Fault Detection and Diagnosis of a Diesel Engine Valve Train

Fault Detection and Diagnosis of a Diesel Engine Valve Train

By Justin Flett

A Thesis

Submitted to School of Graduate Studies

In Partial Fulfillment of Requirements

For the Degree

Master of Applied Science

McMaster University

Master of Applied Science (2013)

McMaster University

(Mechanical Engineering)

Hamilton, Ontario

Title:	Fault Detection and Diagnosis of a Diesel Engine Valve Train
Author:	Justin Flett, B. Eng. (McMaster University)
Supervisor:	Dr. Gary M. Bone, Professor
Number of Pages:	I - XXV
	1 – 140

Abstract

Mechanical faults and failures are a common occurrence within all mechanical systems. Over time a minor mechanical fault can turn into a major failure if not detected and serviced with proper maintenance. The detection and classification of minor faults is essential in reducing the occurrence of major faults, which will result in an increased lifetime of the mechanical system. Additionally, a good fault detection method should lead to increased efficiency, increased safety, improved performance, and reduced total lifetime costs of the system. One of the most commonly used mechanical systems is the internal combustion engine. Internal combustion engines dominate the automotive industry, and have numerous other applications in generation, transportation, etc. This thesis presents the development of a fault detection and diagnosis (FDD) system for use with an internal combustion engine valve train.

A FDD system was developed with a focus on the valve impact amplitudes. Engine cycle averaging and band-pass filtering methods were tuned and utilized for improving the signal to noise ratio. A novel feature extraction method was developed that included a local RMS sliding window method and an adaptive threshold. Faults were seeded in the form of deformed valve springs, as well as abnormal valve clearances. The engine's manufacturer specifies that a valve spring with 3 mm or more of deformation should be replaced. This thesis investigated the detection of a relatively small 0.5mm spring deformation. Valve clearance values were adjusted 0.1mm above and below the nominal clearance value (0.15mm) to test large clearance faults (0.25mm) and small clearance faults (0.05mm). The performance of the FDD system was tested using an instrumented diesel engine test bed. A comparison of numerous signal processing techniques and classification methods was performed.

The FDD system implementing the Naïve-Bayes classification method produced a worst case detection accuracy (DA) of 99.95% and worst case classification accuracy (CA) of 99.95% for spring faults of 0.5mm deformation, tested on multiple valves with a training size of 40 engine cycles. The total FDD execution time including feature extraction, training, and testing over 11,000 engine cycles was 4.5s. Alternative classification methods also worked well with the FDD system, with decision trees and linear discriminant analysis producing worst case CAs of 98.96% and 97.77%, respectively. Further experimental investigations were done where fault scenarios were varied, including simultaneous fault scenarios, and numerous parameter values were altered. The proposed FDD method gave reliable and accurate classification results for many different cases, demonstrating the generality and robustness of the proposed method.

Table of Contents

Acknowledgements

The completion of this thesis would not have been possible without some key contributors. I would like thank my supervisor, Dr. Gary Bone, for his guidance and support throughout this project. I would also like to acknowledge Darren Van Rooyen, a colleague who assisted me throughout the project and helped develop some of the data acquisition system used in this thesis. Additionally I would like to acknowledge the assistance of the Mechanical Engineering Department technicians: Mr. M. MacKenzie, Mr. J. McLaren, Mr. R. Lodewyks, and Mr. J. Verhaeghe.

List of Figures

List of Tables

Abbreviations & Acronyms

ADC	Analog-to-Digital Converter
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
BPN	Back-propagation Network
CWT	Continuous Wavelet Transform
DAC	Digital-to-Analog Converter
DAQ	Data Acquisition System
DFT	Discrete Fourier Transform
DPLS	Discriminant Partial Least Squares
DSP	Digital Signal Processing
DT	Decision Tree
DWT	Discrete Wavelet Transform
EVC	Exhaust Valve Closing
EVC1	Exhaust Valve Closing of Cylinder 1
EVC2	Exhaust Valve Closing of Cylinder 2

FDA	Fisher Discriminant Analysis
FDI	Fault Detection and Isolation
FFT	Fast Fourier Transform
GRNN	Generalized Regression Neural Network
HDDE	Heavy Duty Diesel Engine
ICE	Internal Combustion Engine
IDI	Indirect Injection
IVC	Intake Valve Closing
IVC1	Intake Valve Closing of Cylinder 1
IVC2	Intake Valve Closing of Cylinder 2
k-NN	k-Nearest Neighbour
LCF	Large Clearance Fault
LDA	Linear Discriminant Analysis
MLP	Multilayer Perceptron
MSE	Mean-Squared Error
NB	Naïve-Bayes

NI	National Instruments
PCA	Principal Component Analysis
PCI	Peripheral Component Interconnect
PFSA	Partial Sampling and Feature Averaging
RMS	Root Mean Square
RPM	Revolutions Per Minute
SCF	Small Clearance Fault
SF	Spring Fault
SNR	Signal-to-Noise Ratio
TAWS	Time Averaged Wavelet Spectrum

Nomenclature

b_h	Hidden layer bias value
b_o	Output layer bias value
$C_{i,j}$	Confusion matrix entry of row i and column j
CA_i	Classification accuracy of class i
DA_i	Detection accuracy of class i
E_s	Engine speed
F_L	Spring fault severity level
f_b	Bandwidth frequency of a signal
f_N	Nyquist rate
f_s	Sampling frequency
I_{amp}	Impact amplitude
I_R	Impact region

I_{RE}	Impact region end point
I_{RS}	Impact region start point
N_a	Number of engine cycles averaged
N_d	Number of data points in a single engine cycle, $N_d = 2880$
N_m	Span of the median filter
N_s	Number of neighbouring data points on either side of $y_s(i)$
N_{snr}	Number of values used in the RMS calculation for the SNR
N_T	Number of test cases used for testing
N_t	Training size
N_w	Number of neighbouring data points on either side of $y(i)$
N_{ws}	Window size of sliding window $N_{ws} = 2N_w + 1$ used for LRMS calculation
$P\left(x \mid c\right)$	Probability of some value x residing in some class c
S_{class}	Classification score for LDA for a given class
s_k	Classification coefficients of feature k
s_f	Threshold scaling factor

sn	Segment number
V_{rms}	Vibration RMS value
V_{acc}	Vibration value of the accelerometer
$w_{h,i}$	Weight of ith hidden layer input
w_i	Weight of the network node
$w_{o,i}$	Weight of ith output layer input
x_h	Hidden layer activation input
x_i	Input value to the neural network node
x_o	Output layer activation input
Y	True value of the data
$\hat{Y}$	Prediction value of the neural network
Y_{rms}	Threshold value
$y(i)$	Post-processed vibration signal including averaging and filtering
y_h	Hidden layer output
$y_{lrms}(i)$	Local RMS signal extracted from processed vibration signal
$y_m(i)$	Median filtered value for the ith data point (median average filter)

y_{nn}	Neural network output
y_o	Output layer output
$y_s(i)$	Smoothed value for the ith data point (moving average filter)
$\mu_{class,vn,sn}$	Mean value of a given class, valve number, segment number
$\sigma_{class,vn,sn}$	Standard deviation of a given class, valve number, segment number
$\Phi(x)$	Gaussian distribution function

CHAPTER 1: Introduction

1.1 Motivation for this research

The internal combustion engine (ICE) has been the critical component of the automotive industry since its inception. Even as the green energy initiative moves forward, the ICE still holds its value in both conventional and hybrid vehicles. ICEs are composed of a number of mechanical components, each of which may be subject to various mechanical faults. The valve train is a crucial component of an engine which controls the intake and exhaust timing. Faults within the valve train will result in a reduction of performance and reliability.

Faults occur naturally in mechanical systems over time. The detection of faults in their early stages can be beneficial for the avoidance of larger, more severe faults. Fault detection and diagnosis (FDD) methods are used to monitor a system, identify when a fault has occurred, and identify the type of fault and its location. FDD methods are commonly seen in mechanical systems. If a fault can be correctly detected and diagnosed, corrective measures can be applied to repair the fault and reduce any further damage to the system.

A FDD method applied to an ICE can have significant impact on both the maintenance and research and development (R&D) processes of ICEs in the automotive industry. Fault detection in the R&D stage can decrease the amount of defective products and reduce recalls in the future. Early detection in the maintenance stage can increase asset lifetime, improve performance and reliability, and reduce the frequency of major failures. Further development of the FDD system may produce a real-time online application in the future.

1.2 Fault Detection and Diagnosis Classification Methods

Most fault detection methods consist of three main phases which have distinct functions, where each phase may contain a number of different steps. A FDD system generally requires the following phases:

Signal Acquisition: Acquiring signals that measure the physical conditions of a system and convert them into digital values that can be manipulated and analyzed. Commonly consisting of sensors, data acquisition (DAQ) hardware, and DAQ software.

Signal Processing: Improving the signal to provide better understanding of the signal information. May include noise reduction, filtering, signal compression, and other processing techniques. Feature extraction methods may be used to extract relevant information.

Detection & Diagnosis: Using a monitoring system to identify a fault in a system. Mathematical or statistical operations are performed on a signal to detect and diagnose normal or faulty operation of the system.

Fault detection is commonly separated in two different methods. Signal model-based fault detection as described in Figure 1-1, and process model-based fault detection as described in Figure 1-2. Where **U** is the process input, **Y** is the process output, **N** is some external disturbance, and **S** is the fault symptom.

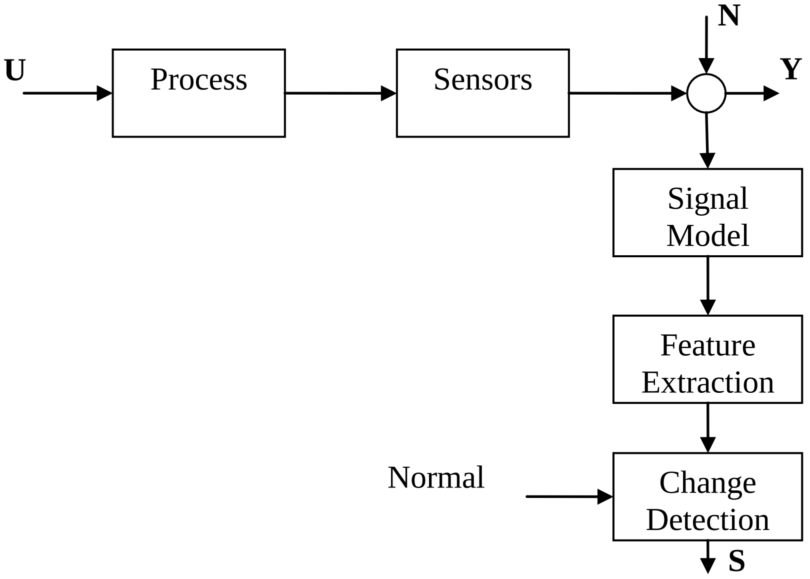

Figure 1-1: Fault detection with signal models (Isermann R. , 2006)

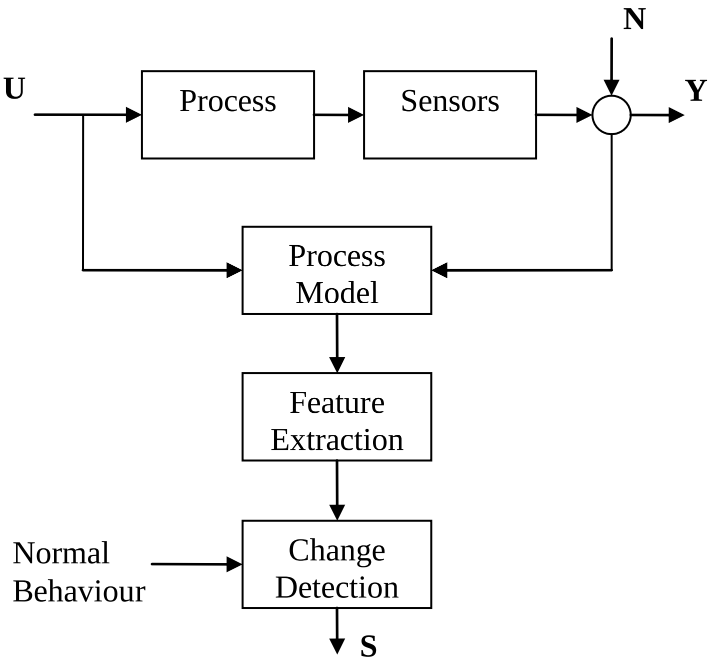

Figure 1-2: Fault detection with process models (Isermann R. , 2006)

Signal models attempt to detect changes of the signal behaviour caused by the process faults. Process models attempt to detect changes in the process behaviour by using some mathematical model of the process. This thesis will focus on fault detection using signal models, as the complexity of a diesel engine makes process modelling difficult and unnecessary for our specific application.

Fault detection includes analyzing the measured signals to extract pertinent features and generate the nominal and faulty symptoms. Once these symptoms have been obtained they can be used to diagnose the faults. The general approach for a signal model-based FDD can be seen in Figure 1-3 (Do & Chong, 2011).

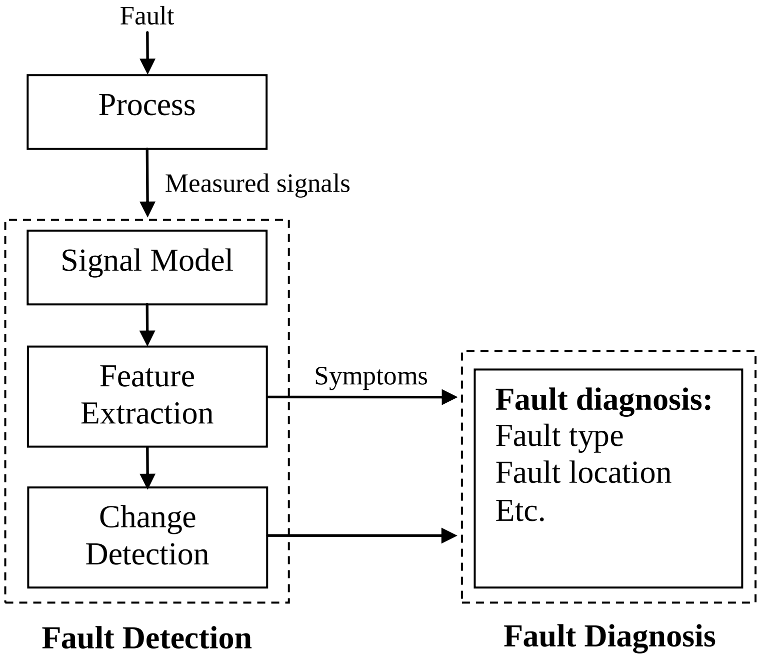

Figure 1-3: General structure of a FDD system (Do & Chong, 2011)

The two main classes of diagnosis are classification methods and inference methods. Figure 1-4 (Isermann, 2006) outlines some fault diagnosis methods. This thesis will focus on classification methods for our FDD.

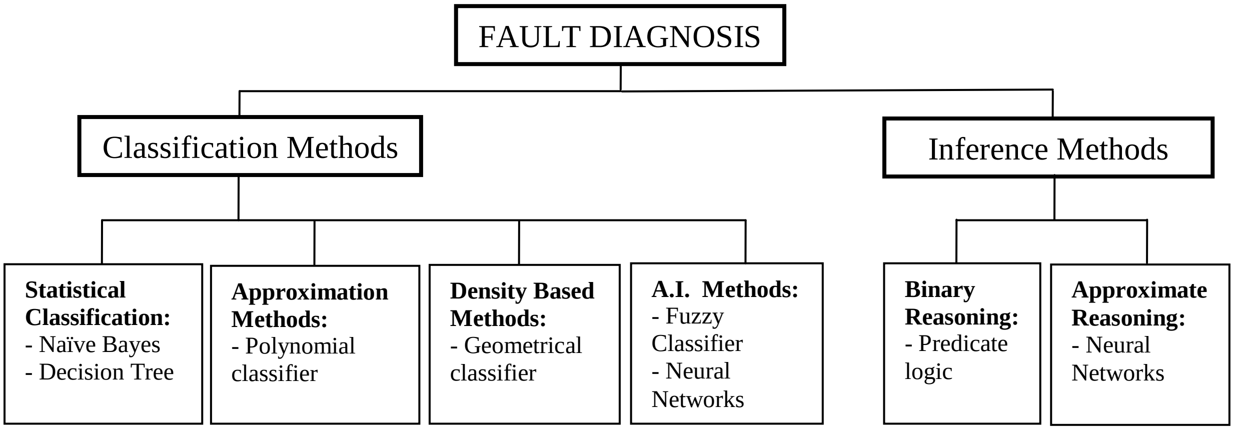

Figure 1-4: Fault diagnosis methods (Isermann R. , 2006)

1.3 Objective and Organization of this Thesis

The objectives of this thesis are to:

1. Develop a DAQ system for a FDD system.
2. Develop a novel FDD method and apply it to an instrumented diesel engine to detect specific valve train faults.
3. Test the FDD method on the diesel engine test bed.
4. Compare the developed method with other common FDD classification methods, and note any improvements or drawbacks.

The organization of the thesis is as follows. In Chapter 2 the relevant literature on fault detection methods is reviewed. The test bed and signal acquisition are described in Chapter 3. In Chapter 4 the signal processing methods are presented. Chapter 5 describes the new fault detection method as well as other common FDD methods. Experimental results are presented in Chapter 6. Finally, conclusions and recommendations for future work are presented in Chapter 7.

CHAPTER 2: Literature Review

2.1 Introduction

In this chapter the research literature related to FDD methods will be reviewed. FDD methods have been developing for numerous years and the related literature is abundant. The literature related to the more specific case of engine valve train faults however is quite sparse. The following aspects will be reviewed: digital signal processing for FDD systems; classification methods for FDD systems; and engine valve train FDD systems.

2.2 Digital Signal Processing for Fault Detection Systems

Modern signal processing includes the operations on, and analysis of, discrete time signals. Signal processing techniques are used to manipulate or modify an information signal in order to improve it (Stranneby & Walker, 2004). Numerous signal processing techniques have been studied in the literature, with advantages and disadvantages determined by the specific application. The following subsections will review the techniques most commonly used with FDD systems.

2.2.1. Fourier Transform

The Fourier transform is commonly used to convert a function from the time domain to the frequency domain. A fast Fourier transform (FFT) is an algorithm for rapidly computing the Fourier transform of discrete time signal.

Obiad and Habetler (2003) developed a FDD for induction motors with arbitrary load conditions. They claimed that mechanical faults such as load unbalance and shaft misalignment in induction motors can be detected by monitoring the stator current frequency spectrum. Load unbalances were seeded by adding nuts and bolts on a balanced metal disk at different radial distances from the shaft. Angular misalignments were seeded by rotating the machine some angle from its original position. They suggested that such faults could be detected by looking at the frequency sidebands in the stator current and that these faults could be successfully detected at various levels of load. They obtained the FFT of the current so the frequency spectrum could be analyzed. A fault detection algorithm was presented by extracting the frequency sidebands from the FFT. No fault detection results were given.

Betta et al. (2001) customized a previously developed FFT analyzer for FDD of an induction motor based on vibration analysis. A non-fault model was developed as well as four fault models: shaft unbalance, misalignment, mechanical looseness in the bearing cap, and an outer ball bearing defect. Their FDD method consisted of a pattern matching procedure by comparing the actual device tests with the non-fault and fault models. Their analysis suggested that the faults considered contained meaningful signals no higher than 1 kHz. Signal processing techniques including filtering, windowing, FFT, and feature extraction were applied to improve the FDD algorithm. Each sample sequence had its FFT evaluated and its characteristics compared to the models. Their experiments included 200 tests for each fault case, with the lowest correct diagnosis rate of 97.5%.

2.2.2. Wavelets

The wavelet transform is similar and often compared to the Fourier transform. The main difference being that wavelet transfoRMS are localized in both time and frequency compared to a Fourier transform which is only localized in frequency. This is advantageous in cases where the time features of a signal are of value.

Lin and Qu (2000) used a Morlet wavelet for feature extraction for mechanical fault diagnosis. A denoising method based on wavelet analysis was used for extracting the vibration period for FDD of rolling bearing and gear box faults. For the bearing size studied, the characteristic impulse period for the rolling bearing inner-race damage was found to be 0.0061s. Using the Morlet wavelet denoising technique gave a period of about 0.006s, which agreed with the characteristic period for the inner-race damaged rolling bearing. Additionally, a gear box was set up with impulse components whose period was equal to 0.24s. Again the Morlet wavelet produced a denoised signal with a period of 0.24s.

Zheng et al. (2002) developed a gear fault detection method based on a continuous wavelet transform. Vibration signal analysis was used to detect cracking teeth in gear systems. They recorded the vibration signal from the beginning to the end of life test, and then picked five time blocks with equal intervals. The blocks represented 0%, 25%, 50%, 75%, and 100% gear fault advancement assuming linear fault advancement with time. Wavelet analysis characterized the signal in both time and frequency domains. A time-averaged wavelet spectrum was developed to extract gear fault feature energy. The feature energy was computed from the wavelet coefficients. Their results showed successful gear fault diagnosis for 10 test cases.

Jian-Da Wu (2009) developed an expert system for fault diagnosis in internal combustion engines using wavelet packet transfoRMS (WPT). The faults detected included: air leakage faults, cam-shaft sensor faults, electronic control thermal faults, and misfiring faults. Sound emission signals were analyzed using the WPT algorithm to effectively extract signal features.

They used Daubechies wavelets as the mother wavelets used to build and perform the WPT method. The experimental data consisted of 150 data sets for each operating condition. Training was done with 30 data sets, and 120 data sets were used for testing. They found that all the recognition rates were over 95% and concluded that the WPT method is useful for FDD systems. They suggest that the WPT technique has several advantages over continuous wavelet transform (CWT) and discrete wavelet transform (DWT) techniques, however they provided no results using CWT or DWT for comparison.

2.3 Classification for Fault Detection Systems

Classification is the process of classifying information into specific categories or classes. Classification methods are useful in FDD systems as specific data sets can be classified into specific faults. The most popular techniques will now be reviewed.

2.3.1. Bayesian Classification

Bayesian classification is a method which implements Bayes' probability theorem. The probability estimate for a hypothesis is updated as additional evidence is learned. Naïve-Bayes (NB) classification generates a probabilistic classifier by applying Bayes' theorem with independence assumptions.

House et al. (1998) developed a classification technique for FDD of an air-handling unit. The objective was to demonstrate the application of several classification techniques to fault detection. Artificial neural networks, nearest neighbour, nearest prototype, a rule-based classifier, and NB classification techniques were considered. Their study showed that most of the classification methods being compared gave similar results, with the exception of NB. Each method was tested against eight different fault types. The NB method gave the largest percentage of correct diagnoses for all but one fault type. The cooling unit fault had correct diagnoses of 96% using NB, while all of the other classification methods gave less then 80% correct diagnoses.

Choi et al. (2009) developed a novel Bayesian fusion classifier for FDD in automotive systems. Classifier fusion is the practice of combining multiple classifiers in an attempt to improve overall classification. They first isolated faults using various neural networks and nearest neighbour methods, and then fused these methods with class-specific Bayesian classifiers in order to classify different fault classes. A realistic automotive engine model was employed which simulates the behaviour of an engine in real time. The engine simulator tested eight fault cases: air flow sensor fault, leakage in air intake manifold, blockage of the air filter, throttle angle sensor fault, air/fuel ratio sensor fault, engine speed sensor fault, less fuel injection, and added friction. The simulation parameters were set at 2485 rpm and were sampled at a rate of 200 Hz for 2000 sample points per run. Their results showed that NB had a correct classification rate of 88.19% which outperformed nearest neighbour 87.06% and neural network 85.17% individual classifications. The novel Bayesian fusion classifier further improved the correct classification rate to 93.75%.

Liu and Chen (2009) developed a FDD method using modified Bayesian classification on a subspace obtained using principle component analysis (PCA). Their approach modifies Bayesian classification by first clustering data into groups and then classifying based on cluster center and covariance. It was applied to the Tennessee Eastman chemical process to demonstrate the effectiveness of the fault detection. In this study the training data sets of normal operating condition as well as overall cooling water flow faults, motor power faults, and inlet guide vane faults were used. The data set contained 4,213 observations for monitoring the motor power at two different classes. The Bayesian clustering method gave a 96.6% correct classification rate for the motor power fault. The other fault cases seemed to also give good results based on clustering, however exact classification rates were not given.

2.3.2. Artificial Neural Network

An artificial neural network (ANN) is a mathematical model which acquires its name due to its similarity to biological neural networks. ANNs consist of numerous processing elements or nodes connected together to form a network. Generally, each node consists of a simple nonlinear processing element. ANNs are a commonly used method for classification.

The FDD system from Wu and Liu (2009) implemented ANNs. A generalized regression neural network (GRNN) was used in comparison with a conventional back-propagation network (BPN). Their study looked into the following six different operating cases of an internal combustion engine: without fault, air leakage, cam-shaft sensor fault, electronic control thermal sensor fault, one-cylinder miss-fire, and two-cylinder miss-fire. The experimental results showed a classification accuracy of over 95% for various engine working conditions. When operating at idle engine speeds GRNN had classification accuracy of over 95% and an execution time of 4.86s, outperforming BPN which had classification accuracy of over 92.5% and an execution time of 13.42s.

Desbazeille et al. (2010) developed a FDD method for large diesel engines using ANNs for fuel leakage faults. ANNs were used for fault classification by looking at the angular speed waveform variations of the crankshaft. At each fault level, 100 healthy samples and 100 fault samples were simulated. 75% of the data were used for training, and the remaining 25% for testing. Their FDD method produced simulation results with greater than 90% classification accuracy for fault levels of 30% and greater. An experimental result was also obtained by decreasing the injected fuel mass by 80% on cylinder A6 and by 50% on cylinder A4. In this experiment, cylinder A6 was correctly classified as a fault, however A4 was not.

ANNs have advantages in that the can perform tasks which a linear classification method cannot. They have been shown to be successful in classification for complex systems and have learning capabilities. Some of the draw backs of ANNs are the large amount of training needed and the large processing time required. ANN training may also become stuck in a local minimum, causing erroneous results. ANNs are considered "black box" compared to "white box" statistical methods such as Bayesian classification or decision trees.

2.3.3. Decision Tree

A decision tree (DT) is a tree-like model of decisions and their possible consequences. DTs can be used as a predictive model to map observations of a known object to generate predictions of the object's target value. DTs are a commonly used method for classification and sometimes are named classification trees. DTs are relatively simple to implement, and easy to interpret.

Sun et al. (2006) developed a DT and PCA-based FDD method for rotating machinery. The Bentley RK4 rotor kit was used for testing a number of cases: normal operation, unbalance, rotor radial rub, oil whirl, shaft crack, and unbalance and radial rub in combination. PCA was used to reduce the number of features from 18 to 6. Then a DT model was trained for classification and diagnosis of faults. Fifty sample sets were taken for each fault case, 60% were used for training and 40% for testing. An average diagnosis accuracy of 98.3% was found, compared to 95.8% from the BPN used for comparison.

Saravanan and Ramachandran (2009) developed a FDD method for a spur bevel gear box. The vibration signal was analyzed using Daubechies wavelets for feature extraction. DT classification was then used for fault diagnosis. Four cases were examined: good gear, gear tooth breakage, gear with crack at root, and gear with face wear. The sampling frequency was 12,000 Hz with a sample length of 8,192 for all operating conditions. The maximum classification accuracy was found to be 98.7%.

DTs are generally simple to understand and interpret, and require minor data preparation. They have advantages of being quite robust and perform well with relatively small computation time with large data sets. Also DTs can be considered "white box" in comparison to some other classification methods. A disadvantage of DTs is that some DT algorithms cannot guarantee that the optimal tree will be generated.

2.3.4. Nearest Neighbour

K-nearest neighbour (k-NN) classification is a commonly used geometric classification method. A data point's class is determined from its distance to some reference points. The object is classified by the majority of its k nearest neighbour's classifications, where k is some positive integer. The nearest neighbour method is a simple and generally provides good results.

He and Wang (2007) developed a k-NN approach for fault detection of semiconductor manufacturing processes. They used an etching process industrial example to demonstrate the performance of their FDD method. The data set consisted of 107 normal wafers and 20 faulty wafers. Results showed k-NN to have a successful detection rate of 85%, a significant increase over the 55% achieved by the PCA method for this application.

Lai and Snider (2005) developed a high electrical impedance fault detection system. Their feature extraction method was based on DWT and RMS voltage and current values. Features were then applied to a k-NN classifier for pattern recognition. 95% of the RMS values were used for training while the remaining 5% were used for testing. In their study, 1000 non fault cases and 1000 fault cases were simulated. Their FDD method gave a successful classification rate of 98% for faults and 97% for non faults.

2.3.5. Linear Discriminant Analysis

Polynomial classification uses a functional approximation to determine the posterior probabilities of the classes. The simplest and most common form is linear discriminant analysis (LDA), commonly used in FDD applications. LDA is a traditional statistical methodology which utilizes analysis of variance and regression analysis to obtain classification.

Altman (1994) performed a comparison between LDA and neural networks for corporate distress diagnosis. Over 1000 healthy, vulnerable, and unsound industrial fiRMS were analyzed. Both types of diagnostic techniques displayed acceptable classification of over 90%. It was stated that although ANNs perform similarly or better than discriminant analysis in most cases, LDA was deemed to be the preferred method due to its simplicity and "white box" structure. The complexity as well as the training time needed for ANNs were listed as their disadvantages.

Chiang et al. (2000) investigated FDD for a chemical process, comparing Fisher discriminant analysis (FDA), discriminant partial least squares (DPLS), and PCA. Their research showed that FDA and DPLS were more proficient that PCA in diagnosing faults. The techniques were applied to simulated data collected from the Tennessee Eastman chemical process simulator. The data consisted of 500 observations for training and 960 observations for testing. There were 21 different simulated process fault cases. Their results showed that PCA techniques produced a 66.7% misclassification rate, DPLS produced a 57.4% misclassification rate, and FDA produced a 20.6% misclassification rate for their specific application.

LDA classification has the advantage of being quite simple to implement and understand. It also has seen extensive use and is one of the most time-tested classification methods. Some disadvantages of LDA are that its linearity and normal distribution assumptions may not be suitable for more complex applications.

2.4 Engine Valve Train Fault Detection

Although the literature on FDD methods is abundant, the literature on FDD of ICE valve trains is very limited. Lui et al. (2005) proposed a simple technique for engine valve fault detection by measuring vibration signals on the cylinder head using accelerometers. This vibration signal was caused by five main impact forces on each cylinder (corresponding to inlet valve opening, inlet valve closing, exhaust valve opening, exhaust valve closing, and combustion). A simple diagnostic technique named partial sampling and feature averaging (PSFA) was presented and used to detect abnormal valve clearances as well as gas leakage. Experiments were performed on a 4135D diesel engine at a sampling frequency of 50 kHz. Inlet valve clearances were set to 0.15, 0.3, 0.6, 0.9, 1.2, and 1.5mm on cylinder 1, where 0.3mm was the nominal case. It was found that both the abnormal valve clearance and gas leak faults could be detected by analyzing the peak impact values and crank angles at those peaks, however no classification rate results were given.

Li et al. (2010) developed a FDD system for abnormal clearance between contacting components in a diesel engine. Two accelerometers were mounted on the cylinder head to measure vibration signals. The vibration impact times and amplitudes were analyzed by re-sampling the time domain signal into the crank angle domain. Empirical mode decomposition was used to decompose the signal in order to minimize signal overlap and enhance the isolation of each event during combustion. Their experiments tested a normal clearance of 0.40mm, a tight clearance of 0.25mm, and an excessive clearance of 0.70mm. Their method was used to successfully detect abnormal valve clearance faults as well as worn piston ring faults. They found that a normal clearance had an average intake valve close at 583.3º, tight clearance at 604.5 º, and excessive clearance at 570.2º in the crank angle domain. No classification rate results were given.

Elamin et al. (2010) proposed a diesel engine valve clearance detection method using acoustic emissions (AE). The AE sensors were mounted of the front and rear sides of the cylinder head to acquire the signals. Nominal valve clearance was 0.38mm, faults were introduced by increasing valve clearance by 0.5mm and 0.8mm in the exhaust valve of cylinder one. The engine was tested at 1000 and 2000 rpm with each signal containing 40000 samples per engine cycle. The 0.8mm valve fault was found to open 46º later and close 25º earlier than the normal case. Their results found that valve clearance faults can indeed be detected by analyzing the magnitude and phase of valve impacts from the AE signals and FFT analysis. They concluded that acoustic emission is a powerful and reliable method of detection and diagnosis for faults in diesel engines, specifically shown for valve faults. However, no specific classification success results were given.

Shen et al. (2000) developed a fault detection system for a diesel engine using FFTs and rough sets theory. The engine valve faults studied were: small intake valve clearance, large intake valve clearance, and large exhaust valve clearance. Vibration signals were collected and analyzed to detect and diagnose these faults. Both the time and frequency domain of the vibration signal were used in developing this system. The focus of this paper was to develop a new method to discretize attributes extracted from the vibration signals. They stated that their system was “effective”, however they did not clearly quantify its detection accuracy or the levels of the faults they were investigating.

2.5 Summary

In this chapter, previous relevant research on FDD was reviewed. Signal processing, signal-based classification and model-based classification algorithms were reviewed. The amount of literature specifically related to valve train FDD of ICE is quite sparse. The most relevant publications were by (Liu, 2005), (Yujun Li, 2010), (Elamin, 2010), and (Shen, 2000) who all used cylinder head vibration signals to detect abnormal valve clearances.

The review of previous research using NB classification showed that this method provides accurate and reliable results, while being a relatively simple classification method. ANNs have been known to be a strong classification method for more complex applications. ANNs are generally more complex than other classification methods and may take more execution time for training. DTs are commonly used for classification and have been found to give reliable results for many applications, they are fairly easy to implement and interpret. The k-NN method was found to be an acceptable classification method for most applications and is also relatively simple to understand and implement. LDA is a simple classification method which has also been shown to give reliable results for many applications.

Although some literature exists for ICE valve train faults, the majority of this research is focused on valve clearance faults. There was no relevant literature found for detecting faulty valve springs. Simultaneous faults on separate cylinders have also not been studied, with the exception of Desbazeille et al. (2010) whose method failed to detect these simultaneous faults. Furthermore, simultaneous faults on the same cylinder have not been studied in any of the literature found. This thesis will look into both of these simultaneous fault cases. Additionally, it was found that the majority of the ICE valve train FDD research did not implement a classification method or did not present adequate classification results.

This thesis will develop a FDD technique for valve train faults by combining signal processing methods and classification algorithms using vibration signals. The focus will be on the NB method due to its relative simplicity and high classification efficiency with FDD systems; however other classification methods will also be studied for comparison.

CHAPTER 3: Experimental Test Bed

3.1 Introduction

In this chapter, the experimental test bed will be discussed. The instrumented engine will be described, followed by a description of the DAQ system including all sensors, conditioning circuitry, DAQ hardware, and DAQ software used. Next, an overview of the engine valve train will be given. Valve springs and valve clearances will be discussed, as well as potential valve faults and their mechanical tolerances.

3.2 Instrumented Engine

The engine used for the experiments consists of a modified Kubota Z482-E diesel engine. This is a vertical two cylinder, water cooled, 4-stroke indirect injection (IDI) diesel engine. The total displacement is 479.0cm^3 with a compression ratio of 23:1. It has a continuous power output of 8.05kW (10.8HP) at 3600rpm with a maximum speed of 3800rpm. A full list of specifications can be found in Table 3-1 (Kubota, 2008).

The Kubota engine is connected to an electric generator. This electric generator is connected to the generator test set which is used to draw a certain load from the generator and ultimately from the engine. The generator is a Markon BL105E. A full list of specifications can be found in Table 3-2 (Cummins Generator Technologies Inc., 2012).

Model	Z482-E
Type	Vertical, water-cooled, 4-cycle diesel engine
Number of cylinders	2
Bore and stroke (mm)	67 x 68
Total displacement (L)	0.479
Combustion chamber type	Spherical type
SAE NET Intermittent power (kW)	9.32 @ 3600 rpm
SAE NET Continuous (kW/rpm)	8.05 @ 3600 rpm
Maximum speed (rpm)	3800
Maximum idling speed (rpm)	800 to 900
Order of firing	1-2
Direction of rotation	Counter-clockwise (viewed from flywheel side)
Injection pump	Bosch MD Type mini pump
Injection pressure	140 kgf/cm^3 (13.73 MPa, 1991 psi)
Injection timing (Before T.D.C)	21^o
Compression ratio	23:1
Fuel	Diesel Fuel No.2-D
Lubricant	Oil, above CC grade
Dimension (length x width x height) (mm)	351 x 389 x 520
Dry weight (kg)	53.1
Starting system	Cell starter (with glow plug)
Starting motor	12 V 0.8 kW
Charging generator	12 V 150 W
Recommended battery capacity	12 V, 28 AH

Table 3-1: Engine specifications

Model	BL 105E
Type	Single phase 4-wire alternator
Voltage	Series connection 110-120V or 220-240V
Design	Brushless self-exciting
Apparent Power	6.0 kVA
Power	6.0 kW
Weight	26.5 kg

Table 3-2: Electric generator specifications

The generator test set is a Sotcher Measurement Inc. model 627. It is a load bank that imitates a real load that the generator would see in normal operation. It is used to put dummy loads on the generator so the engine can be tested under load conditions. A full list of specifications can be found in Table 3-3 (Sotcher Measurement Inc., 2013).

Model	627
Protection	Circuit breaker protection provided
Input Voltage	120/240V, 50-70Hz, single phase
Load Capacity	0 – 10.8kW
Duty Cycle	Continuous
Power Factor	Unity
Accuracies	
Frequency Meter	+/- 3% Center
Volt Meter	+/- 3% FS
Ammeter	+/- 3% FS
Weight	16 kg
Dimensions	500mm high, 400 mm wide, 260mm deep

Table 3-3: Generator test set specifications

The instrumented engine is shown in Figure 3-1. The figure illustrates the Kubota engine connected to the electric generator, and the DAQ system. The sensors are not easily visible in this figure due to their location and small size. Specific sensors will be discussed in the next section.

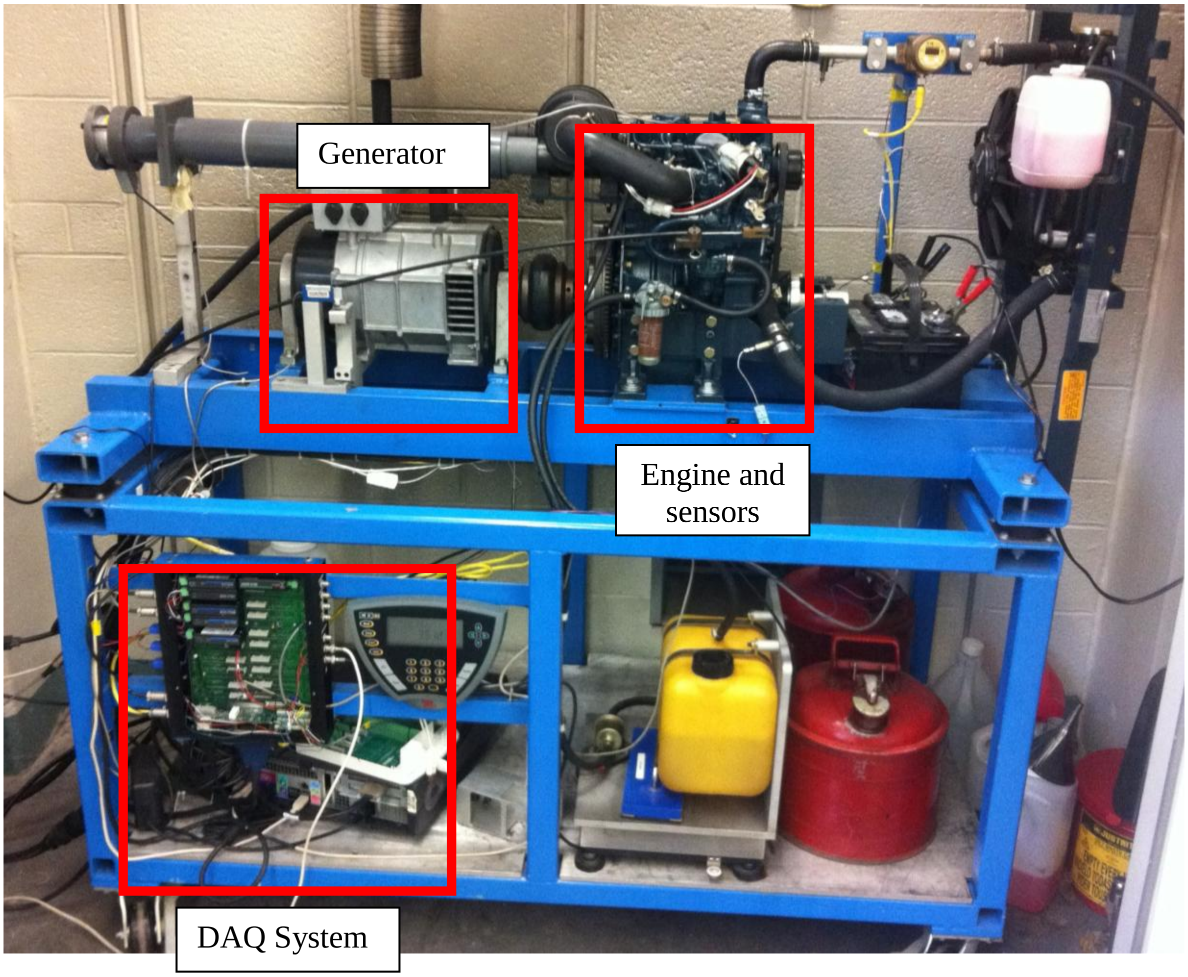

Figure 3-1: Experimental test bed (computer not shown)

3.3 Data Acquisition System

3.3.1 Data Acquisition Hardware

The DAQ hardware consists of a National Instruments (NI) PCIe-6353 card and a desktop computer. This DAQ card has 32 analog inputs with a sampling rate of 1.25 MS/s, 48 digital input/output lines, and four 32-bit counter/timers. A complete list of the NI PCIe-6353 specifications is given in Table 3-4 (National Instruments, 2010).

Product Name	PCIe-6353
Product Family	Multifunction Data Acquisition
Form Factor	PCI Express
Analog Input	
Input Channels (single-ended, differential)	32, 16
Resolution	16 bits
Sample Rate	1.25 MS/s
Throughput (All Channels)	1 MS/s
Simultaneous Sampling	No
Analog Output	
Output Channels	4
Resolution	16 bits
Update Rate	2.85 MS/s
Digital I/O	
Bidirectional Channels	48
Clocked Lines	32
Max Clock Rate	10 MHz
Counter/Timers	
Counters	4
Number of DMA Channels	8
Resolution	32 bits

Table 3-4: Data acquisition card specifications

Signal connections were accomplished using a NI SCC-68 terminal block. The terminal block contains 68 screw terminals for I/O connections, 4 expansion slots for signal conditioning modules, and a general purpose breadboard for custom circuitry. The complete list of NI SCC-68 specifications is given in Table 3-5 (National Instruments, 2006).

Product Name	SCC-68
Product Family	Terminal Block
I/O Connectors	One 68-pin male SCSI connector
Max Working Voltage (Channel-to-earth)	11 VDC
Operating temperature	0 to 55 ^{o}C

Table 3-5: Terminal block specifications

The computer used with the DAQ system for signal processing and analysis was a Windows PC with 8GB of RAM, and a 3.10 GHz Intel i5 processor. The complete list of the computer specs is given in Table 3-6.

Operating System	Windows 7 Professional 64-bit
Processor	Intel Core i5-2400
Number of Cores	4
Number of Threads	4
Clock Speed	3.1 GHz
Intel Smart Cache	6 MB
Memory	8 GB RAM
Hard Drive Model	ST1000DM003-9YN162
Capacity	1 TB
Cache	64 MB
Speed	7200 RPM

Table 3-6: Computer specifications

Connected to the DAQ hardware are a number of analog and digital sensors used to measure the physical phenomena of the mechanical system. A rotary BEI optical encoder was coupled to the crankshaft to measure the crank-angle of the engine. It has a resolution of 1440 counts/rev. A pressure sensing glow plug adapter PSIglow from OPTRAND was used. The pressure sensor has replaced the glow plug on cylinder 1, and reads the absolute pressure of that cylinder. The pressure sensor has a range of 0-200bar and total accuracy of 1% (Optrand, 2013). A PCB 353B18 piezoelectric accelerometer was mounted on the cylinder head to measure vibration signals. The accelerometer has a measurement range of ±500g and a resolution of 0.005g. A complete list of the accelerometer specifications are given in Table 3-7 (PCB Piezotronics, 2011).

Model	353B18
Sensitivity (±**10**%)	10 mV/g
Measurement Range	±500g pk
Frequency Range	1 to 10000 Hz
Resonant Frequency	≥70 kHz
Broadband Resolution (1 to 10000 Hz)	0.005g RMS
Non-Linearity	≤1%
Transverse Sensitivity	≤5%
Overload Limit	±10000g pk
Temperature Range	-65 to + 250^{o} F
Base Strain Sensitivity	≤0.003g/με
Excitation Voltage	18 to 30 VDC
Constant Current Excitation	2 to 20 mA
Output Impedance	≤100 ohm
Output Bias Voltage	8 to 12 VDC
Discharge Time Constant	0.5 to 2.0 sec
Spectral Noise	64μg/$\sqrt{Hz}$

Table 3-7: Accelerometer specifications

Figure 3-2 displays a close-up of the cylinder head with the head cover removed. The valve train as well as both the accelerometer and pressure sensor can be seen clearly. The accelerometer is mounted directly to the engine's cylinder head based on the literature review (e.g. Liu (2005); Li (2010); and Elamin (2010).). The exact location of the accelerometer on the cylinder head is mostly due to the space available.

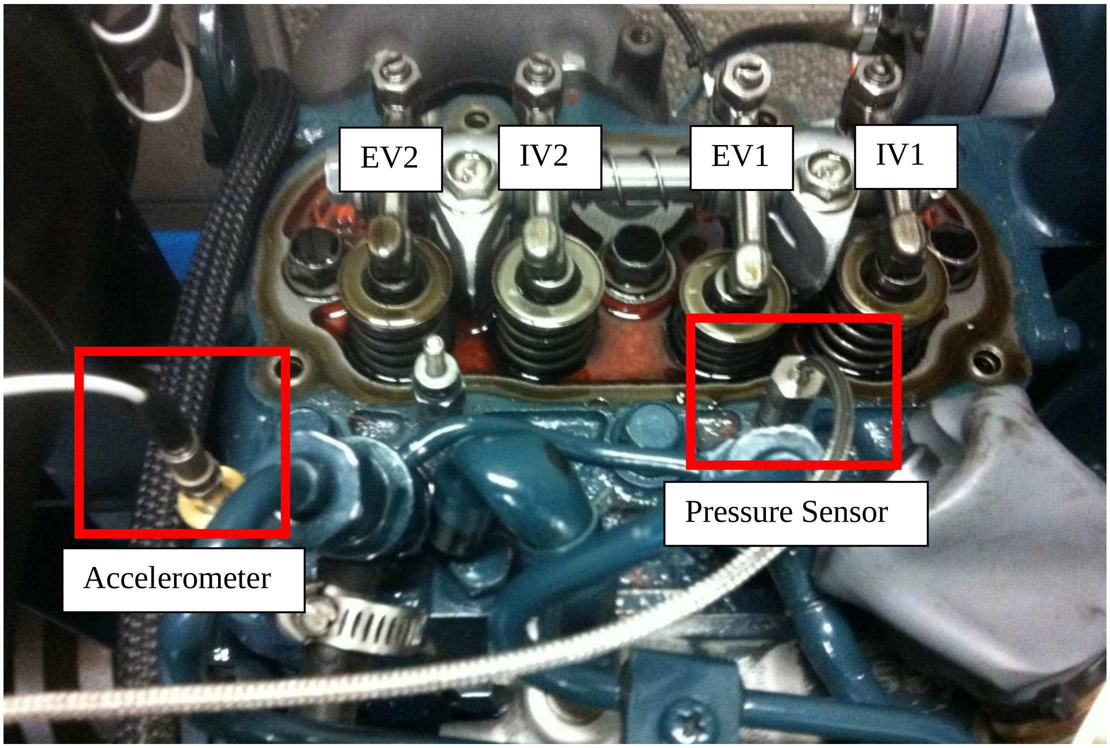

Figure 3-2: Accelerometer and pressure sensor

The four valves of the engine can also be seen clearly. Labels IV1 and EV1 indicate intake valve and exhaust valve of cylinder one, respectively. IV2 and EV2 are the intake and exhaust valves of cylinder two. This naming convention will be used throughout the thesis.

3.4.2 Data Acquisition Software

A program written using NI LabVIEW (National Instruments Corporation, 2013) was used for data acquisition, and to implement some signal processing and analysis techniques. The program was originally written by Darren Van Rooyen and modified by the author for this research. LabVIEW utilizes a visual programming language which is used widely in industry when data acquisition and signal processing are required. NI LabVIEW 8.5.1 was used earlier on in the research, and saw continued use for some of the data acquisition components. Mid-way through the research, updated software was purchased and LabVIEW 10.0 was used for the more critical components. The specific signal processing methods implemented using this software will be further discussed in Chapter Chapter 4:.

Figure 3-3 shows a simplified block diagram of the experimental test bed. It outlines the main components of the system and illustrates how these components interact with each other.

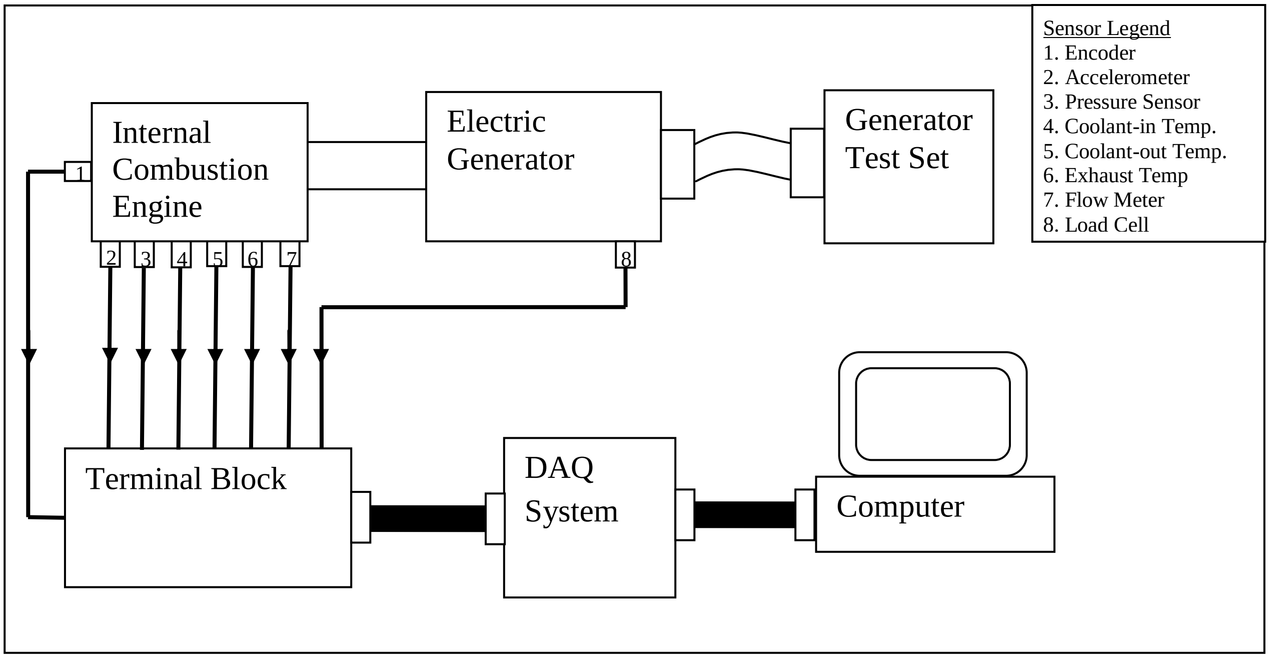

Figure 3-3: DAQ system block diagram

This figure can be broken down into two separate sections: the electro-mechanical system, and the instrumentation. The electro-mechanical system consists of the ICE, the electric generator, and the generator test set. The engine is coupled to the electric generator via a flexible shaft coupler. This electric generator is then connected to the generator test set.

The instrumentation consists of the sensors, terminal block, the DAQ system (signal conditioning circuitry, ADC's, DAC's, etc.), and finally the computer which the data is loaded to and analyzed. The figure shows the flow of information from the physical plant, transformed by sensors and connected to the terminal block. The terminal block receives all sensor data and routes this information to the DAQ. The sampled digital signals are sent to the computer where further signal processing and analysis is performed.

3.4 Engine Valve Train

3.4.1 Valve Trains – Introduction & Background Information

An engine valve train is the device that controls the operation of the valves. These usually consist of at least one intake valve and one exhaust valve for each cylinder in an internal combustion engine. The valve train controls the air flow and fuel flow in and out of the combustion chambers. Common valve trains consist of valves, rocker arms, lash adjusters, pushrods, lifters, camshafts, and valve springs.

A number of different valve train configurations exist, however each layout performs the same basic function of controlling the valves for proper engine operation. The most common valve train designs are cam-in-block and overhead camshaft.

Figure 3-4 shows the workings of a cam-in-block valve design (Automotive Engineering, 2013). In this design, the camshaft is located somewhere within the engine block. The camshaft is affixed to the pushrod, which is actuated as the camshaft rotates and the cam lobe is engaged. The pushrod then actuates the rocker arm which rocks about a pivot point, and compresses the valve spring. This allows the valve to open according to the cam lobe design. Then the camshaft continues to rotate which disengages the cam lobe and the valve spring forces the valve closed. The Kubota Z482-E engine used for this research has a cam-in-block type valve train.

The overhead cam design can be seen in Figure 3-5 (Automotive Engineering, 2013). This design locates the camshaft above the valve; however it serves the same purpose and works similarly to the cam-in-block design. The camshaft is affixed to a camshaft follower, which then compresses the valve spring due to the cam lobe design. As the camshaft rotates, the valve will be forced open by the lobe, and closed by the valve spring. It is important to note that although these two designs are different, both implement a valve spring in the same manner. Thus, valve spring fault detection is important for both valve train designs.

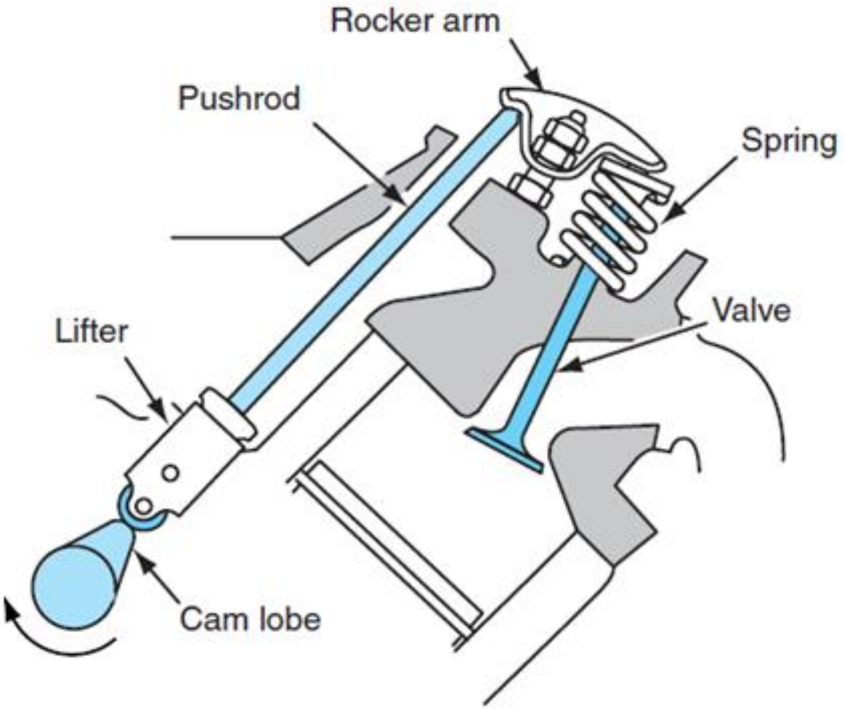

Figure 3-4: Cam-in-block valve design (Automotive Engineering, 2013)

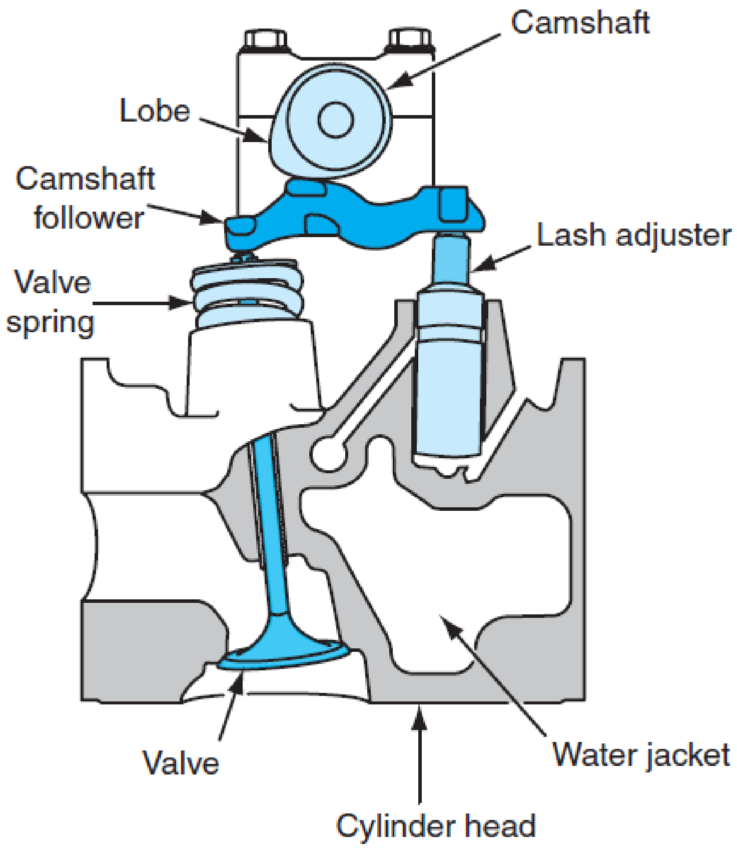

Figure 3-5: Overhead cam design (Automotive Engineering, 2013)

3.4.2 Valve Springs

Valve springs deteriorate and deform gradually over time and extended use. Gradual deformation of a material due to mechanical stresses is known as creep. Creep is exaggerated when the material is subject to high temperature. Creep has been observed in the case of valve springs even at moderate temperatures (Taylor, 1985). A deformed spring results in undesired conditions known as "valve float" and "valve bounce" which reduces engine efficiency and performance due to improper valve closing and sealing. These conditions tend to worsen as engine RPM increases. Deformed valve springs are often difficult to detect and are commonly misdiagnosed as fuel injection or ignition problems. A deformed spring if not detected and properly repaired may lead to spring failure, which could result in catastrophic failure of the engine (Taylor, 1985). Examples are shown in Figure 3-6 (LS1Tech, 2013).

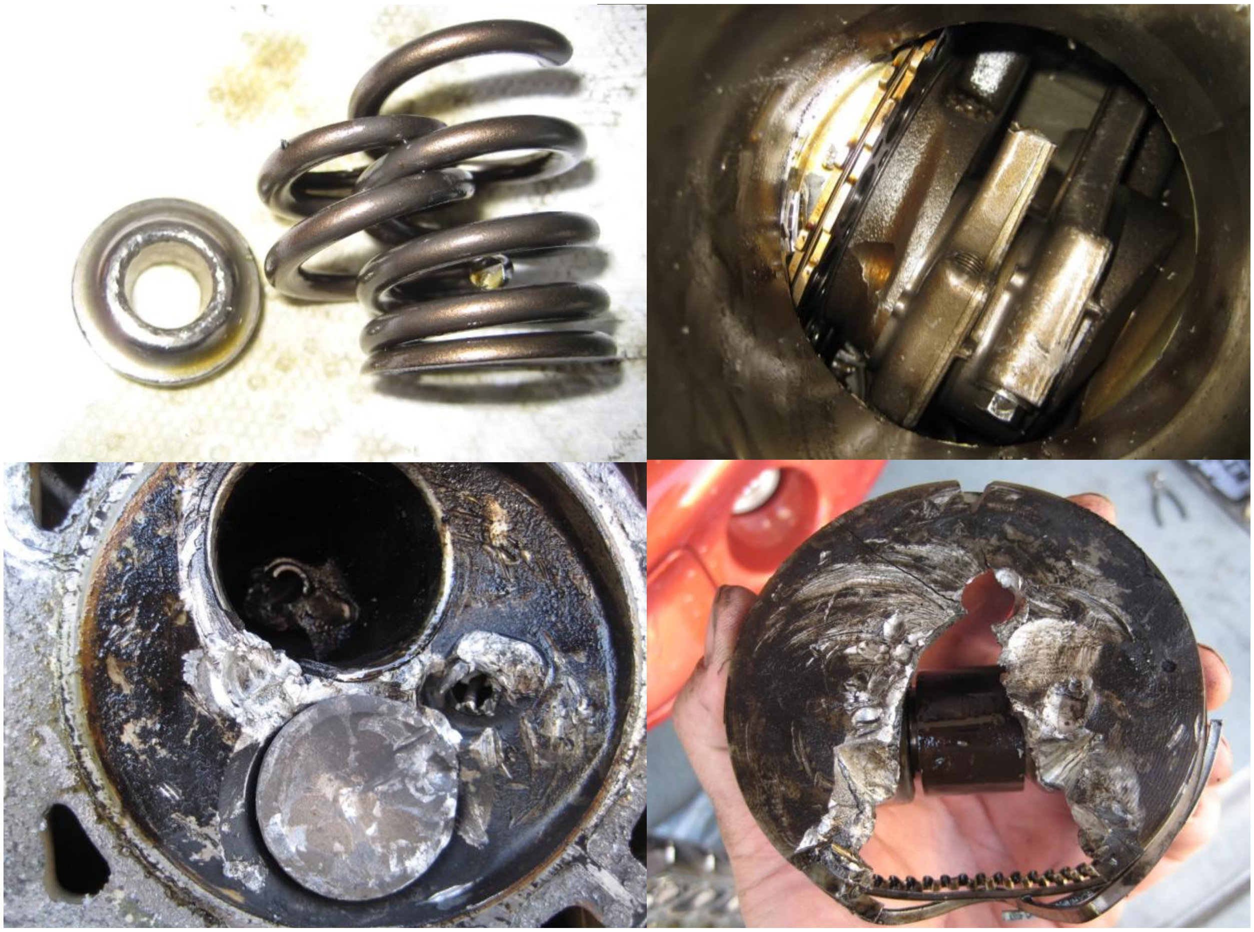

Figure 3-6: Catastrophic engine failure (Top left: Broken spring, Top Right: Broken con-rod, Bottom left: Damaged cylinder head, Bottom Right: Damaged piston)

An engine will specify the acceptable tolerances for the free length of its valve springs. If the valve spring free length goes below the manufacturer's specified threshold, the engine efficiency and performance will decrease, and the possibility of failure will increase. Thus detecting deformed valve springs near or prior to this threshold is desired. The valve spring specifications for the Kubota Z482-E engine used in this thesis are given in Table 3-8 (Kubota, 2008).

Tilt (A)	Allowable limit	1.2 mm

Free length (B)	Nominal size	31.3 to 31.8 mm
	Lower limit	28.4 mm

Table 3-8: Valve spring specification limits (Kubota, 2008)

Figure 3-7 (Kubota, 2008) illustrates both the tilt and free length of a compression valve spring. Where (A) shows the spring tilt and (B) shows the free length of the spring.

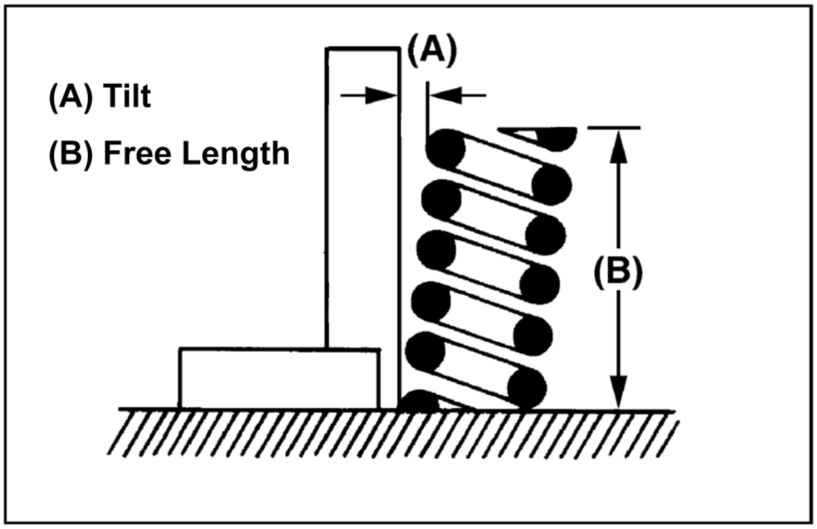

Figure 3-7: Free length and tilt of a valvespring (Kubota, 2008)

In our research we utilized grinding techniques to seed a deformed valve spring fault by reducing its free length by 0.5mm. A faulty spring can be installed on one or both the intake and exhaust valves of each cylinder, giving up to 16 separate fault cases. The fault detection and isolation method developed in this thesis should be able to successfully detect these faults. The term spring fault (SF) will be used throughout this thesis.

3.4.3 Valve Clearance

Valve clearance typically refers to the clearance between the rocker and valve cap. Each ICE has a nominal valve clearance value designed to allow for thermal expansion. An abnormal valve clearance will result in improper valve closing and timing, degrading engine performance. Figure 3-8 gives an illustration of valve clearance.

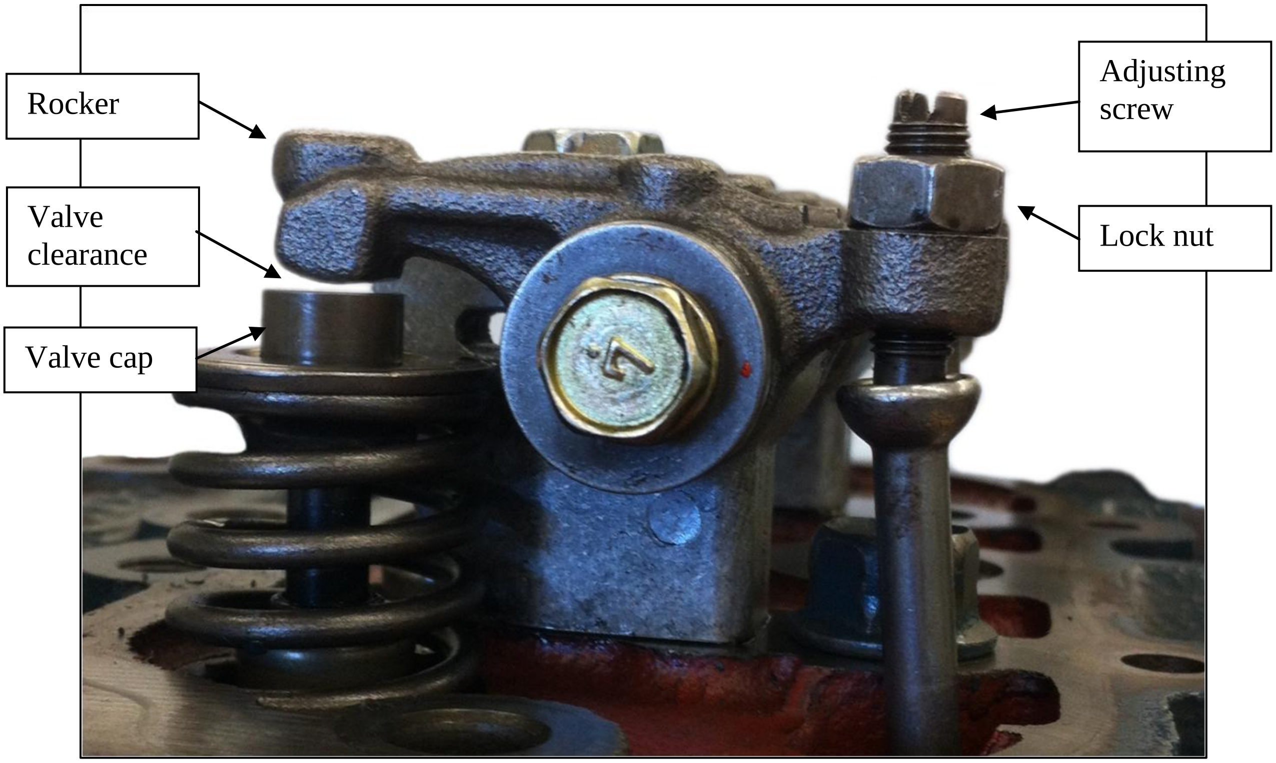

Figure 3-8: Valve clearance

The nominal operating valve clearance specification for the Kubota Z482-E is given in Table 3-9. The manufacturer does not provide an upper or lower limit for the valve clearance. The chosen clearance fault specifications are also given in the table. These values will be used throughout the thesis for the valve clearance fault cases.

Valve Clearance Case	Specification
Small Clearance Fault (cold engine)	0.05 mm
Nominal spec. from manufacturer (cold engine)	0.15 mm
Large Clearance Fault (cold engine)	0.25 mm

Table 3-9: Valve clearance specifications

In our research we adjust the valve clearance with the adjusting screw (shown in Figure 3-8) to seed a valve clearance fault. The clearance faults were seeded on one or both the intake and exhaust valves of each cylinder. The FDD method developed in this thesis should be able to successfully detect these faults. The teRMS small clearance fault (SCF) and large clearance fault (LCF) will be used throughout this thesis.

3.5 Summary

In this chapter, the experimental test bed including the instrumented engine, generator, DAQ hardware, and DAQ software was described. Details and specifications of the sensors and all other pertinent hardware components were provided. The system to be monitored was described and specifications were given. The faults to be seeded were introduced and explained, including the manufacturers allowable limits for the specific component. The next chapter will discuss the signal processing methods used.

CHAPTER 4: Signal Processing Methods

4.1 Introduction

This chapter will describe the signal processing methods used to improve the quality of the signal information. Signal processing commonly consists of signal acquisition, signal improvement, and/or signal compression. These subjects will be discussed in this chapter. First, the method of signal acquisition used will be described including all relevant aspects of the DAQ system. Next signal improvement methods will be discussed, including filtering and denoising techniques. After the signal processing methods are performed, the feature extraction and fault detection methods are applied. These are described in the next chapter.

4.2 Signal Acquisition

Signal acquisition is the process of sampling a signal that describes a physical phenomenon. Sensor output signals are sampled and converted into digital values that can then be analyzed computationally (Nise, 2008). First an appropriate sampling rate will be determined and discussed, and then the data acquisition system utilized for this specific application will be described more thoroughly.

4.2.1 Sampling Rate for the Vibration Signal

Sampling is the process of representing a continuous signal as a discrete signal by taking a number of discrete samples of that continuous signal. The sampling rate or sampling frequency f_s is the number of samples taken per second.

The Nyquist-Shannon sampling theorem states that the sampling frequency of a signal must be greater than twice the maximum frequency of the signal being sampled f_b in order to perfectly reconstruct a signal. This is commonly written as:

$$f_s > 2 f_b \tag{4.1}$$

The Nyquist frequency is equal to two times the bandwidth of a signal:

$$f_N = 2 f_b \tag{4.2}$$

The Nyquist-Shannon sampling theorem states that the sampling frequency thus must be above the Nyquist frequency in order to avoid aliasing.

Unfortunately, an impact or step is not a band limited signal. As mentioned in the previous chapter, our FDD system will be based on the valve closing impacts. This means that our signal does not have a maximum frequency. In other words, the highest possible sampling rate will give the best signal representation with the least amount of aliasing. Of course, our sampling rate will be limited by many factors. This section will describe how the sampling rate used with our FDD system was determined.

First, our signal was sampled at an effectively "infinite" sampling rate of 1 MHz. This rate should show us our "true" signal or at least the best representation of it possible with our DAQ hardware limitations. The signal obtained by sampling at this rate can be seen in Figure 4-1.

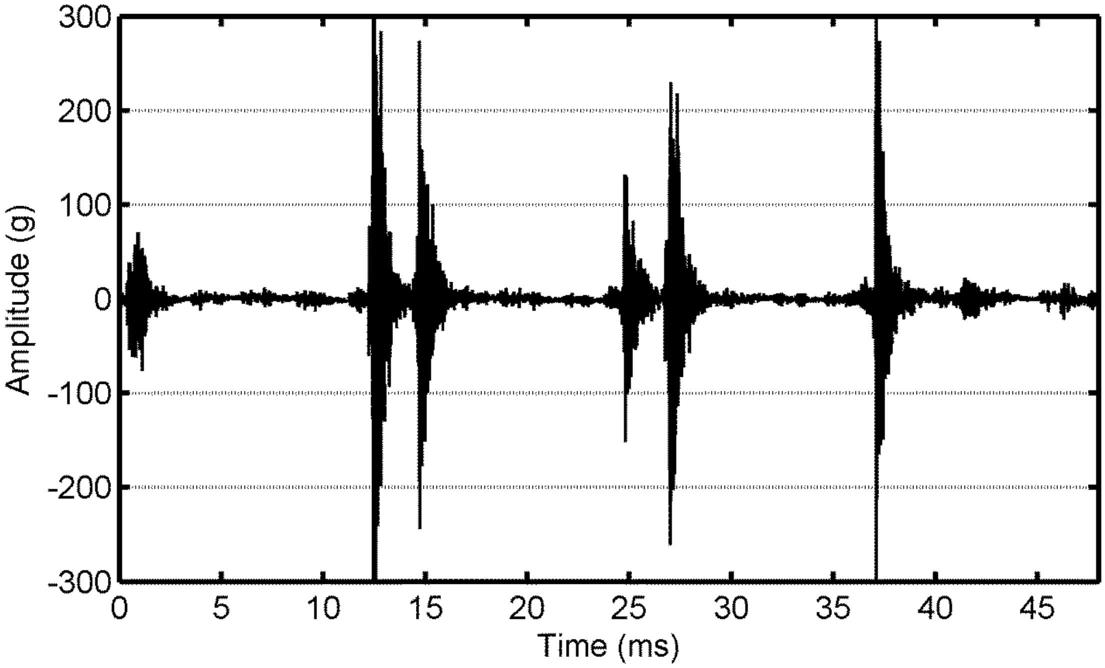

Figure 4-1: Vibration signal in time domain (sampling rate = 1MHz)

Such a large sampling rate is not reasonable due to hardware constraints, memory constraints, computing time constraints, etc. For this reason we must find a reasonable sampling rate, and compare it to our "true" signal to confirm it is appropriate.

By plotting the frequency response of the signal we can see which frequencies have the most importance in our signal. Figure 4-2 shows the frequency response when sampling at 1 MHz.

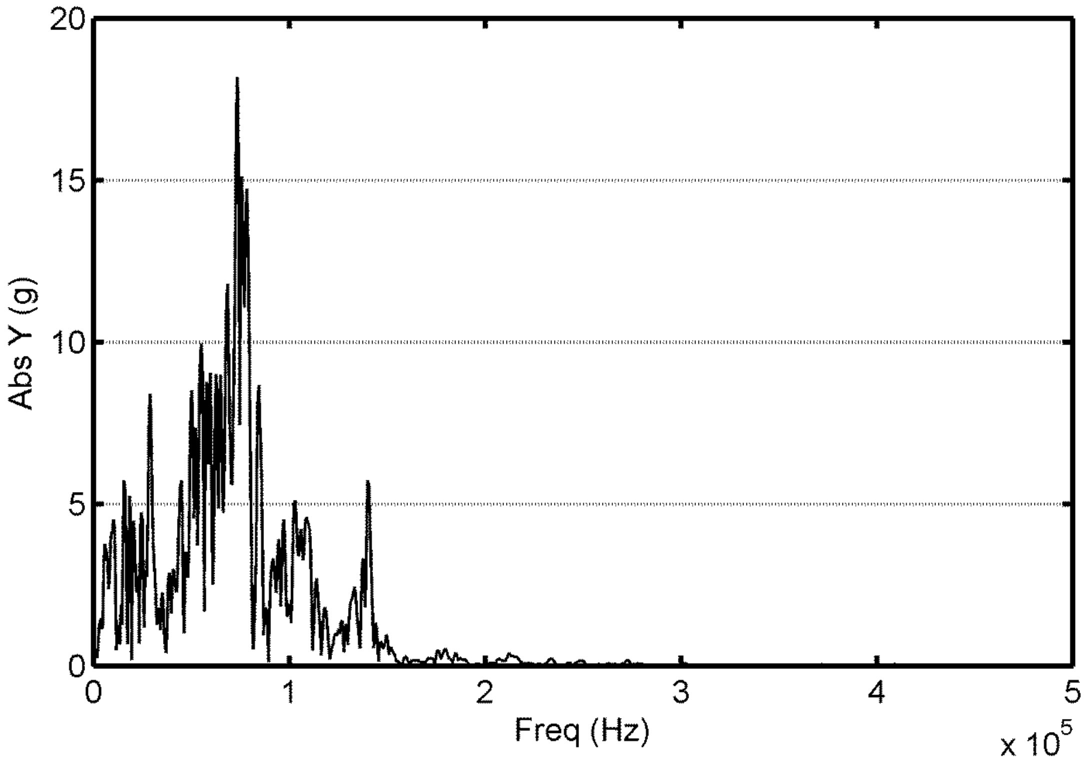

Figure 4-2: Vibration signal (sampling rate = 1MHz) in frequency domain obtained by FFT.

From this figure it can be seen that frequencies after 150 kHz are of little importance, so this shows that 1 MHz is indeed a large over-sampling. Also from our sensor specifications it is known that the accelerometer has a resonance frequency of 70 kHz, which agrees with the spike seen around 70 kHz and a 2nd resonance peak at 140 kHz. Due to these resonance peaks the data above ~60 kHz is unreliable.

There is another very important factor when selecting the sampling rate. Since the impacts correlate to the crank angle, it makes sense to make the sampling rate proportional to the speed of the crankshaft (i.e. the RPM). This is accomplished using the rotary encoder attached to the engine crankshaft via a flexible coupler. In addition, its index channel is used to determine the angle corresponding to the piston of cylinder 1 being at top dead centre. Sampling using this method produces a vibration signal in the crank angle domain rather than the time domain. The encoder has a resolution of 1440 pulses/revolution. This corresponds to an encoder pulse every 0.25° of crank angle. Each pulse triggers the acquisition of one sample. The sampling rate is given by the equation:

$$Sampling\ Rate = Encoder\ Resolution\,\frac{\mathrm{RPM}}{60} \tag{4.3}$$

Most of the tests will be done at an engine speed of about 2000 RPM. This results in a sampling rate of about 48 kHz. Figure 4-3 shows a vibration signal sampled at ~48 kHz.

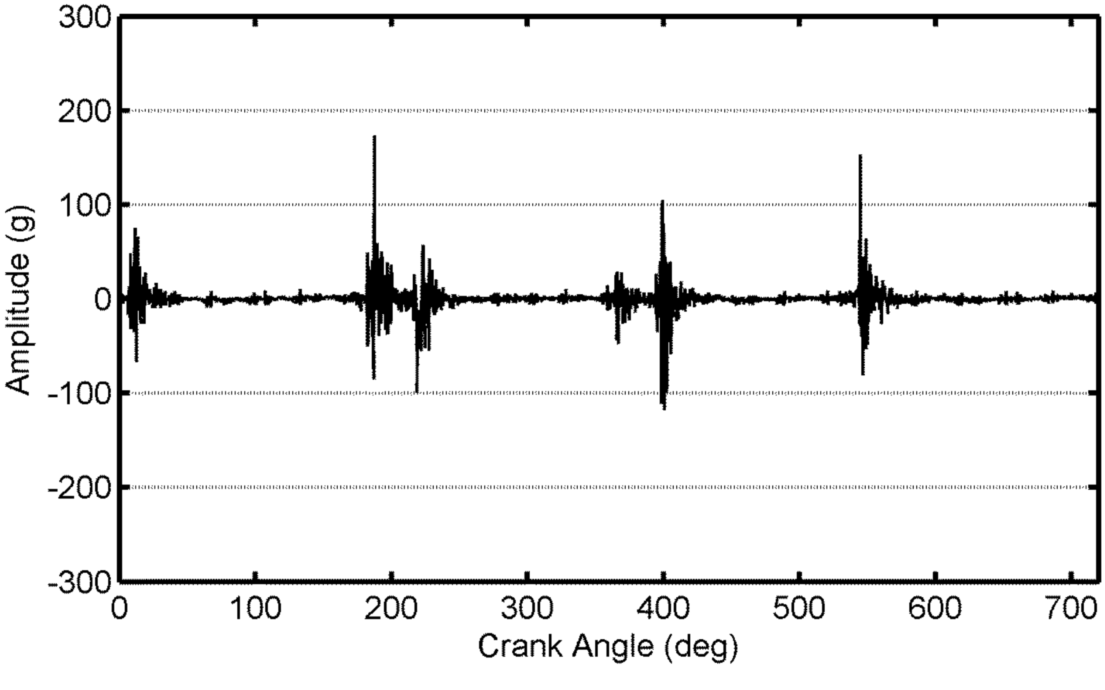

Figure 4-3: Vibration signal in the crank angle domain. The sampling rate is approximately 48 kHz.

From Figure 4-2, a sampling frequency of 48 kHz is less that the Nyquist rate so aliasing of the signal will occur. Aliasing distorts the frequency content of a signal. However, for this application the goal is FDD. As later results will show, with our method it is not necessary to accurately represent the engine vibration. If a more accurate vibration signal is needed in the future, faster sampling can be achieved by replacing the current encoder with a higher resolution model. Note that this could be an expensive option since the encoder will have to withstand the heat, dirt and vibration generated by the ICE.

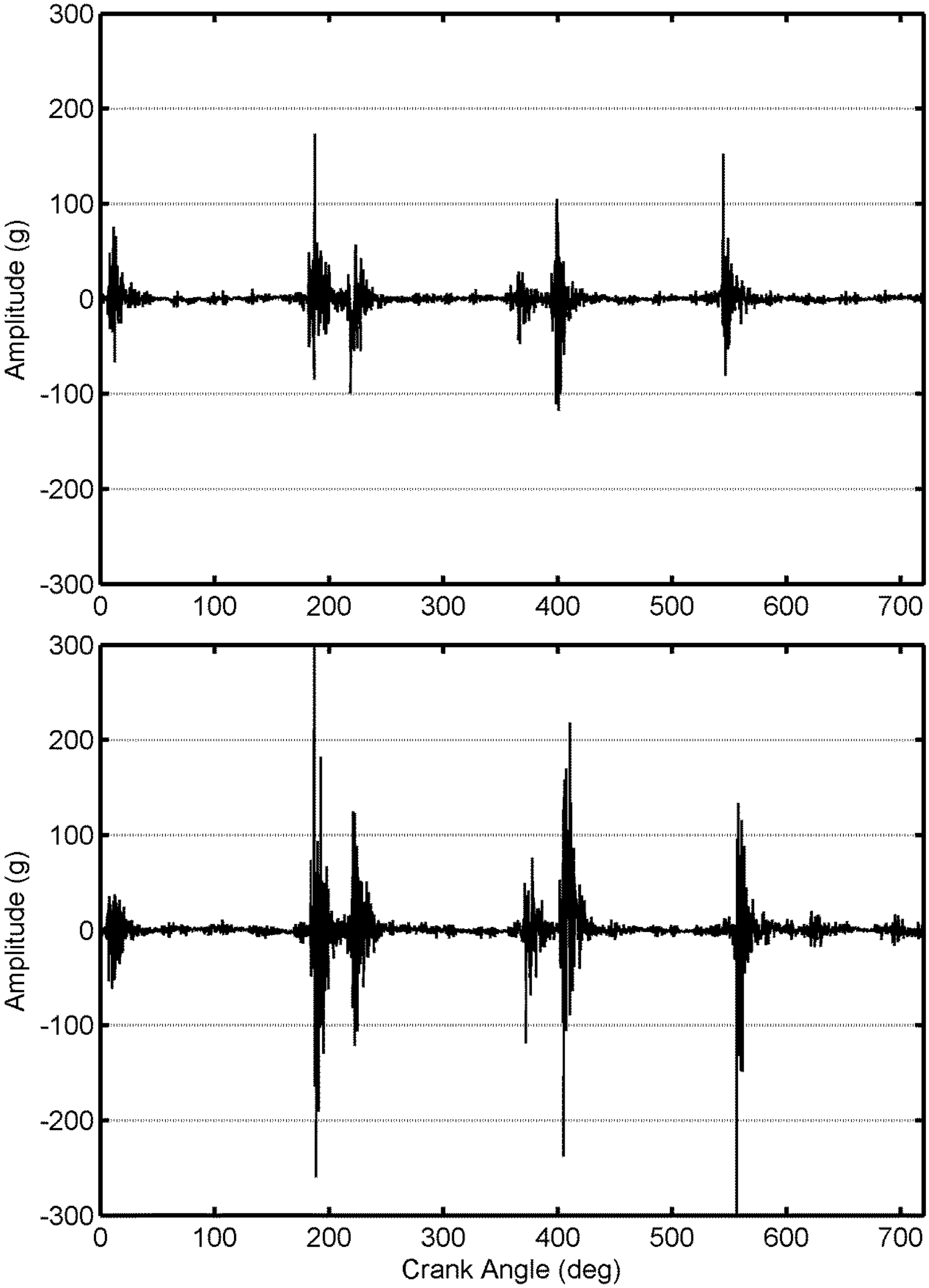

Figure 4-4: Vibration signal - 48 kHz (top) vs. 1 MHz (bottom)

4.2.2 Data Acquisition System

A brief description of the DAQ components and their specifications were outlined in the previous chapter. This section will give more specific details on how the pertinent signals are acquired.

As previously mentioned, the accelerometer is sampled in the crank angle domain at a frequency of about 48 kHz. A number of other sensors are sampled at the start and end of each test set, such as various engine temperatures and flow rates. These signals are sampled less frequently because the signal values do not change as frequently as the vibration and thus don't need as high of a sampling rate.

The data acquisition system was programmed to collect 500 engine cycles per data set. An engine cycle consists of two full crankshaft revolutions, thus 720° or 2880 encoder pulses. Figure 4-5 shows the general raw data vibration signal that is collected by the acquisition system for a single engine cycle.

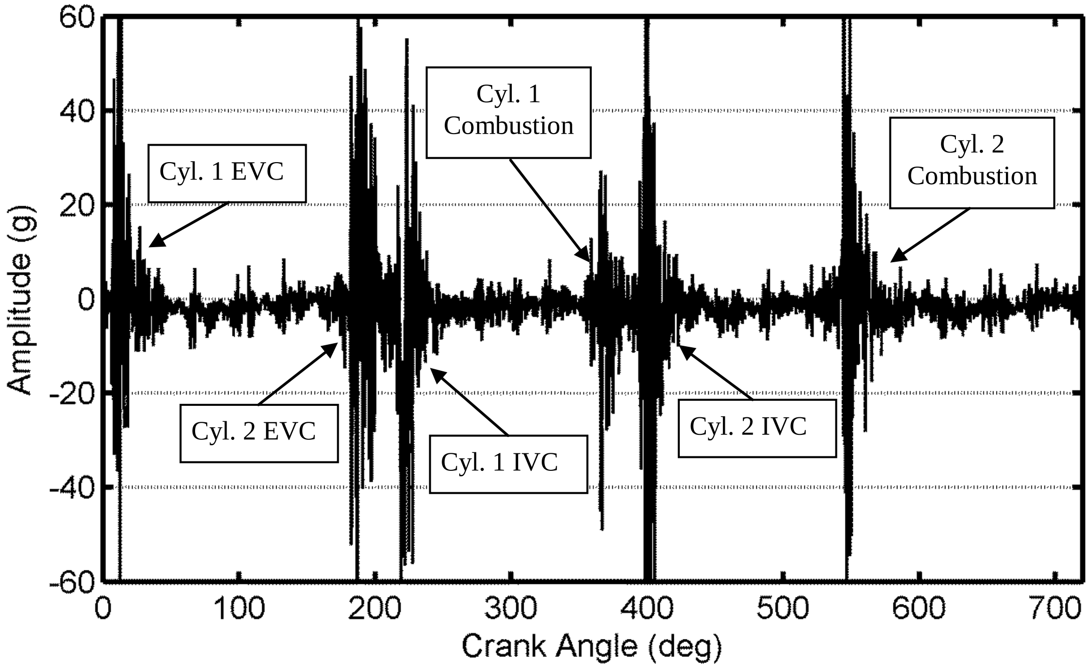

Figure 4-5: Vibration signal of a single engine cycle

Figure 4-5 shows the vibration amplitude acquired from the accelerometer plotted against the crank angle acquired from the rotary encoder. It shows the most important data collected by the DAQ system. From this data we can see six distinct impact points in a single cycle. These impact points represent valve closings and combustions of each cylinder as labelled in the figure. These impacts will be the distinguishing features of the signal which will be used to accomplish the FDD method to be discussed further in the next chapter.

In addition to the vibration signal, some initial and final temperature values are recorded for each data set collected. Initial and final fuel weights are also recorded for each data set, as well as the time and date of the recording. An example of these secondary data recordings can be seen in Table 4-1.

	Initial Value	**Final Value**
Date	12/10/2012	12/10/2012
Time	11:35 AM	11:36 AM
Engine Speed (RPM)	3523	3516
Generator Current (A)	10	10
Engine Load (N)	154	154
Coolant Temp. (°C)	72.9	81.1
Exhaust Temp. (°C)	326.4	327.9
Fuel Weight (kg)	3.362	3.352

Table 4-1: Secondary data recordings for a 3500RPM 10A load data set

4.3 Signal Improvement

Signal improvement is the process of enhancing the acquired signal for the given application. Noise reduction is a commonly used process in signal improvement methods. Some commonly used approaches towards noise reduction are averaging and filtering. These methods will be discussed and explored in relation to the acquired signal.

4.3.1 Averaging

First, an averaging of the vibration signals will be explored. As discussed in the previous section, each data set acquired contains 500 engine cycles. Combustion is known to be an inconsistent event. We do not want this inconsistency to be misinterpreted as a fault. Therefore averaging of a number of engine cycles to reduce the variability should be beneficial for FDD.

For our purpose this variations may be considered "noise". Averaging a small number of cycles will not produce much noise reduction. Averaging a large number of cycles will add a significant time delay to the fault detection. To find an "optimal" number of engine cycles to average, a comparison of various averaging values was performed. A signal-to-noise ratio (SNR) was calculated for a variety of averaging values and then compared in order to choose an appropriate averaging technique. A signal-to-noise ratio is commonly given an informal definition of the ratio of useful information to irrelevant information. In this application the useful information can be thought of as an impact point, and the irrelevant information can be a non-impact point. Thus we can define the SNR to be:

$$SNR = \frac{useful\,information}{irrelevant\,information} = \frac{V_{rms}(impact)}{V_{rms}(non-impact)} \tag{4.4}$$

where $V_{rms}(impact)$ is the vibration RMS value (Volts) in the vicinity of impact point, and $V_{rms}(non-impact)$ is the vibration RMS (Volts) value at the vicinity of a non-impact point. The vibration RMS value can be given by:

$$V_{rms} = \sqrt{\frac{1}{N_{snr}}\left(V_{acc}(1)^2 + V_{acc}(2)^2 + \cdots + V_{acc}(N_{snr})^2\right)} \tag{4.5}$$

where V_{acc} is the vibration value of the accelerometer in Volts, and N_{snr} is the number of values used in the RMS calculation (here $N_{snr} = 80$ was used). Using (4.4) and (4.5) the signal-to-noise ratio can be found for a number of different averaging techniques. Table 4-2 gives a comparison of the SNR values found.

# of Engine Cycles Averaged	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
No Averaging	0.698	0.0430	16.2
20 cycles	0.384	0.0226	17.1
40 cycles	0.365	0.0200	18.3
60 cycles	0.363	0.0192	18.9
80 cycles	0.364	0.0190	19.2

Table 4-2: Signal-to-noise values for changing averages

From Table 4-2 we can see that averaging does in fact reduce the noise, however it also reduces the impact signal that we wish to analyze. Thus the signal-to-noise ratio can be used to better understand the effects of averaging. The general trend shows that as the number of engine cycles averaged increases, both noise and relevant signal are reduced. It can also be seen that the signal-to-noise ratio increases as averaging increases. The table shows that SNR improvement tends to become marginal after about 60 cycles, thus an average of 60 engine cycles was determined to be the appropriate choice. Figure 4-6 illustrates how different averaging values affect the signal. These plots show that averaging reduces noise, however it also reduces the impact amplitudes. For example, the case of no averaging has the largest noise and the most pronounced impacts. Averaging was implemented in the FDD system as follows:

$$y(i,j)=\frac{1}{N_a}\left(V_{acc}(i,j)+V_{acc}(i,j+1)+\cdots+V_{acc}(i,j+N_a-1)\right)\;\text{ for } i=1 \text{ to } N_d \qquad (4.6)$$

where $V_{acc}(i,j)$ is the vibration value for the ith data point in the jth engine cycle, $N_a=60$ is the number of cycles averaged, and $N_d=2880$ is the number of data points in a single engine cycle.

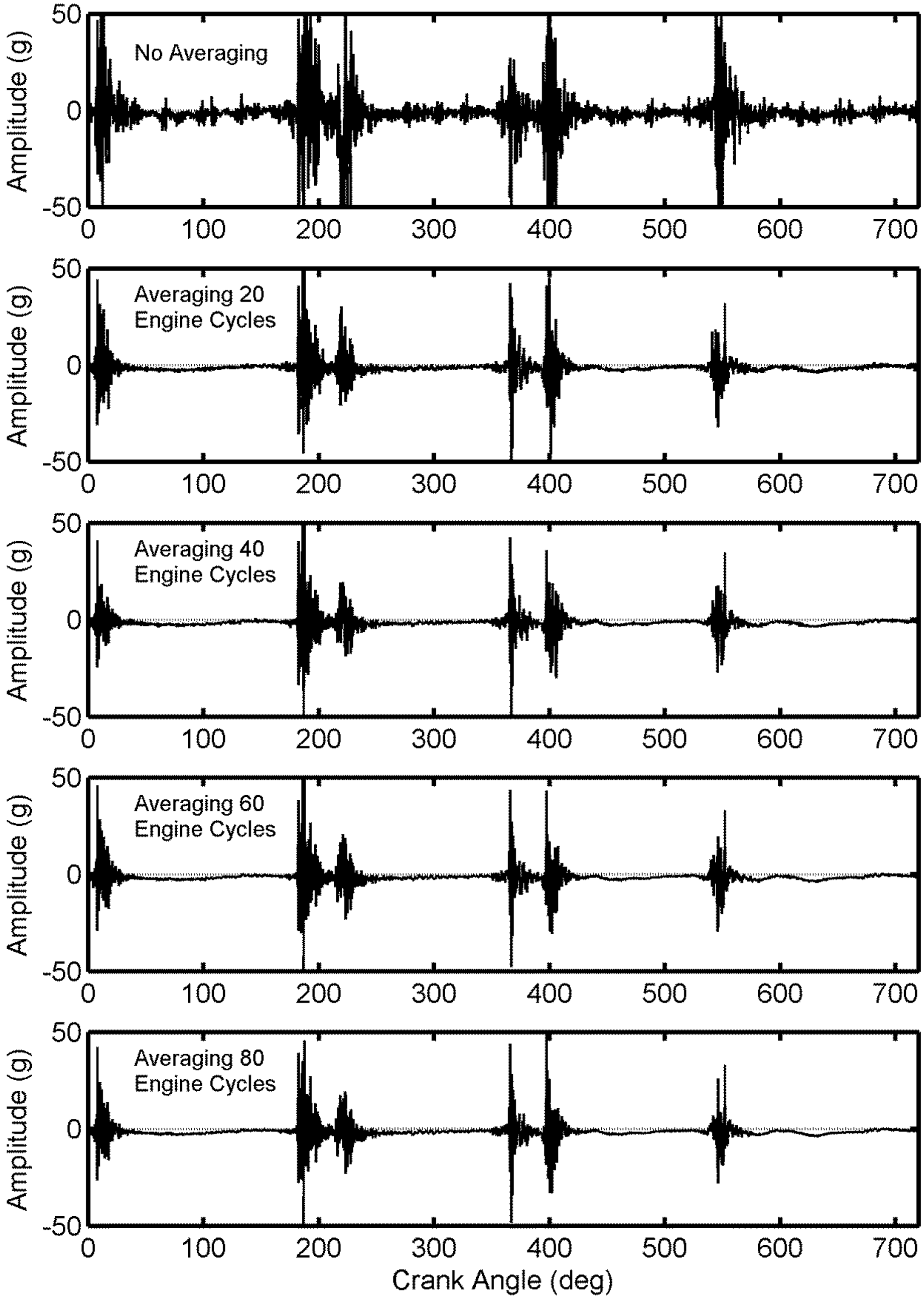

Figure 4-6: Effects of engine cycle averaging.

4.3.2 Filtering

Filtering is the process of removing some unwanted component or feature from a signal. This is often used to suppress unwanted noise. This makes filtering very useful and commonly used in signal processing. The negative aspect of filtering is the unintended loss of information that is associated with it. Thus similar to averaging, the objective of filtering becomes reducing unwanted noise while attempting to maintain the pertinent signal information.

4.3.2.1 Moving Average Filtering

The moving average filter is a filtering technique which smoothes data by replacing each data point by an average of the neighbouring data points within a given span. It has a similar objective to a low-pass filter, and has a similar effect on the corresponding data. This filter has the effect of smoothing; it is used for noise reduction however will also smooth the relevant data. Since this may not be desirable for our FDD, tests were done to assess its usefulness.

A variety of moving averaging spans were tested against the case of no moving average filtering. To implement a moving average filter each data point is replaced by the corresponding smoothed data point given by:

$$y_s(i) = \frac{1}{2N_s + 1}\left(y(i + N_s) + y(i + N_s - 1) + \cdots + y(i - N_s) \right) \tag{4.7}$$

where $y_s(i)$ is the smoothed value for the ith data point, N_s is the number of neighbouring data points on either side of $y_s(i)$, and $2N_s + 1$ is the span of the moving average. Again, a quantifiable comparison was done by using equations (4.4) and (4.5) to calculate the signal-to-noise ratio as described in Section 4.3.1. The table below shows a comparison of SNR values for a variety of different moving average filter spans.

Moving Average Span	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
0 (No filtering)	0.363	0.0192	18.9
3	0.155	0.0185	8.41
5	0.083	0.0184	4.52
7	0.070	0.0183	3.80
9	0.053	0.0183	2.92

Table 4-3: SNR comparison for various moving average spans

From this table we can see that the moving span average filter does in fact reduce the noise, however it also reduces the impact signal. Again we use the signal-to-noise ratio to better understand the effects of this filtering. The general trend shows that as the smoothing is increased both noise and relevant signal are reduced. It can also be seen that the signal-to-noise ratio decreases as smoothing increases. For this reason moving average filtering will not be used for this application. A more visual understanding of the moving average filter effect can be seen in Figure 4-7.

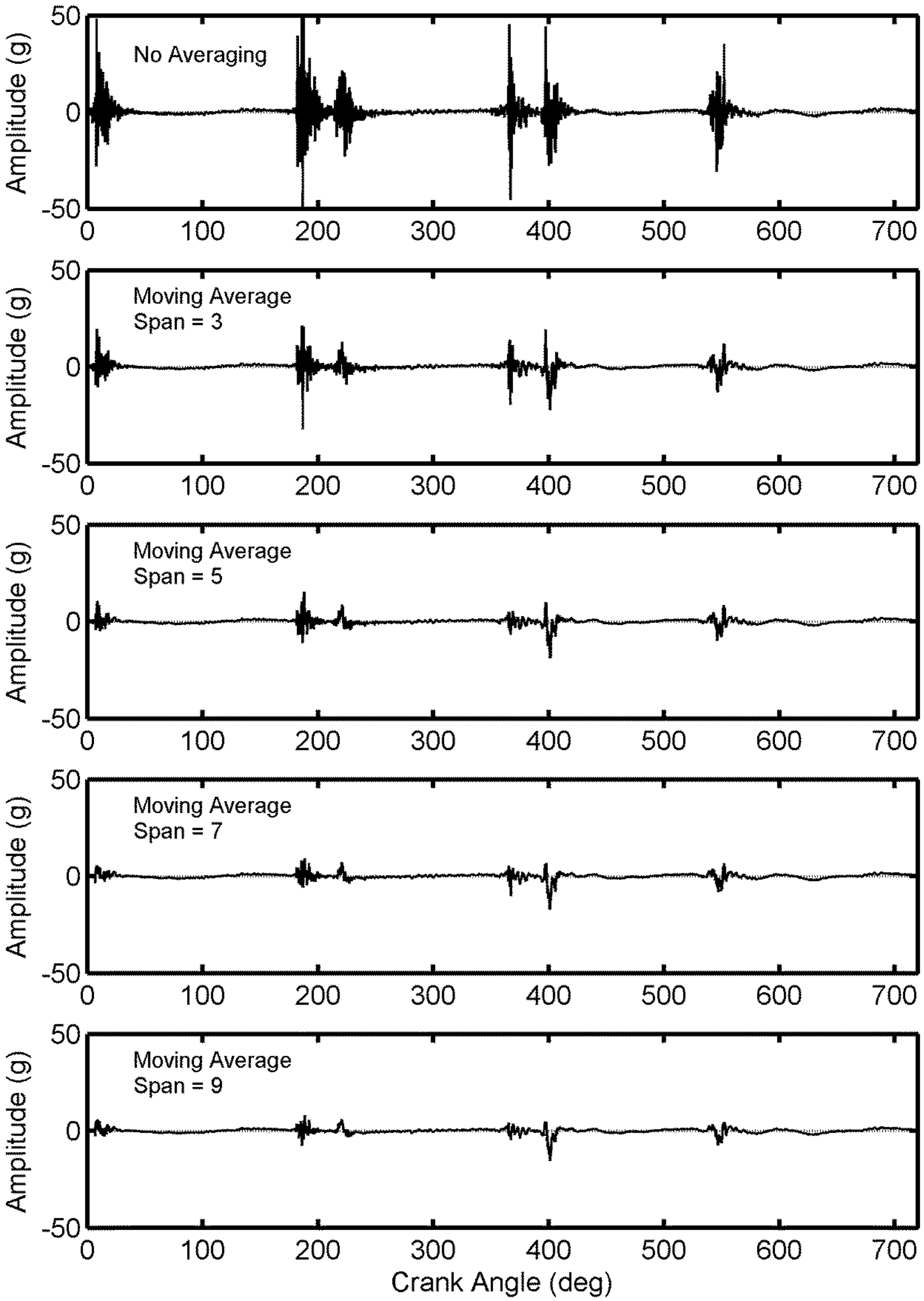

Figure 4-7: Vibration signal for various moving average filter spans

4.3.2.2 Median Filtering

The median filter is a filtering technique that is commonly used in edge detection or step detection applications. It has the advantage of preserving sudden changes in the signal while reducing noise for certain applications. Since the impacts produce sudden changes in the acceleration signal the median filter is an important filter to investigate.

A variety number of median filter spans were tested against the case of no median filtering. To implement a median filter, each data point is replaced by the corresponding median value of a given span:

$$y_m(i) = median\left\{ y\left(i - \frac{(N_m - 1)}{2} \right) : y\left(i + \frac{(N_m - 1)}{2} \right) \right\} \tag{4.8}$$

where $y_m(i)$ is the median filtered value for the ith data point, and N_m is the span of the median filter. Again, a quantifiable comparison was done by using equations (4.4) and (4.5) to calculate the signal-to-noise ratio as described in Section 4.3.1. Table 4-4 shows a comparison of SNR values for a variety of different median filter spans.

Median Filter Span	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
0 (No filtering)	0.363	0.0192	18.94
3	0.190	0.0173	10.96
5	0.144	0.0171	8.45
7	0.129	0.0170	7.61
9	0.113	0.0170	6.60

Table 4-4: SNR comparison for variety of median filter spans

From Table 4-4 we can see that the median filter gives better SNR results than the moving average filter values found in Table 4-3. The median filter does reduce the noise as expected, however the reduction in the impact signal again outweighs the noise reduction. From this it can be seen that using a median filter reduces the SNR. For this reason, although this filter gives better results than previous filtering methods, the median filter will not be used for this specific application. The effects of the median filter can be better visualized by looking at Figure 4-8.

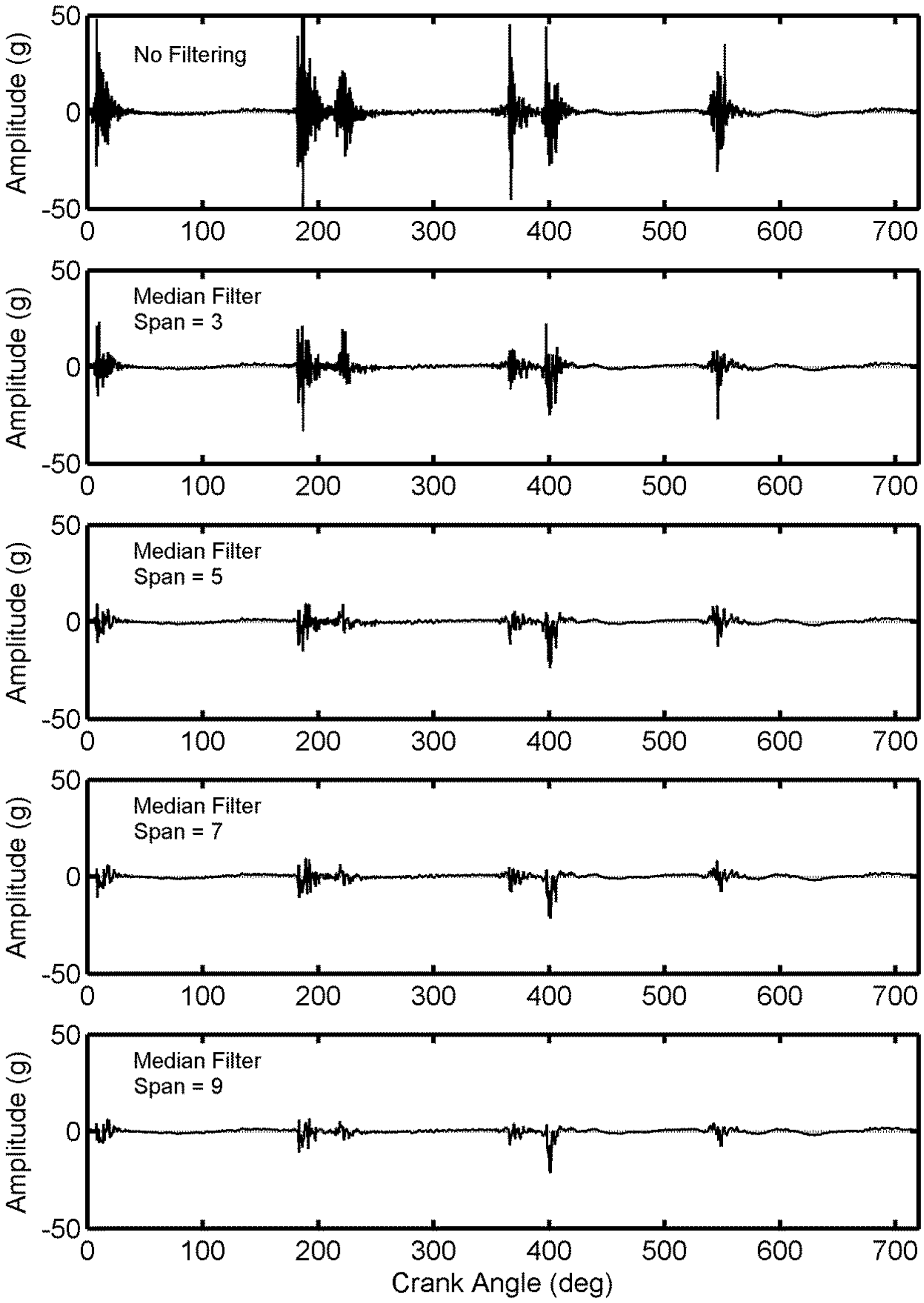

Figure 4-8: Vibration signal for various median filter spans

4.3.2.3 Recursive and Non-Recursive Filters

A recursive filter is a filter that uses one or more of its outputs as an input. These can be effective at noise reduction and are also commonly referred to as infinite impulse response (IIR) filters. An example of a well known IIR filter is the Butterworth filter. In this section a common digital Butterworth filter will be implemented and applied to our data. A variety of different pass-bands and frequencies will be tested and compared with the Butterworth Filter.

First, a tuning of the filtering frequencies was performed. First we looked at the low-pass 8th order Butterworth filter. By trial and error, we found that the low-pass cut-off frequency should be around 20 kHz. We found the best cut-off frequency by comparing the SNR of various frequencies as shown in Table 4-5. From Table 4-5 it can be seen that 19100 Hz gives the best SNR.

Low-pass Cut-off Frequency **(Hz)**	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
18900	0.2934	0.0231	12.72
19000	0.2940	0.0231	12.73
19100	0.2945	0.0231	12.74
19200	0.2950	0.0232	12.73
19300	0.2954	0.0232	12.72

Table 4-5: Low-pass filter frequency tuning

Next, a similar analysis was done for the high-pass Butterworth filter. The SNR obtained with various high-pass cut-off frequencies are shown in the Table 4-6. From Table 4-6 it can be seen that 3400 Hz gives the best SNR.

High-pass Cut-off Frequency **(Hz)**	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
3200	0.360	0.0053	67.51
3300	0.359	0.0053	67.59
3400	0.359	0.0053	67.66
3500	0.359	0.0052	67.66
3600	0.358	0.0052	67.56

Table 4-6: High-pass filter frequency tuning

After low-pass and high pass frequencies have been tuned different filter types can be compared. Table 4-7 shows the comparison between no filter, a low-pass filter, a high-pass filter, and a band-pass filter. From Table 4-7 it can be seen that the high-pass filter or the band-pass filter gives the best results.

Filter Type	$V_{rms}(impact)$ "Signal" **(V)**	$V_{rms}(non-impact)$ "Noise" **(V)**	SNR
No Filter	0.363	0.0192	18.9
Low-Pass Filter	0.358	0.0190	18.9
High-Pass Filter	0.359	0.0053	67.7
Band-Pass Filter	0.352	0.0052	67.8

Table 4-7: Comparison of filter methods

The effects of the recursive 8th order Butterworth filter can be better visualized by looking at Figure 4-9. The vibration signal is shown for the case of no filtering, low pass filtering, high pass filtering, and band pass filtering. From Figure 4-9 it can be seen that no filtering gives the best impact signal, however also has considerable noise. The band pass filter reduces the noise and maintains sufficient impact information. For this reason we will be using an 8th order Butterworth band-pass filter with a 3400 Hz high-pass cut-off and 19100 Hz low-pass cut-off in our application.

Figure 4-10 shows the effect of the signal processing techniques used. The top plot is the raw signal with no processing done, the bottom plot is the vibration signal after the engine cycle averaging and band-pass filtering techniques have been applied. It is evident that the signal processing has reduced the noise drastically and made the impacts more distinguishable and thus easier to detect.

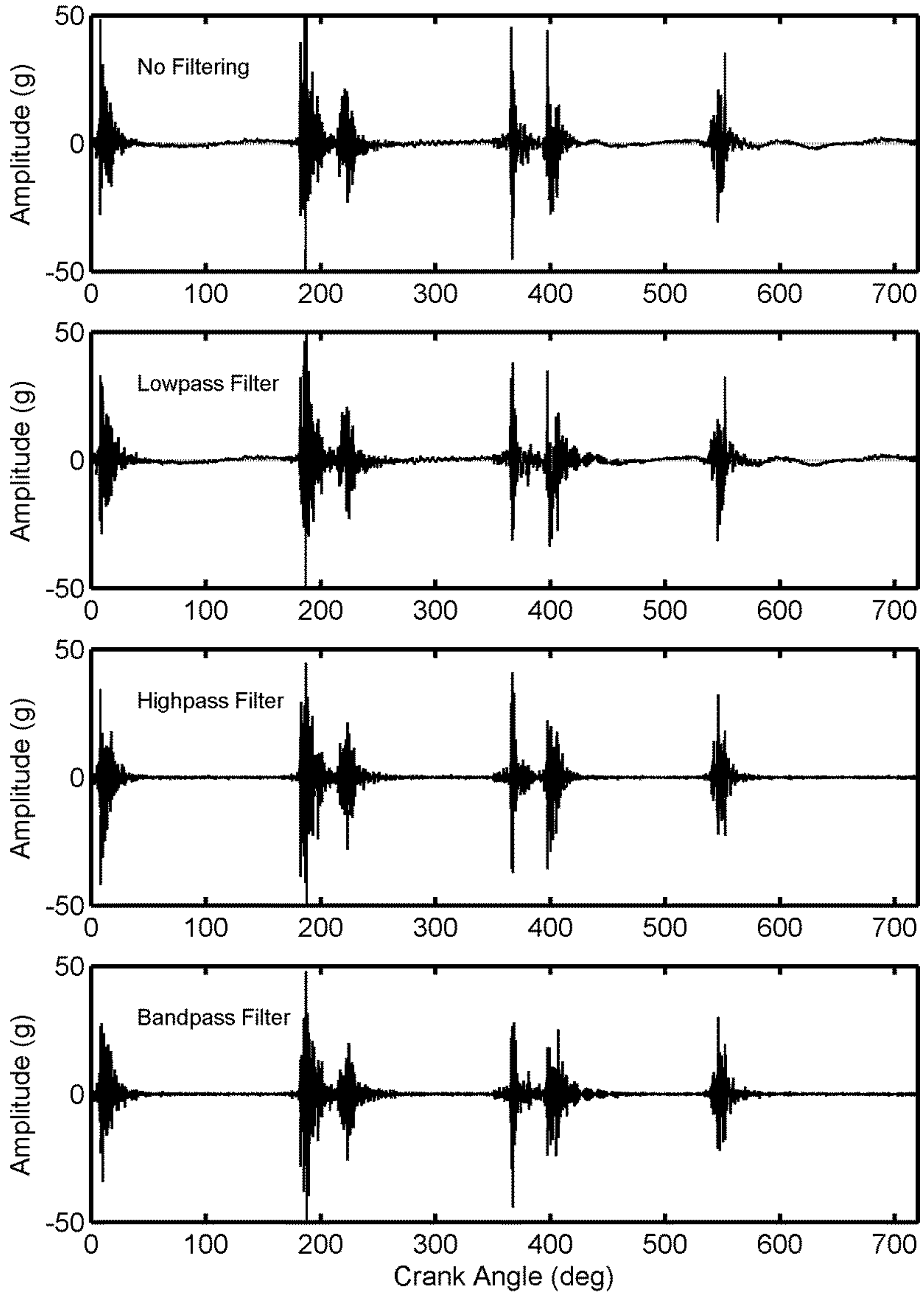

Figure 4-9: Butterworth filter comparison

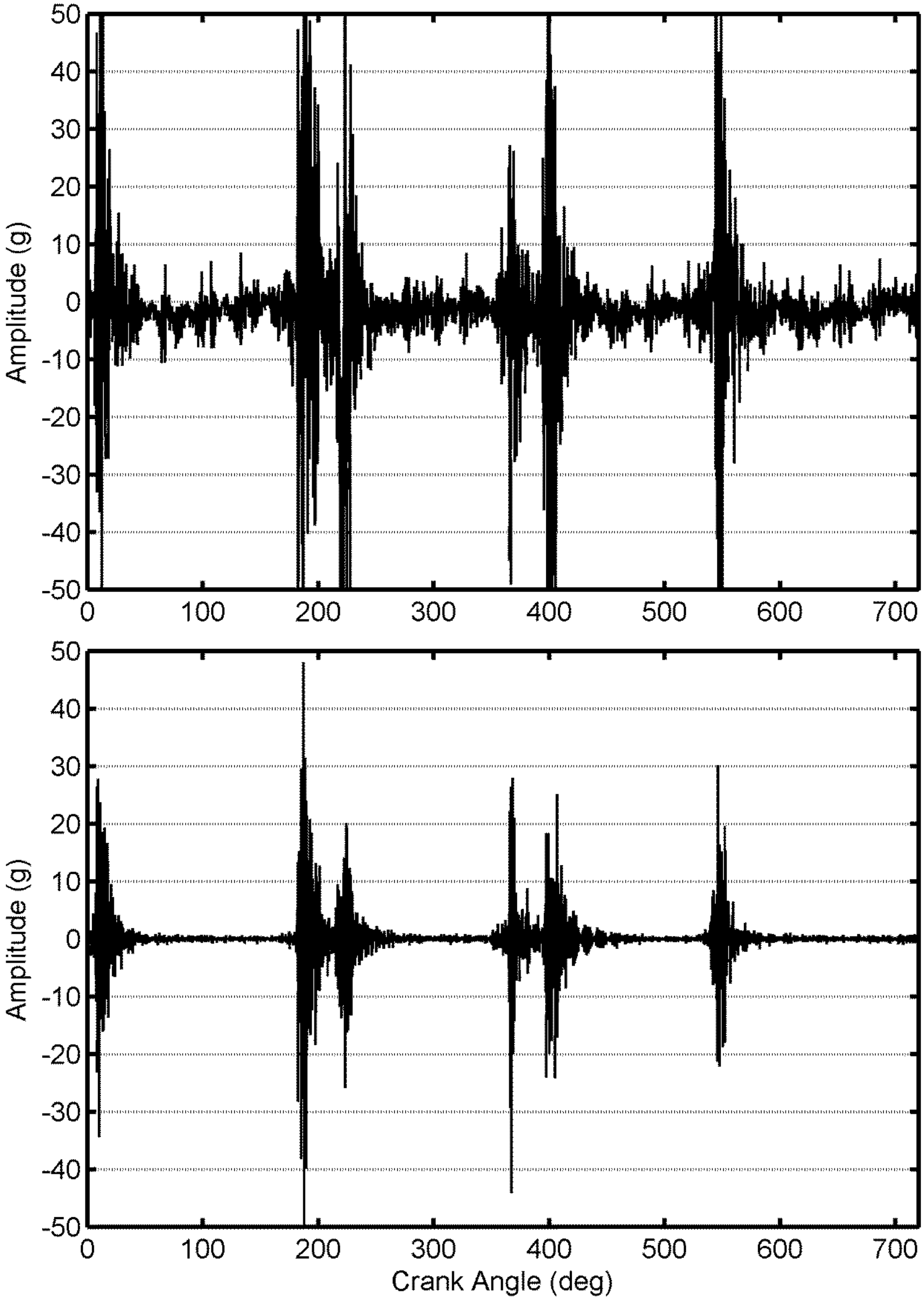

Figure 4-10: Raw signal (top) vs. processed signal (bottom)

4.4 Summary

In this chapter signal processing techniques were described. First the signal acquisition methods were described including the DAQ hardware and software. An explanation of the DAQ system was also given. Signal improvement techniques were then described including the averaging, filtering, and denoising methods to be used in our application. Cycle averaging and band-pass filtering techniques were found to be useful in the processing of our signal and will be used throughout this thesis. The results of the signal processing methods used can be seen clearly in Figure 4-10. The next chapter will discuss feature extraction methods as well as the classification system and FDD method.

CHAPTER 5: Development of Fault Detection and Diagnosis Methods

5.1 Introduction

FDD methods involve monitoring a specific system, and detecting any unexpected or undesired changes in that system. If the system is monitored appropriately, minor faults can be detected in the early stages and corrective or preventative measures can be taken. Detection refers to determining whether the system contains a fault or not. Diagnosis refers to locating and defining the type of fault.

As described in previous chapters, our fault detection method is focused on the analysis of the engine's vibration signal. This chapter will describe feature extraction and classification methods. The feature extraction method was newly developed for this application, and is described in the next section. The base-line classification method, the NB method introduced in Chapter 2, is presented after that. In the remainder of the chapter, several common classification methods are presented. The classification results obtained using these methods will be compared and contrasted in the next chapter.

5.2 Feature Extraction

Feature extraction is the process of analyzing a signal and extracting important attributes or features of that signal. A set of features were chosen that are associated with the valve closing impacts. In this section, the techniques developed to extract these features will be presented.

5.2.1 Local RMS Computation Using a Sliding Window Method

Vibration amplitude describes the severity of a vibration. Several methods to quantify vibration amplitude exist including peak values, peak-to-peak values, average value, and RMS value. The RMS value is a commonly used method with vibration signals as it gives a quantifiable value for amplitude and also maintains information on the time or phase of the signal (Brüel & Kjær, 1982).

For this reason the local RMS values were used to quantify the vibration signal. A sliding window method was used to calculate the local RMS value of a given window for each data point of the signal. Figure 5-1 shows this method.

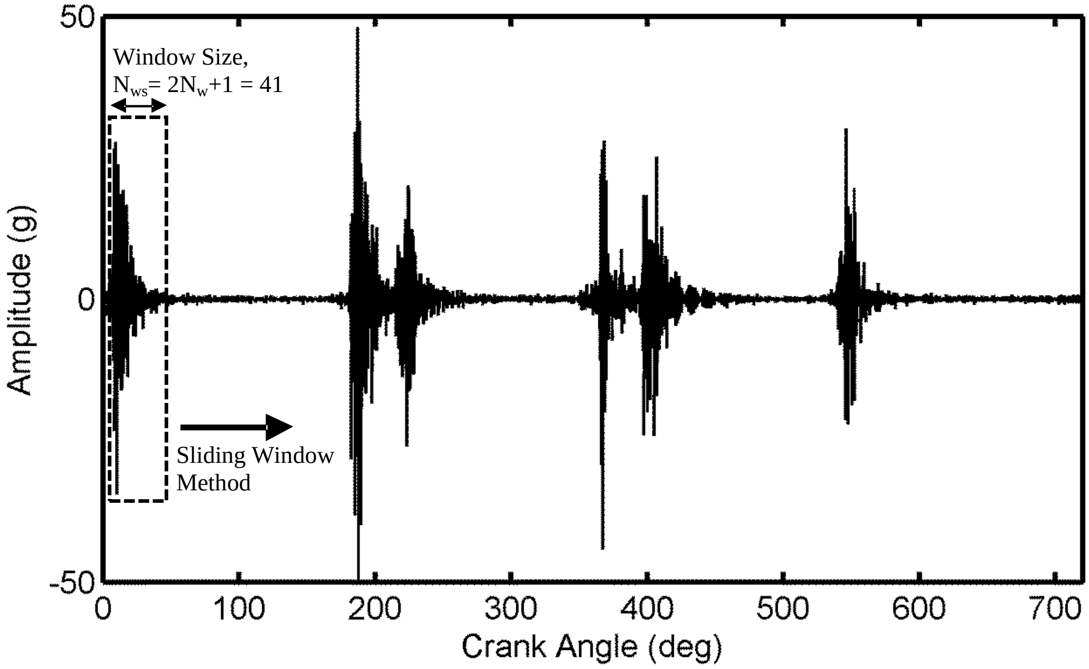

Figure 5-1: Sliding Window Method

The size of the sliding window is 41 data points. The window slides through the vibration signal and calculates the local RMS value for each point. This local RMS signal was given by:

$$y_{lrms}(i)=\sqrt{\frac{1}{2N_w+1}\left(y(i+N_w)^2+y(i+N_w-1)^2+\cdots+y(i-N_w)^2\right)} \qquad (5.1)$$

where $y_{lrms}(i)$ is the local RMS signal, $y(i)$ is the cycle averaged vibration signal, and $N_{ws}=2N_w+1$ is the size of the sliding window. This technique transfoRMS the cycle averaged signal into a more meaningful signal as can be seen by comparing Figure 5-2 to Figure 5-1. The transformed signal describes the pertinent features of the cycle averaged signal including the impact amplitude and timing in a clear fashion. From this signal the six major impact points can be easily distinguished. This RMS signal will be the basis for the feature extraction.

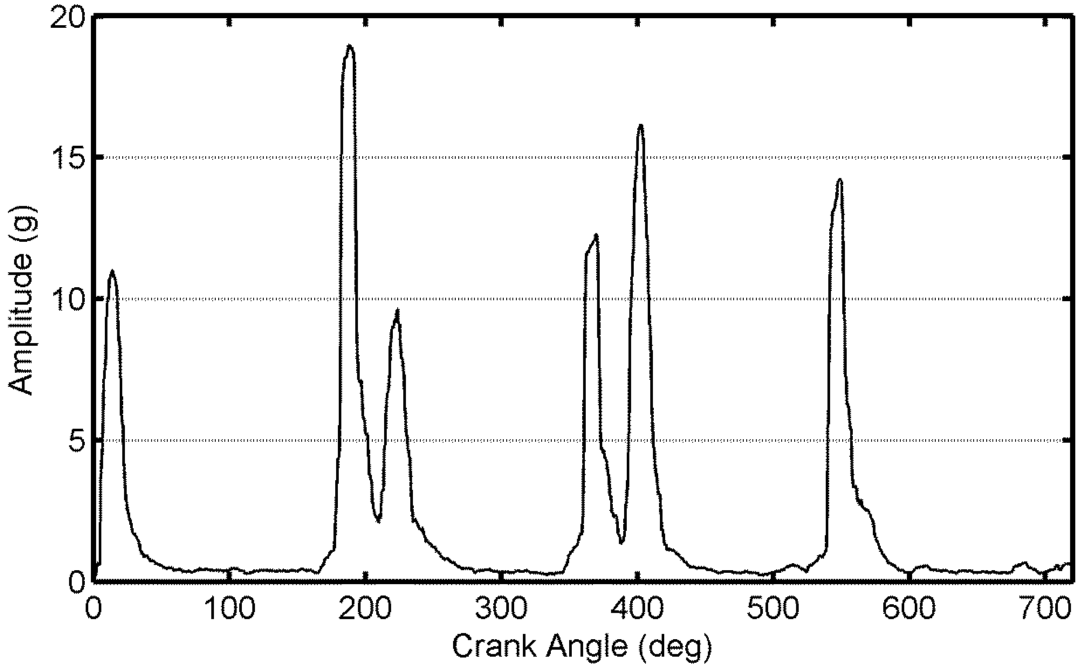

Figure 5-2: Local RMS vibration signature for the signal plotted in Figure 5-3

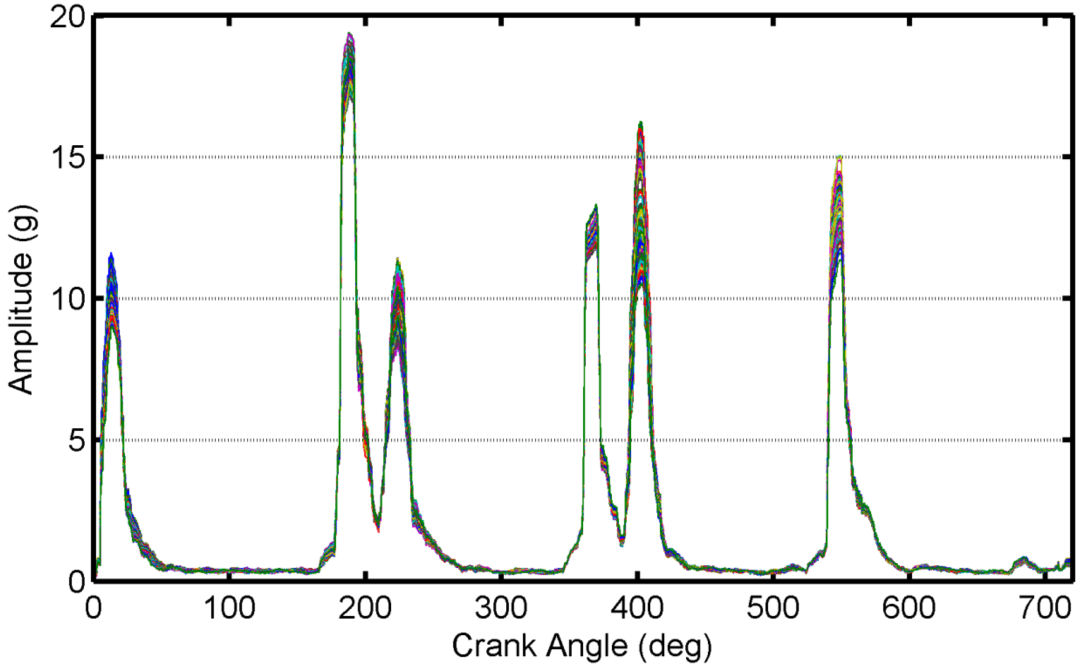

Figure 5-4: Local RMS Vibration Signal for 100 cycles of the Nominal Case

The local RMS vibration signal plotted in Figure 5-2 is for one engine cycle of the nominal (i.e. no fault) case. Figure 5-4 shows the RMS vibration signal for 100 cycles of the nominal case. This illustrates the variation of the signal for the nominal case. Figure 5-5 shows the RMS signal for the one cycle of the nominal case compared to one cycle of the EVC2 fault case.

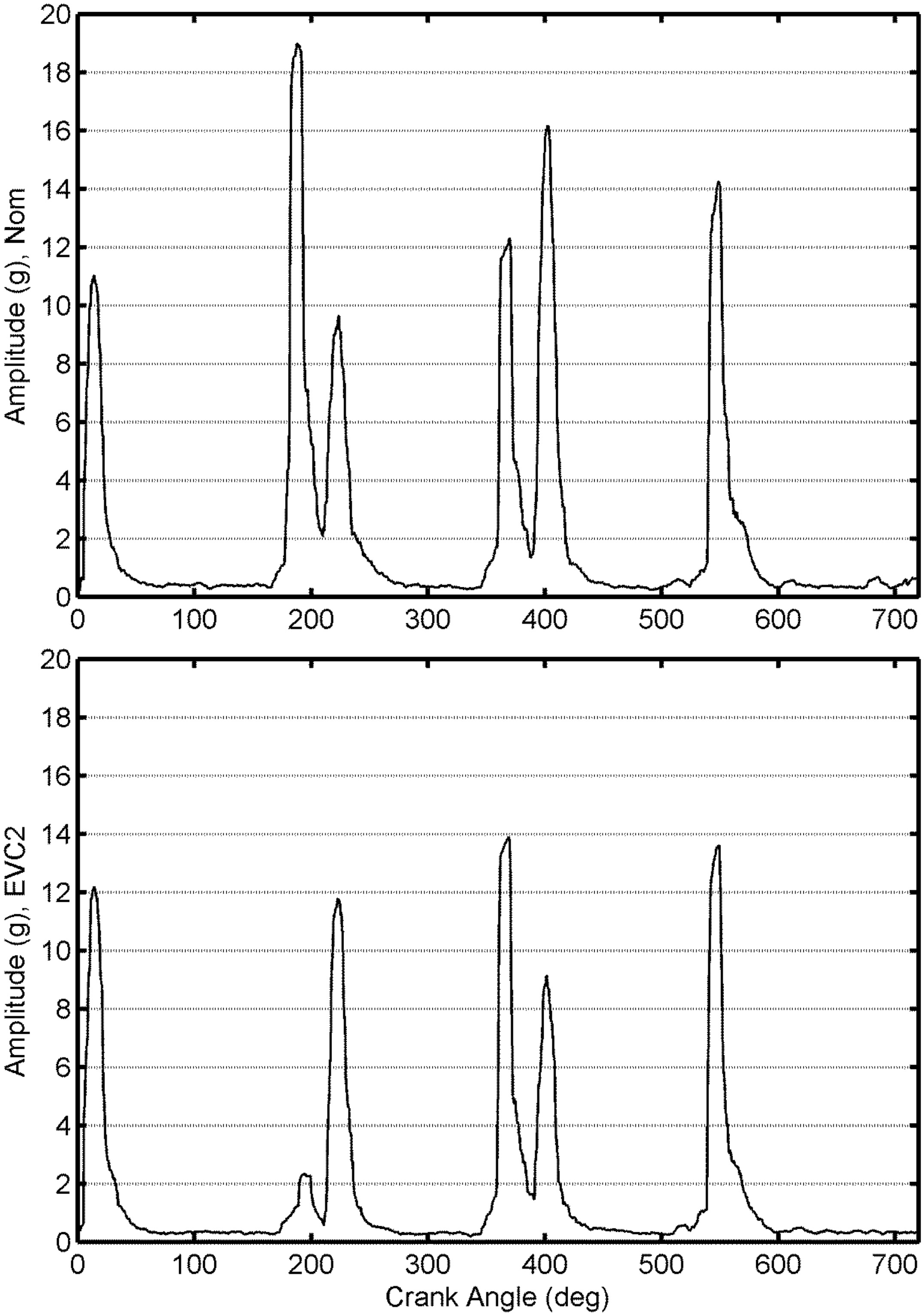

Figure 5-5: Comparison of local RMS Signals - Nominal (Top) vs. EVC2 Fault (Bottom)

From Figure 5-5 it can be seen that the second peak (at about 190^{o}) is much lower in the bottom graph than in the top one. It is also outside of the variation shown in Figure 5-4. The peak was reduced due to a spring fault seeded on EV2. This data suggests that the peak heights of the local RMS signature could be used as the basis of our FDD method.

5.2.2 Impact Detection Using Thresholding

Thresholding is a common technique used in signal processing. A data point can be classified by checking whether it is below or above a given threshold. In this application a threshold will be used to classify whether a specific point on the local RMS signature occurs in the vicinity of an impact (termed an “impact point”), or not (termed a “non-impact” point). When the signature point goes above the threshold value it is said to be an impact point, when the point is below the threshold it is a non-impact point. This can be written mathematically as:

```
$if\ y_{lrms}(i) > Y_{rms}$
        $impact(i) = true$
$else\ if\ y_{lrms}(i) <= Y_{rms}$
        $impact(i) = false$
$end$
```

where $y_{lrms}(i)$ is the local RMS signature at the i^{th} point, and Y_{rms} is the threshold value calculated by:

$$Y_{rms} = s_f \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} y(i)^2} \tag{5.2}$$

where s_f is a scale factor, $0.5 < s_f < 2$, and $N_d = 2880$. Note that if $s_f = 1$ then Y_{rms} is the RMS value of the vibration for one full engine cycle.

From this thresholding method the impact points are determined and the important features can be extracted.

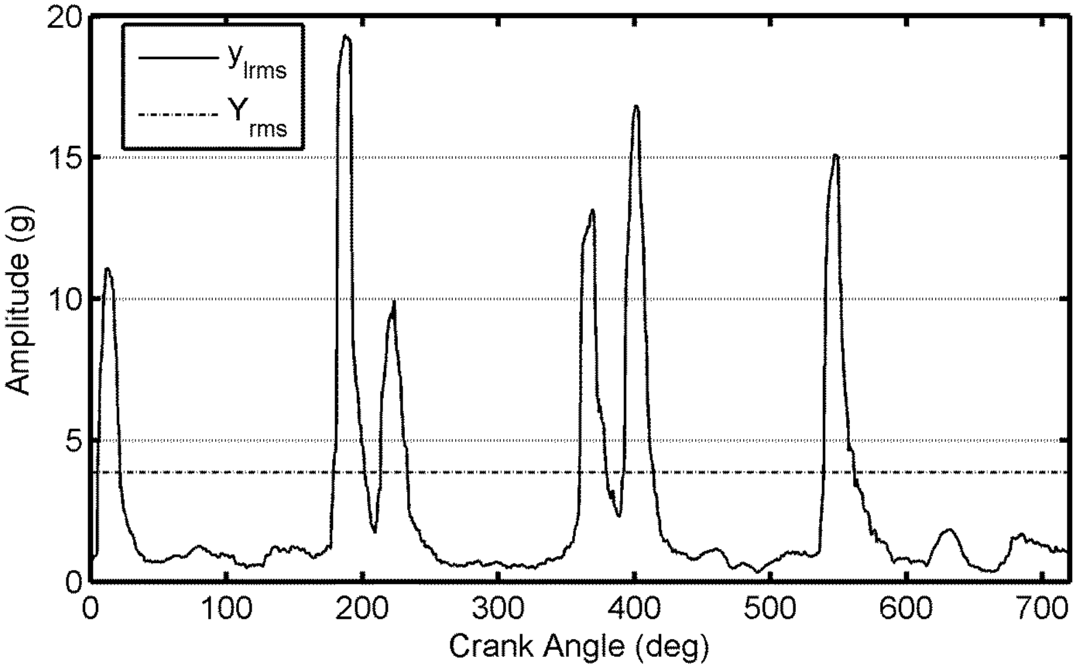

Figure 5-6: Thresholding Method

Figure 5-6 illustrates the thresholding method used. Note that the threshold value successfully intersects the peaks corresponding to the six distinct impacts, and does not intersect with any other points on the signal. The s_f value will be tuned in the next chapter to find an appropriate threshold value. Lowering the threshold value will classify more points as impact points; however it may also cause misdetection of signal peaks that are not impacts, as well as errors distinguishing neighbouring impacts. Raising the threshold value will reduce the occurrence of those problems; however too large of a value may cause an impact to be missed entirely.

If the threshold value is set too high, in the extreme worst case scenario it will not detect any of the impacts. More likely however, a high threshold value may overlook some of the small amplitude impacts. A small impact may be due to some variation in the impact due to random engine cycle variation, or a reduced impact due to a fault being present. If the thresholding method does not detect these impacts, the fault detection method will be degraded. Figure 5-7 below illustrates a threshold value that is too high.

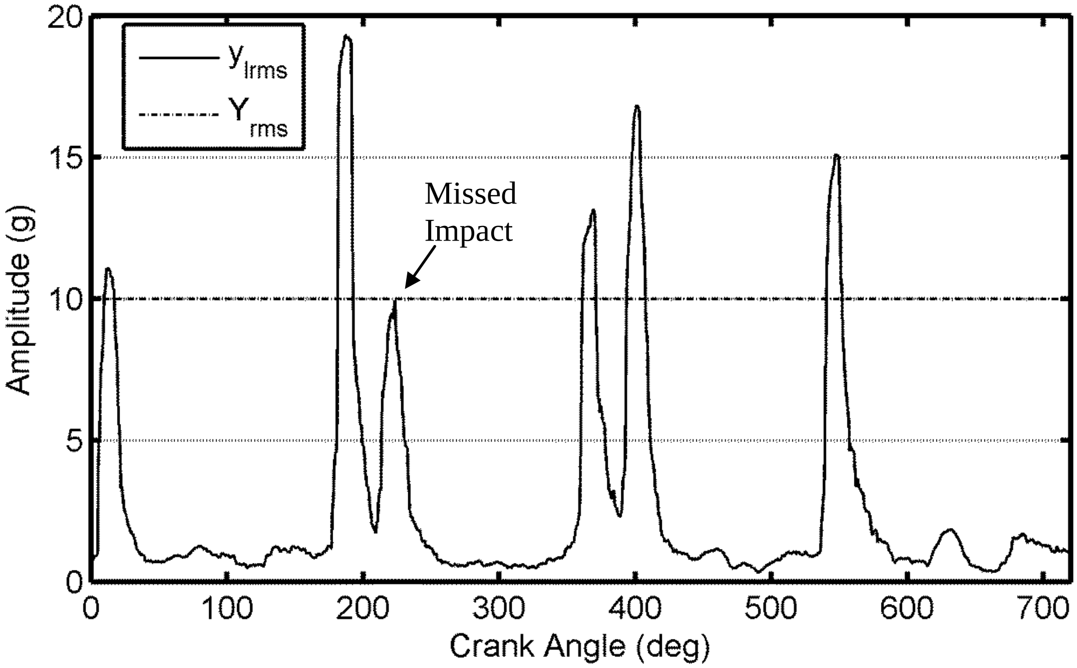

Figure 5-7: High Thresholding

Figure 5-7 shows that when the threshold is too high it misses the third impact, this would then incorrectly assume that only 5 impacts occur when 6 impacts are present. This thresholding error will lead to problems with our FDD method as a missed impact point may incorrectly indicate the presence of a fault when no fault is present.

A threshold value that is too low may falsely detect normal engine vibrations as valve impacts; it may also have trouble separating impacts that occur near each other. Again this is undesirable and will lead to a degradation of the fault detection method. Figure 5-8 below illustrates a threshold value that is too low.

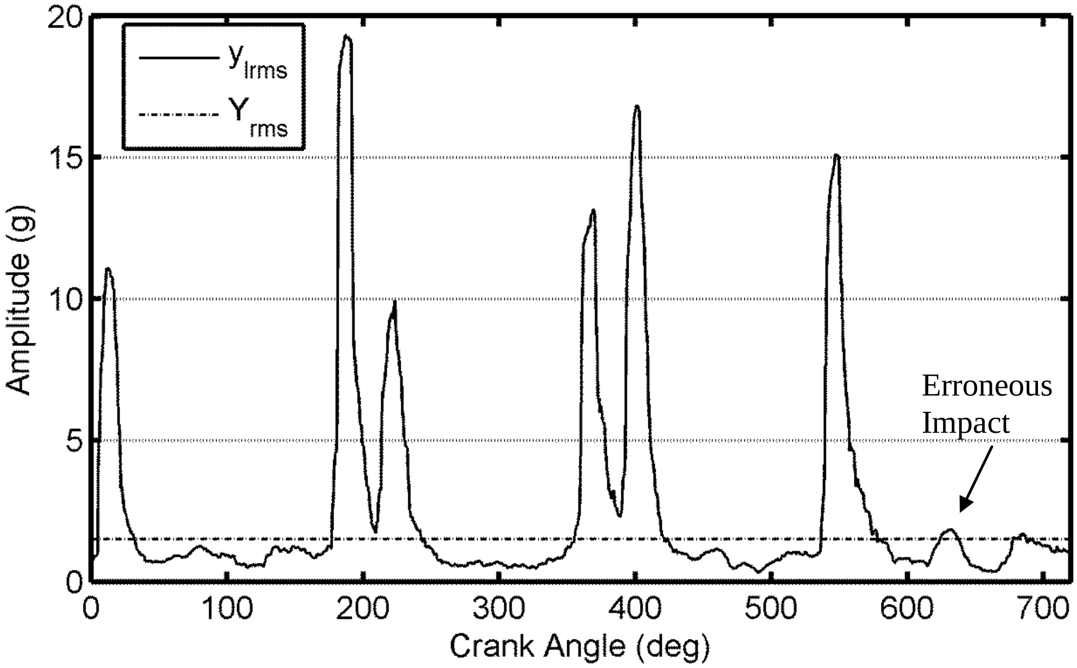

Figure 5-8: Low Thresholding

In the case where additional peaks are detected or expected peaks are undetected, corrective measures are taken. If additional peaks are detected each individual peak is examined and if determined to be inappropriate (i.e. the phase of the peak is unexpected) then this incorrect peak is removed. If expected peaks are undetected then the missed peak is given a standardized impact phase as determined by the manufactures specifications. Table 5-1 gives the standardized expected phase values for each valve impact (Kubota, 2008). Since actual impact phase values will shift slightly, these values are not as accurate as the actual values detected from the vibration signature.

Engine Cycle Impact	Standard Expected Phase
Exhaust Valve Close Cylinder 1	15º
Exhaust Valve Close Cylinder 2	180º
Intake Valve Close Cylinder 1	225º
Combustion Cylinder 1	360º
Intake Valve Close Cylinder 2	405º
Combustion Cylinder 2	540º

Table 5-1: Standardized impact peak phase values (Kubota, 2008)

5.2.3 Impact Amplitude Averaging

An impact region is determined from the local RMS sliding window and threshold methods. Figure 5-9 illustrates the impact region start point I_{RS} , and the impact region end point I_{RE} . The impact region is then calculated by

$$I_R = I_{RE} - I_{RS} \tag{5.3}$$

where I_R is the impact region.

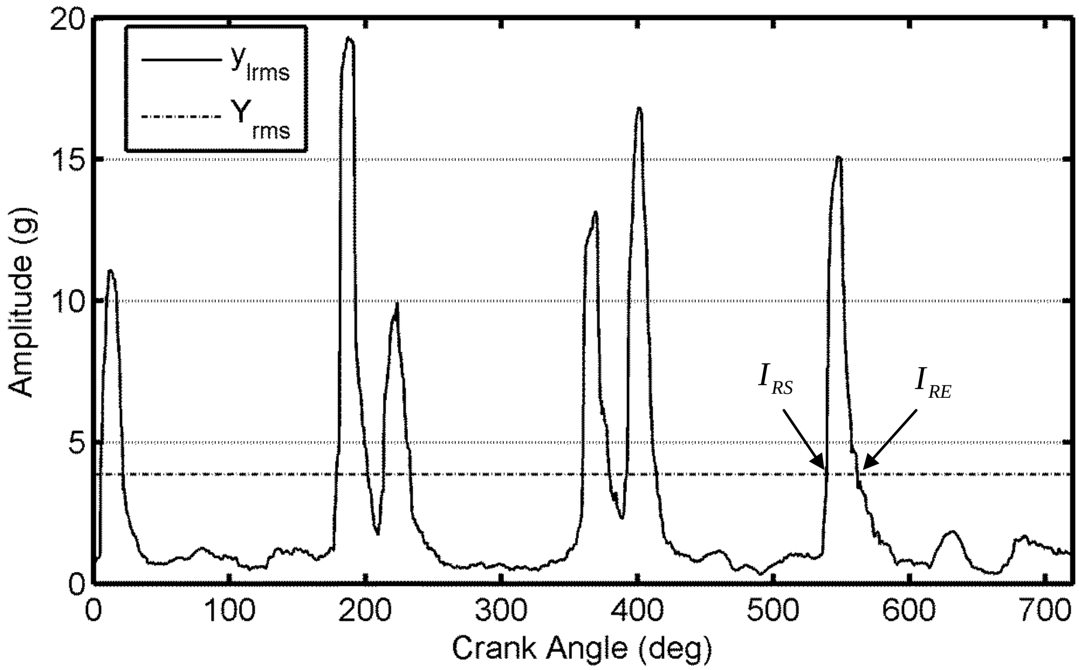

Figure 5-9: Determining the impact region

The impact amplitude is then averaged over this impact region to obtain the impact amplitude feature.

$$I_{amp} = \frac{1}{I_R} \sum_{i=I_{RS}}^{I_R} y_{lrms}(i) \tag{5.4}$$

where I_{amp} is the averaged impact amplitude. This will be the feature used in this thesis.

5.2.4 More Complex Cases: Segmenting Impacts

In some of the more complex testing cases such as faults on multiple valves or multiple faults on the same valve simultaneously it was found that the FDD results were not satisfactory. Figure 5-10 shows the vibration signal of a nominal test case compared to an EVC1 SF + LCF case. From Figure 5-10 it can be seen that the vibration signals look quite similar, and detecting a fault could be difficult. When looking at an RMS average of each individual impact as a whole, faults become difficult to detect for these cases.

For these cases further analysis was done on the vibration signal in an attempt to improve FDD results. It was found that different segments of the vibration signal impacts seemed to behave differently in fault and nominal cases, where sn will be the segment number. Figure 5-11 shows the segmentation method being applied to the EVC impact for cylinder 1. This impact will be named EVC1. Using the same naming convention, the other impacts are named: EVC2, IVC1, and IVC2. It can be seen that the EVC1 impact occurs from 6º to 20º in the crank angle domain. For non-complex cases this entire impact box would be RMS averaged as described in 5.2.1 to 5.2.3 however further analysis shows that segmenting the impact may help distinguish between faults. In this case the impact box was segmented into two equal sections, and then each section is analyzed as described in 5.2.1 to 5.2.3. This method improved the results for these complex cases, as will be shown in the next chapter.

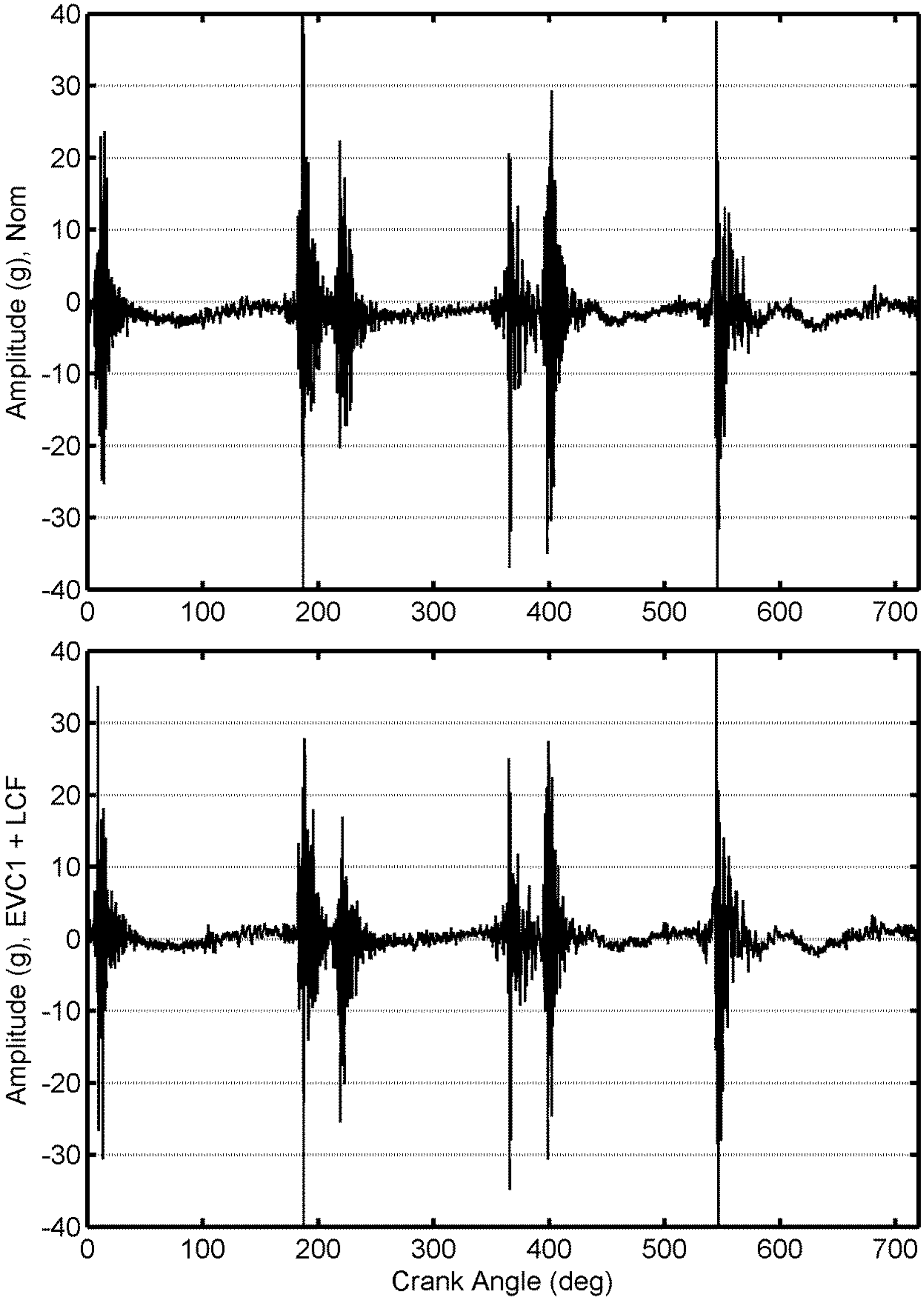

Figure 5-10: Vibration Signal - Nominal (Top), EVC1 SF + LCF (Bottom)

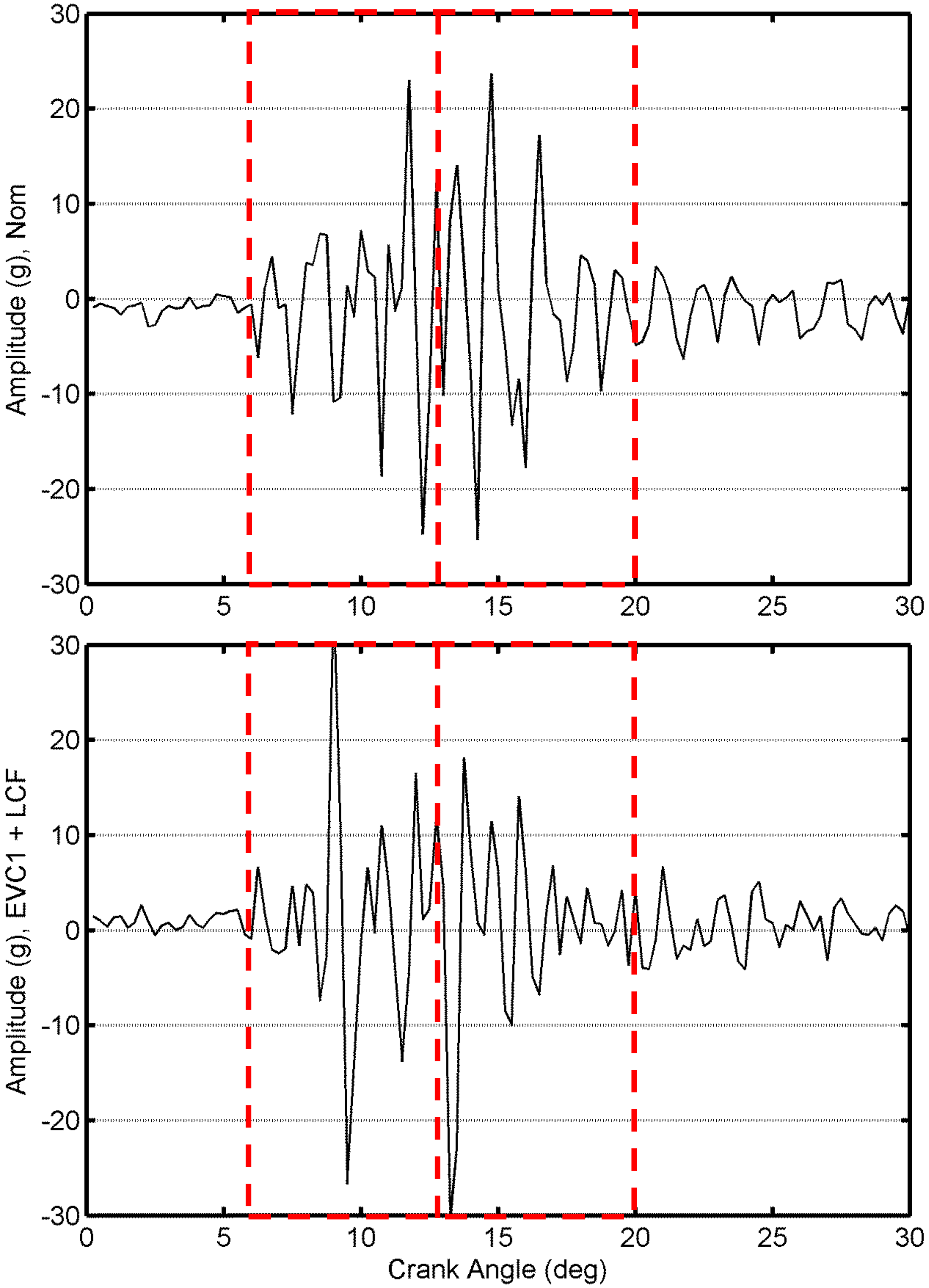

Figure 5-11: Vibration Signal of EVC1 - Nominal (Top), EVC1 SF + LCF (Bottom)

Figure 5-11 illustrates the segmentation method when an impact is segmented into two equal sections. Further segmentation can be done to improve classification results if necessary. However it should be noted that increasing segmentation increases the number of features, which will require longer training time for the FDD.

5.3 Naïve-Bayes Classification

Classification or pattern recognition methods are applied when no prior knowledge of the relation between symptoms and faults is available. Instead, reference patterns are determined for the faults experimentally by learning or training. The NB method will be the base-line classification method for this thesis; however alternative methods will also be studied for comparison. This method has been chosen due to its simplicity, white-box architecture, and the successful results found in the literature review.

5.3.1 Introduction & Background Information

NB classification is one of the most well known classification schemes (Isermann R. , 2006). This method is based on assumptions of the statistical distributions of the fault symptoms. The Gaussian or normal distribution is assumed in this thesis. This assumption is tested in 5.3.4.

5.3.2 Training

First, the amount of data to be used for training the classification method must be decided. In general as the training size (N_t) is increased the probability of successful classification also increases. However, a larger amount of training data also increases the training computation time, and reduces the amount of data available for testing. For this reason determining the smallest training data set without significantly deteriorating classification results is desired. Training is applied to each possible case, including all fault cases and the nominal case. This is done separately for each valve.

Training is done utilizing the features obtained as described in section 5.2. The impact mean μ and standard deviation σ are calculated for each case from a specified training set as follows:

$$\mu_{class,vn,sn} = \frac{\sum_{i=1}^{N_t} I_{amp,vn,sn,i}}{N_t} \tag{5.5}$$

$$\sigma^2_{class,vn,sn} = \frac{1}{N_t}\sum_{i=1}^{N_t}(I_{amp,vn,sn,i} - \mu_{class,vn,sn})^2 \tag{5.6}$$

where $I_{amp,i}$ is the *i*th impact amplitude from the training set for the valve, N_t is the training size used, vn represents the valve number, sn represents the impact amplitude segment number, and $class \in \{\text{nominal, SF, SCF, LCF, SCF+SF, LCF+SF}\}$ represents the class of the segment. Once these means and standard deviations are determined for all classes of each valve, additional data can be tested (i.e. classified) using statistical probability methods.

5.3.3 Testing

Once training is complete all other data is tested against our trained data to determine if the test case represents a fault. This is accomplished by comparing NB probability posteriors for each individual case. The probability of some value x residing in some class c can be computed using normal distribution:

$$P\left(x\,|\,c\right)=\frac{1}{\sqrt{2\pi{\sigma_c}^2}}\exp\left(\frac{-(x-\mu_c)^2}{2{\sigma_c}^2}\right) \tag{5.7}$$

where μ_c is the mean of the class as calculated by (5.5) and σ_c is the standard deviation of the class as calculated by (5.6). If the mean and variance for each class are computed from our training set the probability of test data can be classified into one of the fault classes using some distinct feature. As discussed in the previous sections, impact amplitude has been found to be a critical feature for the distinction of faults. From this we can say:

$$P\left(I_{amp,i}\,|\,class\right)=\frac{1}{\sqrt{2\pi\sigma^2_{class,vn,sn}}}\exp\left(\frac{-(I_{amp,i}-\mu_{class,vn,sn})^2}{2\sigma^2_{class,vn,sn}}\right) \tag{5.8}$$

where $P\left(I_{amp,i}\,|\,class\right)$ is the probability that $I_{amp,i}$ resides in some $class$, and $I_{amp,i}$ represents the amplitude of impact i. Where $\mu_{class,vn,sn}$ and $\sigma_{class,vn,sn}$ were found by training methods as described in Section 5.3.2 These calculations are done for each impact point and probabilities are determined for each fault class. Spring faults are the main fault of interest; however more complex fault classes will be explored. In the case where $sn>1$, the probabilities are calculated using (5.8) for each segment and then multiplied to determine the total probability of the impact.

$$P\left(I_{amp}\,|\,class\right)=\prod_{j=1}^{sn}P\left(I_{amp,j}\,|\,class_j\right) \tag{5.9}$$

where $I_{amp,j}$ and $class_j$ are the impact amplitude and class of the j^th^ segment respectively.

The probability of each of the fault classes is calculated as described using equation (5.8). The probabilities are then compared to determine the class of the test case. In equation form:

$$\hat{Y} = \arg\max P\left(I_{amp} \mid class\right) \tag{5.10}$$

where $\hat{Y}$ is the predicted class. Each of the four valves has 6 possible classes, this give a total of $6^4 = 1296$ possible fault combinations in a single engine cycle. It is unreasonable to attempt to test all of these combinations. However, each individual valve and class will be tested, in addition to a few complex combinations. These tests will be presented in the next chapter.

5.3.4 Normality

The use of these statistical methods and probabilities relies on the assumption of a normal distribution. A normal or Gaussian distribution is a continuous probability distribution defined as:

$$\Phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{5.11}$$

where $\Phi(x)$ is the probability distribution function of variate x, μ is the expected mean and σ is the expected standard deviation. The normal distribution is the most widely used statistical model for the distribution of random variables (Montgomery & Runger, 2007).

In order to justify using these methods we must show that our data can be described by a normal distribution. Figure 5-12 shows a histogram of the RMS EVC1 impact amplitudes for 440[1] engine cycles. A normal distribution curve is plotted over the histogram for visual

[1] Each data set originally consisted of 500 engine cycles. After implementing engine cycle averaging with N_a=60, 440 averaged cycles remained for analysis and testing.

comparison. Based on these results, and similar results found for the other impacts, we concluded that the normal distribution assumption is reasonable for our data.

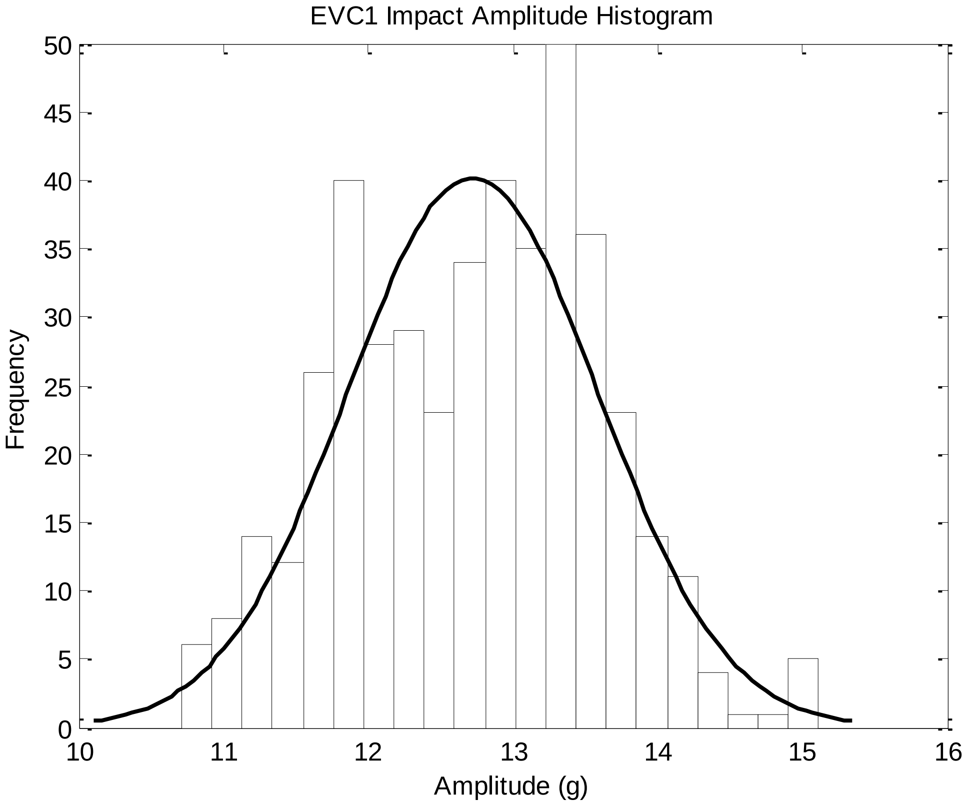

Figure 5-12: Normality testing of EVC1 impact amplitudes.

5.4 Alternative Classification Methods

Based on the literature review presented in Chapter 2:, four well known classification methods were selected for comparison. This section will briefly describe these methods.

5.4.1 Artificial Neural Networks

An ANN consists of numerous processing elements or nodes connected together to form a network. Generally each node consists of a single simple processing element. Each node output is computed by

$$y = \sum_{i=1}^{n} w_i x_i \tag{5.12}$$

where w is the weight of the network node, x is the input value to that node, for all i inputs. In this thesis $x_i = I_{amp,i}$ for the hidden layer. Since they are commonly used for prediction and pattern recognition, a feedforward network will be used in this thesis. Feedforward networks have a directed acyclic graph with one-way connections from input to output layers. Figure 5-13 shows a simplified diagram of the feedforward ANN used for our application. The network consists of 6 inputs which represent the 6 impacts in an engine cycle, 11 nodes or processing elements, and 5 outputs which represent the 5 possible fault cases. The number of hidden layer nodes was obtained by using the general rule of thumb that the hidden layer nodes equal the sum of the inputs and outputs (Gupta, 2001). The simplified case shown in Figure 5-13 is only concerned with detecting SFs, with the addition of other classes and complex cases including combinations of faults the ANN input, output, and hidden parameters will need to be altered accordingly.

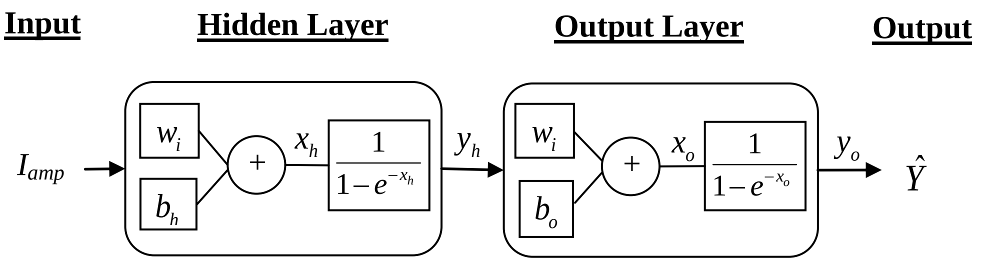

Figure 5-13: Artificial neural network diagram for our application

From Figure 5-13 it can be seen that our ANN has two layers: the hidden layer and the output layer. The hidden layer computes an activation input from the input features for each of its 11 nodes by:

$$x_h = b_h + \sum_{i=1}^{n} I_{amp,i} \times w_{h,i} \tag{5.13}$$

where x_h is the hidden layer activation input, n is the number of hidden layer inputs, $w_{h,i}$ is the weight of the i^{th} hidden layer input, and b is some bias value. This activation input x_h is then put through an activation function, in our case the activation function being used is the sigmoid function:

$$y_h(x_h) = \frac{1}{1-e^{-x_h}} \tag{5.14}$$

This procedure is repeated for the five nodes of the output layer of the ANN, using the hidden layer outputs as the output layer inputs. The output layer computes an activation input from the input features for each of its 5 nodes by:

$$x_o = b_o + \sum_{i=1}^{m} y_h \times w_{o,i} \tag{5.15}$$

where x_o is the output layer activation input, m is the number of output layer inputs, $w_{o,i}$ is the weight of the i^{th} output layer input, and b_o is some output layer bias value. This activation input x_o is then put through an activation function, in our case the activation function being used is the sigmoid function:

$$y_o(x_o) = \frac{1}{1 - e^{-x_o}} \tag{5.16}$$

The weights of the ANN are initialized to small random values. The weights must be updated or trained for the ANN to be useful. Training is done by minimizing some cost function. The cost function used in our application is the mean-squared error (MSE), which attempts to minimize the average squared error between the network's output and the target value. The mean-squared error can be determined by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{5.17}$$

where $\hat{Y}$ is the predicted class and Y is the true value of the class. This cost function is commonly minimized using the gradient decent method (Avriel, 2003). These minimized errors are then used to update the weights for the next training epoch, where an epoch is a single pass through the entire training set. This method is known as back-propagation or backward propagation of errors and is one of the most common methods for training neural networks (Isermann R. , 2006). Once the weights are updated the ANN goes through another cycle and continues this learning algorithm until some stopping condition is met, in this case a minimum gradient magnitude of 1×10^{-6}.

5.4.2 Decision Trees

Decision trees were chosen for this application since they are a commonly used method for classification, and are relatively simple to interpret and implement. Figure 5-14 shows a simplified decision-tree diagram for our application when only concerned with detecting SFs.

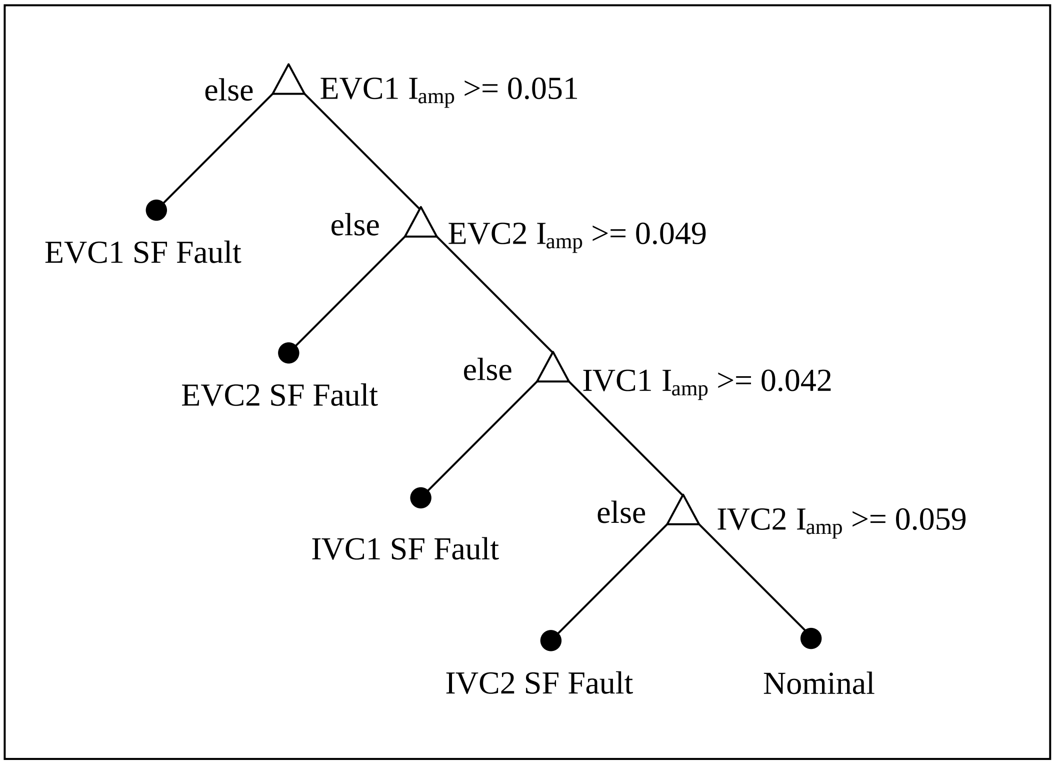

Figure 5-14: Decision tree diagram for our application

Algorithms for constructing DTs generally work top-down and choose a variable at each step to split the set of items. The classification and regression tree (CART) algorithm was chosen for this application (Breiman, Friedman, Stone, & Olshen, 1984). The CART algorithm minimizes the Gini index to determine how to construct the tree (Isermann R. , 2006). This produces the set of node values, e.g. for the root node in Figure 5-14 that node value is 0.051V. The resulting tree can be executed very quickly since it is a series of if-then rules.

5.4.3 k-Nearest Neighbour (k-NN)

The k-NN classification method was chosen for this application since it is popular and generally provides good results. Figure 5-15 shows an example of the k-NN classification method for a simple case of only two classes and 6 training data points for each class. The test data point is classified based on the majority of its k nearest neighbours based on the Euclidean distance. It can be seen that when $k = 1$ the test data point would be classified as class 1, however when $k = 3$ the test data point would be classified as class 2. This example shows that using different k values can change classification results. Unlike the previously described methods, the k-NN method requires no training computations. However, all of the distance computations must be performed during testing which makes this method slow for large testing sets.

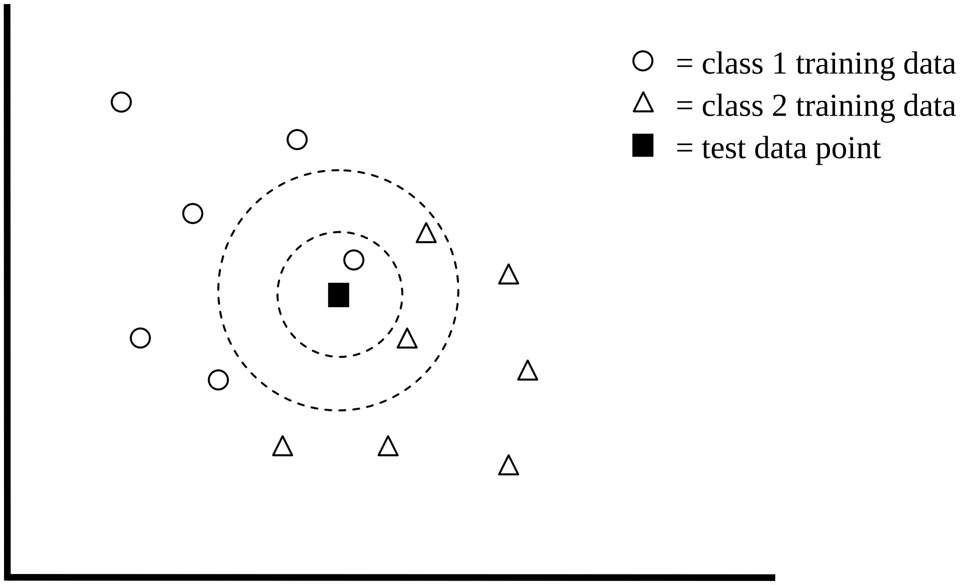

Figure 5-15: k-Nearest Neighbour classification example

5.4.4 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a type of polynomial classification which uses functional approximations to compute probabilities of the classes rather than the independent Gaussian functions assumed for NB classification. It was chosen since it has seen extensive use and is one of the most time-tested classification methods. The standard LDA model can be used to classify samples using:

$$S_{class} = s_0 + s_1 I_{amp,1} + s_2 I_{amp,2} + \cdots + s_k I_{amp,k} \tag{5.18}$$

where S_{class} is the classification score of the fault class, s_k are the classification coefficients where k is the number of features being used dependant on sn and vn, and $I_{amp,k}$ are the feature values. Classification coefficients are calculated during training and then used for testing the unknown data. Each sample tested will have a classification score for each class and will be placed into a specific class based on that score as follows:

$$\hat{Y} = \arg\max S(class) \tag{5.19}$$

5.5 Software Implementation

The FDD methods presented in this chapter were written in Matlab m code. They were implemented using custom developed methods and functions as described in section 5.2, as well as functions from the Matlab Statistics and Neural Network toolboxes as described in section 5.3 and 5.4. This Matlab code was executed on a laptop PC with specifications as given in Table 5-2.

Operating System	Windows 7 Professional 64-bit
Processor	Intel Core 2 Duo T6400
Number of Cores	2
Clock Speed	2.0 GHz
Intel Smart Cache	2 MB
Memory	4 GB RAM
Hard Drive Model	ST9250827AS ATA
Capacity	250 GB
Cache	8 MB
Speed	5400 RPM

Table 5-2: Laptop PC specifications

5.6 Summary

The FDD methods were described including feature extraction and classification methods. First, the newly developed feature extraction method was presented. It features a sliding window RMS calculation, followed by a thresholding approach with the possible use of impact segmentation if necessary for more complex fault scenarios. The main classification method explored was the NB classification method. This method was described in detail for our specific application. The alternative classification methods that will be used for comparison were also described. The next chapter will present and discuss the classification results of the FDD methods described in this chapter.

CHAPTER 6: Experimental Results

6.1 Introduction

The experimental results obtained using the developed FDD system will be described in this chapter. Tuning of parameters and a comparison of classification methods will be presented. Results will be displayed using confusion matrices. A confusion matrix is a visualization tool commonly used to display the performance of a classification algorithm. Each row of a confusion matrix represents an instance in an actual class, while each column represents the predicted class.

6.2 Tuning of the FDD Parameters

In this section the effect of changing various parameter values on final results will be explored, and the parameters tuned to obtain the best FDD results. Parameters including window size, threshold value, engine speed, sampling rate, and filtering frequency will be explored. Each fault case testing group contains 5 sets of 440 engine cycles for a total of 2200 engine cycles where each engine cycle contains 2880 data points.

6.2.1 Tuning Window Size

The first parameter we choose to explore is the window size used for RMS calculation in the feature extraction method (see Section 5.2.1). If the window size is too large, the RMS signal becomes too smooth. This over-smoothing makes detection of peaks or impacts more difficult, and could potentially make two impacts that are near each other become indistinguishable. Some peaks or impacts may be missed due to this over-smoothing. The over-smoothing should reduce the variance, but will also reduce the mean. Recall that our detection method uses both mean and variance as described in Section 5.3. Figure 6-1 shows an RMS vibration signal that is over-smoothed when the window size $N_{ws} = 2N_w + 1 = 101$. With the Y_{RMS} threshold shown, only 4 of the 6 impact regions would be detected.

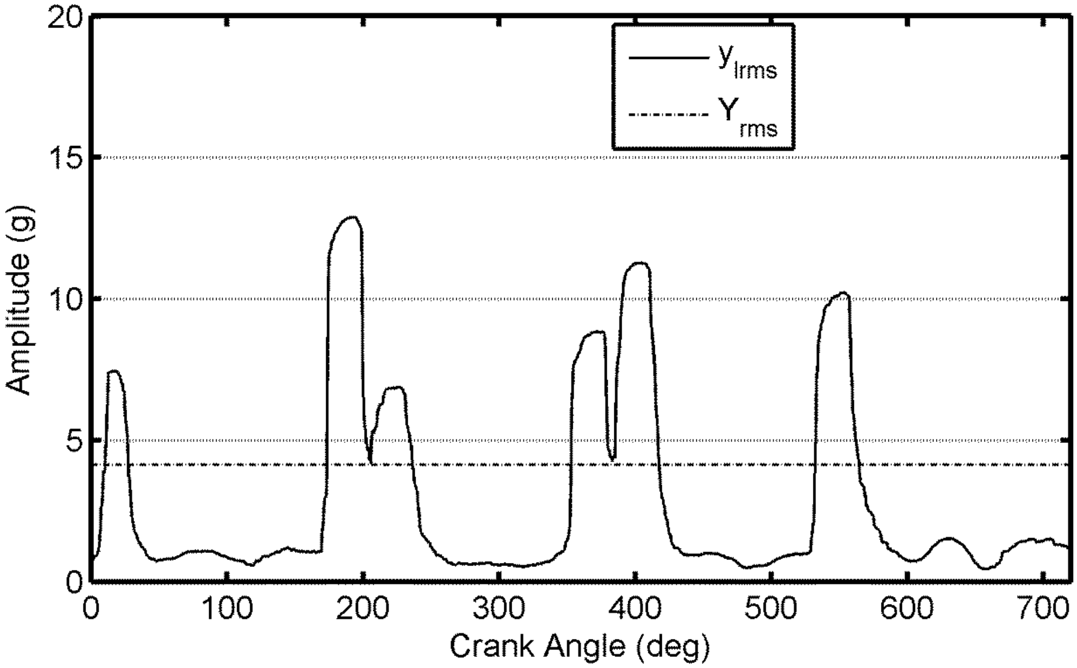

Figure 6-1: Over-smoothed local RMS signature with window size = 101

If the window size becomes too small, the local RMS signal becomes more erratic. The erratic signal may contain jumps which may lead to false impact detections or multiple region detections for a single impact region. This will be shown to decrease the classification success rate. Also an erratic local RMS signal will result in a higher variation between the cycles. This increased variation will also lead to incorrect classification as our base-line detection method is based on signal means and variances as described in the Section 5.3. Figure 6-2 shows a local

RMS vibration signature when the window size $N_{ws} = 2N_w + 1 = 3$. This signature is under-smoothed. With the Y_{RMS} threshold shown (the same as used previously), two erroneous impact regions would be detected as shown.

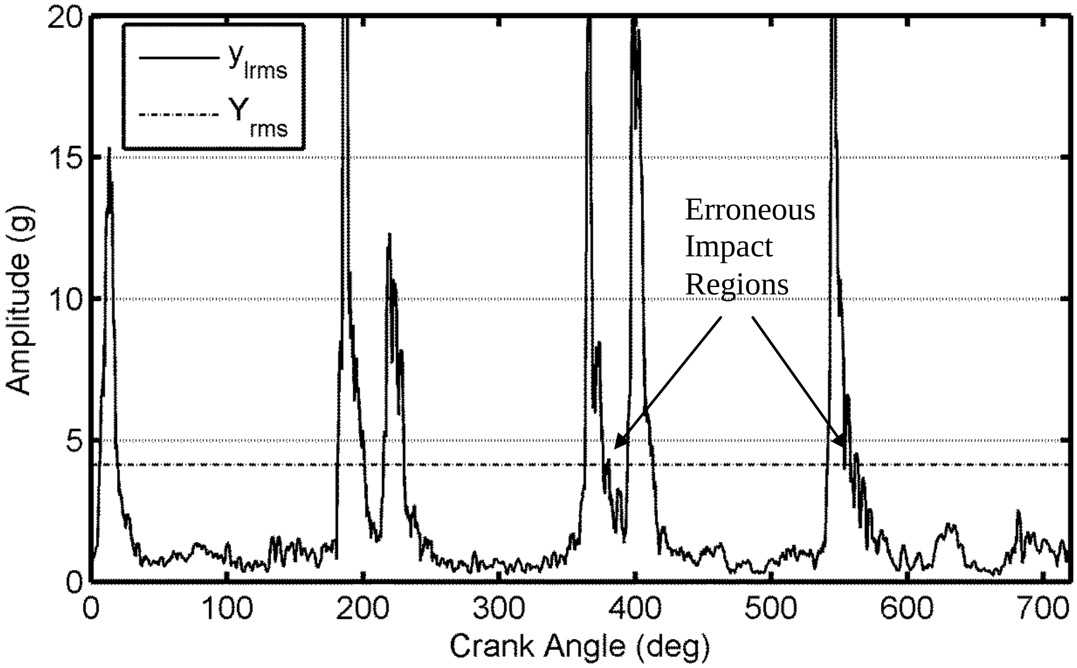

Figure 6-2: Local RMS signature with window size = 3

Figure 6-1 and Figure 6-2 explain visually how over-smoothing and under-smoothing can affect the impact region detection process. Manual tuning was done to find an appropriate window size for our application. Results are given for a number of different window sizes in Table 6-1 through Table 6-12 when $s_f = 1.5$.

Table 6-1 illustrates our classification results using a confusion matrix for the SF classes for window size of 101. A confusion matrix plots the actual known class of the case on the rows of the matrix against the predicted class of the case from our FDD method on the columns of the matrix.

From the confusion matrix both detection accuracy (DA) and classification accuracy (CA) can be determined. We define DA as simply detecting if a fault is present or if the system is operating nominally. CA then is the much more difficult process of classifying a fault into the correct class, location, and combination. Equations and sample calculations for the EVC2 case using Table 6-1 results are given in (6.1) and (6.2).

$$DA_i = \frac{C_{i,1}}{N_T + 1} \qquad when \quad i = 1$$

$$DA_i = \frac{\min\left(N_T, \sum_{j=2}^{n} C_{i,j}\right)}{N_T + 1} = \left.\frac{2200}{2201}\right|_{i=3} = 99.95\%\Big|_{i=3} \qquad when \quad i > 1 \tag{6.1}$$

where DA_i is the detection accuracy of class i, $C_{i,j}$ is the confusion matrix entry of row i and column j, and N_T is the number of test cases used for testing.

$$CA_i = \frac{C_{i,j=i}}{1 + \sum_{j=1}^{n} C_{i,j}} = \left.\frac{2200}{2566}\right|_{i=3} = 85.77\%\Big|_{i=3} \tag{6.2}$$

where CA_i is the classification accuracy of class i. Confusion matrices will be used exclusively for displaying the classification results throughout this chapter.

From equations (6.1) and (6.2) it can be seen that a single sample is added to the denominator. This is done because claiming 100% accuracy on a sample set would not be a fair claim; as increasing the sample size by a single sample could potentially cause a classification error. This ensures that our accuracies are calculated conservatively, and we state our sample size maximum DA and CA as 99.95% rather than 100%.

It is important to note that although there are only 2200 cases for each fault class, it is possible that a single case can have multiple valve faults. This results in the possibility of column and row sums of greater than 2200. From Table 6-1 it can be seen that each fault is successfully detected as a "true positive" which is defined as when a predicted class is classified into the correct actual class. However some cases also contain "false positives" which is defined as a case is predicted to be a fault when the actual case has no fault. In Table 6-1 the actual class of an EVC2-SF fault is correctly predicted to have an EVC2 fault for all 2200 cycles; however an IVC2-SF fault is also incorrectly predicted for 365 cycles when an IVC2 is not in-fact present.

Table 6-2 gives the NB parameter values of μ and σ for each I_{amp} for both the nominal and faulty case with a window size of 101 data points. The values of μ and σ are given in volts, where $1V = 100g$. Tuning the window size effects both I_{amp} feature values as well as the classification parameter values μ and σ. Detection and classification results are determined by a combination of the feature and parameter values. Generally a decrease in window size increases μ. From Table 6-2 and Table 6-12 $\mu_{nom,EVC2} = 0.1193$ when $N_{ws} = 101$, $\mu_{nom,EVC2} = 0.1476$ when $N_{ws} = 3$. However also increases the variance $\sigma_{nom,EVC2} = 2.314 \times 10^{-5}$ when $N_{ws} = 101$, $\sigma_{nom,EVC2} = 1.720 \times 10^{-5}$ when $N_{ws} = 3$. Accuracy should improve with increased μ and decreased σ so tuning is performed to find an acceptable window size.

From Table 6-1 through Table 6-12 we find that the best results occur at a window span of 41 data points. Table 6-7 shows that the lowest classification accuracy for this span occurs at

$CA_4 = 99.77\%$ and all other accuracies are above 99.95%. The results of Table 6-7 are very promising and a window size of 41 will be used for the remainder of this thesis.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	365	99.95	85.74
	IVC1 (SF)	0	0	20	2200	0	99.95	99.05
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-1: Confusion matrix (window size = 101)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0529	0.1193	0.0672	0.1053
	σ	2.895×10^{-6}	2.313×10^{-5}	2.405×10^{-6}	9.831×10^{-6}
Fault	μ	0.0118	0.0190	0.0105	0.0225
	σ	2.428×10^{-7}	2.230×10^{-6}	1.910×10^{-7}	6.296×10^{-7}

Table 6-2: Parameter values μ and σ (window size = 101)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	376	99.95	85.37
	IVC1 (SF)	0	0	18	2200	0	99.95	99.14
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-3: Confusion matrix, (window size = 81)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0818	0.1310	0.0723	0.1135
	σ	5.078×10^{-6}	2.581×10^{-5}	3.030×10^{-6}	9.929×10^{-6}
Fault	μ	0.0180	0.0187	0.0097	0.0231
	σ	6.442×10^{-7}	2.392×10^{-6}	1.038×10^{-7}	7.834×10^{-7}

Table 6-4: Parameter values μ and σ (window size= 81)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	52	99.95	97.65
	IVC1 (SF)	0	0	11	2200	0	99.95	99.46
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-5: Confusion matrix, (window size = 61)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0907	0.1455	0.0842	0.1227
	σ	6.986×10^{-6}	3.721×10^{-5}	2.141×10^{-6}	1.296×10^{-5}
Fault	μ	0.0197	0.0184	0.0092	0.0228
	σ	9.600×10^{-7}	2.779×10^{-6}	6.786×10^{-8}	7.724×10^{-7}

Table 6-6: Parameter values μ and σ (window size= 61)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	4	2200	0	99.95	99.77
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-7: Confusion matrix, (window size = 41)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0995	0.1662	0.0906	0.1340
	σ	2.213×10^{-6}	4.286×10^{-5}	3.094×10^{-6}	1.931×10^{-5}
Fault	μ	0.0200	0.0171	0.0089	0.0221
	σ	9.957×10^{-7}	2.788×10^{-6}	4.524×10^{-8}	6.226×10^{-7}

Table 6-8: Parameter values μ and σ (window size = 41)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	14	2200	0	99.95	99.32
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-9: Confusion matrix, (window size = 21)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1140	0.1846	0.0951	0.1492
	σ	9.982×10^{-6}	1.590×10^{-4}	2.336×10^{-6}	1.656×10^{-5}
Fault	μ	0.0178	0.0152	0.0088	0.0380
	σ	8.655×10^{-7}	2.650×10^{-6}	3.316×10^{-8}	0.0016

Table 6-10: Parameter values μ and σ (window size = 21)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.9
	EVC1 (SF)	0	2200	0	0	0	99.95	99.9
	EVC2 (SF)	293	0	1907	0	0	86.64	86.64
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-11: Confusion matrix, (window size = 3)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1258	0.1476	0.0753	0.1489
	σ	1.927×10^{-5}	0.0017	3.152×10^{-6}	1.140×10^{-4}
Fault	μ	0.0141	0.0133	0.0075	0.0193
	σ	8.405×10^{-7}	1.411×10^{-6}	3.415×10^{-8}	7.451×10^{-7}

Table 6-12: Parameter values μ and σ (window size = 3)

6.2.2 Tuning Threshold Value

The next parameter we explore is the threshold value used for impact detection in the feature extraction method (see section 5.2.2). Manual tuning with $N_{ws} = 41$ was done to find an appropriate threshold value for our application. Results are given for different threshold values in Table 6-13 through Table 6-18. It can be seen that the best classification results occur when $s_f = 1$ which gives a worst case of $DA_i = 99.95\%$ and $CA_i = 99.95\%$ (see Table 6-15). For the reasons explained in section 5.2.2, both low and high threshold values degrade classification results substantially by either missing or falsely detecting impact regions. For the remainder of this thesis $s_f = 1$ will be used in the FDD methods.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2144	0	56	0	0	97.41	97.41
	EVC1 (SF)	0	2200	219	0	0	99.95	90.91
	EVC2 (SF)	386	0	1814	0	0	82.42	82.42
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	317	0	2200	99.95	87.37

Table 6-13: Confusion matrix, ($s_f = 2$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0890	0.0899	0.0818	0.1316
	σ	9.464×10^{-5}	6.726×10^{-4}	2.260×10^{-4}	6.429×10^{-4}
Fault	μ	0.0150	0.0296	0.0098	0.0212
	σ	3.181×10^{-6}	3.081×10^{-4}	1.130×10^{-6}	1.406×10^{-6}

Table 6-14: Parameter values μ and σ ($s_f = 2$)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-15: Confusion matrix, ($s_f = 1$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0999	0.1455	0.0810	0.1332
	σ	1.030×10^{-4}	0.0011	2.070×10^{-4}	7.580×10^{-4}
Fault	μ	0.0170	0.0169	0.0098	0.0212
	σ	5.411×10^{-6}	4.839×10^{-5}	1.130×10^{-6}	1.406×10^{-6}

Table 6-16: Parameter values μ and σ ($s_f = 1$)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	1692	0	508	0	0	76.87	76.87
	EVC1 (SF)	0	2200	185	0	0	99.95	92.20
	EVC2 (SF)	494	0	1706	0	0	77.51	77.51
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	592	0	2200	99.95	78.77

Table 6-17: Confusion matrix, ($s_f = 0.5$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0842	0.0816	0.0806	0.1222
	σ	6.521×10^{-5}	6.899×10^{-4}	2.283×10^{-4}	5.249×10^{-4}
Fault	μ	0.0193	0.0345	0.0098	0.0212
	σ	3.227×10^{-6}	3.081×10^{-4}	1.130×10^{-6}	1.406×10^{-6}

Table 6-18: Parameter values μ and σ ($s_f = 0.5$)

6.2.3 Tuning the Training Size

From the DAQ system described in Chapter 3, we have 2200 engine cycle data sets for each SF fault case (for a total of $5 \times 2200 = 11000$ data sets). To maximize testing and minimize computation time the smallest training size N_t that gives a reliable success rate should be used.

Also note that although training sets of below 30 may give good results, due to the assumption of normal distribution we should not go below 30 training data sets since the normal distribution assumption should consist of at least 30 samples (Montgomery & Runger, 2007).

Manual tuning with $N_{ws} = 41$ and $s_f = 1$ was done to find an appropriate training size for our application. Results for different training sizes are given in Table 6-19 through Table 6-26. It can be seen that using more training cycles produces better classification results. A training size of 40 cycles provides very good results, this is the lowest training size that produces a worst case of $DA_i = 99.95\%$ and $CA_i = 99.95\%$, and will be used for the remainder of this thesis.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-19: Confusion matrix, ($N_t = 50$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0997	0.1628	0.0913	0.1328
	σ	2.469×10^{-6}	8.290×10^{-5}	5.749×10^{-6}	2.716×10^{-5}
Fault	μ	0.0193	0.0174	0.0090	0.0222
	σ	2.720×10^{-6}	2.705×10^{-6}	3.786×10^{-8}	5.614×10^{-7}

Table 6-20: Parameter values μ and σ ($N_t = 50$)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-21: Confusion matrix, ($N_t = 40$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0995	0.1662	0.0906	0.1340
	σ	2.213×10^{-6}	4.286×10^{-5}	3.094×10^{-6}	1.931×10^{-5}
Fault	μ	0.0200	0.0171	0.0089	0.0221
	σ	9.957×10^{-7}	2.788×10^{-6}	4.524×10^{-8}	6.226×10^{-7}

Table 6-22: Parameter values μ and σ ($N_t = 40$)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	52	2200	0	99.95	97.65
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-23: Confusion matrix, ($N_t = 30$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0992	0.1689	0.0904	0.1335
	σ	2.170×10^{-6}	2.702×10^{-5}	3.666×10^{-6}	2.379×10^{-5}
Fault	μ	0.0203	0.0166	0.0089	0.0221
	σ	4.615×10^{-7}	2.703×10^{-6}	5.395×10^{-8}	6.582×10^{-7}

Table 6-24: Parameter values μ and σ ($N_t = 30$)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	316	2200	0	99.95	87.41
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-25: Confusion matrix, ($N_t = 20$)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0987	0.1718	0.0916	0.1309
	σ	2.267×10^{-6}	5.352×10^{-6}	6.938×10^{-7}	1.082×10^{-5}
Fault	μ	0.0203	0.0157	0.0089	0.0218
	σ	4.775×10^{-7}	1.593×10^{-6}	7.325×10^{-8}	3.892×10^{-7}

Table 6-26: Parameter values μ and σ ($N_t = 20$)

6.3 FDD Testing

6.3.1 Comparison of Engine Speeds

Multiple engine speeds (ES) were used to test the generality of our FDD method. A medium engine speed of 2000 RPM has been the engine speed used thus far. A low ES of 1200 RPM and high ES of 3500 RPM were tested for comparison. FDD training was performed at each speed with $N_{ws} = 41, s_f = 1$, and $N_t = 40$. The results are given in Figure 6-3; and Table 6-27 through Table 6-32.

Lower ES give lower impacts ($\mu_{nom,EVC2} = 0.0295$ from Table 6-28) however the detection method still works well at 1200 RPM, with a worst case of $CA_3 = 98.18\%$ and $CA_4 = 93.66\%$. Higher ES give larger impacts ($\mu_{nom,EVC2} = 0.2297$ from Table 6-32) but also have more noise and unwanted vibrations however the method still works fairly well at 3500 RPM, with a worst case DA and CA of $DA_1 = 94.87\%$ and $CA_5 = 87.48\%$ An engine speed of 2000 RPM gives the best results as originally assumed, with a worst case of $DA_i = 99.95\%$ and $CA_i = 99.95\%$; this is what we will use for the majority of our results and discussion in this thesis. These tests show that even at the extreme ends of the engine speed spectrum our FDD method still works well. The FDD is trained for each engine speed so operating over a continuous range of speeds would require extensive training and a method for interpolating the parameters.

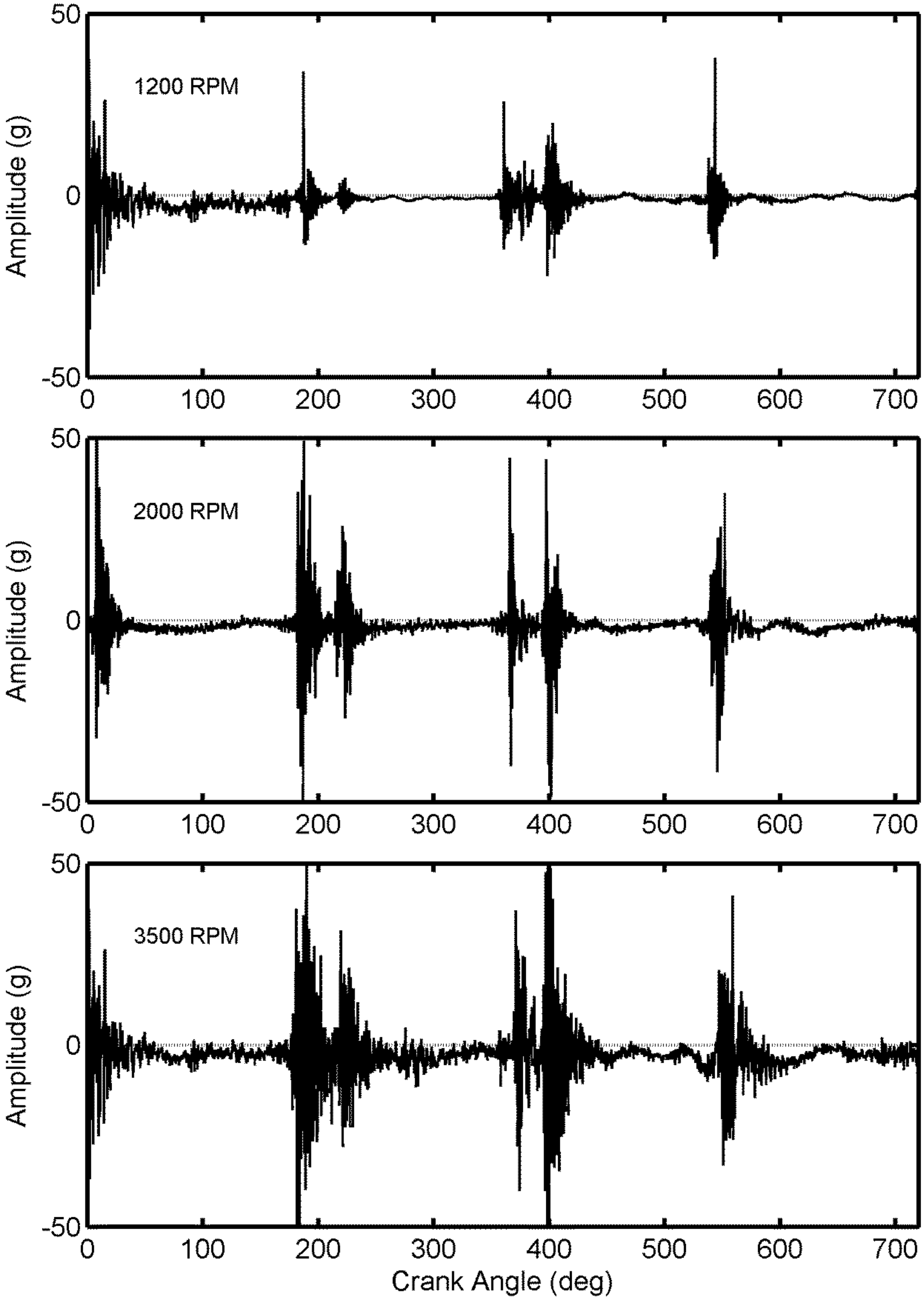

Figure 6-3: Vibration signal for various engine speeds

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	39	60	2161	0	0	98.18	95.58
	IVC1 (SF)	0	148	0	2200	0	99.95	93.66
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-27: Confusion matrix, (engine speed = 1200 RPM)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0148	0.0295	0.0220	0.0272
	σ	8.276×10^{-7}	9.879×10^{-5}	1.414×10^{-5}	1.184×10^{-5}
Fault	μ	0.0093	0.0064	0.0057	0.0062
	σ	1.445×10^{-6}	1.636×10^{-6}	1.218×10^{-6}	4.797×10^{-7}

Table 6-28: Parameter values μ and σ (engine speed = 1200 RPM)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-29: Confusion matrix, (engine speed = 2000 RPM)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0995	0.1662	0.0906	0.1340
	σ	2.213×10^{-6}	4.286×10^{-5}	3.094×10^{-6}	1.931×10^{-5}
Fault	μ	0.0200	0.0171	0.0089	0.0221
	σ	9.957×10^{-7}	2.788×10^{-6}	4.524×10^{-8}	6.226×10^{-7}

Table 6-30: Parameter values μ and σ (engine speed = 2000 RPM)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2088	5	1	106	0	94.87	94.87
	EVC1 (SF)	36	2164	0	0	0	98.32	98.32
	EVC2 (SF)	0	9	2200	0	0	99.95	99.55
	IVC1 (SF)	0	16	41	2200	0	99.95	97.43
	IVC2 (SF)	0	6	291	17	2200	99.95	87.48

Table 6-31: Confusion matrix, (engine speed = 3500 RPM)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1257	0.2297	0.1008	0.2981
	σ	3.093×10^{-4}	6.473×10^{-4}	1.225×10^{-4}	2.186×10^{-4}
Fault	μ	0.0283	0.0474	0.0634	0.0593
	σ	1.976×10^{-5}	4.486×10^{-4}	6.453×10^{-5}	1.407×10^{-5}

Table 6-32: Parameter values μ and σ (engine speed = 3500 RPM)

6.3.2 Sensitivity to Sampling Rate

In this section we explore the effects of changing the sampling rate parameter. In Chapter 4 it was concluded that a 48 kHz sampling rate was adequate for our application. This section gives results of different sampling rates to help confirm or disprove our original conclusion. Results are given for different sampling rates in Table 6-33 through Table 6-42 with $N_{ws} = 41, s_f = 1$, $N_t = 40$, and $ES = 2000RPM$. It can be seen that as the sampling rate is reduced, the DA and CA are reduced. This is due to a combination of the degradation of the signal representation at lower sampling, as well as increased variation at lower sampling rates $\mu_{nom,EVC2}$=4.286×10^{-5} when $f_s = 48kHz$, and $\mu_{nom,EVC2}$=1.249×10^{-4} when $f_s = 6kHz$. This agrees with our comments in section 4.2.1. Obviously due to limitations we cannot have an infinite sampling rate, these results show that a sampling rate of 32 kHz gives fairly reliable results with a worst case of $DA_3 = 99.73\%$ and $CA_4 = 99.10\%$. A sampling rate as low as 12 kHz still gives adequate results with a worst case of $DA_1 = 99.68\%$ and $CA_4 = 85.77\%$ which shows how robust our FDD method is with respect to the sampling rate. As expected, the highest sampling rate of 48 kHz gives the best results with worst case of $DA_i = 99.95\%$ and $CA_i = 99.95\%$. This sampling rate will be used for the remainder of this thesis.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-33: Confusion matrix, (sample rate = 48 kHz)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.0995	0.1662	0.0906	0.1340
	σ	2.213×10^{-6}	4.286×10^{-5}	3.094×10^{-6}	1.931×10^{-5}
Fault	μ	0.0200	0.0171	0.0089	0.0221
	σ	9.957×10^{-7}	2.788×10^{-6}	4.524×10^{-8}	6.226×10^{-7}

Table 6-34: Parameter values μ and σ (sample rate = 48 kHz)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	5	0	2195	0	0	99.73	99.73
	IVC1 (SF)	0	0	19	2200	0	99.95	99.10
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-35: Confusion matrix, (sample rate = 32 kHz)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1027	0.1692	0.0846	0.1375
	σ	4.195×10^{-6}	4.139×10^{-5}	5.021×10^{-6}	5.255×10^{-5}
Fault	μ	0.0180	0.0174	0.0087	0.0201
	σ	9.268×10^{-7}	4.883×10^{-6}	2.716×10^{-8}	4.998×10^{-7}

Table 6-36: Parameter values μ and σ (sample rate = 32 kHz)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	23	99.95	98.92
	IVC1 (SF)	0	0	78	2200	0	99.95	96.53
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-37: Confusion matrix, (sample rate = 24 kHz)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1197	0.1970	0.0769	0.1516
	σ	1.466×10^{-5}	8.085×10^{-5}	4.943×10^{-6}	5.272×10^{-5}
Fault	μ	0.0184	0.0176	0.0091	0.0215
	σ	2.997×10^{-6}	4.123×10^{-6}	8.372×10^{-8}	1.200×10^{-6}

Table 6-38: Parameter values μ and σ (sample rate = 24 kHz)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2194	6	0	0	0	99.68	99.68
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	133	99.95	94.26
	IVC1 (SF)	0	0	351	2200	13	99.95	85.77
	IVC2 (SF)	0	36	0	0	2200	99.95	98.35

Table 6-39: Confusion matrix, (sample rate = 12 kHz)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1075	0.1738	0.0782	0.1390
	σ	1.560×10^{-5}	6.704×10^{-5}	4.267×10^{-6}	1.0777e-04
Fault	μ	0.0166	0.0167	0.0083	0.0260
	σ	3.446×10^{-6}	2.200×10^{-6}	9.693×10^{-8}	1.858×10^{-6}

Table 6-40: Parameter values μ and σ (sample rate = 12 kHz)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	1978	44	178	0	0	89.87	89.87
	EVC1 (SF)	0	2200	62	0	0	99.95	97.22
	EVC2 (SF)	0	0	2200	0	395	99.95	84.75
	IVC1 (SF)	0	0	1200	2200	8	99.95	64.54
	IVC2 (SF)	535	54	150	0	1613	73.28	68.58

Table 6-41: Confusion matrix, (sample rate = 6 kHz)

		EVC1	EVC2	IVC1	IVC2
Nominal	μ	0.1263	0.2232	0.0564	0.1436
	σ	1.828×10^{-5}	1.249×10^{-4}	6.699×10^{-6}	2.658×10^{-4}
Fault	μ	0.0174	0.0187	0.0082	0.0242
	σ	1.282×10^{-6}	3.148×10^{-6}	3.629×10^{-7}	1.008×10^{-6}

Table 6-42: Parameter values μ and σ (sample rate = 6 kHz)

6.3.3 Fault Detection Sensitivity

FDD methods are highly dependant on the severity of the fault in question. A good FDD method should be able to detect small faults in their early stages so preventative maintenance can be done to prevent further damage to the system. As described in Chapter 3, the engine manufacturer specifies that a valve spring should be replaced after a 3mm reduction in the spring's free length. A good FDD method should detect a faulty valve spring far before this limit is reached. Several SF severities were tested to find the fault level that can be detected with a reasonable success rate, with $N_{ws} = 41, s_f = 1$, $N_t = 40$, $ES = 2000RPM$, and $f_s = 48kHz$. Although the SFs were only seeded with IVC1 it is believed that similar results would be obtained with the other valves. Table 6-43 shows that a 0.3mm SF is quite difficult to detect, this is a fairly small fault as it is only 10% of the manufacturers specified SF so it may be unreasonable to attempt to detect such a small fault. Larger faults were tested and the results are compared in Table 6-44 through Table 6-46. The standard SF severity used in this thesis is a valve spring with a 0.5mm deformation. When deformation severity is reduced (0.4mm) the DA and CA decrease as would be expected. When the deformation severity increases (0.55mm) the DA and CA improves. The deformation of 0.5mm was chosen as the standard SF because this is when fault detection success rates become quite reliable. Since this fault level is much lower than the manufacturers specified fault (3.0mm deformation), a 0.5mm fault detection is still a relatively small fault, and early SF detection has been successfully achieved.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	1639	13	5	543	0	74.47	74.47
	IVC1 (SF)	179	0	0	2021	0	91.82	91.82

Table 6-43: Confusion matrix, (small IVC1 0.3mm fault)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	1675	0	0	525	0	76.10	76.10
	IVC1 (SF)	24	0	0	2176	0	98.86	98.86

Table 6-44: Confusion matrix, (small IVC1 0.4mm fault)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95

Table 6-45: Confusion matrix, (0.5mm fault)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	% DA	% CA
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95

Table 6-46: Confusion matrix, (large 0.55mm fault)

6.3.4 Complex Cases: Simultaneous Faults, Distinguishing Fault Types

To test the robustness and generality of our method, a number of more complex scenarios were tested using the fixed set of parameters given in Table 6-47.

FDD Parameter Symbol	FDD Parameter Description	Value Determined
N_{ws}	Window size	41
s_f	Threshold scaling value	1
N_t	Training size	40
ES	Engine speed	2000 RPM
f_s	Sampling rate	48 kHz
F_L	Spring fault severity level	0.5mm

Table 6-47: Set of parameters used for testing

These tests were performed on 440 engine cycles, rather than 2200 engine cycles as done previously to reduce computation time and data storage requirements. First, the case of simultaneous faults on different valves was tested. The results are given in Table 6-48. It can be seen that our FDD method can successfully detect multiple faults on different valves, a problem that previous research failed to solve (M. Desbazeille et al., 2010). Our FDD method successfully detects multiple faults with a worst case of $DA_i = 99.8\%$ and $CA_4 = 97.8\%$.

		Predicted Class									
		Nom	EVC1	EVC2	IVC1	IVC2	EVC2 IVC2	EVC2 IVC2 EVC1	EVC2 IVC2 EVC1 IVC1	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	440	2	0	0	0	0	99.8	99.3
	IVC1	0	0	9	440	0	0	0	0	99.8	97.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC2 IVC2	0	0	0	0	0	440	0	0	99.8	99.8
	EVC2 IVC2 EVC1	0	0	0	0	0	0	440	0	99.8	99.8
	EVC2 IVC2 EVC1 IVC1	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-48: Multiple faults on different valves

It is important to note that thus far our method has only been detecting one type of valve fault, and on which valve it occurred. It has not been distinguishing between types of valve faults. To further test our FDD method, some different types of valve fault were seeded. First, valve SCFs were tested, and the results are given in Table 6-49. These results show that the developed FDD method works very well not only for SFs, but also for SCFs, with all $DA_i = 99.8\%$ and $CA_i = 99.8\%$.

		Predicted Class						
		Nom	EVC1 (SCF)	EVC2 (SCF)	IVC1 (SCF)	IVC2 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	99.8	99.8
	EVC1 (SCF)	0	440	0	0	0	99.8	99.8
	EVC2 (SCF)	0	0	440	0	0	99.8	99.8
	IVC1 (SCF)	0	0	0	440	0	99.8	99.8
	IVC2 (SCF)	0	0	0	0	440	99.8	99.8

Table 6-49: Confusion matrix, (small clearance fault)

Next, the SFs were tested against the SCFs to see if the FDD method could distinguish between the two valve fault types. This is a particularly interesting case because both the spring faults (SF) and the small clearance faults (SCF) have very similar signal characteristics. Both fault types result in diminished valve impact amplitude. For this reason it would be expected that distinguishing between the two similar valve faults may be difficult and lead to poor classification results. Surprisingly, the developed FDD method does a fairly good job distinguishing between such similar faults. The results can be seen in Table 6-50. From these results it can be seen that most of the classification errors are from misdiagnosing a fault type; however the detection of a fault to a specific valve still remains very successful. For example, when the actual class is an EVC1-SCF fault it is successfully classified as an EVC1-SCF fault for 391 tests and misclassified as an EVC1-SF fault for 49 tests; in both cases it is detecting a fault at EVC1. This test illustrates the important difference between detection and classification as the worst case CA is $CA_7 = 88.7\%$ however all $DA_i = 99.8\%$ Even with such similar valve fault types the CA is still surprisingly accurate.

		Predicted Class										
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	IVC2 (SCF)	EVC1 (SCF)	IVC1 (SCF)	EVC2 (SCF)	DA%	CA%
Actual Class	Nominal	440	0	0	0	0	0	0	0	0	99.8	99.8
	EVC1 (SF)	0	440	0	0	0	0	0	0	0	99.8	99.8
	EVC2 (SF)	0	0	440	0	0	0	0	0	0	99.8	99.8
	IVC1 (SF)	0	0	0	440	0	0	0	0	0	99.8	99.8
	IVC2 (SF)	0	0	0	0	435	5	0	0	0	99.8	98.6
	IVC2 (SCF)	0	0	0	0	29	411	0	0	0	99.8	93.2
	EVC1 (SCF)	0	49	0	0	0	0	391	0	0	99.8	88.7
	IVC1 (SCF)	0	0	0	0	0	0	0	440	0	99.8	99.8
	EVC2 (SCF)	0	0	22	0	0	0	0	0	418	99.8	94.8

Table 6-50: Confusion matrix, distinguishing SF and SCF

Further complication was tested by seeding multiple fault types on the engine simultaneously. The FDD method must now distinguish between fault types as well as multiple simultaneous faults on different valves. Table 6-51 gives the experimental results of the multiple faults present on multiple valves at the same time. This table shows that the addition of multiple faults on different valves does not worsen the results of the FDD drastically. It still appears that the major difficulty of the method is distinguishing between the similar fault types and even with this difficulty the FDD still provides DA of $DA_i = 99.8\%$ and a worst case CA of $CA_{12} = 82.1\%$.

		Predicted Class													
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	IVC2 (SCF)	EVC1 (SCF)	IVC1 (SCF)	EVC2 (SCF)	EVC2 (SF) IVC2 (SCF)	EVC2 (SF) IVC2 (SCF) EVC1 (SCF)	EVC2 (SF) IVC2 (SCF) EVC1 (SCF) IVC1 (SCF)	DA%	CA%
Actual Class	Nominal	440	0	0	0	0	0	0	0	0	0	0	0	99.8	99.8
	EVC1(SF)	0	440	0	0	0	0	0	0	0	0	0	0	99.8	99.8
	EVC2(SF)	0	0	440	0	0	0	0	0	0	0	0	0	99.8	99.8
	IVC1(SF)	0	0	0	440	0	0	0	0	0	0	0	0	99.8	99.8
	IVC2(SF)	0	0	0	0	435	5	0	0	0	0	0	0	99.8	98.6
	IVC2(SCF)	0	0	0	0	29	411	0	0	0	0	0	0	99.8	93.2
	EVC1(SCF)	0	49	0	0	0	0	391	0	0	0	0	0	99.8	88.7
	IVC1(SCF)	0	0	0	0	0	0	0	440	0	0	0	0	99.8	99.8
	EVC2(SCF)	0	0	0	0	0	0	0	0	440	0	0	0	99.8	99.8
	EVC2(SF) IVC2(SCF)	0	0	0	0	29	0	0	0	0	411	0	0	99.8	93.2
	EVC2(SF) IVC2(SCF) EVC1(SCF)	0	45	0	0	21	0	0	0	0	0	374	0	99.8	84.8
	EVC2(SF) IVC2(SCF) EVC1(SCF) IVC1(SCF)	0	41	0	0	22	0	0	0	0	0	0	362	99.8	82.1

Table 6-51: Multiple faults on different valves

The final test to challenge our FDD method was seeding multiple faults on the same valve. This is the most challenging scenario and has not been attempted by other researchers based on our literature review. This test contains a spring fault and clearance fault located on the same valve. Both LCFs and SCFs were tested on the same valve as a SF. The case of a LCF+SF fault is of particular interest because the two faults act in an opposing manner. The SF tends to decrease the impact amplitude, while the LCF tends to increase the impact amplitude, making fault detection quite difficult for this specific case. The case of a SCF+SF fault should not be as difficult as these two faults act in a similar manner, both decreasing the impact amplitude. Table 6-52 shows the results of testing multiple faults on a single valve. From these results it can easily be seen that the EVC1-LCF+SF combination caused most of the difficulties, as expected. The worst case DA and CA is quite low with $DA_1 = 63.7\%$ and $CA_1 = 63.7\%$. The reason behind this is that the SF is reducing the valve impact but at the same time the LCF is increasing the valve impact; the net result is a valve impact that looks nominal when in fact two faults are present. This complex case was described in Section 5.2.3, and an impact segmentation method was proposed as the solution. From Table 6-53 we can see that our impact segmentation method improves our results. With $sn = 2$ we obtain a worst case $DA_i = 99.8\%$ of $CA_4 = 87.0\%$ which is a significant improvement over the 63.7% obtained in Table 6-52 without the segmentation method. Further segmentation may further improve results, however diminishing returns may occur after exceeding some sn value, as shown in Table 6-54 and Table 6-55.

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	281	0	0	0	0	0	159	0	63.7	63.7
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	440	0	0	0	174	0	99.8	71.5
	IVC1	0	0	0	440	0	0	54	0	99.8	88.9
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8

	EVC1(SF)
	EVC1(SCF)
		0	0	0	0	0	440	0	0	99.8	99.8

	EVC1(SF)
	EVC1(LCF)
		13	0	0	0	0	0	427	0	96.8	96.8

	IVC1(SF)
	IVC1(SCF)
		0	0	0	0	0	0	0	440	99.8	99.8

Table 6-52: Multiple faults on a single valve

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	440	0	0	0	0	31	99.8	93.2
	IVC1	0	0	58	434	0	0	0	6	99.8	87.0
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1(SF) EVC1(SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1(SF) EVC1(LCF)	0	0	0	0	0	0	440	0	99.8	99.8
	IVC1(SF) IVC1(SCF)	0	0	0	2	0	0	0	438	99.8	99.3

Table 6-53: Multiple faults on a single valve (segmentation method, sn=2)

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	436	0	0	0	4	0	99.8	98.9
	IVC1	0	0	0	440	0	0	0	0	99.8	99.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1(SF) EVC1(SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1(SF) EVC1(LCF)	0	0	0	0	0	0	440	0	99.8	99.8
	IVC1(SF) IVC1(SCF)	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-54: Multiple faults on a single valve (segmentation method, sn=3)

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	4	0	436	0	0	0	0	0	98.9	98.9
	IVC1	0	0	0	440	0	0	0	0	99.8	99.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1(SF) EVC1(SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1(SF) EVC1(LCF)	0	0	0	0	0	0	440	0	99.8	99.8
	IVC1(SF) IVC1(SCF)	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-55: Multiple faults on a single valve (segmentation method, sn=4)

6.4 Comparison with Other Classification Methods

All previous results shown were found using the NB classification method. In this section we will compare the NB method to other well known classification methods. A quantitative comparison will be done by comparing the accuracies DA_i and CA_i as well as the execution times for each classification method. The execution time for the NB method is given in Table 6-56 below for 11000 engine cycles of SFs. The total FDD execution time is separated into feature extraction time, training time, and testing time. The feature extraction time is the same for all classification methods, and the average of 4.5s from Table 6-56 will be used. Training, testing, and total FDD time will change for each classification method and will be used for comparison.

	Feature Extraction Time (11000 Cycles)	Training Time (200 Cycles)	Testing Time (11000 Cycles)	Total FDD Time (11000 Cycles)
Test #1	4.524	0.0156	0.0156	4.555
Test #2	4.290	0.0156	0.0312	4.337
Test #3	4.680	0.0312	0.0624	4.774
Test #4	4.602	0.0156	0.0156	4.633
Test #5	4.324	0.0156	0.0312	4.371
Average	**4.5**	**0.019**	**0.031**	**4.5**

Table 6-56: NB execution time

6.4.1 Artificial Neural Networks

ANNs were chosen for this application since they are one of the most commonly used classification algorithms and have seen application with FDD of ICEs as discussed in Chapter Chapter 2:. This section will discuss the experimental fault classification results obtained using the ANN described in Section 5.4.1. Results can be found in Table 6-57 through Table 6-65.

Table 6-57 shows that the ANN method seems to work well with a worst case of $DA_i = 99.95\%$ and $CA_5 = 99.91\%$. This test was done using the same 40 training cycles used for the NB method per class, for a total of $N_t = 5 \times 40 = 200$ training cycles. This test took 153 epochs for the training, which took 10.64s of training time and 0.078s of testing time.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	1	2199	99.95	99.91

Table 6-57: ANN results, 153 epochs, 200 training cycles

However, further testing with the ANN method gave some surprising results. Each time the network is trained random estimates are used to initialize the node weights. These weights are then updated every epoch as described in Section 5.4.1. It was found that the ANN training was very sensitive to the initial values of the weights, and was quite unreliable when small training sizes were used. The exact same data was re-trained and Table 6-58 shows the surprising results. These results show that the exact same data can provide very poor results if the initial estimates are poor, and has a worst case of $DA_4 = 42.71\%$ and $CA_4 = 0\%$. In general ANN reliability improves as training size and epochs are increased, however these will also increase execution time. Table 6-59 through Table 6-61 show some more tests demonstrating the unreliable results of ANNs with 200 training cycles.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	1261	0	643	0	297	42.71	0
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-58: ANN results, 141 epochs, 200 training cycles

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	4	2196	0	99.95	99.77
	IVC2 (SF)	0	0	0	1	2199	99.95	99.91

Table 6-59: ANN results, 81 epochs, 200 training cycles

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	1714	440	46	0	0	22.08	2.09
	IVC1 (SF)	2200	0	0	0	0	0	0
	IVC2 (SF)	2166	34	0	0	0	1.54	0

Table 6-60: ANN results, 58 epochs, 200 training cycles

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	950	281	466	502	0	56.75	0

Table 6-61: ANN results, 50 epochs, 200 training cycles

Table 6-62 shows the execution times for the ANN method for various numbers of epochs when the training is set at 200 cycles. It can be seen that as the number of epochs increases, the training time tends to increase.

	# of Epochs	Training Time	Testing Time
Test #1	153	10.64	0.078
Test #2	141	8.56	0.094
Test #3	81	6.13	0.094
Test #4	58	5.79	0.038
Test #5	50	4.52	0.078
Average	**97**	**7.13**	**0.076**

Table 6-62: ANN execution times, 200 training cycles

Table 6-63 through Table 6-65 show the classification results of the ANN method when the training is set to 1000 cycles for various iteration values. From these results it can be seen that the increase in training size has improved our classification results, with a lowest classification accuracy of $CA_5 = 99.91\%$. Increasing the training size to 1000 cycles has also made the training produce good results more consistently. However it is still possible that poor initial weight estimates can lead to poor results. Also, the increase of training size leads to a moderate increase in execution time as can be seen in comparing Table 6-66 with Table 6-62.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	1	2199	99.95	99.91

Table 6-63: ANN results, 72 epochs, 1000 training cycles

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	1	2199	99.95	99.91

Table 6-64: ANN results, 65 epochs, 1000 training cycles

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	1	2199	99.95	99.91

Table 6-65: ANN results, 55 epochs, 1000 training cycles

	# of Epochs	**Training Time**	**Testing Time**
Test #1	72	11.11	0.078
Test #2	65	10.55	0.047
Test #3	60	9.29	0.062
Test #4	55	8.62	0.047
Test #5	48	8.13	0.047
Average	**60**	**9.54**	**0.056**

Table 6-66: ANN execution times, 1000 training cycles

6.4.2 Decision Tree

This section will discuss the experimental fault classification results obtained using the DT method described in Section 5.4.2. The results can be found in Table 6-67 and Table 6-68. The worst case accuracies using the DT method are $DA_i = 99.95\%$ and $CA_4 = 98.96\%$ and it has an average training time of 0.0156s and testing time of 0.0156s. These results were for 2200 cycles per class using the same 40 cycles for training as used with NB. These results are comparable to the NB method. Although the results are slightly inferior, we believe that the DT method is an acceptable classification method for our application.

Further comparison was done by implementing the DT method on the difficult case of multiple simultaneous faults on the same valve. The segmentation method was used for this complex case with $sn = 3$, and gave worst case results of $DA_2 = 95.0\%$ and $CA_2 = 94.8\%$ as shown in Table 6-70. Once again these DT results are slightly inferior in comparison with the NB results of Table 6-54, however still acceptable as a FDD classification method.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	22	2178	0	99.95	98.96
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-67: Decision tree confusion matrix

	Training Time	**Testing Time**
Test #1	0.0156	0.0156
Test #2	0.0156	0.0156
Test #3	0.0156	0.0156
Test #4	0.0156	0.0156
Test #5	0.0156	0.0156
Average	**0.016**	**0.016**

Table 6-68: Decision tree execution times

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	392	0	0	0	48	0	0	0	88.9	88.9
	EVC1	0	422	0	0	0	18	0	0	99.8	95.7
	EVC2	0	0	420	20	0	0	0	0	99.8	95.5
	IVC1	0	0	81	336	23	0	0	0	99.8	76.2
	IVC2	8	0	0	2	430	0	0	0	98.0	97.5
	EVC1 (SF) EVC1 (SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1 (SF) EVC1 (LCF)	0	0	0	80	0	0	360	0	99.8	81.6
	IVC1 (SF) IVC1 (SCF)	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-69: Simultaneous faults on same valve, decision tree with sn = 2

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	428	0	0	0	1	0	11	0	97.1	97.1
	EVC1	0	418	0	0	0	22	0	0	95.0	94.8
	EVC2	0	0	440	0	0	0	0	0	99.8	99.8
	IVC1	0	0	0	440	0	0	0	0	99.8	99.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1 (SF) EVC1 (SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1 (SF) EVC1 (LCF)	6	0	0	0	0	0	434	0	98.4	98.4
	IVC1 (SF) IVC1 (SCF)	0	0	0	0	10	0	0	430	99.8	97.5

Table 6-70: Simultaneous faults on same valve, decision tree with sn = 3

6.4.3 k-Nearest Neighbour

The k-NN method was introduced and described in Section 5.4.3. This section will discuss the experimental fault classification results obtained using k-NN. These results were for 2200 cycles per class using the same 40 cycles for training as used with NB. Table 6-71 displays the results with $k = 1$. Increasing the k value was found to either have no effect, or to worsen the classification results. For example, Table 6-72 shows the worse classification results that were produced with $k = 15$. Table 6-73 gives the execution times of the k-NN method when $k = 1$. The worst case DA and CA using the k-NN method when $k = 1$ is $DA_i = 99.95\%$ and $CA_i = 99.95\%$, and has an average testing execution time of 0.9422s. These results are comparable to the NB and DT methods, with the exception of the large increase in testing execution time. Although the testing time may be comparatively poor, it may still be an acceptable classification method depending on the application timing constraints. From these results we conclude that the k-NN method with $k = 1$ is a suitable classification method for our particular application.

Further comparison was once again done by implementing the k-NN method on the difficult case of multiple simultaneous faults on the same valve. The segmentation method was used for this complex case with $sn = 3$, and gave worst case results of $DA_i = 99.8\%$ and $CA_i = 99.8\%$ as shown in Table 6-70. The k-NN results are equally as good as the NB results of Table 6-54, however requires significantly more testing time than the other classification methods. Depending on the timing constraints, k-NN is still very acceptable as a FDD classification method for this our application.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2200	0	0	0	0	99.95	99.95
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-71: k-NN confusion matrix (k=1)

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	1974	51	0	159	16	89.69	89.69
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2200	0	0	99.95	99.95
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.91

Table 6-72: k-NN confusion matrix (k=15)

	Training Time	**Testing Time**
Test #1	0	0.9360
Test #2	0	0.9360
Test #3	0	0.9516
Test #4	0	0.9516
Test #5	0	0.9360
Average	**0**	**0.94**

Table 6-73: k-NN execution times (k=1)

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	440	0	0	0	0	0	99.8	99.8
	IVC1	0	0	0	440	0	0	0	0	99.8	99.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1 (SF) EVC1 (SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1 (SF) EVC1 (LCF)	0	0	0	0	0	0	440	0	99.8	99.8
	IVC1 (SF) IVC1 (SCF)	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-74: Simultaneous faults on same valve, k-NN (*k*=1) with *sn* = 2

		Predicted Class									
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	EVC1 (SF) EVC1 (SCF)	EVC1 (SF) EVC1 (LCF)	IVC1 (SF) IVC1 (SCF)	DA %	CA %
Actual Class	Nominal	440	0	0	0	0	0	0	0	99.8	99.8
	EVC1	0	440	0	0	0	0	0	0	99.8	99.8
	EVC2	0	0	440	0	0	0	0	0	99.8	99.8
	IVC1	0	0	0	440	0	0	0	0	99.8	99.8
	IVC2	0	0	0	0	440	0	0	0	99.8	99.8
	EVC1 (SF) EVC1 (SCF)	0	0	0	0	0	440	0	0	99.8	99.8
	EVC1 (SF) EVC1 (LCF)	0	0	0	0	0	0	440	0	99.8	99.8
	IVC1 (SF) IVC1 (SCF)	0	0	0	0	0	0	0	440	99.8	99.8

Table 6-75: Simultaneous faults on same valve, k-NN (k=1) with *sn* = 3

6.4.4 Linear Discriminant Analysis

The LDA method was introduced and described in Section 5.4.4. This section will discuss the experimental fault classification results obtained using LDA. Results can be found in Table 6-76. The worst case CA and DA using the LDA method is $DA_1 = 98.46\%$ and $CA_3 = 97.77\%$ with an average execution time of 0.0374s. Again, these results were for 2200 cycles per class using the same 40 cycles for training as used with NB. They are comparable to the other classification methods, with the exception of the slightly worse classification accuracies.

		Predicted Class						
		Nom	EVC1 (SF)	EVC2 (SF)	IVC1 (SF)	IVC2 (SF)	DA %	CA %
Actual Class	Nominal	2167	1	0	32	0	98.46	98.46
	EVC1 (SF)	0	2200	0	0	0	99.95	99.95
	EVC2 (SF)	0	0	2152	0	48	99.95	97.77
	IVC1 (SF)	0	0	0	2200	0	99.95	99.95
	IVC2 (SF)	0	0	0	0	2200	99.95	99.95

Table 6-76: LDA confusion matrix

	Training time	**Testing time**
Test #1	0.0624	0.0156
Test #2	0.0156	0.0156
Test #3	0.0312	0.0156
Test #4	0.0156	0.0312
Test #5	0.0156	0.0156
Average	**0.028**	**0.019**

Table 6-77: LDA execution times

6.5 Summary

In this chapter, experiments were performed on the engine test bed. Results were given and compared for a variety of different cases and parameters. Tests were done for different window sizes, threshold values, training cycles, engine speeds, and sampling rates. Additionally, smaller and larger faults were tested. Testing and comparing these results helped demonstrate the generality and robustness of the fault detection method, and also gave insight to manual tuning of the system parameters. From manual tuning it was found that a window size of 41 steps and a s_f value of 1 gave the best classification results. An engine speed of 2000 RPM and sampling rate of 48 kHz provided best results, however other rates also gave acceptable results. Lastly, the NB method developed for this thesis was compared to other common classification techniques, a summary of this comparison can be found in Table 6-78.

Classification Method	Worst Case DA	Worst Case CA	Feature Ext. Time	Training Time	Testing Time	Total FDD Time
NB	99.95%	99.95%	4.5	0.019	0.031	4.5
ANN	0%	0%	4.5	7.13	0.076	11.7
DT	99.95%	98.96%	4.5	0.016	0.016	4.5
k-NN	99.95%	99.95%	4.5	0	0.94	5.4
LDA	98.46%	97.77%	4.5	0.0281	0.0187	4.5

Table 6-78: Classification Method Comparison Summary

From Table 6-78 it can be seen that the NB method seems to give very good results as originally suggested from our initial literature review. The results also show that other classification methods also work well for our application, which is a testament to our FDD signal processing and feature extraction methods. It seems that each classification method tested gives quite reliable and accurate results, with the exception of the ANN method which works well in some cases and very poorly in other cases; seemingly dependant upon the training size being used as well as the initial weight estimates. This is a positive attribute of our FDD method as it is robust for many classification methods, meaning a user can utilize our FDD method and implement the classification method of their choice.

CHAPTER 7: Conclusions

7.1 Summary

In this thesis a FDD system for an ICE application was developed. A novel feature extraction method was developed with a focus on the valve impact amplitudes. An instrumented diesel engine test bed was set up. Faults were seeded in the form of deformed valve springs as well as abnormal valve clearances. The performance of the FDD system was tested using the diesel engine test bed. A comparison of numerous signal processing techniques and classification methods was performed. The experimental results showed that the success rate of the FDD system was quite promising for a number of classification methods. The FDD system was shown to reliably detect and classify faults in their early stages.

7.2 Achievements

The main achievements of this thesis are summarized as follows:

1) This research studied and demonstrated the feasibility of early detection of ICE valve train faults. It was found that the valve train faults under study can be detected and classified with high accuracy using valve-impact vibration analysis.
2) A number of signal processing techniques were tested and compared. Engine cycle averaging and band-pass filtering methods were tuned and utilized for improving the SNR. A novel feature extraction method was developed that included a local RMS sliding window method and an adaptive threshold.

3) Multiple classification methods were applied and tested, it was found that the NB method seems to give superior results, however a number of classification methods gave acceptable results. This is a testament to the developed feature extraction method, as it provides good results for multiple classification methods.
4) The proposed FDD method proved successful using only impact amplitudes as detection features. In addition to single valve faults, multiple simultaneous faults on both separate and identical valves was tested and successfully diagnosed; something that has not been accomplished in previous research literature.
5) A number of testing scenarios were varied and numerous parameter values were altered. The proposed FDD method gave reliable and accurate classification results for many different cases, demonstrating the generality and robustness of the proposed method.

7.3 Recommendations for Future Work

1) The fault severity is the main distinguishing aspect of a FDD method. As the fault severity is decreased it naturally becomes more difficult to detect. Although a minor investigation of fault severities was done in this research, further testing on the severity of valve train faults could be analyzed to see the full effect of fault severities on classification accuracy.
2) Testing the proposed FDD method on a larger engine with more valves and cylinders. Our research was restricted to a two-cylinder engine using a single accelerometer for both cylinders. It would be interesting to explore larger engines such as four-cylinder, six-cylinder, eight-cylinder, etc. and determine how well

the FDD method works for these cases. Also to test a single accelerometer in comparison with multiple accelerometers for the large engine cases.

3) This FDD method uses only sn features to classify each valve. Additional features can be used if needed. It should be possible to detect more complicated cases using the method outlined in this section with additional features.

4) Further development of this FDD method may potentially be used as a real-time online FDD system on a vehicle. This would require alterations including hardware modifications, integration with current vehicle software systems, and addition of dynamic FDD methods to account for varying engine speeds and loads.

References

Altman, E. I., Marco, G., & Varetto, F. (1994, January). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks. *Journal of Banking and Finance*(18), 505-529.

Automotive Engineering. (2013, March). *Valvetrain: One complete four stroke cycle*. Retrieved from Automotive Engineering: http://www.theautomotiveengineer.com/2013/03/ valvetrain-one-complete-four-stroke.html

Avriel, M. (2003). *Nonlinear Programming Analysis and Methods.* Dover Publications.

Betta, G., Liguori, C., Paolillo, A., & Pietrosanto, A. (2001). A DSP-based FFT-Analyzer for the Fault Diagnosis of Rotating Machine Based on Vibration Analysis. *IEEE Instrumentation and Measurement*, 572-577.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. (1984). *Classification and Regression Trees.* Chapman and Hall.

Brüel & Kjær. (1982, September). Measuring Vibration. Nærum, Copenhagen, Denmark: Brüel & Kjær.

Chiang, L. H., Russell, E. L., & Braatz, R. D. (2000, November). Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Iintelligent Laboratory Systems*(50), 243-252.

Choi, K., Singh, S., Kodali, A., Pattipati, K. R., Sheppard, J. W., Namburu, S. M., . . . Qiao, L. (2009). Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems. *IEEE Transactions on Instrumentation and Measurement*, 602-611.

Cummins Generator Technologies Inc. (2012). *Products: Markon BL105*. Retrieved from Cummins Generator Technologies: https://www.cumminsgeneratortechnologies.com /en/products/markon/b105/

Desbazeille, M., Randall, R., Guillet, F., Badaoui, M. E., & Hoisnard, C. (2010). Model-based diagnosis of large diesel engines based on angular speed variations of the crankshaft. *Mechanical Systems and Signal Processing*(24), 1529-1541.

Do, V. T., & Chong, U.-P. (2011). Signal Model-Based Fault Detection and Diagnosis for Induction Motors Using Features of Vibration Signal in Two-Dimension Domian. *Journal of Mechanical Engineering*, 655-666.

Elamin, F., Fan, Y., Gu, F., & Ball, a. A. (2010, April). Diesel Engine Valve Clearance Detection Using Acoustic Emission. *Advances in Mechanical Engineering*.

Gupta, J. (2001). *Neural Networks in Business: Techniques and Applications.* Hershey: IGI Global.

He, Q. P., & Wang, J. (2007, November). Fault Detection using the k-Nearest Neighbour Rule for Semiconductor Manufacturing Processes. *IEEE Transactions on Semiconductor Manufacturing*, *4*(20), 345-354.

House, J. M., Lee, W. Y., & Shin, D. R. (1998). Classification Techniques for Fault Detection and Diagnosis of an Air-Handling Unit. *ASHRAE Transactions: Symposia*, 1087-1097.

Isermann, R. (2006). *Fault-Diagnosis Systems.* Darmstadt: Springer.

Isermann, R., & Ballé, P. (1997, March). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Eng. Practice, 5*(5), 709-719.

Jian-Da Wu, C.-H. L. (2009). An Expert System for Fault Diagnosis in Internal Combustion Engines using Wavelet Packet Transform and Neural Network. *Expert Systems with Applications*(36), 4278-4286.

Jian-Da Wu, C.-H. L. (2009). An expert system for fault diagnosis in internal combustion engines using wavelet packet transforms and neural network. *Expert Systems with Applications*(36), 4278-4286.

Kubota. (2008, March 07). Workshop Manual Diesel Engine SM-E3B Series. Japan.

Lai, T., & Snider, L. (2005, January). High-Impedance Fault Detection Using Discrete Wavelet Transform and Frequency Range and RMS Conversion. *IEEE Transactions of Power Delivery, 20*(1), 397-407.

Li, Y., Tse, P. W., Yang, X., & Yang, J. (2010, June). EMD-based fault diagnosis for abnormal clearance between contacting components in a diesel engine. *Mechanical Systems and Signal Processing*(24), 193-210.

Lin, J., & Qu, L. (2000, January). Feature Extraction Based on Morlet Wavelet and its Application for Mechanical Fault Diagnosis. *Journal of Sound and Vibration*, 135-148.

Liu, J., & Chen, D.-S. (2009, February). Fault Detection and Identification Using Modified Bayesian Classification on PCA Subspace. *American Chemical Society, 48*(6), 3059-3077.

Liu, S., Gu, F., & Ball, A. (2005, November). Detection of engine valve faults by vibration measured on the cylinder head. *Institution of Mechanical Engineers, 220,* 379-386.

LS1Tech. (2013). Retrieved from LS1Tech: http://ls1tech.com/

Montgomery, D. C., & Runger, G. C. (2007). *Applied Statistics and Probability for Engineers.* Danvers: John Wiley & Sons, Inc.

Nabende, P., & Wanyama, T. (2006, August). An Expert System for Diagnosing Heavy-Duty Diesel Engine faults. 1-6.

National Instruments. (2006, August 06). User Guide SCC-68.

National Instruments. (2010, August 10). NI 6351/6353 Specifications.

National Instruments Corporation. (2013). *LabVIEW System Design Software*. Retrieved from National Instruments: http://www.ni.com/labview/

Nise, N. S. (2008). *Control Systems Engineering.* Danvers: Wiley.

Obaid, R. R., & Habetler, T. G. (2003). Current-Based Algorithm for Mechanical Fault Detection in Induction Motors with Arbitrary Load Conditions. *IEEE*, 1347-1351.

Optrand. (2013). *PSIglow: Pressure Sensing Glow Plug Adapter*. Retrieved from Optrand: optrand.com/psiglownfo.htm

PCB Piezotronics. (2011, March 25). Accelerometer, ICP. Depew, New York, United States of America: PCB.

Saravanan, N., & Ramachandran, K. (2009). Fault diagnosis of spur bevel gear box using discrete wavelet features and Decision Tree classification. *Expert Systems with Applications*(36), 9564-9573.

Shen, L., Tay, F. E., Qu, L., & Shen, Y. (2000, January). Fault diagnosis using Rough Sets Theory. *Computers in Industry*(43), 61-72.

Sotcher Measurement Inc. (2013). *Generator Test Set*. Retrieved from Sotcher: http://sotcher.com/Generator_Test_Set/

Stranneby, D., & Walker, W. (2004). *Digital Signal Processing and Applications.* Burlington: Elsevier.

Sun, W., Chen, J., & Li, J. (2006, June). Decision tree and PCA-based fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*(21), 1300-1317.

Zheng, H., Li, Z., & Chen, X. (2002, February). Gear Fault Diagnosis Based on Continuous Wavelet Transform. *Mechanical Systems and Signal Processing*(16), 447-457.