

FAULT DETECTION AND DIAGNOSIS OF AN INTERNAL COMBUSTION ENGINE

FAULT DETECTION AND DIAGNOSIS OF AN INTERNAL COMBUSTION ENGINE

By YIFEI FENG, B.ENG

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Applied Science

McMaster University © Copyright by Yifei Feng, January 2016

McMaster University

Hamilton, Ontario

MASTER OF APPLIED SCIENCE (2016)

(Mechanical Engineering)

TITLE: Fault Detection and Diagnosis of an Internal Combustion Engine

AUTHOR: Yifei Feng, B.ENG (University of Science and Technology Beijing)

SUPERVISORS: Professor Saeid Habibi, Professor Gary Bone

NUMBER OF PAGES: XVII, 108

Lay Abstract

The internal combustion engine (ICE) is widely used. After its long-term use faults occur. Therefore the development of techniques for ICE fault detection and diagnosis (FDD) is important.

The development of an ICE FDD experimental setup is described. The setup uses sensors to measure the ICE vibration and FDD techniques implemented as a computer program. One common ICE fault, the misfire fault, is deliberately introduced in some of the experiments by disconnecting a spark plug. The objective of the FDD is to determine if there is a misfire fault (or not), and which spark plug is faulty.

Several new techniques for FDD are proposed. Many experiments are performed to evaluate their performance. According to the experimental results, all of the proposed techniques can detect and locate the misfire faults with 100% accuracy. Techniques for improving the performance of artificial neural networks for FDD are also proposed and verified experimentally.

Abstract

The internal combustion engine (ICE) is widely used in applications such as automobiles, motorcycles and ships. After its long-term use faults occur that degrade its performance or cause it to malfunction. Therefore ICE fault detection and diagnosis (FDD) research is important for preventing serious economic loss and even human injuries caused by undetected faults.

The development of an ICE FDD experimental setup is described. The setup uses sensors to measure the ICE vibration and FDD techniques implemented as a computer program. One common ICE fault, the misfire fault, is deliberately induced in some of the experiments by disconnecting a spark plug. The objective of the FDD is to determine if there is a misfire fault (or not), and which spark plug is faulty.

Several FDD algorithms are proposed, one category of which is based on data processing techniques such as the variational mode decomposition (VMD) and the wavelet transform. This category of FDD algorithms includes the VMD-based FDD algorithm, wavelet-based kernel principle component analysis (KPCA) and VMD-based KPCA. The VMD-based FDD algorithm introduces a new FDD index based on VMD and statistics. According to the included experimental results, all of these algorithms are capable of detecting and locating the misfire fault with 100% accuracy.

A new SVSF-based training algorithm for the radial-basis-function (RBF) artificial neural network is also proposed. The running-averaged wavelet coefficients of vibration data are used as the network input. The included experimental results show these SVSF-trained networks above achieve 100% accuracy in classifying the misfire faults. The SVSF-based training algorithm also produces a faster convergence rate.

Contents

Lay Abstract	IV
Abstract	V
Contents	VI
Acknowledgements	VIII
List of Abbreviations.....	IX
List of Symbols.....	XI
List of Figures.....	XIV
List of Tables	XVII
Chapter 1 Introduction.....	1
1.1 Research Motivation	1
1.2 Research Objective.....	3
1.3 Thesis Organization	4
Chapter 2 Literature Review.....	5
2.1 Signal Processing Techniques for Fault Detection and Diagnosis	5
2.1.1 Fourier Transform.....	5
2.1.2 Wavelet Transform.....	6
2.1.3 Empirical Mode Decomposition	7
2.1.4 Variational Mode Decomposition	8
2.1.5 Principle Component Analysis.....	9
2.2 Artificial Neural Networks for Fault Detection and Diagnosis.....	10
2.3 Internal Combustion Engine Misfire Fault Detection.....	12
2.4 Summary.....	14
Chapter 3 Fault Diagnosis based on Data Processing Techniques	15
3.1 Underlying Theory	15
3.1.1 Crank Angle Domain Analysis	15
3.1.2 Wavelet Transform.....	16
3.1.3 Variational Mode Decomposition	18
3.2 VMD-Based FDD Algorithm	25
3.3 Nonlinear Multivariate and Multiscale Fault Diagnosis Techniques.....	31
3.4 Summary.....	34
Chapter 4 Artificial Neural Network for Fault Classification	36

4.1 Underlying Theory	36
4.1.1 Multi-layer Perceptron Network	36
4.1.2 Radial Basis Function Network	39
4.1.3 Smooth Variable Structure Filter	40
4.2 Multi-layer Perceptron Network Training Algorithm Using Smooth Variable Structure Filter	43
4.3 Radial Basis Function Network Training Algorithm Using Smooth Variable Structure Filter	46
4.4 Artificial Neural Network Input Selection	50
4.5 Summary.....	51
Chapter 5 Experimental Results, Analysis and Discussion	53
5.1 Experimental Setup	53
5.2 ICE Misfire Fault Induction	59
5.3 Results and Discussion.....	61
5.3.1 ICE Vibration Data in the Crank Angle Domain	61
5.3.2 Experiment Results for the VMD-based FDD Algorithm	63
5.3.3 Experimental Results for the Wavelet-Based KPCA FDD Algorithm.....	71
5.3.4 Experimental Results for the VMD-Based KPCA FDD Algorithm	84
5.3.5 Discussion of the Data-Processing-Technique-Based FDD Results	86
5.3.6 ICE Fault Classification Results for the MLP Network Trained by the SVSF	88
5.3.7 ICE Fault Classification Results for the RBF Network Trained by the SVSF	93
5.3.8 Discussion of the ANN FDD Results.....	96
Chapter 6 Conclusions.....	102
References.....	104

Acknowledgements

I would like to give special thanks to my supervisors, Dr. Habibi and Dr. Bone, for their valuable guidance and advice throughout my research in the past two years and in the completion of this thesis.

I would also like to thank Mr. Cam Fisher for his help in the experimental setup, and Mr. Joe Verhaeghe, for his assistance in designing the signal conditioner in this research.

Finally I would like to express my gratitude to my mom Jinxia and dad Yanli, who give me endless love and tolerance.

List of Abbreviations

ADMM	Alternate direction method of multipliers
ANN	Artificial neural network
ART	Adaptive resonance network
BP	Back-propagation
CAD	Crank angle domain
CID	Cam identification sensor
CWT	Continuous wavelet transform
DAQ	Data acquisition device
DPS	Directional power spectra
D-S	Dempster-Shafer
DWT	Discrete wavelet transform
EMD	Empirical mode decomposition
EWT	Empirical wavelet transform
FDD	Fault detection and diagnosis
FFT	Fast Fourier transform
FT	Fourier transform
ICE	Internal combustion engine
IMF	Intrinsic mode function
KPCA	Kernel PCA
LM	Levenberg-Marquardt
LS-SVM	Least square support vector machines
MLP	Multi-layer perceptron

MSPCA	Multi-scale principal component analysis
PC	Principal components
PCA	Principal component analysis
RBF	Radial basis function
SPE	Squared prediction error
SVSF	Smooth variable structure filter
UPS	Uninterruptible power supply
VMD	Variational mode decomposition

List of Symbols

b_j	MLP Node offset/bias
C^+	Pseudo inverse of C
$C_{k linearized}$	Jacobian matrix of the measurement function at k -th iteration
\overline{C}_k	Running average on the wavelet decomposition coefficients
$e_{z,k k}$	A-posteriori estimation error
$e_{z,k+1 k}$	A-priori estimation error
f	Input signal
\hat{f}	Fourier transform of f
\mathcal{F}	Feature space
$g(\cdot)$	Activation function for RBF
$h(\cdot)$	Nonlinear mapping between the RBF's parameters and outputs
I	VMD-based FDD index
$k(\cdot)$	Kernel function
K_{k+1}	SVSF gain
$L^2(Z)$	Collection of Lebesgue 2 nd integrable functions defined on Z
N_T	Total number of link weights of MLP
\mathbf{P}	PC loading matrix
Q	Q-statistic, also known as SPE
\mathbb{R}	Real space
\mathbf{s}_k	Vibration data of cycle k in CAD
\mathbf{S}	Vibration data matrix
\mathbf{T}	PC score matrix

\mathcal{J}^2	Hotteling's \mathcal{J}^2
\mathbf{T}_k	IMF decomposed from \mathbf{s}_k via VMD
\mathbf{t}_k^i	i -th vector of \mathbf{T}_k
u_k	k -th IMF
\hat{u}	Fourier transform of u
W_ϕ	Approximation coefficients
W_ψ	Detailed coefficients
w_{ji}	Link weight from neuron i to neuron j
\mathbf{X}	Data matrix
\mathbf{x}_{new}	New measurement
$\hat{\mathbf{x}}_{k k}$	A-posteriori estimation of the weight
$\hat{\mathbf{x}}_{k+1 k}$	A-priori estimation of the weight
$\hat{\mathbf{z}}_{k k}$	A-posteriori estimation of the measurement
$\mathbf{z}_{k+1 k}$	A-priori estimation of the measurement
α	Significance level
β	Weight of the penalty term
γ	Parameter in SVSF gain calculation
δ	Dirac distribution
$\delta_{Q,\alpha}$	Upper limits of the Q-statistic
ϵ	Tolerance of the convergence criterion
ζ	Lagrangian multiplier
$\hat{\zeta}$	Fourier transform of zeta
θ	RBF parameters
λ	Eigenvalue

Λ	Eigenvalue matrix
$\widehat{\Lambda}$	Eigenvalue matrix with retained PC
$\widetilde{\Lambda}$	Eigenvalue matrix with residual PC
μ	Normal distribution mean
$\hat{\mu}$	Estimate of normal distribution mean
\mathbf{v}_k	Zero-mean white Gaussian noises in measurement equation at k -th iteration
ρ	Crank angle
σ^2	Normal distribution variance
$\hat{\sigma}$	Estimate of normal distribution standard deviation
τ	Time-step of the dual ascent
τ_α^2	Upper limits of Hotteling's \mathcal{J}^2
$\phi(\cdot)$	Scaling function
$\varphi(\cdot)$	Activation function for MLP
$\psi(\cdot)$	Wavelet function
ψ	Smooth boundary layer width
ω_k	Center frequency for k -th IMF
$\boldsymbol{\omega}_k$	Zero-mean white Gaussian noises in system equation at k -th iteration

List of Figures

Figure 1.1.1 Classification of diagnostic algorithms.....	2
Figure 3.1.1 Schematic diagram for the fast wavelet transform	17
Figure 3.1.2 Schematic diagram for the fast wavelet reconstruction.....	18
Figure 3.1.3 Flow chart of the VMD-based FDD algorithm	30
Figure 3.1.4 Flow chart of the multiscale KPCA algorithm.....	34
Figure 4.1.1 Structure of a MLP network with two hidden layers	37
Figure 4.1.2 Detailed structure of a neuron.....	37
Figure 4.1.3 The log-sigmoid transfer function.....	38
Figure 4.1.4 The linear transfer Function.....	38
Figure 4.1.5 Structure of the RBF network	39
Figure 4.1.6 Concept of the SVSF	41
Figure 4.1.7 Structure of the SVSF	41
Figure 4.1.8 Flow chart of RBF Network Training Algorithm Using SVSF.....	50
Figure 5.1.1 Experimental setup for ICE fault detection.....	53
Figure 5.1.2 Experimental vehicle	54
Figure 5.1.3 CTC AC240-1D accelerometer	54
Figure 5.1.4 Positions and orientations of the accelerometers.....	57
Figure 5.1.5 PSP1001 current source	57
Figure 5.1.6 Signal Conditioner	58
Figure 5.1.7 dSPACE MicroAutoBox 1401	59
Figure 5.2.1 Spark plug numbering	59
Figure 5.2.2 Example of a spark plug connector.....	60

Figure 5.3.01 ICE Sensor Raw Data	61
Figure 5.3.02 ICE Sensor Data Split in Cycles	62
Figure 5.3.03 ICE Sensor Data after CAD Mapping	62
Figure 5.3.04 VMD-based FDD indices for the misfire fault for the normal condition.....	66
Figure 5.3.05 VMD-based FDD algorithm for the misfire fault in Cylinder 2	69
Figure 5.3.06 Confusion matrix for VMD-based FDD algorithm	71
Figure 5.3.07 Wavelet-based KPCA FDD result for the normal condition	77
Figure 5.3.08 Wavelet-based KPCA FDD result for misfire fault in Cylinder 2	83
Figure 5.3.09 Confusion matrix for wavelet-based KPCA algorithm.....	84
Figure 5.3.10 VMD-based KPCA for Misfire in Cylinder 2 (1 st Mode).....	85
Figure 5.3.11 Confusion matrix for VMD-based KPCA algorithm	86
Figure 5.3.12 VMD based KPCA result (T^2 statistics at Mode 1)	88
Figure 5.3.13 Confusion matrices for the SVSF-trained MLP network	89
Figure 5.3.14 Learning curve for the SVSF-trained MLP network.....	90
Figure 5.3.15 Test Confusion Matrix for the LM-trained MLP Network	91
Figure 5.3.16 Learning Curve for the EKF-trained MLP Network.....	91
Figure 5.3.17 Test Confusion Matrix for the LM-trained MLP Network with Running-averaged Vibration in CAD as Input	92
Figure 5.3.18 Test Confusion Matrix for the SVSF-trained MLP Network with Running-averaged Fourier Coefficient as Input.....	92
Figure 5.3.19 Confusion matrices for the SVSF-trained RBF network	94
Figure 5.3.20 Learning curve for the SVSF-trained RBF network.....	94
Figure 5.3.21 Test Confusion Matrix for the EKF-trained RBF Network	95
Figure 5.3.22 Learning Curve for the EKF-trained RBF Network.....	95

Figure 5.3.23 Test Confusion Matrix for the GD-trained RBF Network	96
Figure 5.3.24 Learning Curve for the GD-trained RBF Network.....	96
Figure 5.3.25 Classification Accuracy of SVSF-trained MLP network with Vibration in CAD as input in 10 experiments.	97
Figure 5.3.26 Classification Accuracy of SVSF-trained MLP network with Fourier Coefficient as input in 20 experiments.	98
Figure 5.3.27 Classification Accuracy of SVSF-trained MLP network with Wavelet Coefficient as input in 10 experiments	98
Figure 5.3.28 Classification Accuracy of EKF-trained MLP network with Wavelet Coefficient as input in 10 experiments	99
Figure 5.3.29 Classification Accuracy of LM-trained RBF network with Wavelet Coefficient as input in 10 experiments	99
Figure 5.3.30 Classification Accuracy of SVSF-trained RBF network with Wavelet Coefficient as input in 10 experiments	100
Figure 5.3.31 Classification Accuracy of EKF-trained RBF network with Wavelet Coefficient as input in 10 experiments	100
Figure 5.3.32 Classification Accuracy of GD-trained RBF network with Wavelet Coefficient as input in 20 Experiments	101

List of Tables

Table 5.2.1	The eight ICE working conditions studied.....	60
Table 5.3.1	VMD parameters in the experiment	63
Table 5.3.2	Experiment results for VMD-based FDD algorithm	70
Table 5.3.3	Control limits for wavelet-based KPCA	71
Table 5.3.4	Peak locations for a cylinder 2 misfire fault for the three signal based FDD Algorithms studied	87
Table 5.3.5	Summary of the ANN FDD results	97

Chapter 1 Introduction

1.1 Research Motivation

Internal combustion engines (ICEs) are widely applied in various means of transportation such as automobiles, motorcycles and ships. Like other types of machines, they suffer from performance degradation and malfunction especially after long term use or use in harsh working environments, which may cause serious economic loss and even human injuries. Thus the fault detection and diagnosis (FDD) research is of vital significance for guaranteeing the ICE's normal working condition.

Fault detection and diagnosis systems aim to accomplish the following three tasks: (Gertler, 1998)

1. Fault detection: the indication that something is going wrong in the monitored system.
2. Fault isolation: the determination of the exact location of the fault.
3. Fault identification: the determination of the magnitude of the fault.

So far researchers have proposed a number of diagnostic methods, which Venkatasubramanian et al. (2003) broadly classified into three general categories: quantitative model-based methods, qualitative model-based methods, and process history-based (also known as "signal-based") methods. Each of the three categories can be further subcategorized as shown in Fig. 1.1.1.

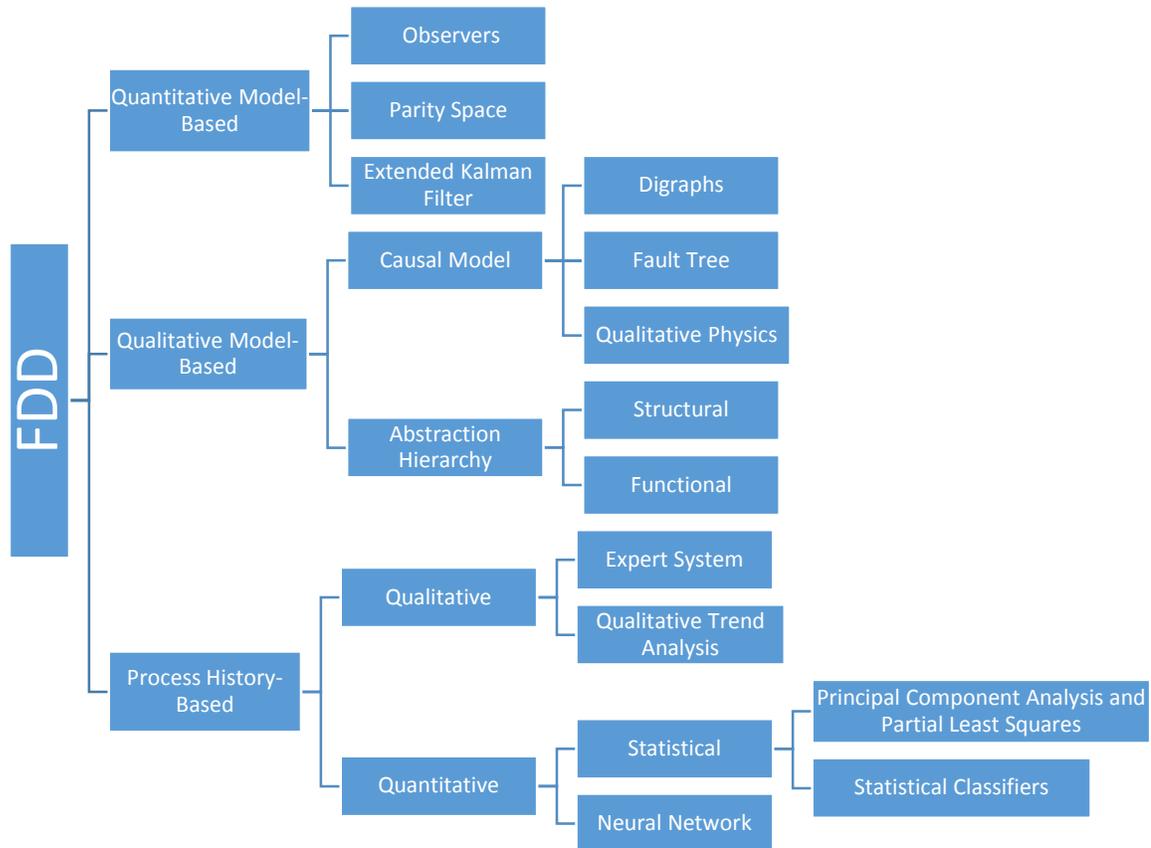


Figure 1.1.1 Classification of diagnostic algorithms (Venkatasubramanian, et al., 2003)

To apply model-based methods in fault detection and diagnosis, one needs to develop the system model according to its underlying principles. In quantitative model-based methods, the mathematical functional relationships between the inputs and outputs of the system are used while in qualitative model equations these relationships are qualitative functions centered on different units in a process. In contrast, the process history-based methods only assume the availability of large amount of historical process data. Further details on many of these methods are presented in (Isermann, 2006).

In this thesis, the studied ICE fault is the misfire fault because it is a common fault in running engines; and it can lead to poor engine performance and cause environmental pollution (Du, et

al., 2012). Knowing that an internal combustion engine is a complex system with many rotary and reciprocating components which makes accurate modeling very difficult, this thesis turns to the process history-based method for the misfire FDD.

1.2 Research Objective

This thesis is based on previous work by Haqshenas (2012) and Ahmed (2011). Haqshenas proposed a powerful data analysis tool based on the concept of multi-scale principal component analysis (MSPCA). MSPCA combines discrete wavelet analysis and principal component analysis (PCA). This tool was able to detect and localize ICE fault conditions such as a lash adjuster fault in both crank angle domain and frequency domain. Another contribution of Haqshenas' research was the covariance-based indices that indicate the faulty variable and faulty scale. Ahmed proposed a multi-layer perceptron (MLP) neural network training technique based on the smooth variable structure filter (SVSF), which was found to have good generalization capability and the ability to avoid premature convergence to local minima. Ahmed also applied this technique to a fault detection and isolation system which classified two faulty conditions (piston chirp and missing bearing faults) plus one baseline condition with high accuracy.

As an extension to Haqshenas' and Ahmed's work, this research aims to achieve the following objectives:

1. Build an FDD experimental setup.
2. Propose novel FDD methods based on data processing techniques and artificial neural networks.
3. Apply these FDD methods to the ICE misfire fault.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews the relevant literature on data processing techniques and artificial neural networks for FDD. This chapter also reviews the prior work on FDD of ICE misfire. Chapter 3 describes fault diagnosis algorithms obtained using data processing techniques, including one new statistical index and two kernel MSPCA algorithms based on wavelet transform and variational mode decomposition respectively. Chapter 4 presents an SVSF-based training algorithm for RBF neural networks. It also describes the method for generating the feature vector used as the networks' input. Chapter 5 first describes the experimental setup and the seeding of the ICE misfire faults. Next it presents experimental results and a comparison of the performance of the FDD methods. Chapter 6 draws the conclusions of the research and makes recommendations for future work.

Chapter 2 Literature Review

This chapter reviews the relevant literature on FDD methods for mechanical systems. Signal processing techniques and their application to FDD are covered first, followed by artificial neural networks and their application to FDD. Second, FDD methods for detecting misfire faults with ICE are reviewed. The chapter ends with a summary of the findings.

2.1 Signal Processing Techniques for Fault Detection and Diagnosis

Signal processing techniques are the basis for many FDD methods. This section presents the application of the Fourier transform, wavelet transform, empirical mode decomposition, variational mode decomposition and principal component analysis to the FDD problem.

2.1.1 Fourier Transform

The Fourier transform (FT) provides an alternative representation of a signal in the frequency domain rather than the time domain. Because of its usefulness in practical applications, FT has become a key tool in numerous engineering branches (Prandoni et al., 2008). The following are examples of its application to FDD.

Shim *et al.* (2010) proposed an FDD system for an uninterruptible power supply (UPS). The system performed real time fast Fourier transform (FFT) on the output current of the UPS' inverter side and analyzed its harmonics. Four cases that were studied in the paper: normal linear load case, normal rectifier load case, turn-on resistance increase fault, and turn-on delay misfiring fault. The faulty cases could be distinguished by comparing the even harmonics in the current. The even harmonics only took 5.3% and 5.6% of the current for the normal linear load and normal rectifier

load respectively. However, when the turn-on resistance increased a fault condition occurred, and the even harmonics increased to 18.8%. Also, the even harmonics increased sharply to 40.2% when there existed the turn-on delay misfiring fault. Their experimental results showed that the FFT was a simple and practical tool for detecting fault conditions in UPS.

Mamis *et al.* (2011) developed a method for fault isolation for power transmission lines. Transient voltage and current waveforms of the transmission lines were first obtained and then transformed into the frequency domain using the FFT. The first harmonic of the spectrum was utilized for determination of the fault location according to the travelling wave theory. The percentage error for the fault distance was within 0.47% to 3.20% when the fault distance changed from 40 km to 320 km. Other experiments also found that the measured fault distance were not affected by resistance, phase angle and source inductance.

2.1.2 Wavelet Transform

The wavelet transform is a useful tool for analysing aperiodic, noisy, intermittent and transient signals. Compared with the Fourier transform, the wavelet transform not only finds the frequency components in the signal but also tell at what instant a particular frequency arises. Due to its advantages, the wavelet transform has been widely applied in different fields, such as fault detection, audio processing and image processing (Sifuzzaman et al., 2009). The following are a few examples of its application to FDD.

Lin and Qu (2000) applied the wavelet transform in gear fault feature extraction. They first proposed a denoising method based on Morlet wavelet whose optimal parameter was obtained using minimal wavelet entropy method. Then this denoising method was used to filter the

vibrational signal of a gear box where a tooth crack failure resulted in a very clear periodic impulse with the feature period of 0.24s.

Sun and Tang (2002) used the wavelet transform in fault detection for bearings in freight cars. First the wavelet transform modulus maxima and their locations were obtained from the vibration signal. Then thresholding and signal reconstruction were performed for denoising. The reconstructed signal consisted of only defect-induced impulse clusters. The errors between the estimated and measured faulty frequencies were only 0.8%.

Kang *et al.* (2001) developed an FDD method for power transformers based on the wavelet transform. The vibration of the equipment was measured by one accelerometer. The envelope of the vibration was extracted using the Hilbert transform and then the continuous wavelet transform (CWT) was performed on the envelope signal to obtain the wavelet coefficients. Next, local maxima of these wavelet coefficients were found and selected to get the CWT vertical ridges. Finally the dominant ridgelines were determined according to the number of expected transient events. This method detected that the weak spring condition caused longer delay time between the dominant ridgelines. For example the delay time between the first two dominant ridgelines was 25.4 ms for the weak spring condition while it was 19.0 ms for the normal condition. This method also found that when the equipment had worn contacts, the strength of the dominant ridgelines decreased significantly, e.g. the strength of the 4th dominant ridgeline decreased from 11.28 to 6.30.

2.1.3 Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) proposed by Huang *et al.* is a time-frequency signal processing method and is useful for analyzing natural signals, which are most often non-linear

and aperiodic. The method is fully adaptive and generates intrinsic mode functions (IMFs) to represent the data. The IMFs represent a complete set of locally orthogonal basis functions whose amplitude and frequency may vary over time. (Zeiler et al., 2010). EMD has been applied to many fields in engineering and science.

Loutridis (2004) proposed a damage detection method for pairs of gears. In this method, the vibration signal resulting from the gears were decomposed into nine intrinsic mode functions (IMF), namely C_1 , C_2 to C_9 . Since C_2 best represented the time scale of the transient caused by gear damage, the energy of mode C_2 was selected as the diagnostic feature. The energy of a healthy gear pair was relatively low at C_2 . It increased when damage happened. When a gear pair had a 75% crack, the peak energy was more than three times that of a pair with only a 15% crack.

Yu *et al.* (2005) proposed a method for the fault diagnosis of roller bearings. The vibration signal of a roller bearing was first translated into time-scale representation by orthogonal wavelet bases. Then the envelope signal was obtained by envelope spectrum analysis of the wavelet coefficients that represent the high frequency components. After applying EMD and Hilbert transform, the local Hilbert marginal spectrum of the envelope signal was obtained, from which the fault pattern can be identified. In the paper, an outer-race fault was introduced in the roller bearing with a characteristic frequency of 76 Hz. The local marginal spectrum shows the frequency peaks of 76 Hz and its double frequency of 152 Hz.

2.1.4 Variational Mode Decomposition

The variational mode decomposition (VMD) is a recently proposed time-frequency signal analysis technique. Like EMD, it decomposes a signal into several intrinsic mode functions with limited frequency bands. In contrast, VMD is entirely non-recursive, theoretically well founded and more

robust to sampling and noise than EMD (Dragomiretskiy et al., 2014). Since VMD is a relatively new signal processing method, there are only a limited number of relevant applications of VMD to the fault detection field to date. The following are two examples.

Wang *et al.* (2015) applied VMD in detecting the rub-impact fault of a gas turbine. Vibration signals were collected from the measuring points of five bearings by eddy current transducers with a sampling frequency of 2000 Hz. The spectrum of the signal included three major frequency components. The decomposition results using VMD successfully detected all three. The empirical wavelet transform (EWT) extracted only the two lower frequency components. In contrast, the empirical mode decomposition (EMD) and the ensemble empirical mode decomposition (EEMD) did not distinguish between the components clearly.

Mohanty *et al.* (2014) used VMD to analyze bearing faults. Vibration data was acquired using one accelerometer. By comparing the VMD decomposition results of the healthy and faulty signal, it was easy to detect the fault from both the spreading in the frequency of the signal, and from the intensity of vibration. Mohanty also explored the effect of VMD parameter α on the decomposition result. An α value, which was too low, led to randomness of the frequency distribution of each intrinsic mode functions and made the FDD problem difficult.

2.1.5 Principle Component Analysis

Principal component analysis (PCA) is a powerful and versatile method capable of providing an overview of complex multivariate data. It aims to represent the data as a set of new orthogonal variables called principal components. PCA can be used for revealing relations between variables/samples, detecting outliers, finding and quantifying patterns, and other functions (Bro et al., 2014). The following are a few examples of its application to FDD.

Villegas *et al.* (2010) developed a PCA-based FDD system for a pilot chemical plant which consisted of two connected cylindrical tanks. The following five operation modes were considered: normal behaviour, tank 1 outlet clogging, tank 2 outlet clogging, level sensor 1 fault and level sensor 2 fault. Five corresponding PCA models were determined for those modes. An experiment was done to validate the FDD system. The experimental data contained 7000 samples, with the first 2300 for the normal mode and the rest for level sensor 1 fault. The results showed that in the first stage only the normal mode model generated Q statistics that were below its threshold. In the later stage, only the level 1 sensor fault model had Q statistics below its threshold.

Cho *et al.* (2004) proposed a new sensor fault identification method based on the kernel principal component analysis (KPCA). Hotelling's T^2 and the SPE statistics were first produced by the KPCA, which detected the fault based on the sudden increase of SPE from 251st sample in the simulated chemical process. Two new statistics were defined using the gradient of kernel function to represent the contribution of each variable to the monitoring statistics. The simulation result showed they could effectively discriminate the faulty variable

2.2 Artificial Neural Networks for Fault Detection and Diagnosis

The artificial neural network (ANN) is a massively parallel distributed network made up of simple processing units called neurons. It imitates the way in which human brain performs a particular task or function of interest. An ANN has the ability to approximate an unknown input-output mapping (Haykin, 2008). Fault classification is one of the important applications of the ANNs.

Ghate *et al.* (2010) developed an MLP neural network classifier for FDD of a three phase induction motor. Three AC current probes were first used to measure the stator current signals, from which several features were extracted such as Pearson's coefficient of skewness, the overall root mean

squared value and so on. After feature extraction, PCA was used for dimensionality reduction of the MLP to reduce its computation time. Experiments showed that the average classification accuracy on the testing instances was as high as 98.25%.

Rajakarunakaran *et al.* (2008) performed fault detection for a centrifugal pumping system by using the binary adaptive resonance network (ART1) in comparison with a feed-forward network trained by the back-propagation (BP) algorithm. A number of system states such as voltage, current, speed and pressure were measured for training data generation. The neural networks were tested for a total of seven categories of faults, such as shaft wear fault and bearing outside diameter reduced fault. The test result showed the ART1 model achieved a superior classification accuracy of 100% with only 20 s training time compared to the BP-trained feed-forward network with 99.3% accuracy and 82 s training time.

Wu *et al.* (2004) developed a radial-basis-function (RBF) neural network for induction machine FDD. Four feature vectors were extracted from power spectra of machine vibration signals measured by one accelerometer. A cell-splitting grid algorithm was proposed to automatically determine the network architecture of the RBF network. The FDD system was tested with unbalanced electrical faults and mechanical faults operating at different rotating speeds. Experimental results showed the accuracy of detecting electrical faults was 96.3% for the training data, and 85.7% for the testing data. The accuracy of detecting the normal condition was 99.5% for the training data, and 96.0% for the testing data. The mechanical faults were well separated from the other two conditions. The accuracy was 100% for both the training and testing data.

2.3 Internal Combustion Engine Misfire Fault Detection

Misfire is a common fault for internal combustion engines that has had significant research activity over the past two decades. Since the FDD techniques proposed in this thesis are also applied to the misfire fault, the literature on FDD of ICE misfire faults is reviewed in this section.

Lee *et al.* (1997) used directional spectra to detect misfire faults in a four-cylinder compression and spark ignition engine. Signals measured from two vibration transducers that were placed perpendicular to each other were combined to construct a complex-valued signal. The maximum likelihood (ML) method was then used to estimate the directional power spectra (DPS) of this complex-valued signal. At the last stage, a MLP neural network was applied to recognize the fault by taking the DPS estimate as its input. Experiments showed the use of DPS achieved as high as 100% classification accuracy while the traditional Fourier transform (FT) method had an accuracy of only 90%. The ML also outperformed FT in terms of input data size: the ML used only 256 data points while 1024 points were used with the FT.

Porteiro *et al.* (2011) developed a multiple ANN system for diesel engine FDD. The system consisted of three levels of subsystems for decision, each comprising an ANN with different inputs. The input features were extracted from vibration signals at 15 positions and the measured temperatures of the exhaust gases, the lubricating oil and coolant. The first ANN estimated the load of the engine. Eight different loads ranging from idling to rated power were used. The second ANN detected the engine's condition, either normal or faulty condition, with the information provided by the first level and new inputs. If the second level decided that the engine was faulty, the third level was employed to determine the probable cause of the failure with the aid of the outputs of the first two levels and some new inputs. The system was capable of perfectly detecting

the engine load and of detecting the failure at the accuracy of 99%. The correctness for failure cause determination was 100% for misfiring in each of the three cylinders, 90.5% for clogged intake, and 91.7%, 75.0% and 66.7% for leaks in the 1st, 2nd or 3rd cylinder, respectively.

Lu *et al.* (2012) proposed an FDD method for a direct-injection three cylinder diesel engine. The FDD method was based on the empirical mode decomposition (EMD) and time-frequency image processing. The EMD decomposed the vibration signal of cylinder head surface into several intrinsic mode function (IMF) components. The first two IMFs were taken to reconstruct the signal in order to eliminate noise. After the pseudo Wigner-Ville time-frequency distribution was applied to obtain the signal's time-frequency image, features such as the absolute distance, the Euclidean distance and the first-order time moment were calculated. If the ratio of the test signal's feature to that of a certain baseline signal was close to 1, then the test condition belonged to that specific baseline condition. The objective was to distinguish the following conditions: normal condition (A), tight gross valve tappet clearance (D), excessive gross valve tappet clearance (C) and misfire condition (B). The experiments showed the three faulty conditions B, C and D could be easily distinguished, but there was some difficulty in distinguishing A.

Moosavian *et al.* (2015) developed an FDD approach to detect the wide spark plug gap fault in ICE based on vibration and acoustic signals. The vibration signal was obtained from an accelerometer mounted on one cylinder head and the acoustic emitted signal was captured by a microphone placed on top of the engine. A wavelet de-noising technique was applied to the signal before feature extraction. Then two Levenberg-Marquardt (LM) trained artificial neural networks (ANNs) as well as two least square support vector machines (LS-SVMs) were used to classify the fault according to the two different signals. Finally, the Dempster-Shafer (D-S) evidence theory was used for fusing classifiers. The experimental results showed that the classification accuracies of

ANN were 67.5% and 65.1% based on the acoustic and vibration signals, respectively. For LS-SVM, the classification accuracies of 65.1% and 57.9% were achieved based on the acoustic and vibration signals, respectively. By employing D–S theory to fuse the results, the classification accuracy reached 98.6%.

2.4 Summary

In this chapter, the applications of several data processing techniques and ANNs to FDD were reviewed, with a focus on the FDD of ICE misfire faults. The reviewed techniques included: Fourier transform, wavelet transform, EMD and VMD. We can clearly see the progress of these techniques. The signal to be processed was first assumed to be linear and periodic and can now be nonlinear and aperiodic. The progress of signal processing also prompted the development of improved FDD systems. Currently the relatively new VMD technique has not aroused significant attention in the FDD field. Since VMD has exhibited better performance than other signal processing techniques, its application to FDD is promising and will be investigated in this thesis. Considering the benefits of multi-variate analysis demonstrated by the literature, the combination of VMD and kernel PCA and their application to misfire FDD will also be investigated.

The application of ANNs to FDD reveals two factors that are crucial to successful FDD: the feature selection for the ANN's input and the use of multiple networks. However, while complicated input features and use of multiple networks may increase the classification accuracy it can also significantly increase the computational load. As a result, the research in this thesis will try to find a simple, but effective, ANN input and a new training algorithm that has more powerful generalization ability for ANNs with simple structures. If these aims are achieved the research will make a new contribution to the application of ANNs to FDD.

Chapter 3 Fault Diagnosis based on Data Processing Techniques

This chapter presents three novel FDD approaches based on data processing techniques. The underlying theory is introduced first, followed by detailed descriptions of the VMD-based FDD algorithm based on a new FDD index and two nonlinear multivariate multiscale FDD algorithms.

3.1 Underlying Theory

3.1.1 Crank Angle Domain Analysis

Crank angle domain (CAD) analysis is a commonly used tool for ICE fault detection. Crank angle refers to the rotation angle of the engine's crankshaft. An internal combustion engine consists of a number of rotary and reciprocating components. They complete a full cycle of movement when the crankshaft rotates from 0 to 720 degrees.

The essence of CAD analysis is to relate a measured signal (e.g. engine block vibration, cylinder pressure) to mechanical events in the engine (e.g. valve train intake, fuel injection) so that signal abnormality caused by defective components can be mapped to its crank angle position. By using CAD analysis, researchers found all engine dynamics, except for fuel dynamics, became less variable compared to time domain analysis. (Chin, et al., 1986)

Mapping data from time domain to crank angle domain can be done via several ways. One of them is to configure the data acquisition system to sample all sensors simultaneously according to the angle of the crankshaft. In other words, the acquisition system collects data only when it receives pulses sent by a camshaft position sensor. Another approach is to record data from all

sensors including the camshaft position sensor in the time domain, and then perform post processing. This approach is adopted in this thesis and its detailed procedure is as follows:

- 1) Collect data from all sensors as mentioned above.
- 2) Split the data into different engine cycles according to the peaks of the camshaft position sensor signal.
- 3) Regard the crank angle of the start sample in each engine cycle as 0 degree and map the data to crank angle domain using linear interpolation.

3.1.2 Wavelet Transform

The advantages of the wavelet transform were presented in Chapter 2. In this section, a brief introduction to the discrete version of the wavelet transform is presented, based on (Mallat, 2009).

Given the scaling function ϕ and wavelet function ψ , a discrete signal $f[n]$ in $L^2(\mathbb{Z})^1$ can be approximated as follows:

$$f[n] = \frac{1}{\sqrt{M}} \sum_k W_\phi[j_0, k] \phi_{j_0, k}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi[j, k] \psi_{j, k}[n]. \quad (\text{Eq. 3.1.2-1})$$

where M is the number of samples of the discrete signal; and $f[n]$, $\phi_{j_0, k}[n]$ and $\psi_{j, k}[n]$ are all discrete functions defined in $[0, M - 1]$. In addition:

$$\phi_{j, k}[n] = 2^{\frac{j}{2}} \phi(2^j n - k); \quad (\text{Eq. 3.1.2-2})$$

¹ The collection of Lebesgue 2nd integrable functions defined on \mathbb{Z} . See page 7 of (Sen, 2013).

$$\psi_{j,k}[n] = 2^{\frac{j}{2}}\psi(2^j n - k) \tag{Eq. 3.1.2-3}$$

where j is the scale parameter about dilation, or the visibility in frequency and k is the shift parameter about the position.

The sets $\{\phi_{j_0,k}[n]\} \in Z$ and $\{\psi_{j,k}[n]\} \in Z^2, j \geq j_0$ form an orthonormal basis of $L^2(Z)$. Thus by taking the inner product, the wavelet coefficients are obtained:

$$W_\phi[j_0, k] = \frac{1}{\sqrt{M}} \sum_n f[n]\phi_{j_0,k}[n] \text{ and} \tag{Eq. 3.1.2-4}$$

$$W_\psi[j, k] = \frac{1}{\sqrt{M}} \sum_n f[n]\psi_{j,k}[n], j \geq j_0 \tag{Eq. 3.1.2-5}$$

where (Eq. 3.1.2-4) are called approximation coefficients while (Eq.3.1.2-5) are called detailed coefficients.

There is a faster way to obtain the wavelet coefficients, which is called the fast wavelet transform. Its diagram is as shown in Fig. 3.1.1. $G_0(z)$ and $H_0(z)$ are low-pass filters, and $G_1(z)$ and $H_1(z)$ are high-pass filters. The notation $\downarrow 2$ means downsampling by 2. The outputs of low-pass filters and high-pass filters define the approximation coefficients (named, A_1, A_2, \dots, A_L) and detail coefficients (named D_1, D_2, \dots, D_L) respectively.

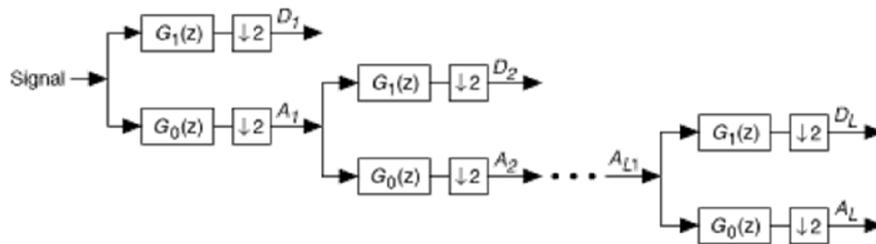


Figure 3.1.1 Schematic diagram for the fast wavelet transform. (NI, 2014)

This algorithm is faster because one can find the coefficients level by level rather than directly using (Eq. 3.1.2-4) and (Eq. 3.1.2-5) to find the coefficients which requires the knowledge of the scaling and dilation version of scaling and wavelet function.

The wavelet transform is also called wavelet decomposition or analysis. And the inverse wavelet transform is also known as wavelet reconstruction or synthesis. The diagram of the latter is shown in Fig. 3.1.2. The notation $\uparrow 2$ means upsampling by 2. The relationship between H_0 and G_0 is $H_0[n] = G_0[-n]$ and that between H_1 and G_1 is $H_1[n] = G_1[-n]$.

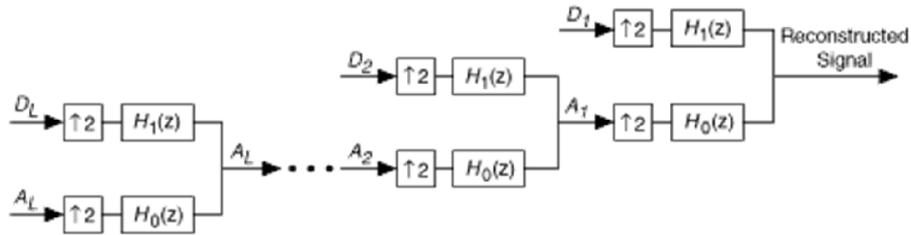


Figure 3.1.2 Schematic diagram for the fast wavelet reconstruction. (NI, 2014)

3.1.3 Variational Mode Decomposition

VMD is a newly developed method for adaptive signal decomposition which can non-recursively decompose a multi-component signal into a number of quasi-orthogonal intrinsic mode functions. Each mode is band-limited and is compact around a center. VMD can be stated as follows (Dragomiretskiy, et al., 2014):

$$\min_{\{u_k\}, \{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t} \right) * u_k(t) \right] e^{-i\omega_k t} \right\|_2^2 \right\} \quad s. t. \sum_k u_k = f \quad (\text{Eq. 3.1.3-1})$$

where f is a real valued input signal. δ is the Dirac distribution and $*$ denotes convolution. i is the imaginary unit and is defined as $i^2 = -1$. K is the number of modes. $\{u_k\} := \{u_1, \dots, u_K\}$ and

$\{\omega_k\} := \{\omega_1, \dots, \omega_K\}$ are shorthand notations for the set of all modes and their center frequencies, respectively. Similarly, $\sum_k := \sum_{k=1}^K$ is understood as the summation over all modes.

To solve the constrained variational problem above, the following augmented Lagrangian is introduced:

$$L(\{u_k\}, \{\omega_k\}, \zeta) := \beta \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t} \right) * u_k(t) \right] e^{-i\omega_k t} \right\|_2^2 + \|f(t) - \sum_k u_k(t)\|_2^2 + \langle \zeta(t), f(t) - \sum_k u_k(t) \rangle. \quad (\text{Eq. 3.1.3-2})$$

The first two terms are the quadratic penalty part and β is the weight of the penalty term, indicating the tightness of the band-limit of each IMF. β is inversely proportional to the noise level in the input signal and by increasing it, the bandwidth of IMF is decreased. ζ in the last term (the inner product term) is the Lagrangian multiplier.

The solution to the original minimization problem (Eq. 3.1.3-1) is now found as the saddle point of the augmented Lagrangian in a sequence of iterative sub-optimizations called alternate direction method of multipliers (ADMM). The complete VMD algorithm is as follows:

Initialize $\{\hat{u}_k^1\}, \{\omega_k^1\}, \zeta^1, n \leftarrow 0$

repeat

$n \leftarrow n + 1$

for $k = 1:K$ **do**

Update \hat{u}_k for all $\omega \geq 0$:

$$\hat{u}_k^{n+1}(\omega) \leftarrow \frac{\hat{f}(\omega) - \sum_{i < k} \hat{u}_i^{n+1}(\omega) - \sum_{i > k} \hat{u}_i^n(\omega) + \frac{\zeta^n(\omega)}{2}}{1 + 2\beta(\omega - \omega_k^n)^2}$$

Update ω_k :

$$\omega_k^{n+1} \leftarrow \frac{\int_0^\infty \omega |\hat{u}_k^{n+1}(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k^{n+1}(\omega)|^2 d\omega}$$

end for

Dual ascent for all $\omega \geq 0$:

$$\hat{\zeta}^{n+1}(\omega) \leftarrow \hat{\zeta}^n(\omega) + \tau(\hat{f}(\omega) - \sum_k \hat{u}_i^{n+1}(\omega))$$

until convergence: $\sum_k \frac{\|\hat{u}_k^{n+1} - \hat{u}_k^n\|_2^2}{\|\hat{u}_k^n\|_2^2} < \epsilon$.

In the algorithm above, the hat notation (\hat{f} , \hat{u} , $\hat{\zeta}$) indicates the Fourier transform of the corresponding function. Two new parameters are introduced *i.e.* τ which indicates the time-step of the dual ascent and ϵ which indicates the tolerance of the convergence criterion. They need to be set manually in the implementation of the algorithm.

3.1.4 Principal Component Analysis

PCA is used to model a multivariate data set and extract the relationship between variables. It has been widely applied to the process monitoring field.

Assuming \mathbf{X} is an $n \times m$ data matrix which encompasses n measurements (or observations) of m variables and the mean of the elements in each column of \mathbf{X} is zero, then PCA aims to find a matrix \mathbf{P} such that after transformation $\mathbf{T} = \mathbf{XP}$, where \mathbf{T} is an $n \times m$ matrix and each pair of columns of \mathbf{T} is uncorrelated. This can be done by taking the singular value decomposition (SVD) on \mathbf{X} , which is shown below:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_m & \\ & & & 0 \end{bmatrix} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_m]^T, \text{ (Eq. 3.1.4-1)}$$

where the columns of \mathbf{U} are orthogonal and so are those of \mathbf{V} . Plus, $\sigma_1 > \sigma_2 > \dots > \sigma_m$. As a result:

$$\mathbf{XV} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V} = \mathbf{U}\mathbf{\Sigma}. \quad (\text{Eq. 3.1.4-2})$$

Taking $\mathbf{P} = \mathbf{V}$, the covariance matrix of \mathbf{X} can be obtained as follows:

$$\begin{aligned} \mathbf{\Sigma}_T &= \frac{1}{n-1} \mathbf{T}^T \mathbf{T} = \frac{1}{n-1} (\mathbf{XP})^T (\mathbf{XP}) = \frac{1}{n-1} (\mathbf{XV})^T (\mathbf{XV}) \\ &= \frac{1}{n-1} (\mathbf{U}\mathbf{\Sigma})^T (\mathbf{U}\mathbf{\Sigma}) = \frac{1}{n-1} \mathbf{\Sigma}^T \mathbf{U}^T \mathbf{U} \mathbf{\Sigma} = \frac{1}{n-1} \mathbf{\Sigma}^T \mathbf{\Sigma} = \mathbf{\Lambda}, \end{aligned} \quad (\text{Eq. 3.1.4-3})$$

where $\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} = \frac{1}{n-1} \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{bmatrix}$ is a diagonal matrix.

The elements of \mathbf{T} in the equations above are called the principal component scores and the columns of \mathbf{P} are called the principal component loadings, or simply the principal components (PCs).

The next step is to threshold these diagonal elements in $\mathbf{\Lambda}$. Suppose the first l elements are retained, then $\mathbf{\Lambda}$ can be written as

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{\Lambda}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Lambda}} \end{bmatrix}, \quad (\text{Eq. 3.1.4-4})$$

where $\hat{\mathbf{\Lambda}}$ is $l \times l$ and $\tilde{\mathbf{\Lambda}}$ is of $(m-l) \times (m-l)$.

Correspondingly,

$$\begin{aligned}
\mathbf{X} &= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_m & \\ & & & 0 \end{bmatrix} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_m]^T \\
&= [\sigma_1 \mathbf{u}_1 \quad \sigma_2 \mathbf{u}_2 \quad \cdots \quad \sigma_m \mathbf{u}_m] \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_m^T \end{bmatrix} \\
&= [\sigma_1 \mathbf{u}_1 \quad \cdots \quad \sigma_l \mathbf{u}_l \quad \sigma_{l+1} \mathbf{u}_{l+1} \quad \cdots \quad \sigma_m \mathbf{u}_m] \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_l^T \\ \mathbf{v}_{l+1}^T \\ \vdots \\ \mathbf{v}_m^T \end{bmatrix} \\
&= [\hat{\mathbf{T}} \quad \tilde{\mathbf{T}}] \begin{bmatrix} \hat{\mathbf{P}}^T \\ \tilde{\mathbf{P}}^T \end{bmatrix} \tag{Eq. 3.1.4-5}
\end{aligned}$$

After the PCA models the baseline dataset, one can monitor the interested process by calculating the Hotelling's \mathcal{J}^2 and the Q-statistic (also known as the squared prediction error, SPE), and then comparing them to their upper limits. (Qin, 2003)

The Hotelling's \mathcal{J}^2 is defined as follows:

$$\mathcal{J}^2 = \mathbf{x}_{new} \hat{\mathbf{P}} \hat{\mathbf{\Lambda}}^{-1} \hat{\mathbf{P}}^T \mathbf{x}_{new}^T, \tag{Eq. 3.1.4-6}$$

where \mathbf{x}_{new} is a vector of new observations with the size of $1 \times m$.

The Q-statistic/SPE is defined as follows:

$$Q = \|\tilde{\mathbf{x}}_{new}\|^2 = \|\mathbf{x}_{new} \tilde{\mathbf{C}}\|^2, \tag{Eq. 3.1.4-7}$$

where $\tilde{\mathbf{C}} = \tilde{\mathbf{P}} \tilde{\mathbf{P}}^T$. In fact $\tilde{\mathbf{x}}_{new}$ can be written as the projection of the new observation vector \mathbf{x}_{new} to the residual subspace using the $\tilde{\mathbf{C}}$ matrix, i.e. $\tilde{\mathbf{x}}_{new} = \mathbf{x}_{new} \tilde{\mathbf{C}} = \mathbf{x}_{new} \tilde{\mathbf{P}} \tilde{\mathbf{P}}^T$.

Assuming the monitored variables possess multivariate normal distribution, for a given significance level α , the upper limits of the Hotelling's \mathcal{J}^2 is:

$$\tau_{\alpha}^2 = \frac{l(n-1)}{n-l} F_{\alpha}(l, n-l), \quad (\text{Eq. 3.1.4-8})$$

where l is the number of selected elements after thresholding and n is the number of samples.

Under the same assumption, the upper limits of the Q-statistic/SPE for a given significance level α is:

$$\delta_{Q,\alpha} = \left(\frac{v}{2m}\right) \chi_{\alpha}^2 \left(\frac{2m^2}{v}\right), \quad (\text{Eq. 3.1.4-9})$$

where m is the sample mean for \mathcal{J}^2 and v is its covariance.

As long as $\mathcal{J}^2 < \tau_{\alpha}^2$ and $Q < \delta_{Q,\alpha}$ the process is considered normal with confidence $(1 - \alpha) \cdot 100\%$.

3.1.5 Kernel Principal Component Analysis

Although PCA is a very effective multivariate statistical method that is widely applied in process monitoring, it is inherently a linear transformation. Thus its performance degrades when it is used to monitor non-linear systems, like the internal combustion engine. To solve this problem, PCA has been extended to several nonlinear versions, among which kernel PCA (KPCA) is one of the most popular methods. The following is a brief description for KPCA based on (Choi, et al., 2005).

As with PCA, the dataset \mathbf{X} has n observations in an m -dimensional space $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m$. Originally, KPCA consisted of two steps, the first of which is non-linear mapping from the original input space to the feature space \mathcal{F} , i.e. $\Phi: \mathbf{x} \in \mathbb{R}^m \rightarrow \Phi(\mathbf{x}) \in \mathcal{F}$. The second step is to perform

the PCA on those mapped samples by finding the eigenvalues λ and corresponding eigenvectors $\mathbf{v} \in \mathcal{F}$ satisfying:

$$\lambda \mathbf{v} = \left(\frac{1}{n} \sum_{j=1}^n \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)^T \right) \mathbf{v}. \quad (\text{Eq. 3.1.5-1})$$

However, to avoid the “curse of dimensionality”, KPCA is performed using the so-called “kernel trick” in order to avoid expressing Φ explicitly. Then the KPCA problem can be solved by finding the eigenvalues λ and eigenvectors ξ from the following equation:

$$n\lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\xi}, \quad (\text{Eq. 3.1.5-2})$$

where

$$\mathbf{K} = \begin{bmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \vdots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{bmatrix}. \quad (\text{Eq. 3.1.5-3})$$

Its element at the i^{th} row and j^{th} column is $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Here the angle bracket is the notation of inner product and $k(\mathbf{x}_i, \mathbf{x}_j)$ is called the kernel function and in this research it adopts the form below:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c)^d, \quad (\text{Eq. 3.1.5-4})$$

where c and d are tuning parameters. Assuming the number of the retained PCs is a , the score vector $\mathbf{t} = [t_1, t_2, \dots, t_a]^T$ is obtained by:

$$t_k = \sum_{i=1}^n \xi_i^l k(\mathbf{x}_i, \mathbf{x}), l = 1, 2, \dots, a. \quad (\text{Eq. 3.1.5-5})$$

where ξ_i^l is the i^{th} element of the l^{th} eigenvector obtained from (Eq. 3.1.5-2).

There are two points that need to be considered before solving the eigenvalue problem. First mean centering has to be performed in feature space \mathcal{F} , which can be done by substituting the kernel matrix \mathbf{K} with

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{E}_M \mathbf{K} - \mathbf{K} \mathbf{E}_M + \mathbf{E}_M \mathbf{K} \mathbf{E}_M. \quad (\text{Eq. 3.1.5-6}).$$

Here $\mathbf{E}_M \in \mathbb{R}^{n \times n}$ and each of its elements equals $1/n$. Moreover, ξ should be scaled in such a manner that $\|\xi\|^2 = 1/n\lambda$ in order to ensure the normality of the PCs, i.e. $\|\mathbf{v}\|^2 = 1$.

To perform KPCA-based monitoring, the Hotelling's \mathcal{J}^2 statistic and the Q-statistic should be modified in the following way:

$$\mathcal{J}^2 = \mathbf{t} \hat{\mathbf{\Lambda}}^{-1} \mathbf{t}^T, \quad (\text{Eq. 3.1.5-7})$$

$$Q = \sum_{i=1}^b t_i^k - \sum_{i=1}^a t_i^k, \quad (\text{Eq. 3.1.5-8})$$

where $\hat{\mathbf{\Lambda}}^{-1}$ is the diagonal matrix of the inverse of the eigenvalues associated with the retained PCs and b is the number of non-zero eigenvalues generated from (Eq. 3.1.5-2) among the total n eigenvalues.

3.2 VMD-Based FDD Algorithm

As mentioned in Section 2.1, VMD is a recently proposed and powerful signal processing tool. Its application to FDD is promising. This section describes a new FDD index based on VMD and the use of vibration data acquired from one sensor.

Assume after the crank angle domain (CAD) analysis, n ICE working cycles of data in CAD are obtained from the data in the time domain and each cycle contains M samples of data. Data in cycle k is denoted as:

$$\mathbf{s}_k = \begin{bmatrix} s_{k,1} \\ s_{k,2} \\ \vdots \\ s_{k,M} \end{bmatrix}. \quad (\text{Eq. 3.2-1})$$

And the vibration dataset after the CAD is denoted as:

$$\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\} = \{\mathbf{s}_k\}_{k=1}^n. \quad (\text{Eq. 3.2-2})$$

Then the data in each cycle are analyzed by VMD using the same settings. Suppose the desired intrinsic mode function (IMF) number is m , then the \mathbf{s}_k will be decomposed into m IMFs by VMD and each IMF has M elements. Denote the VMD result as:

$$\mathbf{T}_k = \{\mathbf{t}_k^1, \mathbf{t}_k^2, \dots, \mathbf{t}_k^m\} = \{\mathbf{t}_k^i\}_{i=1}^m, \quad (\text{Eq. 3.2-3})$$

where the superscript is the index of the IMF, *e.g.* the i^{th} IMF in \mathbf{T}_k is

$$\mathbf{t}_k^i = \begin{bmatrix} t_{k,1}^i \\ t_{k,2}^i \\ \vdots \\ t_{k,M}^i \end{bmatrix}. \quad (\text{Eq. 3.2-4})$$

The next step is to combine each cycle's IMF component that share the same mode index into one matrix. For example, \mathbf{t}_1^i which is the i^{th} IMF of Cycle 1's vibration data \mathbf{s}_1 , \mathbf{t}_2^i which is the i^{th} IMF of Cycle 2's vibration data \mathbf{s}_2 ,, and \mathbf{t}_n^i which is the i^{th} IMF of Cycle n 's vibration data \mathbf{s}_n are combined into one matrix. Denote this matrix as:

$$\mathbf{Z}_i = [\mathbf{t}_1^i \quad \mathbf{t}_2^i \quad \cdots \quad \mathbf{t}_n^i] = \begin{bmatrix} t_{1,1}^i & t_{2,1}^i & \cdots & t_{n,1}^i \\ t_{1,2}^i & t_{2,2}^i & \cdots & t_{n,2}^i \\ \vdots & \vdots & \vdots & \vdots \\ t_{1,M}^i & t_{2,M}^i & \cdots & t_{n,M}^i \end{bmatrix}, \quad (\text{Eq. 3.2-5})$$

where i indicates the i^{th} IMF. So there are totally m such matrices, which are $\mathbf{Z}_1, \mathbf{Z}_2, \dots$, and \mathbf{Z}_m .

Each row of \mathbf{Z}_i corresponds to a certain crank angle. For example, the second row of \mathbf{Z}_i , i.e.

$[t_{1,2}^i \quad t_{2,2}^i \quad \cdots \quad t_{n,2}^i]$ corresponds to $\frac{2}{M} \cdot 720$ degree.

As a process history-based FDD technique, this approach requires data obtained under both healthy (baseline) condition and test condition. The approach denotes the data at a certain mode and at a certain crank angle under the healthy condition as $\{X_i\}_{i=1}^n$ and assumes the random variable complies with the same normal distribution, then $X_i \sim N(\mu, \sigma^2)$, where μ is the distribution mean and σ^2 is the distribution variance. According to statistics (Dekking, et al., 2005), the unbiased estimates of μ and σ calculated using $\{X_i\}_{i=1}^n$ are:

$$\hat{\mu} = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i; \quad (\text{Eq. 3.2-6})$$

$$\hat{\sigma} = S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}, \quad (\text{Eq. 3.2-7})$$

where \bar{X} is the sample mean and S is the sample variance.

The next step is to perform the same operations on the test data and denote the data at the same mode and at the same angle as $\{Y_i\}_{i=1}^n$. If the test condition is healthy, then the following conditions hold:

$$Y_i \sim N(\mu, \sigma^2); \quad (\text{Eq. 3.2-8})$$

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i \sim N\left(\mu, \frac{\sigma^2}{n}\right); \quad (\text{Eq. 3.2-9})$$

$$\frac{\bar{Y} - \mu}{\frac{\sigma}{\sqrt{n}}} \approx \frac{\bar{Y} - \hat{\mu}}{\frac{\hat{\sigma}}{\sqrt{n}}} \sim N(0,1); \quad (\text{Eq. 3.2-10})$$

Therefore, at the probability of $(1 - \alpha) \cdot 100\%$,

$$-z_{\frac{\alpha}{2}} \leq \frac{\bar{Y} - \hat{\mu}}{\frac{\hat{\sigma}}{\sqrt{n}}} \leq z_{\frac{\alpha}{2}}, \quad (\text{Eq. 3.2-11})$$

where $z_{\frac{\alpha}{2}}$ is the upper $\frac{\alpha}{2}$ quantile of standard normal distribution and $P\left\{\left|\frac{\bar{Y} - \hat{\mu}}{\frac{\hat{\sigma}}{\sqrt{n}}}\right| > z_{\frac{\alpha}{2}}\right\} = \alpha$.

Then the new FDD index at mode k and crank angle ρ is designated as

$$I(k, \rho) = \frac{|\bar{Y} - \hat{\mu}|}{\hat{\sigma} \cdot z_{\frac{\alpha}{2}} / \sqrt{n}}. \quad (\text{Eq. 3.2-12})$$

If $I(k, \rho) \leq 1$, the engine is working in its normal condition; if $I(k, \rho) > 1$, a fault has happened at mode k and crank angle ρ with the probability of $(1 - \alpha) \cdot 100\%$. Usually when a fault occurs in an internal combustion engine, several modes and crank angles are affected. Assuming only one fault occurs at a time, the maximum index indicates the mode and the crank angle of the fault.

Now the algorithm for the VMD-based FDD approach described above can be summarized as follows:

Step 1: Get data from one accelerometer²;

Step 2: Map the data from time domain to the crank angle domain;

² Larger data volume increases the FDD accuracy.

Step 3: Perform the VMD and obtain data at different modes;

Step 4: For each mode and each crank angle

- 1) Baseline data (obtained under ICE normal working condition): calculate its sample mean \bar{X} and sample standard deviation S according to (Eq. 3.2-6) and (Eq. 3.2-7);
- 2) Test Data: calculate its sample mean \bar{Y} according to (Eq. 3.2-9);
- 3) Calculate the index $I(k, \rho)$ according to (Eq. 3.2-12);

Step 5: Find the maximum index;

Step 6: If it is above 1, the mode and the crank angle of the fault are those corresponding to the maximum index; and

Step 7: If it is below 1 then no fault has been detected.

The flow chart of this algorithm is exhibited as Fig. 3.1.3.

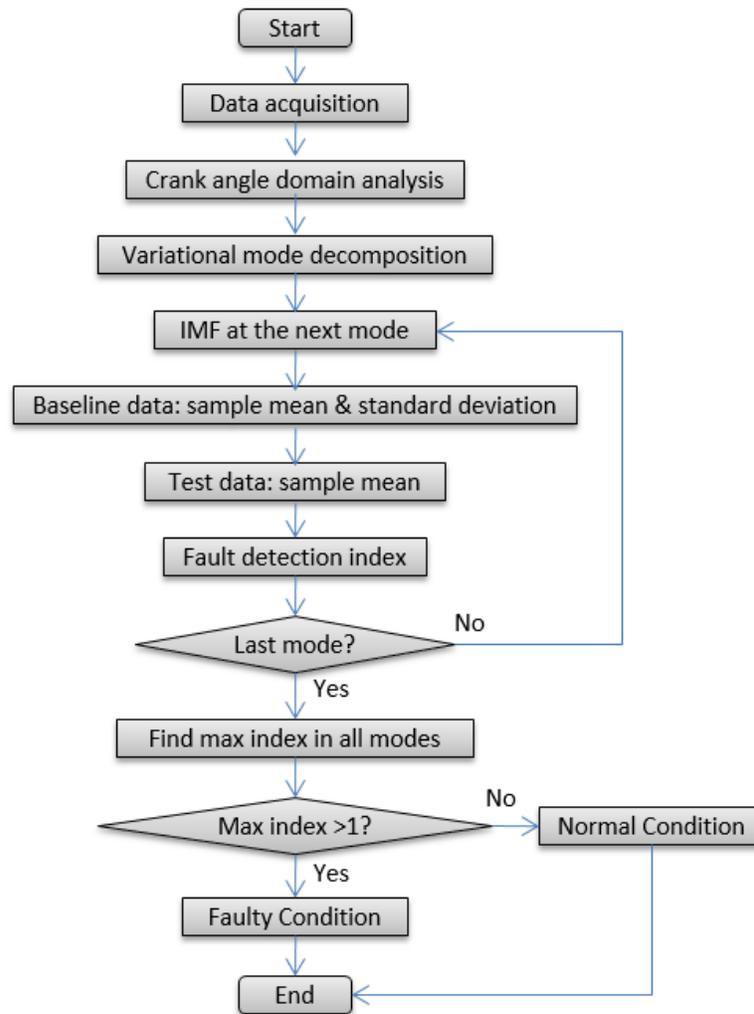


Figure 3.1.3 Flow chart of the VMD-based FDD algorithm.

To implement the algorithm, one needs to properly choose the relevant parameters. The number of modes (K) should be large enough to cover the major frequency peaks of the analyzed signal. According to (Dragomiretskiy, et al., 2014), when K is smaller than the proper value, *i.e.* when underbinning, the spectrum of the IMFs overlap. When K is larger than the proper value, *i.e.* when overbinning, the center frequencies of the IMFs coincide if the weight of the penalty term of (Eq. 3.1.3-2), β , has a large value. One needs to consider the degree of noise of the input signal when tuning β because it is inversely proportional to the noise level in the input signal. The

tolerance of convergence criterion ϵ is typically around 10^{-6} . The time-step of the dual ascent, τ , is picked as 0 for noise-slack.

3.3 Nonlinear Multivariate and Multiscale Fault Diagnosis Techniques

Discrete wavelet transform (DWT) and VMD are both powerful time-frequency analysis tools that decompose a signal into several components with different frequency contents. KPCA is a robust tool for multivariate analysis. The combination of the time-frequency analysis tool (either DWT or VMD) and KPCA creates a strong nonlinear multivariate multiscale FDD method. The complete algorithm of this method is as follows:

Part A: Build Reference Model

Step 1: Under normal ICE working condition, acquire baseline data matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$ from the sensors, where n is the number of observations and m is the number of variables.

Step 2: Shift and scale each variable to have zero mean and unity standard deviation.

Step 3: Decompose each column $\mathbf{y} \in \mathbb{R}^n$ of the data matrix \mathbf{Y} by means of DWT or VMD into S different levels.

Step 4: For each level s ,

- 1) Combine the decomposition components of all variables into an m -by- n matrix $\mathbf{C}_s = [\mathbf{c}_{1s}, \mathbf{c}_{2s}, \dots, \mathbf{c}_{ms}]^T$;
- 2) Compute the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ according to (Eq. 3.1.5-3) and (Eq. 3.1.5-4);
- 3) Mean center \mathbf{K} by replacing it with (Eq. 3.1.5 – 6);
- 4) Find eigenvalues λ and eigenvectors $\boldsymbol{\alpha}$ from (Eq. 3.1.5-2) and scale $\boldsymbol{\alpha}$ so that $\|\boldsymbol{\alpha}\|^2 = 1/n\lambda$;
- 5) Calculate score vector $\mathbf{t} = [t_1, t_2, \dots, t_a]^T$ from (Eq. 3.1.5-5).

- 6) Select a suitable number of PCs and calculate monitoring statistics \mathcal{T}^2 and Q according to (Eq. 3.1.5-7) and (Eq. 3.1.5-8), and calculate their control limits according to (Eq. 3.1.4-8) and (Eq. 3.1.4-9).

Part B: Monitor the system

Step 1: Under test conditions, acquire test data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, where n is the number of observations and m is the number of variables.

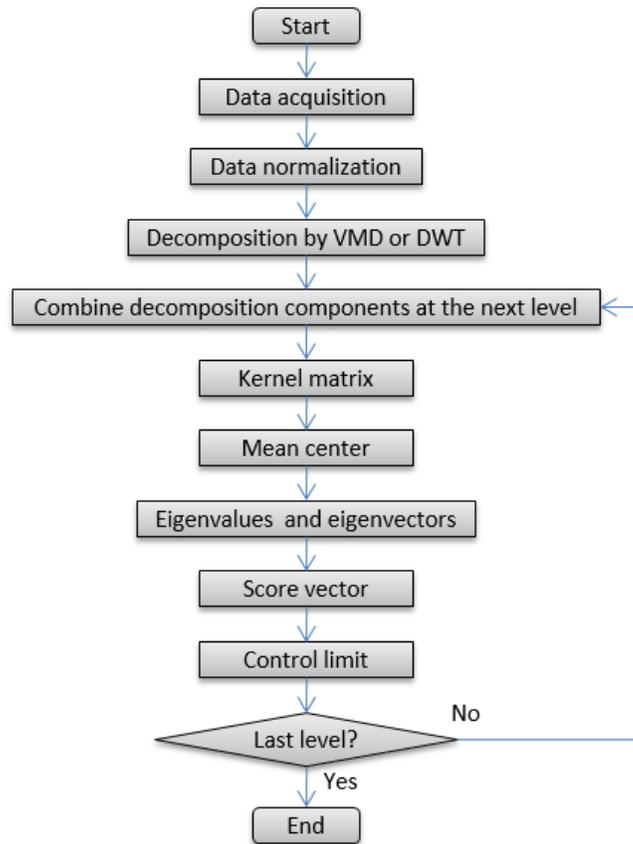
Step 2: Shift and scale each variable to have zero mean and unity standard deviation.

Step 3: Decompose each column $\mathbf{x} \in \mathbb{R}^n$ of the data matrix \mathbf{X} by means of DWT or VMD into S different levels.

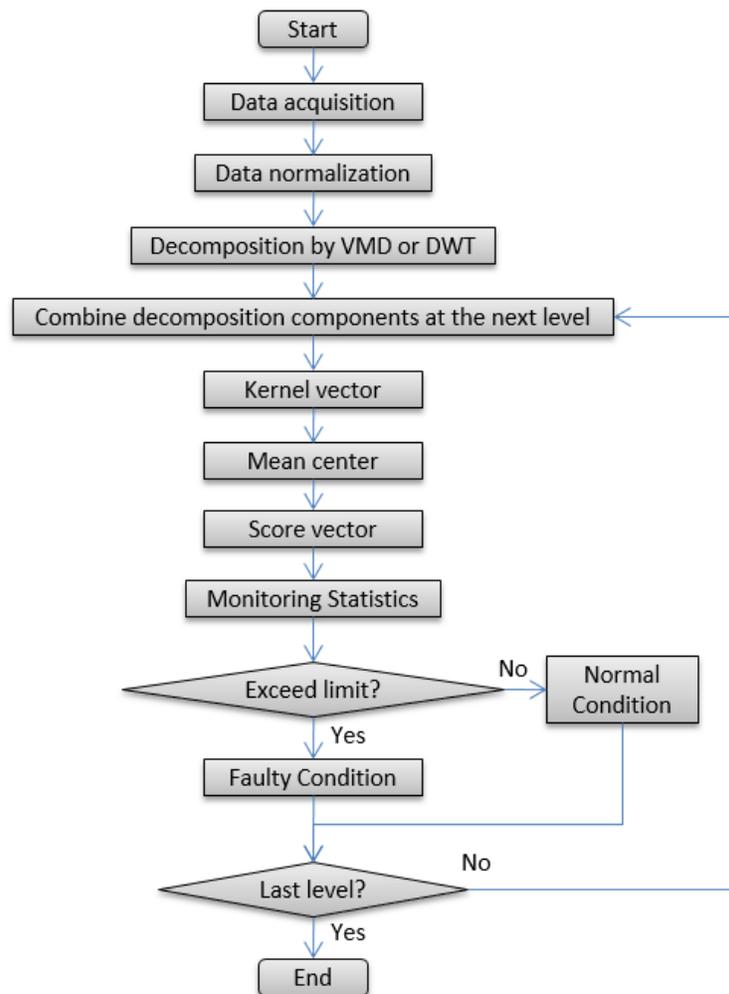
Step 4: For each level s ,

- 1) Combine the decomposition components of all variables into an m -by- n matrix $\mathbf{D}_s = [\mathbf{d}_{1s}, \mathbf{d}_{2s}, \dots, \mathbf{d}_{ms}]^T$;
- 2) Compute the test kernel vector $\mathbf{k} \in \mathbb{R}^{1 \times n}$ where the i^{th} ($i \leq n$) element of \mathbf{k} is $k(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + c)^d$. Here \mathbf{x}_i is the scaled baseline data used in Part A Step 4-b;
- 3) Mean center \mathbf{k} by replacing it with $\tilde{\mathbf{k}} = \mathbf{k} - \mathbf{E}_c \mathbf{K} - \mathbf{k} \mathbf{E}_M + \mathbf{E}_c \mathbf{K} \mathbf{E}_M$ where $\mathbf{E}_c \in \mathbb{R}^{1 \times n}$ and each of its elements is equal to $1/n$;
- 4) Calculate nonlinear components via (Eq. 3.1.5-5);
- 5) Calculate the monitoring statistics \mathcal{T}^2 and Q according to (Eq. 3.1.5-7) and (Eq. 3.1.5-8) and compare them with the control limits obtained in Part A step 4-f.
- 6) If the statistics are below their control limits, then the engine is working at its normal condition; otherwise a fault has been detected and the peak of the statistics indicates the crank angle and (signal decomposition) level of the fault.

The flow chart of this algorithm is exhibited as Fig. 3.1.4.



(a) Flow chart for Part A, *i.e.* Build Reference Model

(a) Flow chart for Part B, *i.e.* Monitor the system**Figure 3.1.4** Flow chart of the multiscale KPCA algorithm.

3.4 Summary

In this chapter, three novel FDD methods based on data processing techniques are proposed. The first method, the VMD-based FDD algorithm, introduces a new FDD index which integrates VMD and statistics. It is expected to inherit the characteristics of VMD of being robust to noise and being suitable to analyzing non-linear and non-stationary signals. The last two FDD methods are types of multi-scale nonlinear PCA, which are based on the VMD/DWT and the KPCA algorithm.

The new VMD/DWT based KPCA is expected to have the ability of extracting fault feature from high-dimensional, noisy and non-linearly correlated data.

Chapter 4 Artificial Neural Network for Fault Classification

This chapter describes the application of artificial neural networks (ANNs) in ICE fault classification. The basic theory of two well-known ANNs — the multi-layer perceptron (MLP) network and the (radial basic function) RBF network are first introduced, followed by a brief description of the smooth variable structure filter (SVSF). Then the SVSF-based training algorithm for the MLP network is reviewed. A novel SVSF-based training algorithm for RBF networks is presented next. The last section of this chapter focuses on the solution to the ANN input selection problem.

4.1 Underlying Theory

4.1.1 Multi-layer Perceptron Network

This section is based on (Haykin, 2008). The structure of a MLP network is shown in Fig. 4.1.1. The network consists of one input layer, several hidden layers (in this example there are two) and one output layer. Each layer has several neurons (also called “nodes”). The network shown here and in this research is fully connected, which means that a neuron in any layer of the network is connected to all the neurons in the previous layer. A neuron has several inputs and one output. If the neuron is in the first hidden layer, then its inputs are external signals; otherwise, they are the outputs of the neurons in the previous layer. For example, the output of the neuron j in Fig. 4.1.2 is $y_j(n)$ and it is one of the inputs for neuron k which lies in the output layer of the MLP. $d_k(n)$ is the target output for neuron k and $e_k(n)$ is the error which equals the difference between the target output and the actual output, i.e. $e_k(n) = d_k(n) - y_k(n)$.

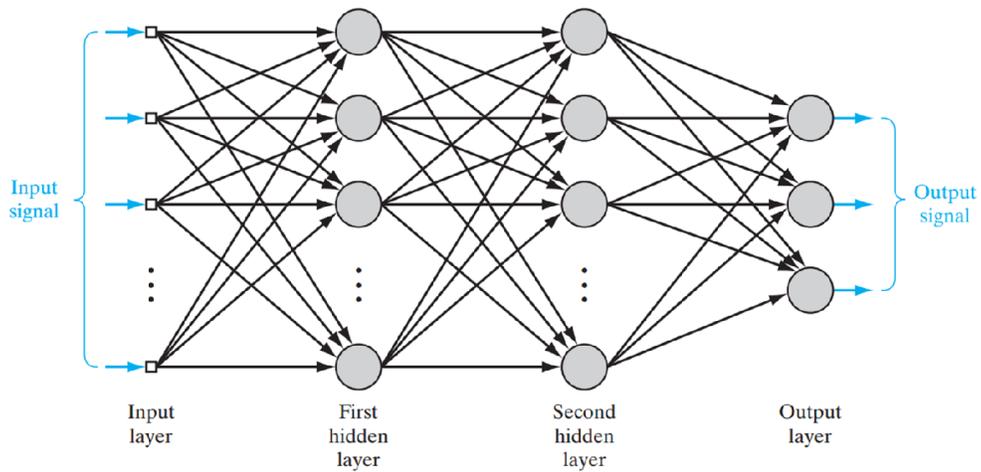


Figure 4.1.1 Structure of a MLP network with two hidden layers (Haykin, 2008).

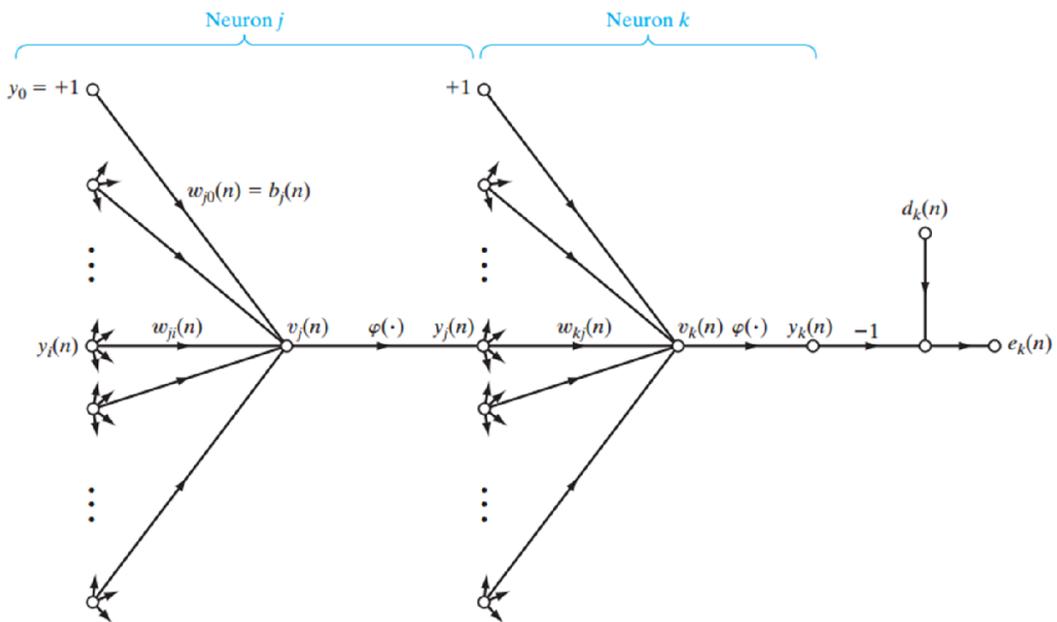


Figure 4.1.2 Detailed structure of a neuron (Haykin, 2008).

Suppose the layer before Neuron j has m neurons. Then taking Neuron j as an example, the relationship between its inputs and output is:

$$y_k(n) = \varphi\left[\sum_{i=1}^m w_{ji}(n)y_i(n) + b_j(n)\right]. \tag{Eq. 4.1.1-1}$$

Here $w_{ji}(n)$ denotes the link weight from neuron i to neuron j , $b_j(n)$ is the node offset (or bias) and $\varphi(\cdot)$ is the activation function may be one of several forms.

In this research, the MLP network has three layers with 10 neurons in the hidden layer and 8 neurons in the output layer. The activation function in the hidden layer is the log-sigmoid transfer function shown in Figure 4.1.3.

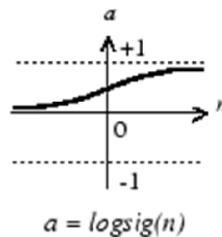


Figure 4.1.3 The log-sigmoid transfer function (Mathworks, logsig, 2015)

The activation function in the output layer is the linear transfer function shown in Figure 4.1.4.

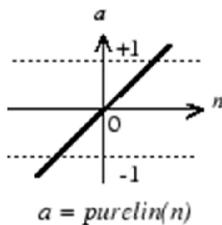


Figure 4.1.4 The linear transfer Function (Mathworks, purelin, 2015)

An important stage of using an ANN is training. Supervised training is used in this research. The essence of supervised training is to find the values of the network parameters that minimize the error between the target output and the actual output. In the MLP network, that refers to determining the link weights and biases for each neuron.

4.1.2 Radial Basis Function Network

This section is based on (Simon, 2002). The structure of the RBF network is shown in Fig. 4.1.5. It consists of three layers — the input layer, the hidden layer and the output layer. Unlike the MLP network, the RBF network has only one hidden layer and the link weights between the input layer and the hidden layer are all set equal to unity.

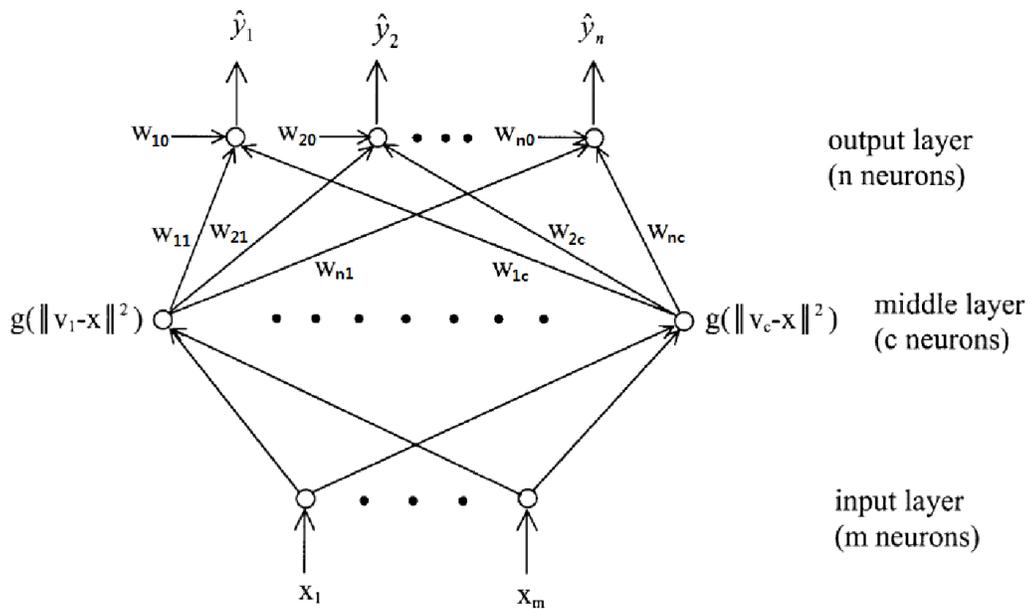


Figure 4.1.5 Structure of the RBF network (Simon, 2002).

The RBF network performs two operations on the input signal before it produces the output signal. The first operation is a nonlinear mapping, after which a complex pattern classification problem is more likely to be linearly separable. The nonlinear mapping is done by the activation function $g(\cdot)$ whose expression in this research is as follows (Simon, 2002) :

$$g(v) = [g_0(v)]^{\frac{1}{1-p}}, \tag{Eq. 4.1.2-1}$$

$$g_0(v) = v + \tau^2 \text{ and} \quad (\text{Eq. 4.1.2-2})$$

$$v = \|\mathbf{v}_i - \mathbf{x}\|^2, i = 1, 2, \dots, c. \quad (\text{Eq. 4.1.2-3})$$

Here $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m]$ is the input signal. $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$ are the prototype vectors, m is the dimension of the inputs, c is the number of the hidden neurons, p is a real number that is larger than 1 and τ is a manually tuned parameter.

The second operation is a linear transformation. Referring to Figure 4.1.5, $\hat{\mathbf{y}} = [\hat{y}_1 \ \hat{y}_2 \ \dots \ \hat{y}_n]$ is the actual output of the RBF network, $[w_{10} \ w_{20} \ \dots \ w_{n0}]$ are the biases for each output

neuron and $\begin{bmatrix} w_{11} & \dots & w_{1c} \\ \vdots & \vdots & \vdots \\ w_{n1} & \dots & w_{nc} \end{bmatrix}$ are the link weights. The relationship between the RBF network's

input and its output is as follows:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} w_{10} & w_{11} & \dots & w_{1c} \\ w_{20} & w_{21} & \dots & w_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n0} & w_{n1} & \dots & w_{nc} \end{bmatrix} \begin{bmatrix} 1 \\ g(\|\mathbf{x} - \mathbf{v}_1\|) \\ \vdots \\ g(\|\mathbf{x} - \mathbf{v}_c\|) \end{bmatrix}, \quad (\text{Eq. 4.1.2-4})$$

The training process of the RBF network should determine the prototype vectors, the biases and the link weights.

4.1.3 Smooth Variable Structure Filter

The smooth variable structure filter (SVSF) proposed by Habibi (2007) is a robust estimator which can guarantee convergence given bounded uncertainties on the states. This estimator involves an inherent switching action that forces the estimated states to converge towards the true state trajectory and remain within a neighbourhood referred to as the existence subspace. The concept of the SVSF is shown in Fig. 4.1.6.

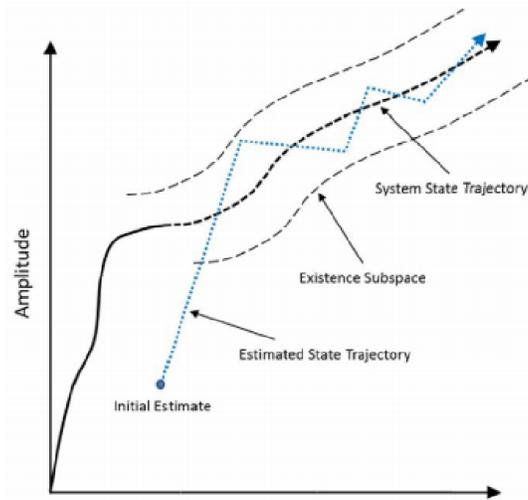


Figure 4.1.6 Concept of the SVSF (Habibi, 2007).

The SVSF can be applied to both linear and nonlinear systems and has a predictor-corrector structure as shown in Fig. 4.1.7.

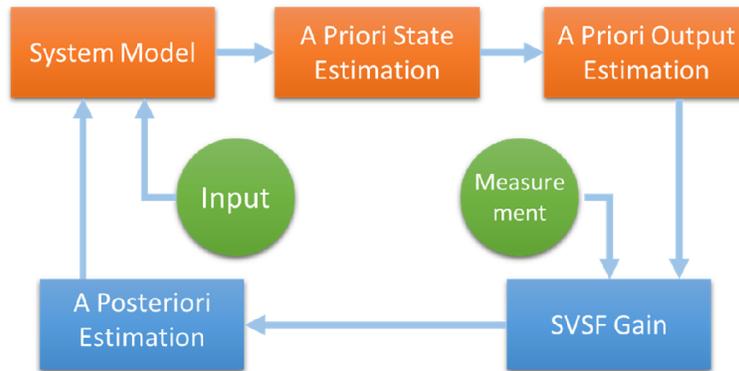


Figure 4.1.7 Structure of the SVSF (Song, 2013).

The SVSF assumes a system model formulated as follows:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \text{ and} \tag{Eq. 4.1.3-1}$$

$$\mathbf{z}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} + \mathbf{v}_k. \tag{Eq. 4.1.3-2}$$

Here $\mathbf{x}_k \in \mathbb{R}^n$ is the system state, \mathbf{u}_k is the system input, \mathbf{w}_k is the system noise, $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement and \mathbf{v}_k is the measurement noise. Function f can be either linear or nonlinear. \mathbf{C} is the measurement matrix and is assumed linear, positive and pseudo diagonal. If the SVSF is applied to a system with a nonlinear measurement equation $\mathbf{z}_{k+1} = g(\mathbf{x}_{k+1}) + \mathbf{v}_k$, \mathbf{C} becomes the Jacobian matrix of the system, *i.e.*

$$\mathbf{C} = \mathbf{C}_{linearized} = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} & \dots & \frac{\partial g}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \dots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \dots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}. \quad (\text{Eq. 4.1.3-3})$$

At each iteration, the SVSF starts with the a-priori estimation of the state and the output, which is expressed as:

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \quad \text{and} \quad (\text{Eq. 4.1.3-4})$$

$$\hat{\mathbf{z}}_{k+1|k} = \mathbf{C}\hat{\mathbf{x}}_{k+1|k}. \quad (\text{Eq. 4.1.3-5})$$

The a-posteriori estimation is then updated with the estimation above and the SVSF corrective gain:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1} \quad \text{and} \quad (\text{Eq. 4.1.3-6})$$

$$\hat{\mathbf{z}}_{k+1|k+1} = \mathbf{C}\hat{\mathbf{x}}_{k+1|k+1}. \quad (\text{Eq. 4.1.3-7})$$

The SVSF gain with a fixed smoothing boundary layer and a saturation function is formulated as follows:

$$K_{k+1} = \mathbf{C}^+ (|\mathbf{e}_{z,k+1|k}| + \gamma |\mathbf{e}_{z,k|k}|) \circ \text{sat} \left(\frac{\mathbf{e}_{z,k+1|k}}{\psi} \right), \quad (\text{Eq. 4.1.3-8})$$

$$\text{sat}\left(\frac{a}{\psi}\right) = \begin{cases} \text{sgn}(a), & \text{if } |a| > \psi \\ a/\psi, & \text{if } |a| \leq \psi \end{cases} \quad (\text{Eq. 4.1.3-9})$$

where ψ denotes the smooth boundary layer width; γ is a diagonal matrix whose value is between 0 and 1; and $\mathbf{e}_{z,k+1|k}$ and $\mathbf{e}_{z,k|k}$ are the a-priori and a-posteriori state estimation errors respectively with the following definition:

$$\mathbf{e}_{z,k+1|k} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k} \quad (\text{Eq. 4.1.3-10})$$

$$\mathbf{e}_{z,k|k} = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k}. \quad (\text{Eq. 4.1.3-11})$$

In (4.1.3-8), \mathbf{C}^+ is the pseudo inverse of \mathbf{C} . The pseudo inverse is used to avoid the instability caused by the possible singularity of the Jacobian matrix $\mathbf{C}_{linearized}$ when the system is nonlinear.

$\mathbf{C}_{linearized}^+$ is expressed in the following way:

$$\mathbf{C}_{linearized}^+ = \mathbf{C}_{linearized}^T (\mathbf{C}_{linearized} \mathbf{C}_{linearized}^T + \rho^2 \mathbf{I})^{-1}, \quad (\text{Eq. 4.1.3-12})$$

where ρ is called the damping parameter and it can mitigate the effect of small singular values in the computation of the inverse. When the system is linear and has a full set of measurements corresponding to the full state vector, $\mathbf{C}^+ = \mathbf{C}^{-1}$.

4.2 Multi-layer Perceptron Network Training Algorithm Using Smooth Variable

Structure Filter

As mentioned before, the essence of training an ANN is to determine the value of its parameters.

Ahmed (2011) demonstrated that the SVSF can be applied to a MLP network to effectively train its link weights. His work will be summarized in this section since it forms the foundation for the RBF network training algorithm presented in the next section.

Suppose there are N pairs of inputs and outputs samples $\{\mathbf{x}_i, \mathbf{y}_i\}$ for training, the output vector \mathbf{y} for all the outputs is denoted as

$$\mathbf{y} = [y_{1,1} \quad y_{1,2} \quad \cdots \quad y_{1,m} \quad \cdots \quad y_{N,1} \quad y_{N,2} \quad \cdots \quad y_{N,m}], \quad (\text{Eq. 4.2-1})$$

where m is the number of neurons in the MLP's output layer, i.e. the output dimension. Then the MLP network can be represented by the following nonlinear discrete-time state space functions:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\omega}_k, \text{ and} \quad (\text{Eq. 4.2-2})$$

$$\mathbf{y}_k = \mathbf{C}_k(\mathbf{w}_k, \mathbf{x}_k) + \mathbf{v}_k, \quad (\text{Eq. 4.2-3})$$

where \mathbf{w}_k is a vector consisting of all the link weights including biases, \mathbf{C}_k represents the measurement function, and $\boldsymbol{\omega}_k$ and \mathbf{v}_k are zero-mean white Gaussian noises.

In this research, the MLP network has only one hidden layer. Assuming the dimension of the input layer is l , that of the hidden layer is n and that of the output layer is m , then the relationship between its input and output is:

$$y_h = \varphi\left[\sum_{j=0}^n \varphi\left(\sum_{i=0}^l u_i w_{I,i,j}\right) w_{O,j,h}\right], h = 1, 2, \dots, m. \quad (\text{Eq. 4.2-4})$$

where $w_{I,i,j}$ stands for the link weight between the i -th input node and the j -th neuron in the hidden layer; and $w_{O,j,h}$ stands for the link weight between the j -th neuron in the hidden layer and the h -th output node.

The linearization of the MLP model is performed by differentiating the network transfer function with respect to the link weights. The Jacobian matrix $\mathbf{C}_{k|linearized}$ can be mathematically expressed as follows:

$$\mathbf{C}_{k|linearized} = \begin{bmatrix} \frac{\partial y}{\partial w_1} & \frac{\partial y}{\partial w_2} & \cdots & \frac{\partial y}{\partial w_{N_T}} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_1}{\partial w_2} & \cdots & \frac{\partial y_1}{\partial w_{N_T}} \\ \frac{\partial y_2}{\partial w_1} & \frac{\partial y_2}{\partial w_2} & \cdots & \frac{\partial y_2}{\partial w_{N_T}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_m}{\partial w_1} & \frac{\partial y_m}{\partial w_2} & \cdots & \frac{\partial y_m}{\partial w_{N_T}} \end{bmatrix}, \quad (\text{Eq. 4.2-5})$$

where N_T denotes the total number of link weights including biases and can be calculated from the following equation:

$$N_T = (l + 1) \cdot n + (n + 1) \cdot m. \quad (\text{Eq. 4.2-6})$$

The SVSF-based MLP training algorithm is as follows:

Step 1: Network parameter initialization

A priori weight estimates $\hat{\mathbf{w}}_{k|k}$ are randomly initialized ranging from -1 to +1.

Step 2: Calculation of the predicted (a posteriori) weight estimates $\hat{\mathbf{w}}_{k+1|k}$

$$\hat{\mathbf{w}}_{k+1|k} = \hat{\mathbf{w}}_{k|k}. \quad (\text{Eq. 4.2-7})$$

Step 3: Calculation of Jacobian matrix of the measurement function $\mathbf{C}_{k|linearized}$.

Step 4: Calculation of the estimated network output (measurements) $\hat{\mathbf{y}}_{k+1|k}$

$$\hat{\mathbf{y}}_{k+1|k} = \mathbf{C}_{k|linearized} \hat{\mathbf{w}}_{k+1|k}. \quad (\text{Eq.4.2-8})$$

Step 5: Measurement error $\mathbf{e}_{y_{k+1|k}}$ calculation

$$\mathbf{e}_{y,k+1|k} = \mathbf{y}_k - \hat{\mathbf{y}}_{k+1|k}. \quad (\text{Eq.4.2-9})$$

Step 6: SVSF gain calculation

$$\mathbf{K}_{k+1} = \mathbf{C}_{k|linearized}^+ (|\mathbf{e}_{y,k+1|k}| + \gamma |\mathbf{e}_{y,k|k}|) \circ \text{sat} \left(\frac{\mathbf{e}_{y_{k+1|k}}}{\psi} \right). \quad (\text{Eq.4.2-10})$$

Step 7: Calculation of the updated state estimates $\hat{\mathbf{w}}_{k+1|k+1}$

$$\hat{\mathbf{w}}_{k+1|k+1} = \hat{\mathbf{w}}_{k+1|k} + \mathbf{K}_{k+1}. \quad (\text{Eq.4.2-11})$$

Step 8: Calculation of the updated state estimates and the updated weights

$$\hat{\mathbf{y}}_{k+1|k+1} = \mathbf{C}_{k|\text{linearized}} \hat{\mathbf{w}}_{k+1|k+1} \text{ and} \quad (\text{Eq.4.2-12})$$

$$\mathbf{e}_{y,k+1|k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1|k+1}. \quad (\text{Eq.4.2-13})$$

Step 9: Calculation of the mean square error (MSE). If it is larger than the desired value, then shuffle the training data and go to **step 3**; otherwise, the training procedure stops.

4.3 Radial Basis Function Network Training Algorithm Using Smooth Variable

Structure Filter

The RBF training problem is also regarded as a parameter estimation problem here. Suppose there are M responses of $\{\mathbf{x}_i, \mathbf{y}_i\}$, the (Eq. 4.1.2-4) can be augmented as follows:

$$\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1 \hat{\mathbf{y}}_2 \cdots \hat{\mathbf{y}}_M] = \begin{bmatrix} \hat{y}_{11} & \cdots & \hat{y}_{n1} \\ \hat{y}_{21} & \cdots & \hat{y}_{n2} \\ \vdots & \vdots & \vdots \\ \hat{y}_{n1} & \cdots & \hat{y}_{nM} \end{bmatrix}$$

$$= \begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1c} \\ w_{20} & w_{21} & \cdots & w_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n0} & w_{n1} & \cdots & w_{nc} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ g(\|\mathbf{x}_1 - \mathbf{v}_1\|) & \cdots & g(\|\mathbf{x}_M - \mathbf{v}_1\|) \\ \vdots & \vdots & \vdots \\ g(\|\mathbf{x}_1 - \mathbf{v}_c\|) & \cdots & g(\|\mathbf{x}_M - \mathbf{v}_c\|) \end{bmatrix}$$

(Eq. 4.3-1)

For simplicity the following notations are introduced:

$$\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1c} \\ w_{20} & w_{21} & \cdots & w_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n0} & w_{n1} & \cdots & w_{nc} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_n^T \end{bmatrix} = \mathbf{W}, \quad (\text{Eq. 4.3-2})$$

$$h_{0k} = 1 \quad (k = 1, \dots, M), \quad (\text{Eq. 4.3-3})$$

$$h_{jk} = g(\|\mathbf{x}_k - \mathbf{v}_j\|), \quad k = 1, \dots, M, \quad j = 1, \dots, c \text{ and} \quad (\text{Eq. 4.3-4})$$

$$\begin{bmatrix} h_{01} & \cdots & h_{0M} \\ h_{11} & \cdots & h_{1M} \\ \vdots & \vdots & \vdots \\ h_{c1} & \cdots & h_{cM} \end{bmatrix} = [h_1 \quad \cdots \quad h_M] = \mathbf{H}. \quad (\text{Eq. 4.3-5})$$

Then Eq. 4.3-1 can be rewritten as

$$\hat{\mathbf{Y}} = \mathbf{W}\mathbf{H} \quad (\text{Eq. 4.3-6})$$

The system model of the RBF network is established as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\omega}_k \text{ and} \quad (\text{Eq. 4.3-7})$$

$$\mathbf{y}_k = h(\boldsymbol{\theta}_k) + \mathbf{v}_k \quad (\text{Eq. 4.3-8})$$

where

$$\boldsymbol{\theta} = [\mathbf{w}_1^T \quad \cdots \quad \mathbf{w}_n^T \quad \mathbf{v}_1^T \quad \cdots \quad \mathbf{v}_c^T]^T, \quad (\text{Eq. 4.3-9})$$

$$\mathbf{y} = [y_{11} \quad \cdots \quad y_{1M} \quad \cdots \quad y_{n1} \quad \cdots \quad y_{nM}]^T \text{ and} \quad (\text{Eq. 4.3-10})$$

$$\hat{\mathbf{y}} = h(\boldsymbol{\theta}_k) = [\hat{y}_{11} \quad \cdots \quad \hat{y}_{1M} \quad \cdots \quad \hat{y}_{n1} \quad \cdots \quad \hat{y}_{nM}]_k^T \quad (\text{Eq. 4.3-11})$$

The vector $\boldsymbol{\theta}$ consists of all $(n(c + 1) + mc)$ RBF parameters; \mathbf{y} and $\hat{\mathbf{y}}$ are the target and actual outputs and both have nM elements; $h(\cdot)$ is the nonlinear mapping between the RBF's parameters and outputs; and $\boldsymbol{\omega}_k$ and \mathbf{v}_k are the system and measurement noise.

The partial derivative of the system with respect to the RBF parameters at the k th iteration of the SVSF recursion is

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_w \\ \mathbf{H}_v \end{bmatrix} \quad (\text{Eq. 4.3-12})$$

where

$$\mathbf{H}_w = \begin{bmatrix} H & O & \cdots & O \\ O & H & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & \cdots & O & H \end{bmatrix} \quad (\text{Eq. 4.3-13})$$

\mathbf{H}_v

$$= \begin{bmatrix} -w_{11}g'_{11}2(x_1 - v_1) & \cdots & -w_{11}g'_{m1}2(x_m - v_1) & \cdots & -w_{n1}g'_{11}2(x_1 - v_1) & \cdots & -w_{n1}g'_{m1}2(x_m - v_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -w_{1c}g'_{1c}2(x_1 - v_c) & \cdots & -w_{1c}g'_{mc}2(x_m - v_c) & \cdots & -w_{n1}g'_{1c}2(x_1 - v_c) & \cdots & -w_{nc}g'_{mc}2(x_m - v_c) \end{bmatrix} \quad (\text{Eq. 4.3-14})$$

\mathbf{H}_w is an $n(c + 1) \times nM$ matrix and \mathbf{H}_v is an $mc \times nM$ matrix.

The complete SVSF-based RBF network training algorithm is as follows:

Step 1: Network parameter initialization

The weights between the hidden layer and output layer $[\mathbf{w}_1^T \ \cdots \ \mathbf{w}_n^T]$ are set to $\mathbf{0}$. The prototype vectors in the hidden layer $[\mathbf{v}_1^T \ \cdots \ \mathbf{v}_c^T]$ are randomly chosen from the input samples.

Step 2: Calculation of the predicted weight estimates

$$\hat{\boldsymbol{\theta}}_{k+1|k} = \hat{\boldsymbol{\theta}}_{k|k}. \quad (\text{Eq. 4.3-15})$$

Step 3: Calculation of Jacobian matrix of the measurement function

$$\mathbf{C}_{k|linearized} = \mathbf{H}_k^T. \quad (\text{Eq. 4.3-16})$$

Step 4: Calculation of the estimated network output

$$\hat{\mathbf{y}}_{k+1|k} = h(\hat{\boldsymbol{\theta}}_{k+1|k}). \quad (\text{Eq. 4.3-17})$$

Step 5: Measurement error calculation

$$\mathbf{e}_{y_{k+1|k}} = \mathbf{y}_k - \hat{\mathbf{y}}_{k+1|k}. \quad (\text{Eq. 4.3-18})$$

Step 6: SVSF gain calculation

$$\mathbf{K}_{k+1} = \mathbf{C}_{k|linearized} + \left(\left| \mathbf{e}_{y_{k+1|k}} \right| + \gamma \left| \mathbf{e}_{y_{k|k}} \right| \right) \circ \text{sat} \left(\frac{e^{y_{k+1|k}}}{\psi} \right). \quad (\text{Eq. 4.3-19})$$

Step 7: Calculation of the updated state estimates

$$\hat{\boldsymbol{\theta}}_{k+1|k+1} = \hat{\boldsymbol{\theta}}_{k+1|k} + \mathbf{K}_{k+1}. \quad (\text{Eq. 4.3-20})$$

Step 8: Calculation of the updated state estimates and the updated weights

$$\hat{\mathbf{y}}_{k+1|k+1} = h(\hat{\boldsymbol{\theta}}_{k+1|k+1}) \text{ and} \quad (\text{Eq. 4.3-21})$$

$$\mathbf{e}_{y_{k+1|k+1}} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1|k+1}. \quad (\text{Eq. 4.3-22})$$

Step 9: Calculation of error defined by (Eq. 4.3-23). If it is larger than the desired value, then repeat step 2 to 8; otherwise the training procedure stops. (Note that \mathbf{y}_k is actually constant although it is written as a function of the SVSF iteration number k .)

$$\mathbf{E} = \frac{1}{2} \|\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}\|^2. \quad (\text{Eq. 4.3-23})$$

The flow chart for this algorithm is exhibited as Fig. 4.1.8. In real implement, the parameter c , which is the number of the prototype vectors in the hidden layer should be chosen such that a proper balance between the computational cost and the classification accuracy is achieved.

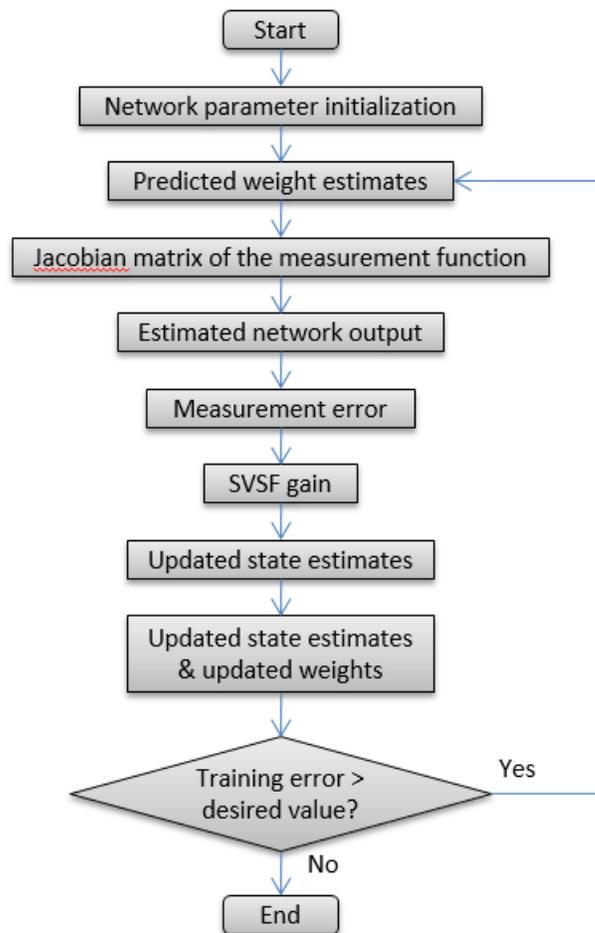


Figure 4.1.8 Flow chart of RBF Network Training Algorithm Using SVSF.

4.4 Artificial Neural Network Input Selection

In addition to the ANN training algorithm, the network's input (feature) is also important for obtaining a high classification accuracy. With vibration data collected from the accelerometer, the ANN's input is constructed as follows.

Assume the vibration data of one ICE working cycle has M samples after it is mapped to the crank angle domain. The i -th cycle data is denoted as

$$\mathbf{v}_i = [v_{1i} \quad v_{2i} \quad \cdots \quad v_{Mi}]^T. \quad (\text{Eq. 4.4-1})$$

The first step in the feature construction is to perform the discrete wavelet transformation. The wavelet decomposition coefficient vector for \mathbf{v}_i is denoted as $\mathbf{C}_i = [C_{1i} \quad C_{2i} \quad \cdots \quad C_{Mi}]^T$, which is also of size $M \times 1$. The wavelet transform was selected since the wavelet decomposition coefficients are determined by both time factors and frequency factors. This should make the feature superior compared to features in purely the time or frequency domains.

The last step to obtain the ANN's input is to perform a running average on the wavelet decomposition coefficients. A running average serves the function of smoothing the data. The effect of outliers can be greatly reduced by this operation. Supposing n cycles (starting from the k -th cycle) of data are collected and decomposed by the discrete wavelet transform, the result of the running average operation is:

$$\overline{\mathbf{C}}_k = \sum_{i=k}^{k+n-1} \mathbf{C}_i. \quad (\text{Eq. 4.4-2})$$

4.5 Summary

This chapter reviewed an existing SVSF-based training algorithm for MLP networks, and proposed a new SVSF-based training algorithm for RBF networks. The two algorithms are expected to inherit the robust characteristics of SVSF and have good generalization ability. This chapter also discusses the ANN input selection problem, and proposes that the running-averaged wavelet coefficients of vibration data in CAD are used as the input for ICE FDD. The ANNs with the SVSF-based training

algorithm and the input mentioned above are expected to achieve high performance in solving ICE FDD problems.

Chapter 5 Experimental Results, Analysis and Discussion

This chapter begins with a description of the experimental setup for the ICE fault detection system. The method for inducing the ICE misfire faults is described next. Finally, experimental results obtained using the FDD techniques from the previous chapters are presented, analysed and discussed.

5.1 Experimental Setup

As shown in Fig. 5.1.1 the FDD experimental setup consists of one ICE with a built-in cam identification sensor, four externally mounted accelerometers, four current sources, four signal conditioners, and a data acquisition device (DAQ). The cam identification sensor is also termed the “CID sensor.”

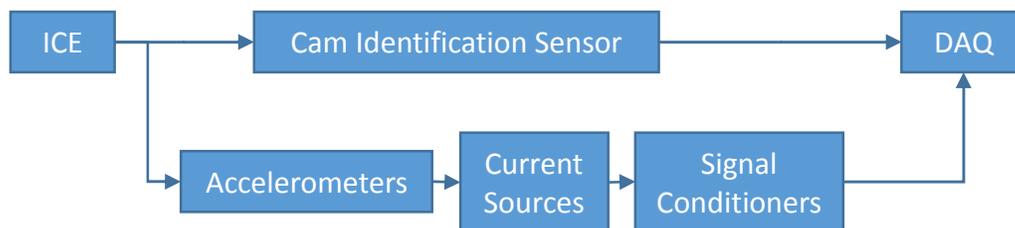


Figure 5.1.1 Experimental setup for ICE fault detection.

The experimental ICE is a Ford V8 4.6L engine. The experiments are performed on the vehicle shown in Fig. 5.1.2.



Figure 5.1.2 Experimental vehicle.

The accelerometers as shown in Fig. 5.1.3 are AC240-1D piezoelectric accelerometers from Connection Technology Center (CTC) Inc., whose sensitivity is 100 mV/g. They are the ones that Ford Motor Company use. Since they cannot be seen inside the engine compartment of the experimental vehicle, their locations and orientations are shown in Fig. 5.1.4, where the engine shown is a spare of the same model as that in the experimental car.



Figure 5.1.3 CTC AC240-1D accelerometer. (CTC, 2012)



(a) Positions and orientations of the accelerometers shown in the front view of the ICE. (The blue shapes stand for the sensors. Same with View 1 & 3.)



(b) View 1.



(c) View 2 (Blue circles stand for sensors. Same with View 4).



(d) View 3.



(e) View 4.

Figure 5.1.4 Positions and orientations of the accelerometers.

Each accelerometer requires a current source power supply. The current source takes the accelerometer as an input and outputs the vibration signal to the signal conditioners. The current source used is a PSP1001 Model from Inter Technology Inc. (ITI) and is shown in Fig. 5.1.5.

**Figure 5.1.5** PSP1001 current source. (ITI, 2008)



Figure 5.1.7 dSPACE MicroAutoBox 1401.

5.2 ICE Misfire Fault Induction

The experimental ICE has eight cylinders and each cylinder has a spark plug which is used to trigger the combustion. These plugs are numbered as shown in Fig. 5.2.1. The experiments will test the performance of the FDD techniques proposed in previous chapters on the ICE misfire fault. To induce a misfire fault, one can simply disconnect the spark plug connector of a certain cylinder as shown in Fig. 5.2.2.



Figure 5.2.1 Spark plug numbering.



Figure 5.2.2 Example of a spark plug connector.

In this research the experimental ICE has eight working conditions—one healthy condition and seven faulty conditions. Under each faulty condition, only one spark plug connector is removed with the others still working properly. The eight working conditions are listed in the Table 5.2.1.

Table 5.2.1: The eight ICE working conditions studied³.

Condition Index	Description
1	Cylinder 1 Misfire
2	Cylinder 2 Misfire
3	Cylinder 3 Misfire
4	Cylinder 4 Misfire
5	Cylinder 5 Misfire
6	Cylinder 6 Misfire

³ Cylinder 8 misfire was not studied because the corresponding spark plug connector is surrounded by many pipes and wires and is difficult to remove.

7	Cylinder 7 Misfire
8	Healthy Condition

5.3 Results and Discussion

5.3.1 ICE Vibration Data in the Crank Angle Domain

The original vibration data collected from the accelerometer and the data of the CID sensor are exhibited in Fig. 5.3.1. According to the peaks of the CID signal, the vibration data can be divided into several working cycles, as shown in Fig. 5.3.2. Vibration data in the CAD obtained by interpolation are shown in Fig. 5.3.3.

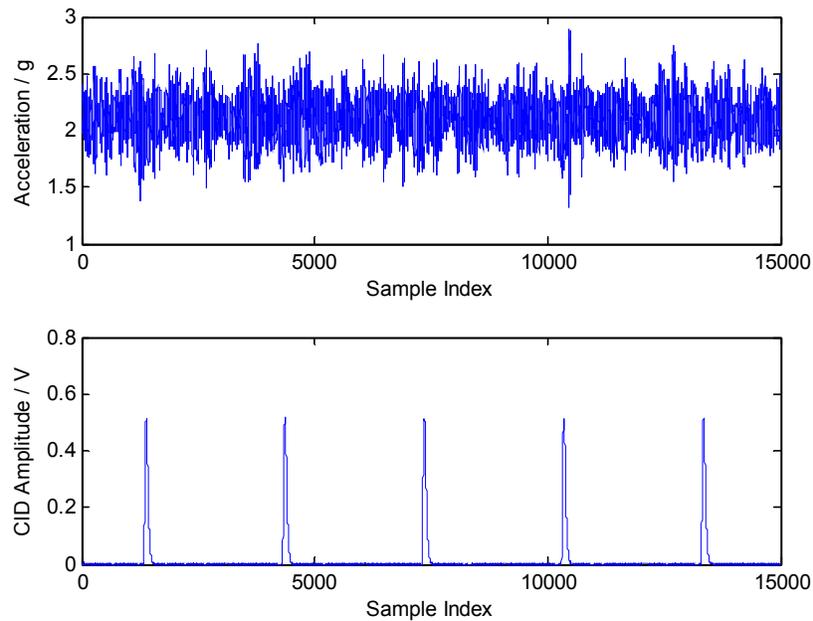


Figure 5.3.1 ICE Sensor Raw Data (One accelerometer and the CID sensor data).

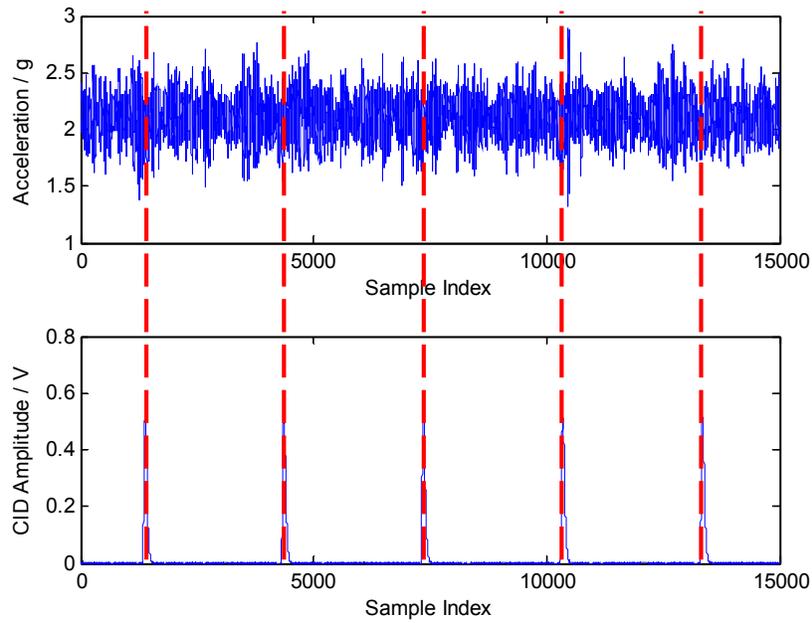


Figure 5.3.2 ICE Sensor Data Split in Cycles.

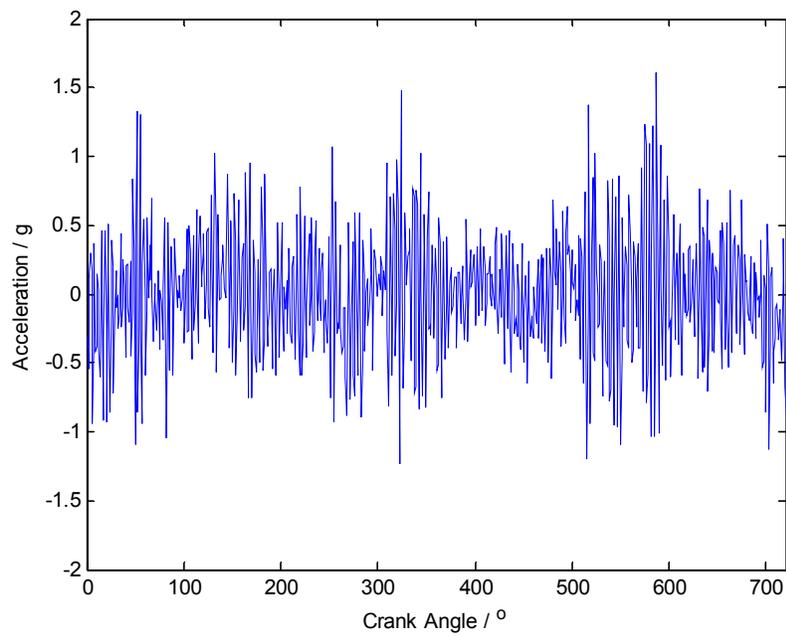


Figure 5.3.3 ICE Sensor Data after CAD Mapping (One Cycle Data).

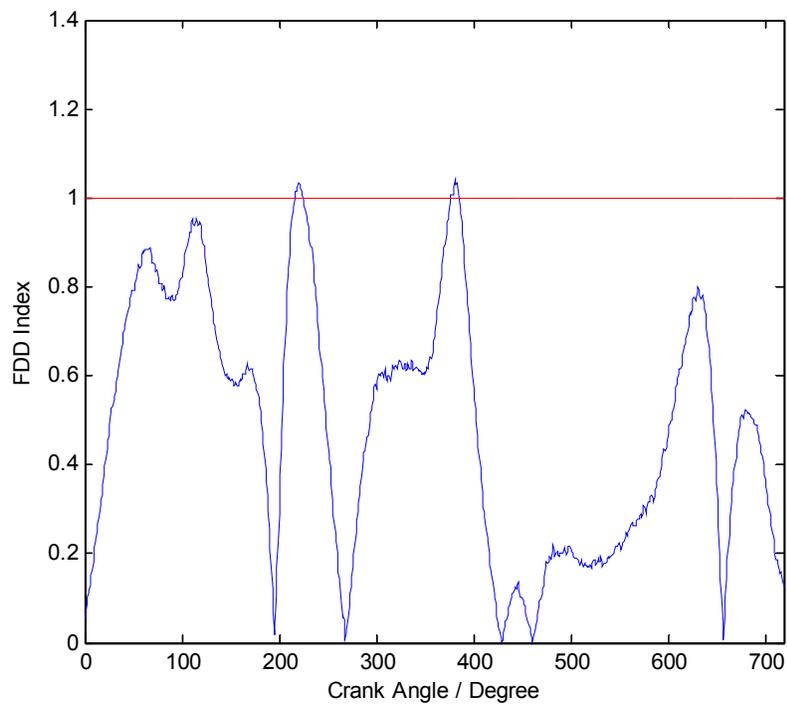
5.3.2 Experiment Results for the VMD-based FDD Algorithm

This experiments were done using the VMD parameters listed in Table 5.3.1 and the VMD software provided by (Dragomiretskiy, et al., 2014).

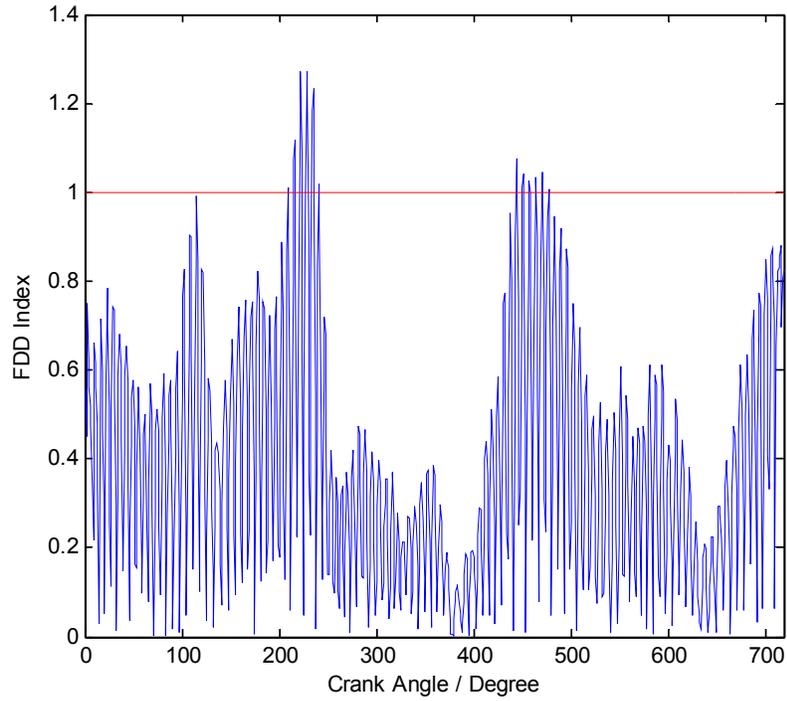
Table 5.3.1: VMD parameters in the experiment.

Parameter	Value
K	6
β	20000
τ	0
ϵ	10^{-7}

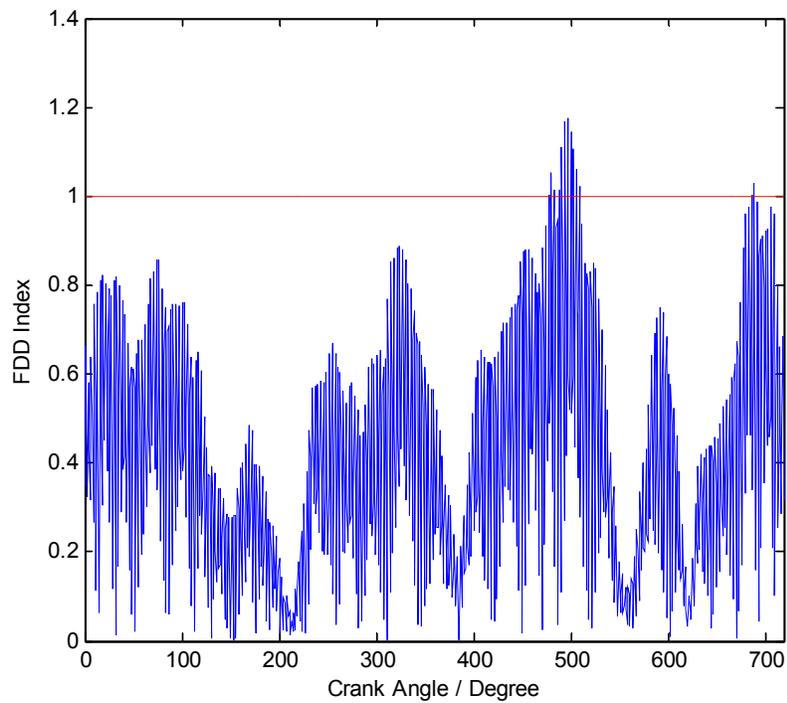
The experiment results for the normal condition are shown in Fig. 5.3.4.



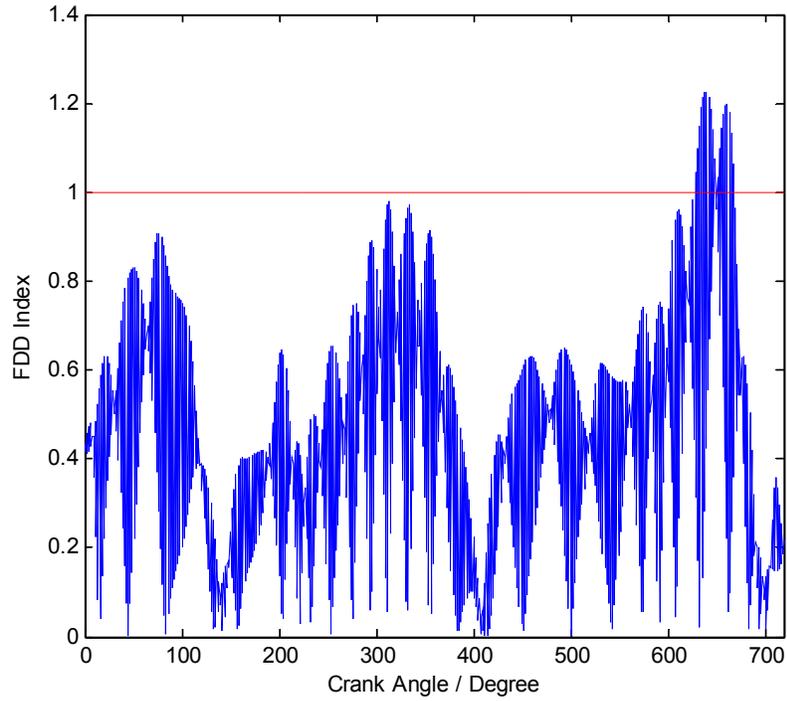
(a) FDD Index for Intrinsic Mode 1



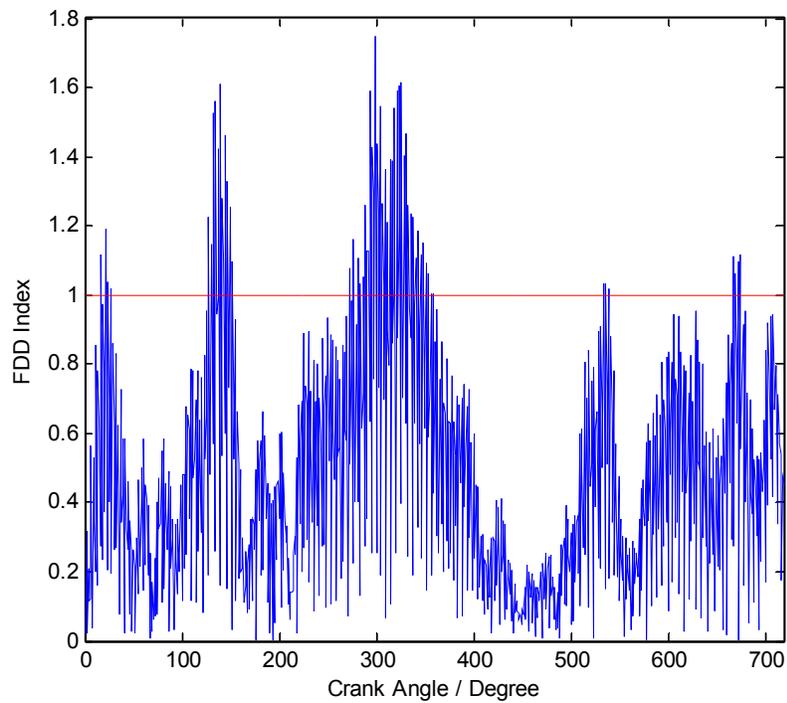
(b) FDD Index for Intrinsic Mode 2



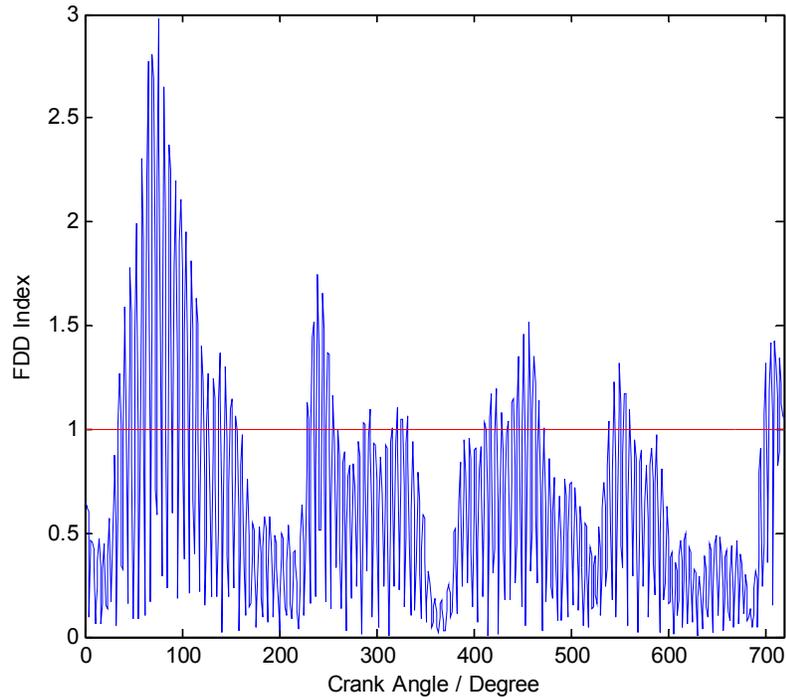
(c) FDD Index for Intrinsic Mode 3



(d) FDD Index for Intrinsic Mode 4



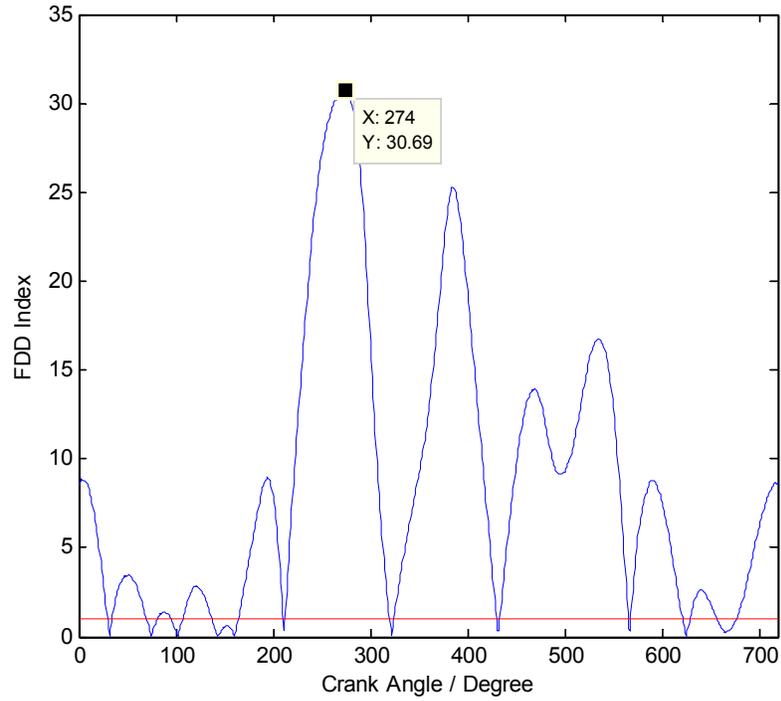
(e) FDD Index for Intrinsic Mode 5



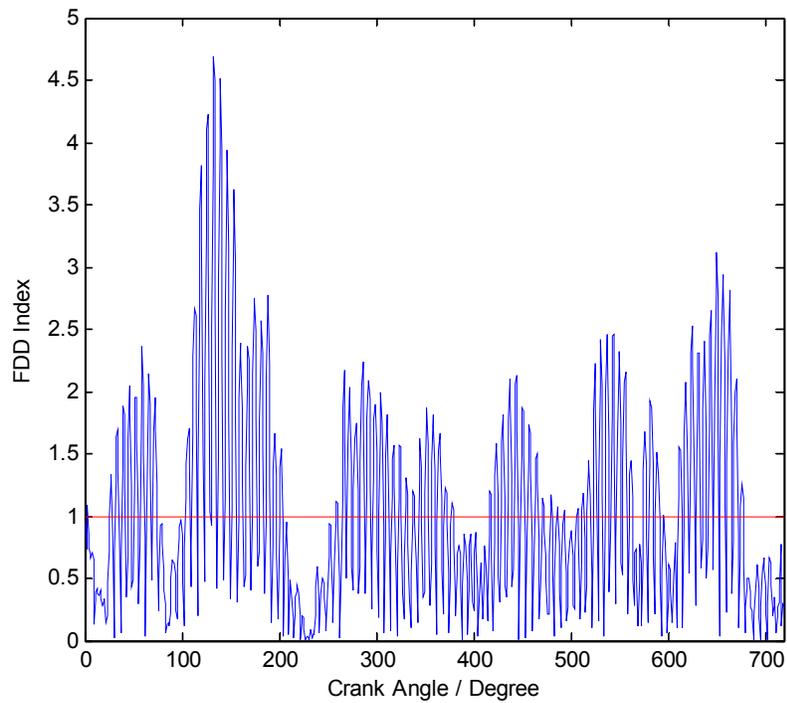
(f) FDD Index for Intrinsic Mode 6

Figure 5.3.4 VMD-based FDD indices for the misfire fault for the normal condition.

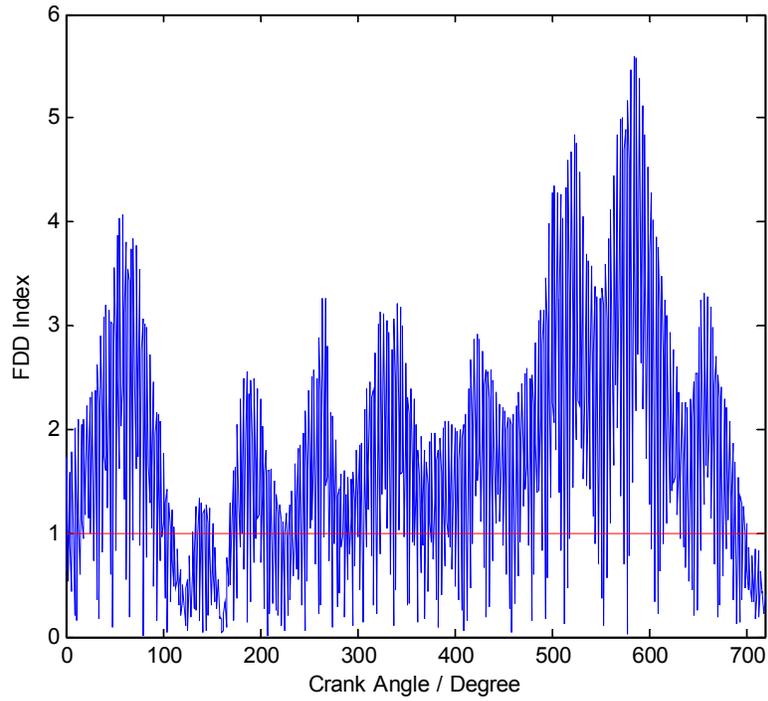
We can see that the FDD indices exceed 1 (represented by the red line in the plots) only at a few crank angles (*e.g.* 382° at mode 1). According to the theory presented in section 3.2, the FDD index should be smaller than 1 all the time. This variance can be explained by the fact that although $\hat{\mu}$ is the unbiased estimate of μ which is the Gaussian distribution mean, it is not the true value of the mean. Similarly, $\hat{\sigma}$ is only an estimate of the Gaussian distribution variance. Thus using these estimates to calculate the FDD index (Eq. 3.2-12) will lead to some incorrect values. The FDD index for the normal condition is not much greater than 1 (no more than 3 times in this experiment) compared with that for the misfire condition *e.g.* the FDD index for the misfire fault in Cylinder 2 as shown in Fig. 5.3.5.



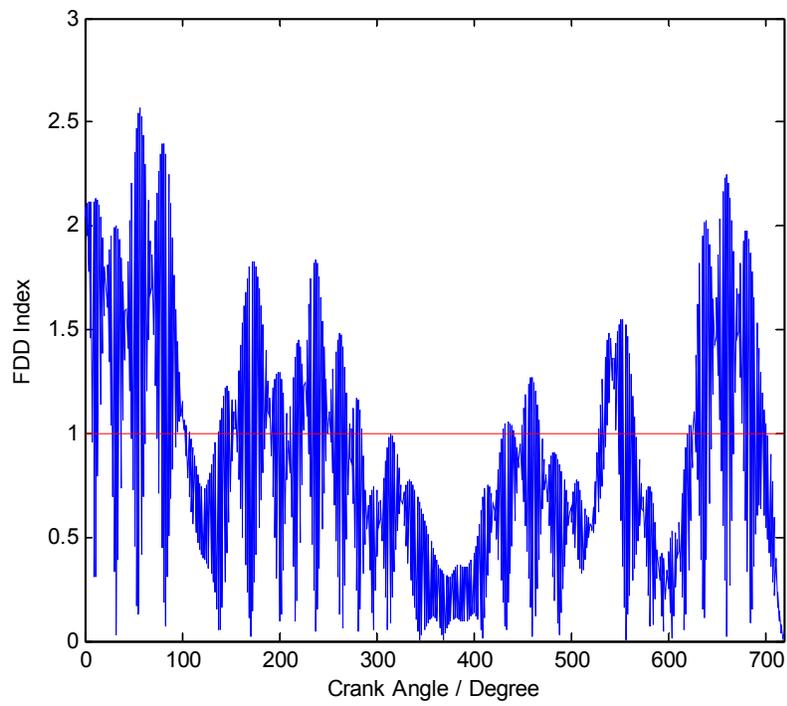
(a) FDD Index for Intrinsic Mode 1



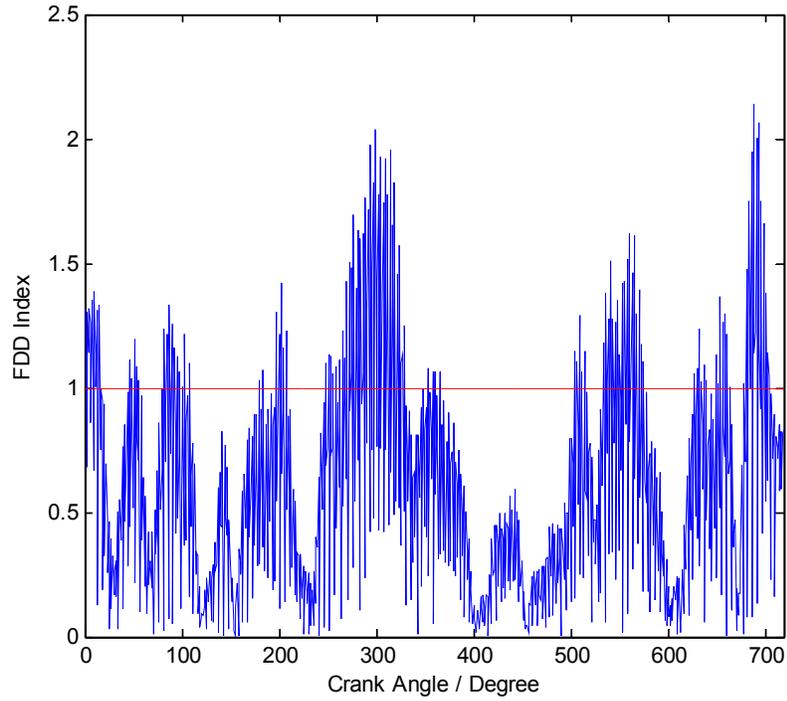
(b) FDD Index for Intrinsic Mode 2



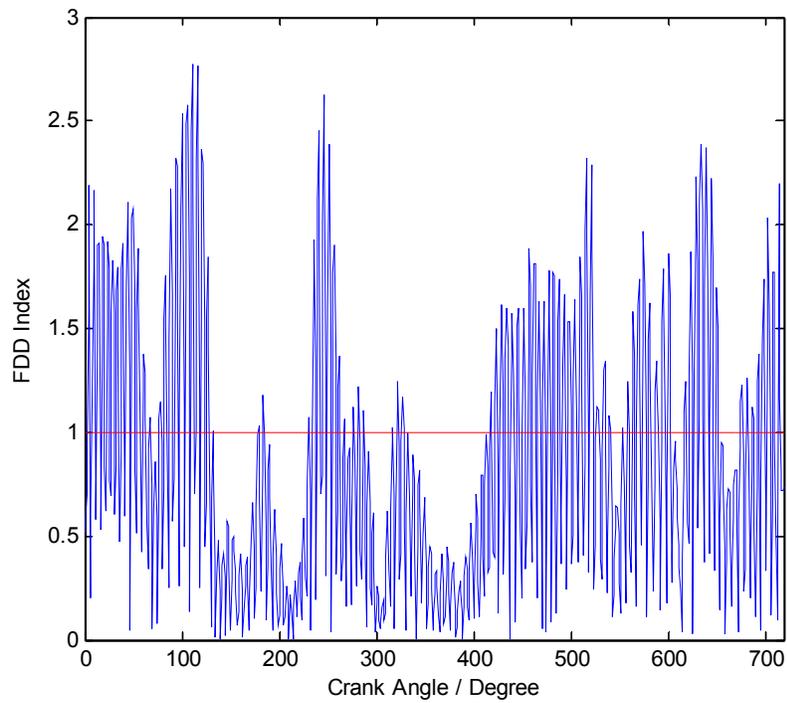
(c) FDD Index for Intrinsic Mode 3



(d) FDD Index for Intrinsic Mode 4



(e) FDD Index for Intrinsic Mode 5



(f) FDD Index for Intrinsic Mode 6

Figure 5.3.5 VMD-based FDD algorithm for the misfire fault in Cylinder 2.

We can find that for modes 1, 2 and 3, the ICE working conditions at most crank angles are significantly affected by the misfire fault since the corresponding indices are much greater 1. However, the index for mode 1 (with the lowest frequency range) is obviously much larger than those of the other modes. This implies that mode 1 is the most useful for misfire FDD. The maximum index whose value is 30.69 appears at 274° in the CAD, which indicates the corresponding mechanical event at this angle is affected by the misfire fault to the most extent.

Application of this FDD method to the misfire faults in the other cylinders reveals a similar finding, *i.e.* Mode 1 has the largest index among all the modes while the difference among the experiments is that the peak location for each fault is different. The FDD result can be summarized in table 5.3.2:

Table 5.3.2: Experiment results for VMD-based FDD algorithm.

Faulty Location	Max Index Mode	Max Index Angle
Cylinder 1	1	7°
Cylinder 2	1	274°
Cylinder 3	1	467°
Cylinder 4	1	541°
Cylinder 5	1	92°
Cylinder 6	1	365°
Cylinder 7	1	201°

To check the performance of this algorithm, 40 cycles⁴ of data were generated and tested for each of the eight conditions listed in section 5.2. The confusion matrix for the test is shown in Fig. 5.3.6.

We can see the accuracy of the algorithm was 100%.

⁴ 50 cycles of vibration data in CAD were obtained for each ICE condition first. But running averages with the window size of 10 were performed on these data, so 40 cycles of data were generated for testing the algorithm.

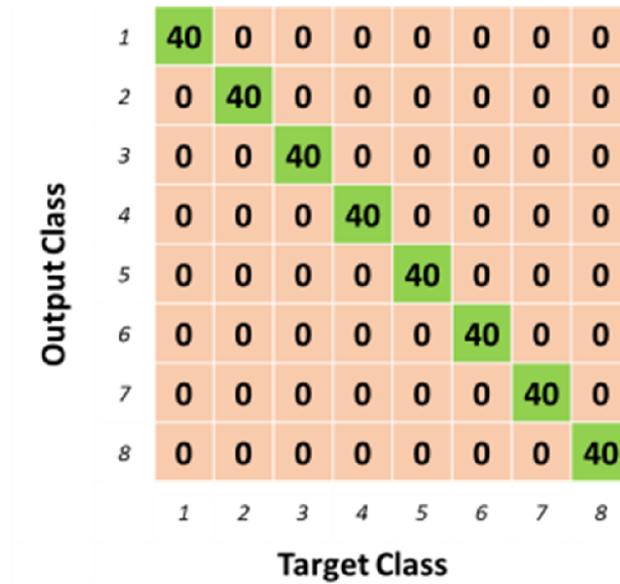


Figure 5.3.6 Confusion matrix for VMD-based FDD algorithm.

To sum up, this VMD-based FDD algorithm is capable of detecting a fault with the mode of the maximum index indicating the type of the fault (misfire in this experiment) and its angle indicating the fault's location.

5.3.3 Experimental Results for the Wavelet-Based KPCA FDD Algorithm

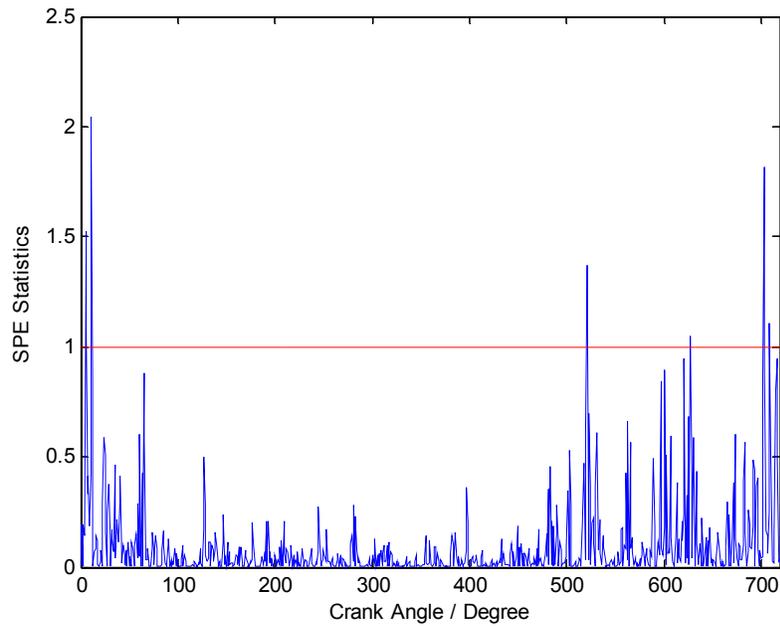
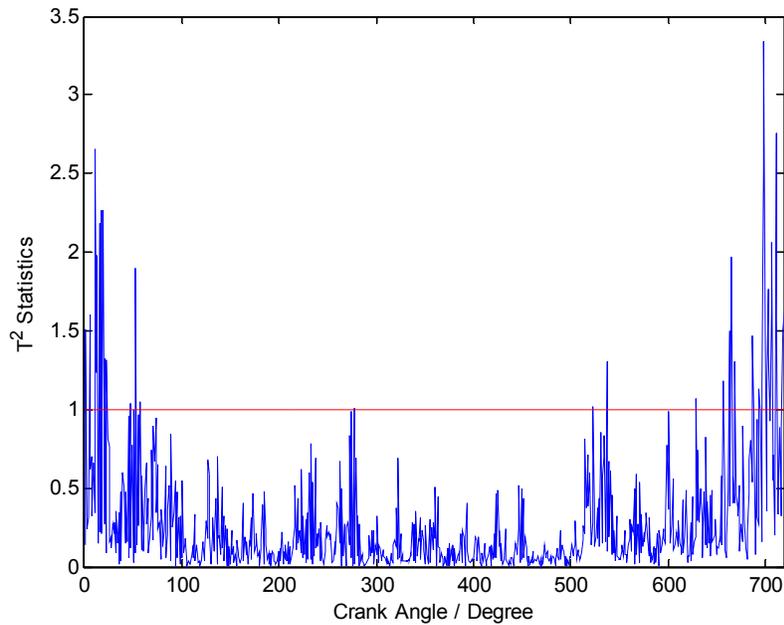
The wavelet function used in this experiment is Coiflets 5 and there are 6 levels (5 detail levels and 1 approximation level) for the wavelet decomposition. The wavelet based KPCA FDD result for the normal condition is shown in Fig. 5.3.7. The vertical axis shows the ratio of the KPCA to its control limit whose values are shown in table 5.3.3.

Table 5.3.3: Control limits for wavelet-based KPCA.

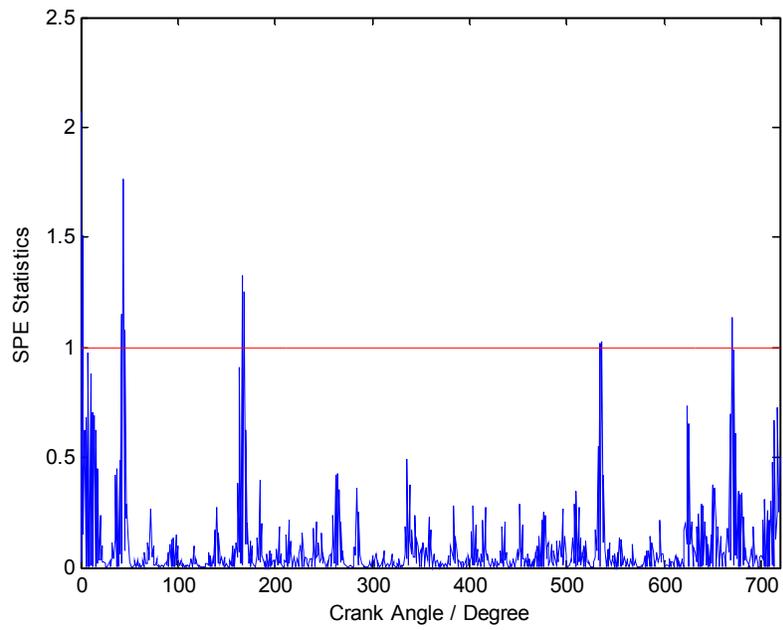
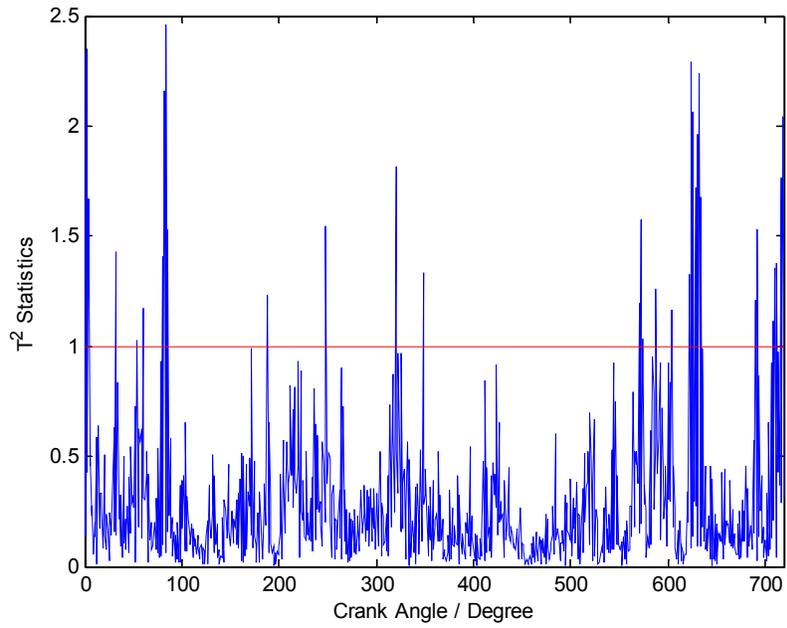
Level	Control limit for \mathcal{J}^2	Control limit for Q
1	11.4590 ⁵	6.9345
2	11.4590	2.8922

⁵ The number of retained PCs is manually set to 3 according to the eigenvalues obtained from (Eq. 3.1.5-2). As a result, the control limit for \mathcal{J}^2 becomes a fixed number according to (Eq. 3.1.4-8).

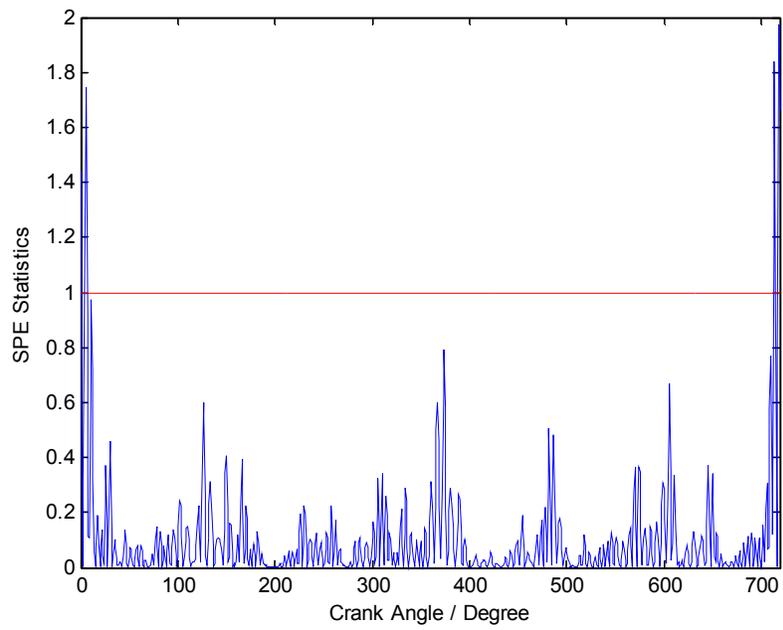
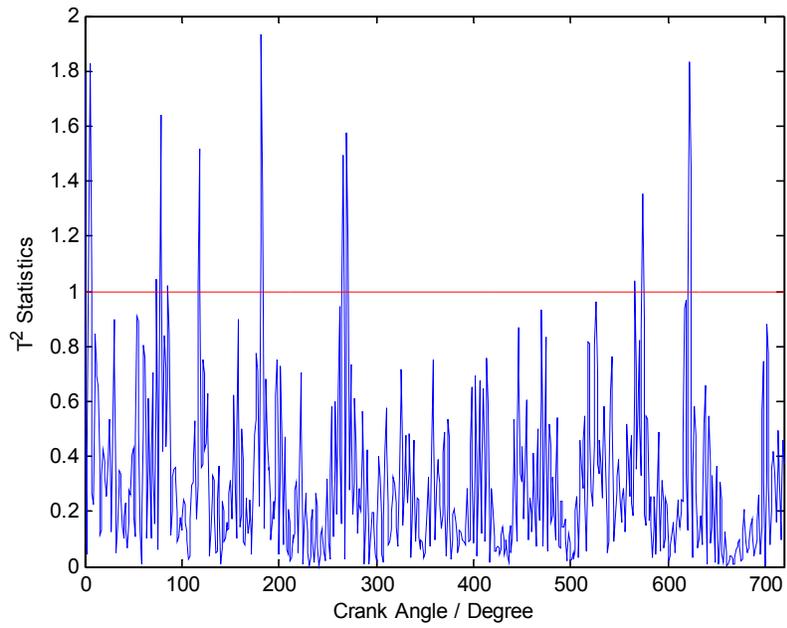
3	11.4590	1.8935
4	11.4590	0.9914
5	11.4590	2.89×10^{-2}
Approximation	11.4590	2.01×10^{-4}



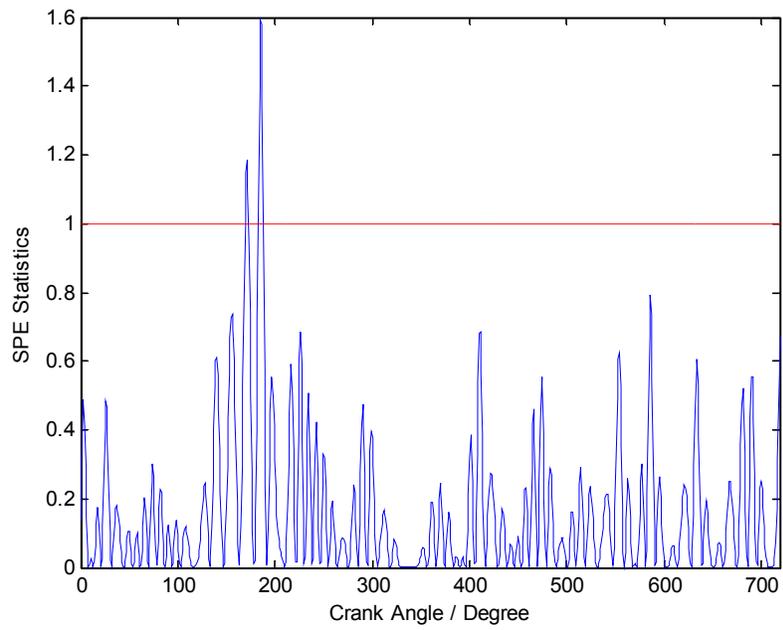
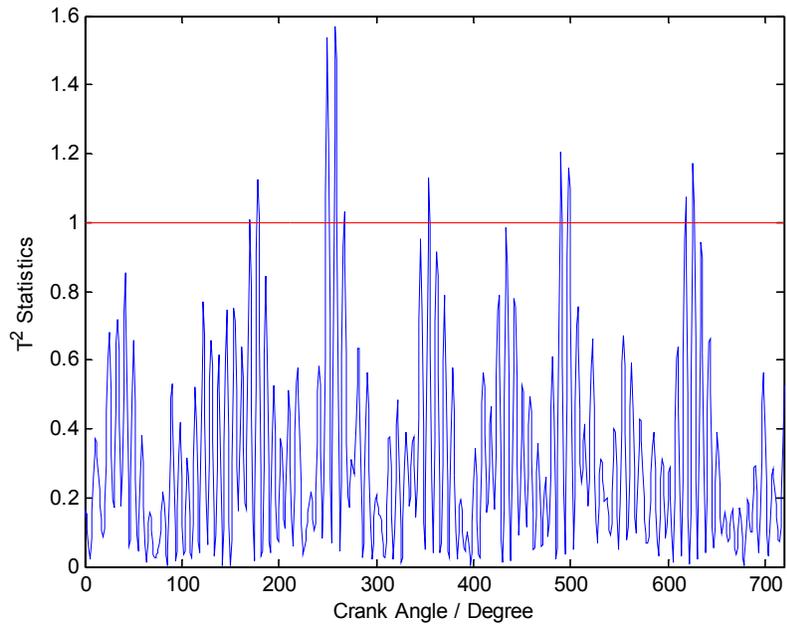
(a) KPCA for Level 1



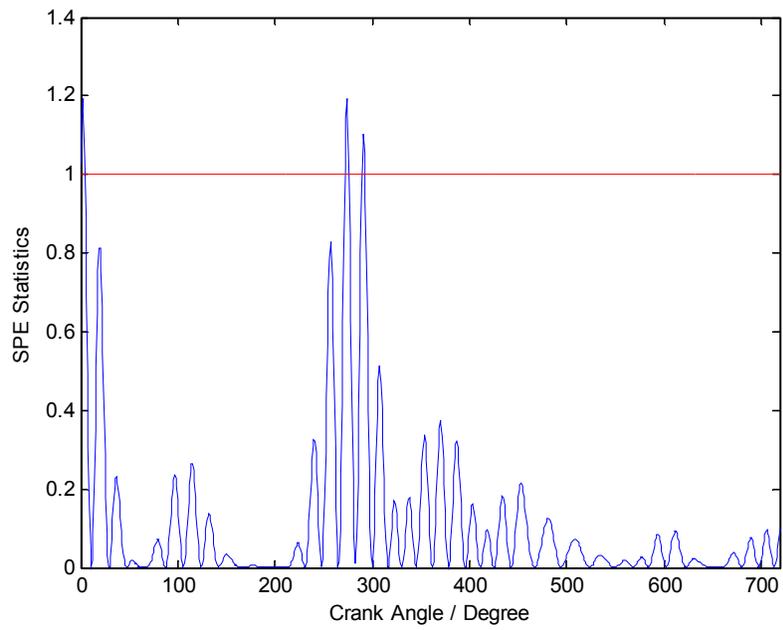
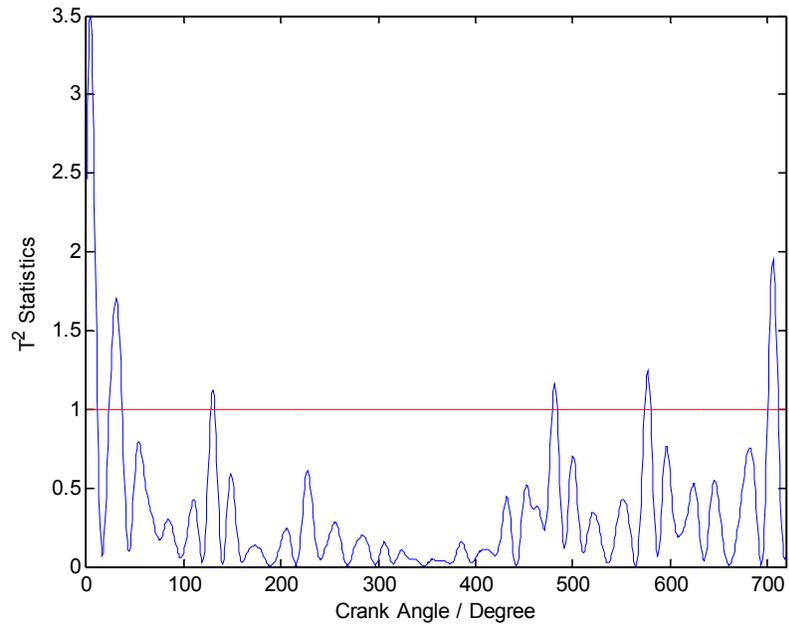
(b) KPCA for Level 2



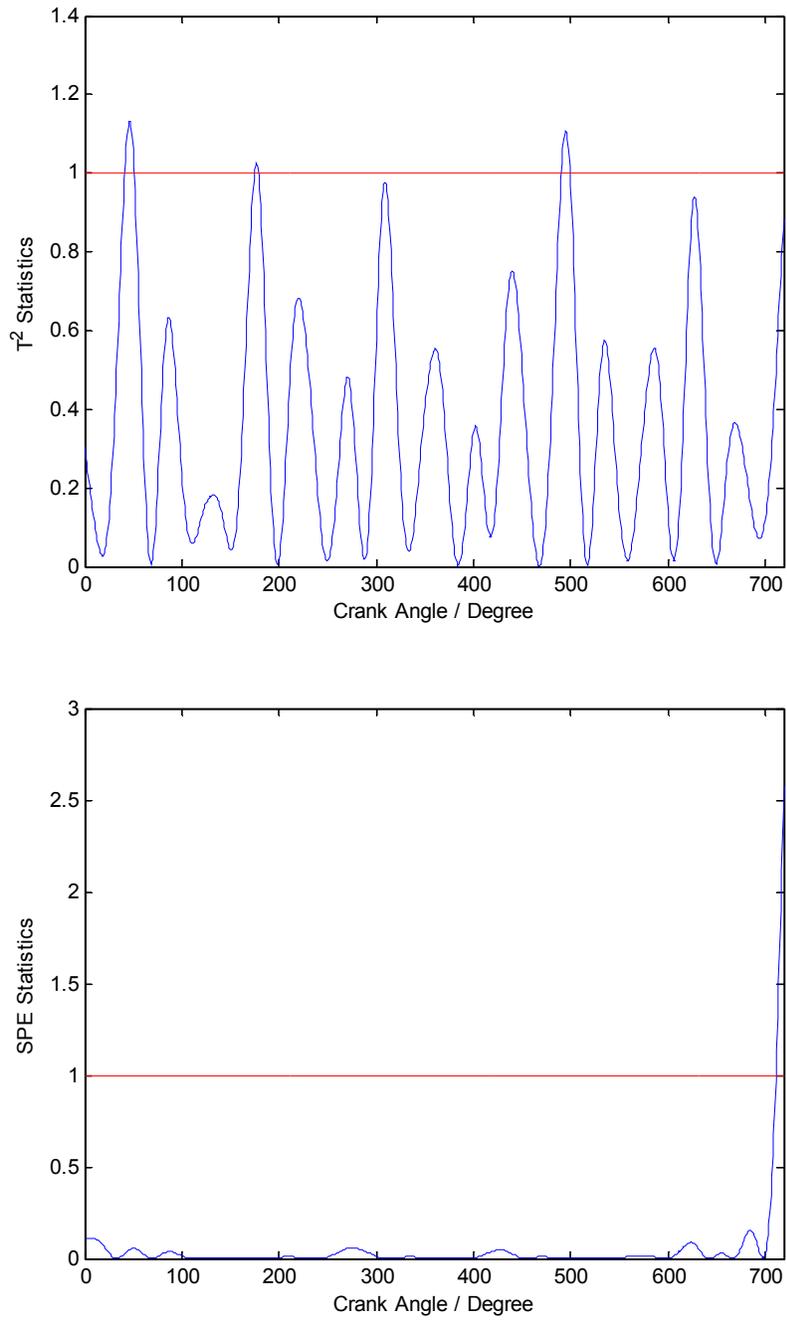
(c) KPCA for Level 3



(d) KPCA for Level 4



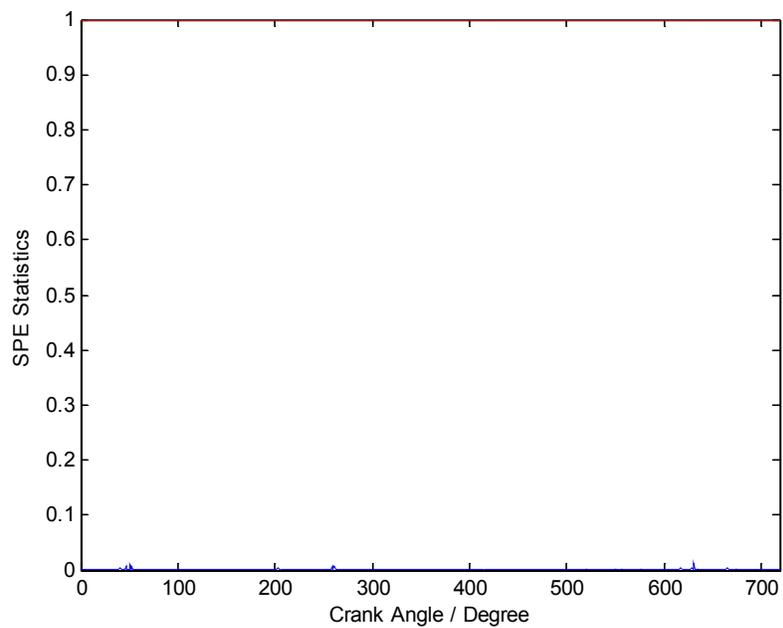
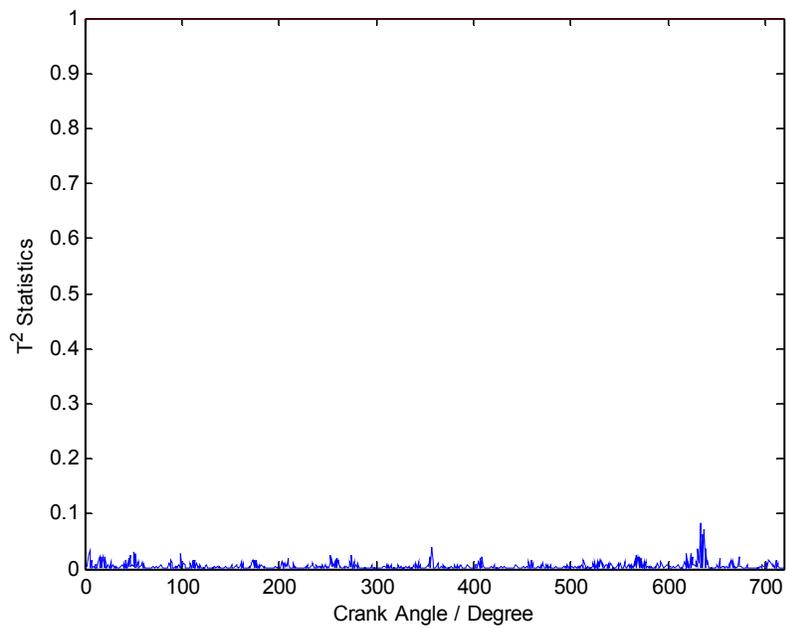
(5) KPCA for Level 5



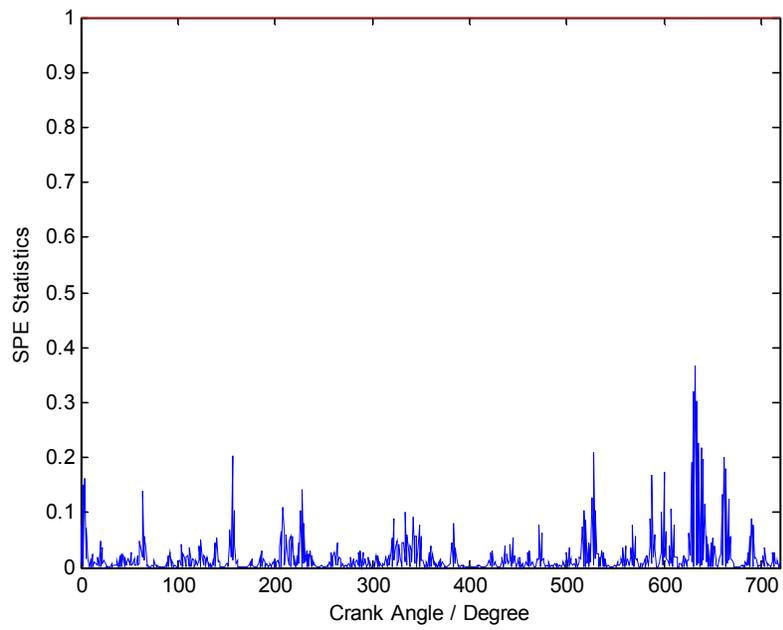
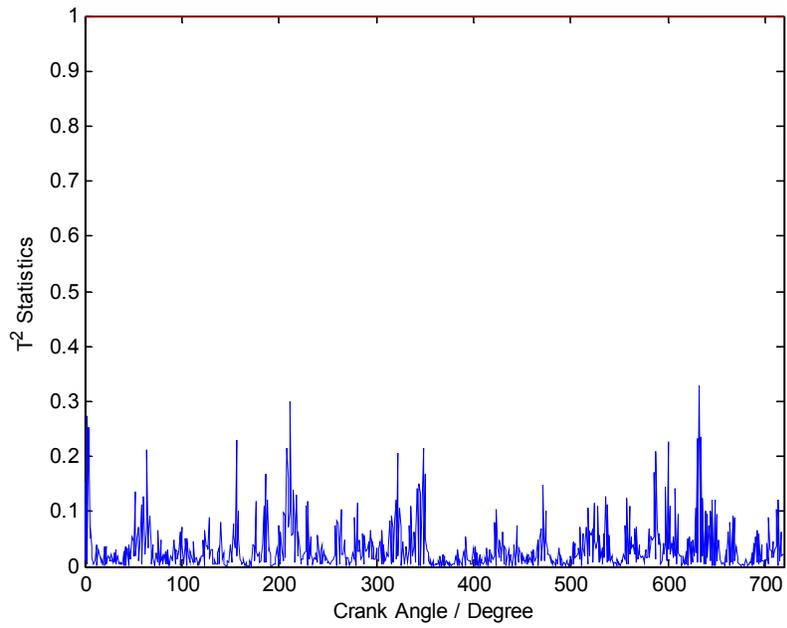
(6) KPCA for the Approximation Level

Figure 5.3.7 Wavelet-based KPCA FDD result for the normal condition.

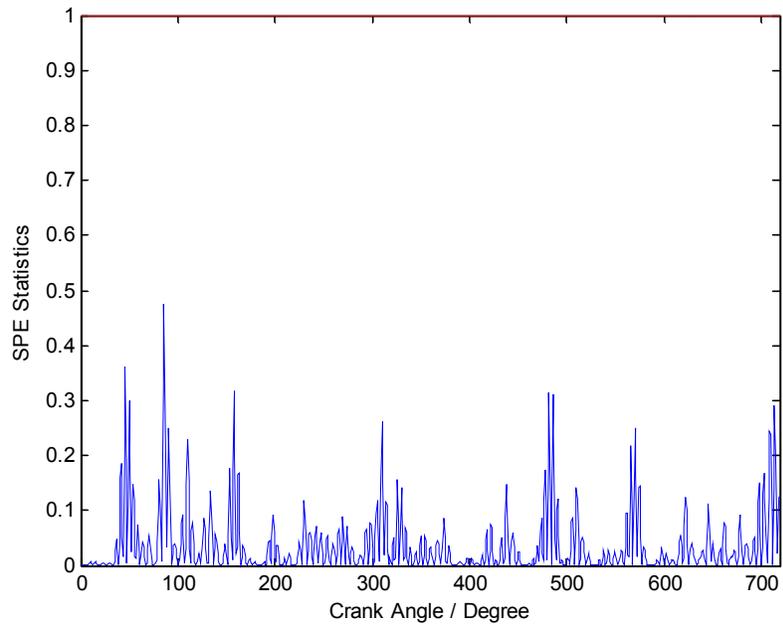
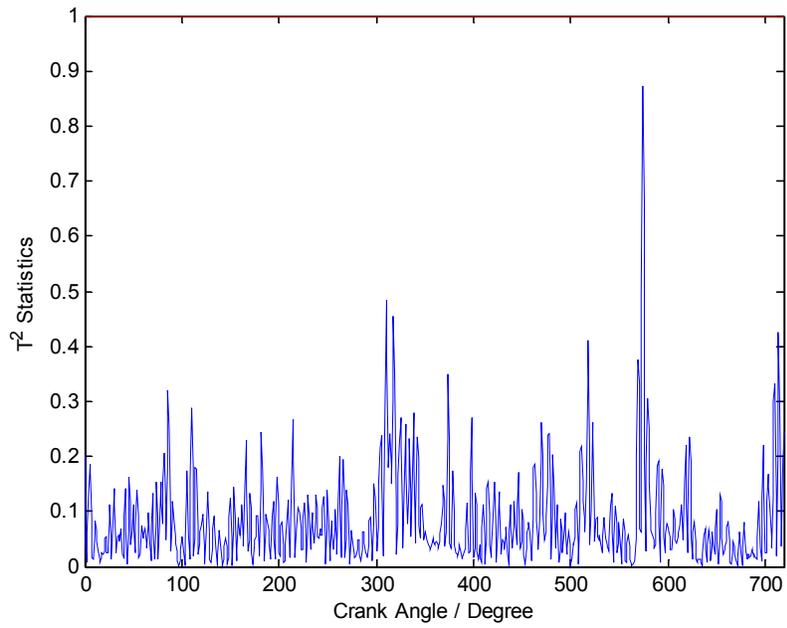
The wavelet-based KPCA FDD result for misfire in Cylinder 2 is shown in Fig. 5.3.8. The vertical axis as Fig. 5.3.7 shows the ratio of the KPCA to its limit and thus if the value is over 1, a fault may have happened at a certain crank angle and at a certain level.



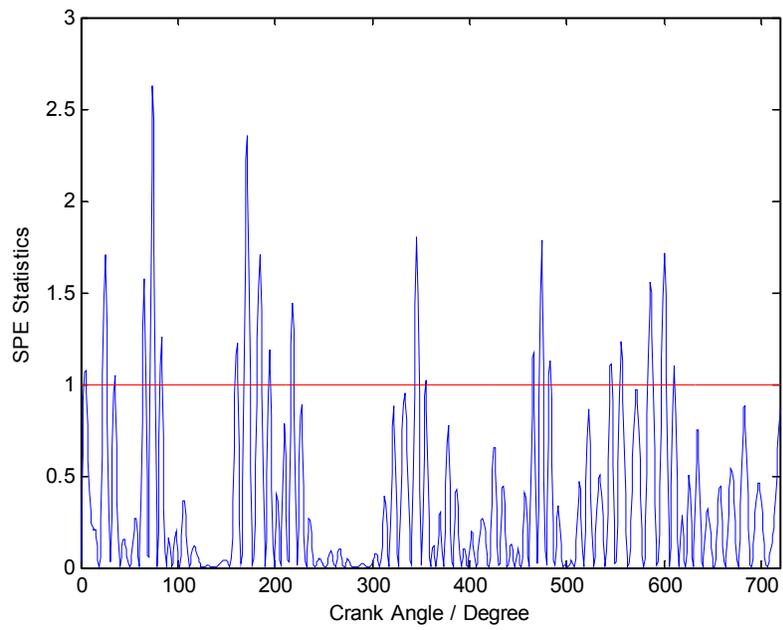
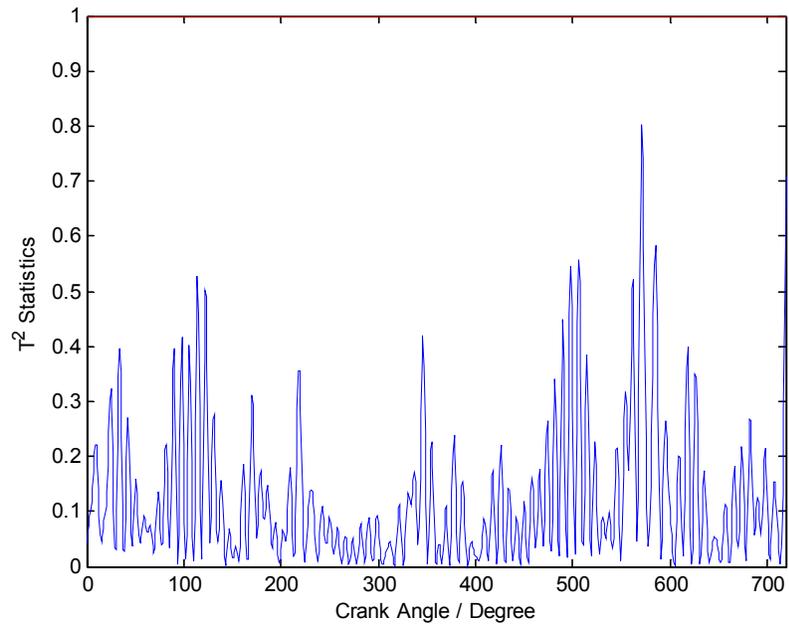
(a) KPCA for Level 1



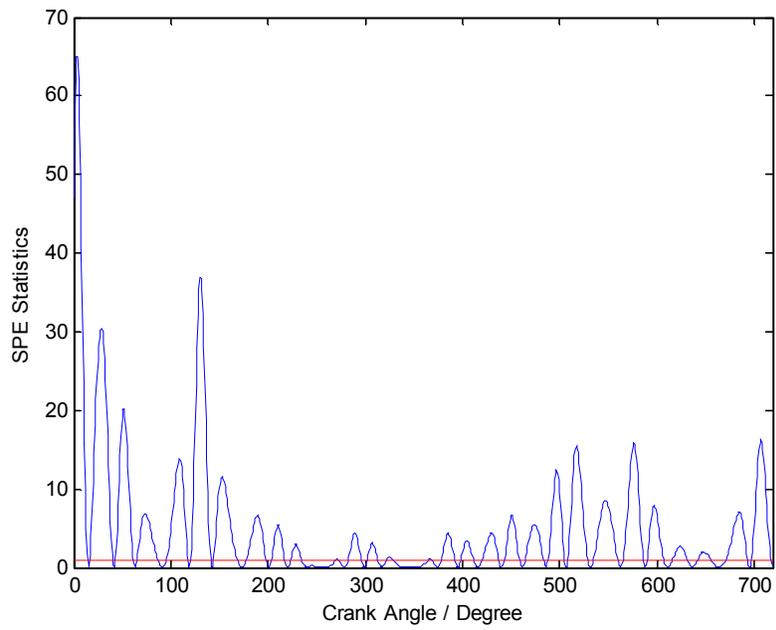
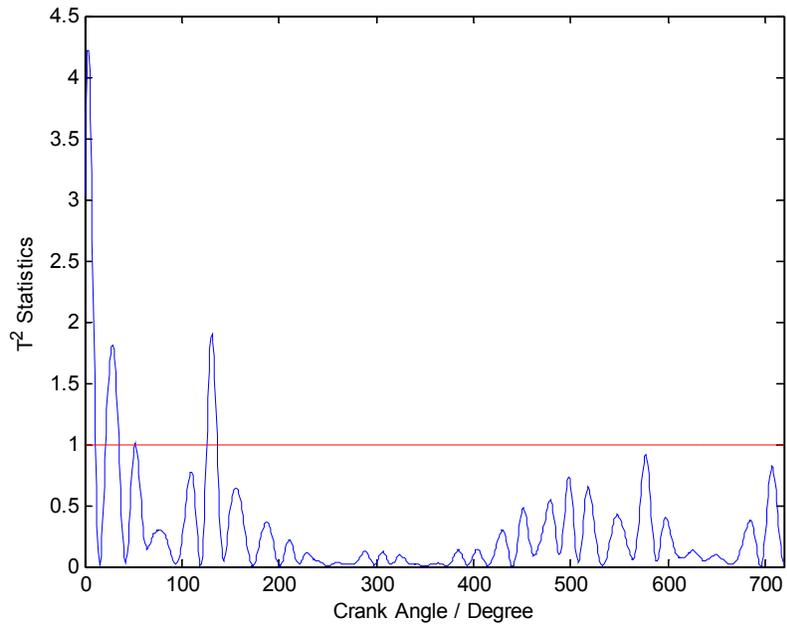
(b) KPCA for Level 2



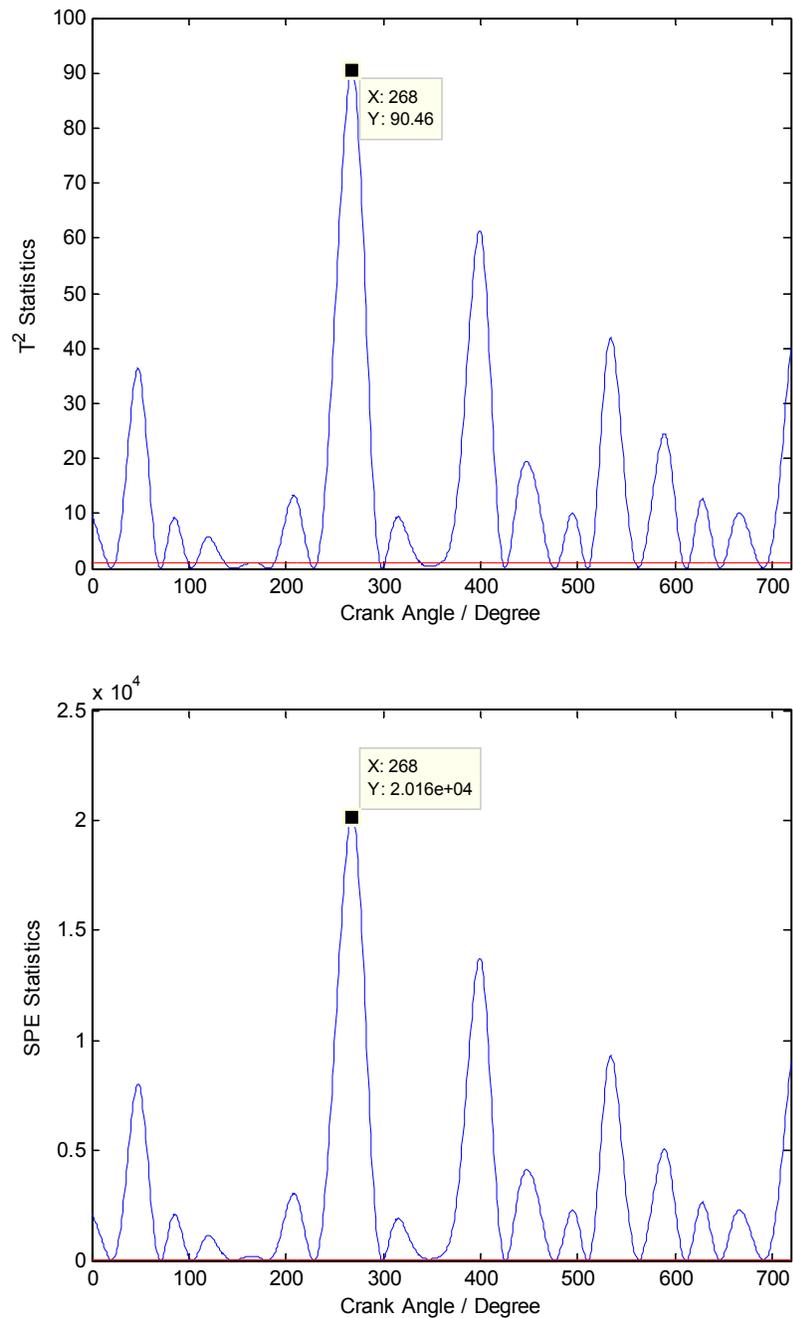
(c) KPCA for Level 3



(d) KPCA for Level 4



(e) KPCA for Level 5



(f) KPCA for Approximation Level

Figure 5.3.8 Wavelet-based KPCA FDD result for misfire fault in Cylinder 2.

From Fig. 5.3.8 we can find that the highest value for both the T^2 statistic and the SPE statistic appear on the Approximation Level at 268° in the CAD. This angle localizes the misfire fault in

Cylinder 2. The misfire faults in other cylinders also have the largest T^2 statistic and SPE statistic on the Approximation Level.

As with the VMD-based FDD algorithm in 5.3.2, the wavelet-based KPCA algorithm was also tested using 320 cycles of data. Its confusion matrix is shown in Fig. 5.3.9. This algorithm also achieves 100% accuracy.

1	40	0	0	0	0	0	0	0
2	0	40	0	0	0	0	0	0
3	0	0	40	0	0	0	0	0
4	0	0	0	40	0	0	0	0
5	0	0	0	0	40	0	0	0
6	0	0	0	0	0	40	0	0
7	0	0	0	0	0	0	40	0
8	0	0	0	0	0	0	0	40
	1	2	3	4	5	6	7	8

Figure 5.3.9 Confusion matrix for wavelet-based KPCA algorithm.

5.3.4 Experimental Results for the VMD-Based KPCA FDD Algorithm

The VMD parameters used here are the same as those in Table 5.3.1. For the misfire fault in Cylinder 2, the maximum T^2 statistics and SPE statistics are at 265° in the crank angle domain and at the 1st intrinsic mode as shown in Fig. 5.3.10. The misfire faults in other cylinders also have the largest T^2 statistics and SPE statistics in the 1st mode.

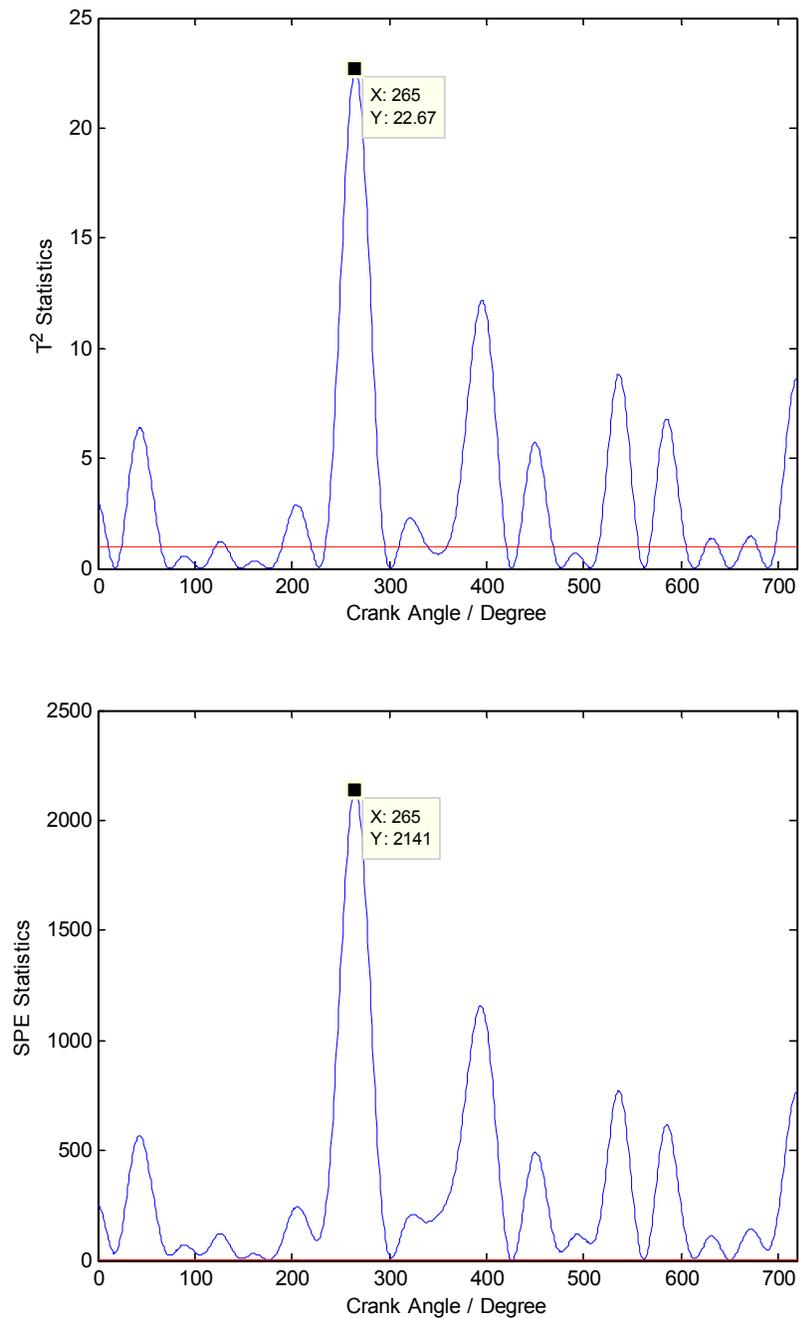


Figure 5.3.10 VMD-based KPCA for Misfire in Cylinder 2 (1st Mode)

The VMD-based KPCA algorithm was also tested using 320 cycles of data as with the wavelet-based KPCA algorithm. The confusion matrix (Fig. 5.3.11) shows that this algorithm also achieves 100% accuracy.

Output Class	1	40	0	0	0	0	0	0	
	2	0	40	0	0	0	0	0	
	3	0	0	40	0	0	0	0	
	4	0	0	0	40	0	0	0	
	5	0	0	0	0	40	0	0	
	6	0	0	0	0	0	40	0	
	7	0	0	0	0	0	0	40	
	8	0	0	0	0	0	0	40	
		1	2	3	4	5	6	7	8
		Target Class							

Figure 5.3.11 Confusion matrix for VMD-based KPCA algorithm.

5.3.5 Discussion of the Data-Processing-Technique-Based FDD Results

All of the three proposed FDD algorithms based on data processing techniques detect the misfire fault detection using the lowest frequency range. However, the crank angles where the peaks occurred differ from one algorithm to another. For example, Table 5.3.4 shows the locations of the peaks for the misfire fault in Cylinder 2. They are not identical, but are close to each other, especially for the two KPCA algorithms (265° vs 268°). The explanation for the small difference between the VMD-based KPCA and wavelet-based KPCA is that the variational mode decomposition and wavelet transform are two different time-frequency analysis tools, and the reconstructed signals produced by them are what these two KPCA algorithms rely on.

Table 5.3.4: Peak locations for a cylinder 2 misfire fault for the three signal based FDD algorithms studied.

Algorithm	Peak Location
The New FDD Index	274°
VMD KPCA	265°
WT KPCA	268°

The difference between the peak location of the new FDD index obtained by the VMD-based FDD algorithm and that of statistical indices obtained by the VMD/wavelet-based KPCA algorithm are much bigger. This is because the former algorithm considers only one variable (*i.e.* one accelerometer) while the latter is multi-variable (*i.e.* it uses all four accelerometers). The impact of the misfire fault on different parts of the engine is different, and can be captured by the accelerometers that are glued onto those parts. So when an algorithm uses one channel vs. multiple channels of data simultaneously it tends to produce a different result. This also helps explain why the KPCA algorithms sometimes have multiple dominant peaks whose amplitudes are the highest and closest at the lowest frequency range level (*i.e.* Intrinsic Mode 1 for VMD and the approximation level for the wavelet transform). For example, in the FDD of misfire in Cylinder 6, the T^2 statistics for mode 1 has two dominant peaks. As shown in Fig. 5.3.12, one is at 367° with an amplitude of 23.48 and the other at 488° with an amplitude of 23.24. In contrast, there is only one dominant peak with the new FDD indices obtained by the VMD-based FDD algorithm for all faulty conditions. Specifically for misfire fault in Cylinder 6, the peak location is 370° at mode 1.

The comparison between DWT and VMD is also worth mentioning. Although they bring very similar results when they are combined with KPCA, the wavelet-based linear PCA (Haqshenas, 2012) has better performance than VMD-based linear PCA by showing the faults clearer.

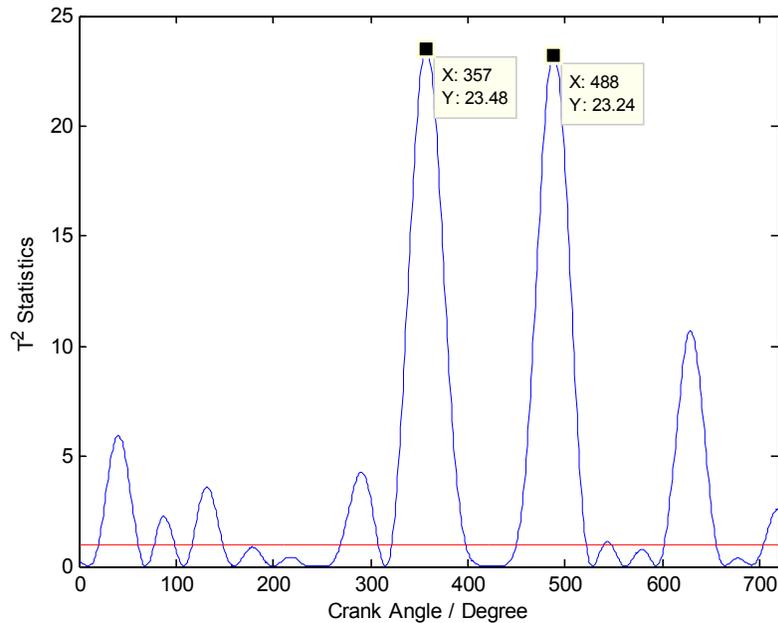


Figure 5.3.12 VMD based KPCA result (T^2 statistics at Mode 1)

5.3.6 ICE Fault Classification Results for the MLP Network Trained by the SVSF

In this experiment, 100 cycles of data are collected for each of the eight ICE conditions listed in section 5.2. 60% of them are used as training data and the remaining 40% as test data. The input to the ANN is the set of running-averaged wavelet coefficients of the vibration signal from one accelerometer. The SVSF parameters are set as $\gamma = 0$ and $\psi = 0.1$. The training stops when the mean square error (MSE) is smaller than 0.017. The performance of the SVSF-trained MLP ANN for misfire FDD is illustrated by the confusion matrices for both training data and test data in Fig. 5.3.13 and by the learning curve in Fig. 5.3.14. The condition index was defined in Table 5.2.1. According to the experimental results, the SVSF-trained MLP network is capable of classifying the engine's working conditions with 100% accuracy and the parameters of the network converge within 5 epochs⁶.

⁶ The number of epochs indicate how many times all of the training examples have been used to train the neural network. See Page 127 of (Haykin, 2008).

Output Class	1	60	0	0	0	0	0	0	
	2	0	60	0	0	0	0	0	
	3	0	0	60	0	0	0	0	
	4	0	0	0	60	0	0	0	
	5	0	0	0	0	60	0	0	
	6	0	0	0	0	0	60	0	
	7	0	0	0	0	0	0	60	
	8	0	0	0	0	0	0	60	
			1	2	3	4	5	6	7
		Target Class							

(a) Training Confusion Matrix

Output Class	1	40	0	0	0	0	0	0	
	2	0	40	0	0	0	0	0	
	3	0	0	40	0	0	0	0	
	4	0	0	0	40	0	0	0	
	5	0	0	0	0	40	0	0	
	6	0	0	0	0	0	40	0	
	7	0	0	0	0	0	0	40	
	8	0	0	0	0	0	0	40	
			1	2	3	4	5	6	7
		Target Class							

(b) Test Confusion Matrix

Figure 5.3.13 Confusion matrices for the SVSF-trained MLP network.

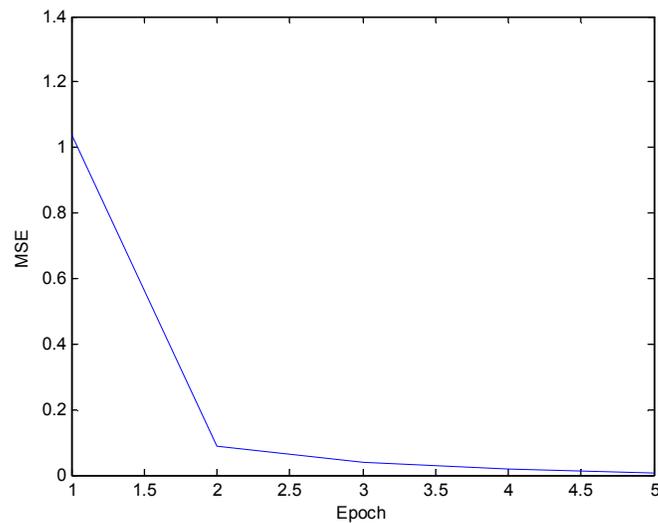


Figure 5.3.14 Learning curve for the SVSF-trained MLP network.

In comparison, the Levenberg–Marquardt (LM) algorithm makes a number of mistakes in the classification, which is shown in Fig. 5.3.15. Moreover, although the extended Kalman filter (EKF) is also able to achieve perfect classification results, it needs more iterations of training to converge. According to its learning curve (shown in Fig. 5.3.16), the convergence requires about 8 epochs, which is nearly twice as many as that of the SVSF training algorithm.

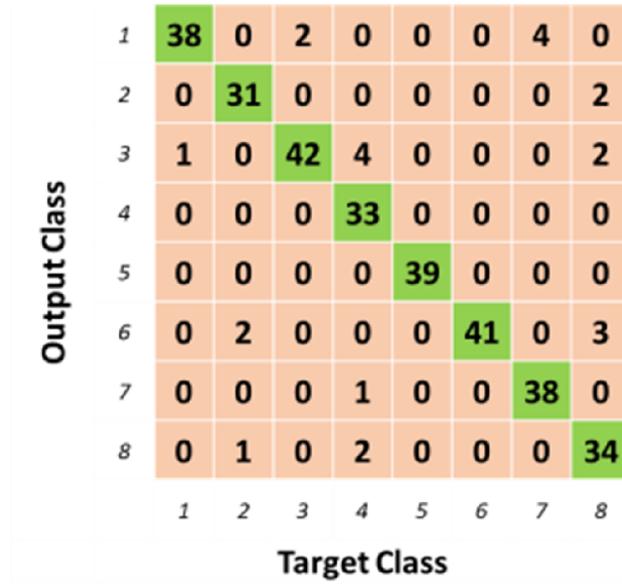


Figure 5.3.15 Test Confusion Matrix for the LM-trained MLP Network.

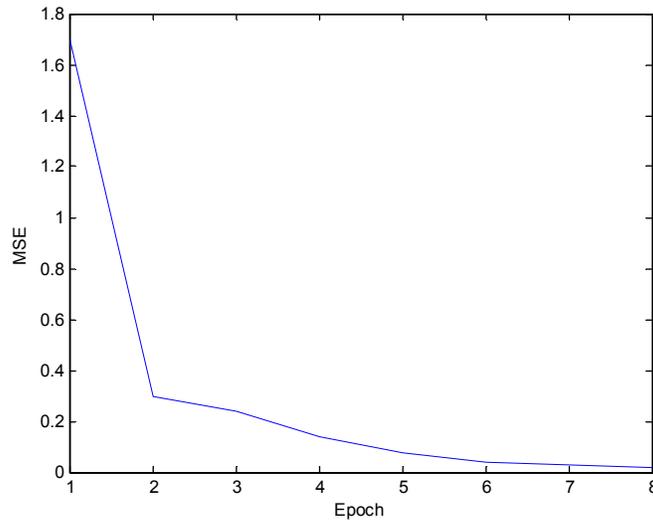


Figure 5.3.16 Learning Curve for the EKF-trained MLP Network.

As mentioned in Chapter 4, the selection of the network input is also an important problem. Apart from the running-averaged wavelet coefficients, the running-averaged vibration data in CAD and the running-averaged Fourier coefficient are also candidate inputs. However, according to their

confusion matrices (Fig. 5.3.17 and 5.3.18) they both lead to worse performance, especially the Fourier coefficients.

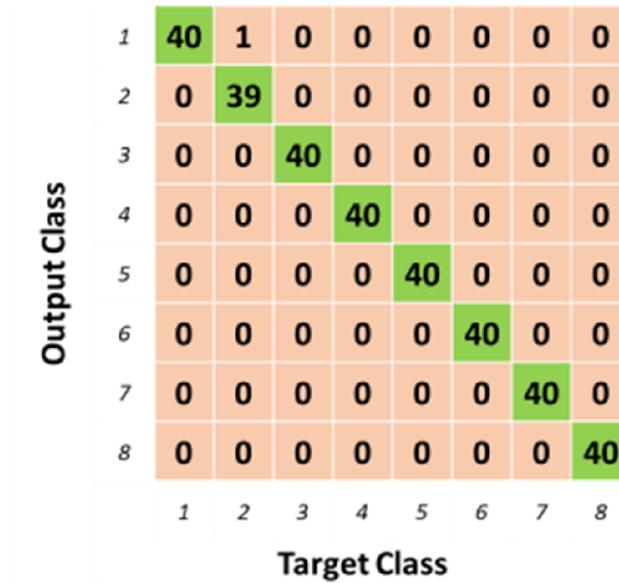


Figure 5.3.17 Test Confusion Matrix for the LM-trained MLP Network with Running-averaged Vibration in CAD as Input.

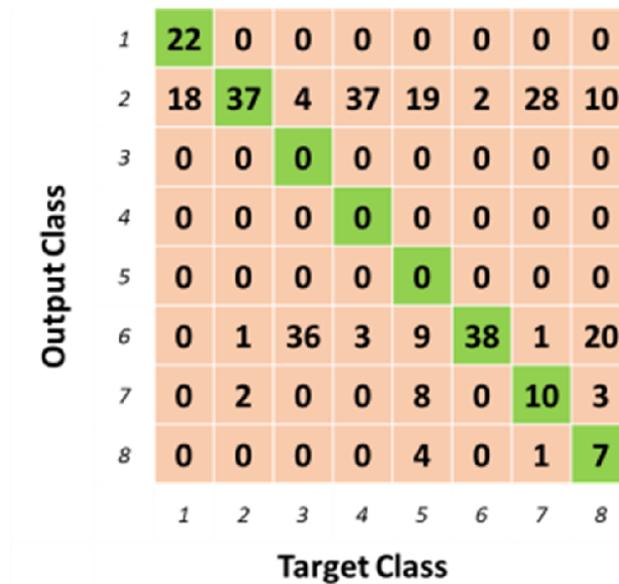


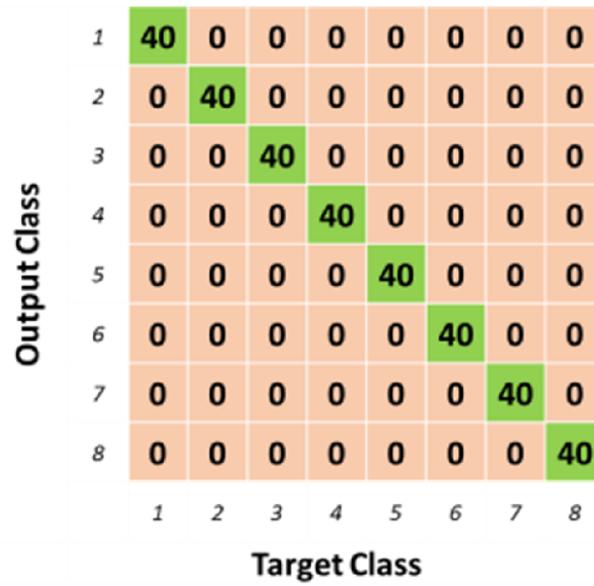
Figure 5.3.18 Test Confusion Matrix for the SVSF-trained MLP Network with Running-averaged Fourier Coefficient as Input.

5.3.7 ICE Fault Classification Results for the RBF Network Trained by the SVSF

The data used for RBF network is the same as in section 5.3.6. With the running-averaged wavelet coefficient of the vibration signal as input, the SVSF-trained RBF network is also able to classify the engine's working conditions perfectly for both training data and test data as shown in Fig. 5.3.19. The training converges within 4 epochs according to its learning curve in Fig. 5.3.20. In comparison, the EKF and gradient descent (GD) algorithms cannot achieve 100% classification accuracy and have a slower convergence rate. These are revealed by Fig. 5.3.21 to 5.3.24.

Output Class	1	60	0	0	0	0	0	0	
	2	0	60	0	0	0	0	0	
	3	0	0	60	0	0	0	0	
	4	0	0	0	60	0	0	0	
	5	0	0	0	0	60	0	0	
	6	0	0	0	0	0	60	0	
	7	0	0	0	0	0	0	60	
	8	0	0	0	0	0	0	0	60
		1	2	3	4	5	6	7	8
		Target Class							

(a) Training Confusion Matrix.



(b) Test Confusion Matrix.

Figure 5.3.19 Confusion matrices for the SVSF-trained RBF network.

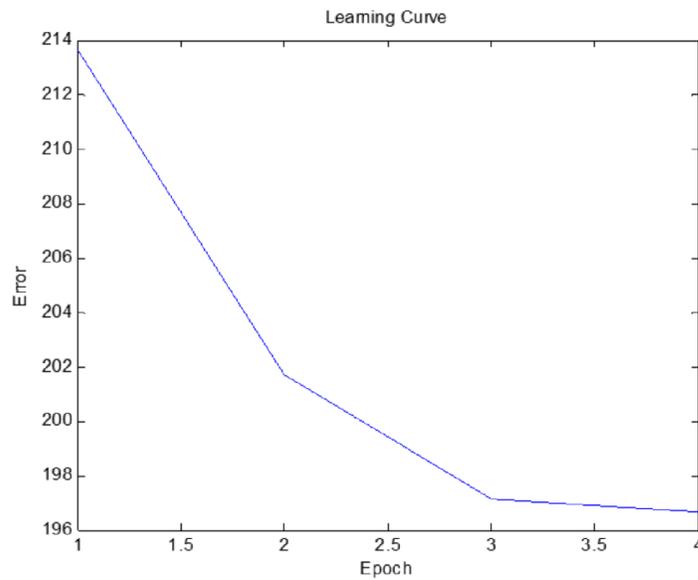


Figure 5.3.20 Learning curve for the SVSF-trained RBF network.⁷

⁷ The error in this figure is defined in (Eq. 4.3-23).

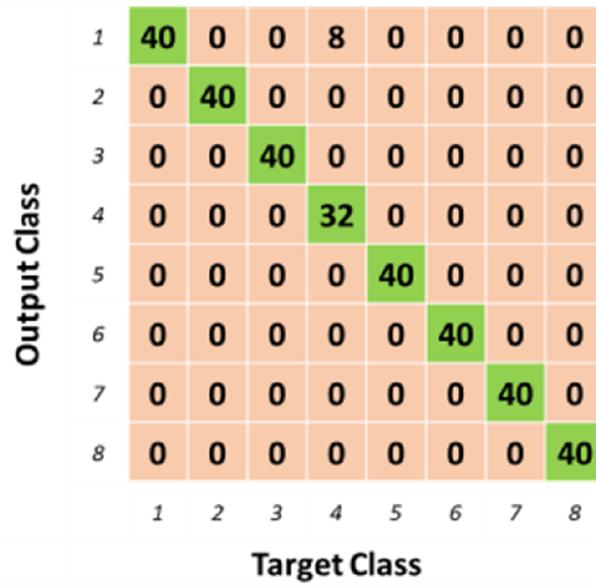


Figure 5.3.21 Test Confusion Matrix for the EKF-trained RBF Network.

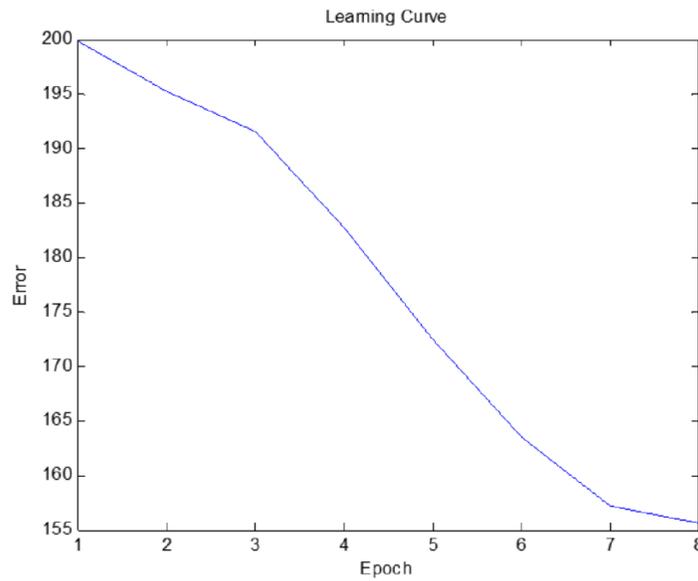


Figure 5.3.22 Learning Curve for the EKF-trained RBF Network.

Output Class	1	40	0	0	8	0	0	0	
	2	0	37	0	0	0	0	0	
	3	0	0	40	0	0	0	0	
	4	0	0	0	32	0	0	0	
	5	0	3	0	0	40	0	21	
	6	0	0	0	0	0	40	0	
	7	0	0	0	0	0	0	40	
	8	0	0	0	0	0	0	19	
		1	2	3	4	5	6	7	8
		Target Class							

Figure 5.3.23 Test Confusion Matrix for the GD-trained RBF Network.

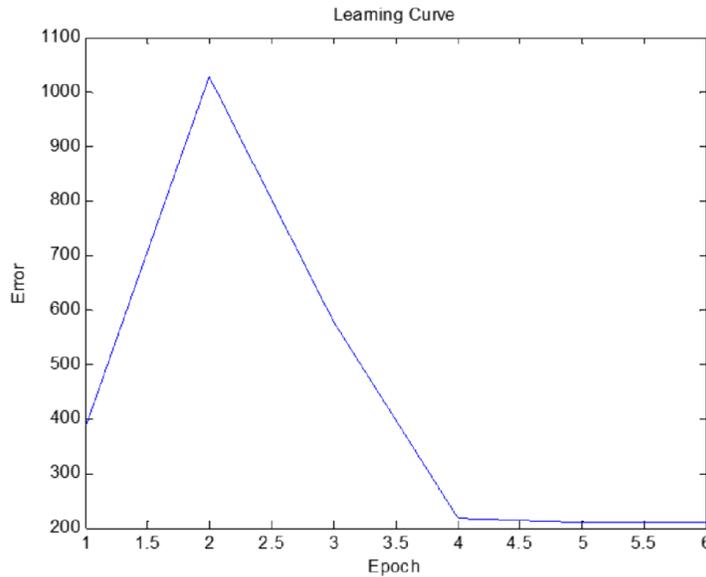


Figure 5.3.24 Learning Curve for the GD-trained RBF Network.

5.3.8 Discussion of the ANN FDD Results

The ANN FDD results are summarized in Table 5.3.5. We can observe that the wavelet coefficient is the most suitable ANN input. It is also clear that the SVSF training algorithm surpasses the other training algorithms in terms of accuracy and/or convergence.

Table 5.3.5: Summary of the ANN FDD results.

Network	Training Algorithm	Input	Accuracy (%)	Convergence (Epoch)
MLP	SVSF	Vibration in CAD	99.88	4
		Fourier Coefficient	44.83	3
	Wavelet Coefficient		100	5
		EKF	100	8
		LM	92.19	16
RBF	SVSF		100	4
	EKF		94.38	8
	GD		92.45	6

Note: The accuracy shown in Table 5.3.5 is calculated based on corresponding test confusion matrices and the value is the average over multiple experiments. The detailed accuracy distribution of each combination of ANN and training algorithm are exhibited in Fig. 5.3.25 to Fig. 5.3.32.

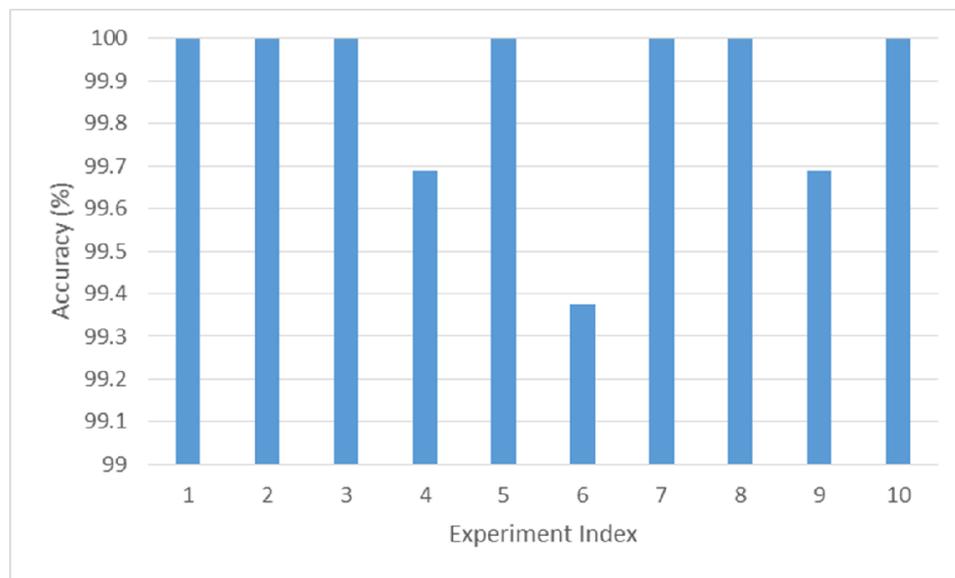


Figure 5.3.25 Classification Accuracy of SVSF-trained MLP network with Vibration in CAD as input in 10 experiments.

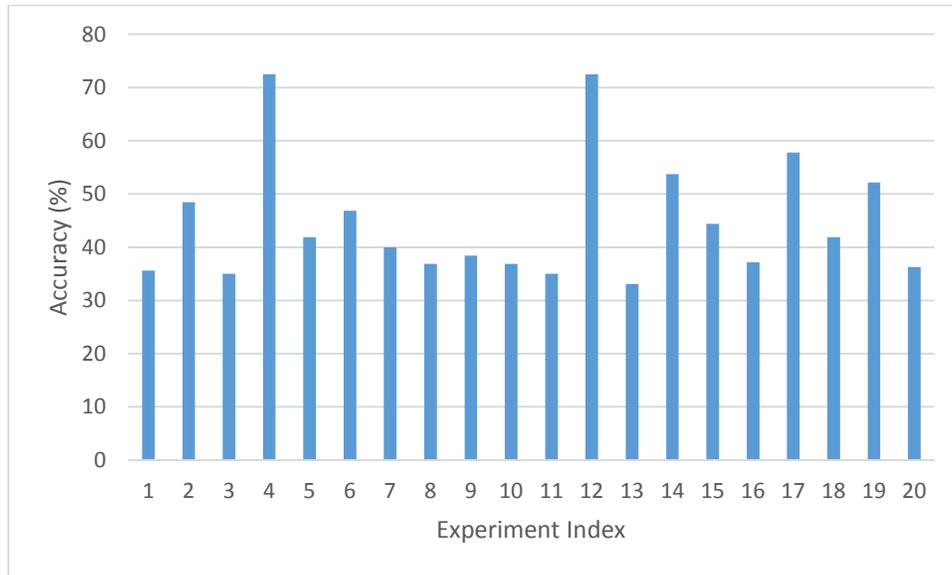


Figure 5.3.26 Classification Accuracy of SVSF-trained MLP network with Fourier Coefficient as input in 20 experiments.

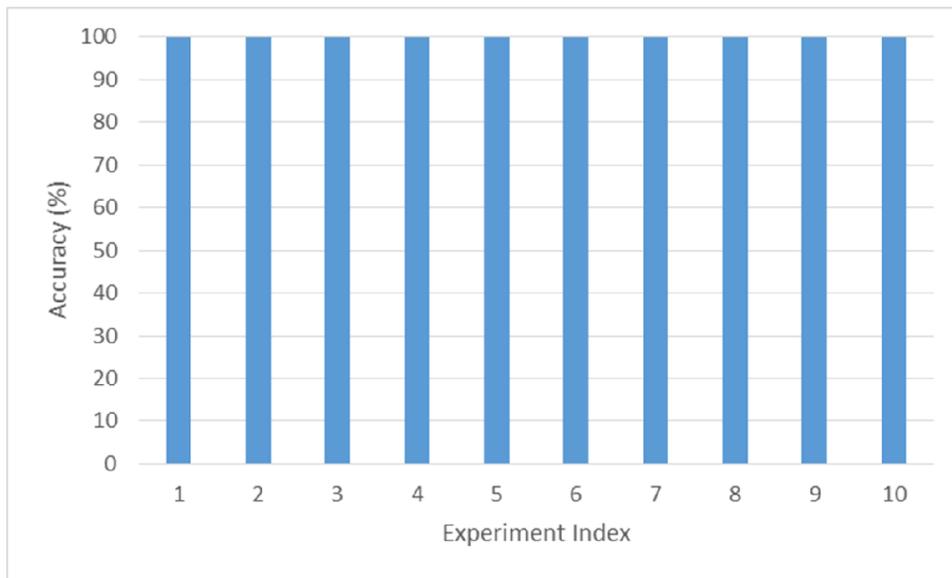


Figure 5.3.27 Classification Accuracy of SVSF-trained MLP network with Wavelet Coefficient as input in 10 experiments.

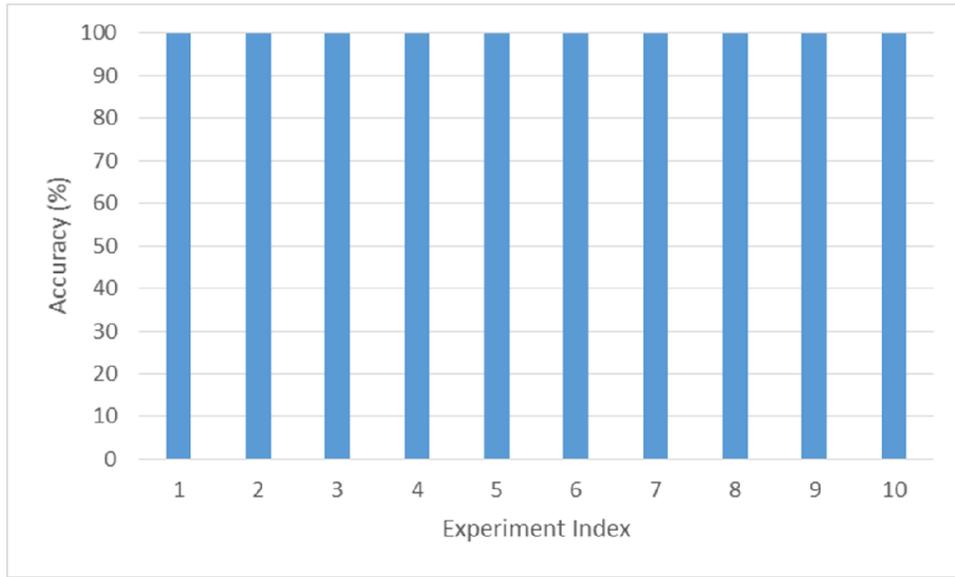


Figure 5.3.28 Classification Accuracy of EKF-trained MLP network with Wavelet Coefficient as input in 10 experiments.

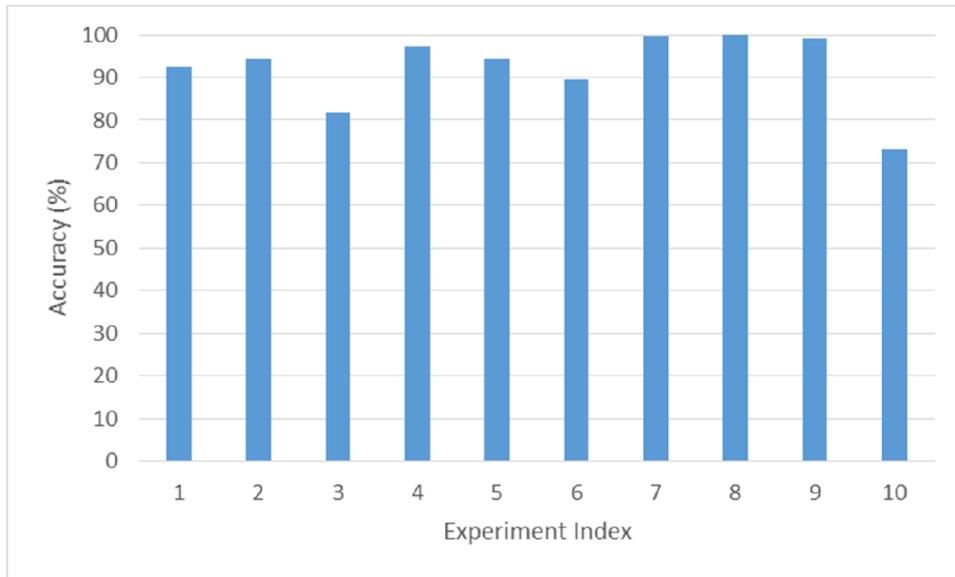


Figure 5.3.29 Classification Accuracy of LM-trained RBF network with Wavelet Coefficient as input in 10 experiments.

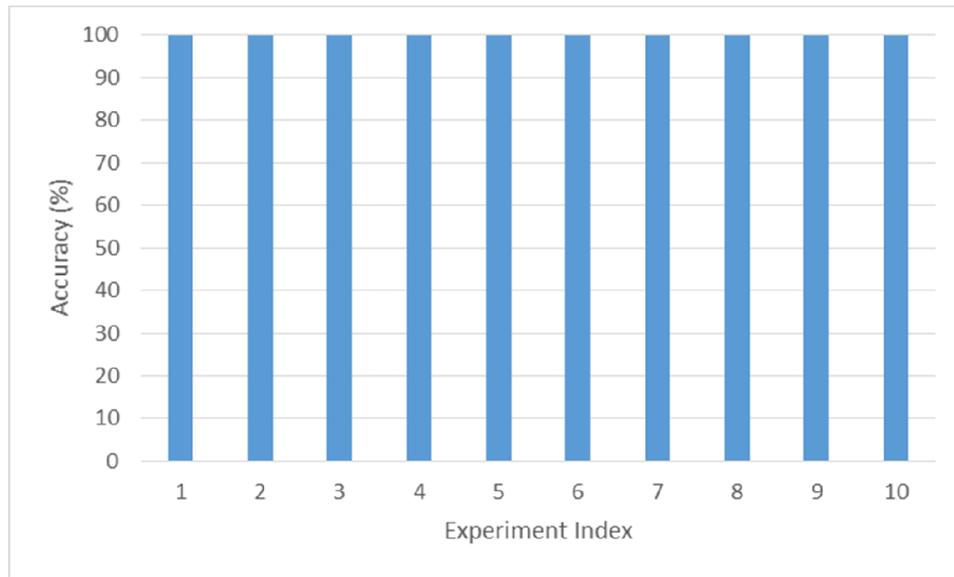


Figure 5.3.30 Classification Accuracy of SVSF-trained RBF network with Wavelet Coefficient as input in 10 experiments.

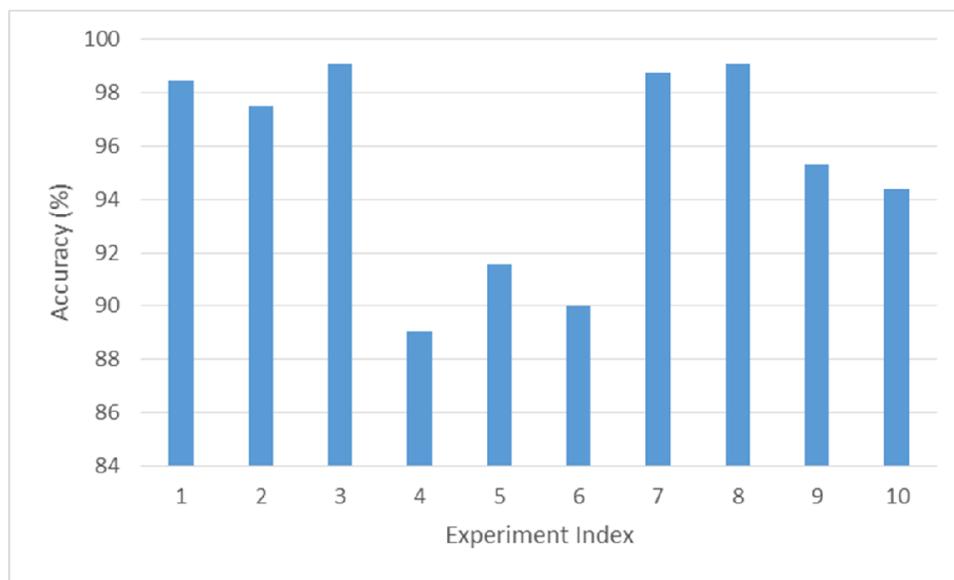


Figure 5.3.31 Classification Accuracy of EKF-trained RBF network with Wavelet Coefficient as input in 10 experiments.

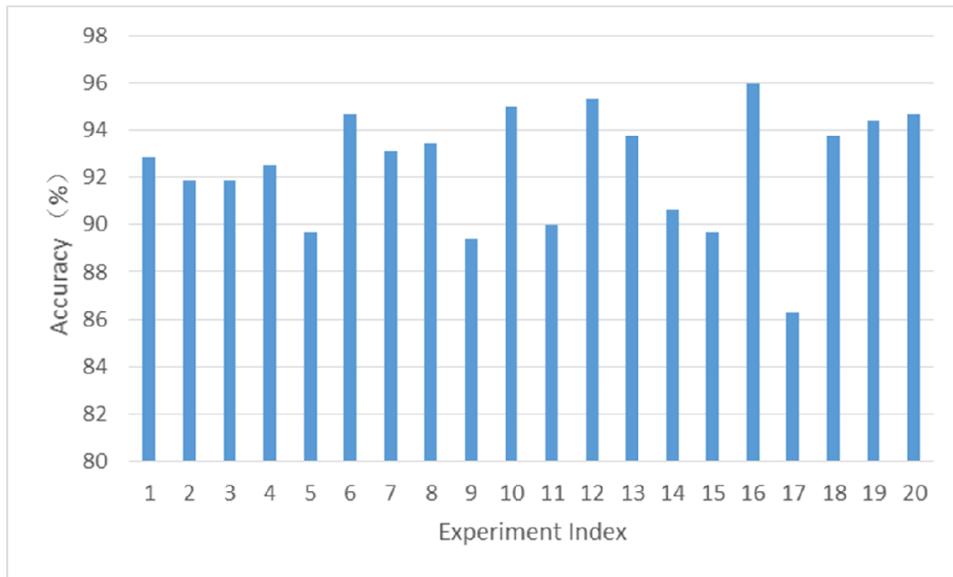


Figure 5.3.32 Classification Accuracy of GD-trained RBF network with Wavelet Coefficient as input in 20 Experiments

Chapter 6 Conclusions

This thesis presents the development of a FDD system for ICEs. The studied fault is the misfire fault which is induced by disconnecting the spark plug at a certain cylinder. After describing the experimental setup for the FDD research, several novel FDD methods based on data processing techniques and artificial neural network techniques are proposed, and experimentally verified.

The specific original contributions of the thesis are as follows:

1. Development of a new VMD-based FDD index and associated FDD algorithm. This index is calculated using data from a single accelerometer and its associated FDD algorithm is found to localize misfire faults at a certain crank angle with 100% accuracy.
2. Two nonlinear multi-variable multi-scale FDD algorithms, namely the wavelet-based KPCA and the VMD-based KPCA, are proposed. The new algorithms can localize the fault while taking the effect of multiple sensors into consideration. They are more suitable to nonlinear systems like engines compared with linear PCA algorithms. They are both shown to detect and localize the misfire faults with 100% accuracy.
3. The input selection problem for ANNs applied to ICE FDD is investigated. The running-averaged wavelet coefficients of the engine vibration signal is found to be the best input in comparison with the Fourier coefficients and pure vibration data in the CAD.
4. A MLP network trained by the SVSF is shown to achieve 100% fault classification accuracy when the running-averaged wavelet coefficients are used as its input.
5. A novel training algorithm for the radial basis function network based on the SVSF is proposed. The SVSF-trained RBF network is capable of 100% accurately classifying the engine's working conditions with the running-averaged wavelet coefficients as its input.

It outperformed EKF-trained and GD-trained RBF networks in terms of accuracy and/or convergence.

References

- Ahmed, R. (2011). *Training of Neural Networks Using the Smooth Variable Structure Filter with Application to Fault Detection*. McMaster University.
- Bro et al. (2014). Principal component analysis. *Analytical Methods*, 2812-2831.
- Chin, et al. (1986). Engine dynamics: Time-based versus crank-angle based. *SAE Technical Paper*.
- Cho et al. (2004). Sensor Fault Identification based on Kernel Principal Component Analysis. *Proceedings of the 2004 IEEE International Conference on Control Applications*.
- Choi, et al. (2005). Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 55-67.
- CTC. (2012). AC240. Retrieved from Connection Technology Center (CTC) Inc.:
https://www.ctconline.com/_100_mv_g_accelerometer___small_size.aspx?qcwg=116_333&qman=
- Dekking, et al. (2005). *A Modern Introduction to Probability and Statistics*. Springer.
- Dragomiretskiy et al. (2014). Variational Mode Decomposition. *IEEE Transactions on signal processing*, 531-544.
- Dragomiretskiy, et al. (2014). Variational Mode Decomposition. *IEEE Transactions on signal processing*, 531-544.
- Du, et al. (2012). Diagnosis for Engine Misfire Fault Based on Torsional Vibration and Neural-Network Analysis. *Advanced Materials Research. Vol. 433.*, 7240-7246.
- Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. CRC Press.

- Ghate et al. (2010). Optimal MLP neural network classifier for fault detection of three phase induction motor. *Expert Systems with Applications*, 3468-3481.
- Habibi, S. (2007). The smooth variable structure filter. *Proceedings of the IEEE*, 1026-1059.
- Haqshenas, S. R. (2012). *Multiresolution-multivariate analysis of vibration signals; application in fault diagnosis of internal combustion engines*. McMaster University.
- Haykin, S. (2008). *Neural Networks and Learning Machines (Third Edition)*. Prentice Hall.
- Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer.
- ITI. (2008). *Model PS01 Product Manual*.
- Kang et al. (2001). Condition assessment of power transformer on-load tap-changers using wavelet analysis. *IEEE Transactions on Power Delivery*, 394-400.
- Lee et al. (1997). Use of directional spectra for detection of engine cylinder power fault. *Shock and Vibration*.
- Lin et al. (2000). Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis. *Journal of sound and vibration*, 135-148.
- Loutridis, S. (2004). Damage detection in gear systems using empirical mode decomposition. *Engineering Structures*, 1833–1841.
- Lu et al. (2012). Fault Diagnosis of Diesel Engine Based on EMD and TFD. *Advances in Mechanical and Electronic Engineering*, 261-266.

- Mallat, S. (2009). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way (3rd Edition)*. Academic Press.
- Mamis et al. (2011). FFT based fault location algorithm for transmission lines. *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on. IEEE*.
- Mathworks. (2015). *logsig*. Retrieved from Matlab Support:
<http://www.mathworks.com/help/nnet/ref/logsig.html>
- Mathworks. (2015). *purelin*. Retrieved from Matlab Support:
<http://www.mathworks.com/help/nnet/ref/purelin.html>
- Mohanty et al. (2014). Bearing Fault Analysis using Variational Mode Decomposition. *Journal of Instrumentation Technology & Innovations*, 20-27.
- Moosavian et al. (2015). Spark plug fault recognition based on sensor fusion and classifier combination using Dempster–Shafer evidence theory. *Applied Acoustics*, 120-129.
- NI. (2014, June). *Discrete Wavelet Transform (Advanced Signal Processing Toolkit)*. Retrieved from National Instruments: http://zone.ni.com/reference/en-XX/help/372656C-01/lvasptconcepts/wa_dwt/
- Porteiro et al. (2011). Diesel engine condition monitoring using a multi-net neural network system with. *Applied Thermal Engineering*, 4097-4105.
- Prandoni et al. (2008). *Signal Processing for Communications*. EPFL Press.
- Qin, S. J. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 480-502.

- Rajakarunakaran et al. (2008). Artificial neural network approach for fault detection in rotary system. *Applied Soft Computing*, 740-748.
- Shim et al. (2010). A real-time fault diagnosis system for UPS based on FFT frequency analysis. *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*.
- Sifuzzaman et al. (2009). Application of wavelet transform and its advantages compared to Fourier transform. *Journal of Physical Sciences*, 121-134.
- Simon, D. (2002). Training radial basis neural networks with the extended Kalman filter. *Neurocomputing* , 455-475.
- Song, Y. (2013). *Electro-Hydrostatic Actuator Fault Detection and Diagnosis*. Hamilton: McMaster University.
- Sun et al. (2002). Singularity analysis using continuous wavelet transform for bearing fault diagnosis. *mechanical systems and signal processing*, 1025-1041.
- Venkatasubramanian, et al. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & chemical engineering*, 293-311.
- Villegas et al. (2010). Principal component analysis for fault detection and diagnosis. experience with a pilot plant. *Proceedings of the 9th WSEAS international conference on computational intelligence, man-machine systems and cybernetics*.
- Wang et al. (2015). Research on variational mode decomposition and its application in detecting rub-impact fault of the rotor system. *Mechanical Systems and Signal Processing*, 243-251.

Wu et al. (2004). Induction machine fault detection using SOM-based RBF neural networks.

Industrial Electronics, IEEE Transactions on, 183-194.

Yu et al. (2005). Application of EMD method and Hilbert spectrum to the fault diagnosis of roller bearings. *Mechanical Systems and Signal Processing*, 259–270.

Zeiler et al. (2010). Empirical Mode Decomposition - an introduction. *Neural Networks (IJCNN)*, *The 2010 International Joint Conference on* (pp. 1-8). Barcelona: IEEE.