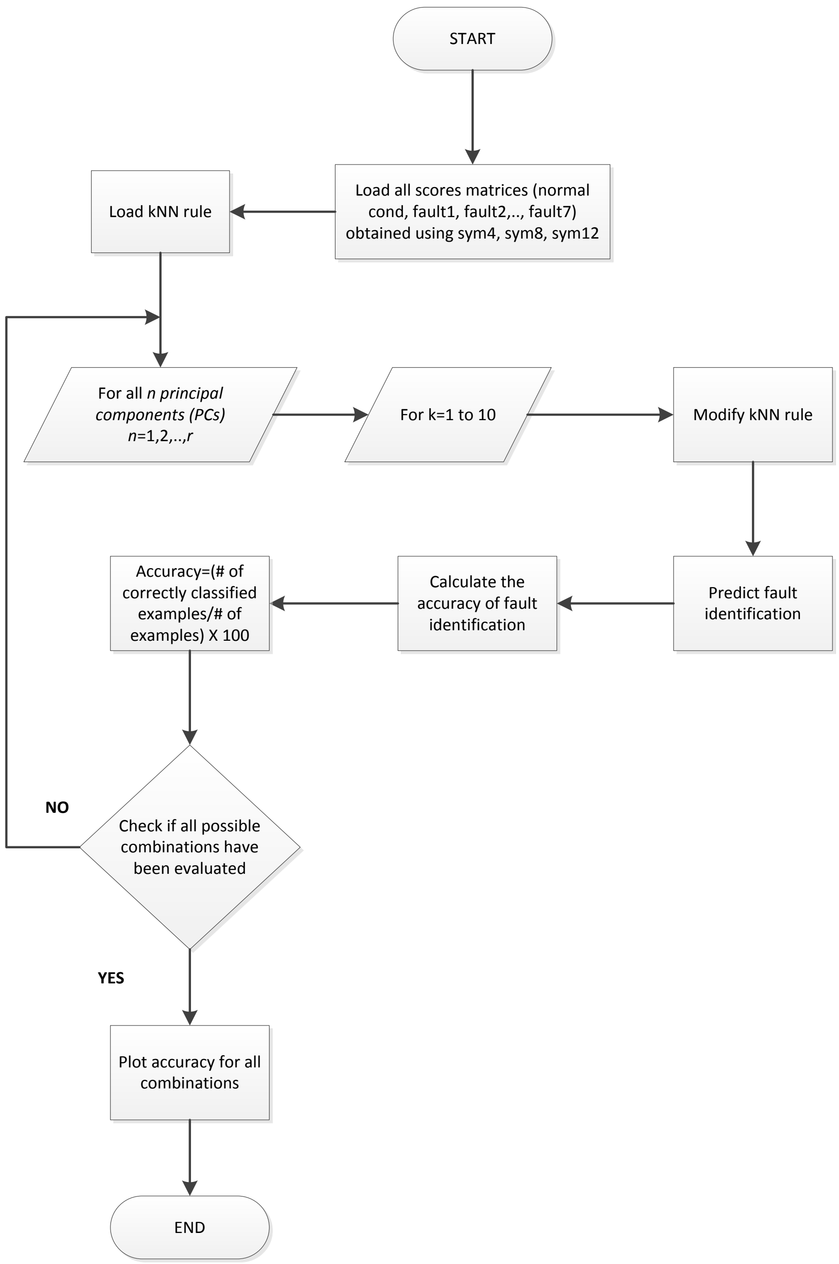


Figure 5.22 Evaluation of fault identification performance

All score matrices obtained using sym4, sym8, sym12 of the normal condition (no fault), fault1, fault2,.., fault7 were loaded from the hard drive. The procedure then continued to apply the kNN rule used to identify the fault.

Prior to calculating the accuracy, the kNN rule was modified by swapping its parameters; the number of the predictor was evaluated from 1 to *r* principal components (PCs) and the number of neighbors k was evaluated from 1 to 10. This step was iterated until all of the possible pairs were performed. For instance, if there were 7 PCs retained in each scores matrix, there would be 70 (7×10) modified kNN rules produced by the end of the iteration process.

Fault prediction was carried out for all fault scenarios using 70 modified kNN rules. At this stage, there were 3 groups of scores matrices (corresponds to sym4, sym8, sym12) where each group consisted of 8 scores matrices: normal condition, fault1, fault2,…, fault7. The accuracy of fault prediction of 70 modified kNN rules was then carried out for these 3 groups of scores matrices using the formula (# of correctly classified examples / # number of examples) × 100.

The process continued with checking if there was a case of fault prediction that had not been evaluated. The process resumed by returning to the beginning of the iteration process if the checking condition was not satisfied. In the case the checking condition was satisfied, the process continued with plotting the value of the accuracy of prediction for all cases and subsequently the whole process was ended.

In the following section, the proposed method was tested using external vibration data from ball bearing testing found in the NASA website at URL: http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/. The data set was taken from the Centre for Intelligent Maintenance Systems (IMS), University of Cincinnati [172]. The test aims to investigate the applicability and performance of the method in fault diagnosis.

5.10 Test of the Proposed Method Using External Vibration Data

The details related to the test rig setup are explained on a readme document provided in the zip file on the website. The bearing data set was provided in ASCII format which was compressed in ‘rar’ format and then zipped.

There were three groups of data where each group of data describes a test-to-failure experiment. The vibration data was recorded as 1-second snapshots and saved into individual files with recording interval of 10 minutes. For the purpose of testing the proposed method, the 2nd group which contained 984 individual files (data set) was used. Each data set had four channels which collected the vibration data of test-to-failure experiment of bearing 1, bearing 2, bearing 3, and bearing 4 respectively. At the end of the experiment, an outer race fault was detected in bearing 1 (channel 1).

The first channel (bearing 1) of the 2nd group of data was used to test the applicability and performance of the proposed method. All of the 984 data set was used in the test and the plot of four data sets which describe four different conditions of the bearing during the experiment is depicted in Figure 5.23. Data set #10, #450, #650 and # 890 corresponded to vibration data at the beginning, middle, region where sign of fault observed and end of the experiment respectively.

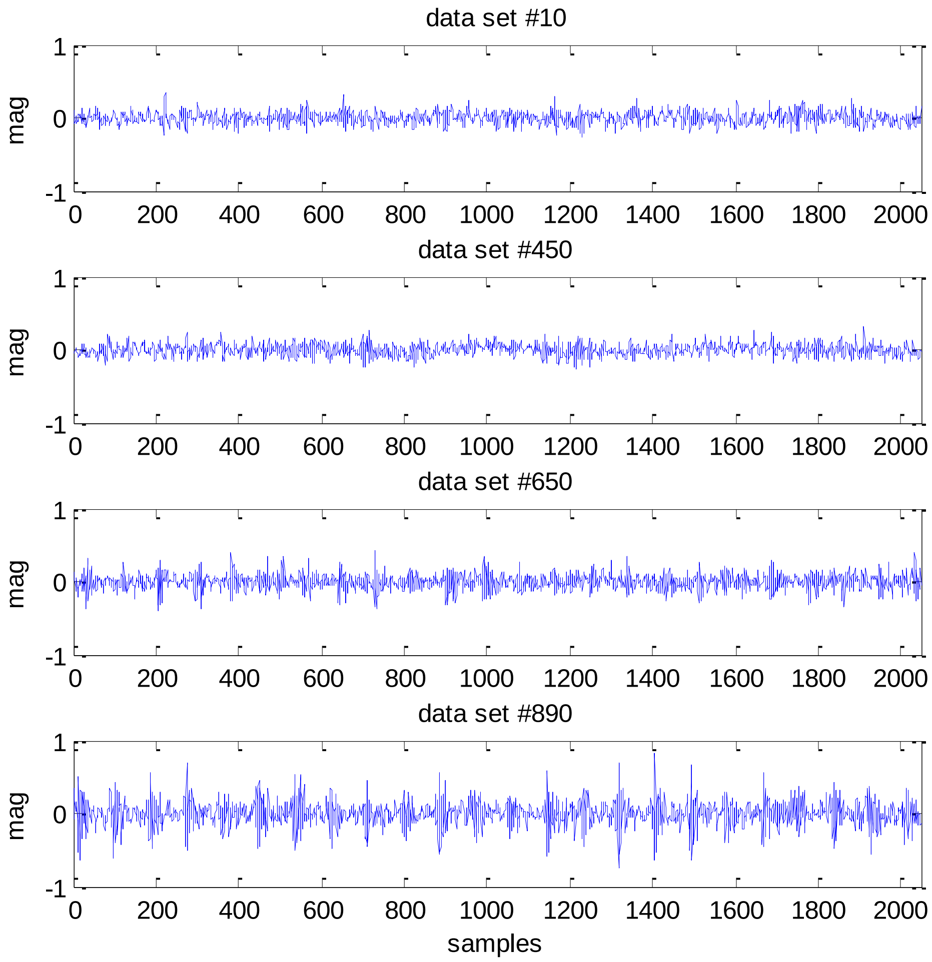


Figure 5.23 Time waveform of vibration signal of bearing 1, data set # 10 (at the beginning region of test), data set #450 (at the middle region of test), data set #650 (sign of fault observed) and data set #890 (at the end region of test)

The whole data set which consisted of 984 individual files recorded a test-to-failure experiment. Based on the time waveform as shown in Figure 5.23, the whole data set may be divided into two regions, #1-450 for the no fault region and #650-984 for the fault region. The data set #1-100 was used to build the PCA model. For k-Nearest Neighbors (kNN) training, the data set #101-200 and data set #651-750 were used to train kNN rule for the no fault (normal condition) and outer race fault respectively, while data set #201-300 and data set #801-900 respectively was used for testing data of the normal condition and outer race fault.

Symlet 8 (sym 8) wavelet decomposed the vibration data of channel 1 through the use of MRA up to 5 levels. The six features were then extracted from each of the cA parts of the decomposition results, as proposed in this research. These features were used to build the PCA model for fault diagnosis purposes.

The results of fault detection are shown in Figure 5.24. It is shown that both T^2 and Q-statistic are able to detect the fault occurring at the end region of the experiment. The T^2 indicated that abnormal behaviour of the system (fault occurred) was observed at the data set #600 as indicated by increasing T^2 values. The abnormal behaviour was also detected by Q-statistics at an earlier region (at the data set #500), since the Q-statistic is more sensitive to changes of system behaviour than the T^2.

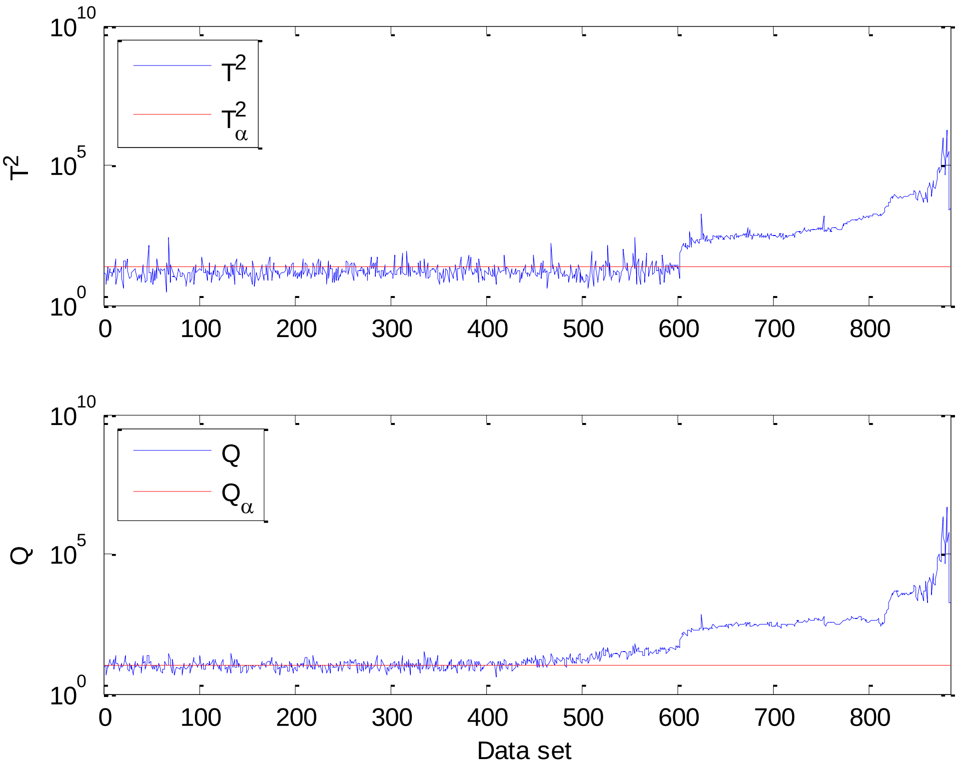


Figure 5.24 Fault detection (outer race fault) using T^2 and Q-statistic of external vibration data

From the results, the proposed method, by means of T^2 and Q-statistic are able to detect the fault occurring at the end of the test-to failure of bearing 1. The result matched with the actual bearing condition at the end of the experiment where the outer race fault was detected.

The next step of the test which used the external data was to test the fault classification performance using scores on principal components. The PCA model of external data obtained by using the proposed method retained 8 principal components (PCs) which had 95% of variance. The first two scores on PC1 and PC2 of the testing data was plotted as depicted in Figure 5.25. It is shown that the clustering effect is clearly visible. This means that the first two scores on the principal components was able to separate the normal condition and the outer race fault of the vibration signal from the external data with very good result.

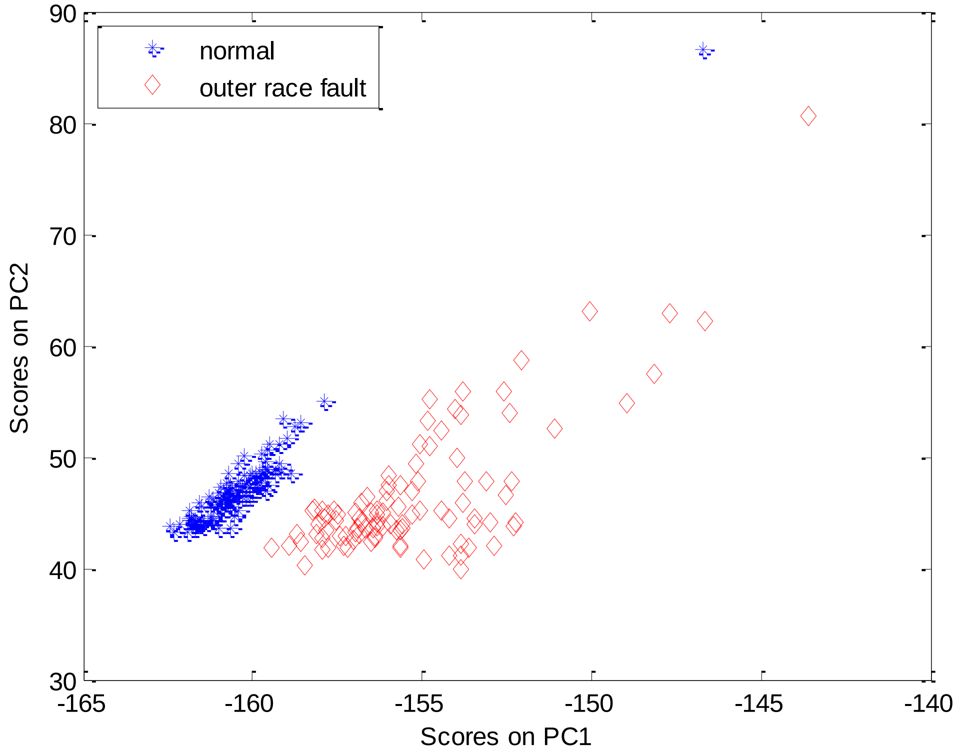


Figure 5.25 Plot of scores on PC1 and PC2

In this test, the performance of the kNN rule was evaluated by varying the number of neighbors *k* and the number of predictors (features). The number of *k* was varied from 1 to 30 while the number of predictors was varied from 1 to 8. Figure 5.26 shows the identification accuracy of the kNN while varying the number of *k* and PC(s). The lowest identification accuracy is for the kNN rule using only 1 predictor (1 PC) while the highest accuracy is that with 8 PCs. The results are as expected since 8 PCs contained more essential information, hence resulting in an improved kNN rule. In the case of varying *k*, the highest identification accuracy of the kNN rule of 8 PCs was obtained with 10 neighbors. It is also shown that adding additional neighbors into the model reduced identification accuracy. All of the findings as presented and discussed above suggest promising results, by means of general applicability and performance of the proposed method.

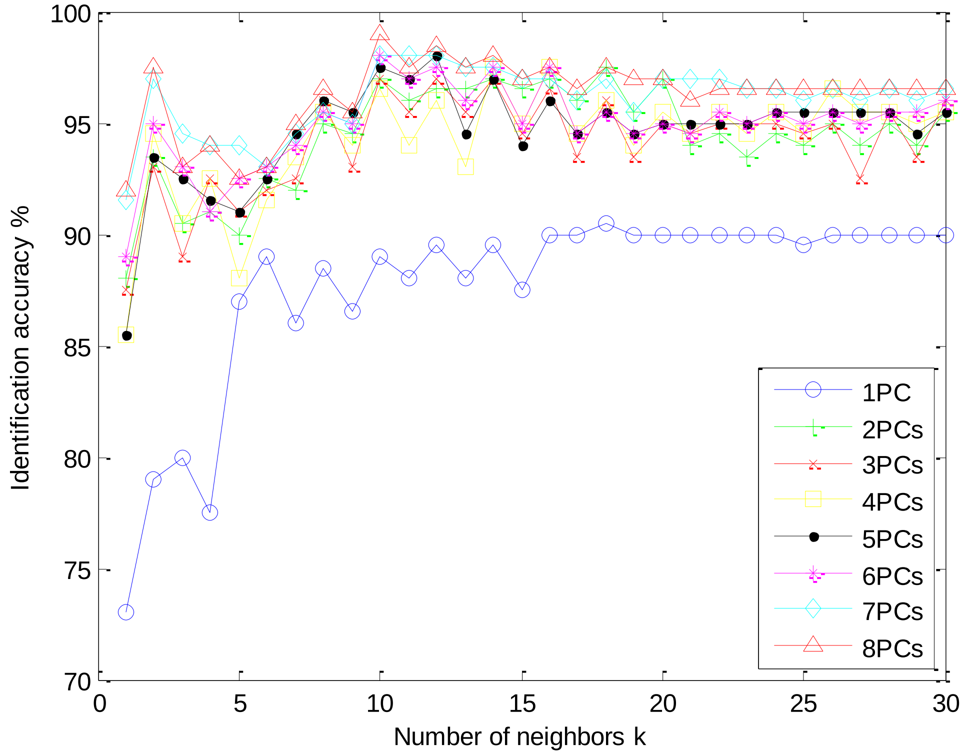


Figure 5.26 Performance of k-Nearest Neighbors (kNN) rule

The next chapter describes setup of the centrifugal pump test rig and the vibration data acquisition process.

CHAPTER SIX

6 The Centrifugal Pump Test Rig and Vibration Data Acquisition

The proposed method for fault detection and diagnosis of the centrifugal pump described in the previous chapter was tested using the vibration data taken from a Spectra Quest machinery fault simulator which was set up with a centrifugal pump configuration. The vibration data was acquired for two operating conditions of the centrifugal pump which are the normal condition (without fault) and a faulty condition (with fault). There were 7 types of artificial faults introduced in the test rig, namely cavitation (fault1), impeller fault (fault2), bearing fault (fault3), blockage condition (fault4), impeller fault and cavitation (fault5), impeller fault and blockage condition (fault6), and bearing fault and cavitation (fault7). The faulty conditions were carefully controlled and sequentially introduced to the test rig. The controlled vibration data was used to train the proposed method in order to evaluate the performance and effectiveness of fault detection and diagnosis for the centrifugal pump. Figure 6.1 shows a schematic diagram of the vibration data acquisition process used in the proposed method.

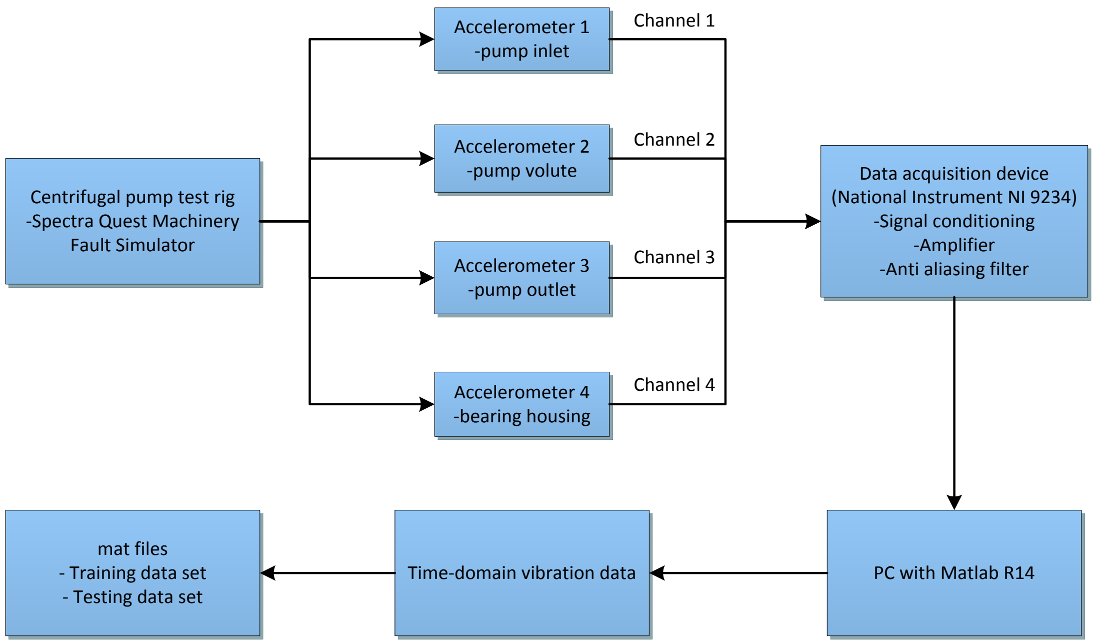


Figure 6.1 Data acquisition process

6.1 Centrifugal Pump Test Rig

The vibration signals for both normal and faulty condition of a centrifugal pump were collected from a Spectra Quest machinery fault simulator. Figure 6.2 and Figure 6.3 shows photographs of the centrifugal pump-tank configuration and the detail of the accelerometers' locations respectively. The suction and discharge sides of the centrifugal pump were fitted with manual valves and pressure gauges. The test rig consists of a centrifugal pump driven by a variable-speed AC motor through a shaft and belt-pulley mechanism. The shaft was connected to the AC motor by a fixed coupling and the belt-pulley mechanism which had a ratio of 1:1. The connecting shaft was supported by two roller bearings that were known to be in good condition. The rotational speed of AC motor was controlled by a speed controller and a tachometer was used to monitor its speed. The rotational speed was set to 35 Hz (2100 RPM) for the whole of the data acquisition process.

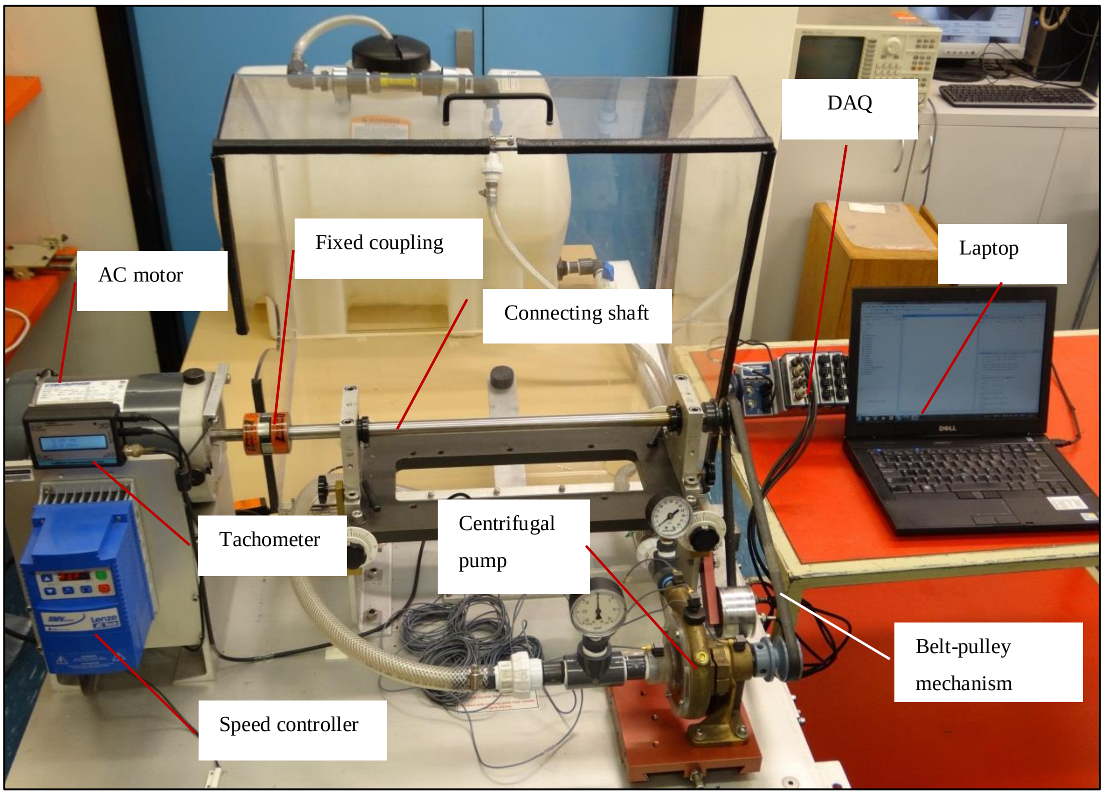


Figure 6.2 Centrifugal pump test rig set up

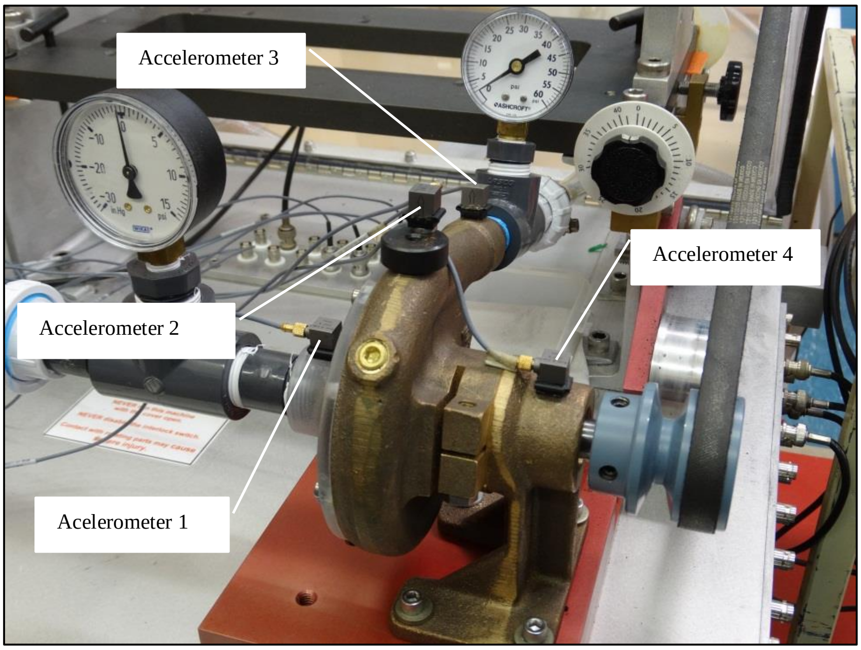


Figure 6.3 Detail of the accelerometer (channel) locations

6.2 Artificial Fault Conditions

There were 7 types of fault conditions considered in this research which consisted of 4 single fault conditions and 3 multi-fault conditions. The single fault conditions included cavitation, impeller fault, bearing fault, and blockage condition.

The cavitation condition was artificially introduced to the system by reducing the flow rate of the pump’s suction which was done by partially closing the suction valve. The cavitation condition was allowed to fully develop before the data was taken. The pump cover was made of transparent polycarbonate cover which allowed the cavitation to be easily observed. Figure 6.4 shows the transparent pump cover with the fully developed cavitation clearly visible.

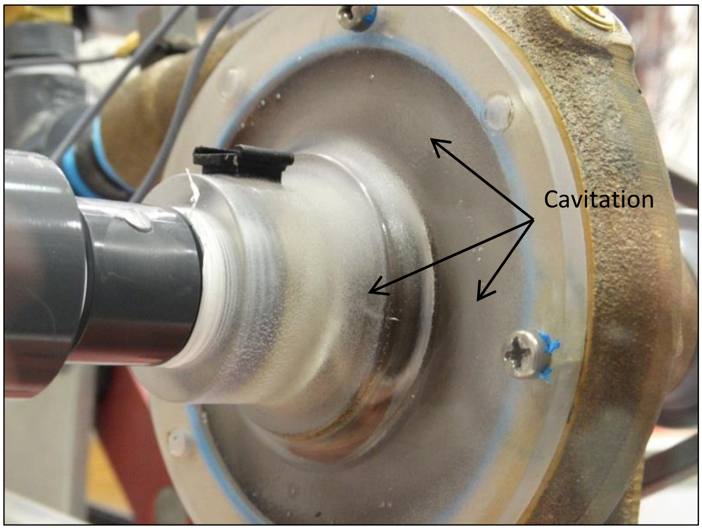


Figure 6.4 Cavitation condition observed in the centrifugal pump

The pump’s impeller used in this study was made of brass and consisted of five vanes. The impeller was an open impeller type that was overhung. The impeller was fitted in the volute house by using a shaft bearing. The artificial fault was introduced by cutting the impeller vanes at two locations for each blade as shown in Figure 6.5.

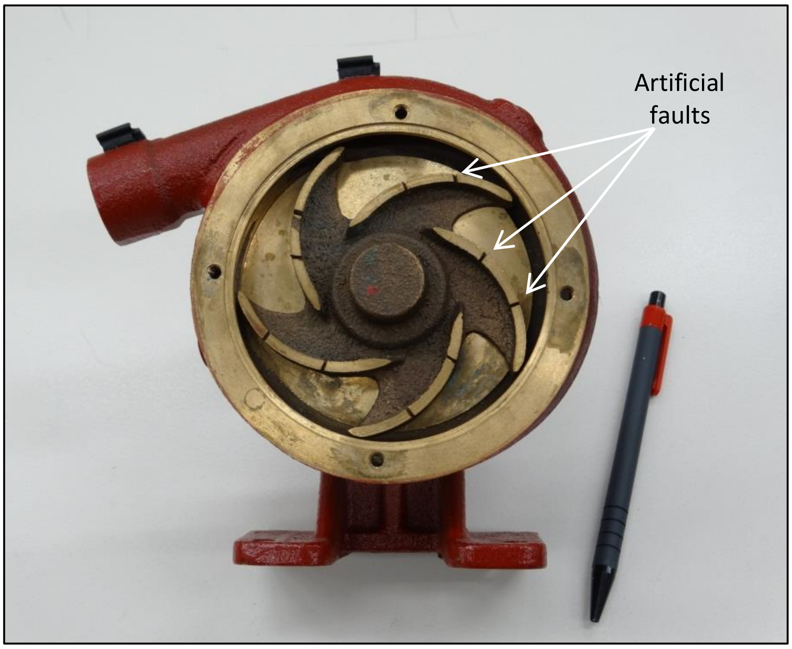


Figure 6.5 Faulty impeller

The type of bearing used in the centrifugal pump was a shaft bearing as depicted in Figure 6.6 which is commonly used in a water pump. The fault was introduced into the shaft bearing by applying an impact to the shaft bearing housing. The impact was not applied directly to the bearing housing. However it was inserted into a metal sleeve and an impact from a rubber head hammer was then applied. This procedure was used since the shaft bearing was manufactured as one integrated piece; hence disassembling the bearing without damaging the whole shaft bearing was not possible. The type of fault obtained was small localized spalling in the outer race as can be seen in Figure 6.7.

Figure 6.6 Shaft bearing

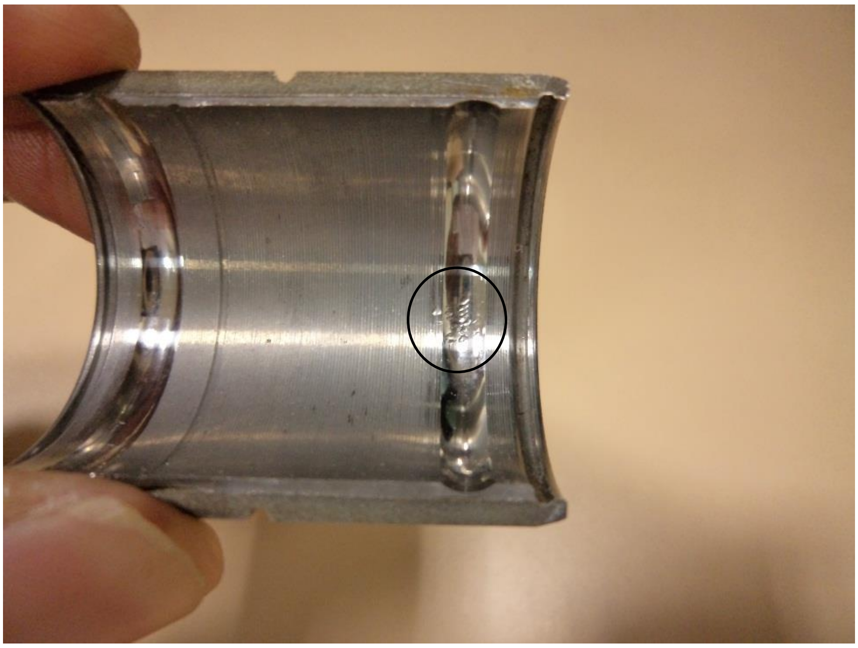


Figure 6.7 Faulty shaft bearing with small localized spalling

The blockage condition was artificially introduced by reducing the flow rate in pump suction. This was done by partially closing the manual valve in the suction side until the pressure gauge in the suction side indicated -4 in.Hg. This set up would make a condition just before cavitation where the cavitation bubbles were not observed, yet the pressure in the suction side has dropped.

6.3 Accelerometers and Data Acquisition Device

There were four accelerometers used in this experiment which were Deltatron accelerometers type 4507 B from Bruel & Kjaer. The accelerometers no 1 to 4 (or channel 1 to channel 4) were mounted respectively on the pump's inlet, pump's volute, pump's outlet and bearing housing as illustrated in Figure 6.3. All the accelerometers were glued-mounted on the centrifugal pump and connected to the transducer interface connectors which were then connected to the data acquisition device inputs.

Data acquisition module from National Instruments type NI 9234 was used for the vibration data acquisition. The module has been equipped with accelerometers, signal conditioning and anti-aliasing filters. The module was fitted to the NI cDAQ-9178 chassis which was connected to the PC via USB cable. MATLAB R2014a with Data Acquisition Toolbox was used to control the data acquisition process. MATLAB codes to control the data acquisition process are listed in Appendix A.

The vibration data of each normal and faulty condition was acquired using a simultaneous sampling rate of 48 kHz (48192 Hz) for each channel. The vibration data was recorded in multiple segments of 1 second duration and was then saved in the hard drive in the form of mat files.

6.4 Data Structure

Vibration data acquisition was carried out for each normal and faulty condition for 1 second duration with 1 second pause between recordings. The data acquisition process resulted in 120 segments for the normal condition (no fault) and for each fault condition and was saved using the mat file format. Hence, in total there were 960 files saved (120 segments x 8 normal and faulty conditions) during the data acquisition process. The data segments were subsequently used to train the proposed fault diagnosis method and to test its performance in detecting and diagnosing faults in the centrifugal pump.

The next chapter presents the results of the proposed wavelet-PCA method for fault diagnosis of the centrifugal pump. It present the results of each PCA model generated from all channels and the comparison of the accuracy performance of each PCA model is also provided.

CHAPTER SEVEN

7 Results and Discussion of the Proposed Method

This chapter presents results of fault diagnosis of a centrifugal pump obtained using the proposed method. The use of sym-n wavelet types in the feature extraction step was evaluated and compared in order to select the most appropriate Symlet wavelet type to be used in a centrifugal pump fault diagnosis. A number of single and multi-faults namely, cavitation (fault1), impeller fault (fault2), bearing fault (fault3), blockage (fault4), impeller fault with cavitation (fault5), impeller fault with blockage (fault6), and bearing fault with cavitation (fault7) along with the normal condition (no fault) were introduced to the pump test rig to test the performance of the proposed method.

In this research, there were four groups of PCA models produced by the proposed method. These four group models related to the four accelerometers mounted at different locations on the centrifugal pump. Therefore PCA model 1, 2, 3, and 4 were based on the vibration data collected from pump's inlet (channel 1), pump's volute (channel 2), pump's outlet (channel 3), and bearing housing (channel 4) of the centrifugal pump respectively.

The identification accuracy of the proposed method was examined for all four groups of PCA models. This was aimed to find the best mounting location of the accelerometer on a centrifugal pump for fault diagnosis purposes.

7.1 PCA Model Developed Using Vibration Signals Collected from Pump's Inlet

The time waveform of the normal condition (no fault) and faulty conditions of vibration signals acquired from channel 1 are depicted in Figure 7.1. They show time waveform plots of the vibration acceleration of the centrifugal pump under the normal condition (normal), pump with cavitation (fault1), pump with impeller fault (fault2), pump with bearing fault (fault3), pump with blockage condition (fault4), pump with both impeller fault and cavitation (fault5), pump with both impeller fault and blockage condition (fault6), and pump with both bearing fault and cavitation (fault7), respectively. The time waveform plots were taken from data set #15 (sequence 15 of 120 data sets) and the first 1377 samples corresponding to 1x shaft rotation are shown.

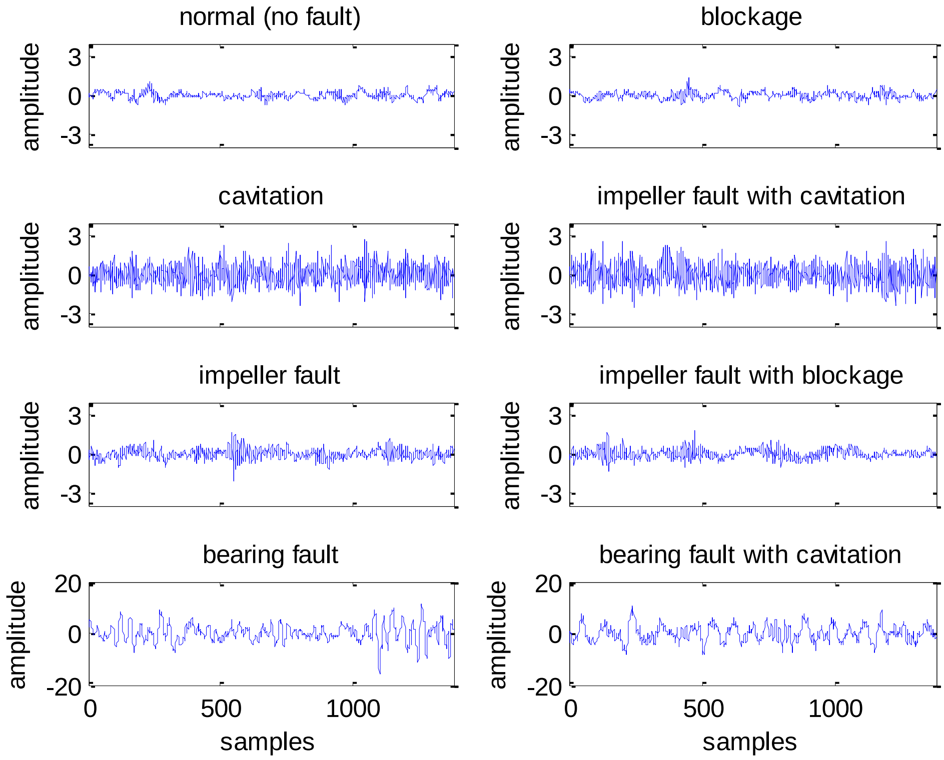


Figure 7.1 Time waveform of normal (no fault) and faulty components acquired from channel 1 (data set #15)

To build the PCA model, a set of training data was selected from the normal condition vibration signal. In this research, data set #1 to 70 was used as training data for building the PCA model while the data set # 71 to 120 was chosen as testing data. The training data was processed using the proposed method which produced a PCA model consisting of the eigenvalue (matrix $\boldsymbol{\Lambda}$) and loading matrix $\boldsymbol{P}$. The results of eigenvalues obtained using sym4 (PCA model 1a), sym8 (PCA model 1b), and sym12 (PCA model 1c) for all principal components are depicted in Table 7.1.

From Table 7.1, it is shown that the proposed PCA method has transformed the original variables (30 features) into 30 principal components which are uncorrelated.

The sym-n wavelet decomposition produces different eigenvalues for each principal component with half of them having values close to zero. Consequently, only half of the principal components are found to retain almost all the essential information in the model. For the purpose of PCA modelling, the total variance to be accounted for in the model was set at 95% and the cumulative percent variance (CPV) was calculated using Equation 4.22. The result of the CPV is depicted in Figure 7.2 in the Pareto chart which shows the eigenvalues for each principal component and their cumulative percentage values as well. It is shown that, for the 95% variance level to be accounted for in the model, the PCA model obtained using sym4 wavelet decomposition retained 7 PCs, while that using sym8 and sym12 retained 8 PCs.

Table 7.1 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 1)

Principal component	Eigenvalue			Principal component	Eigenvalue		
	sym4	sym8	sym12		sym4	sym8	sym12
PC1	17.3	17.0	16.8	PC16	6E-04	1E-03	2E-03
PC2	3.8	4.7	4.4	PC17	2E-04	3E-04	4E-04
PC3	3.1	2.9	2.7	PC18	7E-05	2E-04	3E-04
PC4	1.7	1.6	1.8	PC19	2E-05	2E-05	2E-05
PC5	1.1	1.0	1.1	PC20	1E-05	7E-06	1E-05
PC6	0.9	0.7	0.8	PC21	3E-06	5E-06	8E-06
PC7	0.6	0.5	0.7	PC22	2E-06	2E-06	2E-06
PC8	0.5	0.5	0.5	PC23	4E-08	3E-07	7E-07
PC9	0.3	0.4	0.4	PC24	2E-08	9E-08	2E-07
PC10	0.2	0.2	0.2	PC25	1E-09	4E-09	4E-09
PC11	0.2	0.1	0.2	PC26	4E-10	9E-10	2E-09
PC12	0.1	0.1	0.1	PC27	1E-10	2E-10	3E-10
PC13	0.1	0.1	0.1	PC28	7E-11	8E-11	2E-10
PC14	0.1	0.1	0.1	PC29	9E-12	2E-11	2E-11
PC15	0.0	4E-02	5E-02	PC30	2E-12	2E-12	7E-12

The number of retained PCs in the model resulted in the reduction of matrix $\boldsymbol{\Lambda}$ and matrix $\boldsymbol{P}$. Therefore in channel 1, three PCA models were obtained from sym4, sym8, and sym12 wavelet decomposition, each of which contained the reduced matrix $\widehat{\boldsymbol{\Lambda}}$ and reduced matrix $\widehat{\boldsymbol{P}}$. The sym4 wavelet produced the reduced matrix $\widehat{\boldsymbol{\Lambda}}$ and reduced matrix $\widehat{\boldsymbol{P}}$ with dimension 7×7 and 30×7 respectively while the sym8 and sym12 wavelet produced matrices with dimension 8×8 and 30×8 for reduced matrix

$\widehat{\boldsymbol{\Lambda}}$ and reduced matrix $\widehat{\boldsymbol{P}}$ respectively. The schematic diagram of the resulting PCA models obtained from channel 1 are depicted in Figure 7.3.

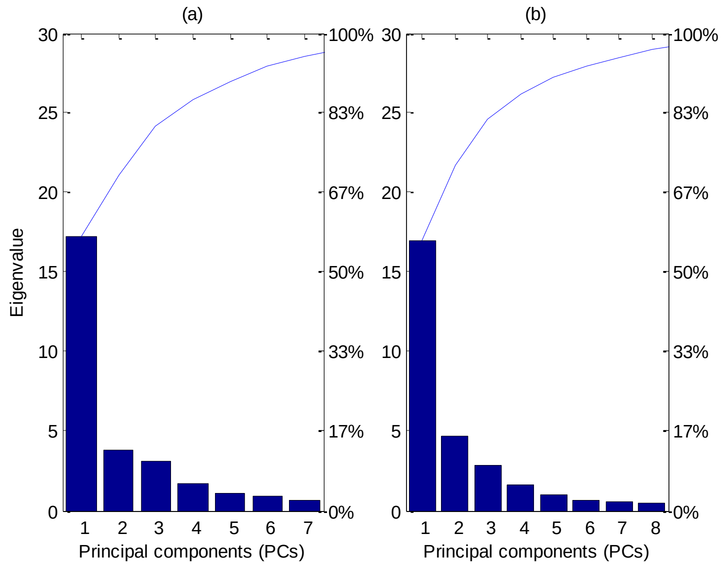


Figure 7.2 Number of principal components retained in the model based on 95% of the variance. (a) 7 PCs retained in the model by using sym4, (b) 8 PCs retained using sym8 and sym12

The relationship between the original features and principal components can be examined by looking at the correlation matrix which is sometimes referred to as *principal component loadings*. The visualization of the correlation matrix between the 8 retained principal components and the features obtained using sym8 wavelet decomposition is depicted in Figure 7.4. It is shown that the first principal component PC1 is positively correlated with all of the original features and has high correlation value with most of the original features. This means that PC1 has the strongest correlation with the original feature, hence it can sufficiently characterize the centrifugal pump condition; this is consistent with the idea of cumulative percent variance (CPV) in Figure 7.2 where PC1 has the highest eigenvalue i.e., contains the most essential information within the PCA model. The other principal components, PC2 and PC3 are positively related with 12 original features and negatively related with 18 original features. Both of the principal components also have relatively high correlation value with the original features compared to the rest of the principal components. Therefore, the first three principal components (PC1, PC2, and PC3) were used in the fault classification process through the scores plot.

Visualization of correlation matrices between principal components and the original features obtained using sym4, sym8 and sym12 wavelet decomposition for channel 1, 2, 3, and 4 are presented in Appendix B.

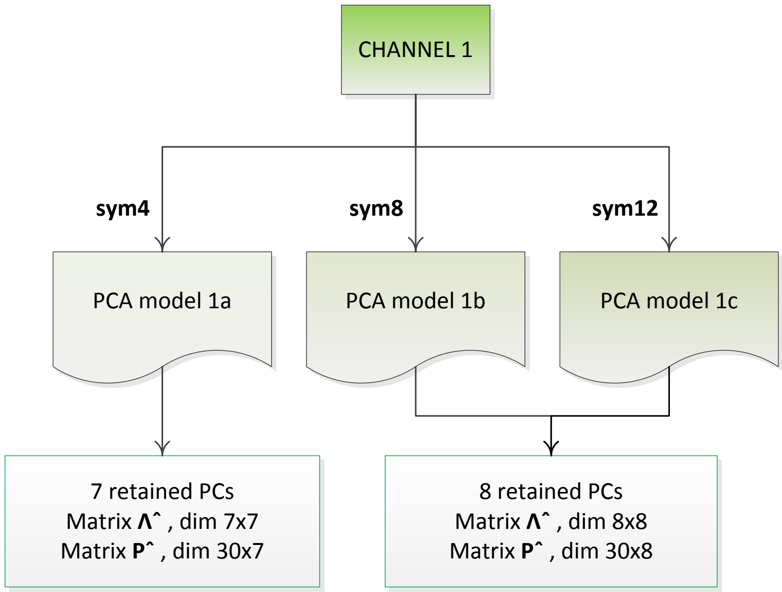


Figure 7.3 Three PCA models obtained from channel 1

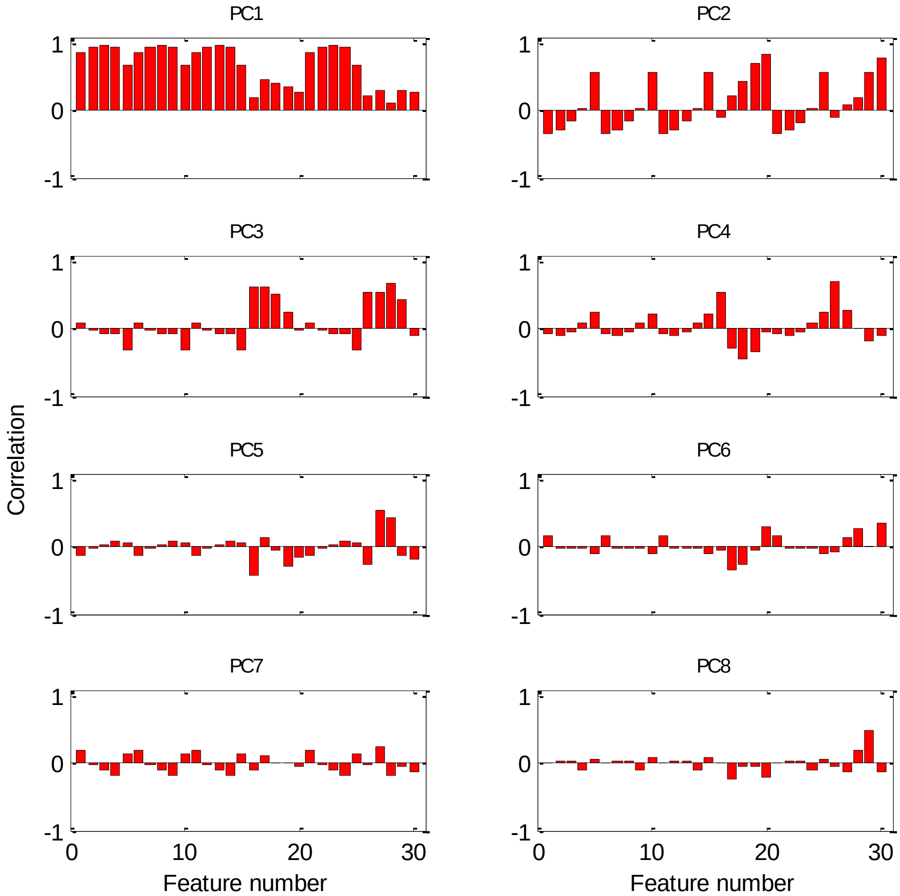


Figure 7.4 Correlation between principal components (PCs) and the features of channel 1 (using sym8 decomposition)

7.2 PCA Model Developed Using Vibration Signals Collected from Pump's Volute

The vibration characteristic collected from channel 2 for all of pump conditions are shown in Figure 7.5. For the consistency with the channel 1 data, data set #15 with 1377 samples is again used here to visualize the time waveform.

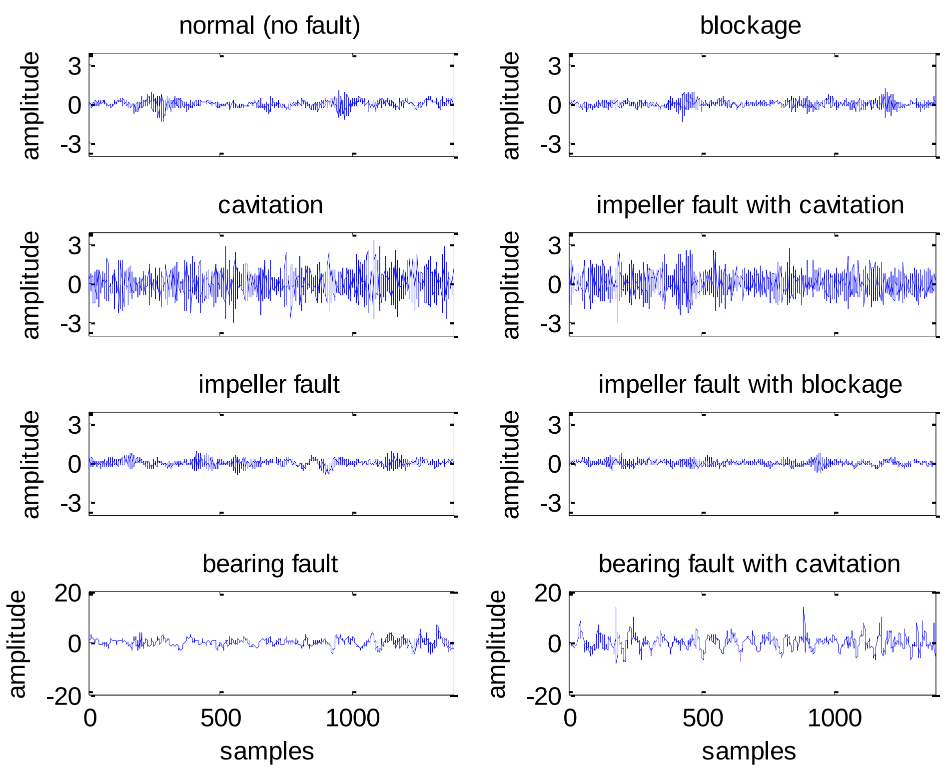


Figure 7.5 Time waveform of normal (no fault) and faulty components acquired from channel 2 (data set #15)

The eigenvalues obtained from channel 2 is depicted in Table 7.2. It shows a similar pattern with those obtained from channel 1 where half of the PCs are having the eigenvalue amplitude close to zero. With the same amount of variance accounted for as in the previous model (channel 1), that is 95%, the PCA model generated from channel 2 retained 7 principal components for all types of Symlet wavelet decomposition. The Pareto chart showed 95% cumulative percent variance as depicted in Figure 7.6.

For this vibration channel, based on the 95% of variance to be accounted for in the model, the proposed method produces three PCA models which have identical dimensions for reduced matrix $\widehat{\boldsymbol{\Lambda}}$ and reduced matrix $\widehat{\boldsymbol{P}}$ as shown in Figure 7.7.

The visualization of the correlation matrix between the PCs and the features obtained using sym8 wavelet decomposition is represented in Figure 7.8 which shows a strong positive correlation between the PC1 and the features. It also shows that the first three PCs have the highest correlation values which mean they characterize the most important information of the original features for channel 2.

Table 7.2 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 2)

Principal component	Eigenvalue			Principal component	Eigenvalue		
	sym4	sym8	sym12		sym4	sym8	sym12
PC1	18.6	18.1	17.7	PC16	9E-04	9E-04	1E-03
PC2	3.8	3.7	3.5	PC17	2E-04	2E-04	2E-04
PC3	2.0	2.6	3.2	PC18	3E-05	5E-05	1E-04
PC4	2.0	2.0	1.9	PC19	2E-05	3E-05	4E-05
PC5	0.9	1.1	1.1	PC20	8E-06	1E-05	1E-05
PC6	0.9	0.8	0.8	PC21	1E-06	2E-06	4E-06
PC7	0.6	0.6	0.5	PC22	1E-06	8E-07	1E-06
PC8	0.5	0.4	0.5	PC23	2E-08	2E-07	2E-07
PC9	0.3	0.2	0.3	PC24	4E-09	2E-08	1E-07
PC10	0.2	0.2	0.2	PC25	2E-10	5E-10	2E-09
PC11	0.1	0.1	0.2	PC26	9E-11	4E-10	8E-10
PC12	0.1	0.1	0.1	PC27	2E-11	3E-11	8E-11
PC13	5E-02	4E-02	4E-02	PC28	1E-11	2E-11	5E-11
PC14	4E-02	4E-02	3E-02	PC29	5E-12	8E-12	1E-11
PC15	1E-02	1E-02	1E-02	PC30	5E-13	7E-13	1E-12

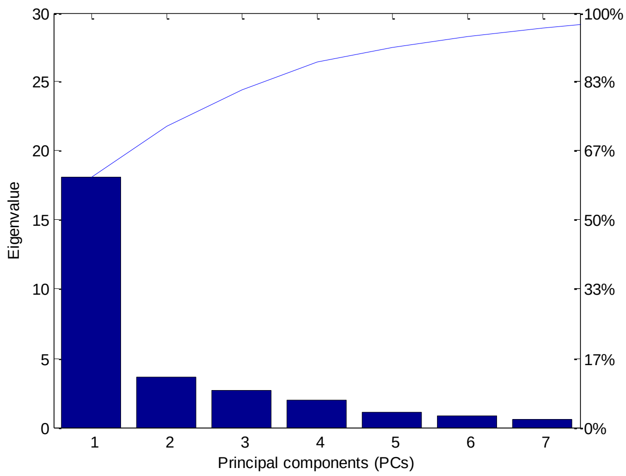


Figure 7.6 Seven PCs retained in the model by using sym4, sym8, and sym12 (based on 95% of the variance)

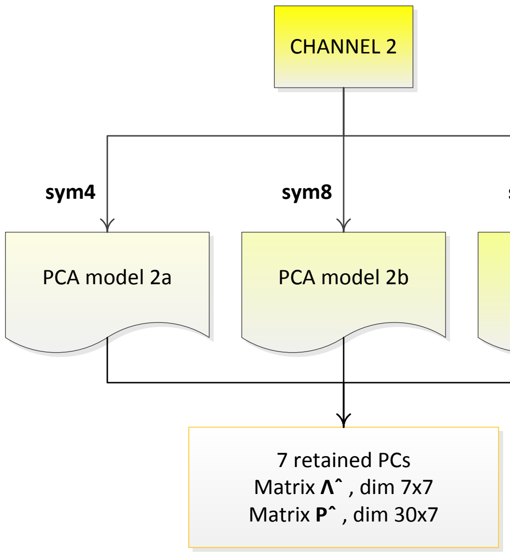


Figure 7.7 PCA models of channel 2

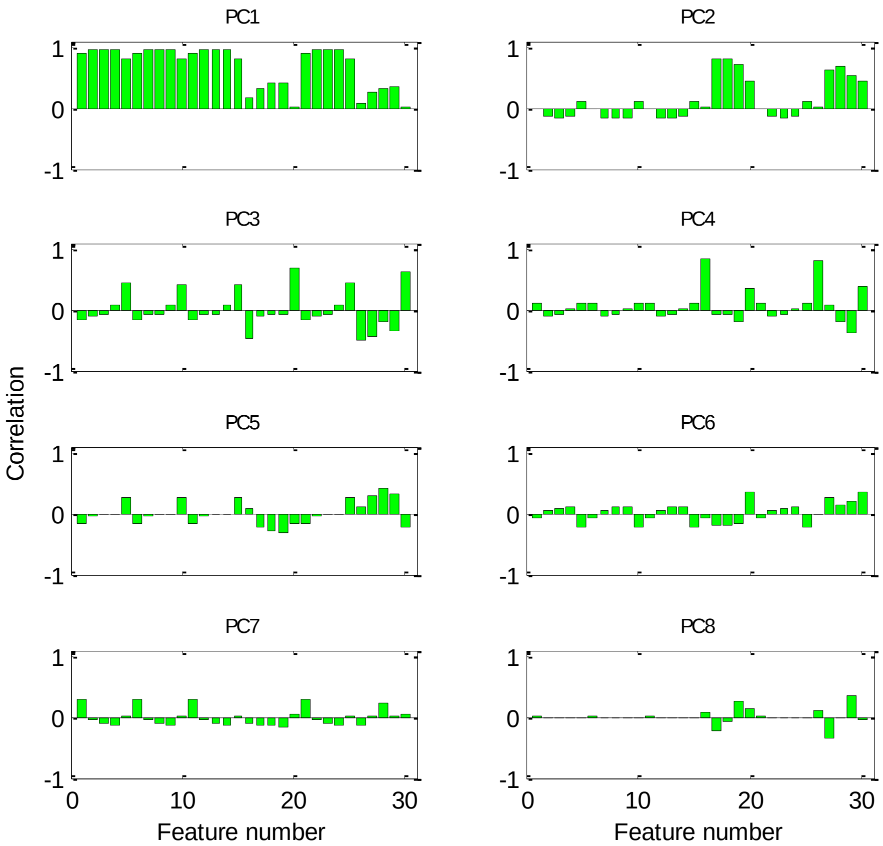


Figure 7.8 Correlation between principal components (PCs) and the features of channel 2 (using sym8 decomposition)

7.3 PCA Model Developed Using Vibration Signals Collected from Pump's Outlet

The time-domain of the vibration signal for normal and faulty conditions taken from channel 3 is shown in Figure 7.9. The time waveform is again plotted using data set #15 with 1377 samples to be consistent to those shown previously.

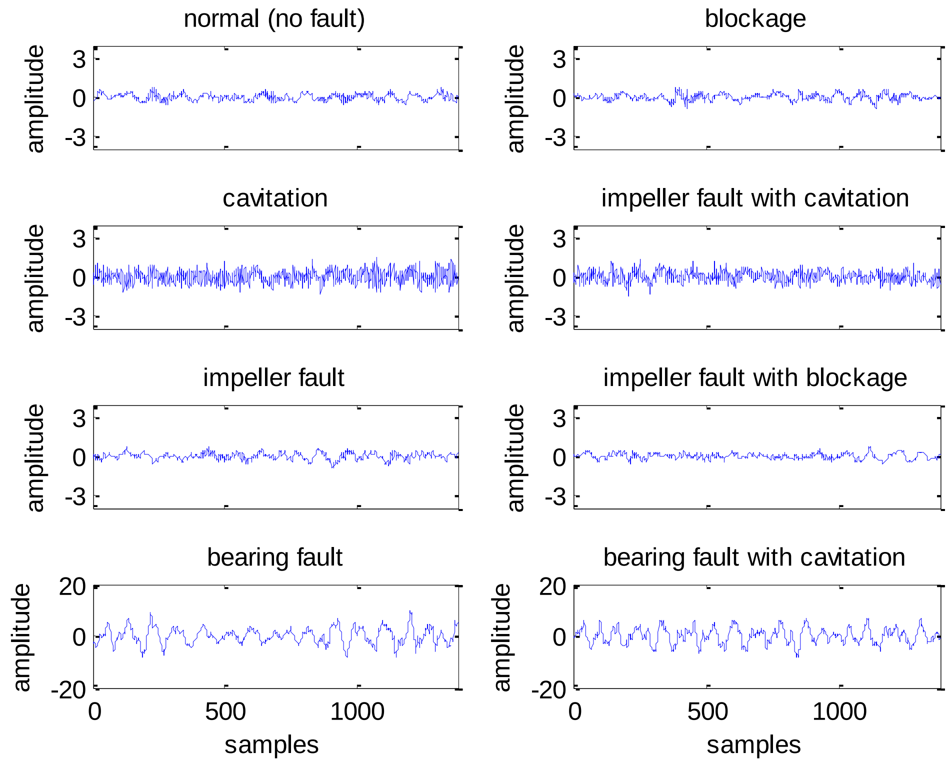


Figure 7.9 Time waveform of normal (no fault) and faulty components acquired from channel 3 (data set #15)

Table 7.3 depicts the eigenvalues of each principal component obtained using the sym4, sym8, and sym12 wavelet decomposition. Similar patterns are observed compared with the eigenvalues obtained from channel 1 and 2 where the last half of the PCs has eigenvalues close to zero.

For channel 3, all types of symlet wavelets produced 7 retained principal components in the model as shown in the Pareto chart in Figure 7.10. Therefore, the proposed method results in the three PCA models as described in Figure 7.11. The correlation matrix between the PCs and the features calculated using sym8 decomposition is shown in Figure 7.12. A high correlation value is exhibited for the first three principal components which represent the most essential information of the original features.

Table 7.3 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 3)

Principal component	Eigenvalue			Principal component	Eigenvalue		
	sym4	sym8	sym12		sym4	sym8	sym12
PC1	16.2	15.9	15.5	PC16	9E-04	1E-03	2E-03
PC2	4.3	5.0	6.0	PC17	3E-04	3E-04	3E-04
PC3	3.4	3.8	3.8	PC18	5E-05	2E-04	2E-04
PC4	2.0	1.5	1.2	PC19	1E-05	2E-05	3E-05
PC5	1.4	1.2	1.0	PC20	3E-06	5E-06	1E-05
PC6	0.9	0.8	0.7	PC21	2E-06	2E-06	2E-06
PC7	0.6	0.6	0.7	PC22	3E-07	4E-07	8E-07
PC8	0.5	0.5	0.4	PC23	6E-08	3E-07	6E-07
PC9	0.3	0.3	0.3	PC24	2E-08	2E-07	3E-07
PC10	0.3	0.2	0.2	PC25	1E-09	2E-09	5E-09
PC11	0.1	0.1	0.1	PC26	4E-10	1E-09	4E-09
PC12	0.1	0.1	0.1	PC27	1E-10	1E-10	3E-10
PC13	3E-02	3E-02	4E-02	PC28	4E-11	5E-11	1E-10
PC14	3E-02	3E-02	2E-02	PC29	3E-12	6E-12	1E-11
PC15	6E-03	6E-03	7E-03	PC30	2E-13	4E-13	7E-13

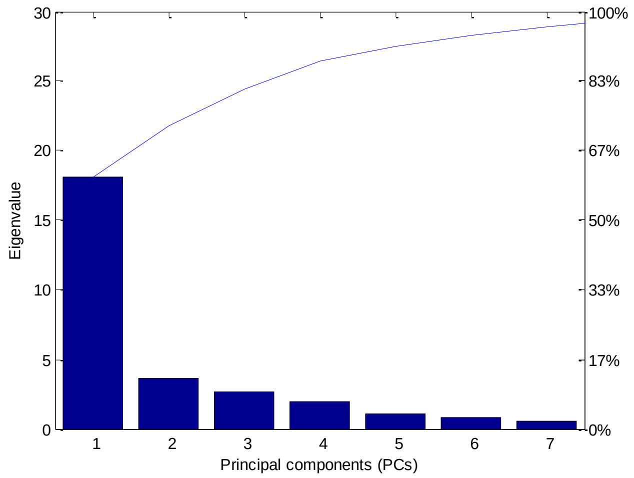


Figure 7.10 Seven PCs retained in the model by using sym4, sym8, and sym12 (based on 95% of the variance)

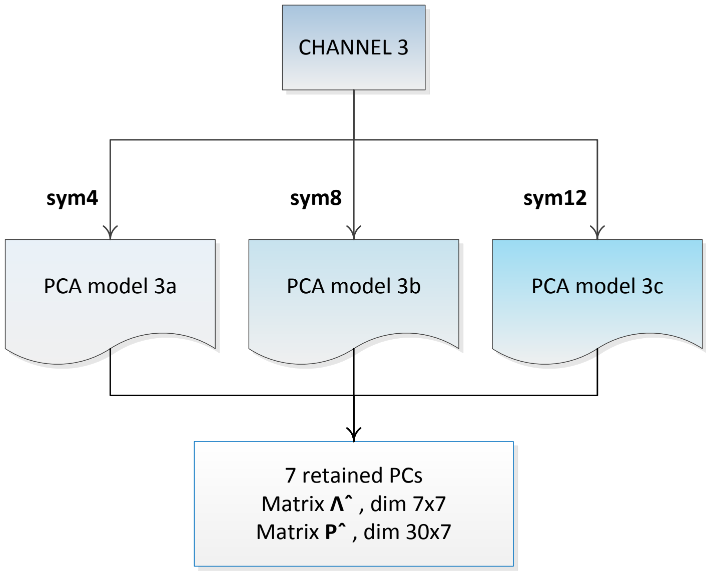


Figure 7.11 PCA models of channel 3

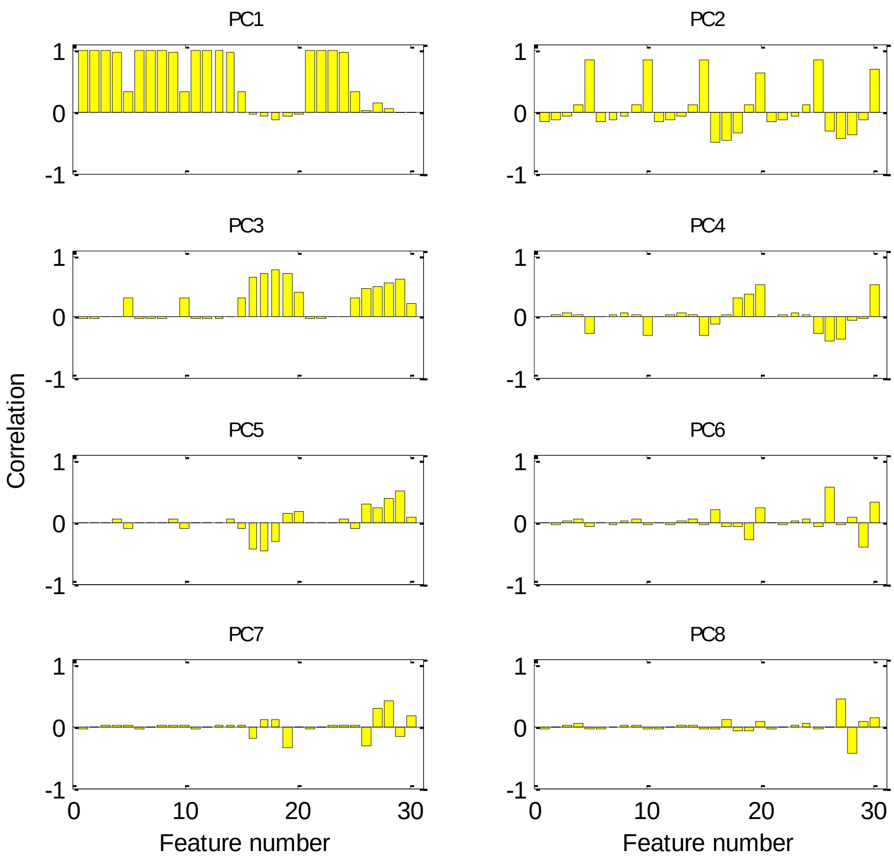


Figure 7.12 Correlation between principal components (PCs) and the features of channel 3 (using sym8 decomposition)

7.4 PCA Model Developed Using Vibration Signals Collected from Bearing Housing

The time waveform of the vibration signal from channel 4 is also plotted using data set #15 with 1377 samples as depicted in Figure 7.13.

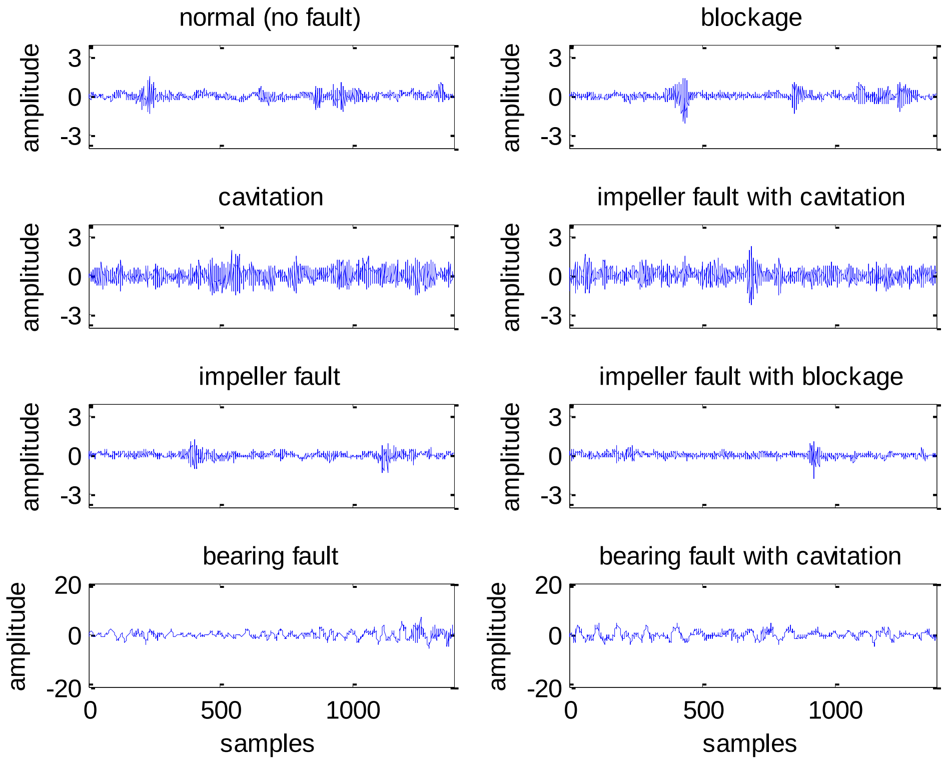


Figure 7.13 Time waveform of normal (no fault) and faulty components acquired from channel 4 (data set #15)

The eigenvalues of the principal components obtained using sym4, sym8, and sym12 are depicted in Table 7.4. More than half the number of principal components had eigenvalues close to zero; this is similar with the three previous channels. The Pareto chart in Figure 7.14, based on the 95% variance level to be accounted for in the model, shows that sym4 wavelet decomposition suggests 5 PCs, while the sym8 and sym12 wavelet decomposition recommends that 6 PCs be retained. The PCA models produced on this channel are depicted in Figure 7.15.

Figure 7.16 depicts the correlation matrix between PCs obtained from the vibration signal of channel 4 using sym8 wavelet decomposition and the features. The first three PCs also have high correlation value, the same as in the three previous channels.

Table 7.4 Eigenvalue of PCA model obtained using sym4, sym8, and sym12 (channel 4)

Principal component	Eigenvalue			Principal component	Eigenvalue		
	sym4	sym8	sym12		sym4	sym8	sym12
PC1	19.7	19.9	19.9	PC16	1E-03	1E-03	1E-03
PC2	5.3	5.0	4.7	PC17	2E-04	4E-04	9E-04
PC3	1.8	1.8	1.6	PC18	1E-04	2E-04	2E-04
PC4	1.2	1.0	1.1	PC19	3E-05	4E-05	4E-05
PC5	0.7	0.8	0.8	PC20	6E-06	9E-06	1E-05
PC6	0.4	0.5	0.7	PC21	2E-06	2E-06	2E-06
PC7	0.4	0.4	0.4	PC22	5E-07	7E-07	1E-06
PC8	0.2	0.2	0.3	PC23	1E-07	4E-07	6E-07
PC9	0.2	0.2	0.2	PC24	2E-08	1E-07	4E-07
PC10	0.1	0.1	0.2	PC25	6E-09	8E-09	1E-08
PC11	0.0	0.1	0.1	PC26	1E-09	5E-09	7E-09
PC12	0.0	0.0	0.0	PC27	3E-10	1E-09	2E-09
PC13	1E-02	1E-02	2E-02	PC28	3E-10	1E-10	6E-10
PC14	1E-02	8E-03	9E-03	PC29	1E-11	3E-11	6E-11
PC15	3E-03	4E-03	4E-03	PC30	2E-12	3E-12	4E-12

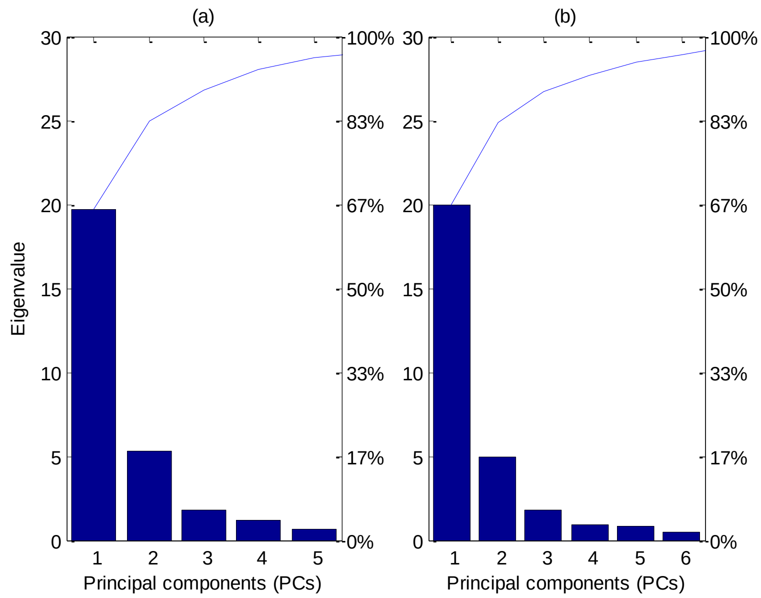


Figure 7.14 Number of principal components retained in the model based on 95% of the variance. (a) 5 PCs retained in the model by using sym4, (b) 6 PCs retained using sym8 and sym12

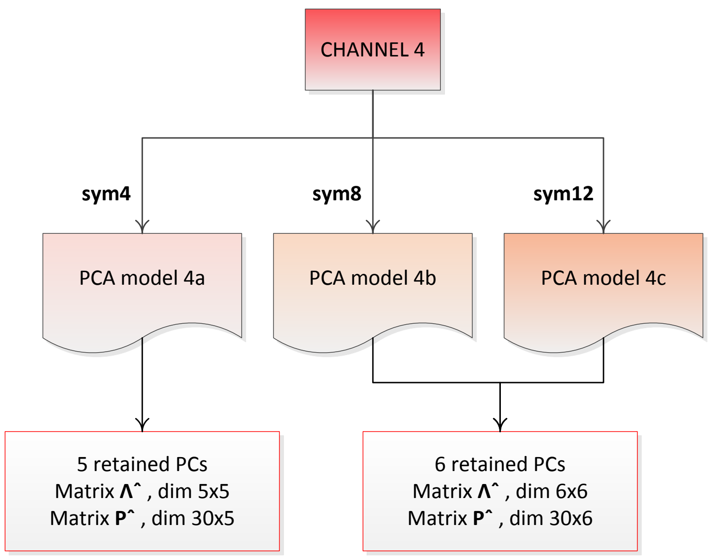


Figure 7.15 PCA models obtained from channel 4

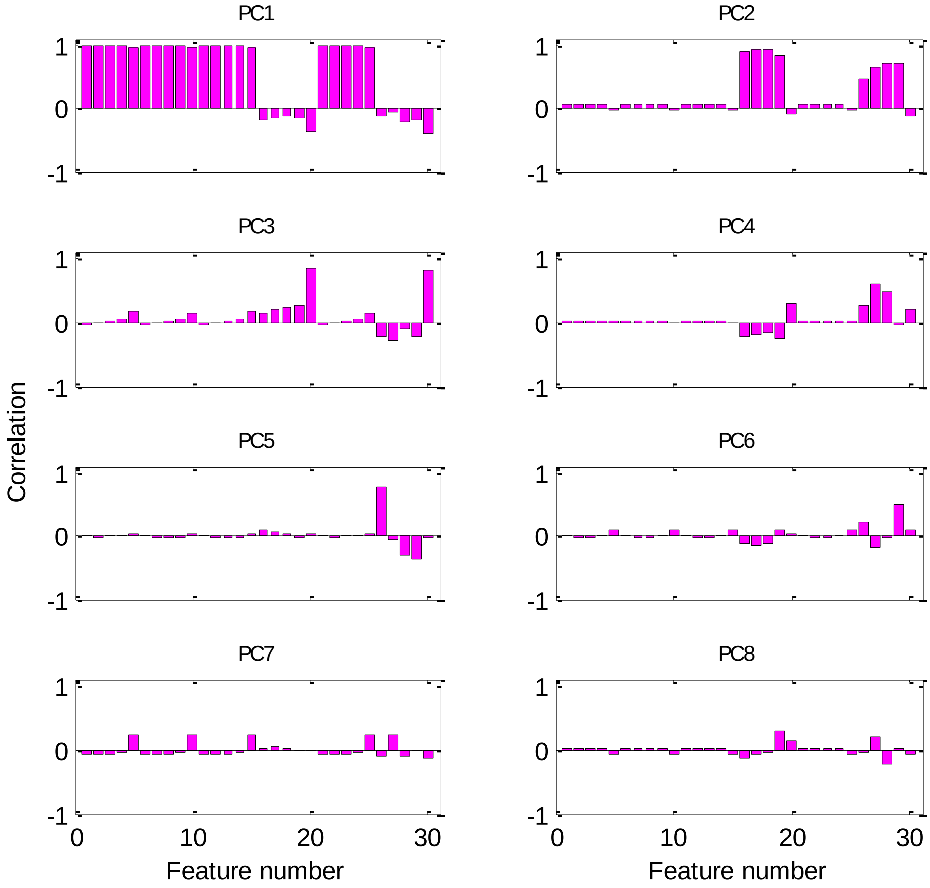


Figure 7.16 Correlation between principal components (PCs) and the features of channel 4 (using sym8 decomposition)

7.5 Summary of PCA Models Obtained from Channel 1 to 4

A summary of PCA models produced by the proposed method for all the channels is presented in Table 7.5. It can be concluded that the proposed method produced a total of 12 PCA models which were built using input features obtained from the cA parts of the sym4, sym8, and sym12 wavelet decomposition.

Table 7.5 PCA models obtained from channel 1 to 4

PCA model		Number of PCs retained (based on 95% variance)	Eigenvalue matrix (dim)	Loading matrix (dim)
Channel 1	**sym4**	7	7x7	30x7
	sym8	8	8x8	30x8
	sym12	8	8x8	30x8
Channel 2	**sym4**	7	7x7	30x7
	sym8	7	7x7	30x7
	sym12	7	7x7	30x7
Channel 3	**sym4**	7	7x7	30x7
	sym8	7	7x7	30x7
	sym12	7	7x7	30x7
Channel 4	**sym4**	5	5x5	30x5
	sym8	6	6x6	30x6
	sym12	6	6x6	30x6

7.6 Fault Detection and Diagnosis: Evaluation and Results

The fault detection and fault diagnosis was carried out using the 12 PCA models obtained from channel 1 to 4. For each channel, the proposed method produced three PCA models corresponding to the sym4, sym8, and sym12 wavelet decomposition. Each PCA model had different characteristics where the number of PCs retained, eigenvalue and loading matrix dimensions were varied. All of the PCA models were evaluated and tested using the normal and faulty condition schemes of the centrifugal pump.

The fault detection was carried out through the calculation of the T^2 and Q-statistic of the testing data from normal and faulty conditions. The performance of each PCA model in detecting various types of faults in the centrifugal pump was examined and tested by calculating the detection error which was obtained by dividing the number of false detection samples with the total samples. Figure 7.17 depicts the block diagram of the evaluation process of PCA models. The PCA model with the lowest identification error was selected as the best PCA model.

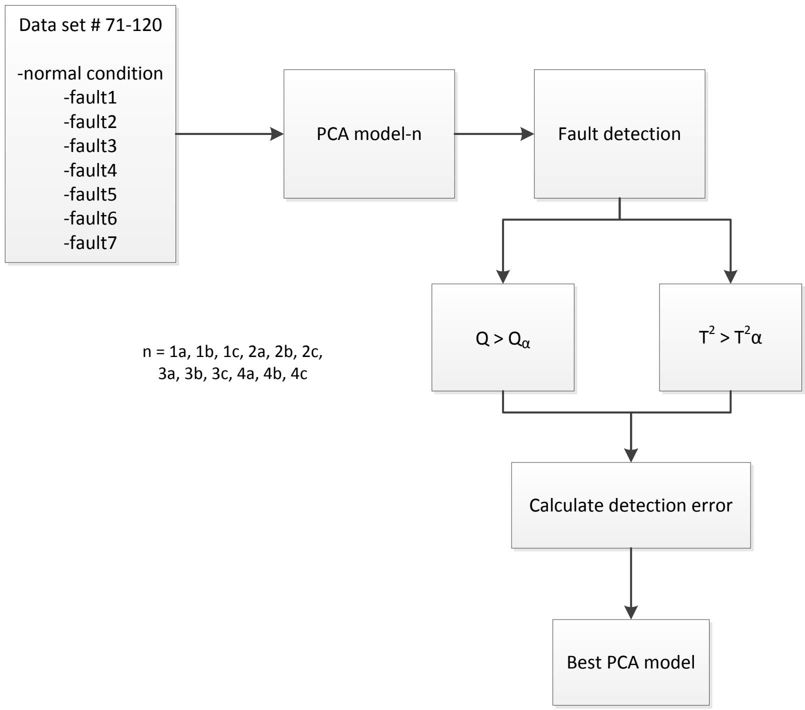


Figure 7.17 PCA models evaluation process

The fault classification was carried out through examining the plot of scores matrix on PC1, PC2, and PC3. The scores of each normal and faulty condition were plotted in the same graph so that the effect of clustering could be observed. The scores matrix obtained from all PCA models were evaluated and compared.

The continuation of the fault identification process applied the kNN rule with *k* (number of neighbors) and the number of PCs as the parameters. To select the best kNN rule, the scores matrix obtained from all PCA models was used as one of the parameters. The number of PCs used to build the kNN rule was varied from 1 to the maximum number of PCs for each model. Another parameter used was the number of *k* nearest neighbors. The *k* parameter was also varied from 1 to 30. The performance of the kNN rule obtained from the combination parameter number of PCs and *k* was calculated in terms of accuracy; in this case accuracy means the number of correctly classified examples per number of examples multiplied by 100. Figure 7.18 shows the block diagram of the evaluation process of fault classification and fault identification.

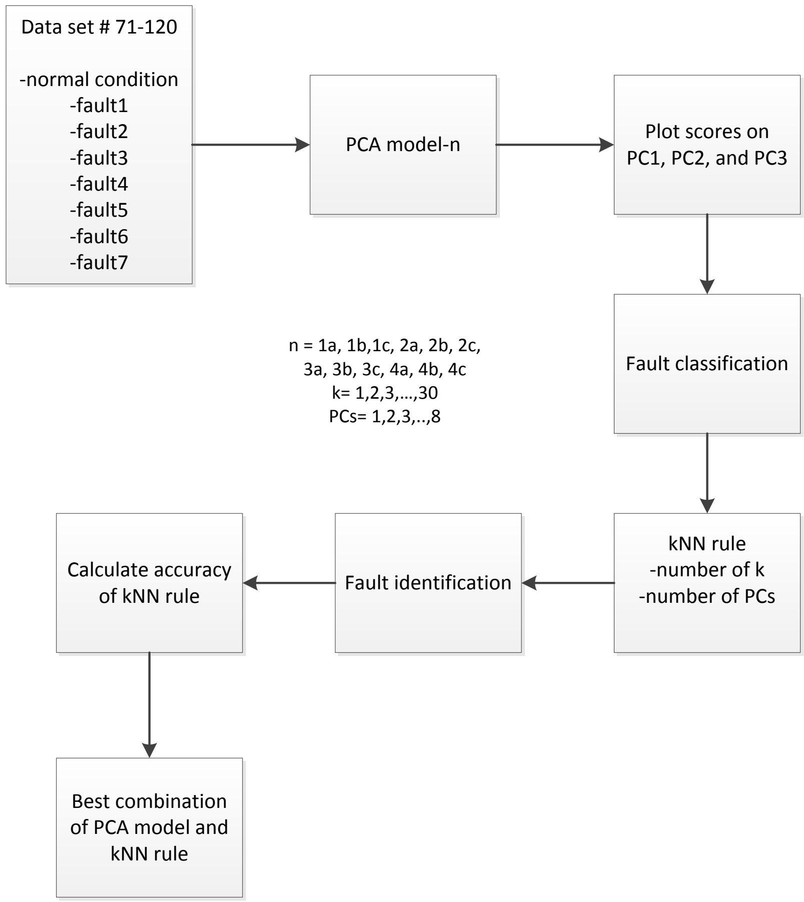


Figure 7.18 Fault classification and fault identification evaluation process

7.6.1 Fault Diagnosis Results Using PCA Model 1

In order to detect when a fault occurred in the centrifugal pump test rig, the algorithm explained in Figure 5.14 was carried out. The training data obtained from data set #1 to 70 of the normal condition (no fault) was pre-processed using the wavelet decomposition that employed the sym4, sym8, and sym12 wavelets. There were three feature matrices obtained from the pre-processed step where each matrix was generated using the sym4, sym8, and sym12 process respectively. These three feature matrices were then used to build the PCA models; therefore channel 1 produced PCA model 1a, model 1b and, model 1c where they were constructed using sym4, sym8, and sym12 respectively. The details of each PCA model was given in section 7.1.

The fault detection analysis was performed on the testing data from normal and faulty conditions (data set #71-120). There were 50 samples for each of the normal and faulty conditions tested in order to detect the presence of a fault. The T^2-statistic and Q-statistics were used to assess the fault detection performance using all the PCA models obtained in channel 1 as shown in Figure 7.19. The graph of the T^2-statistic consisted of the T^2 value calculated from normal and faulty condition and the threshold lines for each PCA model. Since the PCA model 1b and 1c had the same number of retained PCs as described in Figure 7.3, they shared the same threshold line (horizontal line) which was shown as a green horizontal line in Figure 7.19 while the blue horizontal line was the threshold line for PCA model 1a.

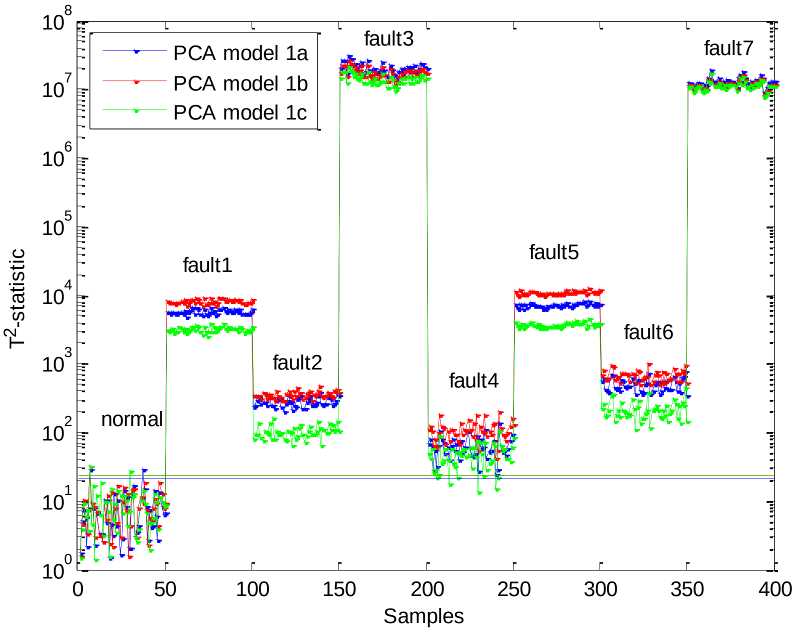


Figure 7.19 T^2 chart for normal and all faulty conditions obtained using PCA model 1a, model 1b, and model 1c

In Figure 7.19, the y-axis was in logarithmic scale which represented the T^2-value of testing data of normal and faulty condition. The T^2 value obtained from each PCA model was plotted in the order of normal, fault1, fault2, fault3,…, fault7. Since there were 50 samples for each set, there were 400 samples in total. The T^2 value obtained from PCA models 1a, 1b and 1c were plotted in blue, red, and green line respectively. The T^2 values which lay below the threshold line indicated the normal condition, as suggested in Equation 4.25, otherwise the fault condition occurred.

Figure 7.19 shows that in general all PCA models from channel 1 were able to detect the normal and faulty condition with some exception in several samples of the normal and fault4 condition for which the threshold was exceeded. Several misdetections were also observed in normal condition for PCA model 1a and 1c and in fault4 for PCA model 1c.

Figure 7.20 shows, the plot of Q value of normal and faulty conditions along with the threshold line for each of PCA model as suggested in Equation 4.28.

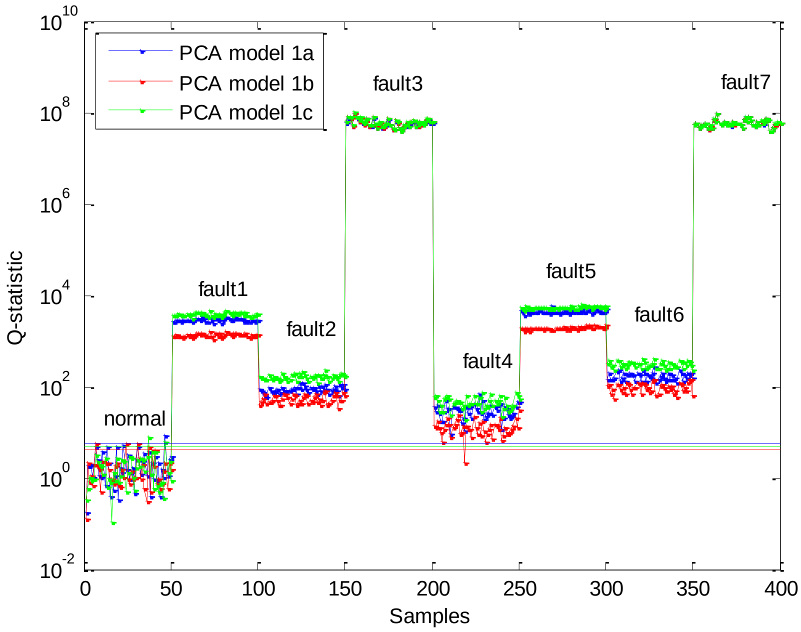


Figure 7.20 Q chart for normal and all faulty conditions obtained using PCA model 1a, model 1b, and model 1c

It is shown that, similar with the T^2 result, in general all PCA models were able to correctly detect the normal and faulty condition of components. In this case, several misdetections were observed in the normal condition for all PCA models and in fault4 for the PCA model 1b. For comparison, Table 7.6 summarized the misdetection rate for all PCA models.

Table 7.6 Comparison of fault detection accuracy of PCA model in channel1

	Misdetection rate %		
	PCA model 1a	PCA model 1b	PCA model 1c
T^2	0.5	0.3	2.5
Q	0.3	1.3	0.5
Weighted average	0.5	**0.3**	2.4

From Table 7.6, it is shown that the PCA model 1b obtained the lowest weighted average misdetection rate which indicates the highest fault detection accuracy. The PCA model 1b obtained the weighted average misdetection rate of 0.3 compared to 0.5 and 2.4 for PCA model 1a and 1c respectively. The weighted average was calculated by considering the total variance retained in the principal component subspace for T^2 (95%) and the total variance retained in the residual subspace for Q (5%). For instance, the weighted average of misdetection for PCA model 1b was calculated by $0.3\times 0.95+1.3\times 0.05=0.3$. The PCA model 1b was then selected as the best PCA model obtained from channel 1 in detecting the fault conditions.

In Figure 7.21 to Figure 7.23, the scores obtained from PCA model 1a, 1b, and 1c of normal and all faulty conditions were plotted on PC1, PC2, and PC3 to reveal the clustering effect of each normal and faulty condition. For each figure, exhibit (b) shows the zoom in of the circled area in the exhibit (a). It is observed that PCA model 1a and PCA model 1b shows clear clustering effect as depicted in Figure 7.21 and Figure 7.22 respectively, although some overlaps occur in both models. However, the overlaps can be reduced by considering more dimensions (PCs). Meanwhile in the PCA model 1c the clustering effect is not clearly visible particularly for the normal condition, fault2, fault4, and fault6. This result is consistent with the misdetection rate listed in Table 7.6 where the PCA models 1a and 1b have the higher identification accuracy compared with PCA model 1c.

Figure 7.24 to Figure 7.26 depict the identification accuracy for each of the PCA models with various combinations of the number of PCs and k (neighbors). The number of PCs included in the calculation may be different for PCA model 1a, 1b, and 1c. This is in accordance with the number of PCs in each model as described in the previous section.

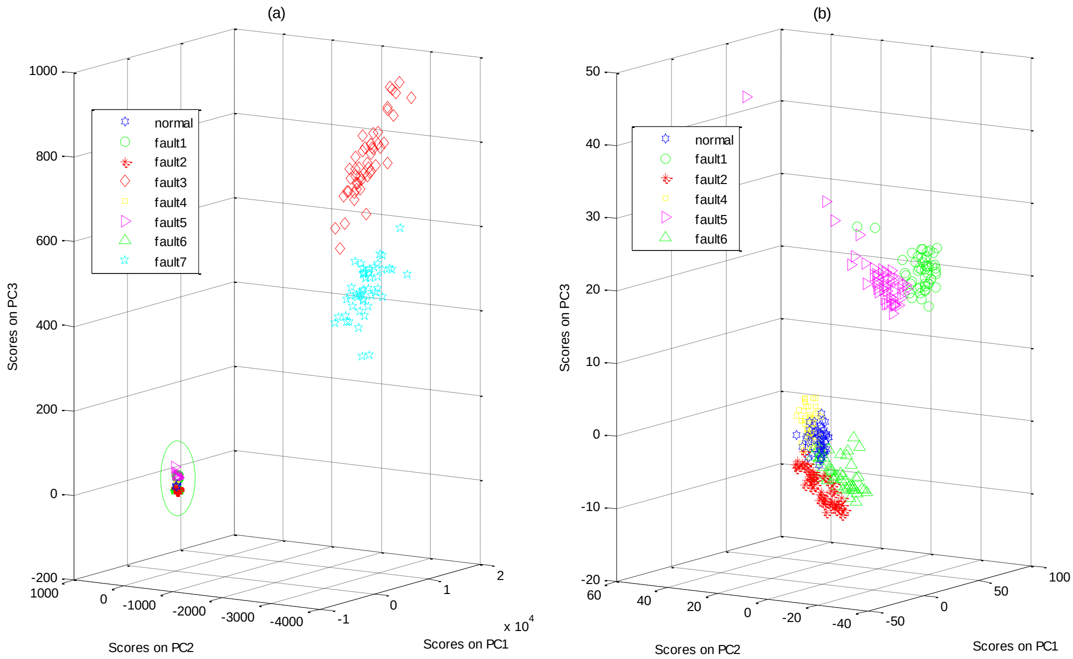


Figure 7.21 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1a. Note that exhibit (b) zooms into the green ellipse area in exhibit (a)

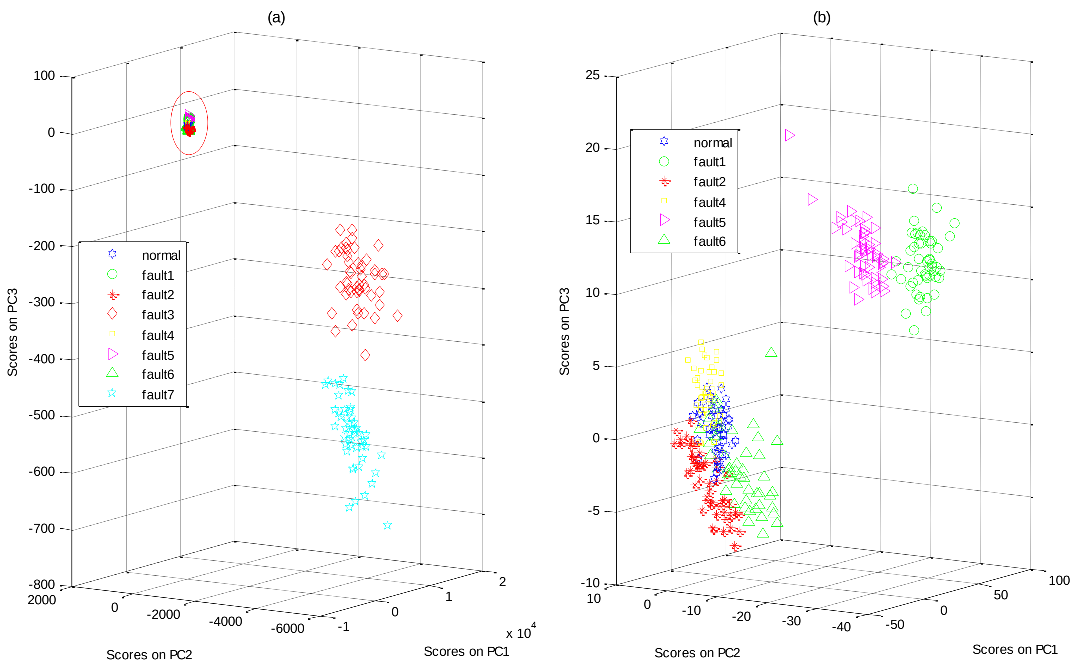


Figure 7.22 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1b. Note that exhibit (b) zooms into the red ellipse area in exhibit (a)

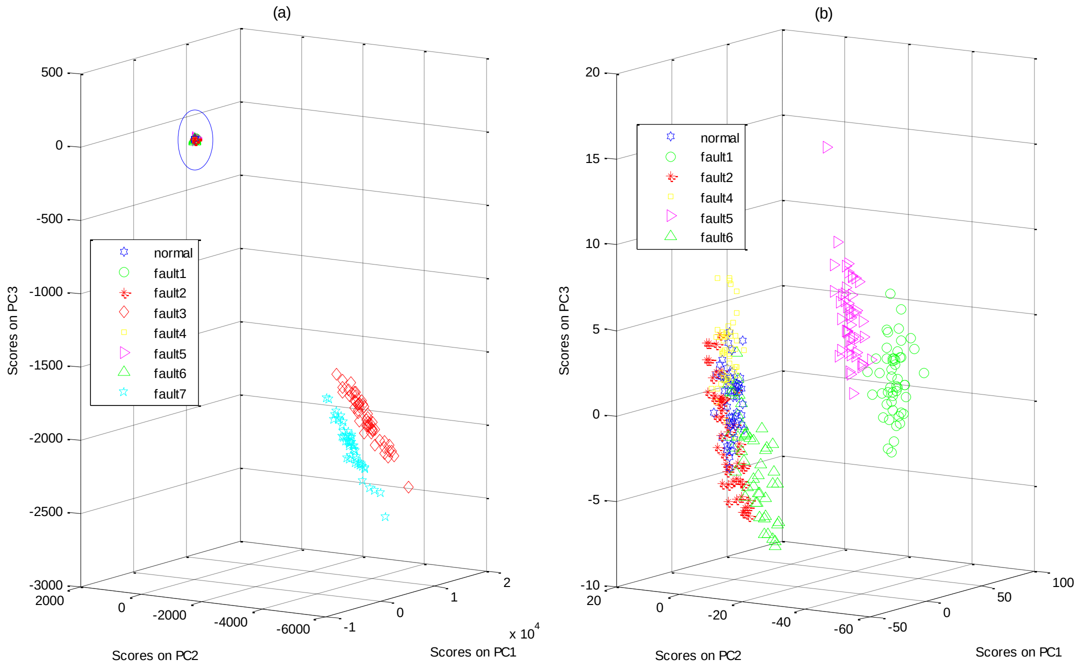


Figure 7.23 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 1c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a).

The identification accuracy of PCA model 1a is shown in Figure 7.24. The *x*-axis represents the number of neighbors *k* which is varied from 1 to 30, while the *y*-axis represents the identification accuracy in percentage. There are seven lines with different markers and colours which denote the identification accuracy of the PCA model employing from 1 PC to 7 PCs. As expected, the model with more numbers of PCs provides the higher identification accuracy. The model with 1 PC gives the worst identification accuracy from 47.0% to 55.7%, while the model with 7 PCs has the highest identification accuracy from 96.7% to 99.5%. This result agrees with the PCA theory where the amount of information corresponds to the number of PCs retained in the PCA model

Figure 7.25 shows the identification accuracies of PCA model 1b. In this case, 1 to 8 PCs were employed; the number of PCs in this model is as described in Figure 7.3. The result has the same trend as the results obtained in the PCA model 1a where the model which employed more number of PCs gives the higher accuracies; 47.2% to 56.0% for PCA model with 1PC and 98.0% to 99.7% for PCA model with 8 PCs.

Meanwhile, the identification accuracies of PCA model 1c is depicted in Figure 7.26, which consists of the 8 results that correspond to 1PC to 8PCs that are included in the model. This model also shows identification accuracy from 44.7% to 54.2% for the model with 1PC and 94.5% to 97.7% for the model with 8PCs.

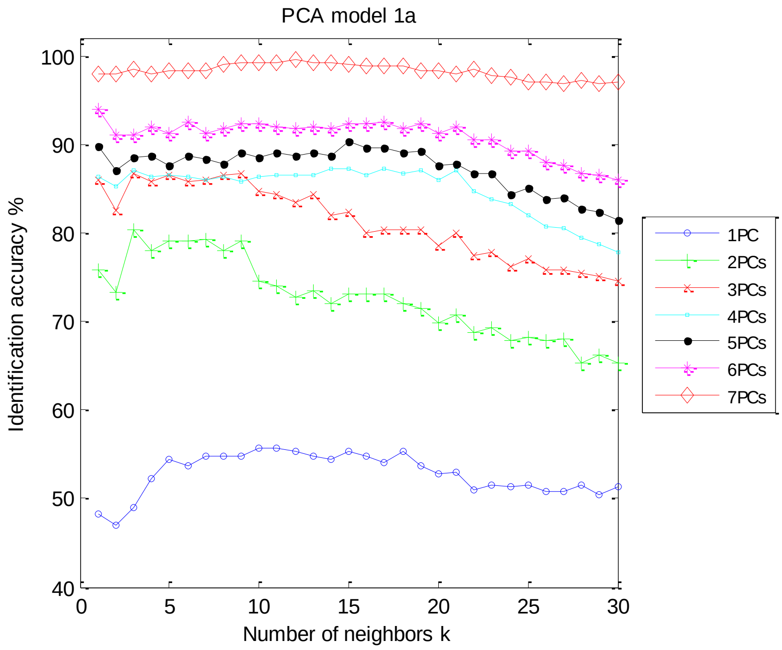


Figure 7.24 Identification accuracy comparison of the PCA model 1a using 1 to 7 PCs.

From Figure 7.24 to Figure 7.26, it can be seen that in general, the highest identification accuracies obtained by PCA model 1a and 1b is higher than those for PCA model 1c. The average identification accuracies of PCA model 1a with 7 PCs is around 98.2% and PCA model 1b with 8 PCs is 99.2%, whilst, the average identification accuracies of PCA model 1c with 8 PCs is 96.3%. The results confirm the findings in the previous discussion where from Table 7.6 and from Figure 7.21 to Figure 7.23, PCA model 1a and 1b were found to have better fault detection and classification performance than PCA model 1c.

It is observed from Figure 7.24 to Figure 7.26 that the identification accuracies depend on the number of neighborhood parameters, *k*. There is a downward trend in the level of accuracy with the increase in the number of parameters *k* which indicates that the model is sensitive on the changes in the number of *k*. However, this is not the case for the PCA model with more PCs included in the model. For example, the PCA model 1a with 7 PCs, PCA model 1b with 8 PCs and PCA model 1c with 8 PCs provide relatively the same identification accuracies as the parameter *k* varies from 1 to 30. It is because the more PCs included in the model reduces the overlap between the classes therefore they may avoid misidentification by increasing the number of neighborhood parameter *k*.

Theoretically, the improvement of identification accuracy becomes less as more PCs are added into the model because less information is held by high order PCs. However, in some instances the higher improvement of accuracy may occur on addition of high order PCs into model. These occurrence can be observed in model 1a where the addition of the 7th PC increases the accuracy more than when the 6th PC is added into the model. This can be due to the non-stationary behavior of the vibration signal which sometimes occurs on the centrifugal pump.

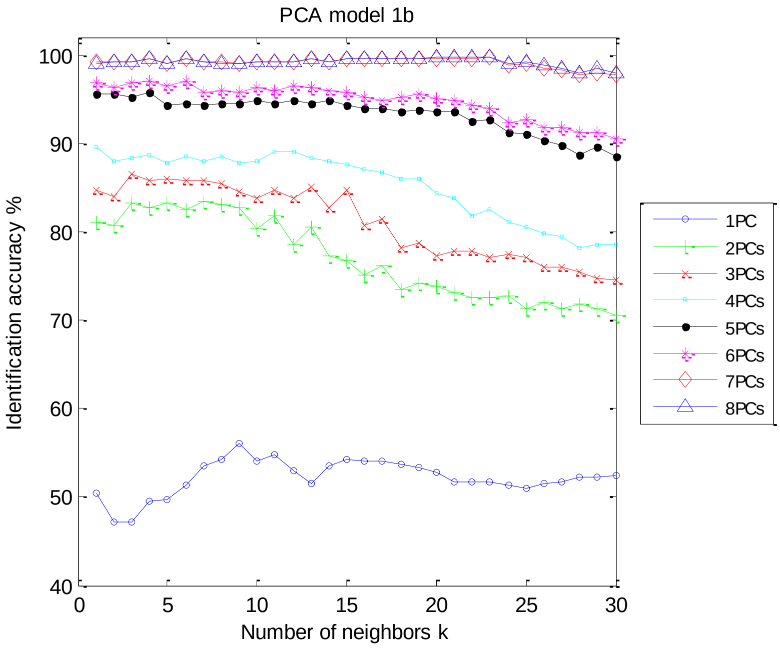


Figure 7.25 Identification accuracy comparison of the PCA model 1b using 1 to 8 PCs.

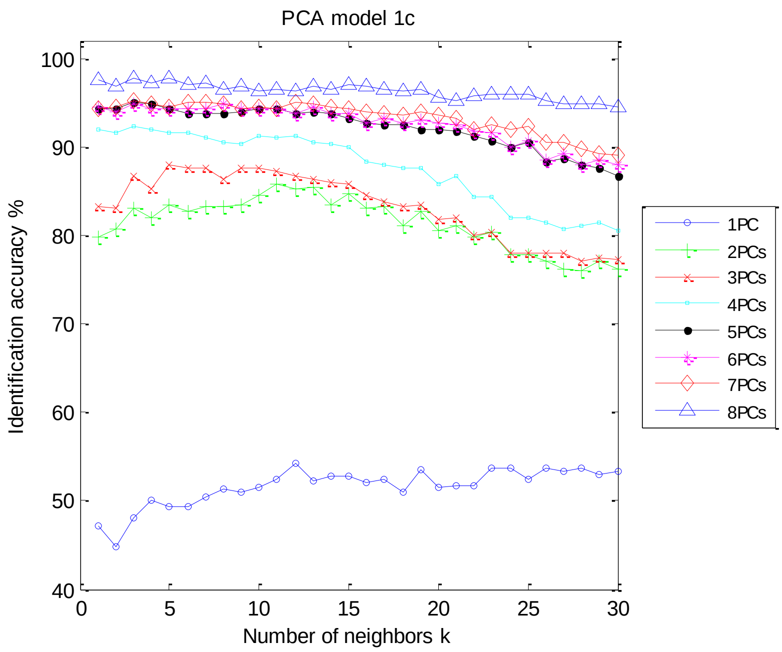


Figure 7.26 Identification accuracy comparison of the PCA model 1c using 1 to 8 PCs.

In view of the above result and discussion, it is appropriate to conclude that PCA model 1b achieves the higher performance in detection, classification and identification of faults in a centrifugal pump compared to the other two models. From the mechanical point of view, channel 1 where the acceleration sensor was mounted at the pump inlet, gives an excellent clustering effect for fault1 (cavitation) for all PCA models as depicted in Figure 7.21 to Figure 7.23. This is because the sensor is closest to the location of cavitation; hence the change of fluid condition (i.e., cavitation) in the pump inlet can be easily identified.

7.6.2 Fault Diagnosis Results Using PCA Model 2

The procedures used to evaluate the T^2- and Q-statistic from channel 2 data were similar to that used for channel 1. There were three PCA models namely, PCA model 2a, model 2b, and model 2c obtained from channel 2. The result of the T^2- and Q-statistic is shown in Figure 7.27 and Figure 7.28 respectively.

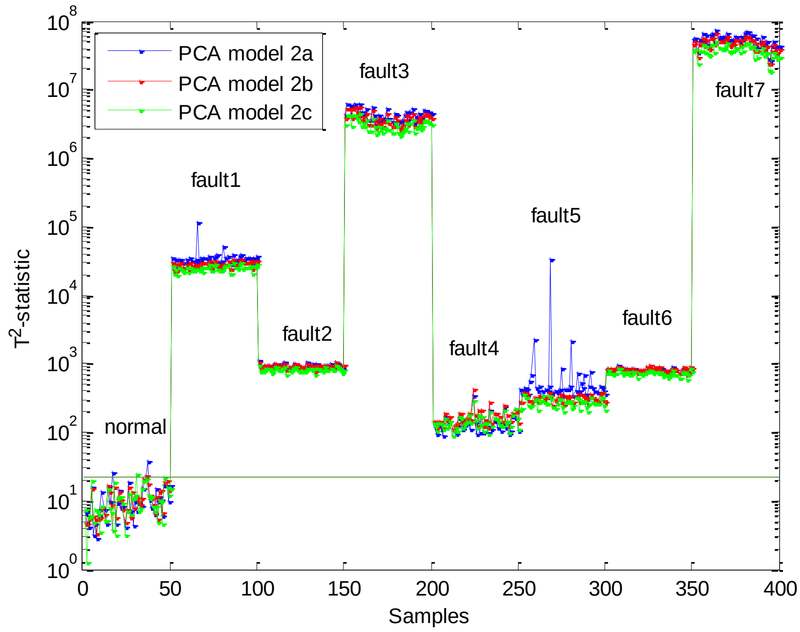


Figure 7.27 T^2 chart for normal and all faulty conditions obtained using PCA model 2a, model 2b, and model 2c

Figure 7.27 shows the T^2-statistic calculated from PCA model 2a, 2b, and 2c compared to the threshold line. There was only one threshold line in this plot, since the PCA models 2a, 2b and 2c share the same threshold line; this is the consequence of the PCA model constructed in section 7.2 where all channel 2 PCA models retained 7 PCs. Therefore according to Equation 4.24 there was only one threshold for all models which is represented as a green horizontal line in Figure 7.27.

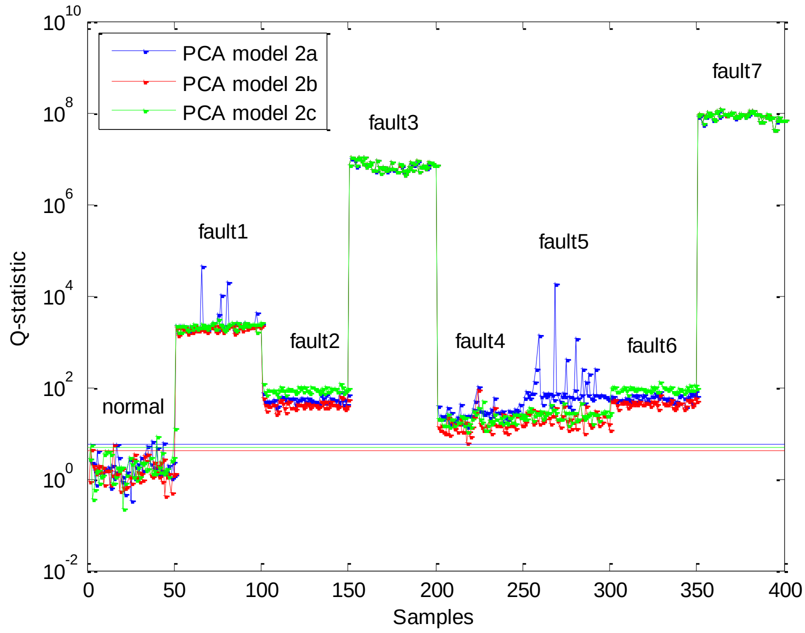


Figure 7.28 *Q* chart for normal and all faulty conditions obtained using PCA model 2a, model 2b, and model 2c

Figure 7.27 shows that all PCA models made good predictions for all fault conditions where all values of T^2 were above the threshold line. However, several samples in the normal condition exceeded the threshold which indicated misdetection. The *Q*-statistic, as depicted in Figure 7.28, shows a similar prediction with the one in the T^2. The *Q*-statistic was able to detect all types of fault with 100% accuracy as indicated by the fault values that were all above the threshold. Yet, several samples in the normal condition were observed exceeding the threshold which denoted misdetection.

Overall, the misdetection rate of the T^2- and *Q*-statistic is listed in Table 7.7 along with its weighted average for all PCA models. The results show that PCA model 2b obtained the lowest misdetection rate (0.3%) compared to the others.

It was found that the lowest misdetection rate of channel 1 and 2 was obtained from the PCA model constructed from the same wavelet type (sym8). The finding was based on the comparison of weighted average misdetection rate from all PCA models in channel 1 and channel 2.

Table 7.7 Comparison of fault detection accuracy of PCA model in channel 2

	Misdetection rate %		
	PCA model 2a	PCA model 2b	PCA model 2c
T^2	0.8	0.3	0.5
Q	0.8	0.5	0.8
Weighted average	0.8	**0.3**	0.5

The three dimensional plot of the scores on PC1, PC2 and PC3 for PCA model 2a, 2b, and 2c is depicted in Figure 7.29 to Figure 7.31. In general, the clustering effect achieved from all PCA models was visible. In Figure 7.29, the PCA model 2a revealed a good separation among classes except between fault2 (impeller fault) and fault6 (impeller fault and blockage) where several overlaps occurred.

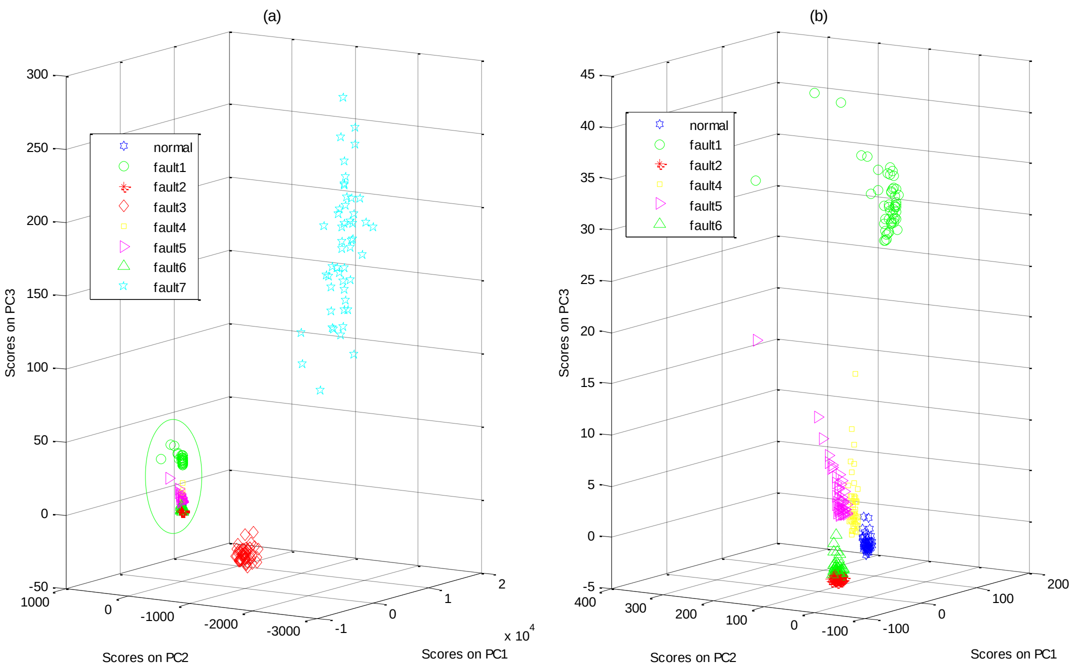


Figure 7.29 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2a. Note that exhibit (b) zooms into the green ellipse area in exhibit (a)

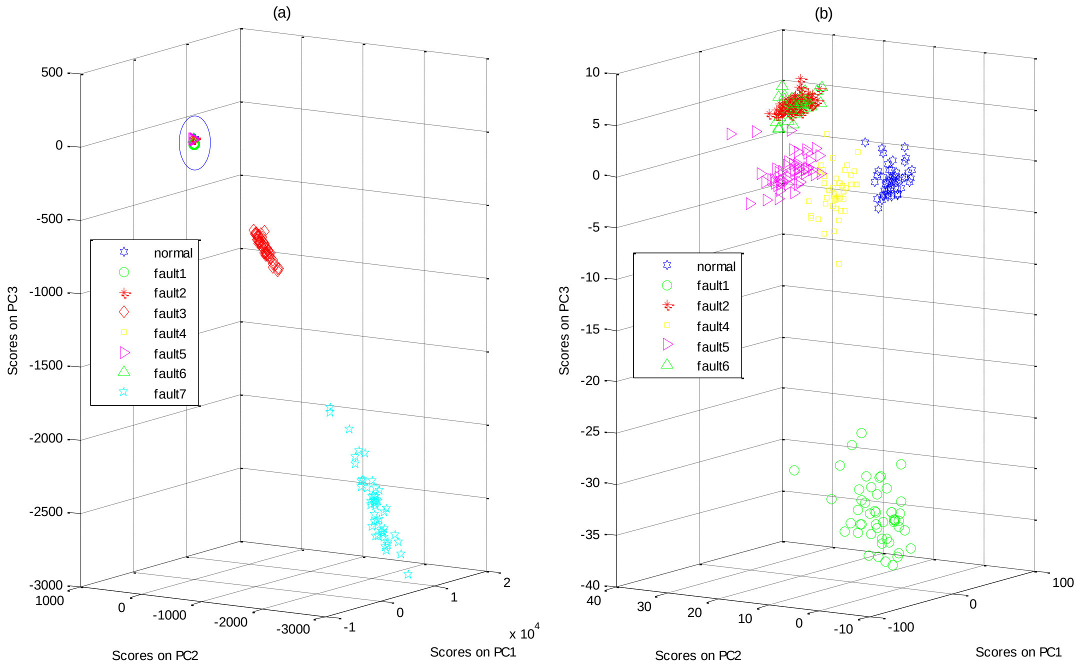


Figure 7.30 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2b. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a)

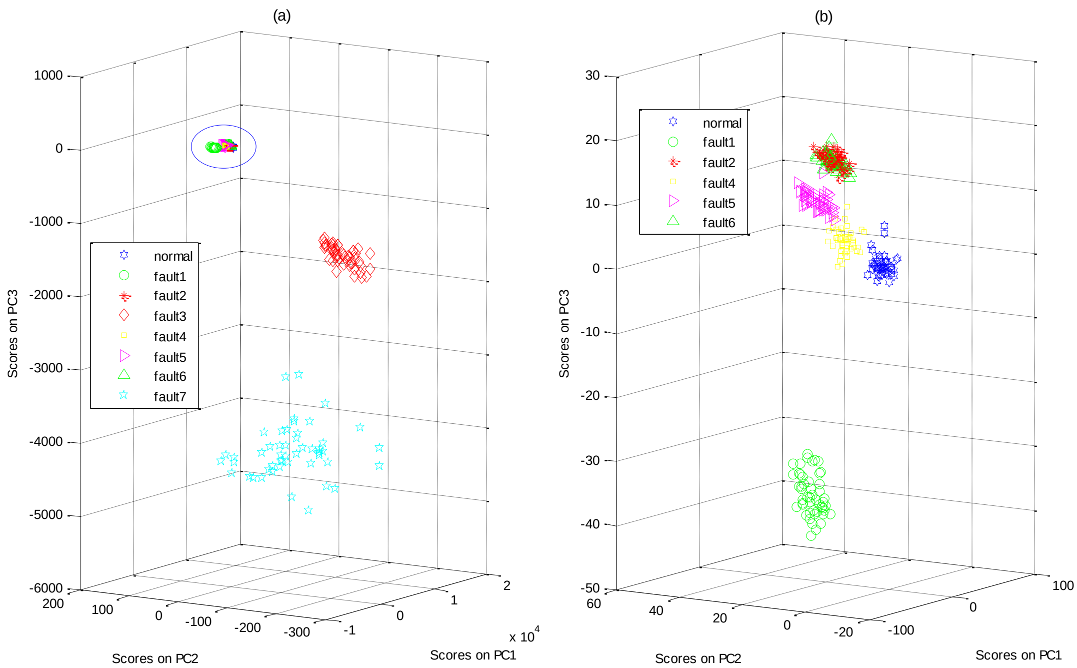


Figure 7.31 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 2c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a)

Figure 7.30 shows the clustering effect produced by PCA model 2b. It was observed that some overlap occurred between the class of fault2 and fault6. The same overlap was also observed in the PCA model 2c as shown in Figure 7.31. In general, the performance of all PCA models obtained from channel 2 in classifying the normal and fault condition is relatively the same. As can be seen from Figure 7.29 to Figure 7.31, all classes can be completely separated except for fault2 and fault6. The relatively similar classification performance among the models confirmed the detection accuracies as summarised in Table 7.7 where the misdetection rate among the models was very similar.

The identification accuracies of PCA models in channel 2 are depicted in Figure 7.32 to Figure 7.34. It was observed from Figure 7.32 that the PCA model 2a with 7 PCs achieved the highest identification accuracy from 95.7% to 98.7% as expected where the model with 1 PC obtained the lowest accuracy from 79.7% to 84.5%. The effect of the number of neighbors, *k*, in the model does not seem to indicate a trend that can be observed. However, in the model with 7 PCs, increasing the number *k* showed no significant change in the accuracy.

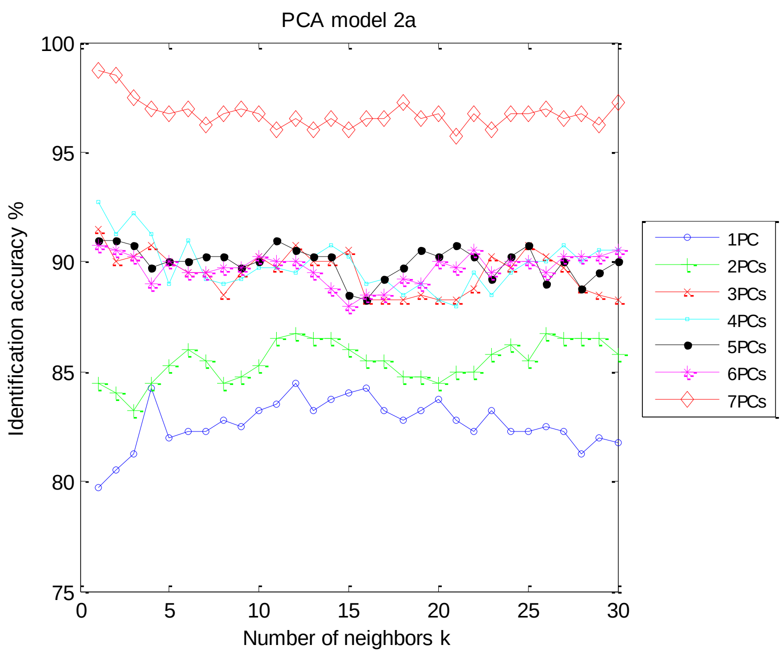


Figure 7.32 Identification accuracy comparison of the PCA model 2a using 1 to 7 PCs.

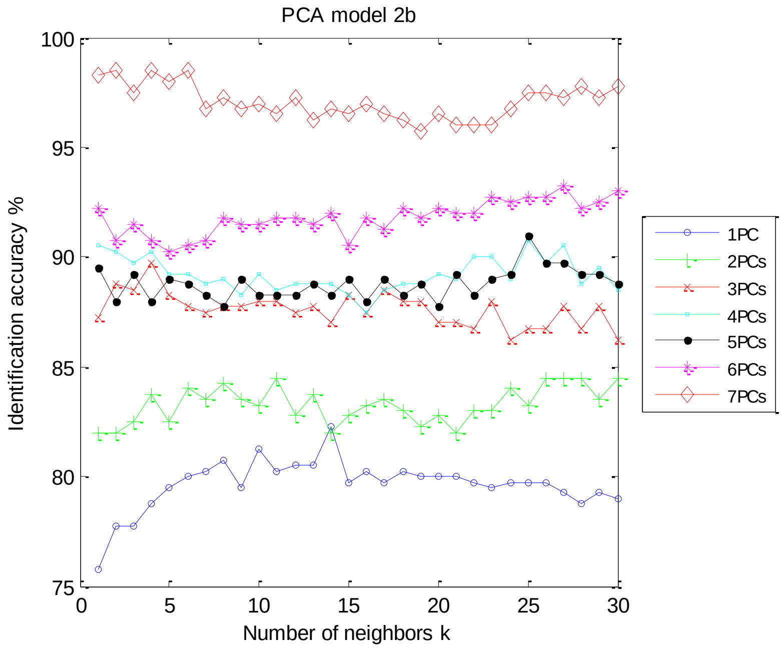


Figure 7.33 Identification accuracy comparison of the PCA model 2b using 1 to 7 PCs.

Employing 1 to 7 PCs, the identification accuracy of PCA model 2b is given in Figure 7.33. It is shown clearly that the model with 7 PCs obtained the highest identification accuracy from 95.7% to 98.5% while the model with 1 PC resulted in the lowest accuracy from 75.7% to 82.2%. The result of the model with 7 PCs was similar to that of PCA model 2a, which indicated that the highest performance of PCA model 2a and 2b was relatively the same.

The similar pattern was also observed for PCA model 2c as shown in Figure 7.34. The highest identification accuracy was achieved by the model with 7 PCs from 95.2% to 97.7% while the model with 1 PC achieved the lowest accuracy from 76.2% to 80.7%. In general, for all PCA models, it could be inferred that the model was not sensitive on the changes in the number of *k*.

It was found that all PCA models achieved the highest identification accuracy by employing all of the PCs available in the model. This is not a surprising result since employing all of the PCs in the model provided the maximum information available in the model for fault identification process. On the contrary, employing only 1 PC resulted in the lowest identification accuracy since the model provided less information to the fault identification process.

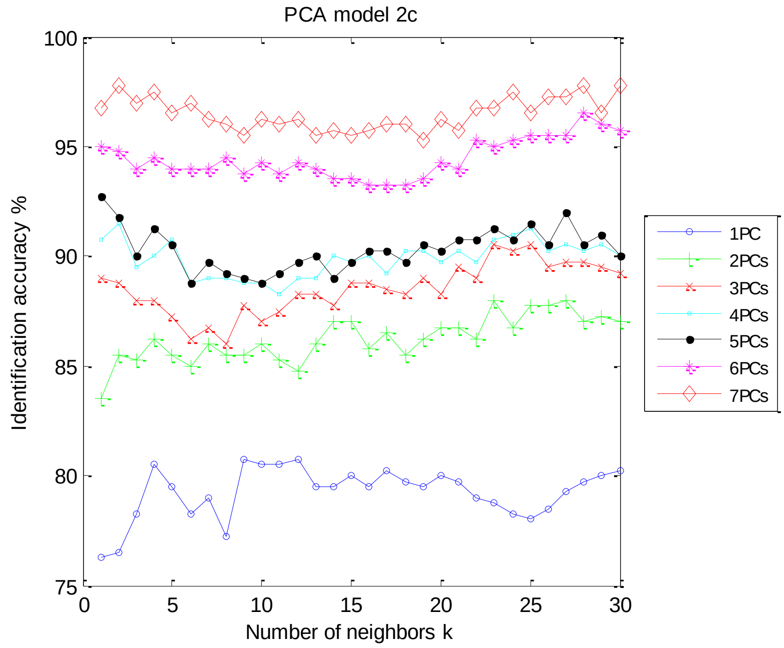


Figure 7.34 Identification accuracy comparison of the PCA model 2c using 1 to 7 PCs.

From the discussion above, all PCA models in channel 2 produced relatively the same performance in terms of fault detection, classification and identification. Although they gave relatively the same performance, Table 7.7 indicated that PCA model 2b (constructed using sym8) achieved the lowest misdetection rate, that was 0.3%, compared to the other two models of 0.8% and 0.5%. The classification result as given from Figure 7.29 to Figure 7.31 pointed out that all PCA models obtained relatively the same performance which was indicated by the overlap between the same classes. Similarly, the average identification accuracy showed that all PCA models obtained relatively the same performance which was 96.7%, 97.1% and 96.5% for 1a, 1b and 1c respectively.

Considering the above results analysis, one may conclude that overall the PCA model 2b achieves the higher performance in fault diagnosis of the centrifugal pump compared to the other two models. Recalling the results of channel 1, it was found that the PCA model constructed using sym8, i.e. PCA model 1b, provided the higher fault diagnosis performance than the others. Therefore, the PCA model constructed using sym8 was found to be the most suitable approach applied to both channels 1 and 2.

From the mechanical point of view, in channel 2 the accelerometer was mounted on the pump's volute which gave a good reading of the vibration signal from the hydraulic related impeller fault, cavitation, and blockage conditions. This caused the clustering effect as illustrated from Figure 7.29 to Figure 7.31 to be more visible compared to channel 1 except in the case of fault2 and fault6. An overlap was observed between fault2 (impeller fault) and fault6 (impeller fault-blockage) which indicated that the proposed method was not able to distinguish the uniqueness of vibration signals between fault2 and fault6 by using only the first three PCs. However, the addition of more PCs would be expected to reduce the overlap as demonstrated in the identification accuracy results (Figure 7.32 to Figure 7.34).

7.6.3 Fault Diagnosis Results Using PCA Model 3

The proposed method applied to channel 3 data produced T^2- and Q-statistic charts as depicted in Figure 7.35 and Figure 7.36 respectively. In the T^2 chart, there is only one threshold line (horizontal green line) since all PCA models in this channel retained the same number of PCs i.e., 7 PCs. Therefore, one threshold was applied for all PCA models. It can be seen from Figure 7.35 that the T^2-statistic calculated using all PCA models was able to detect all the fault cases with 100% accuracy. It was obvious that all types of fault samples exceeded the threshold line (fault occurred). However, misdetection occurred for the normal condition case where several sample points exceeded the threshold. The misdetection rate for all PCA models was listed in Table 7.8.

The performance of the Q-statistic for all PCA models is shown in Figure 7.36. Overall, the Q-statistic obtained from PCA model 3b and 3c showed an excellent performance (100% correct detection) in detecting fault cases where all associated samples lay above the threshold. Meanwhile, one sample from fault4 of the PCA model 3a crossed the threshold which indicated a false alarm.

The situation was slightly different for the normal condition samples. All the models made false detections several times as the *Q*-statistic values exceeded the threshold. This could be due to the non-stationary behaviour of the vibration signal acquired from channel 3 (pump's outlet). Table 7.8 summarized the misdetection rate of the *Q*-statistic for all models.

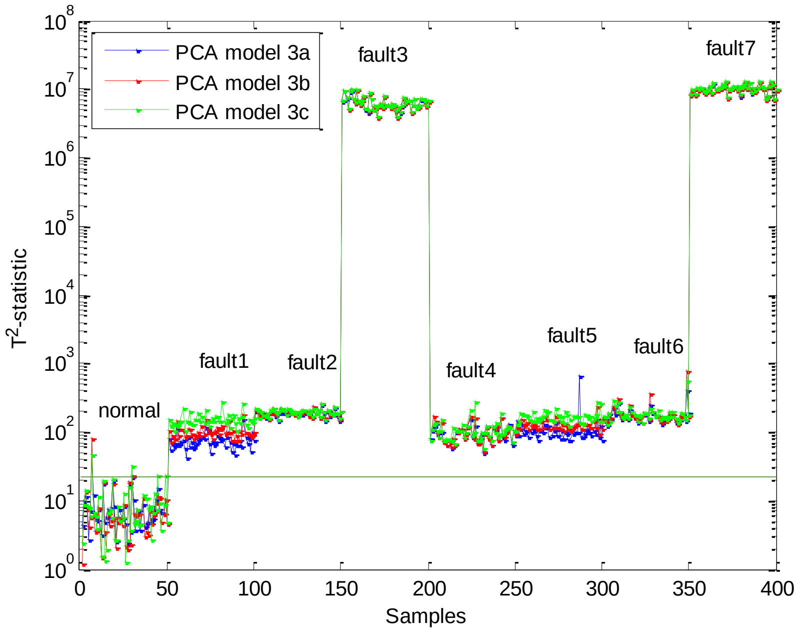


Figure 7.35 T^2 chart for normal and all faulty conditions obtained using PCA model 3a, model 3b, and model 3c

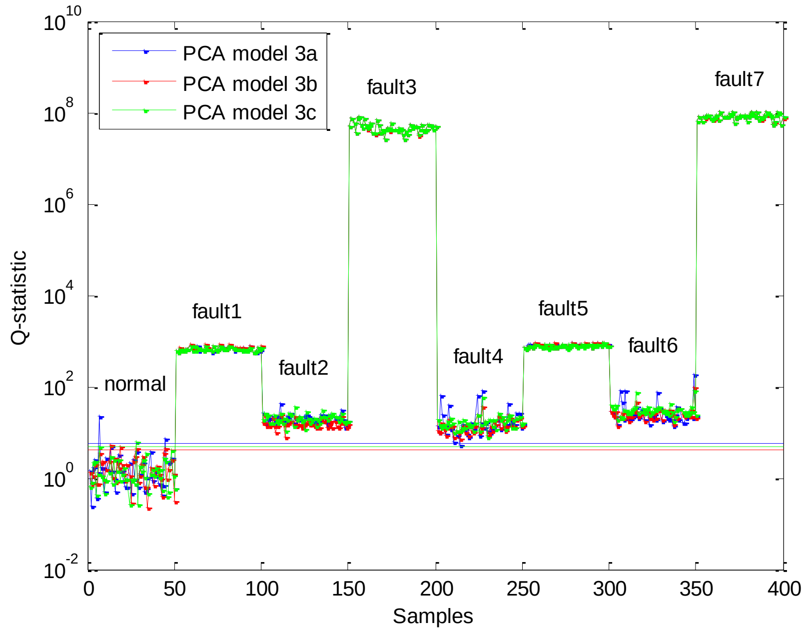


Figure 7.36 Q chart for normal and all faulty conditions obtained using PCA model 3a, model 3b, and model 3c

Table 7.8 shows that PCA model 3a and 3b achieved the lower misdetection rate (0.5%) compared to model 3c (1.0%). The PCA models 3a and 3b which were constructed using sym4 and sym8 respectively showed superior detection performance. This indicated that PCA models 3a and 3b could be more suitably applied in channel 3 for fault detection than the other one.

Table 7.8 Comparison of fault detection accuracy of PCA model in channel 3

	Misdetection rate %		
	PCA model 3a	PCA model 3b	PCA model 3c
T^2	0.5	0.5	1.0
Q	0.8	1.0	0.3
Weighted average	**0.5**	**0.5**	1.0

Figure 7.37 to Figure 7.39 show the first three dimensions of the projected scores for PCA models 3a, 3b, and 3c respectively. As can be seen in the figures, the projected scores by all PCA models were not separated clearly. The overlaps between classes

occurred in all models where the worst was in PCA model 3a. In Figure 7.37, the projected scores of all the fault conditions were mixed up and the clustering effect was not visible. The same thing also occurred in the PCA model 3b and 3c where the proposed method was not able to properly separate the classes using the projected scores in the first three PCs.

The identification accuracy of the proposed method is illustrated in Figure 7.40 to Figure 7.42 for the PCA model 3a, 3b, and 3c respectively. In Figure 7.40, it can be seen that the highest identification accuracy was from 80.2% to 85.0% where it was achieved not only by the PCA model with 7 PCs but also with the ones with 5 and 6 PCs. This suggested that the identification accuracy of the PCA model 3a did not increase by adding more PCs into model. Recalling the identification accuracy from PCA model 1a and 2a, it was found that PCA model 3a achieved a lower result. This was consistent with the classification result where the clustering effect obtained from PCA model 3a was less visible compared to PCA model 1a and 2a.

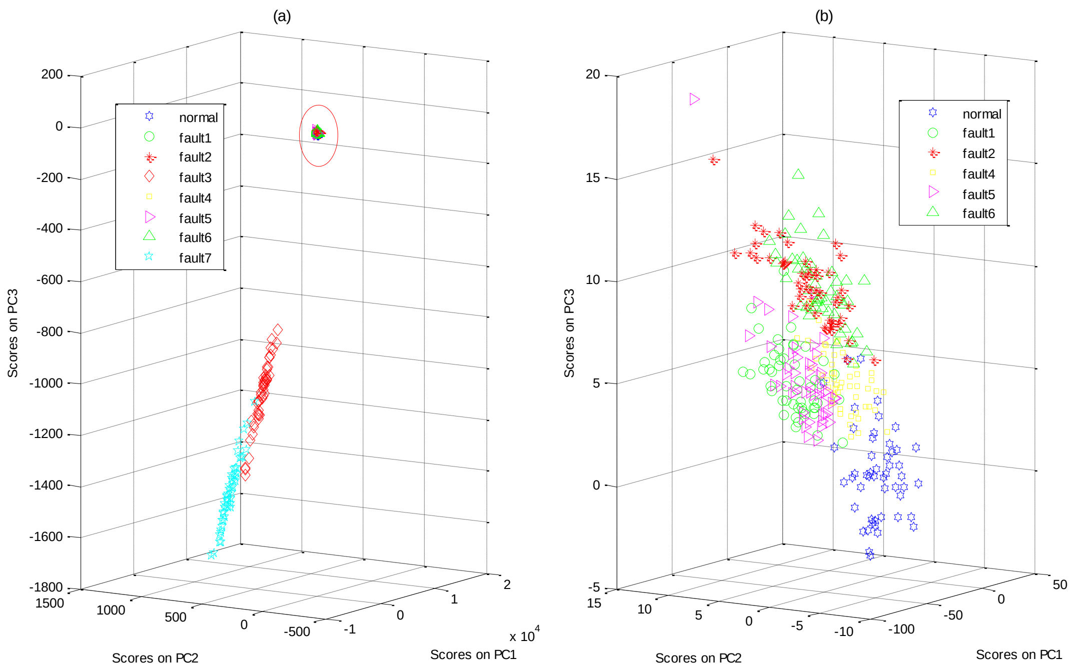


Figure 7.37 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3a. Note that exhibit (b) zooms into the red ellipse area in exhibit (a)

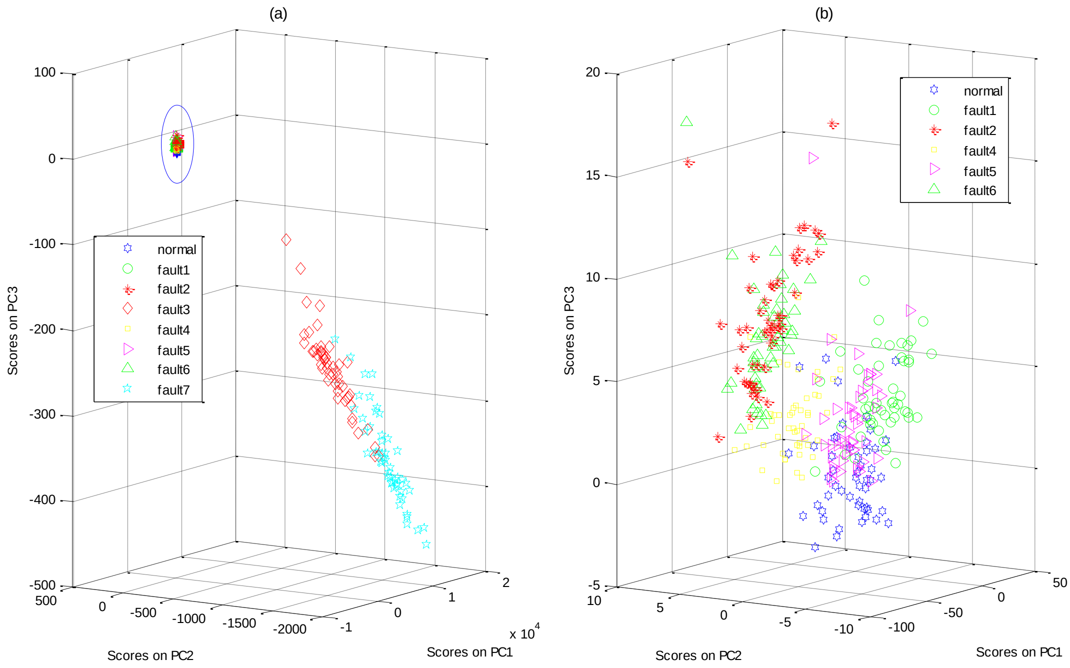


Figure 7.38 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3b. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a)

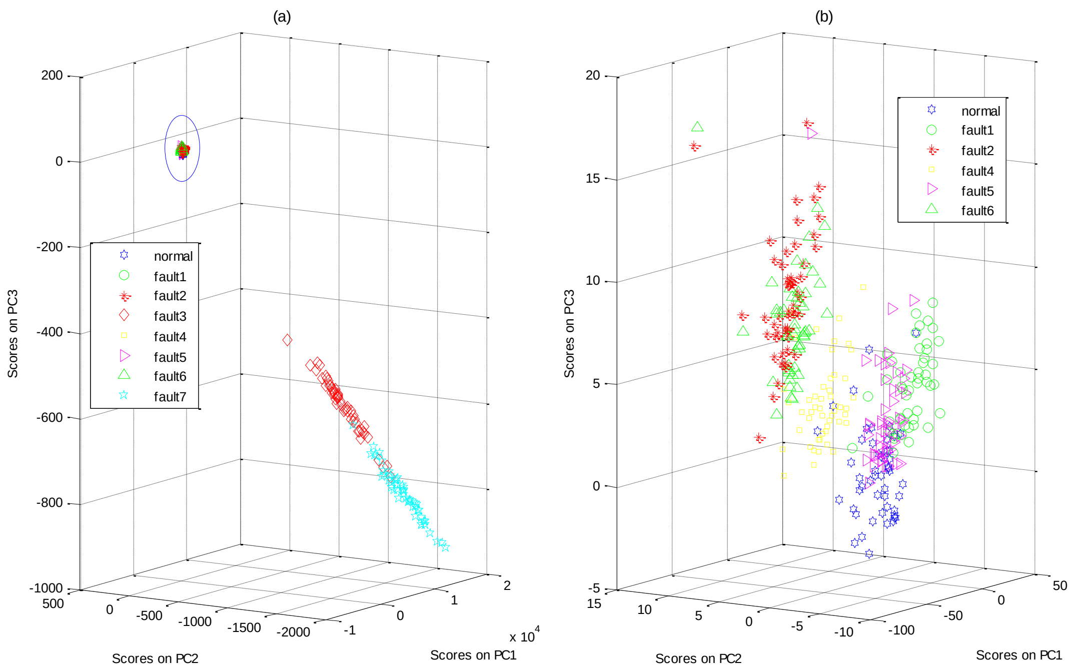


Figure 7.39 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 3c. Note that exhibit (b) zooms into the blue ellipse area in exhibit (a)

There was a slight improvement in identification accuracy of the PCA model 3b compared to the PCA model 3a as illustrated in Figure 7.41. The model with 7 PCs achieved identification accuracy from 85.0% to 87.5% while the one with 1 PC achieved the lowest accuracy ranged 63.2% to 71.5%. It can be seen that the performance of the model with 5, 6, and 7 PCs was not too much different. This observation confirmed the finding in PCA model 3a where the model with 5-7 PCs, did not significantly improve the identification accuracy by increasing the number of PCs into model.

It is shown in Figure 7.40 and Figure 7.41 that adding the number of neighbors *k* into the model yielded varying results. For PCA model 3a, increasing the number *k* would increase the identification accuracy for the model with 1 and 2 PC(s) but it would reduce the identification accuracy for the others. Meanwhile, for the PCA model 3b, increasing the number *k* in general did not significantly improve the identification accuracy of the models except in the model with 1PC. This finding indicated that in most cases, increasing the number of neighbors *k* employed in the PCA model would not significantly increase the identification accuracy of the proposed method.

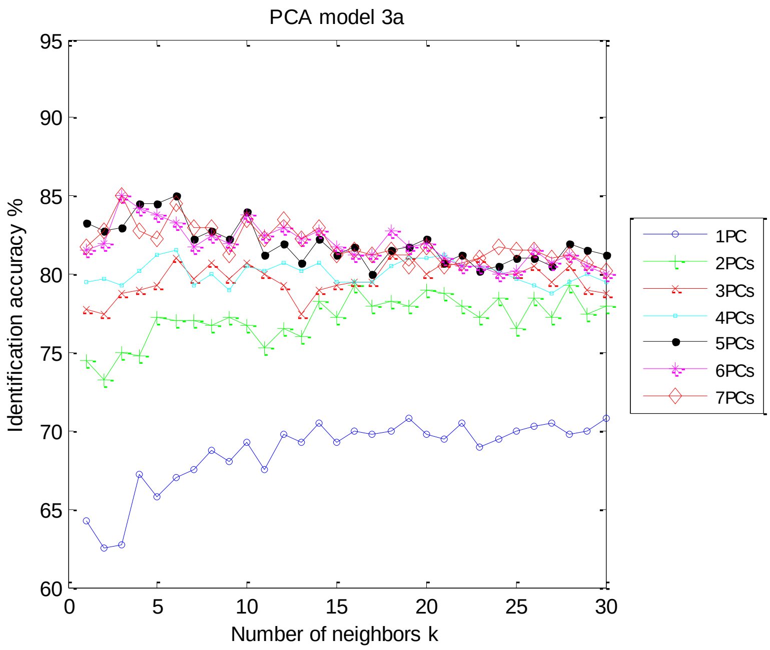


Figure 7.40 Identification accuracy comparison of the PCA model 3a using 1 to 7 PCs.

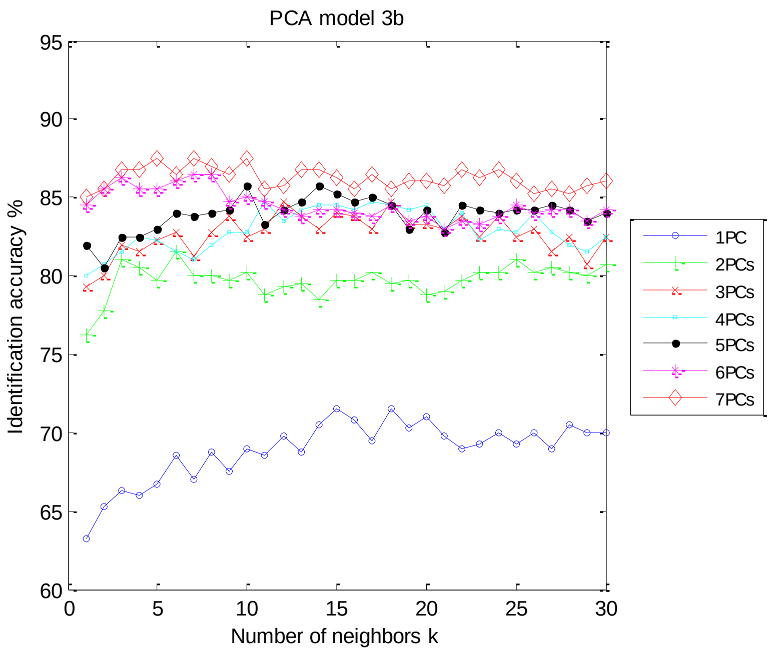


Figure 7.41 Identification accuracy comparison of the PCA model 3b using 1 to 7 PCs.

The identification accuracy of PCA model 3c as presented in Figure 7.42 shows that the model with 6 and 7 PCs achieved superior performance (87.0%-89.7%) compared to the other models. The PCA model with 2 to 5 PCs produced the middle performance from 77.2%-87.5% while the model with only 1 PC obtained the worst accuracy from 63.2%-71.0%. Furthermore, the identification accuracy for all models indicated that the number "*k*" was not sensitive to the increase of accuracy except for those with 1PC.

Three PCA models in channel 3 had different performance results in fault detection, classification and identification. The PCA model 3a and 3b obtained the best fault detection performance with 0.5% misdetection rate compared to 1.0% of model 3c. However, the PCA model 3a produced the worst performance in the fault classification process where it produced a least visible clustering effect using the scores in the first three PCs. The result of fault identification also agreed with the fault classification where the PCA model 3a obtained the lowest result (80.2%-85.0%). Meanwhile, the PCA model 3c achieved better fault classification and identification performance than the other two models although it had the lowest performance in fault detection (1.0% misdetection rate).

In view of the above results and analyses, it is appropriate to select PCA model 3c as the best PCA model in channel 3 to diagnose the faults in the centrifugal pump. Moreover, it can be concluded that sym12 provided a better result in constructing the PCA models in channel 3.

In channel 3, the accelerometer was mounted on the pump's outlet where the discharge pressure was relatively high. The condition led to an unfavourable vibration environment acquired by the sensor. This caused considerable mistakes in fault classification and identification. It was found that in channel 3, the proposed method provided less accuracy in fault classification and identification than that for the other two channels (channel 1 and 2).

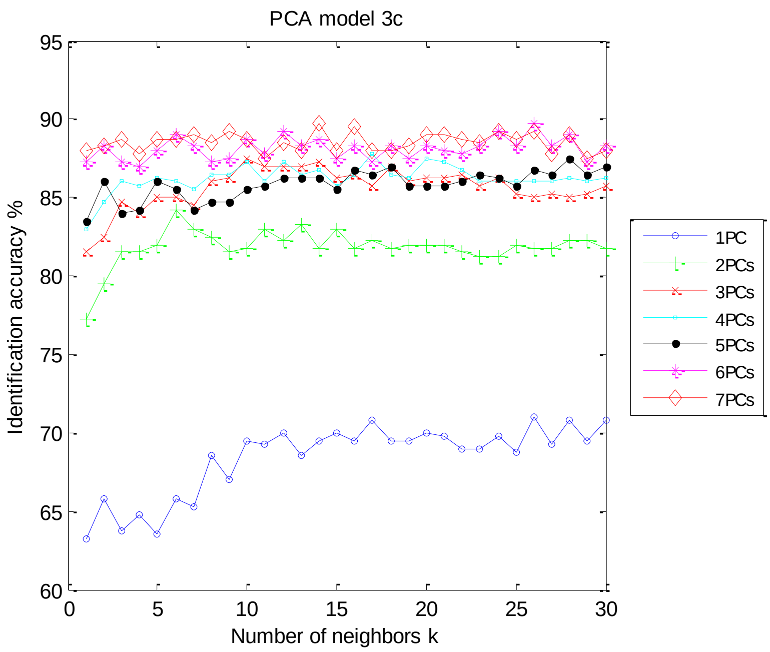


Figure 7.42 Identification accuracy comparison of the PCA model 3c using 1 to 7 PCs.

7.6.4 Fault Diagnosis Results Using PCA Model 4

The proposed method was again carried out for vibration signals measured from channel 4 and the obtained result of fault detection is presented in Figure 7.43 and Figure 7.44. In the case of the T^2 chart, there are two threshold lines, the black horizontal line and the blue horizontal line. The black horizontal line corresponds to PCA model 2b and 2c, where the blue one corresponds to PCA model 2a. The result of the T^2-statistic shows that the proposed method provided poor performance in fault detection as depicted in Figure 7.43 where the T^2 value of faulty condition samples (fault1 and fault4) crossed the threshold many times. This also occurred for normal condition samples where the T^2 value exceeded the threshold line several times.

In Figure 7.44, the proposed method demonstrates a better performance in fault detection through the use of the Q-statistic. It could be seen clearly that the proposed method was able to detect all types of faults. All the Q values from the fault condition samples lay above the threshold which indicated that the faults were correctly detected although several normal condition samples were misdetected.

The results mentioned above showed that the Q-statistic was able to better detect the faults correctly than the T^2-statistic. This is because the Q-statistic was calculated from the residual subspace of the PCA model which made it more sensitive to fault occurrence. The quantitative result of detection performance is listed in Table 7.9 as represented in misdetection rate.

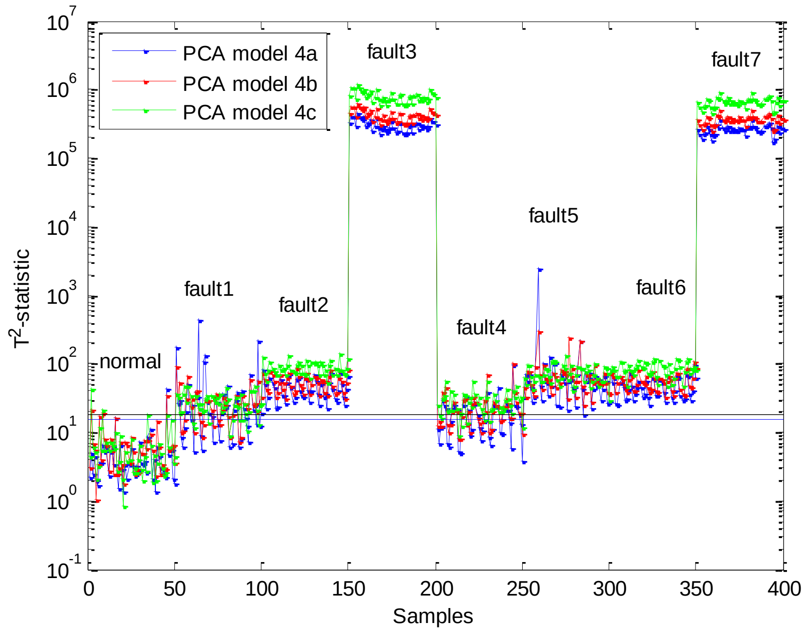


Figure 7.43 T^2 chart for normal and all faulty conditions obtained using PCA model 4a, model 4b, and model 4c

The misdetection rate in Table 7.9 shows that PCA model 4c achieved the lowest weighted average value (4.6%) which meant that the model had a relatively better detection performance than the other two models. In this channel, the PCA model 4c which was constructed using sym12 showed superior performance to the ones constructed using sym4 and sym8 therefore it was appropriate to conclude that sym12 was best applied in channel 4.

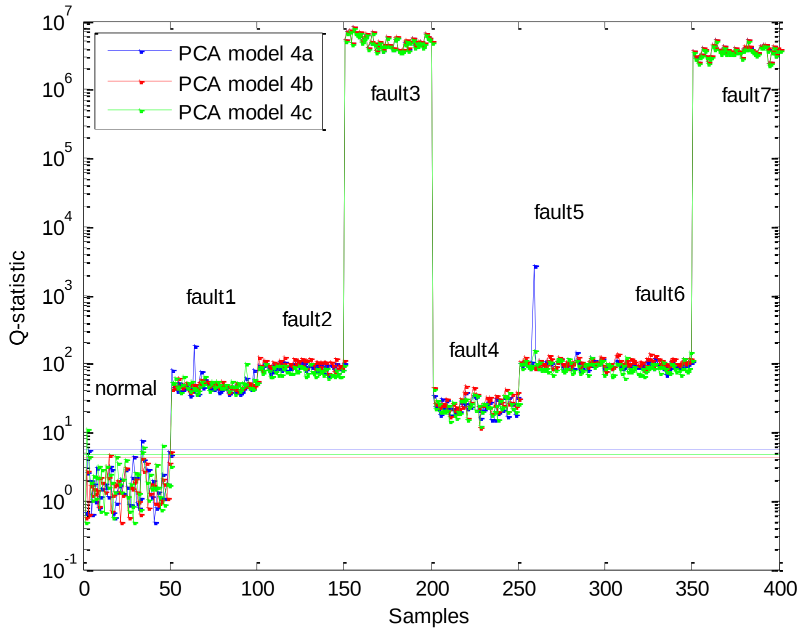


Figure 7.44 *Q* chart for normal and all faulty conditions obtained using PCA model 4a, model 4b, and model 4c

Table 7.9 Comparison of fault detection accuracy of PCA model in channel 4

	Misdetection rate %		
	PCA model 4a	PCA model 4b	PCA model 4c
T^2	11.0	10.0	4.8
Q	0.5	0.8	1.0
Weighted average	10.5	9.5	**4.6**

The proposed method produced fault classification results which are shown in Figure 7.45 to Figure 7.47. It can be seen that the PCA model 4a in Figure 7.45 was not able to separate the different types of faults. The clusters were not visible since all score samples were mixed up with each other. The similar results were also shown for PCA model 4b in Figure 7.46 and PCA model 4c in Figure 7.47. It was observed again that overlaps occurred among the classes in both models. The proposed method seems to fail to classify the fault by using the scores on the first three dimensions of the PCA model. Comparing with the results obtained from channel 1 to 3, it was found that the proposed method in channel 4 achieved the worst fault classification performance.

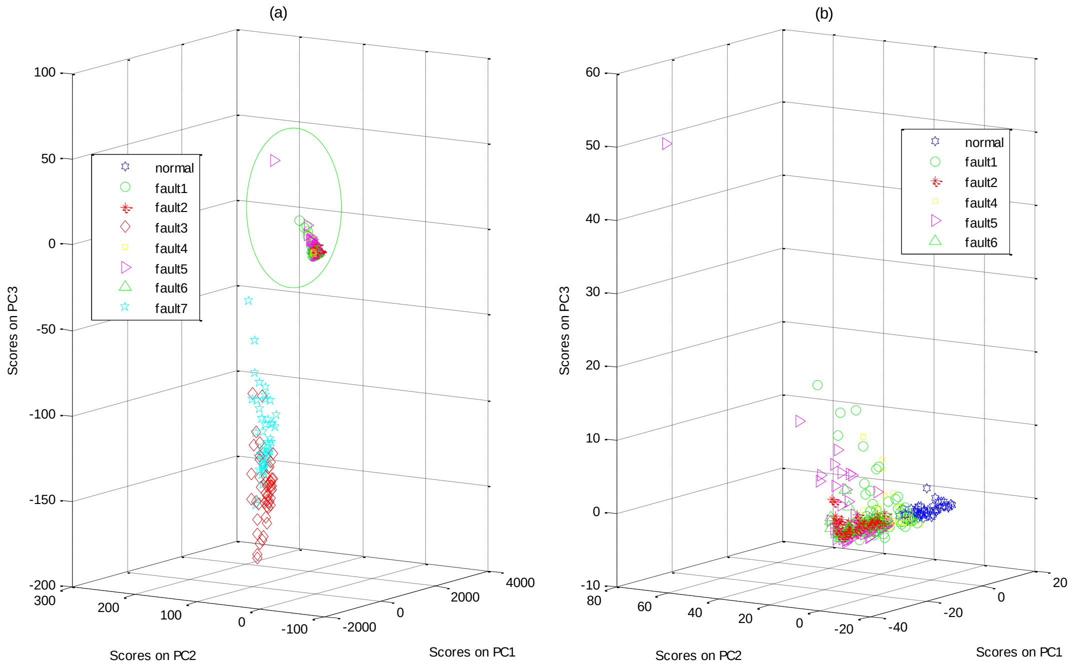


Figure 7.45 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4a. Note that exhibit (b) zooms into the red ellipse area in exhibit (a)

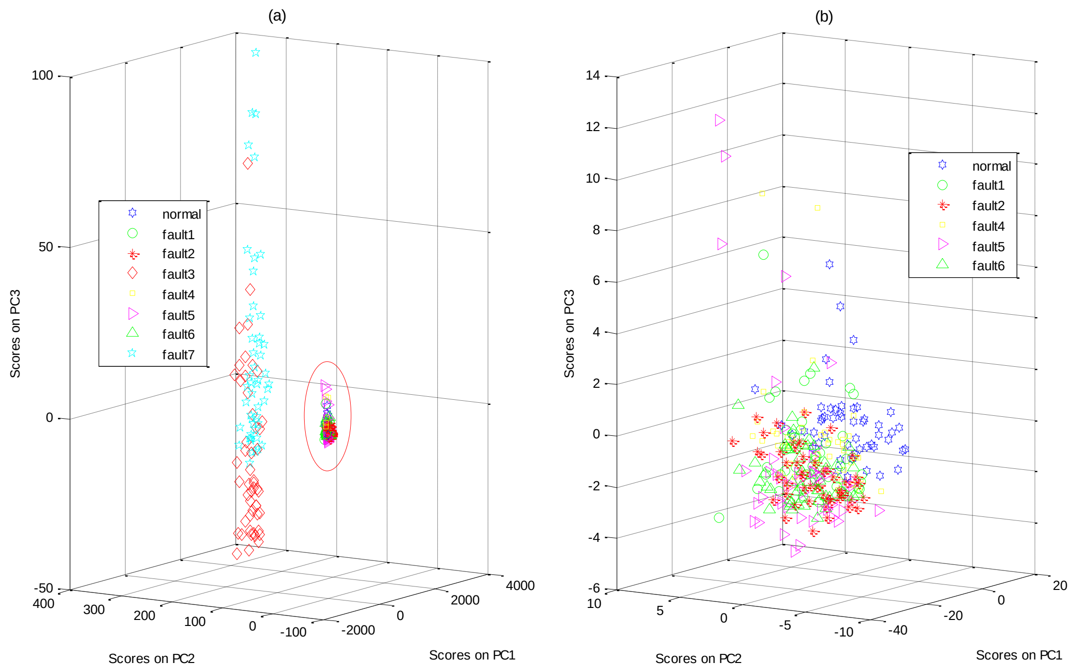


Figure 7.46 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4b. Note that exhibit (b) zooms into the red ellipse area in exhibit (a)

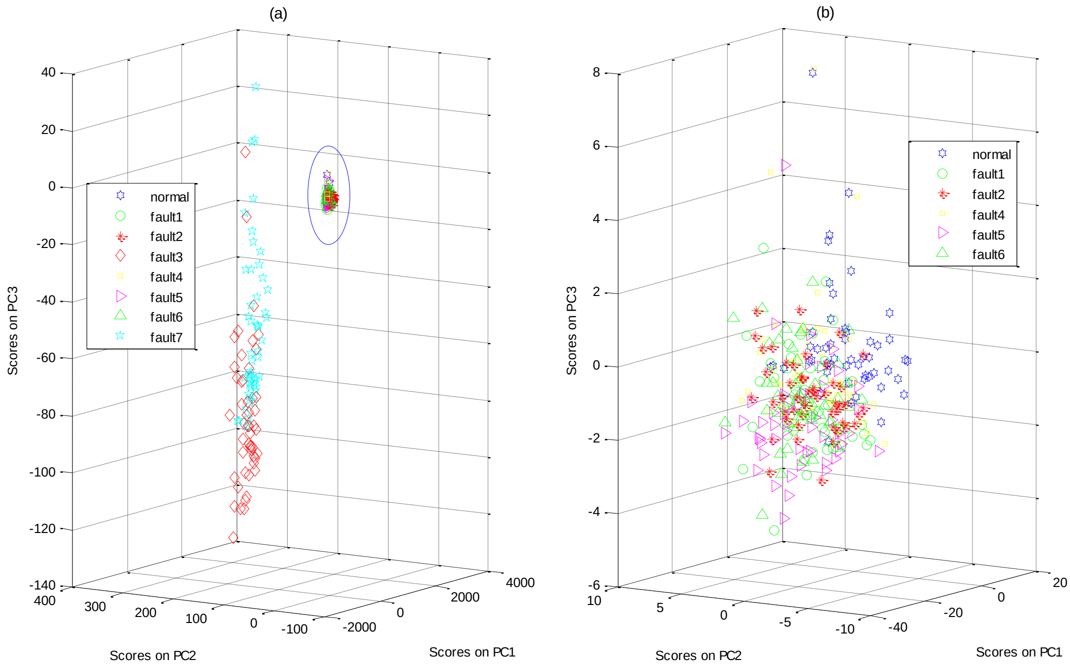


Figure 7.47 Three-dimensional scatter plots of PC1, PC2, and PC3 constructed from PCA model 4c. Note that exhibit (b) zooms into the red ellipse area in exhibit (a)

Figure 7.48 shows the identification accuracy of the PCA model 4a with 1 to 5 principal components (PCs). The PCA model 4a retained only 5 PCs therefore the proposed method evaluated its performance for 1 to 5 PCs. It can be seen that the highest identification performance was obtained by the model with 5 PCs ranged from 55.0% to 61.2% and the lowest one was obtained by the model with 1 PC from 45.5% to 51.2%. It was also found that the model with 3 and 4 PCs had a quite similar identification accuracy with the model with 5PCs which indicated that the model with 3-5 PCs was not sensitive to the increasing number of PCs.

The PCA model 4b obtained better identification accuracy than the PCA model 4a. It achieved identification accuracy from 54.7% to 62.7% (the model with 6 PCs) and the lowest one from 46.5% to 53.7% (the model with 1 PC). The PCA model 4b consisted of 6 PCs in the model so that the proposed method evaluated the identification performance of the model with 1 to 6 PCs. Note that in this case, the number of neighbors *k* included into the model had a significant effect on the identification performance as can be seen for the PCA model 4a and 4b. However the number of *k* was not sensitive to the identification accuracy on PCA model 4c.

Figure 7.50 shows the identification performance of PCA model 4c with 1 to 6 PCs. The models retained 6 PCs as described in section 7.4 where the performance was higher for the model with 6 PCs and was lower for the model with 1PC. It was also observed that the model with 3-6 PCs achieved a relatively similar level of performance as shown by the overlapping lines; the highest accuracy was however obtained by the model with 6 PCs (57.7% to 62.5%). Meanwhile, the model with 1PC provided the lowest accuracy from 43.7% to 53.2%.

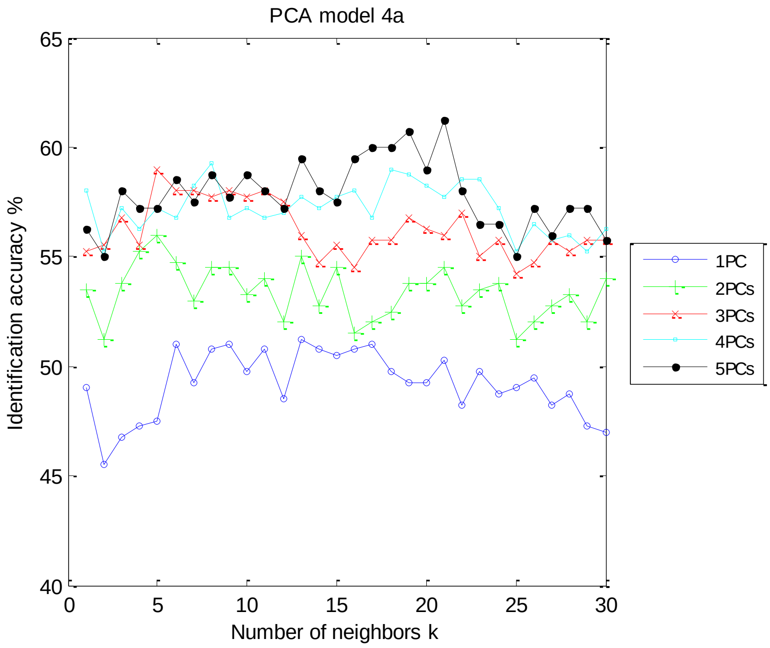


Figure 7.48 Identification accuracy comparison of the PCA model 4a using 1 to 5 PCs.

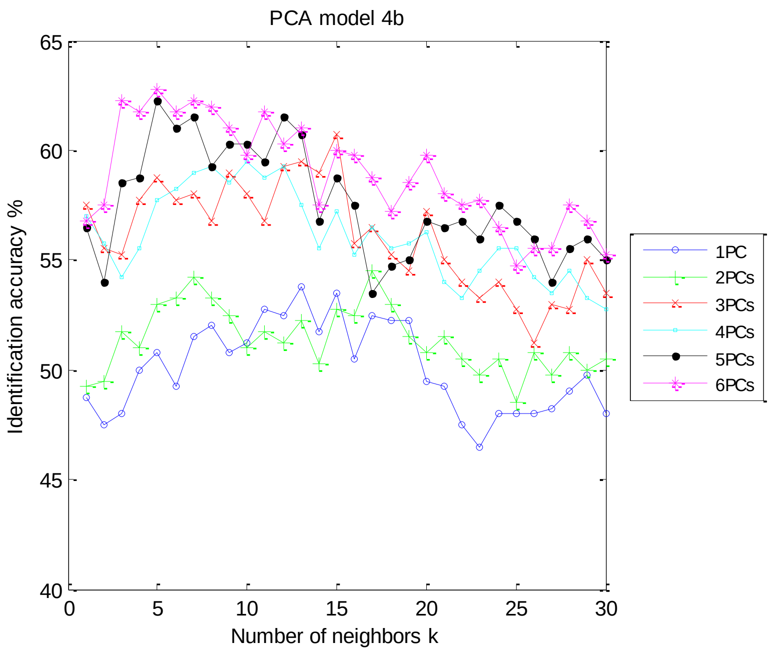


Figure 7.49 Identification accuracy comparison of the PCA model 4b using 1 to 6 PCs.

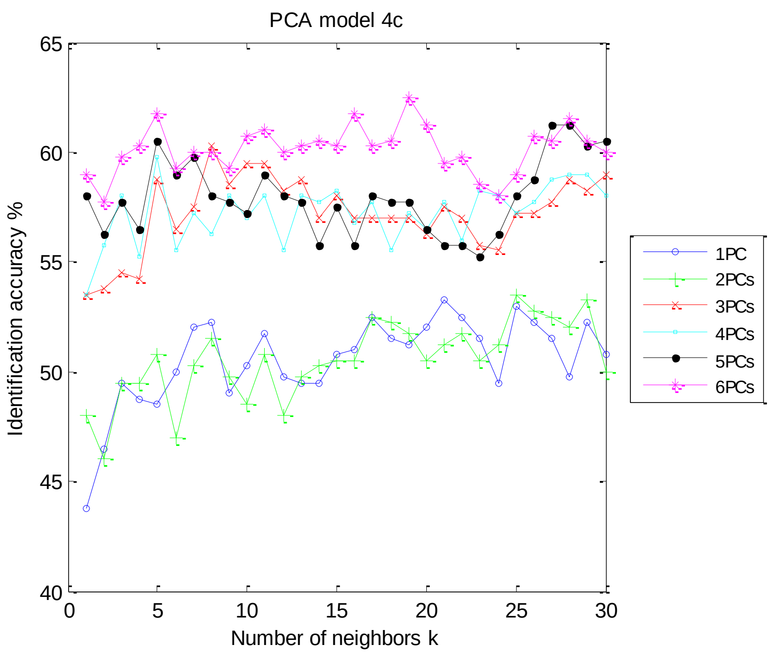


Figure 7.50 Identification accuracy comparison of the PCA model 4c using 1 to 6 PCs.

The performance of three PCA models in channel 4 (PCA model 4a, 4b, and 4c) as discussed above, showed the lowest performance compared to those in channel 1, 2 and 3. The poor performance of the models occurred in fault detection, classification, and the identification step. The highest fault detection performance was obtained by PCA model 4c with 4.6% misdetection rate, while the lowest fault detection performance obtained from other channels was 0.3%, 0.3%, and 0.5% for 1b, 2b, and 3a (and 3b) respectively.

In the fault classification, all models in channel 4 failed to separate the classes and the clustering effect was not visible. In general, PCA model 4b and 4c achieved slightly better identification accuracy which ranged from 43.7% to 62.7%. Although the performance among the models in channel 4 was quite similar, by considering the above result analyses, it is appropriate to select PCA model 4c as the best model to diagnose faults in the centrifugal pump using channel 4.

The mounting location of the accelerometer in channel 4 was on the bearing housing of the centrifugal pump. The location of the accelerometer was relatively far from the location of faults where they occurred, which was in the area close to the pump's impeller (cavitation, impeller fault, and blockage) except for the bearing fault. The noisy vibration due to severe damage of the shaft bearing was also observed. This caused the proposed method to be unable to distinguish the uniqueness of the raw vibration signals associated with the faults and led to the poor performance of the diagnosis result.

In the following section, the performance comparison among the selected PCA model from each channel was carried out and evaluated in order to find the best model to be employed for the general centrifugal pump fault diagnosis.

7.7 Performance Comparison of the PCA models

In this section, the performance comparison of the PCA models is analysed, aimed at finding which location in the centrifugal pump gave the best fault diagnosis results using the proposed method. The performance comparison was based on the misdetection rate obtained by the PCA models for each channel. Moreover, the average misdetection rate was taken from three PCA models in each channel and the channel that had the lowest misdetection rate was then selected as the most appropriate channel for acquiring the vibration signal. Since each channel was related to a specific accelerometer location, hence the best location in the centrifugal pump to acquire the vibration signal for fault diagnosis purposes can then be concluded.

The misdetection rate comparison for all four channels is illustrated in Figure 7.51. There were three PCA models for each channel where each was indexed using the letter ‘a’, ‘b’, and ‘c’. Therefore in channel 1 there were PCA models 1a, 1b, and 1c and this naming convention also applied for the other channels.

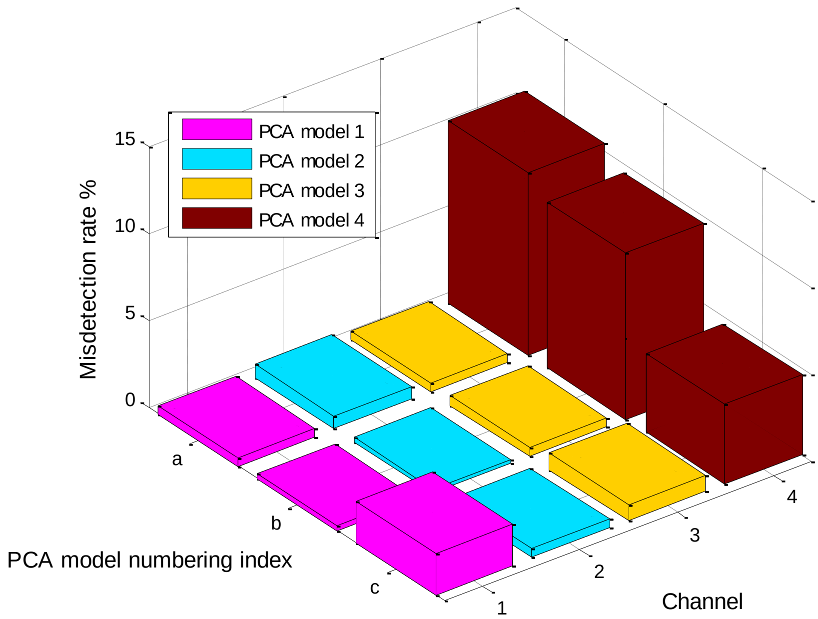


Figure 7.51 Misdetection rate of all PCA models from channel 1 to 4

In Figure 7.51 and Figure 7.52, it is obvious that the lowest misdetection rate was obtained by the PCA model constructed from channel 2. This finding implies that channel 2 was the best channel for the purpose of fault detection. Since the accelerometer sensor in channel 2 was mounted onto the pump’s volute casing then it could be also inferred that the volute of a centrifugal pump was the most appropriate accelerometer’s mounting location for fault detection, for the range of faults being considered here.

Figure 7.51 also shows the lowest misdetection rate obtained by PCA models in each channel. The PCA model 1b, 2b, 3a, and 4c obtained the lowest misdetection rate for channel 1, 2, 3, and 4 respectively. It could be inferred that sym8 was most suitably applied in channel 1 and 2, sym4 in channel 3 and sym12 in channel 4 for fault detection in a centrifugal pump.

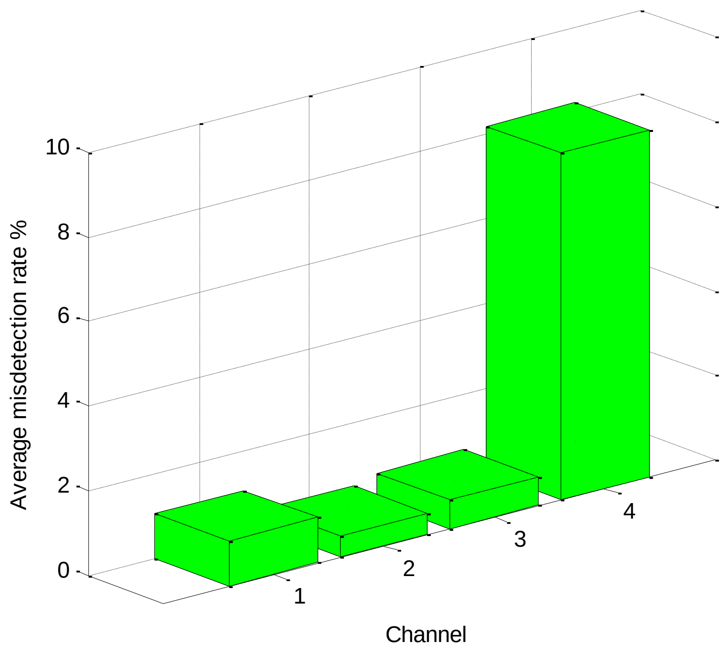


Figure 7.52 Average misdetection rate from channel 1 to 4

From section 7.6, it was shown that for all channels the highset identification accuracy was achieved by the PCA model with the most number of PCs. It was found that the highest identification accuracy was obtained by PCA model 1b, PCA model 2b, PCA model 3c, and PCA model 4c. The comparison of the average identification accuracy among those models is illustrated in Figure 7.53. In this figure, the z-axis was calculated from the average value of the identification accuracy obtained by employing 1 to 30 number of neighbors, *k*. It is obvious that the PCA model 1b achieved the highest identification accuracy of 99.2% while PCA model 2b, 3c, and 4c achieved an identification accuracy of 97.1%, 88.5%, and 60.1% respectively. It can be concluded that PCA model 1b which was constructed using sym8 is the best model to identify fault in the centrifugal pump.

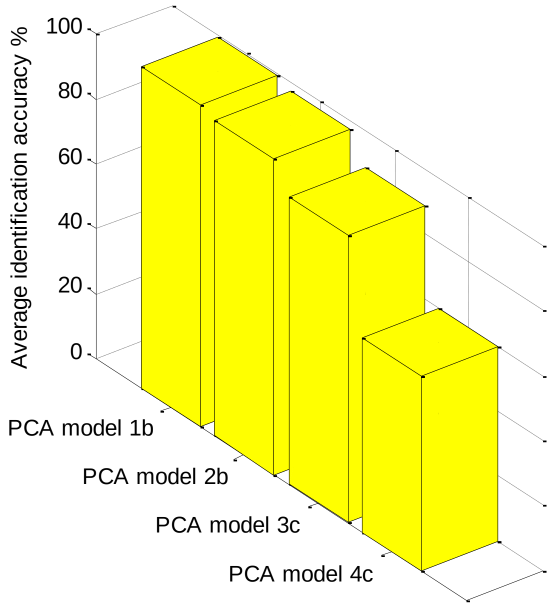


Figure 7.53 Identification accuracy of the best PCA models in each channel

Among the 30 neighbors employed in PCA model 1b, the model with 20 neighbors achieved the highest identification accuracy of 99.7%. The confusion matrix which shows information about the actual and predicted class of faults done by the proposed method is given in Table 7.10. In the table, each column of the matrix indicates the instances in a predicted class, while each row indicates the instances in an actual class.

It is obvious that there was only one mislabelling/misidentification (highlighted in yellow) of a fault class where the normal class (actual) was predicted as fault4. This resulted in an identification error 1 out of 400 samples or 0.3% which means 99.7% identification accuracy as mentioned above.

Table 7.10 Confusion matrix of PCA model 1b employing 20 neighbors *k*

		PREDICTED							
		normal	fault1	fault2	fault3	fault4	fault5	fault6	fault7
ACTUAL	normal	49	0	0	0	1	0	0	0
	fault1	0	50	0	0	0	0	0	0
	fault2	0	0	50	0	0	0	0	0
	fault3	0	0	0	50	0	0	0	0
	fault4	0	0	0	0	50	0	0	0
	fault5	0	0	0	0	0	50	0	0
	fault6	0	0	0	0	0	0	50	0
	fault7	0	0	0	0	0	0	0	50

The confusion matrix for the PCA model 2b, 3c, and 4c is given in Table 7.11 to Table 7.13 respectively. The PCA model 2b obtained the highest identification accuracy of 98.5% by employing 20 neighbors, the PCA model 3c achieved the highest identification accuracy of 89.7% by employing 14 neighbors, and the PCA model 4c had the highest identification accuracy of 62.5% by employing 19 neighbors.

Table 7.11 Confusion matrix of PCA model 2b employing 20 neighbors *k*

		PREDICTED							
		normal	fault1	fault2	fault3	fault4	fault5	fault6	fault7
ACTUAL	normal	50	0	0	0	0	0	0	0
	fault1	0	50	0	0	0	0	0	0
	fault2	0	0	46	0	0	0	4	0
	fault3	0	0	0	50	0	0	0	0
	fault4	0	0	0	0	50	0	0	0
	fault5	0	0	0	0	0	50	0	0
	fault6	0	0	2	0	0	0	48	0
	fault7	0	0	0	0	0	0	0	50

Table 7.12 Confusion matrix of PCA model 3c employing 14 neighbors *k*

		PREDICTED							
		normal	fault1	fault2	fault3	fault4	fault5	fault6	fault7
ACTUAL	normal	50	0	0	0	0	0	0	0
	fault1	0	49	0	0	0	1	0	0
	fault2	0	0	42	0	0	0	8	0
	fault3	0	0	0	43	0	0	0	7
	fault4	0	1	0	0	47	0	2	0
	fault5	0	5	0	0	0	44	1	0
	fault6	0	0	16	0	0	0	34	0
	fault7	0	0	0	0	0	0	0	50

Table 7.13 Confusion matrix of PCA model 4c employing 19 neighbors *k*

		PREDICTED							
		normal	fault1	fault2	fault3	fault4	fault5	fault6	fault7
ACTUAL	normal	45	3	0	0	2	0	0	0
	fault1	0	29	0	0	21	0	0	0
	fault2	0	1	23	0	0	6	20	0
	fault3	0	0	0	38	0	0	0	12
	fault4	1	19	0	0	30	0	0	0
	fault5	0	1	12	0	0	24	13	0
	fault6	0	0	19	0	0	8	23	0
	fault7	0	0	0	12	0	0	0	38

The next chapter presents conclusions and suggestions for future work of the thesis.

CHAPTER EIGHT

8 Conclusions and Suggestions for Future Work

8.1 Conclusions

The proposed method was investigated in order to find a new technique for centrifugal pump fault diagnosis. It was based on a combination of wavelet transform, statistical parameters, and PCA modelling. The area of centrifugal pump fault diagnosis has not yet been fully researched and moreover the fault diagnosis techniques based on a wavelet-PCA model are still wide open to discovery. The investigation in this study contributes new knowledge on the use of wavelet transforms using Symlet and PCA models in building an integrated centrifugal pump fault diagnosis framework. It produces integrated algorithms for statistical feature extraction of decomposed vibration signals and for PCA modelling. The findings of the investigation also included PCA-based fault detection, fault classification and fault identification.

Six statistical features, namely energy level, standard deviation, RMS, kurtosis, variance, and crest factor were extracted from the decomposed vibration signals obtained from four accelerometers located on the pump. The investigation of a new technique for feature extraction was carried out and led to the construction of the PCA model-based fault diagnosis for a centrifugal pump. The feature extraction technique combined the application of wavelet-based decomposition and the statistical parameters. The PCA models generated by using wavelet-based statistical parameters were used successfully in fault detection, fault classification, and fault identification in the proposed method.

The development of the wavelet-PCA method in this study aimed to find a vibration-based multi fault diagnosis method for centrifugal pumps. The proposed wavelet-PCA method produced a set of PCA models obtained from four channels of vibration signals from four different accelerometers located on the pump. The channel number 1 to 4 corresponded to accelerometers mounted on the pump inlet, pump volute, pump outlet, and the pump bearing house respectively. Each channel produced 3 PCA models which were built using Symlet 4, 8, 12 wavelets respectively.

The Multi Resolution Analysis (MRA) of Symlet wavelets were used to decompose the time-domain vibration signal at up to 5 levels. The decomposition process produced Approximation coefficients (*cA*) and Detailed coefficients (*cD*). Only the *cA* parts were selected in this study for further processing. Six statistical features were extracted from each of the approximation coefficients (*cA*); therefore 30 statistical features were produced for each PCA model building process. The results of the feature extraction process were used to build the PCA model which consisted of the reduced loading matrix $\widehat{\mathbf{P}}$ and the reduced eigenvalue matrix $\widehat{\mathbf{\Lambda}}$. The matrices $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{\Lambda}}$ were used as the core matrices for the fault detection, classification, and identification process.

The fault detection process was carried out using the T^2 and Q-statistic where their values were compared with the threshold lines of T_α^2 and Q_α respectively. Twelve PCA models obtained from the four channels were then tested during the fault detection process. There were four single faults and three multi faults introduced for testing, namely: cavitation, impeller fault, bearing fault, blockage, impeller fault-cavitation, impeller fault-blockage, and bearing fault-cavitation. The fault detection process was performed on all types of faults. The test results showed that the PCA model constructed from the vibration data obtained from channel 2 using the sym8 (PCA model 2b) achieved the lowest misdetection rate of 0.3%. The finding implies that channel 2, where the accelerometer was mounted on the pump volute, provided the best mounting location of the accelerometer for the purpose of fault detection. Moreover, it concluded that sym8 provided the best decomposition of the time-domain waveform for statistical feature extraction purposes where it gave the PCA model 2b the lowest misdetection rate.

The highest misdetection rate in the fault detection was 10.5%, which was achieved by PCA model 4a. This rate occurred in the test results using the PCA model constructed from channel 4. The result concludes that the mounting location on the bearing house provides the worst performance in the fault detection rate. In addition, among the three Symlet decompositions in channel 4, sym12 performed better in the results since it achieved the lowest misdetection rate.

Moreover, the test result also showed that the misdetection rate achieved when using the Q-statistic, in general was lower than that from the T^2-statistic. This finding indicates that the Q-statistic is more sensitive in detecting the small behaviour of the system so that a small change in vibration signals indicating the occurrence of a fault can be detected earlier. However, in a particular case, the Q-statistic attains a worse misdetection rate than the T^2-statistic. This condition refers to a situation where the residual subspace of the PCA model tends to be too sensitive to the change of vibration behaviour, therefore several Q-statistic values exceeded the threshold.

The scores obtained from all PCA models were used to classify faults introduced on the centrifugal pump. The scores from normal (no fault) and faulty conditions were plotted using the first three principal components (PCs) in order to reveal the clustering effect. The PCA model 1a and 1b showed that the clustering effect was visible in which each of the classes were separated and could be distinguished. A small overlap was observed to occur in both of the models. However, the overlap could be avoided by adding more PCs in the classification process. Meanwhile, the situation was different for PCA model 1c where the clustering effect was not obviously visible. A large number of overlaps seem to occur among the classes which made them difficult to be classified.

All the PCA models from channel 2 revealed a clear separation among classes where each class could be classified clearly and only a small amount of overlap occurred. The classification results obtained from channel 3 and 4 showed that the projected scores on the first three PCs were not obviously separated. The overlap occurred among all classes so that the fault classification was not properly achieved.

The test results showed that the performance of fault identification from the PCA model in channel 1 was obviously quite high where the highest identification accuracy was achieved by PCA model 1b, with 99.2%. This accuracy was achieved by employing all PCs (8PCs) available in the model. The PCA models 2a, 2b and 2c achieved relatively the same identification accuracy which are 96.7%, 97.07% and 96.5% respectively. Meanwhile, the fault identification performance of PCA models obtained from channel 3 and 4 was significantly lower than the previous ones.

The confusion matrix presented information about actual and predicted classes of faults achieved by the proposed method. The matrix showed that there was only one mislabelling of a fault class for PCA model 1b and six mislabellings for PCA model 2b.

The following section presents the final conclusions of the investigation of the proposed wavelet-PCA method for fault diagnosis of centrifugal pumps.

As can be concluded from the literature review, the use of a combined statistical feature extraction, wavelet transform and PCA methods in the application of vibration-based fault diagnosis of centrifugal pumps are still wide open to discovery. This study focused on investigating a new method for multi fault diagnosis by using a combination of six statistical parameters, wavelet transform, and PCA model.

There was no single PCA model tested that achieved the best performance for all fault detection, fault classification, and fault identification; there were also no single PCA models suited for all channels. Rather, each case had to be diagnosed with a different PCA model.

The test results of the proposed method shows that PCA model 2b achieved the lowest (and therefore the best) misdetection rate of 0.3% among the other models. In addition, it also achieved very high performance in fault classification in which the separation among classes was obviously visible. However in fault identification, it did not attain the best performance where it achieved 97.1% of the identification accuracy, although this result was obviously quite high. As a comparison, the highest fault identification performance was achieved by PCA model 1b with 99.2% identification accuracy.

The PCA model 2b was generated from the time-domain vibration signal of channel 2 by using sym8 wavelet family. This implies that channel 2 which corresponds to the mounting location of the accelerometer on the pump volute provides a better location for vibration acquisition than the other locations on the pump. Furthermore, in the PCA model 2b building process, Symlet 8 (sym8) decomposed the time-domain vibration signal at up to 5 levels and six statistical parameters were extracted from each level. The test results show that the decomposition processes using sym8 wavelet produced a better PCA model than the others.

As a final conclusion and in answering the main objective of the study, it can be expressed that the proposed wavelet-PCA based method can be used for vibration-based multi fault diagnosis−consisting of fault detection, fault classification and fault identification−of centrifugal pumps, with some limitations as explained in the previous discussions. That is, different PCA models have to be employed for each transducer location in order to achieve the best performance.

8.2 Suggestions for Future Work

There are several aspects related to the study which can be explored further in order to improve the fault diagnosis performance of the proposed wavelet-PCA method. The future work may also aim to expand the generality of the method in fault diagnosis for other rotating machinery applications.

It is important to investigate the use of different wavelet types to decompose the time-domain vibration signal in order to generate the most suitable features for the PCA model. For example, the application of different wavelet types like Daubechies, Coiflets, and Reverse Biorthogonal for feature extraction processes may be valuable for further investigation.

There is a need to investigate the effect of Symlet types (sym-n) used to decompose the time-domain vibration signal in feature generation processes. This is due to the test results showing that a particular sym-n type provides a better PCA model than the other types. For instance, the sym8 wavelet type produced the PCA model 1b which gave a high performance in fault detection and PCA model 2b which achieved the best fault classification and fault identification. On the other hand, the results showed that sym4 (PCA model 1a and 2a) and sym12 (PCA model 1c and 2c) gave lower accuracy classification. This implies that sym4 and sym12 are not best suited for feature generation of PCA modelling processes with channels 1 and 2.

The findings showed that the statistical features extracted from the time-domain vibration signals produced a strong fault diagnosis performance. However, an investigation to explore the use of statistical features extracted from the frequency-domain and a combination from time- and frequency-domain is advisable. To enhance the quality of the features providing centrifugal pump fault-related information, it is important to develop a feature selection procedure in order to discard or weaken the irrelevant features so that only the most prominent features would be fed to the PCA model building process. Moreover, the feature selection procedure also aims to avoid the difficulties with dimensionality.

Further exploration of the use of the other variants of PCA like kernel-PCA (KPCA) may be beneficial to be examined. While the PCA is performed in the original sample space, the KPCA is carried out in the extended feature space. The method uses the kernel function and is most commonly applied in the system where a nonlinear condition most likely occurs. Even though the vibration signal obtained from the test did not indicate any non-linearity, the investigation of the use of KPCA is needed in order to evaluate its performance compared to the proposed method.

The proposed wavelet-PCA method needs to be tested using different time-domain vibration signals acquired from other pump sources. The test would evaluate the generality of the proposed method to fault diagnosis of various sizes of centrifugal pumps. It is also needed to test the proposed method using different capacity and operational speeds of the centrifugal pumps in order to assess the generalisation ability of the proposed method to the pump’s operational parameters.

There is an aspect to evaluate other distance metrics in the kNN rule. Instead of applying the Euclidean distance metric, the use of Mahalanobis distance, Hamming distance, cosine distance or others are beneficial to be investigated in order to search better identification accuracy on fault identification. Furthermore, the use of other variants of kNN such as the weighted kNN (WkNN) would be interesting to be explored.

There is scope to investigate other types of pump faults which are not examined in this study. The additional faults could be used to evaluate the applicability of the proposed method to the diversity of faults which are most likely to occur in the real world application.

There is scope to extend the use of the proposed method; that is, instead of using vibration signals, it is a challenge to use acoustic emission signals to generate potential features for the PCA model building process.

The future work may also include the application of the proposed method to fault diagnosis for other rotating machinery like induction motors, turbines and gearboxes.

References

1. Tran, V.T. and B.-S. Yang, *An Intelligent Condition-based Maintenance Platform for Rotating Machinery*. Expert Systems with Applications, 2012. **39**(3): p. 2977-2988.
2. Jardine, A.K.S., D. Lin, and D. Banjevic, *A Review on Machinery Diagnostics and Prognostics Implementing Condition-based Maintenance.* Mechanical Systems and Signal Processing, 2006. **20**(7): p. 1483-1510.
3. Tsang, A.H.C., *Condition-based Maintenance: Tools and Decision Making.* Journal of Quality in Maintenance Engineering, 1995. **1**(3): p. 3 - 17.
4. Gertsbakh, I.B., *Models of Preventive Maintenance*, 1977, North -Holland Publishing Company: New York-Oxford.
5. Ahmad, R. and S. Kamaruddin, *An Overview of Time-based and Condition-based Maintenance in Industrial Application.* Computers & Industrial Engineering, 2012. **63**(1): p. 135-149.
6. Canfield, R.V., *Cost Optimization of Periodic Preventive Maintenance.* Reliability, IEEE Transactions on, 1986. **35**(1): p. 78-81.
7. Labib, A.W., *A Decision Analysis Model for Maintenance Policy Selection Using A CMMS.* Journal of Quality in Maintenance Engineering, 2004. **10**(3): p. 191-202.
8. Seera, M., et al., *Condition Monitoring of Induction Motors: A Review and An Application of An Ensemble of Hybrid Intelligent Models.* Expert Systems with Applications, 2014. **41**(10): p. 4891-4903.
9. Campos, J., *Development in The Application of ICT in Condition Monitoring and Maintenance.* Computers in Industry, 2009. **60**(1): p. 1-20.
10. Bloch, H.P. and F.K. Geitner, *Machinery Failure Analysis and Troubleshooting*, 1983, Gulf: Houston, TX.
11. Assaad, B., M. Eltabach, and J. Antoni, *Vibration Based Condition Monitoring of A Multistage Epicyclic Gearbox in Lifting Cranes.* Mechanical Systems and Signal Processing, 2014. **42**(1–2): p. 351-367.
12. Karassik, I.J., et al., *Pump Handbook*, 1986, McGraw-Hill Inc.
13. Thobiani, F.A., *The Non-intrusive Detection of Incipient Cavitation in Centrifugal Pumps*, 2011, University of Huddersfield.
14. Girdhar, P., *Practical Centrifugal Pumps: Design, Operation and Maintenance*, 2005, Elsevier: U.K.
15. Karassik, I.J. and T. McGuire, *Centrifugal Pumps: Selection, Operation, And Maintenance*, 1998, Chapman & Hall: New York.
16. McKee, K., et al. *A Review of Major Centrifugal Pump Failure Modes with Application to The Water Supply and Sewerage Industries*. in *ICOMS Asset Management Conference*. 2011. Gold Coast, QLD, Australia.
17. *Pump and Pump System Glossary*. [cited 2014 4 February]; Available from: http://www.pumpfundamentals.com/pump_glossary.htm.
18. Bachus, L. and A. Custodio, *Know and Understand Centrifugal Pumps*, 2003, Elsevier: U.K.
19. ESDU. *Radial, Mixed and Axial Flow Pumps*. 2007 [cited 2014 4 February]; Available from: http://www.idmeb.org/contents/resource/80030b_15_23.pdf.

20. Parrondo, J.L., S. Velarde, and C. Santolaria, *Development of A Predictive Maintenance System for A Centrifugal Pump*. Journal of Quality in Maintenance Engineering, 2006. **4**(3).
21. Volk, M., *Pump Characteristics and Applications*, 2005, Volk & Associates Inc.: USA.
22. *Centrifugal Pumps*. [cited 2014 23 January]; Available from: http://www.processpump.info/centrifugal-pumps/centrifugal-pumps.html.
23. *Pump Spares*. [cited 2014 6 January]; Available from: http://www.mfginvestmentcasting.com/pump-spares.html.
24. *Scanpump Original Pump Spare Parts*. [cited 2013 20 November]; Available from: http://www.sulzer.com/en/Products-and-Services.
25. Karassik, L., et al., *Pump Handbook*, 2007, McGraw-Hill: New York.
26. *Water Pump Bearings*. [cited 2013 11 March]; Available from: http://www.nskamericas.com/cps/rde/xchg/na_en/hs.xsl/electrical-accessories.html.
27. *Engineer's World: Basic Pump Principles*. [cited 2013 4 December]; Available from: http://eng-in-eng.blogspot.com.au/2012/10/basic-pump-principles.html.
28. Bloch, H.P. and A.R. Budris, *Pump User's Handbook Life Extension*, 2010, Lilburn, GA: Fairmont Press.
29. Beebe, R.S., *Predictive Maintenance of Pumps Using Condition Monitoring*, 2004, Elsevier Ltd: Oxford, UK.
30. *What is Head?* [cited 2013 22 October]; Available from: http://www.pumpfundamentals.com/what%20is%20head.htm.
31. Jacobsen, C.B. *The Centrifugal Pump*. [cited 2013 16 November]; Available from: http://www.grundfos.com/content/dam/Global%20Site/Industries%20%26%20solutions/Industry/pdf/The_Centrifugal_Pump.pdf.
32. Alfayez, L., D. Mba, and G. Dyson, *The Application of Acoustic Emission for Detecting Incipient Cavitation and The Best Efficiency Point of A 60kW Centrifugal Pump: Case Study*. NDT & E International, 2005. **38**(5): p. 354-358.
33. Cudina, M., *Detection of Cavitation Phenomenon in A Centrifugal Pump Using Audible Sound*. Mechanical Systems and Signal Processing, 2003. **17**(6): p. 1335-1347.
34. Ganeriwala, S. and V. Kanakasabai, *Using Vibration Signatures Analysis to Detect Cavitation in Centrifugal Pumps*, in *Rotating Machinery, Structural Health Monitoring, Shock and Vibration, Volume 5*, T. Proulx, Editor 2011, Springer New York. p. 499-507.
35. Gorla, R. and A. Khan, *Turbomachinery: Design and Theory*, 2003, Marcel Dekker Inc: USA.
36. Herbich, J., *Handbook of Dredging Engineering*, 2000, McGraw Hill: USA.
37. Shah, Y., A. Pandit, and V. Moholkar, *Cavitation Reaction Engineering*, 1999, Kluwer Academic, Springer Verlag: USA.
38. Turton, R.K., *An Introductory Guide to Pumps and Pumping Systems*, 1993, Mechanical Engineering Publication: London.
39. Collins, J., *Failure of Materials in Mechanical Design; Analysis, Prediction, Prevention*, 1993, John Wiley and Sons Ltd: UK.

40. Birajdar, R., R. Patil, and K. Khanzode. *Vibration and Noise in Centrifugal Pumps-Sources and Diagnosis Methods*. in *Third International Conference on Integrity, Reliability and Failure*. 2009. Porto, Portugal.
41. Klubnick, R. *Detecting Pump Cavitation with a Vibration Sensor*. Pumps & Systems 2007 [cited 2014 19 March]; Available from: http://www.pump-zone.com/topics/instrumentationcontrols/detecting-pump-cavitation-vibration-sensor.
42. *Centrifugal Pump Help*. [cited 2013 20 January]; Available from: http://jacpump.files.wordpress.com/2011/04/thermalchgo-017w2.jpg.
43. Albraik, A., et al., *Diagnosis of Centrifugal Pump Faults Using Vibration Methods*. Journal of Physics: Conference Series, 2012. **364**.
44. Rayner, R., *Pump Users Handbook*, 1995: Oxford: Elsevier Advanced Technology.
45. *Bearing Damage Analysis*. [cited 2014 6 March]; Available from: http://www.timken.com/.
46. *Pumps For Sewage, Wastewater & Sludge Applications*. [cited 2014 18 February]; Available from: www.hidrostal.co.uk/docs/pdf_docs/pumping%20sewage.pdf.
47. Sethiya, S.K. *Condition Based Maintenance (CBM)*. 2006 [cited 2014 27 February]; Available from: http://irsme.nic.in/files/cbm-sethiya.pdf.
48. Deng, L. and R. Zhao, *A Vibration Analysis Method Based on Hybrid Techniques And Its Application to Rotating Machinery*. Measurement, 2013. **46**(9): p. 3671-3682.
49. Feng, Z., F. Chu, and M.J. Zuo, *Time–frequency Analysis of Time-varying Modulated Signals Based on Improved Energy Separation by Iterative Generalized Demodulation*. Journal of Sound and Vibration, 2011. **330**(6): p. 1225-1243.
50. Norton, M. and D. Karczub, *Fundamentals of Noise and Vibration Analysis For Engineers*, 2003: CUP, UK.
51. Sakthivel, N.R., V. Sugumaran, and S. Babudevasenapati, *Vibration Based Fault Diagnosis of Monoblock Centrifugal Pump Using Decision Tree*. Expert Systems with Applications, 2010. **37**(6): p. 4040-4049.
52. Ahmed, M., et al. *Fault Detection And Diagnosis Using Principal Component Analysis of Vibration Data from A Reciprocating Compressor*. in *Control (CONTROL), 2012 UKACC International Conference on*. 2012.
53. Lin, J. and M.J. Zuo, *Gerbox Fault Diagnosis Using Adaptive Wavelet Filter*. Mechanical Systems and Signal Processing, 2003. **17**(6): p. 1259-1269.
54. Antoni, J. and R.B. Randall, *The Spectral Kurtosis: Application to The Vibratory Surveillance And Diagnostics of Rotating Machines*. Mechanical Systems and Signal Processing, 2006. **20**(2): p. 308-331.
55. Sawalhi, N., R.B. Randall, and H. Endo, *The Enhancement of Fault Detection And Diagnosis in Rolling Element Bearings Using Minimum Entropy Deconvolution Combined with Spectral Kurtosis*. Mechanical Systems and Signal Processing, 2007. **21**(6): p. 2616-2633.
56. Samanta, B. and K.R. Al-Balushi, *Artificial Neural Network Based Fault Diagnostics of Rolling Element Bearings Using Time-domain Features*. Mechanical Systems and Signal Processing, 2003. **17**(2): p. 317-328.
57. Rafiee, J., M.A. Rafiee, and P.W. Tse, *Application of Mother Wavelet Functions for Automatic Gear And Bearing Fault Diagnosis*. Expert Systems with Applications, 2010. **37**(6): p. 4568-4579.

58. Bechhoefer, E. and M. Kingsley. *A Review of Time Synchronous Average Algorithms*. in *Annual Conference of the Prognostics and Health Management Society*. 2009. San Diego.
59. Latuny, J., *A Sensitivity Comparison of Neuro-fuzzy Feature Extraction Methods from Bearing Failure Signals*, in *Mechanical Engineering* 2013, Curtin University: Bentley.
60. Halim, E.B., et al., *Time Domain Averaging Across All Scales: A Novel Method for Detection of Gearbox Faults.* Mechanical Systems and Signal Processing, 2008. **22**(2): p. 261-278.
61. Dalpiaz, G., A. Rivola, and R. Rubini, *Effectiveness and Sensitivity of Vibration Processing Techniques for Local Fault Detection in Gears.* Mechanical Systems and Signal Processing, 2000. **14**(3): p. 387-412.
62. Baillie, D.C. and J. Mathew, *A Comparison of Autoregressive Modeling Techniques for Fault Diagnosis of Rolling Element Bearings.* Mechanical Systems and Signal Processing, 1996. **10**(1): p. 1-17.
63. Garga, A.K., B.T. Elverson, and D.C. Lang, *AR Modeling With Dimension Reduction for Machinery Fault Classification, In: Critical Link: Diagnosis To Prognosis*1997: Haymarket. 299-308.
64. Zhan, Y., V. Makis, and A.K.S. Jardine, *Adaptive Model for Vibration Monitoring Of Rotating Machinery Subject To Random Deterioration.* Journal of quality in maintenance engineering, 2003. **9**: p. 351-375.
65. Wang, W.Q., F. Ismail, and M. Farid Golnaraghi, *Assessment of Gear Damage Monitoring Techniques Using Vibration Measurements.* Mechanical Systems and Signal Processing, 2001. **15**(5): p. 905-922.
66. Randall, R.B., *Vibration-based Condition Monitoring: Industrial and Automotive Applications*, 2011, Bruel & Kjaer: Naerum, Denmark.
67. Pan, M.-C. and W.-C. Tsao, *Using Appropriate IMFs for Envelope Analysis in Multiple Fault Diagnosis of Ball Bearings.* International Journal of Mechanical Sciences, 2013. **69**(0): p. 114-124.
68. Guo, Y., et al., *Envelope Order Tracking for Fault Detection in Rolling Element Bearings.* Journal of Sound and Vibration, 2012. **331**(25): p. 5644-5654.
69. Park, C.-S., Y.-C. Choi, and Y.-H. Kim, *Early Fault Detection in Automotive Ball Bearings Using the Minimum Variance Cepstrum.* Mechanical Systems and Signal Processing, 2013. **38**(2): p. 534-548.
70. Tsao, W.-C., et al., *An Insight Concept to Select Appropriate IMFs for Envelope Analysis Of Bearing Fault Diagnosis.* Measurement, 2012. **45**(6): p. 1489-1498.
71. Hauer, J.F., C.J. Demeure, and L.L. Scharf, *Initial Results in Prony Analysis of Power System Response Signals.* Power Systems, IEEE Transactions on, 1990. **5**(1): p. 80-89.
72. Chen, Z. and C.K. Mechefske, *Diagnosis of Machinery Fault Status Using Transient Vibration Signal Parameters.* Journal of Vibration and Control, 2002. **8**(3): p. 321-335.
73. Yumin, S. and K. Nezu, *Detection of Self-aligning Roller Bearing Fault by Asynchronous Adaptive Noise Cancelling Technology.* JSME International Series C: Dynamics, Control, Robotics, Design and Manufacturing, 1999. **42**(1): p. 33-43.
74. Logan, D., *Using the Correlation Method for the Detection of Faults in Rolling Ekement Bearings*, 1996, Monash University: Melbourne.

75. Mevel, L., L. Hermans, and H. Van Der Auweraer, *Application of a Subspace-Based Fault Detection Method to Industrial Structures.* Mechanical Systems and Signal Processing, 1999. **13**(6): p. 823-838.
76. Gelle, G., M. Colas, and G. Delaunay, *Blind Sources Separation Applied To Rotating Machines Monitoring By Acoustical and Vibrations Analysis.* Mechanical Systems and Signal Processing, 2000. **14**(3): p. 427-442.
77. Servière, C. and P. Fabry, *Principal Component Analysis and Blind Source Separation of Modulated Sources for Electro-Mechanical Systems Diagnostic.* Mechanical Systems and Signal Processing, 2005. **19**(6): p. 1293-1311.
78. Servière, C. and P. Fabry, *Blind Source Separation of Noisy Harmonic Signals for Rotating Machine Diagnosis.* Journal of Sound and Vibration, 2004. **272**(1–2): p. 317-339.
79. Ahmed, M., F. Gu, and A. Ball, *Feature Selection and Fault Classification of Reciprocating Compressors using a Genetic Algorithm and a Probabilistic Neural Network.* Journal of Physics: Conference Series, 2011. **305**(1).
80. Honarvar, F. and H.R. Martin, *New Statistical Moments for Diagnostics of Rolling Element Bearings.* Journal of Manufacturing Science and Engineering, 1997. **119**(3): p. 425-432.
81. McCormick, A.C. and A.K. Nandi, *Classification of the rotating machine condition using artificial neural networks.* Proceedings of the Institution of Mechanical Engineers, 1997. **211**(6): p. 439.
82. Wang, H. and P. Chen, *Sequential Condition Diagnosis for Centrifugal Pump System Using Fuzzy Neural Network.* Neural Information Processing, 2007. **11**(3).
83. Lei, Y. and M.J. Zuo, *Gear Crack Level Identification Based on Weighted K Nearest Neighbor Classification Algorithm.* Mechanical Systems and Signal Processing, 2009. **23**(5): p. 1535-1547.
84. Sakthivel, N.R., V. Sugumaran, and B.B. Nair, *Comparison of Decision Tree-Fuzzy and Rough Set-Fuzzy Methods for Fault Categorization of Mono-Block Centrifugal Pump.* Mechanical Systems and Signal Processing, 2010. **24**(6): p. 1887-1906.
85. Al Thobiani, A., F. Gu, and A. Ball. *The Monitoring Of Cavitation in Centrifugal Pumps Based On the Analysis of Vibro-Acoustic Measurements*. in *CM 2010 and MFPT 2010 : The Seventh International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*. 2010. Stratford-upon-Avon, UK.
86. Al-Hashmi, S.A., *Statistical Analysis of Acoustic Signal for Cavitation Detection*, in *International Journal of Emerging Technology and Advanced Engineering* 2013: Dubai, UAE.
87. Nasiri, M.R., M.J. Mahjoob, and H. Vahid-Alizadeh. *Vibration Signature Analysis for Detecting Cavitation in Centrifugal Pumps Using Neural Networks*. in *IEEE International Conference on Mechatronics*. 2011. Istanbul, Turkey.
88. Sun, Q., et al., *Pattern Recognition for Automatic Machinery Fault Diagnosis.* Journal of Vibration and Acoustics, 2004. **126**(2): p. 307-316.
89. Niu, X., L. Zhu, and H. Ding, *New Statistical Moments for the Detection of Defects in Rolling Element Bearings.* The International Journal of Advanced Manufacturing Technology, 2005. **26**(11-12): p. 1268-1274.

90. Hope, A.D. and L. Wang, *Fault Diagnosis of Centrifugal Pumps Using Vibration Analysis*, in *Twelfth International Cogress on Sound and Vibration* 2005: Lisbon, Portugal.
91. Rafiee, J., et al., *Intelligent Condition Monitoring of A Gearbox Using Artificial Neural Network.* Mechanical Systems and Signal Processing, 2007. **21**(4): p. 1746-1754.
92. Yang, H., J. Mathew, and L. Ma, *Vibration Feature Extraction Techniques for Fault Diagnosis of Rotating Machinery*, in *Asia-Pacific Vibration Conference* 2003: Gold Coast, Australia.
93. Duhamel, P. and M. Vetterli, *Fast Fourier Transforms: A Tutorial Review and A State of The Art.* Signal Processing, 1990. **19**(4): p. 259-299.
94. Baydar, N. and A. Ball, *Detection of Gear Deterioration under Varying Load Conditions by Using the Instantaneous Power Spectrum.* Mechanical Systems and Signal Processing, 2000. **14**(6): p. 907-921.
95. Yang, L., et al., *The Tunable Third-Order Optical Nonlinearities of a Diarylethene-Zinc Phthalocyanine Hybrid.* Dyes and Pigments, 2014. **102**(0): p. 251-256.
96. Li, C.J., J. Ma, and B. Hwang, *Bearing Condition Monitoring By Pattern Recognition Based On Coherence Analysis of Vibration.* Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 1996. **210**(3): p. 277-285.
97. Tandon, N. and A. Choudhury, *A Review of Vibration and Acoustic Measurement Methods for The Detection Of Defects In Rolling Element Bearings.* Tribology International, 1999. **32**(8): p. 469-480.
98. Capdessus, C., M. Sidahmed, and J.L. Lacoume, *Cyclostationary Processes: Application in Gear Faults Early Diagnosis.* Mechanical Systems and Signal Processing, 2000. **14**(3): p. 371-385.
99. McCormick, A.C. and A.K. Nandi, *Cyclostationarity In Rotating Machine Vibrations.* Mechanical Systems and Signal Processing, 1998. **12**(2): p. 225-242.
100. Klein, R., D. Ingman, and S. Braun, *Non-Stationary Signals: Phase-Energy Approach—Theory and Simulations.* Mechanical Systems and Signal Processing, 2001. **15**(6): p. 1061-1089.
101. Lee, S.K. and P.R. White, *Higher-Order Time–Frequency Analysis and Its Application to Fault Detection in Rotating Machinery.* Mechanical Systems and Signal Processing, 1997. **11**(4): p. 637-650.
102. Randall, R.B., *Frequency Analysis, Third Edition*, 1987, Bruel & Kjaer: Naerum, Denmark.
103. Ramirez, R.W., *The FFT Fundamentals and Concepts,* 1988, Prentice-Hall, Inc: Englewood Cliffs, New Jersey.
104. Cochran, W.T., et al., *What is the Fast Fourier Transform.* IEEE Transaction on Audio And Electroacoustics, 1967. **Au-15**(2).
105. Bergland, G.D. *A Guided Tour of the Fast Fourier Transform*. 1969.
106. Cooley, J.W. and J.W. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series.* Mathematics of Computation, 1965. **19**(90): p. 297-301.
107. Tan, C.Z. and M.S. Leong, *An Experimental Study of Cavitation Detection in a Centrifugal Pump Using Envelope Analysis.* Journal of System Design and Dynamics, 2008. **2**(1): p. 274-285.

108. Patidar, S. and P.K. Soni, *An Overview on Vibration Analysis Techniques for the Diagnosis of Rolling Element Bearing Faults.* International Journal of Engineering Trends and Technology, 2013. **4**(5).
109. Randall, R.B. and J. Antoni, *Rolling Element Bearing Diagnostics—A tutorial.* Mechanical Systems and Signal Processing, 2011. **25**(2): p. 485-520.
110. McFadden, P.D. and J.D. Smith, *Vibration Monitoring of Rolling Element Bearings by the High-Frequency Resonance Technique - A Review.* Tribology International, 1984. **17**(1): p. 3-10.
111. Mollazade, K., et al., *Vibration-Based Fault Diagnosis of Hydraulic Pump of Tractor Steering System by Using Energy Technique.* Modern Applied Science, 2009. **3**(6).
112. Ho, D. and R.B. Randall, *Optimisation of Bearing Diagnostic Techniques Using Simulated and Actual Bearing Fault Signals.* Mechanical Systems and Signal Processing, 2000. **14**(5): p. 763-788.
113. Rai, V.K. and A.R. Mohanty, *Bearing Fault Diagnosis Using FFT of Intrinsic Mode Functions in Hilbert–Huang Transform.* Mechanical Systems and Signal Processing, 2007. **21**(6): p. 2607-2615.
114. Farokhzad, S., et al., *Artificial Neural Network Based Classification of Faults in Centrifugal Pump.* Journal of Vibroengineering, 2012. **14**(4): p. 1734.
115. Farokhzad, S., *Vibration Based Fault Detection of Centrifugal Pump by Fast Fourier Transform and Adaptive Neuro-Fuzzy Inference System.* Journal of Mechanical Engineering and Technology, 2013. **1**(3): p. 82-87.
116. Moosavian, A., et al., *Comparison of Two Classifiers; K-Nearest Neighbor and Artificial Neural Network, for Fault Diagnosis on a Main Engine Journal-Bearing.* Shock and Vibration, 2013. **20**(2).
117. Chun-Lin, L. *A Tutorial of the Wavelet Transform*. 2010.
118. Yan, R., R.X. Gao, and X. Chen, *Wavelets for Fault Diagnosis of Rotary Machines: A Review with Applications.* Signal Processing, 2014. **96, Part A**(0): p. 1-15.
119. Saravanan, N. and K.I. Ramachandran, *Incipient Gear Box Fault Diagnosis Using Discrete Wavelet Transform (DWT) for Feature Extraction and Classification Using Artificial Neural Network (ANN).* Expert Systems with Applications, 2010. **37**(6): p. 4168-4181.
120. Graps, A., *An Introduction to Wavelets.* IEEE Computational Science and Engineering, 1995. **2**(2): p. 50-61.
121. Vetterli, M. and C.Harley, *Wavelets and Filter Banks: Theory and Design.* IEEE Transaction on Signal Processing, 1992. **40**: p. 2207-2232.
122. Misiti, M., Y. Misiti, and G. Oppenheim, *Wavelets and Their Applications*, 2006, ISTE.
123. Muralidharan, V. and V. Sugumaran, *Rough Set Based Rule Learning and Fuzzy Classification of Wavelet Features for Fault Diagnosis of Monoblock Centrifugal Pump.* Measurement, 2013. **46**(9): p. 3057-3063.
124. Muralidharan, V. and V. Sugumaran, *A Comparative Study of Naïve Bayes Classifier and Bayes Net Classifier for Fault Diagnosis of Monoblock Centrifugal Pump Using Wavelet Analysis.* Applied Soft Computing, 2012. **12**(8): p. 2023-2029.
125. Barford, L.A., R.S. Fazzio, and D.R. Smith *An Introduction to Wavelets*. 1992.
126. Mallat, S., *A Wavelet Tour of Signal Processing* 1989: Academic Press.

127. Trendafilova, I., *An Automated Procedure for Detection and Identification of Ball Bearing Damage Using Multivariate Statistics and Pattern Recognition.* Mechanical Systems and Signal Processing, 2010. **24**(6): p. 1858-1869.
128. Wu, J.-D. and C.-H. Liu, *An Expert System for Fault Diagnosis in Internal Combustion Engines Using Wavelet Packet Transform and Neural Network.* Expert Systems with Applications, 2009. **36**(3, Part 1): p. 4278-4286.
129. Mallat, S.G., *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1989. **11**(7): p. 674-693.
130. Wu, J.-D. and J.-M. Kuo, *An Automotive Generator Fault Diagnosis System Using Discrete Wavelet Transform and Artificial Neural Network.* Expert Systems with Applications, 2009. **36**(6): p. 9776-9783.
131. Wu, J.-D. and C.-H. Liu, *Investigation of Engine Fault Diagnosis Using Discrete Wavelet Transform and Neural Network.* Expert Systems with Applications, 2008. **35**(3): p. 1200-1213.
132. Leducq, D. *Hydraulic Noise Diagnostics Using Wavelet Analysis* in *Proceeding of the International Conference on Noise Control Engineering*. 1990.
133. Prabhakar, S., A.R. Mohanty, and A.S. Sekhar, *Application of Discrete Wavelet Transform For Detection of Ball Bearing Race Faults.* Tribology International, 2002. **35**(12): p. 793-800.
134. Lou, X. and K.A. Loparo, *Bearing Fault Diagnosis Based on Wavelet Transform and Fuzzy Inference.* Mechanical Systems and Signal Processing, 2004. **18**(5): p. 1077-1095.
135. Gao, Y., X. Kong, and Q. Zhang. *Wavelet Approach for Performance Monitoring and Diagnosis of A Hydraulic Pump* in *Proceeding of the 6th JFPS International Symposium on Fluid Power*. 2005. Tsukuba, Japan.
136. Lattin, J., J.D. Carroll, and P.E. Green, *Analyzing Multivariate Data.* 2003, CA, USA: Thomson Learning, Inc.
137. Nguyen, V.H. and J.-C. Golinval, *Fault Detection Based on Kernel Principal Component Analysis.* Engineering Structures, 2010. **32**(11): p. 3683-3691.
138. Mujica, L., et al., *Q-statistic and T^2-statistic PCA-based Measures for Damage Assessment in Structures.* Structural Health Monitoring, 2011. **10**(5): p. 539-553.
139. Chiang, L.H. and L.F. Colegrove, *Industrial Implementation of On-line Multivariate Quality Control.* Chemometrics and Intelligent Laboratory Systems, 2007. **88**(2): p. 143-153.
140. Tao, E.P., et al., *Fault Diagnosis Based on PCA for Sensors of Laboratorial Wastewater Treatment Process.* Chemometrics and Intelligent Laboratory Systems, 2013. **128**(0): p. 49-55.
141. Jolliffe, I.T., *Principal Component Analysis.*, 2002, Springer: Secaucus, NJ, USA.
142. Chen, Y. and L. Lan, *A Fault Detection Technique for Air-Source Heat Pump Water Chiller/Heaters.* Energy and Buildings, 2009. **41**(8): p. 881-887.
143. Abdi, H. and L.J. Williams *Principal Component Analysis*. 2010.
144. Liying, J., et al. *Improved Confidence Limits of T^2 Statistic for Monitoring Batch Processes*. in *Control and Decision Conference (CCDC), 2012 24th Chinese*. 2012.

145. Benaicha, A., et al., *Determination of Principal Component Analysis Models for Sensor Fault Detection and Isolation.* International Journal of Control, Automation, and Systems, 2013. **11**(2): p. 296-305.
146. MacGregor, J.F. and T. Kourti, *Statistical Process Control of Multivariate Processes.* Control Engineering Practice, 1995. **3**(3): p. 403-414.
147. Harrou, F., et al., *Statistical Fault Detection Using PCA-based GLR Hypothesis Testing.* Journal of Loss Prevention in the Process Industries, 2013. **26**(1): p. 129-139.
148. Joe Qin, S., *Statistical Process Monitoring: Basics and Beyond.* Journal of Chemometrics, 2003. **17**(8-9): p. 480-502.
149. Li, W., et al., *Feature Extraction and Classification of Gear Faults Using Principal Component Analysis.* Journal of quality in maintenance engineering, 2003. **9**(2).
150. Malhi, A. and R.X. Gao, *PCA-based Feature Selection Scheme for Machine Defect Classification.* Instrumentation and Measurement, IEEE Transactions on, 2004. **53**(6): p. 1517-1525.
151. Sun, W., J. Chen, and J. Li, *Decision Tree and PCA-based Fault Diagnosis of Rotating Machinery.* Mechanical Systems and Signal Processing, 2007. **21**(3): p. 1300-1317.
152. Lu, S. and M. Li. *Bearing Fault Diagnosis Based on PCA and SVM*. in *International Conference on Mechatronics and Automation*. 2007.
153. Trendafilova, I., M.P. Cartmell, and W. Ostachowicz, *Vibration-Based Damage Detection in an Aircraft Wing Scaled Model Using Principal Component Analysis and Pattern Recognition.* Journal of Sound and Vibration, 2008. **313**(3–5): p. 560-566.
154. He, Q., et al., *Machine Condition Monitoring Using Principal Component Representations.* Mechanical Systems and Signal Processing, 2009. **23**(2): p. 446-466.
155. Li, Z., et al. *Gear Faults Diagnosis Based on Wavelet-AR Model and PCA*. in *International Conference on Image Processing and Pattern Recognition in Industrial Engineering*. 2010.
156. Pirra, M., et al., *PCA Algorithm for Detection, Localisation and Evolution of Damages in Gearbox Bearings.* Journal of Physics: Conference Series, 2011. **305**(1): p. 012019.
157. Zimroz, R. and A. Bartkowiak, *Investigation on Spectral Structure of Gearbox Vibration Signals by Principal Component Analysis for Condition Monitoring Purposes.* Journal of Physics: Conference Series, 2011. **305**(1): p. 012075.
158. Abouhnik, A., et al., *Novel Approach to Rotating Machinery Diagnostics Based on Principal Component and Residual Matrix Analysis.* ISRN Mechanical Engineering, 2012. **2012**: p. 7.
159. Zhao, X., M.J. Zuo, and T.H. Patel, *Generating an Indicator for Pump Impeller Damage Using Half and Full Spectra, Fuzzy Preference-Based Rough Sets and PCA.* Measurement Science and Technology, 2012. **23**(4): p. 045607.
160. Liying, J., et al. *Fault Detection of Rolling Element Bearing Based on Principal Component Analysis*. in *Control and Decision Conference (CCDC), 2012 24th Chinese*. 2012.
161. Dong, S. and T. Luo, *Bearing Degradation Process Prediction Based on the PCA and Optimized LS-SVM Model.* Measurement, 2013. **46**(9): p. 3143-3152.

162. Gharavian, M.H., et al., *Comparison of FDA-based and PCA-based Features in Fault Diagnosis of Automobile Gearboxes.* Neurocomputing, 2013. **121**(0): p. 150-159.
163. Xi, J., Y. Han, and R. Su. *New Fault Diagnosis Method for Rolling Bearing Based on PCA*. in *Control and Decision Conference (CCDC), 2013 25th Chinese*. 2013.
164. Sakthivel, N.R., et al., *Comparison of Dimensionality Reduction Techniques for the Fault Diagnosis of Mono Block Centrifugal Pump Using Vibration Signals.* Engineering Science and Technology, an International Journal, 2014. **17**(1): p. 30-38.
165. Shao, R., et al., *The Fault Feature Extraction and Classification of Gear Using Principal Component Analysis and Kernel Principal Component Analysis Based On the Wavelet Packet Transform.* Measurement, 2014. **54**(0): p. 118-132.
166. Widodo, A. and B.-S. Yang, *Wavelet Support Vector Machine for Induction Machine Fault Diagnosis Based On Transient Current Signal.* Expert Systems with Applications, 2008. **35**(1–2): p. 307-316.
167. Casimir, R., et al., *The Use of Features Selection and Nearest Neighbors Rule for Faults Diagnostic in Induction Motors.* Engineering Applications of Artificial Intelligence, 2006. **19**(2): p. 169-177.
168. Pandya, D.H., S.H. Upadhyay, and S.P. Harsha, *Fault Diagnosis of Rolling Element Bearing with Intrinsic Mode Function of Acoustic Emission Data Using APF-KNN.* Expert Systems with Applications, 2013. **40**(10): p. 4137-4145.
169. Bouguerne, A., et al. *Classification of Induction Machine Faults by K-nearest neighbors*. in *7th International Conference on Electrical and Electronics Engineering (ELECO)* 2011. Bursa.
170. Latuny, J. and R. Entwistle. *Bearing Fault Analysis through the Application of ANFIS and Vector Array Indicators Based on Statistical Parameters of Wavelet Transformation Component*. in *Australasian Congress on Applied Mechanics (ACAM 6) Conference*. 2010.
171. Jiang, S., et al., *An Improved K-nearest-neighbor Algorithm for Text Categorization.* Expert Systems with Applications, 2012. **39**(1): p. 1503-1509.
172. Lee, J., et al., *Bearing Data Set*, U.o.C. Center for Intelligent Maintenance System (IMS), Editor 2007: NASA Ames, Moffett Field, CA.

Appendix A – MATLAB Script Code for Data Acquisition Using NI-9234

```
%Script to run data acquisition using National Instrument NI 9234
%Created: Dec 2013, Berli Kamiel

clear all;
clc;
close all;

plot_graph = 1;
simpan_data = 1;
tic;


 s = daq.createSession('ni');
 s.DurationInSeconds = 1;
 Dur = s.DurationInSeconds;
 s.Rate = 48192; % sample rate 48192 / sec
 s.addAnalogInputChannel('dev1', 'ai0', 'Accelerometer');
 s.addAnalogInputChannel('dev1', 'ai1', 'Accelerometer');
 s.addAnalogInputChannel('dev1', 'ai2', 'Accelerometer');
 s.addAnalogInputChannel('dev1', 'ai3', 'Accelerometer');

 s.Channels(1).Sensitivity = 95.83E-3;
 s.Channels(2).Sensitivity = 95.90E-3;
 s.Channels(3).Sensitivity = 96.93E-3;
 s.Channels(4).Sensitivity = 94.50E-3;



for i=1:120

    data = s.startForeground(); % start recording vibration data

    data_ch1 = data(:,1);
    data_ch2 = data(:,2);
    data_ch3 = data(:,3);
    data_ch4 = data(:,4);

    if plot_graph == 1

    figure(1)
    subplot(4,1,1)
    plot(data_ch1,'r-');
    xlabel('Samples');
    ylabel('Signal (Volts)');
    title('Acceleration data: Channel 1');

    subplot(4,1,2)
    plot(data_ch2,'b-');
    xlabel('Samples');
    ylabel('Signal (Volts)');
    title('Acceleration data: Channel 2');

    subplot(4,1,3)
    plot(data_ch3,'k-');
    xlabel('Samples');
    ylabel('Signal (Volts)');
    title('Acceleration data: Channel 2');
```

```
        subplot(4,1,4)
        plot(data_ch4,'g-');
        xlabel('Samples');
        ylabel('Signal (Volts)');
        title('Acceleration data: Channel 2');

        fft_data_ch1 = abs(fft(data_ch1))/length(data_ch1);
        [ymax1,xmax1] = max(fft_data_ch1);

        fft_data_ch2 = abs(fft(data_ch2))/length(data_ch2);
        [ymax2,xmax2] = max(fft_data_ch2);

        fft_data_ch3 = abs(fft(data_ch3))/length(data_ch3);
        [ymax3,xmax3] = max(fft_data_ch3);

        fft_data_ch4 = abs(fft(data_ch4))/length(data_ch4);
        [ymax4,xmax4] = max(fft_data_ch4);

        t = 1:1:length(fft_data_ch1)/2;
        max_freq1 = num2str(xmax1);
        max_freq2 = num2str(xmax2);
        max_freq3 = num2str(xmax3);
        max_freq4 = num2str(xmax4);

        figure(2)
        subplot(4,1,1)
        plot(t/Dur,fft_data_ch1(1:1:length(fft_data_ch1)/2),'r-', xmax1,
ymax1, 'bo');
        axis([0 (length(fft_data_ch1)/2+length(fft_data_ch1)*0.001) 0
ymax1+ymax1*0.1]);
        judul = ['FFT of signal: Max Freq: ' max_freq1 ,' Hz'];
        title(judul);
        xlabel('Frequency (Hz)');
        ylabel('Magnitude');
        grid on;

        subplot(4,1,2)
        plot(t/Dur,fft_data_ch2(1:1:length(fft_data_ch2)/2),'b-', xmax2,
ymax2, 'ro');
        axis([0 (length(fft_data_ch2)/2+length(fft_data_ch2)*0.001) 0
ymax2+ymax2*0.1]);
        judul = ['FFT of signal: Max Freq: ' max_freq2 ,' Hz'];
        title(judul);
        xlabel('Frequency (Hz)');
        ylabel('Magnitude');
        grid on;

        subplot(4,1,3)
        plot(t/Dur,fft_data_ch3(1:1:length(fft_data_ch3)/2),'k-', xmax3,
ymax3, 'ro');
        axis([0 (length(fft_data_ch3)/2+length(fft_data_ch3)*0.001) 0
ymax3+ymax3*0.1]);
        judul = ['FFT of signal: Max Freq: ' max_freq3 ,' Hz'];
        title(judul);
        xlabel('Frequency (Hz)');
        ylabel('Magnitude');
        grid on;

        subplot(4,1,4)
        plot(t/Dur,fft_data_ch4(1:1:length(fft_data_ch4)/2),'k-', xmax4,
ymax4, 'ro');
        axis([0 (length(fft_data_ch4)/2+length(fft_data_ch4)*0.001) 0
ymax4+ymax4*0.1]);
        judul = ['FFT of signal: Max Freq: ' max_freq4 ,' Hz'];
        title(judul);
        xlabel('Frequency (Hz)');
        ylabel('Magnitude');
```

```
         grid on;

         end

      % helpdoc datestr
     tanggal = datestr(now);
     tgl_date = datestr(tanggal,7);
     tgl_month = datestr(tanggal,5);
     tgl_year = datestr(tanggal,10);
     tgl_time = datestr(tanggal,13);
     tgl_ampm = datestr(tanggal, 15);
     hour = tgl_ampm(:,1:2);
     min = tgl_ampm(:,4:5);
     sec = tgl_time(:,7:8);
     format_tgl = [tgl_date tgl_month tgl_year '_' hour min sec];

     rootname = 'c:\mat_data\';  % drive tujuan dan nama file
     extension = '.mat';     % ekstension utk nama file

     namafile =
[rootname,'ch123_',tgl_date,tgl_month,'_',num2str(i),extension];

         data_all = [data_ch1 data_ch2 data_ch3 data_ch4];

         if simpan_data == 1
            eval(['save ', namafile ,' data_all']);
         end

         pesan = ['Acquiring and saving data at loop number: ',num2str(i)];
         disp(pesan)

     end

     toc;
```

Appendix B − Visualisation of Correlation Matrices for All Channels Using sym4, sym8 and sym12

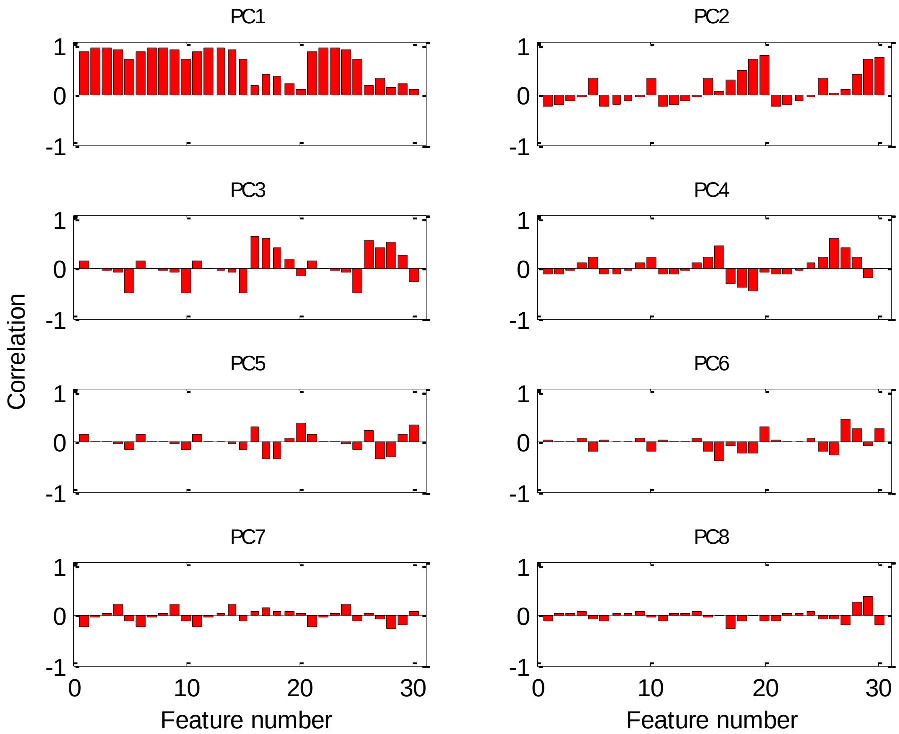


Figure B. 1 Correlation between principal components (PCs) and the features of channel 1 (using sym4 decomposition)

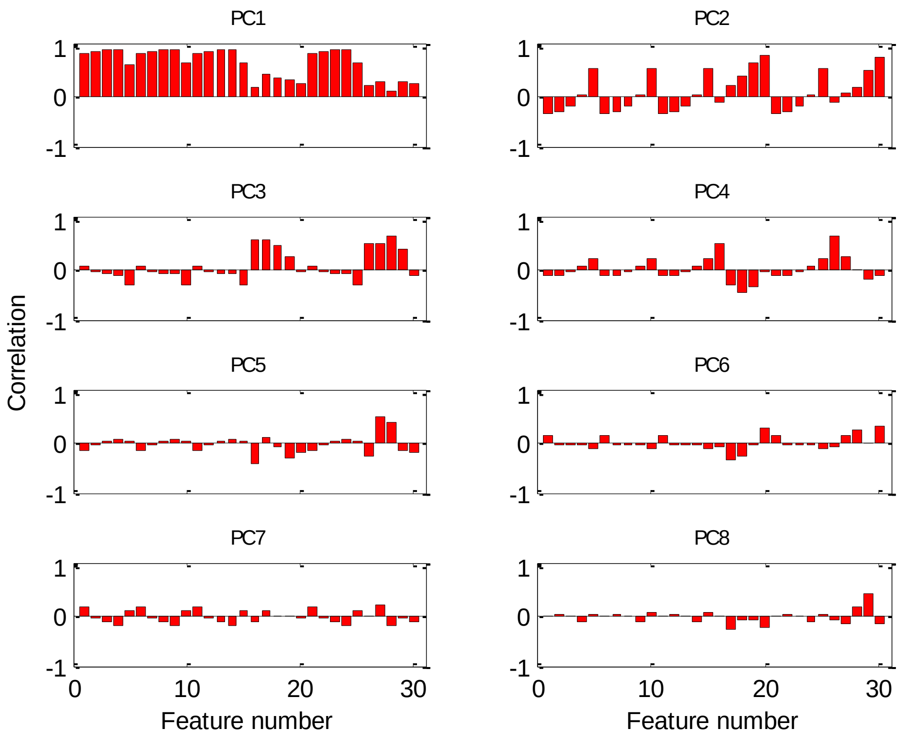


Figure B. 2 Correlation between principal components (PCs) and the features of channel 1 (using sym8 decomposition)

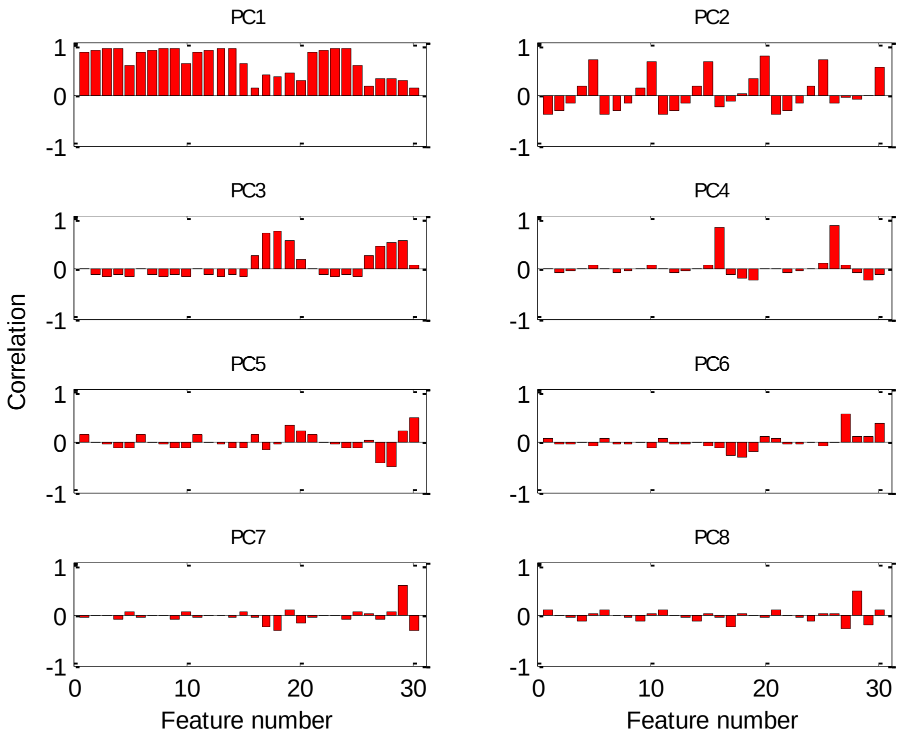


Figure B. 3 Correlation between principal components (PCs) and the features of channel 1 (using sym12 decomposition)

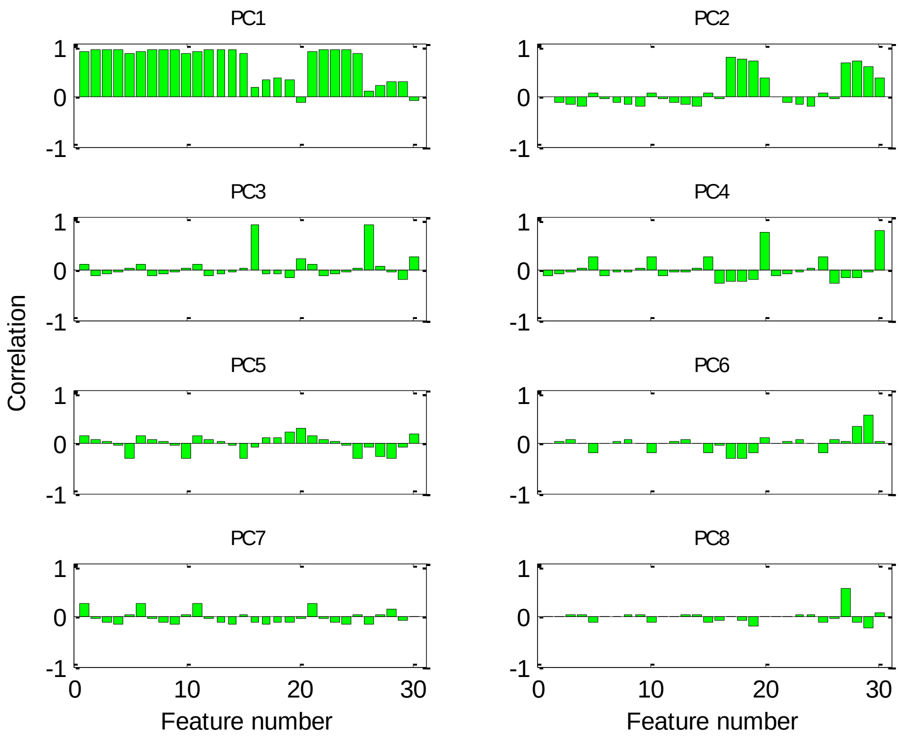


Figure B. 4 Correlation between principal components (PCs) and the features of channel 2 (using sym4 decomposition)

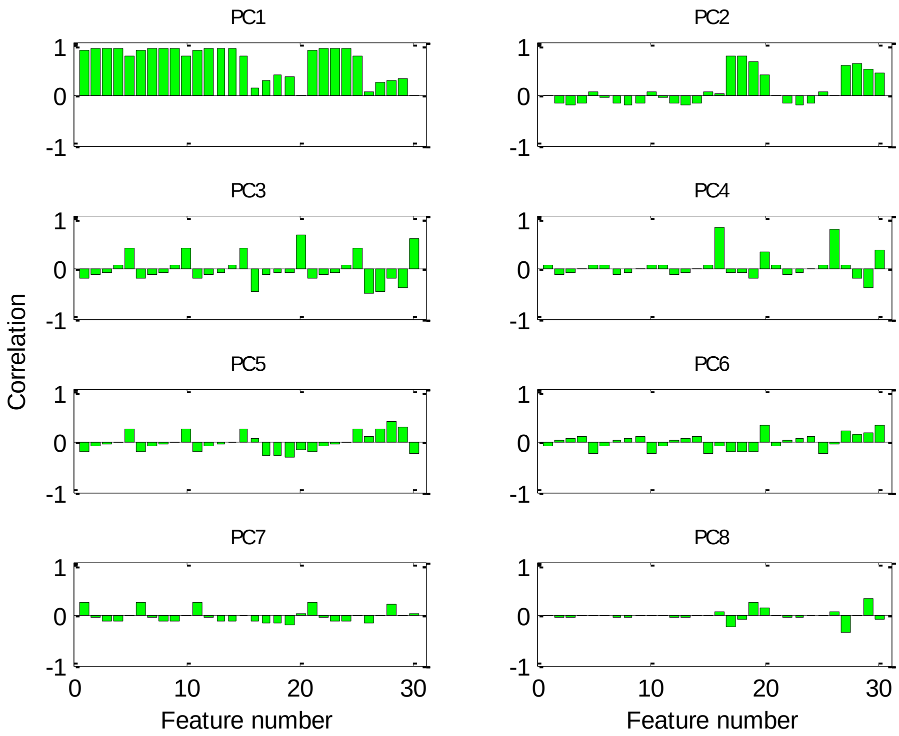


Figure B. 5 Correlation between principal components (PCs) and the features of channel 2 (using sym8 decomposition)

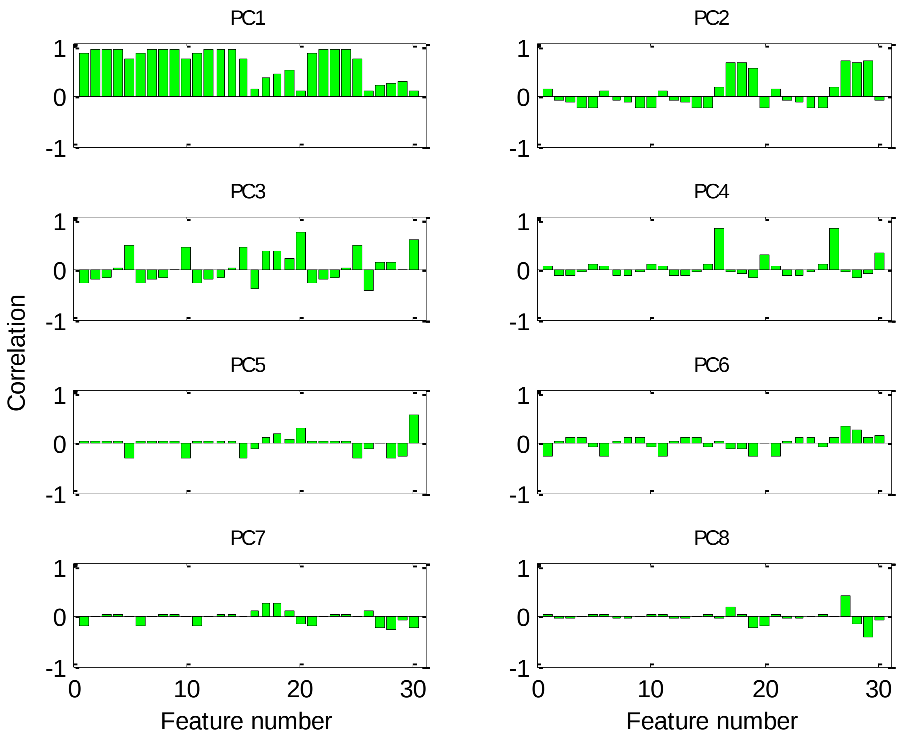


Figure B. 6 Correlation between principal components (PCs) and the features of channel 2 (using sym12 decomposition)

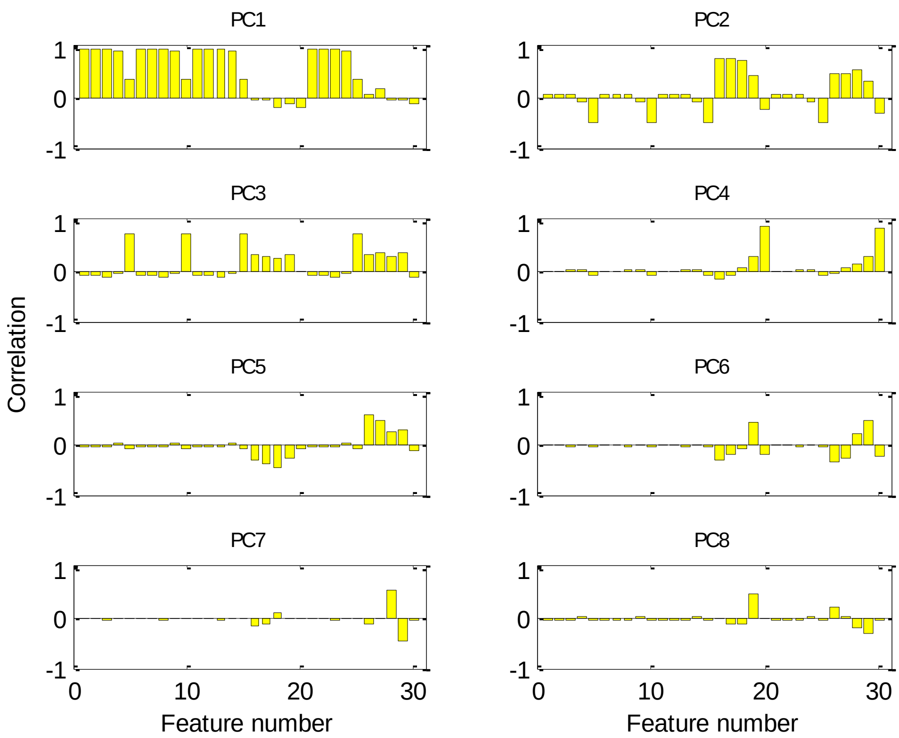


Figure B. 7 Correlation between principal components (PCs) and the features of channel 3 (using sym4 decomposition)

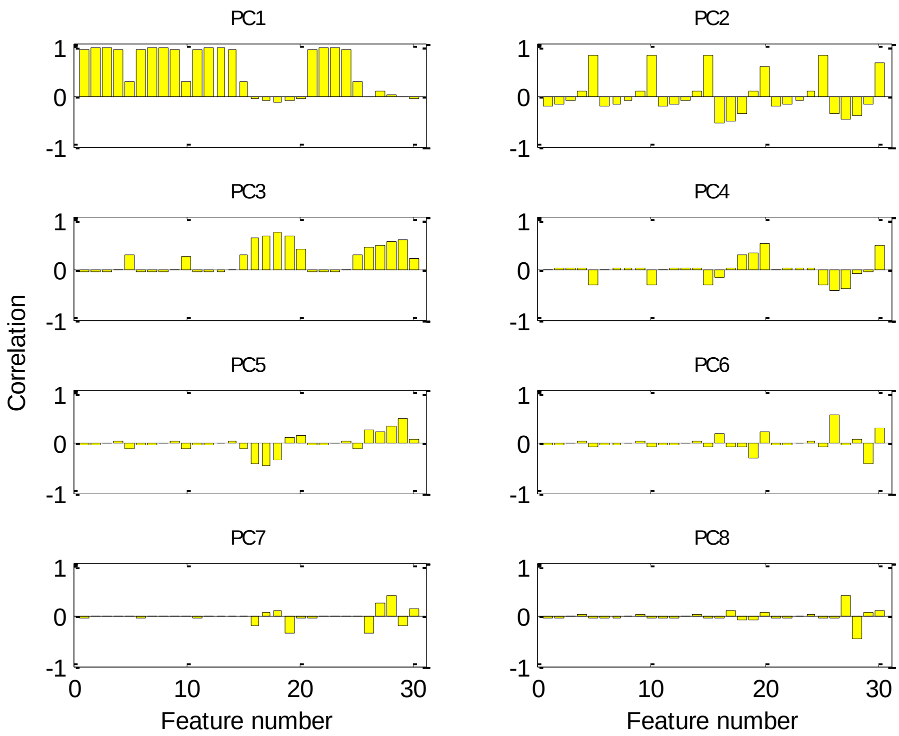


Figure B. 8 Correlation between principal components (PCs) and the features of channel 3 (using sym8 decomposition)

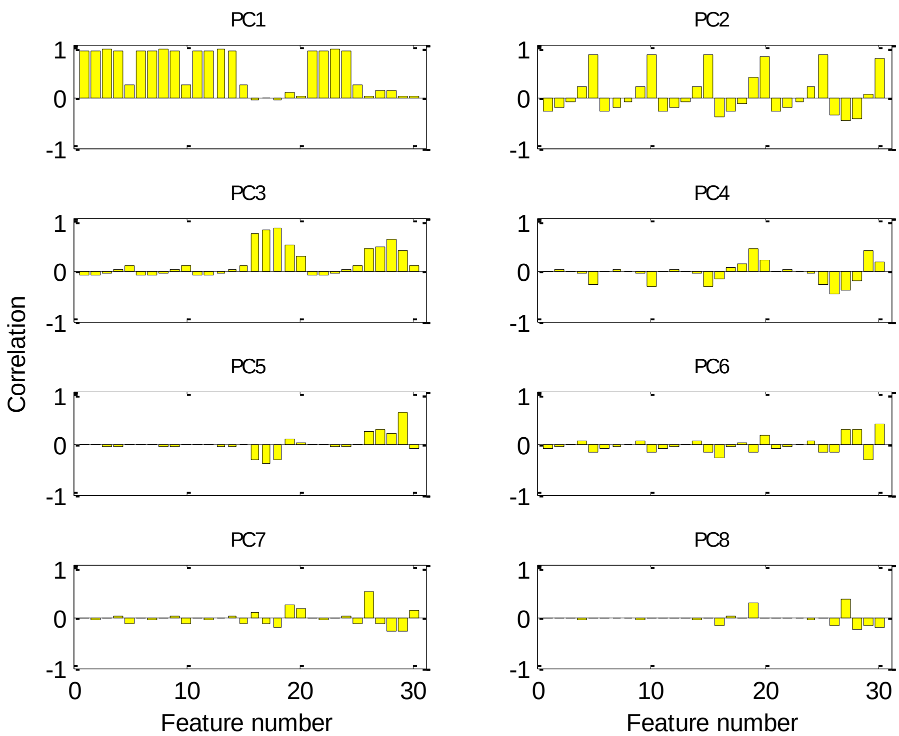


Figure B. 9 Correlation between principal components (PCs) and the features of channel 3 (using sym12 decomposition)

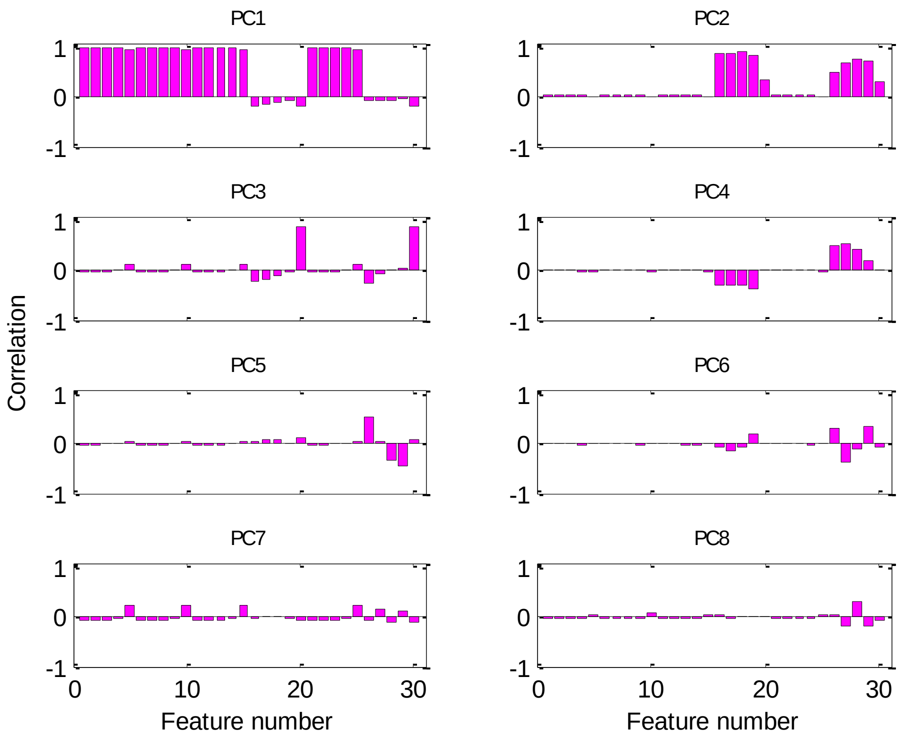


Figure B. 10 Correlation between principal components (PCs) and the features of channel 4 (using sym4 decomposition)

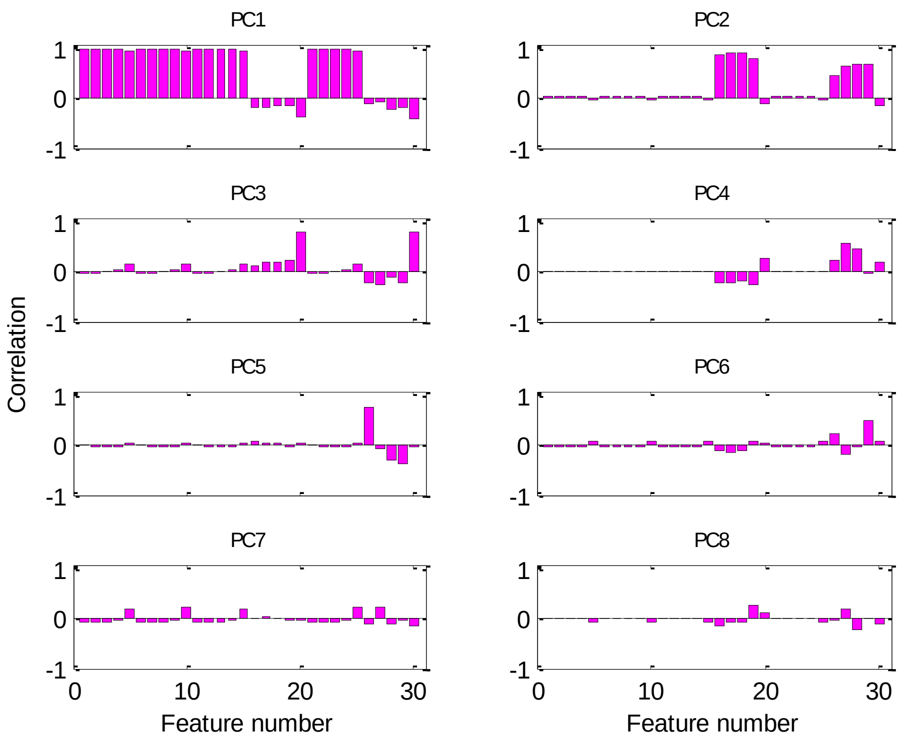


Figure B. 11 Correlation between principal components (PCs) and the features of channel 4 (using sym8 decomposition)

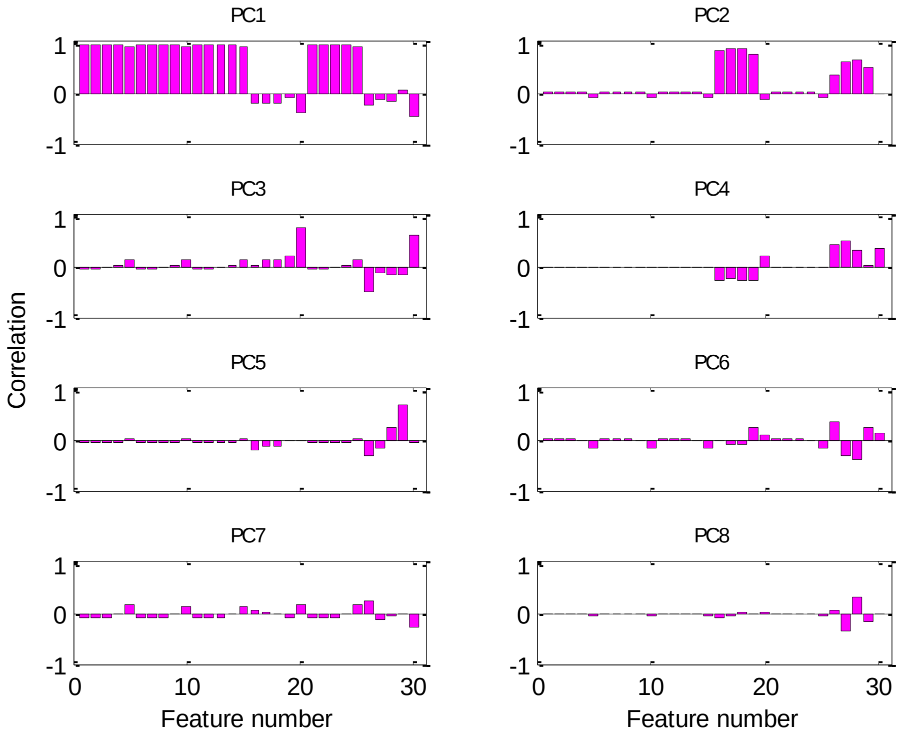


Figure B. 12 Correlation between principal components (PCs) and the features of channel 4 (using sym8 decomposition)

Appendix C – Integrated Framework for Fault Diagnosis Code

Module 1 – PCA Model Building

```
% Script for extracting statistical features from decomposed time-domain
% and then use the features to build PCA model
% Created: Feb 2014, Berli Kamiel

      clear all;
      clc;
      close all;


      decomp_level=5;
      wave='sym12';

      data_awal=1;
      data_akhir=70;

      %prepare matrix zeros for features' vector

      vector_ch1=zeros(data_akhir+1-data_awal,30);
      vector_ch2=zeros(data_akhir+1-data_awal,30);
      vector_ch3=zeros(data_akhir+1-data_awal,30);
      vector_ch4=zeros(data_akhir+1-data_awal,30);

      d1=0;

      for d=data_awal:data_akhir

          signal_in=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_normal\ch123_0412_',int2str(d),'.mat'];
          load (signal_in)

          pesan= ['Processing data sequence no: ',signal_in];
          disp(pesan);


          ch1 = data_all(:,1);
          ch2 = data_all(:,2);
          ch3 = data_all(:,3);
          ch4 = data_all(:,4);


          [ch1cA_0,~]=dwt(ch1,wave);
          [ch2cA_0,~]=dwt(ch2,wave);
          [ch3cA_0,~]=dwt(ch3,wave);
          [ch4cA_0,~]=dwt(ch4,wave);

         for K=1:4

            for I=1:decomp_level


eval(['[ch',int2str(K),'cA_',int2str(I),',~]=dwt(ch',int2str(K),'cA_',int2s
tr(I-1),',wave);']);

               end
         end
```

```
%calculate features from wavelet decomposed signal and put into
corresponding vector_ch

          d1=d1+1;

          for i=1:4
              for j=1:5

                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j),') =
energi2(ch',int2str(i),'cA_',int2str(j),',',int2str(j),');'])
                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+5),') =
std(ch',int2str(i),'cA_',int2str(j),');'])
                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+10),') =
rms(ch',int2str(i),'cA_',int2str(j),',0);'])
                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+15),') =
kurtosis(ch',int2str(i),'cA_',int2str(j),');'])
                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+20),') =
var(ch',int2str(i),'cA_',int2str(j),');'])
                eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+25),') =
crest(ch',int2str(i),'cA_',int2str(j),');'])

              end
          end
      end

      vector_ch1234(:,:,1)=vector_ch1;
      vector_ch1234(:,:,2)=vector_ch2;
      vector_ch1234(:,:,3)=vector_ch3;
      vector_ch1234(:,:,4)=vector_ch4;

      for i=1:4

      eval ( [
'[LOADING_ch',int2str(i),',SCORE_ch',int2str(i),',latent_ch',int2str(i),',T
2_ch',int2str(i),'] =
princomp(zscore(vector_ch1234(:,:,',int2str(i),')));']);
      eval ( [ '[norm_vector_ch1234(:,:,',int2str(i),'),MU(i,:),SIGMA(i,:)]
= zscore(vector_ch1234(:,:,',int2str(i),'));' ])
      eval ( ['loading(:,:,',int2str(i),')=LOADING_ch',int2str(i),';'])
      eval ( ['score(:,:,',int2str(i),')=SCORE_ch',int2str(i),';'])
      eval ( ['latent(:,',int2str(i),')=latent_ch',int2str(i),';'])

      end

      figure(1)
      subplot(2,1,1)
      pareto(latent(1:30,1))

      subplot(2,1,2)
      pareto(latent(1:30,2))

      figure(2)
      subplot(2,1,1)
      pareto(latent(1:30,3))

      subplot(2,1,2)
      pareto(latent(1:30,4))

      figure(3)
      subplot(1,2,1)
      pareto(latent(1:30,1))
```

```
      subplot(1,2,2)
      pareto(latent(1:30,4))


      save ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\baseline_PCAmodel_normal_35hz_sym12.mat','loading','score','latent',
'MU','SIGMA');

      %*****************************
      %*****************************

      clear all;

      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\baseline_PCAmodel_normal_35hz_sym12.mat');

      for i=1:4

      eval ( ['lamda',int2str(i),'=diag(latent(:,',int2str(i),'));' ])
      CPV(:,i)=cumsum(latent(:,i))/sum(latent(:,i))*100;

      end

      % Building PCA model
      % ****************************************************************

      % criteria how many PCs retained in the PCA model

      totalVariance=95;
      for i=1:4
          s=find( CPV(:,i)>totalVariance);
          idx(i)=s(1);
      end

      % truncating lamda and loading matrix based on number of retained PCs

      for i=1:4
          eval ( [
'PClamda_',int2str(i),'=lamda',int2str(i),'(1:idx(',int2str(i),'),1:idx(',i
nt2str(i),'));'])
          eval ( [
'PCloading_',int2str(i),'=loading(:,1:idx(',int2str(i),'),',int2str(i),');'
])
      end

      %PCAmodel created*****************************************************

      filename='C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysi
s chapter 7\mat file sec7_6\truncating_PCAmodel_normal_35hz_sym12.mat';

      save(filename)
```

Module 2 – Fault Detection

```
% Script for fault detection using T2 and Q-statistic
% Created: Feb 2014, Berli Kamiel

clear all;
clc;
close all;

filename2='C:\Users\13212047\Documents\MATLAB\script_thesis2014\anal
ysis chapter 7\mat file
sec7_6\truncating_PCAmodel_normal_35hz_sym12.mat';

load(filename2)



% converting PClamda and PCloading matrix

for i=1:4
    eval (
['PCloading_T_',int2str(i),'=transpose(PCloading_',int2str(i),');'])
    eval (
['invPClamda_',int2str(i),'=inv(PClamda_',int2str(i),');'])
    eval ( ['[r',int2str(i),',~]=size(PCloading_',int2str(i),');'])
    eval ( [ 'E',int2str(i),'=eye(r',int2str(i),');'])
end



% input new data

newdata_awal=71;
newdata_akhir=120;

decomp_level=5;
wave='sym12';

d1=0;
for k=newdata_awal:newdata_akhir


    %newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_normal\ch123_0412_',int2str(k),'.mat'];
    %1newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_normal_cavitation\ch123_0412_',int2str(k),'.mat'];
    %2newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_impeller\ch123_0412_',int2str(k),'.mat'];

%3newdata=['C:\mat_data\Data_2014\22_03_2014_35Hz_bearing\ch123_2203_',in
t2str(k),'.mat'];
    %4newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_normal_blockage\ch123_0412_',int2str(k),'.mat'];
    %5newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_impeller_cavitation\ch123_0412_',int2str(k),'.mat']
;
    %6newdata=['C:\mat_data\Data_2013\sampling rate
48192\04_12_2013_35hz_impeller_blockage\ch123_0412_',int2str(k),'.mat'];

newdata=['C:\mat_data\Data_2014\22_03_2014_35hz_bearing_cav\ch123_2203_',
int2str(k),'.mat'];

    load(newdata)


    pesan= ['Processing new data sequence no: ',newdata];
    disp(pesan);
```

```
        ch1 = data_all(:,1);
        ch2 = data_all(:,2);
        ch3 = data_all(:,3);
        ch4 = data_all(:,4);


        [ch1cA_0,~]=dwt(ch1,wave);
        [ch2cA_0,~]=dwt(ch2,wave);
        [ch3cA_0,~]=dwt(ch3,wave);
        [ch4cA_0,~]=dwt(ch4,wave);

       for K=1:4

          for I=1:decomp_level


eval(['[ch',int2str(K),'cA_',int2str(I),',~]=dwt(ch',int2str(K),'cA_',int
2str(I-1),',wave);']);

           end
       end



     %calculate features from wavelet decomposed signal and put into
corresponding vector_ch

        d1=d1+1;

        for i=1:4
            for j=1:5

              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j),') =
energi2(ch',int2str(i),'cA_',int2str(j),',',int2str(j),');'])
              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+5),') =
std(ch',int2str(i),'cA_',int2str(j),');'])
              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+10),') =
rms(ch',int2str(i),'cA_',int2str(j),',0);'])
              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+15),') =
kurtosis(ch',int2str(i),'cA_',int2str(j),');'])
              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+20),') =
var(ch',int2str(i),'cA_',int2str(j),');'])
              eval( [
'vector_ch',int2str(i),'(',int2str(d1),',',int2str(j+25),') =
crest(ch',int2str(i),'cA_',int2str(j),');'])

            end
        end

        % normalized vector_ch
        for i=1:4
            eval ( ['step_vector_ch',int2str(i),' =
bsxfun(@minus,vector_ch',int2str(i),',MU(i,:));'])
            eval ( ['norm_vector_ch',int2str(i),' =
bsxfun(@rdivide,step_vector_ch',int2str(i),',SIGMA(i,:));'])
        end

     %vector_ch1234(:,:,1)=vector_ch1;
     %vector_ch1234(:,:,2)=vector_ch2;
     %vector_ch1234(:,:,3)=vector_ch3;
     %vector_ch1234(:,:,4)=vector_ch4;
```

```
        % calculating T squared
         for i=1:4
             eval ( ['T2_ch',int2str(i),'(d1)
=norm_vector_ch',int2str(i),'(d1,:)*PCloading_',int2str(i),'*invPClamda_'
,int2str(i),'*PCloading_T_',int2str(i),'*transpose(norm_vector_ch',int2st
r(i),'(d1,:));'])
         end

         % calculating Q statistic

         for i=1:4
             eval ( [ 'Q_ch',int2str(i),'(d1)= ( norm (
norm_vector_ch',int2str(i),'(d1,:)* (E',int2str(i),'-
PCloading_',int2str(i),'*PCloading_T_',int2str(i),')) ).^2;'])
         end


     end

     %calculating score for fault condition

     for i=1:4
         eval(
['score_faulty',int2str(i),'=norm_vector_ch',int2str(i),'*PCloading_',int
2str(i),';'])
     end

     %the following line is to save T2 and Q into file


       %save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\normal_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch3','T2_ch4
','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %1save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\cavitation_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch3','T2
_ch4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %2save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\impeller_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch3','T2_c
h4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %3save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\bearing_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch3','T2_ch
4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %4save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\blockage_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch3','T2_c
h4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %5save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\impellercavitation_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_
ch3','T2_ch4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
     %6save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\impellerblockage_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_ch
3','T2_ch4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');
```

```
    save ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearingcavitation_T_Q_sym12.mat','T2_ch1','T2_ch2','T2_c
h3','T2_ch4','Q_ch1','Q_ch2','Q_ch3','Q_ch4');

    %save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file
sec7_6\T_Q_sym12\bearingcavitation_score_matrix_sym12.mat','score_faulty1
','score_faulty2','score_faulty3','score_faulty4');
    %save
('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter
7\mat file sec7_6\T_Q_sym12\baseline_latent.mat','latent');
```

Module 3 − Plotting T^2 and Q-statistic

```
%Script to plot T and Q statistic
%Created: Mar 2014, Berli Kamiel

clear all;
clc;
close all;

%%%%%%%%%%%%%%%%%%%%%% LOAD T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\normal_T_Q_sym4.mat')
T2_Ch1_normal_sym4=T2_ch1;
Q_Ch1_normal_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\cavitation_T_Q_sym4.mat')
T2_Ch1_cavitation_sym4=T2_ch1;
Q_Ch1_cavitation_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\impeller_T_Q_sym4.mat')
T2_Ch1_impeller_sym4=T2_ch1;
Q_Ch1_impeller_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\bearing_T_Q_sym4.mat')
T2_Ch1_bearing_sym4=T2_ch1;
Q_Ch1_bearing_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\blockage_T_Q_sym4.mat')
T2_Ch1_blockage_sym4=T2_ch1;
Q_Ch1_blockage_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\impellercavitation_T_Q_sym4.mat')
T2_Ch1_impellercavitation_sym4=T2_ch1;
Q_Ch1_impellercavitation_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\impellerblockage_T_Q_sym4.mat')
T2_Ch1_impellerblockage_sym4=T2_ch1;
Q_Ch1_impellerblockage_sym4=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\bearingcavitation_T_Q_sym4.mat')
T2_Ch1_bearingcavitation_sym4=T2_ch1;
Q_Ch1_bearingcavitation_sym4=Q_ch1;

%%%%%%%%%%%%%%%%%%%%%% END LOAD T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%% COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%

T2_Ch1_sym4=[T2_Ch1_normal_sym4,T2_Ch1_cavitation_sym4,T2_Ch1_impeller_sym4,T2_Ch1_bearing_sym4,...

T2_Ch1_blockage_sym4,T2_Ch1_impellercavitation_sym4,T2_Ch1_impellerblockage_sym4,T2_Ch1_bearingcavitation_sym4];

Q_Ch1_sym4=[Q_Ch1_normal_sym4,Q_Ch1_cavitation_sym4,Q_Ch1_impeller_sym4,Q_Ch1_bearing_sym4,...

Q_Ch1_blockage_sym4,Q_Ch1_impellercavitation_sym4,Q_Ch1_impellerblockage_sym4,Q_Ch1_bearingcavitation_sym4];
```

```
    %%%%%%%%%%%%%%%%%%%%% END COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%


    %#############################

    %%%%%%%%%%%%%%%%%%%%%% LOAD T_Q sym8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\normal_T_Q_sym8.mat')
    T2_Ch1_normal_sym8=T2_ch1;
    Q_Ch1_normal_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\cavitation_T_Q_sym8.mat')
    T2_Ch1_cavitation_sym8=T2_ch1;
    Q_Ch1_cavitation_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\impeller_T_Q_sym8.mat')
    T2_Ch1_impeller_sym8=T2_ch1;
    Q_Ch1_impeller_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\bearing_T_Q_sym8.mat')
    T2_Ch1_bearing_sym8=T2_ch1;
    Q_Ch1_bearing_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\blockage_T_Q_sym8.mat')
    T2_Ch1_blockage_sym8=T2_ch1;
    Q_Ch1_blockage_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\impellercavitation_T_Q_sym8.mat')
    T2_Ch1_impellercavitation_sym8=T2_ch1;
    Q_Ch1_impellercavitation_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\impellerblockage_T_Q_sym8.mat')
    T2_Ch1_impellerblockage_sym8=T2_ch1;
    Q_Ch1_impellerblockage_sym8=Q_ch1;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\bearingcavitation_T_Q_sym8.mat')
    T2_Ch1_bearingcavitation_sym8=T2_ch1;
    Q_Ch1_bearingcavitation_sym8=Q_ch1;

    %%%%%%%%%%%%%%%%%%%%%% END LOAD T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%% COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%

    T2_Ch1_sym8=[T2_Ch1_normal_sym8,T2_Ch1_cavitation_sym8,T2_Ch1_impell
er_sym8,T2_Ch1_bearing_sym8,...

T2_Ch1_blockage_sym8,T2_Ch1_impellercavitation_sym8,T2_Ch1_impellerblocka
ge_sym8,T2_Ch1_bearingcavitation_sym8];

    Q_Ch1_sym8=[Q_Ch1_normal_sym8,Q_Ch1_cavitation_sym8,Q_Ch1_impeller_s
ym8,Q_Ch1_bearing_sym8,...

Q_Ch1_blockage_sym8,Q_Ch1_impellercavitation_sym8,Q_Ch1_impellerblockage_
sym8,Q_Ch1_bearingcavitation_sym8];
    %%%%%%%%%%%%%%%%%%%%% END COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%


    %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
%%%%%%%%%%%%%%%%%%%%%% LOAD T_Q sym12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\normal_T_Q_sym12.mat')
T2_Ch1_normal_sym12=T2_ch1;
Q_Ch1_normal_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\cavitation_T_Q_sym12.mat')
T2_Ch1_cavitation_sym12=T2_ch1;
Q_Ch1_cavitation_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\impeller_T_Q_sym12.mat')
T2_Ch1_impeller_sym12=T2_ch1;
Q_Ch1_impeller_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\bearing_T_Q_sym12.mat')
T2_Ch1_bearing_sym12=T2_ch1;
Q_Ch1_bearing_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\blockage_T_Q_sym12.mat')
T2_Ch1_blockage_sym12=T2_ch1;
Q_Ch1_blockage_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\impellercavitation_T_Q_sym12.mat')
T2_Ch1_impellercavitation_sym12=T2_ch1;
Q_Ch1_impellercavitation_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\impellerblockage_T_Q_sym12.mat')
T2_Ch1_impellerblockage_sym12=T2_ch1;
Q_Ch1_impellerblockage_sym12=Q_ch1;

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym12\bearingcavitation_T_Q_sym12.mat')
T2_Ch1_bearingcavitation_sym12=T2_ch1;
Q_Ch1_bearingcavitation_sym12=Q_ch1;

%%%%%%%%%%%%%%%%%%%%%% END LOAD T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%% COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%

T2_Ch1_sym12=[T2_Ch1_normal_sym12,T2_Ch1_cavitation_sym12,T2_Ch1_impeller_sym12,T2_Ch1_bearing_sym12,...

T2_Ch1_blockage_sym12,T2_Ch1_impellercavitation_sym12,T2_Ch1_impellerblockage_sym12,T2_Ch1_bearingcavitation_sym12];

Q_Ch1_sym12=[Q_Ch1_normal_sym12,Q_Ch1_cavitation_sym12,Q_Ch1_impeller_sym12,Q_Ch1_bearing_sym12,...

Q_Ch1_blockage_sym12,Q_Ch1_impellercavitation_sym12,Q_Ch1_impellerblockage_sym12,Q_Ch1_bearingcavitation_sym12];
%%%%%%%%%%%%%%%%%%%%% END COMBINE T_Q sym4 %%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%LOAD EIGENVALUES (or VARIANCE)%%%%%%%%%%%%%%%%%%

load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis chapter 7\mat file sec7_6\T_Q_sym4\baseline_latent_sym4.mat')
latent_sym4=latent;
```

```
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\baseline_latent_sym8.mat')
    latent_sym8=latent;

    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\baseline_latent_sym12.mat')
    latent_sym12=latent;
    clear latent

    %%%%%%%%%%%%%%%%%%%%%%%%% END LOAD EIGENVALUES %%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%% CPV CALCULATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for i=1:4

        %eval ( ['lamda',int2str(i),'=diag(latent(:,',int2str(i),'));'
])

CPV_sym4(:,i)=cumsum(latent_sym4(:,i))/sum(latent_sym4(:,i))*100;

CPV_sym8(:,i)=cumsum(latent_sym8(:,i))/sum(latent_sym8(:,i))*100;

CPV_sym12(:,i)=cumsum(latent_sym12(:,i))/sum(latent_sym12(:,i))*100;
    end

    % criteria how many PCs retained in the PCA model

    totalVariance=95;
    for i=1:4
        s_sym4=find( CPV_sym4(:,i)>totalVariance);
        idx_sym4(i)=s_sym4(1);

        s_sym8=find( CPV_sym8(:,i)>totalVariance);
        idx_sym8(i)=s_sym8(1);

        s_sym12=find( CPV_sym12(:,i)>totalVariance);
        idx_sym12(i)=s_sym12(1);
    end

    %%%%%%%%%%%%%%%%%%%%%%%%% END CPV %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%% T2alfa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %sym4
    T2alfa_sym4=idx_sym4(1)*(30-1)/(30-idx_sym4(1))*2.4422;
    %sym8
    T2alfa_sym8=idx_sym8(1)*(30-1)/(30-idx_sym8(1))*2.3965;
    %sym12
    T2alfa_sym12=idx_sym12(1)*(30-1)/(30-idx_sym12(1))*2.3965;
    %%%%%%%%%%%%%%%%%%%%%%% END T2alfa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %%%%%%%%%%%%%%%%%%%%%%%% Qalfa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Qsym4
    Teta1= sum( (latent_sym4(idx_sym4(1):30,1)).^1);
    Teta2= sum( (latent_sym4(idx_sym4(1):30,1)).^2);
    Teta3= sum( (latent_sym4(idx_sym4(1):30,1)).^3);

    h0=1- ((2*Teta1*Teta3)/(3*Teta2^2));
    c_alpha=1.96;

    a1=h0*c_alpha*sqrt(2*Teta2)/Teta1;
    a2=Teta2*h0*(h0-1)/Teta1^2;

    Q_alfa_sym4=Teta1*(a1+1+a2)^(1/h0);

    %-----------------------------------------------------
```

```
%Q_sym8
Teta1= sum( (latent_sym8(idx_sym8(1):30,1)).^1);
Teta2= sum( (latent_sym8(idx_sym8(1):30,1)).^2);
Teta3= sum( (latent_sym8(idx_sym8(1):30,1)).^3);

h0=1- ((2*Teta1*Teta3)/(3*Teta2^2));
c_alpha=1.96;

a1=h0*c_alpha*sqrt(2*Teta2)/Teta1;
a2=Teta2*h0*(h0-1)/Teta1^2;

Q_alfa_sym8=Teta1*(a1+1+a2)^(1/h0);

%----------------------------------------------------

%Qsym12
Teta1= sum( (latent_sym12(idx_sym12(1):30,1)).^1);
Teta2= sum( (latent_sym12(idx_sym12(1):30,1)).^2);
Teta3= sum( (latent_sym12(idx_sym12(1):30,1)).^3);

h0=1- ((2*Teta1*Teta3)/(3*Teta2^2));
c_alpha=1.96;

a1=h0*c_alpha*sqrt(2*Teta2)/Teta1;
a2=Teta2*h0*(h0-1)/Teta1^2;

Q_alfa_sym12=Teta1*(a1+1+a2)^(1/h0);


%%%%%%%%%%%%%%%%%%%%%%% END Qalfa %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



figure1=figure;

semilogy(T2_Ch1_sym4,'-*b','MarkerSize',2)
hold on
semilogy(T2_Ch1_sym8,'-*r','Markersize',2)
semilogy(T2_Ch1_sym12,'-*g','MarkerSize',2)
line( [0 400],[T2alfa_sym4 T2alfa_sym4],'LineStyle','-','color','b')
line( [0 400],[T2alfa_sym8 T2alfa_sym8],'LineStyle','-','color','r')
line( [0 400],[T2alfa_sym12 T2alfa_sym12],'LineStyle','-
','color','g')
legend('PCA model 1a','PCA model 1b','PCA model
1c','Location','NorthWest')
ylabel('T^{2}-statistic')
xlabel('Samples')

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
% Create textbox T2 STATISTIC
annotation(figure1,'textbox',...
    [0.235285714285714 0.541904761904763 0.0847142857142857
0.0500000000000008],...
    'String',{'fault1'},...
    'FitBoxToText','off',...
    'LineStyle','none');

% Create textbox
annotation(figure1,'textbox',...
    [0.814632895294616 0.863706147586056 0.0847142857142857
0.0500000000000008],...
    'String',{'fault7'},...
    'FitBoxToText','off',...
    'LineStyle','none');

% Create textbox
annotation(figure1,'textbox',...
```

```
          [0.435642857142856 0.876666666666667 0.0847142857142857
0.0500000000000008],...
          'String',{'fault3'},...
          'FitBoxToText','off',...
          'LineStyle','none');

      % Create textbox
      annotation(figure1,'textbox',...
          [0.531742899533701 0.369523809523811 0.0847142857142857
0.0500000000000008],...
          'String',{'fault4'},...
          'FitBoxToText','off',...
          'LineStyle','none');

      % Create textbox
      annotation(figure1,'textbox',...
          [0.620270877490463 0.561165731881668 0.0847142857142857
0.0500000000000008],...
          'String',{'fault5'},...
          'FitBoxToText','off',...
          'LineStyle','none');

      % Create textbox
      annotation(figure1,'textbox',...
          [0.725612123781264 0.441073353128781 0.0847142857142857
0.0500000000000008],...
          'String',{'fault6'},...
          'FitBoxToText','off',...
          'LineStyle','none');

      % Create textbox
      annotation(figure1,'textbox',...
          [0.33443471810089 0.395785769273069 0.0847142857142857
0.0500000000000008],...
          'String',{'fault2'},...
          'FitBoxToText','off',...
          'LineStyle','none');

      % Create textbox
      annotation(figure1,'textbox',...
          [0.142924544298432 0.314285714285714 0.0847142857142857
0.0500000000000008],...
          'String',{'normal'},...
          'FitBoxToText','off',...
          'LineStyle','none');
      % ^^^^^^^^^^^^^^^^^ END CREATE TEXTBOX T2 STATISTIC
^^^^^^^^^^^^^^^^^^^^^^

      figure2=figure;

      semilogy(Q_Ch1_sym4,'-*b','MarkerSize',2)
      hold on
      semilogy(Q_Ch1_sym8,'-*r','Markersize',2)
      semilogy(Q_Ch1_sym12,'-*g','MarkerSize',2)
      line( [0 400],[Q_alfa_sym4 Q_alfa_sym4],'LineStyle','-','color','b')
      line( [0 400],[Q_alfa_sym8 Q_alfa_sym8],'LineStyle','-','color','r')
      line( [0 400],[Q_alfa_sym12 Q_alfa_sym12],'LineStyle','-
','color','g')
      legend('PCA model 1a','PCA model 1b','PCA model
1c','Location','NorthWest')
      ylabel('Q-statistic')
      xlabel('Samples')

      %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      % Create textbox Q-statistic
      annotation(figure2,'textbox',...
          [0.140285714285714 0.319047619047619 0.0686428571428571
0.0452380952380959],...
```

```
        'String',{'normal'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.245807121661721 0.510392609699769 0.0613145400593472
0.0461893764434181],...
        'String',{'fault1'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.342958456973294 0.451131639722864 0.0613145400593472
0.0461893764434181],...
        'String',{'fault2'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.43939762611276 0.816027713625866 0.0613145400593472
0.0461893764434181],...
        'String',{'fault3'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.828121661721069 0.818337182448037 0.0613145400593472
0.0461893764434181],...
        'String',{'fault7'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.73019881305638 0.471916859122402 0.0613145400593472
0.0461893764434181],...
        'String',{'fault6'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.632275964391692 0.536581986143187 0.0613145400593472
0.0461893764434181],...
        'String',{'fault5'},...
        'FitBoxToText','off',...
        'LineStyle','none');

    % Create textbox
    annotation(figure2,'textbox',...
        [0.540287833827893 0.402632794457275 0.0613145400593472
0.0461893764434181],...
        'String',{'fault4'},...
        'FitBoxToText','off',...
        'LineStyle','none');
    %^^^^^^^^^^^^^^^^^^^^^ END CREATE TEXT BOX Q STATISTIC^^^^^^^^^^^^^^

    %OOOOOOOOOOOOOOOOOO CALCULATE MISDETECTION OOOOOOOOOOOOOOOOOOOOOOOO
    %T2

    T2idx_sym4_normal=size(find(T2_Ch1_sym4(1:50)>T2alfa_sym4));
    T2idx_sym4_fault=size(find(T2_Ch1_sym4(51:400)<T2alfa_sym4));
    T2idx_sym4_total=(T2idx_sym4_normal(2)+T2idx_sym4_fault(2));
    misdetection_T2_sym4=T2idx_sym4_total/400 *100
```

```
T2idx_sym8_normal=size(find(T2_Ch1_sym8(1:50)>T2alfa_sym8));
T2idx_sym8_fault=size(find(T2_Ch1_sym8(51:400)<T2alfa_sym8));
T2idx_sym8_total=(T2idx_sym8_normal(2)+T2idx_sym8_fault(2));
misdetection_T2_sym8=T2idx_sym8_total/400*100

T2idx_sym12_normal=size(find(T2_Ch1_sym12(1:50)>T2alfa_sym12));
T2idx_sym12_fault=size(find(T2_Ch1_sym12(51:400)<T2alfa_sym12));
T2idx_sym12_total=(T2idx_sym12_normal(2)+T2idx_sym12_fault(2));
misdetection_T2_sym12=T2idx_sym12_total/400*100
% END T2

% Q

Qidx_sym4_normal=size(find(Q_Ch1_sym4(1:50)>Q_alfa_sym4));
Qidx_sym4_fault=size(find(Q_Ch1_sym4(51:400)<Q_alfa_sym4));
Qidx_sym4_total=(Qidx_sym4_normal(2)+Qidx_sym4_fault(2));
misdetection_Q_sym4=Qidx_sym4_total/400 *100

Qidx_sym8_normal=size(find(Q_Ch1_sym8(1:50)>Q_alfa_sym8));
Qidx_sym8_fault=size(find(Q_Ch1_sym8(51:400)<Q_alfa_sym8));
Qidx_sym8_total=(Qidx_sym8_normal(2)+Qidx_sym8_fault(2));
misdetection_Q_sym8=Qidx_sym8_total/400 *100

Qidx_sym12_normal=size(find(Q_Ch1_sym12(1:50)>Q_alfa_sym12));
Qidx_sym12_fault=size(find(Q_Ch1_sym12(51:400)<Q_alfa_sym12));
Qidx_sym12_total=(Qidx_sym12_normal(2)+Qidx_sym12_fault(2));
misdetection_Q_sym12=Qidx_sym12_total/400 *100
% END Q

%OOOOOOOOOOOOOOOOOO END CALCULATE MISDETECTION OOOOOOOOOOOOOOOOOOOOO
```

Module 4 – Plotting Score Matrix

```
%Script to plot Score Matrix
%Created: Mar 2014, Berli Kamiel

clear all;
clc;
close all;

%$$$$$$$$$$$$$$$$$$$$$ LOAD SCORE sym4 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym4\normal_score_matrix_sym4.mat')
normal_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym4\cavitation_score_matrix_sym4.mat')
cavitation_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym4\impeller_score_matrix_sym4.mat')
impeller_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym4\bearing_score_matrix_sym4.mat')
bearing_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym4\blockage_score_matrix_sym4.mat')
blockage_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellercavitation_score_matrix_sym4.mat')
impellercavitation_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellerblockage_score_matrix_sym4.mat')
impellerblockage_score_faulty1_ch1_sym4=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\bearingcavitation_score_matrix_sym4.mat')
bearingcavitation_score_faulty1_ch1_sym4=score_faulty1;

%$$$$$$$$$$$$$$$$$$$$$ END LOAD SCORE sym4 %%%%%%%%%%%%%%%%%%%%%%%%

%******************** LOAD SCORE sym8 ******************************

load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\normal_score_matrix_sym8.mat')
normal_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\cavitation_score_matrix_sym8.mat')
cavitation_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\impeller_score_matrix_sym8.mat')
impeller_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\bearing_score_matrix_sym8.mat')
bearing_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym8\blockage_score_matrix_sym8.mat')
blockage_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impellercavitation_score_matrix_sym8.mat')
impellercavitation_score_faulty1_ch1_sym8=score_faulty1;
load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impellerblockage_score_matrix_sym8.mat')
impellerblockage_score_faulty1_ch1_sym8=score_faulty1;
```

```
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\bearingcavitation_score_matrix_sym8.mat')
    bearingcavitation_score_faulty1_ch1_sym8=score_faulty1;

    %****************** END LOAD SCORE sym8 ***************************

    %^^^^^^^^^^^^^^^^^^^ LOAD SCORE sym12 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\normal_score_matrix_sym12.mat')
    normal_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\cavitation_score_matrix_sym12.mat')
    cavitation_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\impeller_score_matrix_sym12.mat')
    impeller_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\bearing_score_matrix_sym12.mat')
    bearing_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_6\T_Q_sym12\blockage_score_matrix_sym12.mat')
    blockage_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellercavitation_score_matrix_sym12.mat')
    impellercavitation_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellerblockage_score_matrix_sym12.mat')
    impellerblockage_score_faulty1_ch1_sym12=score_faulty1;
    load('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearingcavitation_score_matrix_sym12.mat')
    bearingcavitation_score_faulty1_ch1_sym12=score_faulty1;

    %^^^^^^^^^^^^^^^^^^^^^ END LOAD SCORE sym12 ^^^^^^^^^^^^^^^^^^^^^^^^^



    %&&&&&&&&&&&&&&&&&&&& PLOT sym4 &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

    figure1=figure;


    plot3(normal_score_faulty1_ch1_sym4(:,1),normal_score_faulty1_ch1_sy
m4(:,2),normal_score_faulty1_ch1_sym4(:,3),'h')
    hold on
    plot3(cavitation_score_faulty1_ch1_sym4(:,1),cavitation_score_faulty
1_ch1_sym4(:,2),cavitation_score_faulty1_ch1_sym4(:,3),'o','color','g')
    plot3(impeller_score_faulty1_ch1_sym4(:,1),impeller_score_faulty1_ch
1_sym4(:,2),impeller_score_faulty1_ch1_sym4(:,3),'*','color','r')
    plot3(bearing_score_faulty1_ch1_sym4(:,1),bearing_score_faulty1_ch1_
sym4(:,2),bearing_score_faulty1_ch1_sym4(:,3),'d','color','r')
    plot3(blockage_score_faulty1_ch1_sym4(:,1),blockage_score_faulty1_ch
1_sym4(:,2),blockage_score_faulty1_ch1_sym4(:,3),'square','color','y','Ma
rkerSize',3)
    plot3(impellercavitation_score_faulty1_ch1_sym4(:,1),impellercavitat
ion_score_faulty1_ch1_sym4(:,2),impellercavitation_score_faulty1_ch1_sym4
(:,3),'>','color','m')
    plot3(impellerblockage_score_faulty1_ch1_sym4(:,1),impellerblockage_
score_faulty1_ch1_sym4(:,2),impellerblockage_score_faulty1_ch1_sym4(:,3),
'^','color','g')
    plot3(bearingcavitation_score_faulty1_ch1_sym4(:,1),bearingcavitatio
n_score_faulty1_ch1_sym4(:,2),bearingcavitation_score_faulty1_ch1_sym4(:,
3),'p','color','c')
    xlabel('Scores on PC1','FontSize',8)
```

```
      ylabel('Scores on PC2','FontSize',8)
      zlabel('Scores on PC3','FontSize',8)
      title('(a)')
      grid
      legend('normal','fault1','fault2','fault3','fault4','fault5','fault6
','fault7','location','NorthWest')
      FigHandle = figure1;
      set(FigHandle, 'Position', [50, 50, 450, 600]);
      set(gca,'FontSize',8)
      view([-57 6])

      figure2=figure;


      plot3(normal_score_faulty1_ch1_sym4(:,1),normal_score_faulty1_ch1_sy
m4(:,2),normal_score_faulty1_ch1_sym4(:,3),'h')
      hold on
      plot3(cavitation_score_faulty1_ch1_sym4(:,1),cavitation_score_faulty
1_ch1_sym4(:,2),cavitation_score_faulty1_ch1_sym4(:,3),'o','color','g')
      plot3(impeller_score_faulty1_ch1_sym4(:,1),impeller_score_faulty1_ch
1_sym4(:,2),impeller_score_faulty1_ch1_sym4(:,3),'*','color','r')
      %plot3(bearing_score_faulty1_ch1_sym4(:,1),bearing_score_faulty1_ch1
_sym4(:,2),bearing_score_faulty1_ch1_sym4(:,3),'d','color','r')
      plot3(blockage_score_faulty1_ch1_sym4(:,1),blockage_score_faulty1_ch
1_sym4(:,2),blockage_score_faulty1_ch1_sym4(:,3),'square','color','y','Ma
rkerSize',3)
      plot3(impellercavitation_score_faulty1_ch1_sym4(:,1),impellercavitat
ion_score_faulty1_ch1_sym4(:,2),impellercavitation_score_faulty1_ch1_sym4
(:,3),'>','color','m')
      plot3(impellerblockage_score_faulty1_ch1_sym4(:,1),impellerblockage_
score_faulty1_ch1_sym4(:,2),impellerblockage_score_faulty1_ch1_sym4(:,3),
'^','color','g')
      %plot3(bearingcavitation_score_faulty1_ch1_sym4(:,1),bearingcavitati
on_score_faulty1_ch1_sym4(:,2),bearingcavitation_score_faulty1_ch1_sym4(:
,3),'p','color','c')
      xlabel('Scores on PC1','FontSize',8)
      ylabel('Scores on PC2','FontSize',8)
      zlabel('Scores on PC3','FontSize',8)
      title('(b)')
      grid
      legend('normal','fault1','fault2','fault4','fault5','fault6','locati
on','NorthWest')
      FigHandle = figure2;
      set(FigHandle, 'Position', [550, 50, 450, 600]);
      set(gca,'FontSize',8)
      view([-57 6])
      %&&&&&&&&&&&&&&&&&&&&&&&&&&& END PLOT sym4 &&&&&&&&&&&&&&&&&&&&&&&&&

      %~~~~~~~~~~~~~~~~~~~~~~~~~ PLOT sym8 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      figure3=figure;

      plot3(normal_score_faulty1_ch1_sym8(:,1),normal_score_faulty1_ch1_sy
m8(:,2),normal_score_faulty1_ch1_sym8(:,3),'h')
      hold on
      plot3(cavitation_score_faulty1_ch1_sym8(:,1),cavitation_score_faulty
1_ch1_sym8(:,2),cavitation_score_faulty1_ch1_sym8(:,3),'o','color','g')
      plot3(impeller_score_faulty1_ch1_sym8(:,1),impeller_score_faulty1_ch
1_sym8(:,2),impeller_score_faulty1_ch1_sym8(:,3),'*','color','r')
      plot3(bearing_score_faulty1_ch1_sym8(:,1),bearing_score_faulty1_ch1_
sym8(:,2),bearing_score_faulty1_ch1_sym8(:,3),'d','color','r')
      plot3(blockage_score_faulty1_ch1_sym8(:,1),blockage_score_faulty1_ch
1_sym8(:,2),blockage_score_faulty1_ch1_sym8(:,3),'square','color','y','Ma
rkerSize',3)
      plot3(impellercavitation_score_faulty1_ch1_sym8(:,1),impellercavitat
ion_score_faulty1_ch1_sym8(:,2),impellercavitation_score_faulty1_ch1_sym8
(:,3),'>','color','m')
```

```
    plot3(impellerblockage_score_faulty1_ch1_sym8(:,1),impellerblockage_
score_faulty1_ch1_sym8(:,2),impellerblockage_score_faulty1_ch1_sym8(:,3),
'^','color','g')
    plot3(bearingcavitation_score_faulty1_ch1_sym8(:,1),bearingcavitatio
n_score_faulty1_ch1_sym8(:,2),bearingcavitation_score_faulty1_ch1_sym8(:,
3),'p','color','c')
    xlabel('Scores on PC1','FontSize',8)
    ylabel('Scores on PC2','FontSize',8)
    zlabel('Scores on PC3','FontSize',8)
    title('(a)')
    grid
    legend('normal','fault1','fault2','fault3','fault4','fault5','fault6
','fault7','location','West')
    FigHandle = figure3;
    set(FigHandle, 'Position', [50, 50, 450, 600]);
    set(gca,'FontSize',8)
    view([-57 6])

    figure4=figure;

    plot3(normal_score_faulty1_ch1_sym8(:,1),normal_score_faulty1_ch1_sy
m8(:,2),normal_score_faulty1_ch1_sym8(:,3),'h')
    hold on
    plot3(cavitation_score_faulty1_ch1_sym8(:,1),cavitation_score_faulty
1_ch1_sym8(:,2),cavitation_score_faulty1_ch1_sym8(:,3),'o','color','g')
    plot3(impeller_score_faulty1_ch1_sym8(:,1),impeller_score_faulty1_ch
1_sym8(:,2),impeller_score_faulty1_ch1_sym8(:,3),'*','color','r')
    %plot3(bearing_score_faulty1_ch1_sym8(:,1),bearing_score_faulty1_ch1
_sym8(:,2),bearing_score_faulty1_ch1_sym8(:,3),'d','color','r')
    plot3(blockage_score_faulty1_ch1_sym8(:,1),blockage_score_faulty1_ch
1_sym8(:,2),blockage_score_faulty1_ch1_sym8(:,3),'square','color','y','Ma
rkerSize',3)
    plot3(impellercavitation_score_faulty1_ch1_sym8(:,1),impellercavitat
ion_score_faulty1_ch1_sym8(:,2),impellercavitation_score_faulty1_ch1_sym8
(:,3),'>','color','m')
    plot3(impellerblockage_score_faulty1_ch1_sym8(:,1),impellerblockage_
score_faulty1_ch1_sym8(:,2),impellerblockage_score_faulty1_ch1_sym8(:,3),
'^','color','g')
    %plot3(bearingcavitation_score_faulty1_ch1_sym8(:,1),bearingcavitati
on_score_faulty1_ch1_sym8(:,2),bearingcavitation_score_faulty1_ch1_sym8(:
,3),'p','color','c')
    xlabel('Scores on PC1','FontSize',8)
    ylabel('Scores on PC2','FontSize',8)
    zlabel('Scores on PC3','FontSize',8)
    title('(b)')
    grid
    legend('normal','fault1','fault2','fault4','fault5','fault6','locati
on','NorthWest')
    FigHandle = figure4;
    set(FigHandle, 'Position', [550, 50, 450, 600]);
    set(gca,'FontSize',8)
    view([-57 6])
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~ END PLOT sym8 ~~~~~~~~~~~~~~~~~~~~~~~~~

    %>>>>>>>>>>>>>>>>>>>>>>>>>>>> PLOT sym12 <<<<<<<<<<<<<<<<<<<<<<<<<<<<

    figure5=figure;

    plot3(normal_score_faulty1_ch1_sym12(:,1),normal_score_faulty1_ch1_s
ym12(:,2),normal_score_faulty1_ch1_sym12(:,3),'h')
    hold on
    plot3(cavitation_score_faulty1_ch1_sym12(:,1),cavitation_score_fault
y1_ch1_sym12(:,2),cavitation_score_faulty1_ch1_sym12(:,3),'o','color','g'
)
    plot3(impeller_score_faulty1_ch1_sym12(:,1),impeller_score_faulty1_c
h1_sym12(:,2),impeller_score_faulty1_ch1_sym12(:,3),'*','color','r')
    plot3(bearing_score_faulty1_ch1_sym12(:,1),bearing_score_faulty1_ch1
_sym12(:,2),bearing_score_faulty1_ch1_sym12(:,3),'d','color','r')
```

```
    plot3(blockage_score_faulty1_ch1_sym12(:,1),blockage_score_faulty1_c
h1_sym12(:,2),blockage_score_faulty1_ch1_sym12(:,3),'square','color','y',
'MarkerSize',3)
    plot3(impellercavitation_score_faulty1_ch1_sym12(:,1),impellercavita
tion_score_faulty1_ch1_sym12(:,2),impellercavitation_score_faulty1_ch1_sy
m12(:,3),'>','color','m')
    plot3(impellerblockage_score_faulty1_ch1_sym12(:,1),impellerblockage
_score_faulty1_ch1_sym12(:,2),impellerblockage_score_faulty1_ch1_sym12(:,
3),'^','color','g')
    plot3(bearingcavitation_score_faulty1_ch1_sym12(:,1),bearingcavitati
on_score_faulty1_ch1_sym12(:,2),bearingcavitation_score_faulty1_ch1_sym12
(:,3),'p','color','c')
    xlabel('Scores on PC1','FontSize',8)
    ylabel('Scores on PC2','FontSize',8)
    zlabel('Scores on PC3','FontSize',8)
    title('(a)')
    grid
    legend('normal','fault1','fault2','fault3','fault4','fault5','fault6
','fault7','location','West')
    FigHandle = figure5;
    set(FigHandle, 'Position', [50, 50, 450, 600]);
    set(gca,'FontSize',8)
    view([-57 6])

    figure6=figure;

    plot3(normal_score_faulty1_ch1_sym12(:,1),normal_score_faulty1_ch1_s
ym12(:,2),normal_score_faulty1_ch1_sym12(:,3),'h')
    hold on
    plot3(cavitation_score_faulty1_ch1_sym12(:,1),cavitation_score_fault
y1_ch1_sym12(:,2),cavitation_score_faulty1_ch1_sym12(:,3),'o','color','g'
)
    plot3(impeller_score_faulty1_ch1_sym12(:,1),impeller_score_faulty1_c
h1_sym12(:,2),impeller_score_faulty1_ch1_sym12(:,3),'*','color','r')
    %plot3(bearing_score_faulty1_ch1_sym12(:,1),bearing_score_faulty1_ch
1_sym12(:,2),bearing_score_faulty1_ch1_sym12(:,3),'d','color','r')
    plot3(blockage_score_faulty1_ch1_sym12(:,1),blockage_score_faulty1_c
h1_sym12(:,2),blockage_score_faulty1_ch1_sym12(:,3),'square','color','y',
'MarkerSize',3)
    plot3(impellercavitation_score_faulty1_ch1_sym12(:,1),impellercavita
tion_score_faulty1_ch1_sym12(:,2),impellercavitation_score_faulty1_ch1_sy
m12(:,3),'>','color','m')
    plot3(impellerblockage_score_faulty1_ch1_sym12(:,1),impellerblockage
_score_faulty1_ch1_sym12(:,2),impellerblockage_score_faulty1_ch1_sym12(:,
3),'^','color','g')
    %plot3(bearingcavitation_score_faulty1_ch1_sym12(:,1),bearingcavitat
ion_score_faulty1_ch1_sym12(:,2),bearingcavitation_score_faulty1_ch1_sym1
2(:,3),'p','color','c')
    xlabel('Scores on PC1','FontSize',8)
    ylabel('Scores on PC2','FontSize',8)
    zlabel('Scores on PC3','FontSize',8)
    title('(b)')
    grid
    legend('normal','fault1','fault2','fault4','fault5','fault6','locati
on','NorthWest')
    FigHandle = figure6;
    set(FigHandle, 'Position', [550, 50, 450, 600]);
    set(gca,'FontSize',8)
    view([-57 6])

    %<<<<<<<<<<<<<<<<<<<<<<<<<<< END PLOT sym12 >>>>>>>>>>>>>>>>>>>>>>>>>
```

Module 5 − Plotting Classification Performance Using k-nn

```
%Script to plot classification performance
%Created: Mar 2014, Berli Kamiel

clear all;
clc;
close all;

%>>>>>>>>>>>>>>>>>>>>> LOAD SCORE MATRIX sym4 <<<<<<<<<<<<<<<<<<<<<<<<<<

% TRAINING DATA
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\normal_training_score_matrix_sym4.mat','score_faulty1');
training_normal_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\cavitation_training_score_matrix_sym4.mat','score_faulty1')
;
training_cavitation_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impeller_training_score_matrix_sym4.mat','score_faulty1');
training_impeller_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\bearing_training_score_matrix_sym4.mat','score_faulty1');
training_bearing_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\blockage_training_score_matrix_sym4.mat','score_faulty1');
training_blockage_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellercavitation_training_score_matrix_sym4.mat','score_f
aulty1');
training_impellercavitation_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellerblockage_training_score_matrix_sym4.mat','score_fau
lty1');
training_impellerblockage_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\bearingcavitation_training_score_matrix_sym4.mat','score_fa
ulty1');
training_bearingcavitation_ch1_sym4=score_faulty1;
% END OF TRAINING DATA

% TESTING DATA
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\normal_score_matrix_sym4.mat','score_faulty1');
testing_normal_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\cavitation_score_matrix_sym4.mat','score_faulty1');
testing_cavitation_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impeller_score_matrix_sym4.mat','score_faulty1');
testing_impeller_ch1_sym4=score_faulty1;
load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\bearing_score_matrix_sym4.mat','score_faulty1');
testing_bearing_ch1_sym4=score_faulty1;
```

```
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\blockage_score_matrix_sym4.mat','score_faulty1');
      testing_blockage_ch1_sym4=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellercavitation_score_matrix_sym4.mat','score_faulty1');
      testing_impellercavitation_ch1_sym4=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\impellerblockage_score_matrix_sym4.mat','score_faulty1');
      testing_impellerblockage_ch1_sym4=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym4\bearingcavitation_score_matrix_sym4.mat','score_faulty1');
      testing_bearingcavitation_ch1_sym4=score_faulty1;
      % END OF TESTING DATA
      %~~~~~~~~~~~~~~~~~~~~~ END OF sym4~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



      k=30; %Number of Neighbors

      [r_training,c_training]=size(training_normal_ch1_sym4);

      X=[training_normal_ch1_sym4;training_cavitation_ch1_sym4;training_impe
ller_ch1_sym4;training_bearing_ch1_sym4;

training_blockage_ch1_sym4;training_impellercavitation_ch1_sym4;training_im
pellerblockage_ch1_sym4;training_bearingcavitation_ch1_sym4];

      [r_X,~]=size(X);

      Y(1:r_training,1)=cellstr('normal');
Y(r_training+1:2*r_training,1)=cellstr('fault1');Y(2*r_training+1:3*r_train
ing,1)=cellstr('fault2');
      Y(3*r_training+1:4*r_training,1)=cellstr('fault3');Y(4*r_training+1:5*
r_training,1)=cellstr('fault4');Y(5*r_training+1:6*r_training,1)=cellstr('f
ault5');
      Y(6*r_training+1:7*r_training,1)=cellstr('fault6');Y(7*r_training+1:8*
r_training,1)=cellstr('fault7');
      r_training=r_training-20;
      Y1(1:r_training,1)=cellstr('normal');
Y1(r_training+1:2*r_training,1)=cellstr('fault1');Y1(2*r_training+1:3*r_tra
ining,1)=cellstr('fault2');
      Y1(3*r_training+1:4*r_training,1)=cellstr('fault3');Y1(4*r_training+1:
5*r_training,1)=cellstr('fault4');Y1(5*r_training+1:6*r_training,1)=cellstr
('fault5');
      Y1(6*r_training+1:7*r_training,1)=cellstr('fault6');Y1(7*r_training+1:
8*r_training,1)=cellstr('fault7');


      Z=[testing_normal_ch1_sym4;testing_cavitation_ch1_sym4;testing_impelle
r_ch1_sym4;testing_bearing_ch1_sym4;

testing_blockage_ch1_sym4;testing_impellercavitation_ch1_sym4;testing_impel
lerblockage_ch1_sym4;testing_bearingcavitation_ch1_sym4];

      for I=1:c_training
          for J=1:k

              X1=X(:,1:I);

              kNN=fitcknn(X1,Y,'NumNeighbors',J);

              label_result(:,I,J) = predict(kNN,Z(:,1:I));
              L(I,J)=loss(kNN,Z(:,1:I),Y1);
```

```
        end
    end

    [r,c]=size(L);

    kNN_performance=(ones(r,c)-L)*100;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    figure1=figure;

    plot(kNN_performance(1,:),'-bo','MarkerSize',2)
    hold on
    plot(kNN_performance(2,:),'-g+')
    plot(kNN_performance(3,:),'-rx')
    plot(kNN_performance(4,:),'-cs','MarkerSize',2)
    plot(kNN_performance(5,:),'-k.')
    plot(kNN_performance(6,:),'-m*')
    plot(kNN_performance(7,:),'-rd')
    %plot(kNN_performance(8,:),'-r^')
    %cek_kNN_sym4=kNN_performance(7,:)
    ylim([40 102])

    h_legend=legend('1PC','2PCs','3PCs','4PCs','5PCs','6PCs','7PCs','Locat
ion','EastOutside');
    set(h_legend,'FontSize',8)


    xlabel('Number of neighbors k')
    ylabel('Identification accuracy %')
    title('PCA model 1a')

    clear all;

    %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    %**********************************************************************
    %######################################################################

    %>>>>>>>>>>>>>>>>>>>> LOAD SCORE MATRIX sym8 <<<<<<<<<<<<<<<<<<<<<<<<<<

    % TRAINING DATA
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\normal_training_score_matrix_sym8.mat','score_faulty1');
    training_normal_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\cavitation_training_score_matrix_sym8.mat','score_faulty1')
;
    training_cavitation_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impeller_training_score_matrix_sym8.mat','score_faulty1');
    training_impeller_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\bearing_training_score_matrix_sym8.mat','score_faulty1');
    training_bearing_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\blockage_training_score_matrix_sym8.mat','score_faulty1');
    training_blockage_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impellercavitation_training_score_matrix_sym8.mat','score_f
aulty1');
    training_impellercavitation_ch1_sym8=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
```

```
sec7_6\T_Q_sym8\impellerblockage_training_score_matrix_sym8.mat','score_fau
lty1');
      training_impellerblockage_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\bearingcavitation_training_score_matrix_sym8.mat','score_fa
ulty1');
      training_bearingcavitation_ch1_sym8=score_faulty1;
      % END OF TRAINING DATA

      % TESTING DATA
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\normal_score_matrix_sym8.mat','score_faulty1');
      testing_normal_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\cavitation_score_matrix_sym8.mat','score_faulty1');
      testing_cavitation_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impeller_score_matrix_sym8.mat','score_faulty1');
      testing_impeller_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\bearing_score_matrix_sym8.mat','score_faulty1');
      testing_bearing_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\blockage_score_matrix_sym8.mat','score_faulty1');
      testing_blockage_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impellercavitation_score_matrix_sym8.mat','score_faulty1');
      testing_impellercavitation_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\impellerblockage_score_matrix_sym8.mat','score_faulty1');
      testing_impellerblockage_ch1_sym8=score_faulty1;
      load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym8\bearingcavitation_score_matrix_sym8.mat','score_faulty1');
      testing_bearingcavitation_ch1_sym8=score_faulty1;
      % END OF TESTING DATA
      %~~~~~~~~~~~~~~~~~~~~~ END OF sym8~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



      k=30; %Number of Neighbors

      [r_training,c_training]=size(training_normal_ch1_sym8);

      X=[training_normal_ch1_sym8;training_cavitation_ch1_sym8;training_impe
ller_ch1_sym8;training_bearing_ch1_sym8;

training_blockage_ch1_sym8;training_impellercavitation_ch1_sym8;training_im
pellerblockage_ch1_sym8;training_bearingcavitation_ch1_sym8];

      [r_X,~]=size(X);

      Y(1:r_training,1)=cellstr('normal');
Y(r_training+1:2*r_training,1)=cellstr('fault1');Y(2*r_training+1:3*r_train
ing,1)=cellstr('fault2');
      Y(3*r_training+1:4*r_training,1)=cellstr('fault3');Y(4*r_training+1:5*
r_training,1)=cellstr('fault4');Y(5*r_training+1:6*r_training,1)=cellstr('f
ault5');
      Y(6*r_training+1:7*r_training,1)=cellstr('fault6');Y(7*r_training+1:8*
r_training,1)=cellstr('fault7');
```

```
      r_training=r_training-20;
      Y1(1:r_training,1)=cellstr('normal');
Y1(r_training+1:2*r_training,1)=cellstr('fault1');Y1(2*r_training+1:3*r_tra
ining,1)=cellstr('fault2');
      Y1(3*r_training+1:4*r_training,1)=cellstr('fault3');Y1(4*r_training+1:
5*r_training,1)=cellstr('fault4');Y1(5*r_training+1:6*r_training,1)=cellstr
('fault5');
      Y1(6*r_training+1:7*r_training,1)=cellstr('fault6');Y1(7*r_training+1:
8*r_training,1)=cellstr('fault7');


      Z=[testing_normal_ch1_sym8;testing_cavitation_ch1_sym8;testing_impelle
r_ch1_sym8;testing_bearing_ch1_sym8;

testing_blockage_ch1_sym8;testing_impellercavitation_ch1_sym8;testing_impel
lerblockage_ch1_sym8;testing_bearingcavitation_ch1_sym8];

      for I=1:c_training
          for J=1:k

              X1=X(:,1:I);

              kNN=fitcknn(X1,Y,'NumNeighbors',J);

              label_result(:,I,J) = predict(kNN,Z(:,1:I));
              L(I,J)=loss(kNN,Z(:,1:I),Y1);

          end
      end

      [r,c]=size(L);

      kNN_performance=(ones(r,c)-L)*100;

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      figure2=figure;

      plot(kNN_performance(1,:),'-bo','MarkerSize',2)
      hold on
      plot(kNN_performance(2,:),'-g+')
      plot(kNN_performance(3,:),'-rx')
      plot(kNN_performance(4,:),'-cs','MarkerSize',2)
      plot(kNN_performance(5,:),'-k.')
      plot(kNN_performance(6,:),'-m*')
      plot(kNN_performance(7,:),'-rd')
      plot(kNN_performance(8,:),'-b^')
      %cek_kNN_sym8=kNN_performance(7,:)
      ylim([40 102])

      h_legend=legend('1PC','2PCs','3PCs','4PCs','5PCs','6PCs','7PCs','8PCs'
,'Location','EastOutside');
      set(h_legend,'FontSize',8)

      xlabel('Number of neighbors k')
      ylabel('Identification accuracy %')
      title('PCA model 1b')

      %save ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_8\TheBestKNN_ch1.mat','kNN_performance')
      %save ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file sec7_8\ForConfusionMat_1b.mat','label_result','Y1')
      clear all;

      %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      %&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
      %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

      %>>>>>>>>>>>>>>>>>>> LOAD SCORE MATRIX sym12 <<<<<<<<<<<<<<<<<<<<<<<<
```

```
    % TRAINING DATA
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\normal_training_score_matrix_sym12.mat','score_faulty1');
    training_normal_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\cavitation_training_score_matrix_sym12.mat','score_faulty1
');
    training_cavitation_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impeller_training_score_matrix_sym12.mat','score_faulty1')
;
    training_impeller_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearing_training_score_matrix_sym12.mat','score_faulty1');
    training_bearing_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\blockage_training_score_matrix_sym12.mat','score_faulty1')
;
    training_blockage_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellercavitation_training_score_matrix_sym12.mat','score
_faulty1');
    training_impellercavitation_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellerblockage_training_score_matrix_sym12.mat','score_f
aulty1');
    training_impellerblockage_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearingcavitation_training_score_matrix_sym12.mat','score_
faulty1');
    training_bearingcavitation_ch1_sym12=score_faulty1;
    % END OF TRAINING DATA

    % TESTING DATA
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\normal_score_matrix_sym12.mat','score_faulty1');
    testing_normal_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\cavitation_score_matrix_sym12.mat','score_faulty1');
    testing_cavitation_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impeller_score_matrix_sym12.mat','score_faulty1');
    testing_impeller_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearing_score_matrix_sym12.mat','score_faulty1');
    testing_bearing_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\blockage_score_matrix_sym12.mat','score_faulty1');
    testing_blockage_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellercavitation_score_matrix_sym12.mat','score_faulty1'
);
    testing_impellercavitation_ch1_sym12=score_faulty1;
```

```
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\impellerblockage_score_matrix_sym12.mat','score_faulty1');
    testing_impellerblockage_ch1_sym12=score_faulty1;
    load ('C:\Users\13212047\Documents\MATLAB\script_thesis2014\analysis
chapter 7\mat file
sec7_6\T_Q_sym12\bearingcavitation_score_matrix_sym12.mat','score_faulty1')
;
    testing_bearingcavitation_ch1_sym12=score_faulty1;
    % END OF TESTING DATA
    %~~~~~~~~~~~~~~~~~~~~~ END OF sym8~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



    k=30; %Number of Neighbors

    [r_training,c_training]=size(training_normal_ch1_sym12);

    X=[training_normal_ch1_sym12;training_cavitation_ch1_sym12;training_im
peller_ch1_sym12;training_bearing_ch1_sym12;

training_blockage_ch1_sym12;training_impellercavitation_ch1_sym12;training_
impellerblockage_ch1_sym12;training_bearingcavitation_ch1_sym12];

    [r_X,~]=size(X);

    Y(1:r_training,1)=cellstr('normal');
Y(r_training+1:2*r_training,1)=cellstr('fault1');Y(2*r_training+1:3*r_train
ing,1)=cellstr('fault2');
    Y(3*r_training+1:4*r_training,1)=cellstr('fault3');Y(4*r_training+1:5*
r_training,1)=cellstr('fault4');Y(5*r_training+1:6*r_training,1)=cellstr('f
ault5');
    Y(6*r_training+1:7*r_training,1)=cellstr('fault6');Y(7*r_training+1:8*
r_training,1)=cellstr('fault7');
    r_training=r_training-20;
    Y1(1:r_training,1)=cellstr('normal');
Y1(r_training+1:2*r_training,1)=cellstr('fault1');Y1(2*r_training+1:3*r_tra
ining,1)=cellstr('fault2');
    Y1(3*r_training+1:4*r_training,1)=cellstr('fault3');Y1(4*r_training+1:
5*r_training,1)=cellstr('fault4');Y1(5*r_training+1:6*r_training,1)=cellstr
('fault5');
    Y1(6*r_training+1:7*r_training,1)=cellstr('fault6');Y1(7*r_training+1:
8*r_training,1)=cellstr('fault7');


    Z=[testing_normal_ch1_sym12;testing_cavitation_ch1_sym12;testing_impel
ler_ch1_sym12;testing_bearing_ch1_sym12;

testing_blockage_ch1_sym12;testing_impellercavitation_ch1_sym12;testing_imp
ellerblockage_ch1_sym12;testing_bearingcavitation_ch1_sym12];

    for I=1:c_training
        for J=1:k

            X1=X(:,1:I);

            kNN=fitcknn(X1,Y,'NumNeighbors',J);

            label_result(:,I,J) = predict(kNN,Z(:,1:I));
            L(I,J)=loss(kNN,Z(:,1:I),Y1);

        end
    end

    [r,c]=size(L);

    kNN_performance=(ones(r,c)-L)*100;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure3=figure;

plot(kNN_performance(1,:),'-bo','MarkerSize',2)
hold on
plot(kNN_performance(2,:),'-g+')
plot(kNN_performance(3,:),'-rx')
plot(kNN_performance(4,:),'-cs','MarkerSize',2)
plot(kNN_performance(5,:),'-k.')
plot(kNN_performance(6,:),'-m*')
plot(kNN_performance(7,:),'-rd')
plot(kNN_performance(8,:),'-b^')
ylim([40 102])

h_legend=legend('1PC','2PCs','3PCs','4PCs','5PCs','6PCs','7PCs','8PCs'
,'Location','EastOutside');
set(h_legend,'FontSize',8)

xlabel('Number of neighbors k')
ylabel('Identification accuracy %')
title('PCA model 1c')
```