Feng, Guojin

Optimisation of vibration monitoring nodes in wireless sensor networks

OPTIMISATION OF VIBRATION MONITORING NODES IN WIRELESS SENSOR NETWORKS

Guojin Feng

Supervised by

Professor Andrew D. Ball and Dr Fengshou Gu

October 2016

School of Computing and Engineering

University of Huddersfield

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AT	Attention
AC	Alternative current
ADC	Analogue to digital converter
ADI	Analog devices Inc.
AM	Amplitude-modulating
API	Application programming interface
ARMA	Autoregressive moving average
ASCII	American standard code for information interchange
BM	Breakdown maintenance
BPFI	Ball pass frequency, inner race
BPFO	Ball pass frequency, outer race
BPSRC	Band-pass squared rectifier
BSF	Ball (roller) spin frequency
CBM	Condition-based maintenance
CCS	Code composer studio
CFFT	Complex-valued fast Fourier transform
CMSIS	Cortex microcontroller software interface standard
CSMA/CA	Carrier sense multiple access with collision avoidance
CTS	Clear to send
CWT	Continuous wavelet
DC	Direct current
DFT	Discrete Fourier transform
DPM	Dynamic power management
DPM-SSM	Dynamic power management with scheduled switching mode
DSP	Digital signal processing
DWT	Discrete wavelet
ECG	Electrocardiogram
EDR	Enhanced data rate
EMD	Empirical mode decomposition
FFT	Fast Fourier transform

FIR	Finite impulse response
FPU	Floating point unit
FTF	Fundamental train frequency
GB	Gigabyte
GPIO	General purpose input and output
GST	Global sensor technology
HART	Highway addressable remote transducer protocol
HHT	Hilbert-Huang transform
HSA	Hilbert spectral analysis
HT	Hilbert transform
Hz	Hertz
I2C	Inter-integrated circuit
ICDI	In-circuit debug interface
IEC	International electrotechnical commission
IEEE	Institute of electrical and electronics engineers
IEPE	Integrated electronics piezo-electric
IFFT	Inverse FFT
IIR	Infinite impulse response
IMFs	Intrinsic mode functions
IOT	Internet of Things
IP	Internet protocol
IPv6	Internet protocol version 6
ISM	Industrial, scientific and medical
ISR	Interrupt service routine
kHz	kilo Hertz
LR-WPAN	Low-rate wireless personal area networks
MAC	Medium access control
MB	Megabytes
MCU	Microcontroller
MEMS	Micro-electro-mechanical systems
MOSFET	Metal–oxide–semiconductor field-effect transistor
MTBF	Mean time between failures
OAP	Over the air programming
PCB	Printed circuit board
PE	Piezo-electric

PM	Preventative maintenance
RAM	Random access memory
RFFT	Real-valued fast Fourier transform
RMS	Root mean square
RTOS	Real-time operating system
RTS	Request to send
RUL	Remaining useful life
SD	Secure Digital
SK	Spectral kurtosis
SNR	Signal to noise ratio
SOC	System on chip
SPI	Serial peripheral interface
SPS	Sample per second
SRAM	Static random access memory
STFT	Short time Fourier transform
SVD	Singular value decomposition
SWI	Software interrupt
TDMA	Time division multiple access
TinyOS	Tiny operating system
TSA	Time synchronous average
USB	Universal serial bus
WLAN	Wireless local area network
WMAN	Wireless metropolitan area network
WPAN	Wireless personal area network
WPT	Wavelet packet transform
WSN	Wireless sensor network
WT	Wavelet transform
WVD	Wigner-Ville distribution
WWAN	Wireless wide area network
µDMA	Micro digital memory access

ABSTRACT

This PhD research focuses on developing a wireless vibration condition monitoring (CM) node which allows an optimal implementation of advanced signal processing algorithms. Obviously, such a node should meet additional yet practical requirements including high robustness and low investments in achieving predictive maintenance.

There are a number of wireless protocols which can be utilised to establish a wireless sensor network (WSN). Protocols like WiFi HaLow, Bluetooth low energy (BLE), ZigBee and Thread are more suitable for long-term non-critical CM battery powered nodes as they provide inherent merits like low cost, self-organising network, and low power consumption. WirelessHART and ISA100.11a provide more reliable and robust performance but their solutions are usually more expensive, thus they are more suitable for strict industrial control applications.

Distributed computation can utilise the limited bandwidth of wireless network and battery life of sensor nodes more wisely. Hence it is becoming increasingly popular in wireless CM with the fast development of electronics and wireless technologies in recent years. Therefore, distributed computation is the primary focus of this research in order to develop an advanced sensor node for realising wireless networks which allow high-performance CM at minimal network traffic and economic cost.

On this basis, a ZigBee-based vibration monitoring node is designed for the evaluation of embedding signal processing algorithms. A state-of-the-art Cortex-M4F processor is employed as the core processor on the wireless sensor node, which has been optimised for implementing complex signal processing algorithms at low power consumption. Meanwhile, an envelope analysis is focused on as the main intelligent technique embedded on the node due to the envelope analysis being the most effective and general method to characterise impulsive and modulating signatures. Such signatures can commonly be found on faulty signals generated by key machinery components, such as bearings, gears, turbines, and valves.

Through a preliminary optimisation in implementing envelope analysis based on fast Fourier transform (FFT), an envelope spectrum of 2048 points is successfully achieved on a processor with a memory usage of 32 kB. Experimental results show that the simulated bearing faults can be clearly identified from the calculated envelope spectrum. Meanwhile, the data throughput requirement is reduced by more than 95% in comparison with the raw data transmission. To optimise the performance of the vibration monitoring node, three main techniques have been developed and validated:

1) A new data processing scheme is developed by combining three subsequent processing techniques: down-sampling, data frame overlapping and cascading. On this basis, a frequency resolution of 0.61 Hz in the envelope spectrum is achieved on the same processor.

2) The optimal band-pass filter for envelope analysis is selected by a scheme, in which the complicated fast kurtogram is implemented on the host computer for selecting optimal band-pass filter and real-time envelope analysis on the wireless sensor for extracting bearing fault features. Moreover, a frequency band of 16 kHz is analysed, which allows features to be extracted in a wide frequency band, covering a wide category of industrial applications.

3) Two new analysis methods: short-time RMS and spectral correlation algorithms are proposed for bearing fault diagnosis. They can significantly reduce the CPU usage, being over two times less and consequently much lower power consumption.

Keywords: Vibration, Wireless sensor network (WSN), Envelope analysis, Distributed computation, Fast kurtogram, Condition monitoring, Fault diagnosis, Short-time RMS, Spectral correlation

DECLARATION

This dissertation is submitted for the degree of Doctor of Philosophy at the University of Huddersfield. I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Huddersfield.

This work is original except where acknowledgement and references are made to the previous work. Neither this nor any substantially similar dissertation has been or is being submitted for a degree, diploma or other qualification at any other university.

ACKNOWLEDGEMENT

This work has been carried out in the Centre for Efficiency and Performance Engineering (CEPE), School of Computing and Engineering at University of Huddersfield, UK. It was financially funded by the Fee-waiver Scholarship scheme from the University of Huddersfield. I would like to give my grateful acknowledgements to the University of Huddersfield and CEPE group for their support during the research.

I would like to thank my main supervisor professor Andrew Ball for his guidance and encouragement during my research. He provides me with an excellent research environment and a lot of opportunities.

I am especially grateful to my co-supervisor Dr Fengshou Gu, principal research fellow, for his sincere and warm-hearted help. His abundant experience and great vision in the condition monitoring fields have greatly broadened my mind and provided me with a lot of inspirations.

My thanks are also to the members in CEPE and my friends in Huddersfield especially Dr Dong Zhen and Xiang'e Tian for their discussion and advice during my PhD study and kind-hearted help in life.

I would also like to express my grateful to my girlfriend Xiaoxia Liang, who has provided me a lot of understanding, encouragement and support at the final stage of my PhD study.

Finally, I would like to thank my parents, my sister and other relatives for their long-term understanding and supporting of my study.

LIST OF PUBLICATIONS

[1] **G. Feng**, N. Hu, Z. Mones, F. Gu, and A. D. Ball, ‘An investigation of the orthogonal outputs from an on-rotor MEMS accelerometer for reciprocating compressor condition monitoring’, Mech. Syst. Signal Process., vol. 76–77, pp. 228–241, Aug. 2016.

[2] **G. Feng**, D. Zhen, X. Tian, F. Gu, and A. D. Ball, ‘A Novel Method to Improve the Resolution of Envelope Spectrum for Bearing Fault Diagnosis Based on a Wireless Sensor Node’, in Vibration Engineering and Technology of Machinery, vol. 23, J. K. Sinha, Ed. Cham: Springer International Publishing, 2015, pp. 765–775.

[3] D.-X. Yang, F.-S. Gu, **G.-J. Feng**, Y.-M. Yang, and A. Ball, ‘Achieving high bit rate logical stochastic resonance in a bistable system by adjusting parameters’, Chin. Phys. B, vol. 24, no. 11, p. 110502, Nov. 2015.

[4] **G. Feng**, J.X. Gu, D. Zhen, A. Mustafa, F. Gu and A. D. Ball (2014). The Implementation of Envelope Analysis on a Wireless Condition Monitoring System for Bearing Fault Diagnosis, in International Journal of Automation and Computing (IJAC).

[5] **G. Feng**, X. Tian, J. X. Gu, D. Yang, F. Gu and A. D. Ball (2014). An adaptive envelope analysis in a wireless sensor network for bearing fault diagnosis using fast kurtogram algorithm, in 27th International Congress of Condition Monitoring and Diagnostic Engineering Management (COMADEM), Brisbane convention and Exhibition Centre, Australia, 16-18 Sep. 2014.

[6] X. Tian, **G. Feng**, A. Albraika, F. Gu and A. D. Ball (2014). The investigation of motor current signals from a centrifugal pump for fault diagnosis, in 27th International Congress of Condition Monitoring and Diagnostic Engineering, Brisbane convention and Exhibition Centre, Australia, 16-18 Sep. 2014.

[7] **G. Feng**, A. Mustafa, J.X. Gu, D. Zhen, F. Gu and A. D. Ball (2013). The Real-time Implementation of Envelope Analysis for Bearing Fault Diagnosis based on

Wireless Sensor Network, in Proceedings of the 19th International Conference on Automation & Computing, Brunel University, London, UK, 13-14 September 2013.

[8] **G. Feng**, D. Zhen, M. Aliwan, F. Gu, and A. Ball (2013). Investigation of Wireless Protocols for Remote Condition Monitoring, in Proceedings of Computing and Engineering Annual Researchers' Conference 2013, G. Lucas, Ed. Huddersfield: University of Huddersfield, 2013, pp. 19–24.

[9] Z. Mones, **G. Feng**, U.E Ogbulaor, F. Gu, A.D. Ball (2016). Performance Evaluation of Wireless MEMS Accelerometer for Reciprocating Compressor Condition Monitoring, in The International Conference on Power Transmissions (ICPT) Chongqing University, China, 2016

[10] Usama Haba, **Guojin Feng**, Abdulkarim Shaeboub, Xinyu Peng, Fengshou Gu and Andrew D. Ball (2016). Detection and Diagnosis of Compound Faults in a Reciprocating Compressor based on Motor Current Signatures, in 29th International Congress of Condition Monitoring and Diagnostic Engineering Management (COMADEM), Xi'an, China on 20th-22th Aug. 2016

[11] Badradin Elforjani, **Guojin Feng**, Fengshou Gu and Andrew Ball (2016). Thermal Harvesting based Wireless Sensor Nodes for the Monitoring Temperature of Gearbox, in 29th International Congress on Condition Monitoring and Diagnostic Engineering Management (COMADEM), Xi'an, China on 20th-22th Aug. 2016

CHAPTER 1

INTRODUCTION

Maintenance has become increasingly important to the industrial automated manufacturing process. Meanwhile, the investment on maintenance has grown quickly with the increasing complexity of machines. This research intends to conduct condition monitoring over the low power consumption wireless sensor network so as to effectively reduce the maintenance cost.

This chapter firstly explains the machinery failure and maintenance strategies. Then, the general procedure of the condition-based maintenance process is explained. The benefits and challenges of employing wireless techniques in the condition monitoring fields are discussed. Due to the limited bandwidth in wireless sensor network, distributed computation shows more superior performance in wireless condition monitoring than centralised computation. Based on these, the research aims and objectives are presented. Finally, the outline of this thesis is briefly explained.

1.1 Importance of maintenance

In today's industrial factories, the manufacturing process has become highly automated which has greatly increased productivity and quality whilst significantly reducing the level of human resources required. In the meantime, machines are getting increasingly complicated and a modern machine may include hundreds of components. The unexpected failure of a single component not only damages itself but may also affect the related equipment. Even worse, the plant operation may be interrupted and cause significant economic loss. In some critical industrial applications, system failures can cause extremely severe consequences, including explosions, personal injury and toxic releases, etc.

In practice, a machine can be classified as out-of-date and will have to be replaced, due to economic reasons, where the reality maybe it has the potential for future use. Also, a machine may be replaced to minimise its failure rate for safety considerations [1]. It is estimated that the cost for replacing equipment still in good condition is more than $1 trillion each year. The reason for this huge waste is due to the lack of a reliable and cost-effective method for predicting the equipment's remaining useful life [2]. Therefore, researchers and manufacturers are trying to find maintenance measures to ensure plant performance, minimise operational costs and extend the life of industrial machines while ensuring this does not compromise safety.

1.2 Machinery failures and maintenance strategies

Machinery failure is defined as 'the inability of a machine to perform its required functions' [1]. Numerous reasons can cause the failure, such as deficiencies in the original design, improper manufacture, inappropriate maintenance, and excessive operational demands [1]. The failure can appear very sudden, be catastrophic and unpredictable whilst on most occasions the machinery failure is caused by a gradual wear on machinery, which is incipient and grows partially and gradually as the machine is running. Such failure evolving can take days, weeks or months and thus it is possible to provide warning of impending failure at an early stage and arrange maintenance operations in time through monitoring the health condition of the machine [3].

1.2.1 Failure bathtub curve

Based on statistical analysis, the frequency of machinery failures can be typically described as “bathtub curve”, as shown in Figure 1.1 [1]. This curve is appropriate for an individual machine or a large number of machines of the same type. It can be seen that a machine experiences three stages: wear in, normal wear, and wear out.

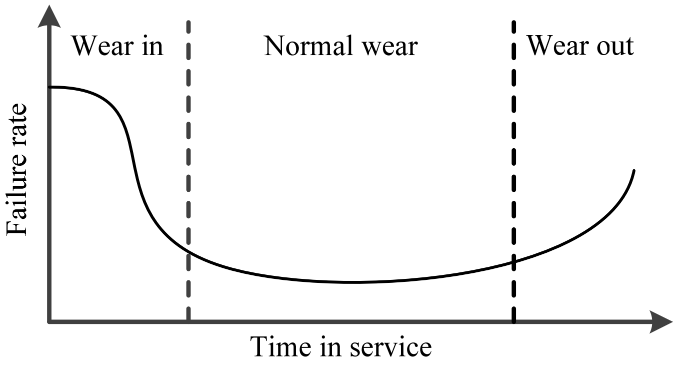


Figure 1.1 Typical bathtub curve [1]

Wear in. This stage is the beginning of a machine’s useful life. At this stage, the failure rate tends to be relatively high due to design errors, manufacturing defects, assembly mistakes, installation problems and commissioning errors [1]. With these failures being discovered and corrected, the failure frequency decreases quickly.

Normal wear. After wear in stage, the machine passes into a relatively stable period, named normal wear period. During this period, there is a relatively low failure rate on condition the machine is running under its design specifications. This period occupies the most of a machine’s life.

Wear out. As a machine gradually approaches the end of its designed life, it enters the wear out stage, during which the failure rate tends to increase. At this period, the failures are usually caused by metal fatigue, wear mechanisms between moving parts, corrosion, and obsolescence [1]. Note that the slope of the wear out part is different from one machine to another and the failure increasing rate is largely determined by the design of the machine and its operational history.

1.2.2 Maintenance strategies

To keep the machine running in a healthy state, different maintenance strategies have been developed. Among them, the most three popular ones are breakdown maintenance

(BM), preventative maintenance (PM), and condition-based maintenance (CBM). The definition, advantages, and disadvantages of these three strategies are described below.

A. Breakdown maintenance (BM)

This strategy is also known as run-to-failure maintenance. By its meaning, it means maintenance actions are carried out only when the machine is completely out of service. In this situation, the machinery failure should not result in critical damage or the cost of a new replacement machine being cheaper than the maintenance costs. Thus no attention is paid to the machines efficiency and maintenance. Another reason is that little knowledge is known about the types of faults and the time of fault occurrence in advance.

The advantage of BM is that maintenance investment is not necessary and therefore costs are especially low. Meanwhile disadvantages are also obvious, for example, an emergency outage cannot be predicted, which may lead to plant related damages, or even cause fatal accidents, production loss, and loss of control over management. Therefore, this method is seldom admirable nowadays.

B. Preventive maintenance (PM)

This strategy is also called scheduled maintenance or time-based maintenance, which means that maintenance actions are prepared and carried out well before the occurrence of the fault, along with the increased knowledge and experience of mechanical principle and likely faults. It is carried out according to a predetermined plan which means that the maintenance is performed at predetermined intervals or according to some prescribed criteria. Because of the limited knowledge of the development of failure rule and lack of scientific methods in fault diagnosis, the determination of the intervals or criteria is usually based on design specifications, previous experience and statistical data. Thereby, it is difficult to prevent accidents caused by stochastic factors and maintenance for the frequent repairs is inevitable.

The benefit of preventive maintenance is that, firstly, maintenance could be planned so that it could be carried out at a convenient time; secondly, fewer accidents happen, especially disastrous accidents; thirdly, it is a more efficient way of controlling spare parts and saving money. However, it is not wise to perform maintenance at a fixed time interval. On the one hand, if the interval is shorter than its functional lifespan, the implementation of maintenance may cause a significant waste of money and labour. On the other hand, longer intervals may be too late to prevent failures or to retain the required function. Additionally, unexpected breakdowns maybe still happen and fatigue life assessments and optimisation are not monitored effectively.

C. Condition-based maintenance (CBM)

For this strategy, maintenance is conducted based on the knowledge of the conditions of the machine instead of the predetermined plan or schedule. The optimal maintenance time and contents are determined according to the on-line detection data and fault condition of a machine predicted by the diagnostic unit. If well conducted, a CBM program can successfully prevent unexpected catastrophic faults, improve the reliability and availability of a machine, decrease downtime, and increase operating efficiency. Therefore, this maintenance strategy is able to exploit the maximum operating time of a machine and minimise maintenance cost. The key of CBM is the knowledge of the condition of a machine which can be obtained by employing various monitoring techniques. Note that CBM requires additional investment in that instrumentations need to be installed to monitor the machine condition and experts may also be required for analysing the acquired data and suggesting maintenance operations.

A comparison of the three types of maintenance strategies is summarised in Table 1.1. Apparently, the CBM method is more effective than other two methods especially for the maintenance of complicated and critical machines. Thus, it is getting widely applied in the industrial world. Besides these three main strategies, there are also several other maintenance strategies, such as reliability/risk centred maintenance, automatic maintenance, and controlled maintenance. The main objectives of all the maintenance actions are to produce required outputs, maximise designed life span, abide by safety standards and minimise maintenance costs [4].

Table 1.1 Comparison of maintenance strategies

Strategy	Advantage	Disadvantage
BM	Low cost	May lead to plant related damages
PM	Reduce the accidents rates, especially catastrophic accidents	➢ Difficult to determine the perfect maintenance intervals ➢ High maintenance cost
CBM	➢ Able to exploit the maximum operating time of a machine ➢ Maintenance cost can be reduced greatly by reducing the number of unnecessary scheduled preventive maintenance operations	➢ Extra investment on instrumentations ➢ Experts needed for data analysis

1.3 Introduction of condition-based maintenance

In a CBM process, maintenance decisions are recommended based on the information collected through condition monitoring (CM). Through automated CM, human inspection requirements and unnecessary maintenance can be reduced whilst the safety and reliability of the machinery can be significantly improved by finding faults at an early stage. Ideally, CBM allows the personnel to carry out the correct maintenance, minimising spare parts cost, system downtime and time spent on maintenance.

1.3.1 Process of condition-based maintenance

As shown in Figure 1.2, a CBM program typically includes three key steps: data acquisition, data processing, and maintenance decision-making [1]. In general, the data acquisition step collects data related to the system health. The data processing step analyses the obtained data to gain a better understanding of the system health and the maintenance decision-making step provides efficient maintenance policies.

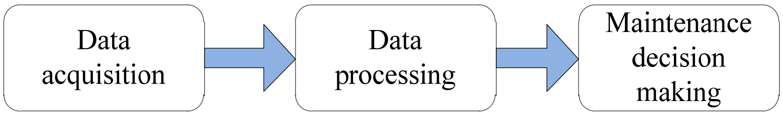


Figure 1.2 Steps in a CBM program [1]

Step 1. Data acquisition

In this step, the useful data or information from the target machines are collected. This collected data can be divided into two main types: event data and condition monitoring data. The event data is the machine operation and maintenance related information, including what happened (e.g., installation, breakdown and overhaul) and what was conducted on the target machine (e.g., minor repair, preventive maintenance and oil change). This information are commonly recorded by the maintenance personnel manually.

Condition monitoring data is the measurements related to the health condition/state of the target machine. Such data is versatile and can be of various types, such as temperature, chemical gases, vibration, motor speed, acoustic, acoustic emission and electrical methods [3], [5]. Among these monitoring techniques, vibration monitoring has been proven to be a sensitive and effective method of detecting faults in rotating machinery [3].

Vibration monitoring refers to ‘the use of sensing and analysis of system characteristics – in the time, frequency or modal domains – for the purpose of detecting changes, which may indicate damage or degradation’ [6]. This kind of non-destructive technique, its first usage for detection can be traced back to the 1970s and early 1980s at which time it was utilised in the aerospace and offshore oil industries [7]. The reasons for vibration based CM being widely employed can be summarised as below:

- **Wide existence.** Almost every machine or process in the manufacturing process produces vibration in one form or another while in operation.
- **High sensitivity to the fault**. The vibration sensors are usually attached to the monitored machine directly, which indicates they are close to the physical location of the fault and the transmission loss is small.
- **Solid theory study.** Vibration mechanisms of most machinery and structures have been understood, which makes it possible to diagnose the characteristics of vibration responses resulting from these faults.
- **Reliable equipment.** Vibration instrumentations with high performance, such as wideband transducers and analysis equipment, have become reliable over years of utilisation.
- **Ongoing development.** The fast development of vibration signal processing methods and computing facilities contribute to its wide applications.

Step 2. Data processing

This step contains two sub-steps: data cleaning and data analysis. Data cleaning is an important step, through which the errors contained in the data especially those entered manually can be removed, thereafter increasing the chance of clean (error-free) data being employed for further analysis and modelling. Without such a step, it is very likely to get into the so-called “garbage in garbage out” situation.

Data analysis is the process of using various models, algorithms and tools to analyse data so as to get better understanding and interpretation of data. The models, algorithms and tools used for data analysis depend mainly on the types of data collected, which are versatile and could fall into three categories: value type (e.g. temperature, pressure and humidity signal), waveform type (e.g. vibration and acoustic signals) and multi-dimension type (e.g. visual images and infrared thermograph).

Usually, the data of value type changes slowly, thus they are usually called static data. The required sampling rate of such data is usually low. For instance, a sampling rate of one sample per second (1 SPS) will be sufficient for most temperature signals. The waveform data changes much faster and is commonly named dynamic data. In order to fully capture their information, a much higher sampling rate is required. For example, for a signal that vibrates at 1 kHz (kilo Hertz), in order to reconstruct the signal after digitisation, the sampling rate should be higher than 2 kHz according to the Nyquist sampling theorem [8], which means 2000 data points per second. Apparently, data analysis techniques for these two types are different.

Data processing for waveform data is also usually named signal processing. There are a number of signal processing techniques developed to analyse and interpret waveform so as to extract useful information for further diagnosis and prognosis. The procedure of extracting useful information from raw signals is the so-called feature extraction. The research focus of this thesis is vibration signal, which is a type of waveform data. The relevant signal processing techniques will be reviewed in Chapter 4.

Step 3. Maintenance decision-making

After data processing, it comes to maintenance decision-making, the last step of a CBM program. The maintenance actions made by maintenance personnel crucially relies on sufficient and efficient decision support. The techniques for maintenance decision support can be usually categorised into two main types: diagnostics and prognostics [9].

Diagnostics. Machine fault diagnostics can be considered as “a procedure of mapping the information obtained in the measurement space and/or features in the feature space to machine faults in the fault space” [9]. This mapping process is also called pattern recognition. Previously, pattern recognition has been performed manually by analysing the graphical tools like power spectrum graph, cepstrum graph and wavelet scalogram. In recently years, techniques like artificial neural networks and machine learning are gaining popularity to achieve automatic pattern recognition based on the information and features extracted from the signals [1], [9].

Prognostics. Machine fault prognostics mainly have two main types. The most commonly used prognostics type is remaining useful life (RUL), which intends to predict the life time left before a failure occurs given the current machine condition and past operation profile [9]. Another type is to predict the chance that a machine operates without a fault or a failure up to some future time based on the current machine condition and its past operation profile. The later prognostic type is suitable for situations when a fault or a failure is catastrophic (e.g., nuclear power plant).

1.3.2 Challenges in condition monitoring

The importance of condition monitoring and fault diagnosis has gained wide acceptance in both academic and industrial areas in the past decades. Every year, there are several international conferences on such topics and thousands of paper publications, studying various topics including monitoring techniques [3], mechanical fault modelling [10], [11], fault diagnosis and prognosis [12], [13], etc. However, such effective and advanced methods are rarely well utilised by industries. Rather, the industrial applications are more likely to employ traditional techniques developed more than 20 or 30 years ago. As a typical case, using vibration technique for condition monitoring and fault diagnosis of rolling element bearing has been well studied in the academic arena [14], however, quite a few industrial applications still tend to rely on the simple vibration root mean square (RMS) as a warning indicator instead of advanced techniques, such as envelope analysis [15], cepstrum [16] and Wavelet [17]. These indicate that a gap exists between academic research and industrial applications. The main reasons for this gap can be summarised as:

- **High cost**. For conducting CBM, a significant investment is required on instrumentations, equipment installation and also software development. With machines becoming increasingly complex, their relevant maintenance cost has been growing quickly as well. It is estimated that approximately half of operating costs in most processing and manufacturing operations can be attributed to maintenance [1].

- **Difficult to interpret fault results**. Although the CBM software has tried to make the results informative, special training is required for using such software and experts may also be needed for interpreting the monitored data and fault diagnosis results.

- **Isolated academic research and industrial implementation**. The dilemma is that the data analysis and fault diagnosis usually involves complicated mathematical equations and is usually implemented and validated using scientific computing packages, such as Matlab and Mathematica. Were as the software developers are good at programming rather than mathematical equations and they tend to use the efficient programming languages, such as C/C++ and Java. Furthermore, these advanced programs are usually restricted inside small research groups, which are not easily available for implementation engineers.

This thesis has intended to solve these challenges by employing the latest technologies in electronics and wireless communication. The emerging wireless sensor network (WSN) and the Internet of Things (IOT) have the potential to significantly reduce cost in CM area and the open source based development idea can potentially help narrow down the gap between academic research and industrial implementation. For this reason, the main embedded development work in this thesis is based on open source hardware and software. The relevant hardware circuits are easy to purchase and the software implementation and validations are explained in detail for easy duplication.

1.3.3 Wireless condition monitoring

Currently, wired online CM systems have been successfully applied in many industrial areas. Wired sensors provide detailed and reliable information for machine health condition analysis and some wired sensors can even shut down a machine automatically if they detect excessive abnormity. However, so far, wired online CM systems have been mainly restricted to large and critical industrial machines due to their high cost. With the increasing complexity of industrial machines, the associated costs of installation, maintenance, troubleshooting and upgrading wiring have escalated [18].

With the recent fast development in electronics and communication technologies, wireless sensor network (WSN) is becoming increasingly popular in CM area. Wireless technique brings both benefits and challenges for CM, which are summarised in the following two subsections.

1.3.3.1 Benefits of employing wireless CM

Wireless CM systems employ electromagnetic waves as the transmission medium and hence avoid the limitations of wired networks and offer a number of attractive advantages [18]:

- **Low cost.** The installation and maintenance costs take a large part in the overall maintenance cost, especially for the installation in harsh environments. As estimated, the wiring can cost £40 to £80 per foot including labour. Specialised wiring for harsh environments can cost as much as £1500 per foot [18]. By utilising the wireless CM system, tens of thousands of feet of wiring can be eliminated from the average industrial site. This has been the driving factor for the employment of wireless systems in the CM fields. According to a market study, the lower cost has been cited as a major reason for adopting wireless technology [18].

- **Ease of maintenance**. The wires are more prone to crack or fail with age. A lot of time and labour can be spent on inspecting, troubleshooting, repairing and replacing the aged wires [18]. If the wiring faults result in production stoppages, the costs can be significant. Therefore, wireless systems can effectively avoid such costs on time, labour and economic.
- **Reduced connector failure**. A common failure mode within a network are the connectors, hence these can be one of the most common reliability risks for the wired systems [18]. Especially for sensors working in harsh environments, the cables can be easily damaged by chemicals, vibration, or moving parts. In the wireless system, such failures can be eliminated.
- **Ease of replacement and upgrading.** Without the constraint of wires, the replacement or upgrading of the wireless CM systems become much easier. It does not need to mark or distinguish different communication wires. By placing the wireless sensors in the right position and after simple configuration, the measurement systems will be functional. Furthermore, the wireless technology also enables the remote monitoring of equipment located in hard to reach or dangerous locations, such as the electrical substation and wind power stations.
- **Security**. Security is a big concern in industrial applications. The continuous development of data encryption and wireless data transmission hiding techniques promises the security of wireless transmission at a level similar or surpasses that of wired systems [18].

1.3.3.2 Challenges in wireless CM

Currently, the successful application of wireless CM can be found in various commercial areas, such as environmental monitoring [19], structure health monitoring [20], temperature monitoring in product distribution [21]. However, it faces a number of challenges for being applied in the industrial field, which has unique characteristics and requires more strict demands, as summarised below [22], [23]:

- **Processing heterogeneous sensor signals:** The industrial monitoring system usually needs to measure various types of signals, including static types like temperature, pressure and dynamic ones like vibration and acoustic. The processing methods for different signals can be different.

- **Higher sampling rate**. The sampling rate of condition monitoring signals can be much higher than that for environmental or structural applications due to the need to perform accurate and comprehensive monitoring based on dynamic signals such as mechanical vibration and electrical terminal data from electrical machines. These signals generally require a bandwidth of dozens of kHz [3].

- **Faster data transmission rates**. The higher sampling rate of dynamic signals produces large amounts of data hence the needs for fast rate transmission. The high-speed data collection and transmission can often be obstructed by the limited wireless bandwidth [23].

- **Energy efficiency**. The battery is usually used as the power source for wireless sensor nodes to achieve real wireless connection. In this case, the life of a sensor node mostly depends on the life of the power resources of the nodes. Restricted by their limited physical size, sensor nodes commonly have limited battery energy supply [22]. Thus, battery life is a big concern for the wireless systems because battery replacement is an additional maintenance activity that can offset the savings provided by wireless sensors.

- **Higher reliability in data transmission**: The industrial machinery applications are usually more intolerant to data loss problems, thus requiring a more reliable communication.

- **Cost effective**. A sensor network is typically made up of many sensor nodes and the cost of a single node determines the cost of the complete network. If the expenditure in implementing sensor network is higher than that for traditional sensors, the sensor network is not applicable. Hence, the cost of each sensor node should be kept low in order to optimise the overall cost of the sensor network.

- **Resource constraints**. For cost and convenience of installation, the sensor nodes are usually designed to be compact in physical size. Hence, battery size (energy) of a sensor node is usually limited. In addition, a sensor node typically has restricted computational capability and limited memory.

- **Higher security requirement**. Industrial applications typically require higher security communication mechanisms than those of the commercial ones due to high confidentiality and criticality of industrial applications.

Due to limitations in current wireless technology, it is not practical to completely replace the existing wired communications in the industrial applications with WSNs. A reasonable approach is to form a system that includes both wireless and wired techniques. For the noncritical tasks like condition monitoring and energy efficiency monitoring, wireless technologies can be employed to reduce the overall cost while for the critical tasks like motor controlling, the wired system can be utilised for reliability and timing considerations [24].

1.3.4 Embedded processing on distributed wireless sensor nodes

The potential benefits and challenges of wireless CM have attracted a number of interests from researchers [22], [24], [25], semiconductor companies [26] and industrial products suppliers [27]–[29]. Different energy harvesting techniques are emerging to prolong the lifespan of sensor nodes by harvesting energy from ambient resources like wasted heat [30], [31], solar [32], [33] and vibration [34], [35]. A number of wireless technologies have been developed or improved to satisfy requirements in industrial applications. For example, WirelessHART and ISA100.11a are proposed to achieve high reliability and security in wireless communication [36]. Intelligent data processing is embedded in the distributed wireless sensor node in order to reduce the transmission data size for signals with high sampling rate [25] and maintain effective communications with a minimum amount of power [18].

According to computing allocation schemes, CM systems can be categorised into two main types: centralised and distributed computation. The advantages and disadvantages of centralised and distributed computation are summarised in Table 1.2. Centralised computation based CM, by its name, performs both data analysis and fault diagnosis in a data processing centre. This architecture has by far been the most widely used scheme in wired CM systems. It allows the operator to view comprehensive monitoring information at a single location; however it can significantly consume the valuable bandwidth in the WSN for data with the high sampling rate, and also cause reliability and transmission efficiency problems.

Table 1.2 Advantages and disadvantages of centralised and distributed computation

	Advantages	**Disadvantages**
Centralised CM	➢ Gathers comprehensive monitoring information at a single location ➢ Simple function sensor nodes	➢ High data transmission load ➢ Intensive computing in the central unit for analysing big data sets ➢ Less reliability and transmission

		efficiency with adding of sensor nodes with high sampling rate data
Distributed CM	➢ Relaxed bandwidth requirement ➢ Off-load computation requirement in the central unit ➢ Able to establish large scale network ➢ Utilise big data process resources	➢ High challenges in implementing signal processing algorithms on sensor nodes with constraint resources.

Unlike centralised computation, distributed computation based CM allows data processing and even fault diagnosis on sensor nodes, which reduces unnecessary data transmission in the network and thus allows a large number of sensor nodes to coexist in one network. In addition, distributed CM system off-loads the intensive computing in the central unit, allowing complicated CM data analysis and decision making to be realised in a medium specification computer, which is much more cost effective compared with centralised CM which often needs a computation intensive computer at a high cost.

With the fast development in electronics technology, more powerful sensor nodes are being developed with higher computation capability but lower power consumption. This has reduced the challenges in embedding intelligent signal processing on sensor nodes. For this reason, distributed computation is becoming increasingly popular in wireless CM in recent years. In [37], Sreenuch et al. proposed an approach for distributed condition monitoring systems that offers a reusable software architecture for a number of CM applications. Tan et al. [38] proposed a distributed fault detection method for monitoring computer numeric control machines based on cutting force, vibration and sound information. Yin and Zhong [39] monitored rotating auxiliaries at power plants based on a distributed wireless vibration CM system, in which they employ a data-level fusion for comparing the similarity of adjacent data and a task-level fusion for providing the strategy of sending data and the way to judge nodes' survival. L. Hou et al. [25] proposed a scheme for induction motor condition monitoring and fault diagnosis based on motor stator current and the vibrational signature. In this system, feature extraction and classification by the neural network classifier are implemented on the node and decision level fusion is executed at the centre.

Another benefit of distributed computation is that it has the potential to reduce power consumption of sensor nodes. According to [22], much more energy is usually required for data communication in comparison to sensing and data processing. Hence, distributed computation is becoming the trend in wireless condition monitoring and different kinds of technologies have been explored by researchers, such as embedding dedicated signal processing algorithms [25], [40], compression [20], [41] and compressive sensing [42], [43].

L. Hou and N. W. Bergmann performed fast Fourier transform (FFT) of 512 points on the vibration signal and extracted 12 most frequently occurring frequency components in the frequency domain as the fault feature for data transmission in a wireless system [25]. With the help of further data fusion methods, the payload transmission data was reduced from 1024 bytes to 8 bytes, achieving a data reduction of more than 99%. In [20], Lynch employed a WSN to monitor the bridge health condition. By embedding computational efficiency FFT and transmitting modal frequencies instead of the time-history record (raw data), a major energy saving of over 98% was achieved. Also, the AR coefficients were calculated, which required longer execution times and an energy saving of about 50% was experienced. In [41], Nachman et al evaluated data processing on a wireless sensor node, named IMote2. By using downsampling and FFT on the sensor node, the transmitted data size and energy consumption is reduced as much as 250x and 48x, respectively, in comparison to not performing any data processing.

In terms of compression, it can be lossless or loss types. In [20], Lynch employs lossless Huffman coding method for processing structural response data and a compression ratio ranging from 0.6 to 0.8 are attained on the wireless sensing unit. For audio signal, typical compression ratios can be 4:1 to 3:1 for lossless type and 10:1 to 20:1 for loss type [41]. Note that the loss type compression usually is more complicated than the lossless one.

Recently, one emerging signal processing method, named compressive sensing, is becoming popular in the wireless sensor network [44], [45]. It is a technique originated from image processing for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. Its idea is to sense information instead of the signal by using a sampling rate much lower than Nyquist sampling frequency [46]. Some research works have utilised this technique for condition monitoring. Bouzid [47], [48] adopted low sampling rate to collect acoustic emission signals for the institute health monitoring of wind turbine blade using the acoustic wireless sensor networks. Bao et al. [42] used compressive sensing for structural vibration data loss recovery and

Zhang et al. [43] explored the usage of compressive sensing for bearing signal processing.

Besides the above techniques, there are also some other effective processing methods, such as frequency band energy extraction [40]. In Chapter 4, more signal processing algorithms will be explored and their suitability for being embedded on the wireless sensor nodes will be discussed.

Considering the benefits of distributed computation, it is more suitable for vibration based wireless condition monitoring. Thus, embedding signal processing algorithms on distributed sensor nodes are considered as the study focus in this thesis.

1.4 Research aims and objectives

This PhD research aims to develop a wireless vibration condition monitoring node which allows an optimal implementation of advanced signal processing algorithms. Distributed computation can utilize the limited bandwidth of the wireless network and battery life of sensor nodes more wisely; making it more suitable for vibration based wireless condition monitoring. Therefore, embedding signal processing algorithms on distributed sensor nodes is considered as the study focus in this thesis.

As a typical vibration response, an impulsive and modulating signal can be found in many key machinery components, such as bearings [15], [49], gears [50], turbines [51] and valves [52]. It is characterised by the presence of a periodic repetition of sharp peaks modulated by high-frequency resonance components [53]. The thesis focuses on this type of vibration signal and develops embedded signal processing techniques that are widely applicable.

To achieve the above research aim, the key objectives are prioritised as follows:

1) Review popular wireless techniques and compare their advantages and disadvantages for being applied for industrial CM applications. Investigate research works performed and current industrial solutions for wireless condition monitoring. On this basis, select one wireless protocol to prototype a wireless CM system for evaluation.

2) Prototype a wireless CM system with a focus on the design of the vibration based wireless sensor node for implementing and evaluating signal processing algorithms

that are commonly used in condition monitoring field. Consideration of its power consumption should be made in order to maintain a service span as long as possible when it is powered by batteries.

3) Study the fault generation mechanism and fault features of rolling element bearings and then investigate popular signal processing algorithms for their fault detection and diagnosis. Discuss the requirements for an algorithm to be implemented on processors with restricted computational capability and limited memory size. On this basis, select one signal processing algorithm for being embedded on the sensor node for bearing fault feature extraction.
4) Implement the chosen envelope analysis algorithm on the proposed wireless sensor nodes for extracting the fault features of rolling bearings. Optimise and validate each implementation step including data acquisition, data processing, and data transmission. Utilise the implemented algorithm for bearing fault feature extraction and evaluate its performance in terms of computation speed, memory consumption and data reduction in comparison with the raw data transmission.
5) Discuss the necessity and possibility for improving the spectrum resolution in the envelope spectrum. Note that this improvement should be accomplished on the same processor with the idea of frame data processing and down-sampling. Explain the implementation procedure and validate the correctness of the implementation. Compare the performance of the improved method on bearing fault feature extraction with that in the initial implementation.
6) Investigate the methods for selecting an optimal band-pass filter for envelope analysis. On this basis, utilise one algorithm in the prototype wireless CM system to achieve automatic band-pass filter selection and parameters updating. Explain and optimise the implementation procedure. Evaluate the implemented scheme on different bearing fault signals to confirm its effectiveness.
7) Investigate and compare common envelope detection methods, including the simple squared rectifier and the precise Hilbert transform based method. On this basis, find the possibility to optimise and speed up the implementation process of envelope detection. Implement, validate and evaluate the proposed methods for processing bearing fault signals.

1.5 Outline of thesis

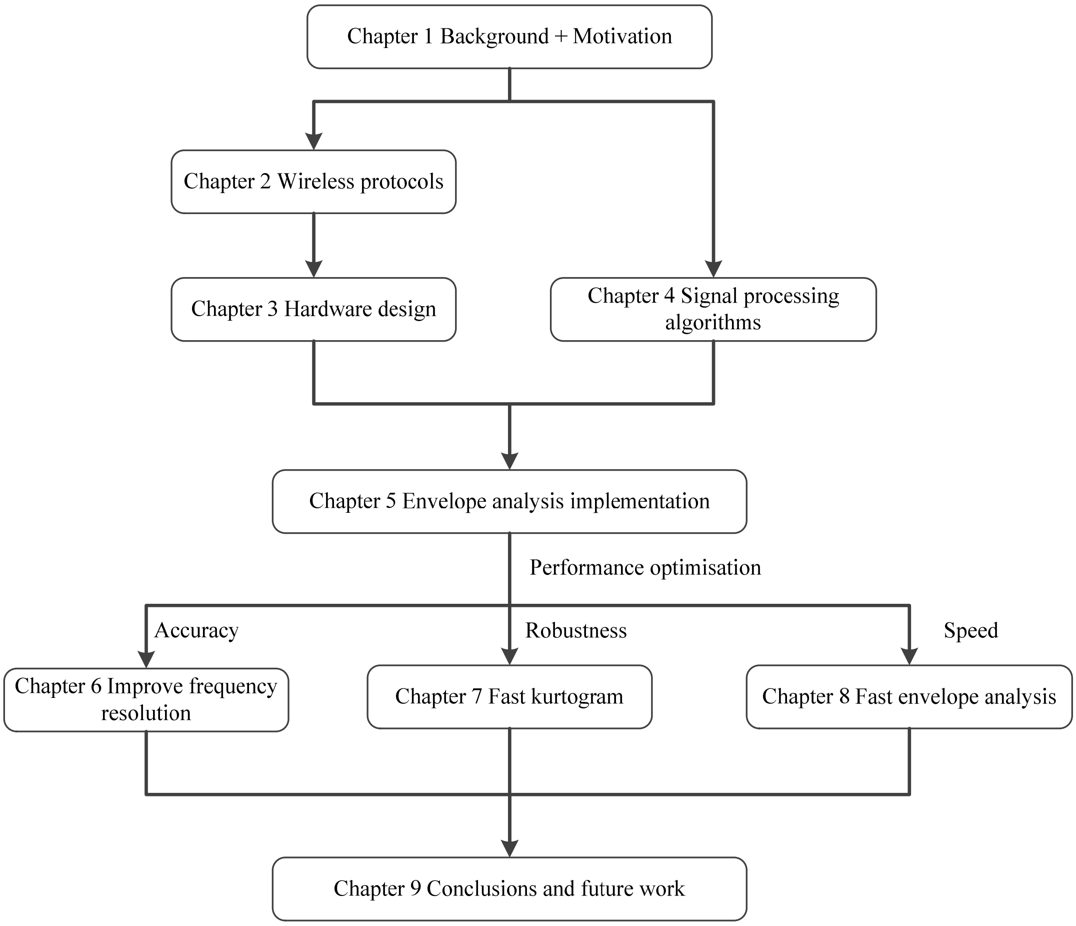

Chapter 1 Background + Motivation
Chapter 2 Wireless protocols
Chapter 3 Hardware design
Chapter 4 Signal processing algorithms
Chapter 5 Envelope analysis implementation
Performance optimisation
Accuracy
Robustness
Speed
Chapter 6 Improve frequency resolution
Chapter 7 Fast kurtogram
Chapter 8 Fast envelope analysis
Chapter 9 Conclusions and future work

Figure 1.3 Thesis structure

Based on the sequence of techniques development, the main structure of the thesis is outlined in Figure 1.3. In general, various wireless communication technologies are explored in Chapter 2 to select appreciate protocols for establishing vibration based wireless condition monitoring system. Then, a vibration based wireless sensor is prototyped in Chapter 3 for the validation of embedded algorithms. In Chapter 4, a host of signal processing algorithms are investigated to select suitable algorithms for processing impulsive and modulating type signals on the prototyped sensor node. On this basis, envelope analysis is selected for implementation on the proposed sensor node in Chapter 5. Then, the implementation of envelope analysis is improved in terms of computation accuracy, robustness, and speed in Chapter 6, Chapter 7 and Chapter 8, respectively. Finally, the conclusions are drawn and future works are suggested in Chapter 9.

CHAPTER 2

INVESTIGATION ON WIRELESS SENSOR NETWORKS AND THEIR APPLICATIONS FOR CONDITION MONITORING

Wireless network technology has gained enormous advancements in the past few years and a variety of wireless standards, such as cellular network, Wi-Fi, Bluetooth and ZigBee, are developed for different kinds of application. These kinds of technologies bring great benefits to our routine work and daily lives, as well as the industrial fields. Condition-based maintenance is one of such benefited areas.

This chapter firstly introduces the wireless sensor network. Then, the current popular wireless protocols for WSN are investigated and their advantages, drawbacks, and suitability for condition monitoring are discussed. In addition, recent research works in the wireless condition monitoring field are also reviewed and the latest products for industrial wireless condition monitoring are summarised. Finally, the up-to-date wireless sensor platforms are investigated as a reference for evaluation system set up in Chapter 3.

2.1 Introduction to wireless sensor network

According to application area and signal range, wireless network techniques can usually be categorised into four groups [54]: wireless personal area network (WPAN), wireless local area network (WLAN), wireless metropolitan area network (WMAN), and wireless wide area network (WWAN), as is shown in Figure 2.1. In general, the power consumption and cost increases with its signal range and bandwidth.

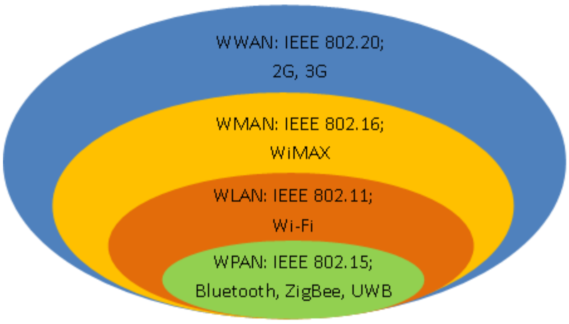


Figure 2.1 Wireless networks division

When talking about wireless network, the cellular network is the most familiar and widely used one, which is a kind of WWAN. It has gained tremendous development over the past decades with its successful applications in the commercial products. The cellular network can cover a large area and its transmission speed varies a lot depending on the standard and network coverage, ranging from 20 kbps in the second generation cellular network to several hundred Mbps in the latest fourth generation (4G) ones. However, it should be noted that the usage of cellular network will bring significant network traffic costs, which is not negligible for the applications that need to transmit a large volume of data. A reasonable application area for the cellular network is to use it for sending alerting information to personnel via Email/short message service [55].

In practice, the industrial machines are usually concentrated in a factory, which can be easily covered by a network like WLAN or WPAN. Such a network is usually called wireless sensor network (WSN), which includes sensor nodes, sink node, a connection to the Internet or satellite and a task manager node, as illustrated in Figure 2.2 [56]. Initially, monitoring data are collected by sensor nodes and routed by nearby nodes through wireless communication to the sink node. The sensor node manages operation of the wireless network and uploads the acquired data within the WSN to the task management node through Internet or satellite, allowing the end user to analyse the monitored data remotely.

Due to its great advantages like low cost, ease of maintenance and reduced connector failure, etc. discussed in Section 1.3.3.1, WSN has gained a wide range of applications in various areas, such as environment monitoring, healthcare, military and industrial condition monitoring and control applications. It plays an important role in the emerging Internet of Things (IOT) [57].

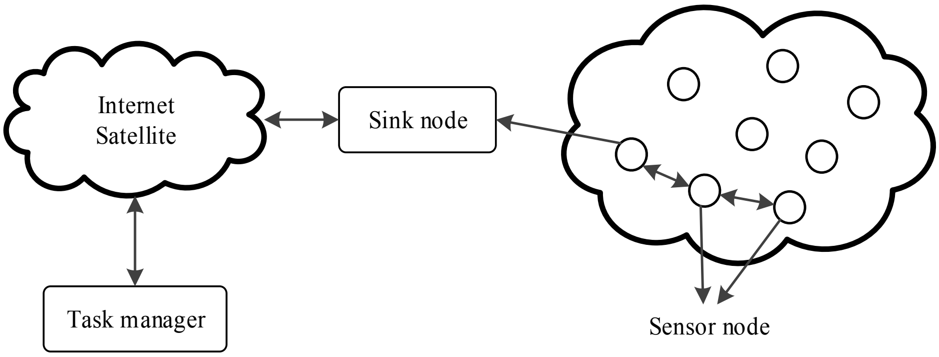


Figure 2.2 Structure of a typical wireless sensor network [56]

2.2 Wireless protocols for WSN

Due to various application area requirements, a host of wireless communication protocols have been developed and employed in the WSN, such as WiFi, Bluetooth, ZigBee and 6LoWPAN. Furthermore, there are also protocols developed specially for for satisfying special industrial requirements like WirelessHART and ISA100.11a. These protocols will be studied in the following sub sections.

2.2.1 WiFi and WiFi HaLow

WiFi is defined by WiFi Alliance as any "wireless local area network (WLAN) products that are based on the IEEE802.11 standards" [58]. In the real world, WiFi is much more often used in comparison with WLAN. WiFi has been created for the connection of two computers together and typically have a restricted signal range within the scope of one building and one room (indoor range is approximately 40 metres).

WiFi operates on the 2.4/5 GHz frequency band, one of the industrial, scientific and medical (ISM) license-free frequency bands. This means it does not produce additional traffic fees. In terms of data transmission speed, the theoretical data rate ranges from several Mbps in IEEE802.11b to 600 Mbps in IEEE 80.2.11n. Note that this data rate can be affected significantly by the signal strength and number of users coexist the network.

Fast data transmission speed is an advantage of Wi-Fi and it has been explored for remote monitoring systems [59]. But issues like significant power consumption and long latency have made Wi-Fi not a good candidate for battery powered wireless sensor node applications. Furthermore, due to the wide availability of Wi-Fi in the commercial area, its security is also a big concern when applying it for industrial applications.

In recent years, a new low power Wi-Fi solution, namely Wi-Fi HaLow, is introduced by the Wi-Fi Alliance® based on IEEE 802.11ah specification to satisfy the requirement in the scenario of Internet of Things (IOT) applications [60]. Wi-Fi HaLow operates in 900 MHz frequency band, offering longer range, lower power connectivity. The transmission range of Wi-Fi HaLow's is expected to be twice of today's Wi-Fi [61]. In addition, it is expected to provide a more robust connection in challenging environments due to that it has better penetration capability to walls or other barriers.

In terms of transmission speed, the lowest rates of Wi-Fi HaLow is only 150 kbps, achieved by using a 1 MHz channel, and can be up to 18 Mbps by employing a 4 MHz-wide channel. Due to the wide availability of the existing Wi-Fi, Wi-Fi HaLow is expected to gain wide application quickly and can be a good candidate for wireless condition monitoring systems.

2.2.2 Bluetooth classic and Bluetooth low energy

Bluetooth is an open wireless technology standard for exchanging data over short distances based on IEEE802.15.1 standard. It is designed for continuous, streaming data applications and has gained wide usage in computer centred peripheral applications.

The network of Bluetooth devices is known as a piconet, which can contain one master and 7 slaves Bluetooth devices. The connection of two or more piconets forms a scatternet in which certain devices simultaneously play the master role in one piconet and the slave role in another. Bluetooth works in 2.4 GHz frequency band and can achieve a maximum data rate of 3 Mbit/s for version 2.0 + enhanced data rate (EDR).

The power consumption of Bluetooth is lower than Wi-Fi, making it more suitable for battery powered wireless applications. A commercial off-the-shelf product, called Intel Mote [62], was designed based on the top of a point-to-point Bluetooth baseband in order to be used in industrial monitoring using vibration sensors. The Intel Mote is high-performance sensor node platform integrated 12 MHz CPU, 64 kB SRAM, and 512 kB Flash. However, it is reported that the Bluetooth-based WSN applications has decreased due to the high complexity and inadequate power characteristics of Bluetooth [63].

On the other hand, a new specification, namely Bluetooth low energy (BLE), is becoming popular in recent years with the emerging of smartphones, wearable devices and IOT. BLE is an ultralow-power technology addressing devices with very low battery capacity. It allows for a maximum data rate of up to 1 Mbps over distances of about 50 m in the 2.4 GHz band. Although BLE is similar to Bluetooth, it has made some important improvement to satisfy IOT applications. For example, BLE has a variable-length packet structure, compared to Bluetooth's fixed length.

BLE is suitable for applications that require the episodic or periodic transmission of small amounts of data. The maximum power consumption of BLE is about 15 mA but the average power consumption can be only about 1 uA. This is achieved by making the actual connection time of only a few mili-seconds. Although the theoretical data rate is up to 1 Mbps, the actual throughput is much less due to packet overhead, implementation constraints, and processing delays. For consultancy, one measurement of maximum application layer throughput is about 58.48 kbps in [64].

In addition, the network capacity for BLE is enhanced compared to Bluetooth classic, making it more suitable for IOT applications that usually include many sensor nodes. According to [64], the number of simultaneous slaves per master can be up to more than 5000. Moreover, BLE might support mesh network in the future version [65].

2.2.3 Protocols based on IEEE802.15.4

IEEE 802.15.4 is a technique designed to provide simple wireless communications with relatively short range, limited power, relaxed data throughput, low production cost and small size. It originates from the low-rate wireless personal area networks (LR-WPAN) standard, which was inspired by the requirement to enable inexpensive WSNs for remote monitoring and control of noncritical functions in the residential, commercial, and industrial applications.

Three frequency bands are supported by the IEEE 802.15.4 standard, as given in Table 2.1, including two low bands at 868/915 MHz and a high band at 2.4 GHz. The data transmission rates vary from 20 kbps to 250 kbps depending on the operating band. The most commonly used band is 2.4 GHz one, which is the same as the Wi-Fi and Bluetooth.

One major benefit of 802.15.4 is that they can form a mesh network, where each sensor does not only transmit its own data but also serve as a relay for other sensors. Due to its attractive characteristics and the emerging IOT applications, a number of techniques have been developed to operate on top of 802.15.4 standard, such as ZigBee, WirelessHART, 6LoWPAN and the recently Thread from Google Nest.

Table 2.1 IEEE 802.15.4 standard details

Frequency band (MHz)	Coverage	Maximum theoretical data rate (kbps)	Channels
868.0-868.6	Europe	20	1
902-928	Americas	40	10
2400-2483.5	Worldwide	250	16

2.2.3.1 ZigBee

ZigBee is a mesh-networking standard targeted at industrial control and monitoring, building, and home automation, embedded sensing, and energy system automation. The advantages of ZigBee include extremely low energy consumption and support for several different topologies, making it a good candidate for wireless sensor network applications [63]. According to a comparison in [66], it is found that ZigBee is easier to implement than Wi-Fi and Bluetooth, which makes it more suitable for sensor networking applications due to their limited memory and computational capacity. Furthermore, ZigBee provides better efficiency for data size smaller than 102 bytes [66] and better performance in low SNR environment than Wi-Fi and Bluetooth [67].

As an early starter in the WSN area, ZigBee has been supported by many semiconductor companies, such as TI, NXP and Silicon Labs. Its successful application can be found in motor broken bar monitoring [68], energy monitoring [24] and fault diagnosis based on motor stator current and vibration signals [25]. However, it is reported that ZigBee cannot satisfy strict industrial-grade reliability and robustness [30, 45].

2.2.3.2 WirelessHART

WirelessHART is a protocol especially for industrial wireless network applications approved in June 2007. It is an extension of the highway addressable remote transducer (HART) protocol. In 2010, WirelessHART was approved by the International Electrotechnical Commission (IEC), making it first wireless international standard as IEC 62591[70].

WirelessHART operates on the 2.4 GHz frequency band and utilises a time synchronised, self-organizing, and self-healing mesh architecture. Due to special considerations for the industrial requirement, it has been reported more superior for industrial applications than ZigBee in terms of robustness, co-existence, power consumption and security [69]. However, the commercially available wireless modules for WirelessHART are still rare when the author starts prototyping the wireless CM system.

2.2.3.3 ISA100.11a

ISA100.11a is another wireless system standard for industrial automation, which is also based on IEEE802.15.4. A hybrid medium access control (MAC) layer is adopted in ISA100.11a by combining time division multiple access (TDMA) and carrier sense multiple access with collision avoidance (CSMA/CA) [71]. The flexible time slots with configurable length can satisfy different system requirements and frequency hopping and channel blacklisting techniques can promise robust wireless communications. Moreover, ISA100.11a can be mixed with existing wired networks, including Modbus, Profibus, and HART, using tunnelling techniques [72].

Similar as WirelessHART, ISA100.11a outperforms ZigBee for industrial applications. However, the commercially available wireless modules for ISA100.11a are still rare. In addition, it is reported that it is quite technically challenging to implement the full ISA100.11a stack on low-cost hardware [73].

2.2.3.4 6LoWPAN and Thread

6LoWPAN is built with the aim of standard Internet protocol (IP) version 6 (IPv6) communication over low power wireless IEEE 802.15.4 networks [74]. Its advantages lie in its capability to communicate directly with other IP devices locally or via the existing and well-defined IP network (e.g., Ethernet). Thereafter, it can utilise the existing architecture and security, application level data model and services, network management tools etc. In order to send and receive IPv6 packets over the bandwidth limited IEEE 802.15.4 based networks, 6LoWPAN has employed encapsulation and header compression mechanisms to reduce packet size.

Another competitive protocol named Thread is launched in 2015 [75]. It is a royalty-free protocol and runs over 6LoWPAN specialised for home automation. Due to good commercial support from companies like Google's Nest, Thread is expected to provide better security and connectivity support than the basic 6LoWPAN.

2.2.4 Comparison of wireless protocols

In the above sections, several popular wireless protocols commonly employed in WSN are investigated. They have their own advantages for satisfying some specific requirements but maybe not so suitable for other scenarios. For the condition monitoring application that requires a large volume of data and high data rates, Wi-Fi can be a good option supposing the AC/DC power supply can be easily accessed.

Meanwhile, the low-power wireless protocols bring another kind of condition monitoring scenarios, in which the wireless sensor node can be powered by a battery for a long time, e.g. several months or several years, or can harvest energy directly from its ambient, such as exhausted heat [18, 19] and mechanical vibration [3, 4]. Such wireless CM can greatly ease the installation, maintenance and possible redeployment process. In factory automation systems, most industrial monitoring and control are generally not large and does not need frequent updating, making low-power wireless protocols a suitable solution for such scenarios.

The popular low-power wireless communication protocols are summarised in Table 2.2 in terms of operating frequency band, data rate, range, network capacity and support level. Basically, the lower operating frequency enables a longer range for the same transmission power condition and higher data rate causes higher power consumption. WirelessHART and ISA.100.11a have already provided industrial level support, which makes them more suitable for applications that have strict reliability requirements. Meanwhile, it is quite challenging to implement these two robust protocols on resource constraint microcontrollers [36], making their solutions costly. Note that these two protocols are not ideal for industrial applications and further improvements are required with more experience being gained in practical applications. For example, to solve the drawbacks in the centralised management scheme, Zand et al presented a distributed network for real-time monitoring and process control applications in wireless industrial automation [76].

Table 2.2 Comparison of investigated wireless communication protocols

Protocol	Operating Band (Hz)	Data rate (bps)	Range	Support level
BLE	2.4G	1 M	Short	Commercial
Wi-Fi HaLow	900M	150 k - 18 M	Long	Commercial
ZigBee	2.4G 915M 868M	20 k – 250 k	Medium to long	Commercial
WirelessHART				Industrial
ISA100.11a				Industrial
Thread				Commercial

Other protocols like ZigBee, BLE, Wi-Fi HaLow and Thread have gained good commercial support and thus their relevant solutions can be provided at a low price. They can be employed in the situations where reliability requirements are not so strict. However, it is difficult to tell which protocol will win out in the future. Possibly they will co-exist and find their own application areas in the future. In this thesis, ZigBee is employed to build the wireless sensor network due to the solutions and supports for ZigBee have been mature whilst other solutions are still under development when the author starts the work.

2.3 Review of WSN for condition monitoring

Attracted by the potential benefits and opportunities of WSN, many research works have been carried out for condition monitoring purposes and some industrial products have become available on the market. This section will review work and products in the condition monitoring field with a focus on vibration based techniques.

2.3.1 Review of researcher’s works

In 2007, F. Salvadori and M. De Campos, et al [77], proposed a monitoring system with both wired and wireless communication approaches for energy usage evaluation in industrial electric systems. The system provided two kinds of intelligent nodes: one battery powered intelligent sensors module and one high-performance remote data acquisition unit. The first type is designed to acquire static signals and the second type is employed for acquiring data with high sampling rate. Through packet losses experiments, it shows the first type module is able to capture up to 11 samples per cycle while the second one can achieve 68 samples per cycle, which allowed analysing up to the 34th harmonics. Based on the above system, they added the dynamic power management with scheduled switching mode (DPM-SSM) protocol for the sensor nodes to extend the network lifetime [78]. By switching the node to a sleep state after a data packet transmission, the method takes advantage of the battery capacity recovery effect. The experimental results show that even for very short sleep times (i.e., 1 ms), the sensor node is able to transmit about one-third more packets compared with a sensor node without DPM-SSM. The sensor lifetime could be further extended by applying longer sleep periods.

In 2007, Xin Xue [79] proposed a CM system for three-phase induction motors, especially small motors, using a WSN to measure the temperature and the vibration signals. The sensor node was from Crossbow Inc., which hosted an Atmel 128L CPU running the Tiny Operating System (TinyOS). In the experiment, the sensors were placed on the rotor and wirelessly transmitted to a base station located 6 feet away. The results showed that the wireless sensor data was not as reliable as wired sensors. However, it provided an important way to collect data when wired sensors cannot reach the place. The magnetic field caused less than 10% packet loss to the wireless communication between the sensor nodes and the base station. Note that the sampling rate of the temperature sensor and acceleration sensors were set at 10 Hz and 100 Hz respectively.

In 2005, B. Lu et al. [80] proposed an industrial plant energy management system with a WSN architecture, which achieved the evaluation of the industrial motor energy usage using only motor terminal quantities through WSN and nameplate information without interfering with the motor’s normal operation. Later, they upgraded the energy management scheme to a closed-loop one and discussed the scheme and the design of the sensor in [81] and [82] separately. The applicability of the proposed scheme is analysed in terms of data throughput and communication latency, power consumption, energy evaluation and condition monitoring accuracy. A simplified demo system has been constructed and details of the design and implementation of the wireless sensor device are given in [83].

In 2009, B. Lu et al. [24] improved the system and gave a more mature and comprehensive solution, which identified the synergies between wireless sensor networks (WSNs) and nonintrusive electrical-signal-based motor signature analysis and proposed a scheme of applying WSNs in online and remote energy monitoring and fault diagnostics for industrial motor systems. The paper also provided detailed analysis to address the real-world challenges in designing and deploying WSNs in practice, including wireless-link-quality dynamics, noise and interference and environmental impact on communication range and reliability. The overall system feasibility is investigated through a series of laboratory experiments and field tests.

In 2007, P. S. Sausen et al. [84] presented a dynamic power management (DPM) technique derived from a more realistic analysis of the battery capacity recovery effect and the switching energy. The results showed the potential for improving the battery lifetime by taking advantage of the battery recovery effect when a node transitions to a sleeping state, and mostly when transitions are scheduled after packet transmissions. The proposed technique provided several DPM modes which would be triggered depending on the battery remaining capacity. Simulations results showed this technique could provide real battery power recovery without compromising the timeliness of the applications running on the wireless sensor network.

In 2010, J. Pedro Amaro et al. [85] developed a low-cost radio frequency communication system to monitor the operation conditions of three-phase squirrel cage induction motors. The system was based on a low-cost electronic device that can acquire and pre-process current, voltages and temperatures and transmit processed key information related to the motor operation conditions. Additionally, it protected the motor from abnormal operation conditions and managed its stator winding connection mode. Hardware platforms for WSN were briefly described and compared. A Nordic nRF9E5 device was selected for this system and was tested in terms of power consumption and radio distance.

In 2010, Yuan Hong-fang [86] presented a ZigBee WSN for vibration signal acquisition of pump. Two vibration sensors ADXL105 and ADXL78 with the range of ±5g and ±35g separately were used to ensure the scope of measure and improve the measure precision of the signal. Normally, only three symptom parameters in the frequency domain will be sent for fault recognition; therefore, the transmission load of WSN could be decreased. If an abnormal state was detected, original data could be sent to a fielder server for further analysis and diagnosis. The ZigBee module in this system is developed with ZigBee processor CC2480, RF front end CC2591, and microcontroller ATmega128L.

In 2012, L. Hou and N. W. Bergmann performed FFT of 512 points on the vibration signal and extracted 12 most frequently occurring frequency components in the frequency domain as the fault feature for data transmission in a wireless system [25]. After data fusion methods were added, the payload transmission data was reduced from 1024 bytes to 8 bytes, which is a high rate compression with a data reduction of more than 99%. However, the sampling rate for the vibration signal was only at 3.1 kHz, which would be a bit low for some applications to accurately detect results obtained.

These above investigations have shown a trend of processing the data with a high sampling rate on the distributed wireless sensor nodes instead of transmitting the raw data directly, in which case, the wireless bandwidth requirements would be greatly decreased and the power consumption of the sensor nodes can be significantly reduced.

2.3.2 Investigation of available wireless CM products

Some leading industrial instrumentation companies, such as GE, ABB, Emerson, Siemens, and Honeywell, have released their wireless condition monitoring products. These products are briefly described below.

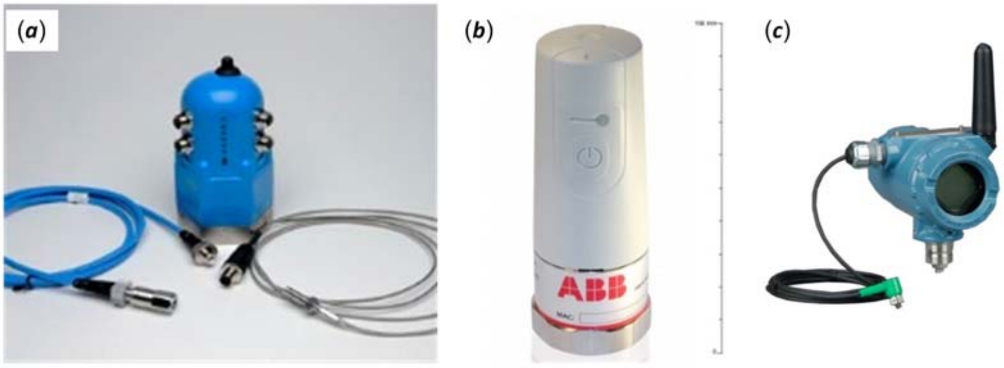


Figure 2.3 (a) Essential Insight mesh wireless condition monitoring platform, (b) WiMon 100 and (c) CSI 9420 wireless vibration transmitter

GE Measurement & Control. The Essential Insight mesh wireless condition monitoring platform (shown in Figure 2.3(a)) of GE Measurement & Control [29] is based on ISA100.11a protocol. It is capable of collecting acceleration and temperature signal with the resolution at 16 bit and 24 bit respectively. For static data, the update rate can be set from 1 minute to once per day and for the dynamic data, it can be updated from once per day to once per every 30 days. Also, both the static and dynamic data can be updated on the user's demand. It is worth mentioning that the sampling rate can be set as high as 51.2 kHz.

ABB. The WiMon 100 (shown in Figure 2.3(b)) from ABB is based on WirelessHART [27]. It includes a vibration sensor, a temperature sensor, a long life battery and a WirelessHART radio. It can provide a conversion resolution at 16 bit with sampling rate at 5,859 kSPS.

Emerson. The CSI 9420 wireless vibration transmitter (Figure 2.3(c)) from Emerson [28] is based on WirelessHART protocol. It can acquire 4096 points of the raw waveform on demand with sampling rate as high as 51.2 kHz. The upload interval can be set from 1 minute to 60 minutes in standard mode and up to 24 hours at power save mode. In addition, this device provides three kinds of data compression techniques to extract useful information whilst facilitating easier transmission over the network. Such techniques are expected to accelerate responsiveness of the system whilst reducing power consumption.

The first compression technique is FFT analysis, which reduces the data size by over 60%. The second technique is named thumbnail spectrum which contains the same frequency and amplitude information as the high-resolution spectrum, while the data set is compressed by an additional 98%. Figure 2.4 (a) and (b) illustrates an example of the FFT analysis and thumbnail spectrum.

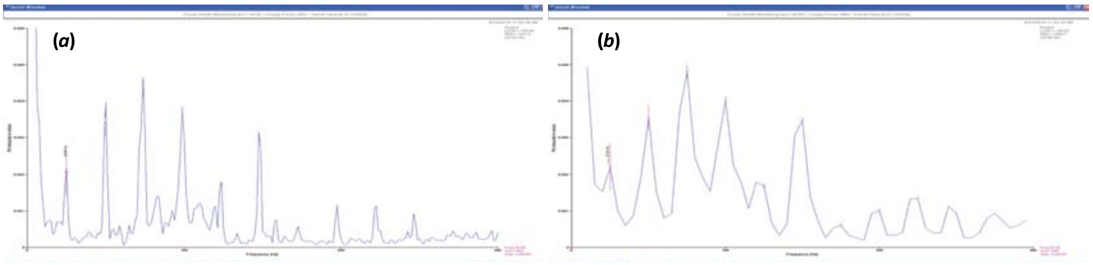


Figure 2.4 Spectrum results of CSI 9420: (a) high-resolution spectrum and (b) thumbnail spectrum [28]

The third technique is energy bands, which divides the spectrum into three predetermined energy bands (as shown Table 2.3) and then calculates the vibration energy within each band. By dividing the spectrum into energy bands, the user can isolate frequencies associated with different categories of faults.

Table 2.3 Energy bands with frequency ranges

Band	Fault types	Range
1	Rotor vibration: imbalance, misalignment (also defects on belt drives)	2-65Hz
2	Rotor harmonics: looseness, electrical faults, blade and vane pass	65-300Hz
3	High frequency: bearing and gear defects, lubrication and cavitation	300-1000Hz

GST. The A306 and A302 (shown in Figure 2.5) from Global Sensor Technology (GST) [87] is based on IEEE802.15.4 protocol. A306 is capable of acquiring three channels of vibration and temperature signals with a sampling rate of up to 10 kHz and 16-bit resolution. In order to solve the bandwidth limitations of IEEE802.15.4, a large memory of 1 gigabytes (GB) is provided for temporarily storing the large dataset acquired with the high sampling rate. Unlike A306, A302 includes a tri-axil MEMS accelerometer and is self-powered while the maximum sampling rate is about 4 kHz and its resolution is 12 bit.

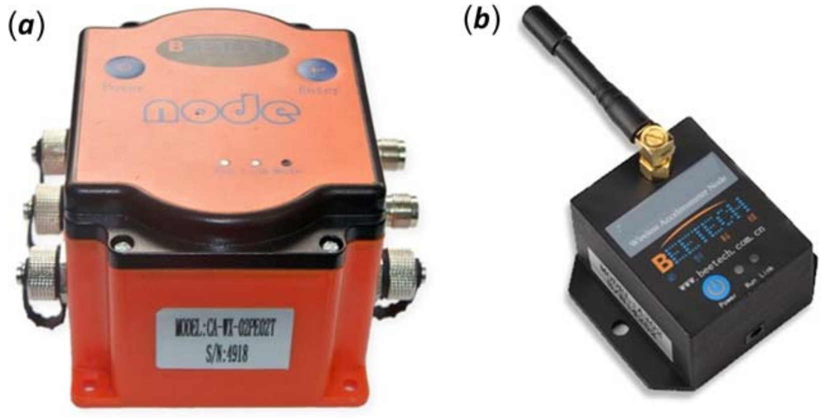


Figure 2.5 Wireless sensor nodes from GST: (a) A306 model (b) A302 model

MicroStrain. A wireless IEPE sensor node (Figure 2.8(b)) is provided from MicroStrain [88]. The node provides 24-bit resolution and up to 104 kHz sampling rates, operating at IEEE802.15.4 radio. Besides, it integrates a low-pass Butterworth filter with programmable anti-aliasing filter ranging from 26 Hz to 33 kHz.

Figure 2.6 Wireless IEPE sensor node from MicroStrain

Siemens and Honeywell. There are also wireless solutions from other companies, like Siemens [89] and Honeywell [90]. Differently, Siemens adopts the WirelessHART protocol while Honeywell employs the ISA100.11a. The wireless transmitters from Siemens and Honeywell are shown in Figure 2.7 (a) and (b), respectively. Only products for collecting temperature and pressure signals have been found.

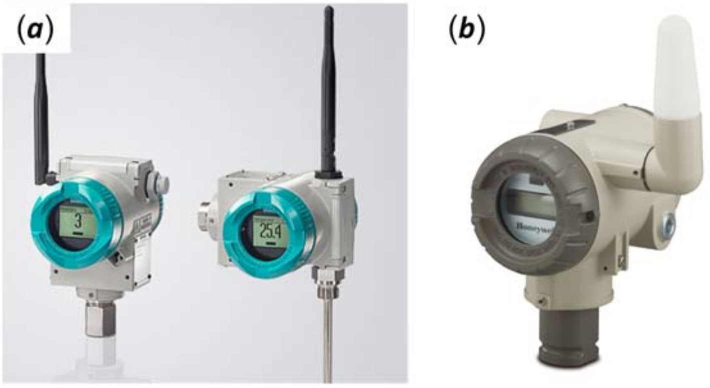


Figure 2.7 (a) Wireless transmitters from Siemens and (b) wireless transmitters from Honeywell

Apart from the above products that are based on IEEE802.15.4, there are also some devices based on the sub-GHz frequency band, which can provide a longer transmission range than those wireless techniques working on 2.4 GHz.

ADI. The wireless vibration sensor ADIS16229 [26] (Figure 2.8(a)) from Analog Devices Inc. (ADI) operates at 902.5 MHz to 928 MHz frequency band. It includes dual-axis digital micro-electro-mechanical systems (MEMS) with resonant frequency up to 5.5 kHz and sample rate up to 20 kSPS. In addition, it features a programmable decimation filter and selectable windowing function in time domain signal processing and a 512-point, real-valued FFT; FFT magnitude averaging; and programmable spectral alarms in frequency domain processing.

PCB Piezotronics. Echo® Wireless Vibration Sensor Model 670A01 (as shown in Figure 2.8(b) from PCB Piezotronics [91], is based on 900MHz ISM band. The sensor is designed to wake up at preprogrammed intervals, measures a series of the overall vibration signal, transmit them to a receiver for processing and goes back to sleep to conserve battery power. In detail, it can sample the vibration with 16-bit resolution, calculate the RMS acceleration at the frequency range from 2.2 to 15 kHz and transmit the data using the 900MHz ISM band with a data rate at 20 bps.

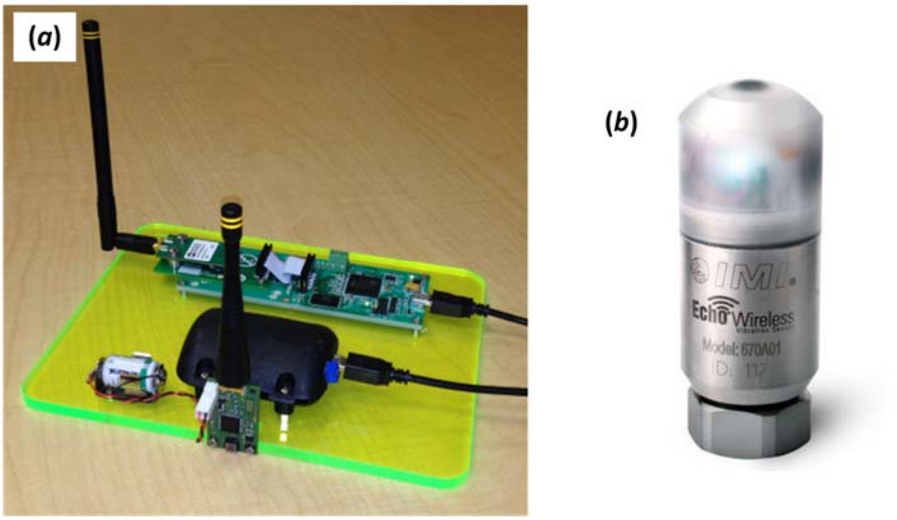


Figure 2.8 (a) ADIS16229 from ADI and (b) Echo® wireless vibration sensor model 670A01

These above products are the latest developments and most of them have been introduced within the last five years. Some of them have provided high specifications and even integrated intelligent signal processing algorithms, like the thumbnail spectrum on CSI9420 and FFT on ADIS16229. However, the majority of them don’t allow users to embed customised signal processing algorithms.

2.4 Investigation on wireless sensor platforms

Attracted by the great potential applications of WSN, many leading semiconductor manufacturers have provided some system on chip (SOC) wireless modules, which include both the microcontroller and the RF components, e.g., CC2530 from TI [92] and JN5148 from NXP [93]. These modules enable users to implement WSNs with minimum time to market and low development cost.

A number of SOC for WSN (built on IEEE802.15.4) is summarised in Table 2.4, including the sensitivity and power consumption features. It can be observed the sensitivity and power consumption of the chips vary a little. It is reported that SOC solutions significantly reduced the amount of power consumption in the sleep mode, but not so much in transmission and receiving modes [24].

Table 2.4 Commercially available SOC for WSN

Corp.	RF Module	Sensitivity (dBm)	Power Consumption(A)		
			Rx	Tx	Standby
TI	CC2530	-97	20.5m	29m(1dBm)	0.4u
	CC2538	-97	20m	24m(0dBm)	0.4u
Freescale	KW20	-102	15m	15m(0dBm)	0.6u
	MC13226V	-100	21m	29m(0dBm)	0.85u
	MC13237	-94	26.6m	22.3m(0dBm)	<1u
Silicon Labs	EM357	-102	26m	28.5m(0dBm)	0.4u
ST	STM32W108CC	-99	27m	31m(3dBm)	0.4u
NXP	JN5148	-95	17.5m	15.0m	0.1u
	JN5168	-95	17m	15.3m	0.1u
Atmel	ATmega256RFR2	-100	12.5m	14.5m	<0.7u
Radio Pulse	MG2470	-99	--	--	1u

As we intend to embed intelligent signal processing algorithms on the sensor node, the processor specifications of the above SOC are given in Table 2.5. The CPU architecture of the processor is a key factor influencing the computing capabilities and power consumptions of the sensor node. Various CPU structures have been adopted by different chips, from the earliest 8 bit 8051 core to the latest Cortex-M3 and Cortex-M4 architecture. For high computing capability and low power consumption considerations, it is advised to choose the chip that employs latest Cortex-M3 or even Cortex-M4 core. Among these chips, KW20 from Freescale is the most high-performance one, which adopts the latest Cortex-M4 core running at a maximum speed up to 50 MHz. Most importantly, the DSP core inside the chip could greatly improve the computing capabilities. The data memory is another important factor influencing the computing capabilities of the node. From Table 2.5, it can be seen that JN5148, MC13226, LTC5800 and KW20 outstand themselves in this respect.

Table 2.5 Wireless sensor network chips (processor)

Corporation	RF Module	Processor			
		Architecture	Flash (kB)	SRAM (kB)	Speed (MHz)
TI	CC2530	8 bit, 8051 core	256	8	32
	CC2538	32 bit, Cortex-M3	512	32	32
Freescale	KW20	32 bit, Cortex-M4	512	64	50
	MC13226	32 bit, ARM7	128	96	26
	MC13237	8 bit, HCS08	128	8	32
Silicon Labs	EM357	32 bit, Cortex-M3	192	12	24
ST	STM32W108CC	32 bit, Cortex-M3	192	16	24
NXP	JN5148	32 bit, RISC	128	128	32
	JN5168	32 bit, RISC	256	32	32
Atmel	ATmega256RFR2	8 bit, AVR	256	32	16
Radio Pulse	MG2470	8 bit, AVR	64	6	16

In general, IEEE802.15.4 provides a maximum data rate of 250 kbps, however, some chips have broken through this limitation. For example, the JN5148 can provide high data rate modes at 500 and 667 kbps; MG2470 can provide a maximum data rate of 1 Mbps, and the radio transceiver of ATmega2564RFR2 can provide high data rates up to 2 Mbps. However, it should be noted that higher data transmission rate usually consumes larger power.

Usually, the above chips only provide solutions for ZigBee or 6LoWPAN and the solutions for WirelssHart or ISA100.11a are still rare. Linear Technology's Dust Networks has a series of products named SmartMesh that can support WirelessHART. The SmartMesh family consists of several products to simplify system development: LTC5800 (SOC) and LTP5900 (module). The LTC5800 integrates all radio circuitry components, including an onboard power amplifier and a Cortex-M3 microprocessor with 512 kB Flash and 72 kB SRAM, requiring only power, ground and an antenna for robust wireless connectivity.

Besides the above SOC solutions, there are also some advanced wireless sensor nodes developed by some universities of institutes using a commercially available microcontroller, RF transceiver and other electronic components, such as MICA series from UCB [94] and XYZ from Yale [95]. Table 2.6 lists the microcontroller (MCU) chips and RF modules for different WSN nodes, from which, it can be observed that new generation WSN nodes have been developed with both a high-performance processor like ARM and an advanced RF module that contains 802.15.4/ZigBee chips. Specifically, the CC2420 chip from TI is widely employed due to its low power and high-performance design. The newer version, named CC2530 shown in Table 2.4, provides even lower power consumption and higher performance.

Table 2.6 Wireless sensor network nodes

Platform	Microcontroller			Memory Size (kBytes)		RF chip
	Model	Architecture	Frequency (MHz)	Program	Data	Model
MICA2	Atmel128L	8 bit AVR	8	128	4	TR1000
MICAZ	Atmel128L	8 bit AVR	8	128	4	CC2420
Telos	MSP430F149	16 bit RISC	8	60	2	CC2420
WMNMCS-5	ADuC832/834	8 bit 8052	1.3	62	6	nRF905
Preon 32	--	32 bit Cortex M3	72	256	64	AT86RF231
BTNode	Atmel128L	8 bit AVR	8	128	180	CC1000
SUN Spot	ARM 920T	32 bit ARM9	180	4000	512	802.15.4
Imote2.0	PXA271	32 bit RISC	11-400	32000	32000	CC2420
XYZ	ML67	32 bit RISC	60	256	32	CC2420

Considering the complexity of the WSN, several specialised companies have provided integrated wireless module solutions to ease the development process and shorten the time to market. The benefits for employing an integrated wireless module specially for handling wireless network task include:

- **Save memory consumption**. A robust wireless network stack typically consumes large Flash and RAM memory. For instance, the implementation of a full ZigBee stack for a coordinator on a CC2530 SOC consumes approximately 120 kB of Flash and 8 kB of RAM [96], which is a significant usage even for the advanced processor TM4C1233H6PM employed in this thesis.
- **Reduce interference to signal processing**. The wireless stack operation involves several kinds of operations; including sending data, receiving data, acknowledgement and retransmission etc. Such operations may interrupt the signal processing and reduce their computation efficiency.
- **Avoid software related faults**. It is never an easy job to achieve a robust network stack. Software related faults may result in a long development period.

The Xbee wireless RF module is an outstanding example and is employed future in Chapter 3. The Xbee modules provide a variety of wireless solutions, such as ZigBee, IEEE802.15.4, Wi-Fi and sub 1GHz RF communication. After a proper configuration through the X-CTU tool provided by the company, the modules can set up a network automatically. Moreover, these modules are pin-to-pin compatible with one another, which means the wireless protocol could be easily updated by utilising a different kind of Xbee module.

2.5 Summary

Through the investigation in this chapter, it has been found that a variety of wireless protocols are available for satisfying different application requirements. Especially, the low power consumption wireless protocols are gaining fast development in recent years with the emerging IOT. For industrial applications that have strict reliability requirement, WirlessHart and ISA100.11a are good choices, which have been designed to address the critical needs of industry for reliable, robust and secure wireless communication. For other protocols, such as WiFi HaLow, BLE, ZigBee, and Thread, have good commercial support, resulting in their relevant solutions to be provided at low cost. They can be utilised for the condition monitoring area that requires less strict reliability but is more sensitive to costs.

The conflict between high data throughput and limited bandwidth in the WSN can be mitigated by embedding intelligent signal processing algorithms on the wireless sensor node and just transmitting the extracted features that have much smaller size than the raw data. For evaluating the signal processing algorithms, a wireless CM system will be prototyped in Chapter 3. Previous studies have shown satisfactory results for such schemes and more signal processing algorithms will be explored in Chapter 4.

CHAPTER 3

HARDWARE DESIGN FOR A WIRELESS CONDITION MONITORING SYSTEM

This chapter presents the hardware design of a prototype wireless CM system, which will be employed for the evaluation of embedding signal processing algorithms in later chapters. The overall structure of the proposed wireless condition monitoring system and its working mechanism is firstly explained. Then, the design of the wireless sensor node is explained as the emphasis. A state-of-the-art Cortex-M4F MCU is employed as the core processor. The hardware is developed with the idea of stackable boards with different functions, in which way, each part can be upgraded easily.

3.1 Overall structure of the wireless CM system

A prototype wireless CM system based on ZigBee is presented in Figure 3.1. It consists of one sink node and several sensor nodes. Vibration signals are collected and locally processed at the sensor node and the processed results are transmitted over the ZigBee network to the sink node. Then, the host computer reads data from the sink node and the analysed envelope spectrum of different sensor nodes are visualised.

The sink node is the heart of the ZigBee network. It is composed of an XBee Pro module for wireless connectivity and an FT232 board for communicating with the host computer through a USB port. All the data from the sensor nodes are transmitted to the host computer through the sink node and the commands from the host computer are routed to the sensor nodes through the sink node as well.

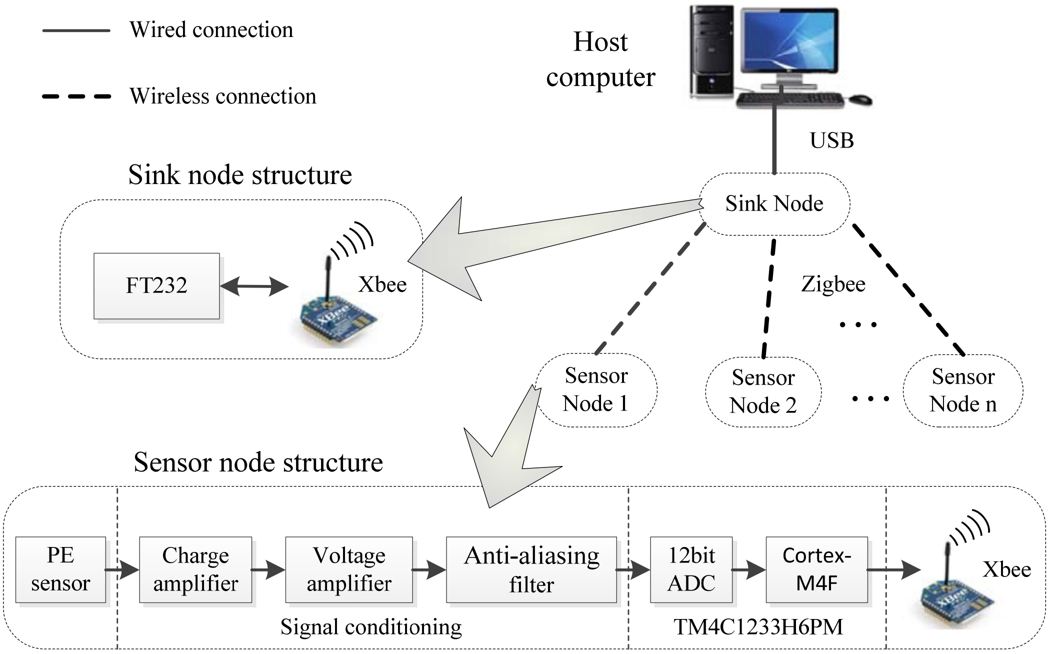


Figure 3.1 Structure of the wireless CM system

The sensor node is the key design of this thesis and will be explained in detail in the following sections. As wireless sensors are required to be deployed in large networks and need to be low cost and have low power consumption, they are usually designed with limited memory size and restricted computational capabilities [22]. For instance, the popular wireless module XBee ®ZB has only 2 kilobytes (kB) RAM and an 8-bit processor running at 50.33 MHz [97]. The system-on-chip (SOC) solution CC2530 has

8 kB RAM and an 8-bit processor running at 32 MHz [92]. These wireless modules are usually not powerful enough for signal processing methods like real-time FFT analysis. Therefore, an additional external processor, which has much better performance yet still has low power consumption, is needed to fulfil the complex signal processing algorithms.

For rapid prototyping, a Launchpad development board is selected as the main base board and several booster pack boards are stacked on this board to extend different kinds of functions. Figure 3.2 gives the picture for the assembled wireless sensor nodes, which consists of four boards: main baseboard, vibration sensor conditioning, wireless extension and battery booster pack, which will be explained separately in the following sections. The dimensions of assembled sensor node are measured as about 66 mm × 51 mm × 50 mm (length × width × height).

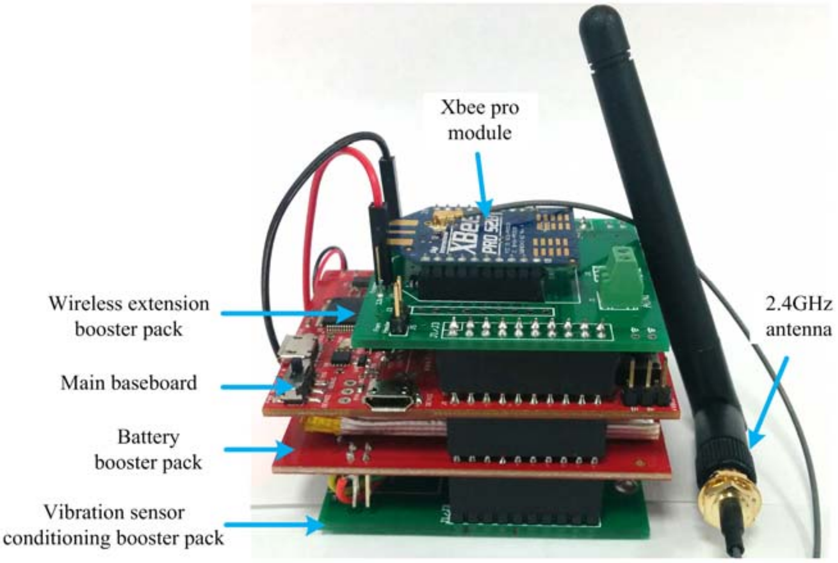

Xbee pro module
Wireless extension booster pack
Main baseboard
Battery booster pack
Vibration sensor conditioning booster pack
2.4GHz antenna

Figure 3.2 Assembled wireless sensor node

3.2 Main baseboard

The earlier real-time DSP applications are accomplished using dedicated DSP processors, such as TMS320C6x and TMS320C5x from Texas Instruments [98], [99], Blackfin and SHARC from ADI [100]. These processors are usually optimised for DSP processing, including dedicated structure or hardware units, such as pipeline, multiply accumulate unit, floating point unit (FPU) and FFT co-processors. Such optimisations have made them especially efficient and capable of real-time signal processing but have also made them complicated and expensive too. Furthermore, one needs special knowledge of the hardware structure and requires programming skills to achieve high computation performance.

With the fast development of processors and requirement for ubiquitous computing, more and more microcontrollers have become capable of performing signal processing with the integration of DSP unit or even FPU unit. This has made an additional DSP processor redundant and hence reduced the system cost [101]. Nowadays, the ARM cortex-M based processors have been extensively adopted by leading semiconductor vendors, such as Texas instrument, STM, NXP. According to a report in 2014, the shipment of Cortex-M units have reached 8 billion in about 10 years and this number is increasing quickly [102].

The Cortex-M family has several subfamilies: Cortex-M0, Cortex-M1, Cortex-M3 and Cortex-M4, etc. which are designed for satisfying various application requirements. The Cortex-M4 family is specially optimised for realising signal processing algorithms. Compared to its former Cortex-M3 family, the Cortex-M4 can achieve twice as fast for fixed-point algorithms and ten times faster for floating-point algorithms [101]. Such performance enhancement has enabled faster implementation of signal processing algorithms on low power consumption MCU, making the strategy of embedding signal processing on distributed wireless sensor node even more attractive.

One great benefit of Cortex-M processors is the Cortex microcontroller software interface standard (CMSIS) library, which is free of charge on all ARM processor-based MCU systems and has the potential of becoming an industry standard [101]. Furthermore, CMSIS-DSP library makes it even more attractive for embedded signal processing applications, which includes more than 60 common DSP algorithms, such as filter functions, Fourier transforms and vector operations. Thereafter, a state-of-the-art Cortex-M4 processor is employed as the core processor on the wireless sensor node.

3.2.1 Launchpad introduction

A TIVA LaunchPad board [103] (as shown in Figure 3.3) from TI is employed as the main baseboard for embedding signal processing algorithms. This board integrates a new generation 32-bit ARM Cortex-M4F microcontroller TM4C1233H6PM, which has the feature of both high performance and low power consumption while the price is just less than £3. The key features of the processor are listed below [104]:

- ARM Cortex-M4 with floating point with capabilities for digital signal control applications
- 256 kB Flash, 32 kB SRAM
- 32-channel configurable µDMA controller, providing a way to offload data transfer tasks from the Cortex™-M4 processor, allowing for more efficient use of the processor and the available bus bandwidth
- 12-bit ADC accuracy is achievable at the full 1-MSPS rating with no hardware averaging
- Generous communication options available: 8 UARTs, 6 I2C, 4 SPI / SSI

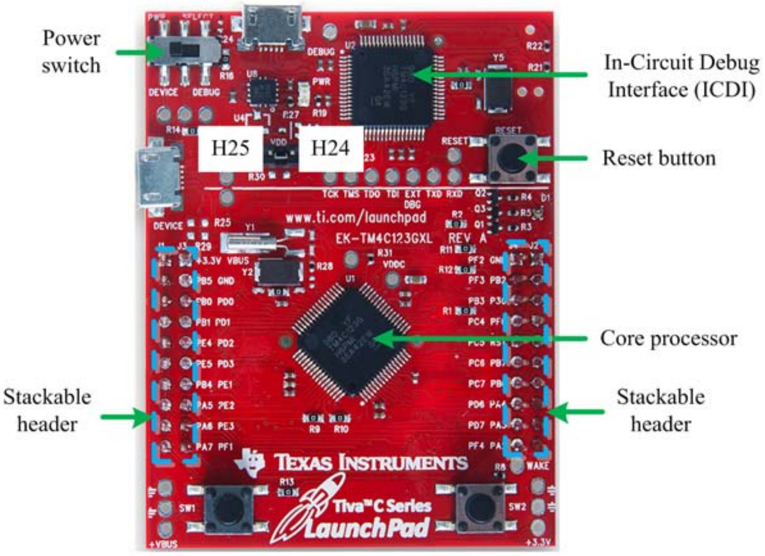


Figure 3.3 TIVA C LaunchPad development board

Another reason for choosing LaunchPad board is its excellent extensibility. Different function boards can be chosen and stacked on the LaunchPad board, or users can design their own booster pack boards. The booster pack boards available currently include colourful LCD display, Wi-Fi connection and Li battery, etc. [105] Besides, the onboard in-circuit debug interface (ICDI) makes the development process much more cost effective and the abundant software resources and libraries can greatly simplify and speed up the development process.

The main function of the core processor includes an analogue to digital conversion, implementing local signal processing algorithms and sending analysis results to the wireless module. Here, the on-chip 12 bit ADC is used to convert the vibration signals for low power and cost considerations. Although the resolution is a bit lower for vibration signals, it is competent enough for a demonstration. The reference voltage for the on-chip is the power supply of the processor (3.3V) and it can only accept positive voltage input. If the voltage input to the analogue pins is higher than the power supply or lower than 0V, the relevant pins might be damaged permanently. Therefore, the analogue signal should be properly biased before connecting to the ADC.

The floating point unit inside the processor is especially useful for implementing the signal processing algorithms efficiently. The µDMA controller can act as a co-processor to move data between registers and memories, in which way, the CPU unit could perform the algorithm much efficiently.

3.2.2 Power supply considerations

In the development process, the board can get its power supply simply from the USB port in the ICDI section or through its own USB port. The typical power consumption of the core processor is 370 µA/MHz. This means that it consumes about 30 mA when running at the maximum system frequency (80 MHz). This value varies depending on the resources used.

After the development process, the board is expected to be powered by a battery and the ICDI unit is not anticipated to work because its power consumption can significantly reduce the lifespan of the battery. To achieve this, the jumper on H24 and H25, as shown in Figure 3.3 should be removed and the 3.3V external power supply should connect to H24.

3.3 Vibration sensor conditioning booster pack

Vibration signals are usually collected through accelerometers, which can be either piezoelectric (PE) type or integrated electronics piezoelectric (IEPE). The PE type accelerometer outputs an electric charge signal proportional to vibration acceleration and needs no external power supply but an external charge amplifier is normally required to convert its high impedance output to a low impedance one. In this way, the signal can be effectively measured by an ADC. In comparison, the IEPE type accelerometer integrates a preamplifier for impedance conversion, which enables the output signal to be directly collected by the ADC or transmitted over long distances. However, the price is that it needs an excitation power from a constant-current source (4-20mA) [106].

Apart from the traditional PE and IEPE accelerometers, the micro-electro-mechanical systems (MEMS) type accelerometer is getting popular due to its low cost and low power consumption characteristics. The MEMS accelerometers have integrated conditioning circuit in a compact package or can output digitised acceleration directly. Currently, they have gained wide application for motion, shock or falling detections in the area of the automobile, smartphones and wearable devices, which does not require a very high sampling frequency. Also, they have been explored for condition monitoring. According to the investigation in [71], [107], the MEMS accelerometers under tests have similar frequency peaks content in comparison with the conventional accelerometer for the impact excitation but they also have substantial noise and shift in phase, meaning there still needs to be significant improvements before applying them in practice. Furthermore, the MEMS accelerometers usually have limited frequency response within 2 kHz. Meanwhile, the fault frequencies of mechanical systems are often modulated in several kHz ranges [14], which is out of the range of most MEMS accelerometers.

From the power consumption point of view, the PE type sensor consumes less power than the IEPE sensor and this is an especially important factor for a battery powered WSN node. Therefore, a PE type sensor is employed for vibration measurement.

3.3.1 Conditioning circuit

The schematic of a charge amplifier is presented in Figure 3.4. The output voltage V_{out} is determined by the input charge q and the feedback capacitor C_f:

$$V_{out} = \frac{q}{C_f} \tag{3.1}$$

It can be noticed that that the smaller the capacitor is, the larger the gain of the charge amplifier. The resistor R_2, R_4 and R_5 compose a T-type resister network, which is equivalent to a large feedback resister R_f:

$$R_f = R_2 + R_5 + \frac{R_2 R_5}{R_4} \tag{3.2}$$

The feedback resistor R_f is employed to create proper bias for the amplifier. The feedback resister R_f and capacitor C_f creates a high-pass filter [108], whose cut-off frequency f_{HP} can be given as:

$$f_{HP} = \frac{1}{2\pi R_f C_f} \tag{3.3}$$

It can be observed from (3.1) and (3.3) that the capacitance C_f needs to be lowered so as to increase the gain while the resistance R_f needs to be increased to keep the cut-off frequency low. However, very large resister yields poor SNR and is not easy to implement in practice [108]. This is the reason why a T-type resister network is used to produce the equivalent feedback resister in Figure 3.4. Because the impedance of the feedback resistor is so high, even a small bias current at the input of the amplifier might create a large output offset therefore an amplifier with low bias current is preferred [109]. In this design, a low bias current (only 1 pA at 25 ºC) rail-to-rail input/output amplifier LTC6078 [110] is chosen to build the charge amplifier. The parameters of the charge amplifier are computed and shown in Table 3.1.

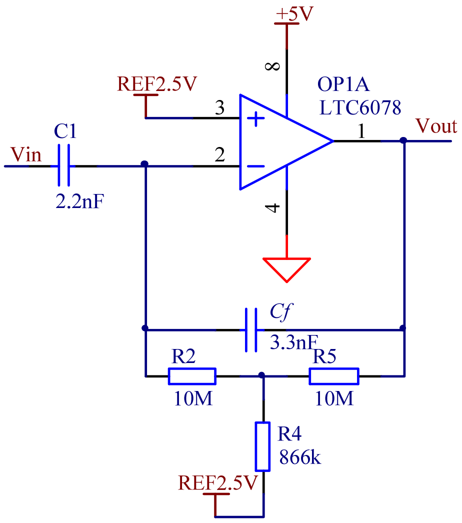


Figure 3.4 Schematic of charge amplifier and voltage amplifier

Table 3.1 Parameters of the charge amplifier

Parameter	**Value**
Gain	0.303mV/pC
Equivalent feedback resistor	135.5 MΩ
Cut-off frequency	0.356Hz

After the charge amplifier, the signal is amplified by 10 times and then filtered using a Sallen-key type filter shown in Figure 3.5 to suppress the high-frequency noises. The circuit is constructed using a low-power dual operational amplifier LTC2051 [111] and the desired parameters of the filter are given in Table 3.2. The frequency response of the designed circuit is given in Figure 3.6, which shows its -3dB gain is located at approximately 12 kHz.

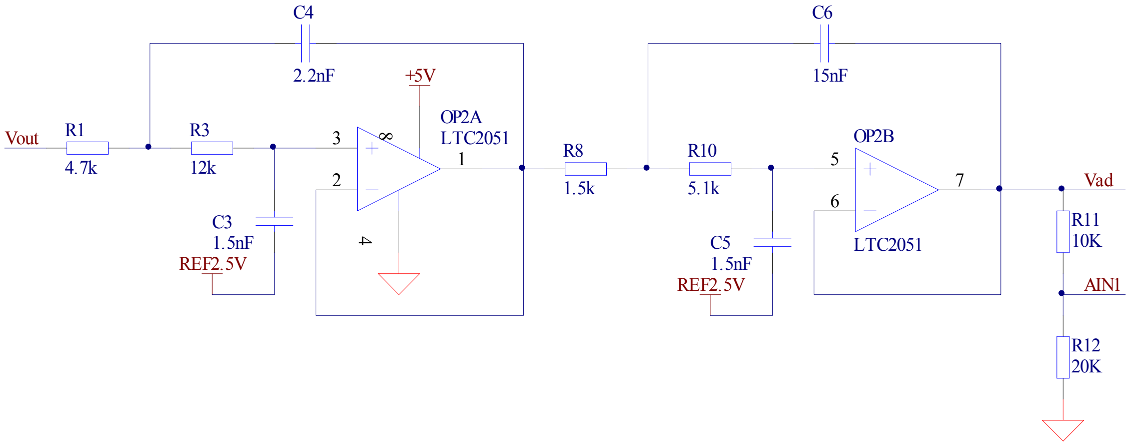


Figure 3.5 Schematic of 12 kHz Butterworth Sallen-Key low-pass filter

Table 3.2 Parameters of the 12 kHz low-pass filter

Parameter	**Value**
Order	4
Gain	0dB(1V/V)
Allowable passband ripple	1dB
Passband frequency	12 kHz
Corner frequency attenuation	-3dB

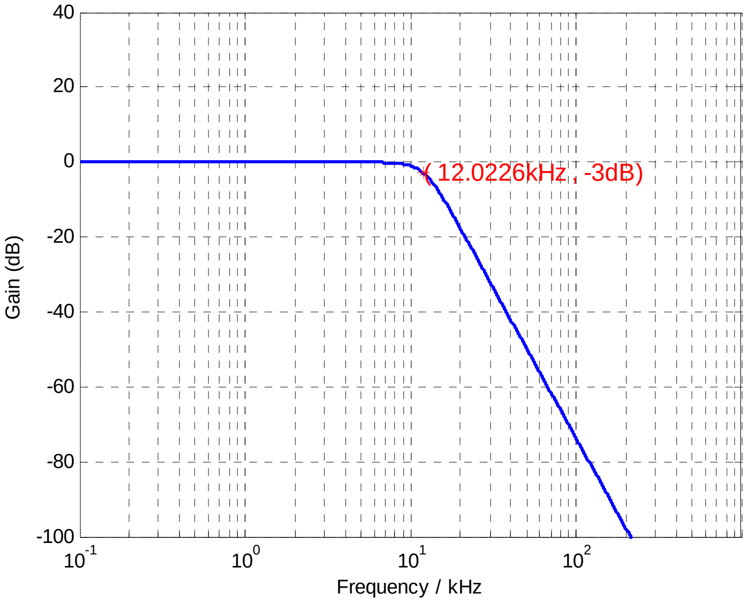


Figure 3.6 Frequency response of the designed low-pass filter

3.3.2 Power supply and reference voltage

As the wireless sensor node is powered by a battery, the power consumption always matters. In order to obtain a longer battery life, a simple solution is to shut off applications when they are not being used. The mechanism used to shut off the application is referred to load switch since the current drawn by each application is considered a load on the battery.

Typically, load switches are implemented using two metal–oxide–semiconductor field-effect transistor (MOSFET) [112], as shown in Figure 3.7. Here, a small N-channel MOSFET (Q1) is employed to control the state of the P-channel MOSFET (Q2) that has a low resistance in on state. When the logic input signal $V_{on/off}$ is high, both the N-channel MOSFET (Q1) and P-channel MOSFET (Q2) are turned on, allowing current to flow from port V_i to V_O. Otherwise, the current flow between V_i and V_O is blocked. This mechanism is quite effective in multiplexing battery power to the areas where it is needed and powering off when idle [112].

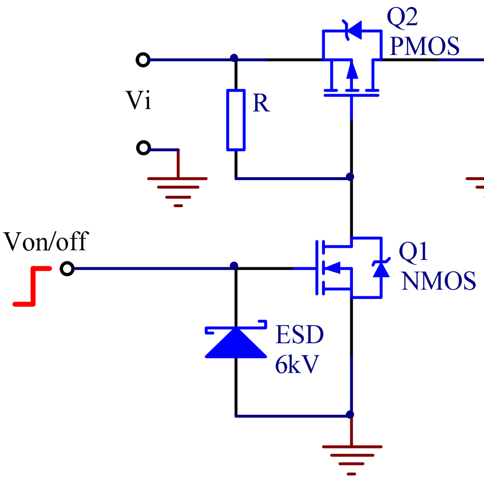


Figure 3.7 Basic load switch schematic [112]

With the quick development of portable devices, a host of integrated load switch chips become available. Not only have they saved the board space but they have also been optimised to obtain lower quiescent current, voltage drop, power dissipation, rising time and protection mechanism (current/temperature limits), etc. In this design, an integrated load switch TPS22929D [113] is employed, which allows a maximum current of 1.8 A. The schematic for load switch is shown in Figure 3.8(a). A logic high signal on *PE_ON* will enable the +5 V power output while it can be simply shut down by a low logic signal. By default, the +5 V power output is disabled by pulling the *PE_ON* pin down to the ground using a resistor.

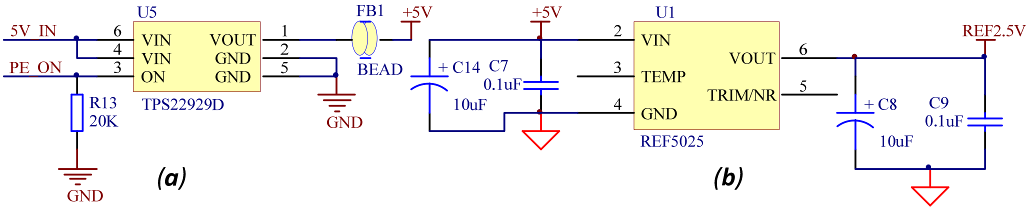


Figure 3.8 (a) Schematic of 5V power load switch and (b) schematic of 2.5 V reference voltage

As shown in Figure 3.4 and Figure 3.5, single power supply operational amplifier is employed for power consumption considerations and a +2.5 V reference voltage signal is needed to bias the operational amplifier properly. Here, a low noise precision voltage reference REF5025 [114] is adopted to provide the reference voltage. The schematic of 2.5 V reference voltage is shown in Figure 3.8(b).

3.3.3 External ADC and MicroSD card

By default, the vibration signal is collected by the on-chip 12 bit ADC of the core processor. In some cases, the signal can be very weak and the resolution of the ADC may be not sufficient, therefore, an external 16 bit ADC is also added on the conditioning booster pack as a backup solution and has not been used in this thesis.

The external ADC employed is LTC1864 [115], which is from Linear Technology. Its maximum sampling frequency can be as high as 250 KSPS and it can connect to the LaunchPad board through a serial peripheral interface (SPI) interface, as shown in Figure 3.9(a). The vibration signal can be connected to its input by placing a jumper on J1.

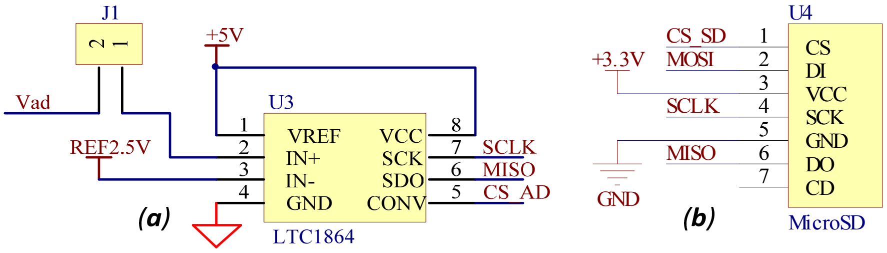


Figure 3.9 (a) Schematic of external ADC and (b) schematic of MicroSD card connection

In situations where the wireless signal is not stable, the data from the WSN node maybe not able to be uploaded in time, therefore, a MicroSD card is added for storing the data. The schematic for the MicroSD card connection is shown in Figure 3.9(b). Its connection to the LaunchPad is also through the SPI interface, which is the same as the external ADC but different chip selection signal is used to distinguish their data.

3.3.4 Port occupation analysis

Although the LaunchPad has provided good extensibility, careful attentions should be taken to avoid port usage conflicts. The port usage of the conditioning booster pack is shown in Table 3.3. As it shows, the analogue input is occupied on *J3_9* and the SPI ports are occupied as well with one selection pin for external ADC (*CS_AD*) and one for MicroSD card (*CS_SD*).

Table 3.3 Port occupation of conditioning booster pack

Function	GPIO	J1	J3	GPIO	Function
		1	1		
	PB5	2	2	GND	
	PB0	3	3	PD0	SCLK
	PB1	4	4	PD1	
	PE4	5	5	PD2	MISO
	PE5	6	6	PD3	MOSI
	PB4	7	7	PE1	
	PA5	8	8	PE2	
	PA6	9	9	PE3	AIN1
	PA7	10	10	PF1	

Function	GPIO	J4	J2	GPIO	Function
	PF2	1	1	GND	
	PF3	2	2	PB2	CS_SD
CS_AD	PB3	3	3	PE0	
	PC4	4	4	PF0	
	PC5	5	5		
PE_ON	PC6	6	6	PB7	
	PC7	7	7	PB6	
	PD6	8	8	PA4	
	PD7	9	9	PA3	
	PF4	10	10	PA2	

3.4 Wireless extension booster pack

3.4.1 Wireless module introduction

According to the investigation in Chapter 2, ZigBee is chosen to establish the prototype wireless condition monitoring mainly due to its low cost, extremely low power consumption and potential to create large-scale networks [23]. For fast prototyping, the off-the-shelf XBee ®ZB wireless module from Digi international [97] is employed in this design. This popular commercial module is effective, reliable and practical to establish the wireless network [116].

One advantage of XBee module is its pin-to-pin compatibility. It can be easily migrated to another kind of wireless protocol, such as WIFI and sub-GHz, without changing the hardware design. For the ZigBee network alone, there are also several options. Table 3.4 lists the specifications of Xbee ZB and Xbee-PRO ZB. In comparison with the

XBee ZB, XBee-Pro ZB has wider transmission range and higher power consumption. They can be chosen according to the application's range requirements.

Table 3.4 XBee ZB and XBee-PRO ZB specifications [117]

Platform	XBee® ZB	XBee-PRO® ZB
Performance		
RF Data Rate	250 kbps	
Indoor /Outdoor Range	40 m / 120m	90 m / 3200m
Transmit Power	2 mW (+3 dBm) boost mode	63 mW (+18 dBm)
Receiver Sensitivity (1% PER)	-96 dBm in boost mode	-102 dBm
Features		
Frequency Band	2.4 GHz	
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)	
Serial Data Rate	1200 bps - 1 Mbps	
ADC Inputs	(4) 10-bit ADC inputs	
Operating Temperature & Humidity	-40° C to +85° C, 0-95% humidity non-condensing	
Power Requirements		
Supply Voltage	2.1 - 3.6VDC	2.7 - 3.6VDC
Transmit Current	40 mA (@3.3 V)	205 mA
Receive Current	38 mA / 40 mA boost mode @ 3.3VDC	47 mA
Power-Down Current	<1 uA @ 25° C	3.5 uA @ 25° C

In addition, there are several options for the antennas, as shown in Figure 3.10. The Xbee modules with wire or printed circuit board (PCB) antennas are ready to use because the antenna is already on the board while an external antenna is needed for the Xbee modules with U.FL or RPSMA types since only the external connectors are provided on the board. However, the U.FL and RPSMA types provide more flexibility and are suitable for applications that want the antenna to be outside the box.

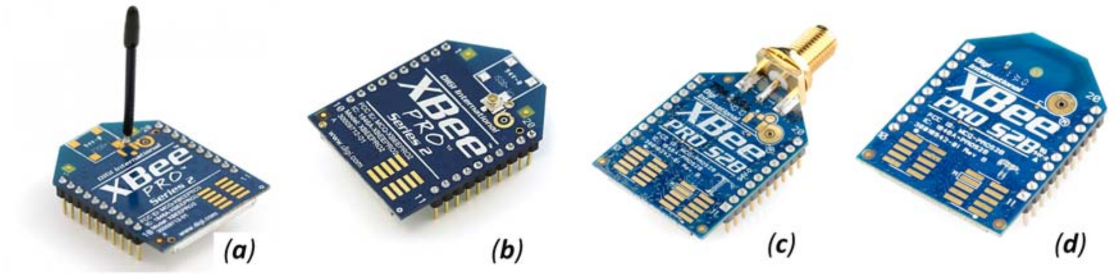


Figure 3.10 XBee Pro ZB module with different kinds of antennas: (a) wire antenna, (b) U.FL antenna, (c) RPSMA antenna and (d) PCB antenna

The WSN node is intended to be installed in a box, maybe a metal box for blocking potential large noises in the environment. Thus, the module with U.FL antenna is preferred. A U.FL female to RPSMA female connector is used for the connection with the 2.4GHz male SMA antenna, which is given in Figure 3.11(b) and (c), respectively.

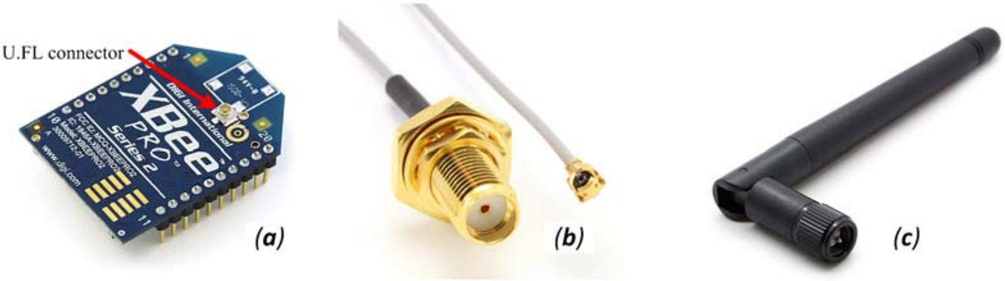


Figure 3.11 (a) XBee-PRO module with U.FL connector, (b) U.FL female to RP-SMA female connector and (c) 2.4GHz male SMA antenna

3.4.2 Connections with Launchpad board

Similar to the design in the power supply in Section 3.3.2, a load switch is added for the wireless module, which is also implemented using the load switch TPS22929D [113] as shown in Figure 3.12 (a). A logic high signal on *XBEE_ON* enables the +3.3V power output and it can be turned off by a low logic signal. By default, the +3.3V power output is enabled by pulling the XBEE_ON up using a resistor.

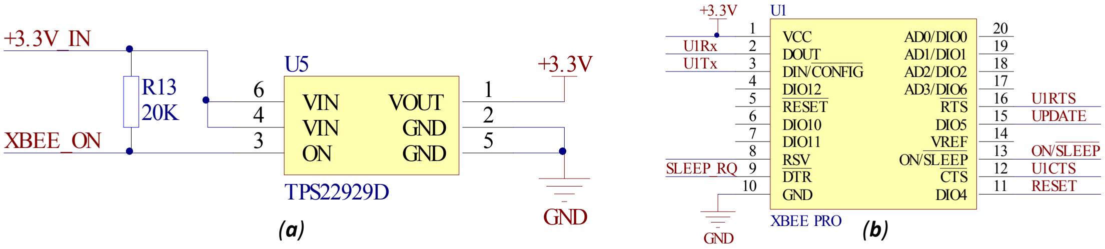


Figure 3.12 (a) Schematic for 3.3V power load switch and (b) schematic for Xbee module connections

The schematic for the Xbee module connections is shown in Figure 3.12 (b). Several signals are routed out for future usage. The *SLEEP_RQ* signal is used to request the wireless module to enter sleep mode and the $ON/\overline{SLEEP}$ signal is the indication for the working status of the module with low voltage output in sleep mode and high voltage in working state. The *RESET* signal is used to reset the processor on the Launchpad, which may be useful when the processor does not function correctly. The *UPDATE* signal is reserved for the GPIO check for the remote program updating.

In this application, because data throughput of the serial port is relatively high, clear to send (CTS) and request to send (RTS) flow control methods are employed to avoid overflowing the serial buffer on the wireless module and prevent the loss of data packets. RTS and CTS flow control signals are connected to the U1RTS and U1CTS on the Launchpad board and can be enabled by the D6 and D7 attention (AT) commands, separately [117].

3.4.3 Wireless module configuration

The wireless module can be configured using the XCTU software through the UART port. An XBee USB adapter is employed for carrying out the configuration, as shown in Figure 3.13(a) [118]. Once it is powered on, the board can be found in the device manager, as shown in Figure 3.13(b) and the port number (COM14 in this case) will be used in the XCTU software for recognising this board. By default, the Xbee modules are configured working at 9600/8-N-1 (9600bps, 8 data bits, no parity and 1 stop bit), which is used for the first time configuration of the modules.

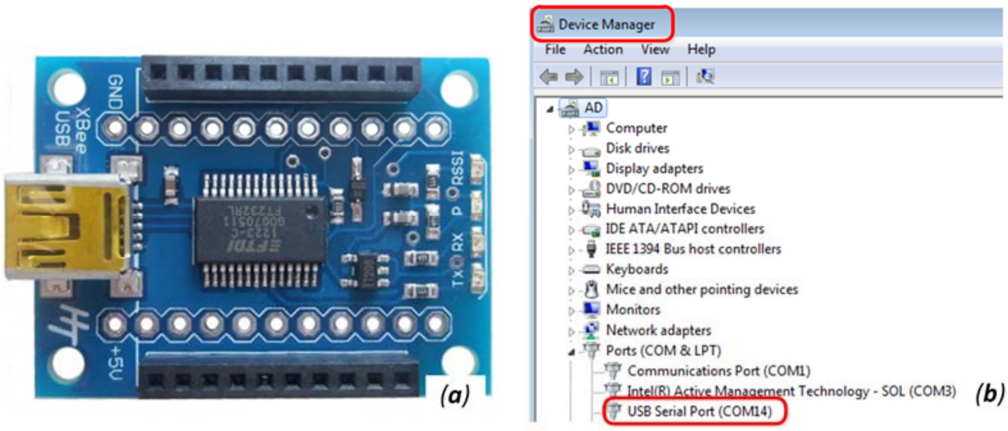


Figure 3.13 (a) XBee USB adapter and (b) view the port of the Xbee USB adapter

The Xbee module can be configured as three kinds of nodes: coordinator, router and end device and for each kind of node, there are two operating modes: attention (AT) and application programming interface (API). AT mode is simple and suitable for networks with just a few nodes while API provides the user more flexibility and specifies how commands, command responses, and module status messages are sent and received from the module [117]. In this application, the sink node is configured as a coordinator at API mode and the sensor nodes can be configured as router or end device at AT mode. The altered settings of this application are listed in Table 3.5. As it shows, the

RTS flow control is enabled manually and the CTS flow control is already enabled by default.

Table 3.5 Configurations of XBee module

Parameter	Value
ID – PAN ID	121
SC – Scan Channels	2000
NO- Node Discovery Options	2
BD – Baud Rate	115200 [7]
D6 – DIO6 Configuration	RTS flow control [1]
AO – API Output Mode	Explicit [1]

Apart from being configured via an adapter, the Xbee modules can also be configured through over the air programming (OAP). After the sink node is set as a coordinator, the other modules will join the ZigBee network automatically once they are powered on and can be found using the X-CTU tool for configuration. This programming mode has avoided the procedure of using the adapter to configure the Xbee modules and can be quite helpful when there are plenty sensor nodes.

3.4.4 Port occupation analysis

The port usage of the wireless extension booster pack is shown in Table 3.6. The UART1 port is occupied and cannot be used for other applications. The functions of *XBEE_ON*, *SLEEP_RQ*, *RESET* and *UPDATE* signals have been explained in Section 3.4.2. The *AIN1* and *AIN2* are just routed out from the relevant pins for future usages.

Table 3.6 Port occupation of wireless extension booster pack

Function	GPIO	J1	J3	GPIO	Function
3.3V		1	1	5.0V	
	PB5	2	2	GND	
U1Rx	PB0	3	3	PD0	
U1Tx	PB1	4	4	PD1	
	PE4	5	5	PD2	
SLEEP_RQ	PE5	6	6	PD3	
	PB4	7	7	PE1	XBEE_ON
	PA5	8	8	PE2	AIN2
	PA6	9	9	PE3	AIN1
	PA7	10	10	PF1	

Function	GPIO	J4	J2	GPIO	Function
	PF2	1	1	GND	
	PF3	2	2	PB2	
	PB3	3	3	PE0	
U1RTS	PC4	4	4	PF0	
U1CTS	PC5	5	5	RESET	
	PC6	6	6	PB7	
UPDATE	PC7	7	7	PB6	
	PD6	8	8	PA4	
	PD7	9	9	PA3	
	PF4	10	10	PA2	

3.5 Battery booster pack

3.5.1 Fuel tank booster pack introduction

An off-the-shelf battery booster pack for the Launchpad board is available from Element14, named as fuel tank booster pack [119]. As shown in Figure 3.14, the pack includes one board and a 3.7V, 1200 mAH Lithium-ion polymer battery. On the board, it includes a battery charger and a gas gauge with the capability of providing critical parameters of the battery, including temperature, charging state and capacity, etc. This information can be accessed through the inter-integrated circuit (I2C) communication port. The pack can be charged through the USB connector or through the connector near the USB socket.

As mentioned in Section 3.2.2, the ICDI unit is not expected to be powered by the battery unit. Hence, the two jumpers for 3.3V and 5V output near the USB connector in Figure 3.14 should be removed and these two outputs will connect to the vibration sensor conditioning booster pack directly.

Note that the two regulators on the battery booster pack work in the normal mode and consume about 14 mA even with no loads. It can be changed to work in the power save mode by moving the resistor on R18 and R20 to R17 and R19, separately. After this modification, the power consumption reduces to only $120\ \mu A$.

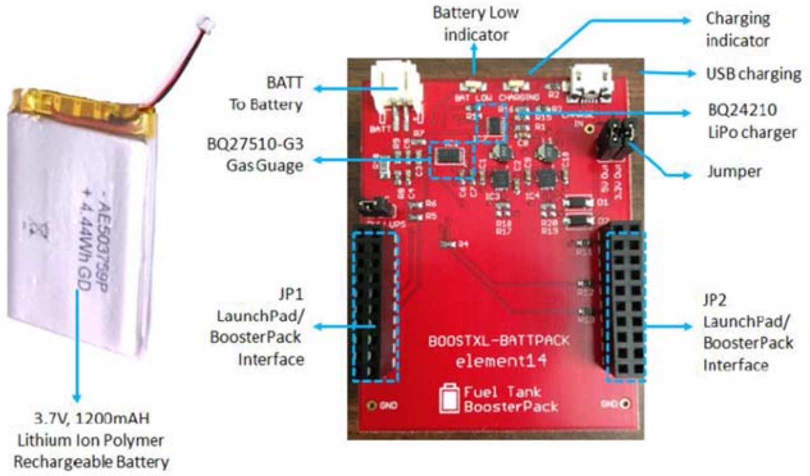


Figure 3.14 Picture of the battery booster pack

3.5.2 Port occupation analysis

The port usage of the battery booster pack is shown in Table 3.7. The I2C port is implemented using software simulation and is connected to two ordinary GPIO pins. $CHARGE$, $\overline{EN}$ and PG are three signals of the charge unit BQ24210. The CHARGE signal indicates the charge status with low voltage for charging and high impedance for other cases. The $\overline{EN}$ signal controls the chip with low voltage to enable the charger and high voltage to enter suspend mode. The PG signal indicates the power presentation with low voltage for normal range and high impedance for other cases.

Table 3.7 Port occupied by battery booster pack

Function	GPIO	J1	J3	GPIO	Function
3.3V		1	1	5.0V	
CHARGE	PB5	2	2	GND	
	PB0	3	3	PD0	
	PB1	4	4	PD1	
$\overline{\text{EN}}$	PE4	5	5	PD2	
	PE5	6	6	PD3	
PG	PB4	7	7	PE1	
	PA5	8	8	PE2	
	PA6	9	9	PE3	
	PA7	10	10	PF1	

Function	GPIO	J4	J2	GPIO	Function
	PF2	1	1	GND	
	PF3	2	2	PB2	
	PB3	3	3	PE0	
	PC4	4	4	PF0	
	PC5	5	5		
	PC6	6	6	PB7	SDA(SW)
	PC7	7	7	PB6	SCL(SW)
	PD6	8	8	PA4	
	PD7	9	9	PA3	
	PF4	10	10	PA2	

3.6 Summary

In this chapter, a wireless condition monitoring structure is proposed for the evaluation of distributed wireless CM based on vibration. A wireless sensor node is prototyped for the evaluation of embedding signal processing algorithms. A state-of-the-art Cortex-M4 microcontroller with good computing capability and low power consumption performance is employed as the core processor and an off-the-shelf wireless module is utilised to establish the ZigBee wireless network. The wireless sensor node is designed with a stackable structure with one main base board for data processing and three booster packs for vibration signal condition, wireless extension and battery management. In the design, low power consumption and future upgrading are also considered.

CHAPTER 4

INVESTIGATION OF SIGNAL PROCESSING TECHNIQUES FOR ROLLING BEARING CONDITION MONITORING

As discussed in Section 1.3.4, embedding signal processing algorithms on distributed wireless sensor nodes can effectively reduce data throughput over the bandwidth limited wireless network and also has the potential to decrease the power consumption of sensor nodes. Meanwhile, the low cost and low power consumption feature of the wireless sensor node does not allow complicated and memory-consuming signal processing algorithms to be implemented on it. Thus, a careful selection of suitable signal processing algorithms is required.

As a typical impulsive and modulating type signature, fault signal from rolling element bearing is selected for processing. This chapter starts with the introduction of rolling element bearing to study its fault generation mechanism and thus get an understanding of its fault features. Then, the current signal processing methods for vibration signal analysis in the CM area are discussed in three categories: time-domain, frequency-domain and time-frequency. In the corresponding section, the relevant signal processing algorithms are explained from the aspect of theory and applications for bearing signal processing. Furthermore, their advantages and disadvantages are also discussed with special attention to the feasibility for being implemented on a microcontroller with limited memory and computing capability.

4.1 Introduction to rolling element bearing

Rolling element bearings are one of the most widely used components in industrial rotating machines [14], [120]. Their applications can be widely found in various areas, such as automobiles, aeroplanes, computers and machine tools. Bearings play an important role in the linking and guiding of various rotating parts in industrial machines. These precision-made mechanical components are usually subjected to harsh running conditions, such as high speed, remarkable load and possibly severe vibrations. Although bearings are designed to be durable for a long lifespan, defects often develops under such severe operating conditions [121]. Actually, the failure of bearings is responsible for the most frequent reasons of machine breakdown [14]. According to a survey of faults on induction motors conducted in 1995, bearing failure accounts for the majority of the failures, as much as 51% [122].

4.1.1 Bearing faults and its vibration characteristics

A rolling element bearing generally consists of four parts (as shown in Figure 4.1): an inner race, an outer race, rolling elements and a cage which holds the rolling elements in certain relative positions. Some bearings have additional components, for example, spherical roller bearings have guide race and seals.

Typically, the inner race rotates with the shaft and the outer race is mounted onto a stationary bearing house. The rolling elements can be a ball or roller type. In comparison, the load in the ball type bearing is transferred to a very small surface on the raceways while that in the roller type one is through a line contact. This indicates that roller type bearing can carry a larger load than the ball type bearing of similar dimensions. The rolling elements are separated by the cage to prevent their contacts during operation. Furthermore, the cage also helps to prevent poor lubrication conditions and hold the bearing together during handling [15].

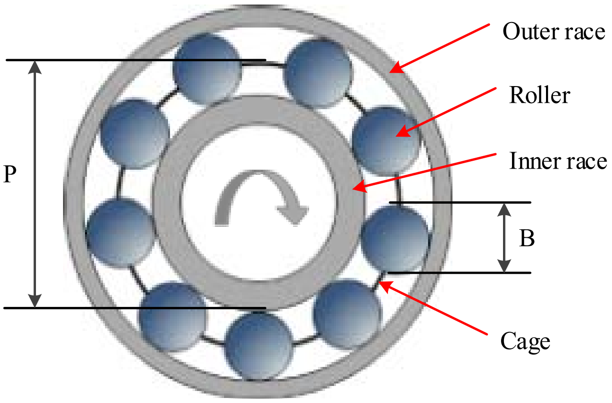


Figure 4.1 Bearing components

In general, defects of bearings can be categorised as distributed or localised. Distributed defects include surface roughness, waviness, misaligned races and off-size rolling elements and are usually caused by manufacturing error, improper installation or abrasive wear defects [123]. Localised defects include cracks, pit and spall on the rolling surfaces. The major failure mode of rolling element bearings is spalling of the races or the rolling elements. This is typically caused by an initial fatigue crack below the surface of the metal, which then propagates towards the surface until a piece of metal breaks away to leave a small pit or spall [123].

Consider an example where the outer race of the bearing has a spall. Each time the spall is rolled over; a pulse of very short duration is incurred that causes the bearing to vibrate at its resonance frequency. As illustrated in Figure 4.2 (T denotes the time interval between impacts), the response decays quickly due to damping [15].

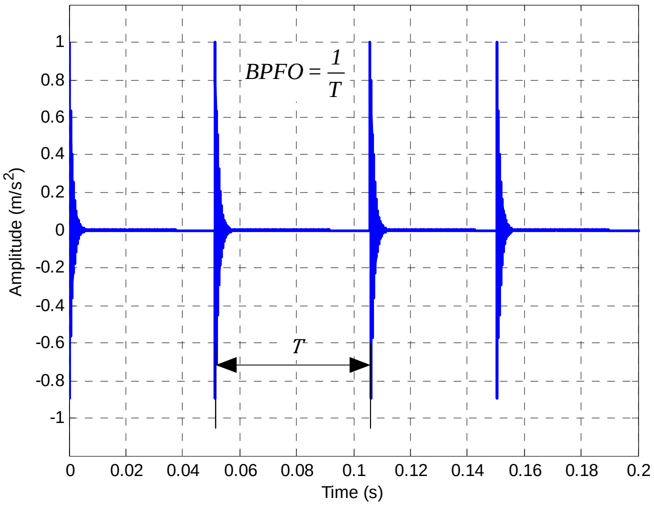


Figure 4.2 Idealised vibration signature due to fault in outer race

When the bearing is rotating at a steady speed, a periodical vibrating response can be captured by an accelerometer mounted on the bearing house. The frequency of this periodical response is named as the fault characteristic frequency, which is determined by the faulty component, geometric dimensions, and the rotational speed. The fault frequencies are specific for defects existing on different components including the outer race, inner race, balls, or cage. It is the fault characteristic frequency that is of interest in the detection of bearing faults, rather than the large amplitude responses at the high-frequency resonance frequency of the bearing rings induced by the short duration impacts. For a bearing with its outer race fixed, its theoretical characteristic fault frequencies can be calculated using Eq. (4.1) - (4.4), and a derivation of these equations is given in [15].

Table 4.1 Equation for local bearing fault characteristic frequencies

Bearing fault frequencies	Definition	No.
Ball pass frequency, outer race (BPFO)	$\mathrm{BPFO} = \frac{Nf_r}{2}(1 - \frac{d}{D}\cos\phi)$	(4.1)
Ball pass frequency, inner race (BPFI)	$\mathrm{BPFI} = \frac{Nf_r}{2}(1 + \frac{d}{D}\cos\phi)$	(4.2)
Ball (roller) spin frequency (BSF)	$\mathrm{BSF} = \frac{Df_r}{2d}(1 - \frac{d^2}{D^2}\cos^2\phi)$	(4.3)
Fundamental train frequency (FTF)	$\mathrm{FTF} = \frac{f_r}{2}(1 - \frac{d}{D}\cos\phi)$	(4.4)

where d is the ball diameter; D is the pitch diameter; N is the number of balls; ϕ is the contact angle and f_r is the shaft rotation rate in Hertz (Hz). These equations are theoretical and discrepancies arise when bearings carry significant thrust loads or if there is any slippage. Typically, the slip causes change in bearing frequencies in the order of 1-2%, both as a deviation from the calculated value and also as a random variation around the mean frequency [14].

4.1.2 Bearing test rig description

To evaluate the performance of the wireless condition monitoring system, a bearing test rig was set up. As shown in Figure 4.3 (a), the rig is composed of five main parts: an electrical induction motor, shaft couplings, a DC generator, bearings and motion shaft.

The bearing in the test rig is N406 cylindrical roller type. During the test, the shaft ran at a full speed of 1460 rpm, i.e. 24.3 Hz. The four localised fault frequencies for the bearing are calculated according to Eq. (4.1) - (4.4) and listed in Table 4.2. Among the four fault frequencies, the highest one is the inner race fault frequency at 135.5 Hz, whose 3rd harmonic frequency (406.5 Hz) is within 500 Hz. This means fault frequencies of interests are all within 500 Hz, which is a significant basis for conducting the down-sampling process in the algorithm proposed in Chapter 6.

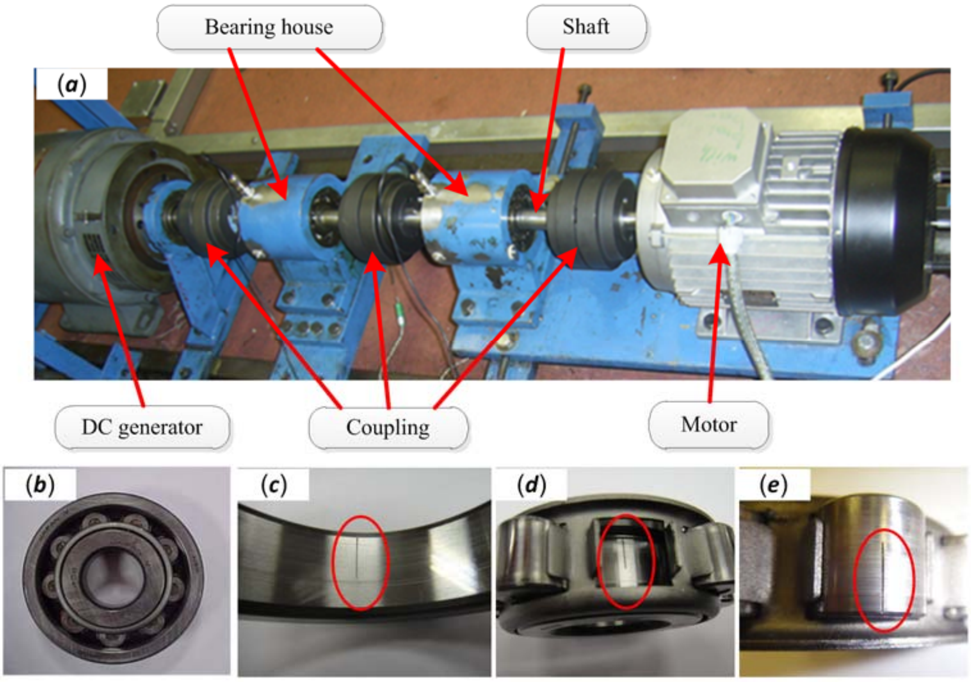


Figure 4.3 Bearing test rig and bearings for experiments: (a) bearing test rig, (b) healthy bearing, (c) bearing with outer race defect, (d) bearing with inner race defect and (e) bearing with roller defect

Table 4.2 Bearing fault characteristic frequencies

Defect location	Fault frequency (Hz)
Inner race (BPFI)	**135.5**
Outer race (BPFO)	**83.5**
Ball (BSF)	**48.4**
Cage (FTF)	**9.3**

Four bearing conditions as shown in Figure 4.3 (b-d) were tested on the rig. The healthy bearing was employed for producing the baseline signal and the other three bearings were seeded with outer race fault, inner race fault and roller fault. A piezoelectic (PE) type accelerometer was mounted on the bearing house horizontally to collect the vibration signal. A group of vibration signals from four bearing conditions were acquired and these signals were employed as the study target to help understand the various signal processing techniques in the following sections.

One segment of the vibration signals is presented in Figure 4.4, where it includes 8192 points of data sampled at 32 kHz. It can be seen that signals from a defective bearing show obvious dissimilar features from the healthy one. The vibration levels from the faulty bearing are higher than those from the healthy one and periodical impacts can be clearly observed in the waveform of faulty bearings.

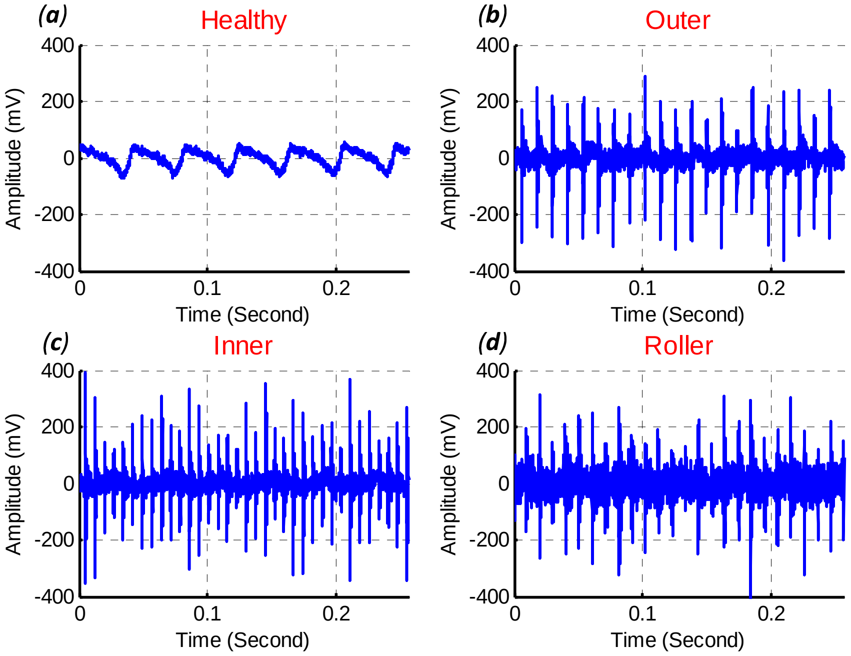


Figure 4.4 Typical vibration signal from different bearing condition: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

4.2 Time-domain analysis

One simple and straightforward approach for bearing fault detection and diagnosis is to analyse the measured vibration signal in the time domain. The method can be as simple as visually observing the vibration signal or further processing can be used to extract features. A number of statistical parameters have been developed to characterise various signal characteristics, such as RMS, peak, crest factor, skewness, and kurtosis. A new processing method, named morphological filter, has recently been applied for extracting envelope from bearing vibration signals [124] and shows a promising potential for local processing on a microcontroller due to its simple and efficient computing capability.

4.2.1 Statistical parameters

The statistical parameters are helpful for characterising specific signal characteristics and have been used widely for expressing bearing fault features. For one discrete signal $x(n)$, its r^{th} order moment can be written as:

$$M_r = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^r \tag{4.5}$$

where N is the number of data points and r is the order of the moment. The other common statistical parameters for the discrete signal x_n are defined in Table 4.3.

Table 4.3 Definition of common statistical parameters

Statistical parameters	Definition	No.
RMS	$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2}$	(4.6)
Peak	$peak = \frac{1}{2}[\max(x_n) - \min(x_n)]$	(4.7)
Stand deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2}$	(4.8)
Skewness	$skew = \frac{M_3}{\sigma^3}$	(4.9)
Crest factor	$cf = \frac{peak}{RMS}$	(4.10)
Kurtosis	$kurt = \frac{M_4}{\sigma^4}$	(4.11)

Among these statistical parameters, root mean square (RMS), crest factor, skewness, and kurtosis are most commonly employed for bearing signal analysis [120]. Therefore, these three parameters of are calculated for 39 segments of bearing signals shown in Figure 4.4 and presented in Figure 4.5. The statistical parameters can be calculated in two approaches: one is to compute the statistical parameters for the whole frequency range of the signal and the second one is to break the signal into discrete frequency bands and compute the parameters for each band [120]. Here, the first method is employed for calculation.

From Figure 4.5, it can be observed that all these three parameters can well separate the signal from faulty bearings with that from the healthy one. In addition, the statistical parameters for different signal segments are very close, which can be explained by the smooth running condition of the bearing test rig.

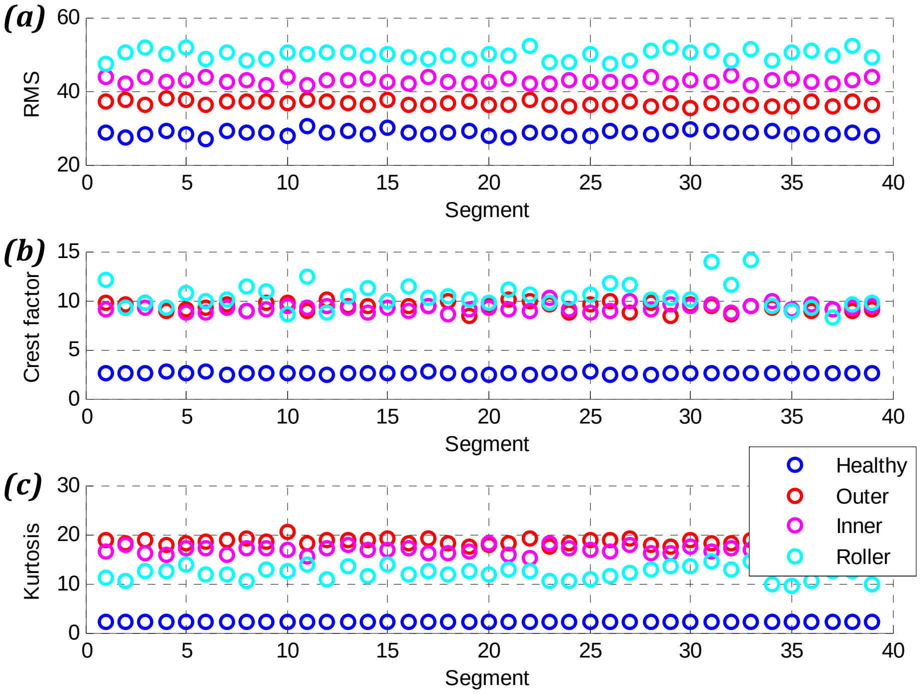


Figure 4.5 Statistical parameters for different signal segments: (a) RMS, (b) crest factor and (c) kurtosis

➢ RMS

Root mean square (RMS) is the simplest and most common approach for measuring defects. It provides a measure of power contents of the vibration signal. This feature is helpful for tracking the overall energy level, but it provides no useful information on which component is failing [125]. Besides, it may not show appreciable changes in the early stages of bearing damage [125].

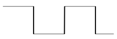
➢ Crest factor

The crest factor is a good measure for detecting early stages of damage. It is defined as the ratio of the peak over the RMS level. This means that for two signals with the same RMS value, the one with larger peaks will have a higher crest factor value. According to [125], the crest factor normally lies between 2 and 6 and a crest value higher than 6 may indicate an abnormal machinery condition. This feature is effective for detecting changes in an impulsive type vibration source such as a localised fault on a gear or a bearing.

- **Kurtosis**

Kurtosis is the fourth moment of the distribution. It can characterise the pulse character of a signal by reflecting the relative peakness of a distribution over a normal distribution. One benefit of Kurtosis is that its value is determined not by the signal amplitude but by its shape factor. Table 4.4 shows kurtosis values for four types of typical signals [126]. In practice, Kurtosis is often calculated after the signal is filtered. Its value can be higher than 100, especially when it is computed inside the band of the structure resonance [127].

Table 4.4 Kurtosis value for typical signals

Signal type	**Kurtosis value**
Sine signal	1.5
Square signal 	1
Gaussian signal	3
Pulse signal	>3

Besides, some other statistics are also used for characterising bearing signals, such as clearance factor, impulse factor, shape factor and beta monuments based upon the beta distribution. According to [120], the overall performance of beta function parameters to identify and classify bearing defects is almost identical to that of crest factor and kurtosis.

Although statistical parameters are easy to calculate and can indicate bearing conditions, most of these monitoring parameters are sensitive to noise and operating conditions [128]. Their limitations are summarised as follows:

- Incipient damage could not be detected in some cases
- Damage could not be detected in early stages
- Value of parameters reduces to normal levels when fault develops
- Difficulty in the diagnosis of localised fault types

Furthermore, by calculating statistical features in a short-time, the resultant signal can give the envelope related information from the bearing vibration signal and its effectiveness has been studied and verified by Kilundu and Behzad [129], [130]. This will be further studied in Chapter 8.

4.2.2 Morphological filter

The morphological filter is firstly developed to analyse binary image data, later extended to grey-level images and has become a common image processing method [131]. In 1987, Maragos and Schafer first presented the definition of morphological filter for one-dimensional time series data [132]. Now, it has been applied in several areas, such as surface profiling [133], medical electrocardiogram (ECG) signal processing [134] and also vibration signal processing [124]. The application of a morphological filter for impulsive signal processing is mainly to suppress the noise and extract the envelope in the measured signal. Nikolaou and Antoniadis [124] applied it on a bearing fault signal and extracted its envelope which offers a better visual inspection of the impulsive signal. Through a corresponding power spectra analysis, it shows the extracted envelope contains sufficient information about the repetition period and the intensity of the impacts.

Suppose there is a structural element $g(m)$ of length M and $g(m)$ is an even function, i.e. $g(m) = g(-m)$, for a discrete signal $x(n)$ of length N, the four basic operations of morphological filter are defined in Table 4.5 [124].

Table 4.5 Definition of four basic operations of morphological filter

Operator	Definition	No.
Erosion	$(f \ominus g)(n) = min\{f(n+m) - g(m)\}\ (n = 0, 1, \cdots, N - M - 1)$	(4.12)
Dilation	$(f \oplus g)(n) = max\{f(n-m) + g(m)\}\ \ (n = 0, 1, \cdots, N + M - 2)$	(4.13)
Opening	$(f \circ g)(t) = [(f \ominus g) \oplus g](t)$	(4.14)
Closing	$(f \bullet g)(n) = [(f \oplus g) \ominus g](t)$	(4.15)

Basically, the erosion operation reduces the peaks and enlarges the minima of $f(n)$, while the dilation operation increases the valleys and enlarges the maxima of $f(n)$. The closing operation can detect the upper envelope of $f(n)$ while opening operation can detect its lower envelope. By combining the opening and closing operator properly, an averaged bi-directional envelope can also be obtained [135].

The structure element acts as an important role in the morphological filter and a small change in it can cause a very different detection result. The basic principle to design structural elements is to select the structural element which is similar to the geometric characteristic of the target signal [136]. The choice of structural elements includes the shape and size. In [124], Nikolaou and Antoniadis recommended the length of a structure element to be 0.6–0.7 times of the pulse repetition period, which can minimise the noise effects.

Take the same data set in Figure 4.4 for example; the upper envelope of the vibration signals are calculated using closing operator and presented in Figure 4.6. In order to keep a periodical pulse of period T, the length of the structure element should be less than T. Thus, to keep all these three bearing fault signals, the structure element should be smaller than the period of the maximum interested fault frequency. In this case, the maximum fault frequency is the inner race fault with 135 Hz, corresponding to a period of 7.4 ms. Thereafter, a flat structure element of length 95, corresponding to 3 ms, is applied on the four signals. As shown in Figure 4.6, the envelope of the signals is well tracked and the extracted envelope provides a clearer visual display than the raw data form.

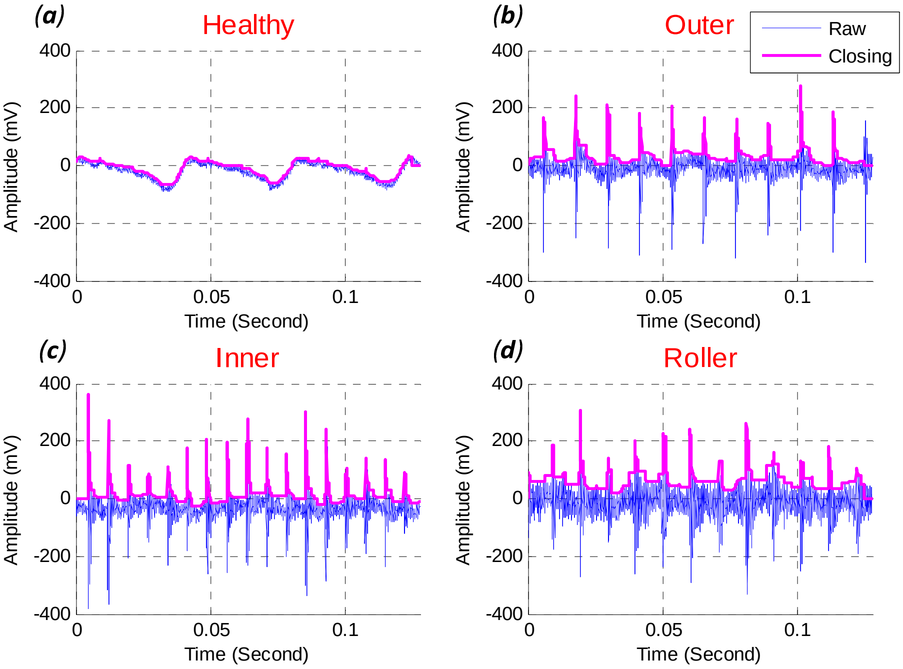


Figure 4.6 Closing envelope detection result for vibration signal from different bearing condition: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

The main advantage of the morphological filter is that its computation only involves addition and subtraction, which makes it very fast and thus especially suitable for a processor with restricted computation capability and memory size. In addition, morphological can be combined together with other signal processing algorithms to obtain better processing results. For example, Sui and Zhang combined wavelet transform and morphological filters to extract features of defects in bearings [137].

The detected envelope by the morphological filter can usually be down-sampled to produce a small data for transmission. A better way is to use it as part of the envelope analysis and get a more meaningful result. However, the disadvantage is that there are no available libraries found for its implementation on embedded processors. Meanwhile, the Hilbert transform based envelope analysis can well utilize the finite impulse response (FIR) or fast Fourier transform (FFT) library functions optimized for embedded processors.

4.2.3 Other time-domain techniques

One way to express the amplitude characteristics of a vibration signal is using the probability density distribution [120], which can be estimated by determining the time duration for which a signal remains in a set of amplitude windows. For a typical window at amplitude x and of width Δx, the probability $P(x)$ is defined as:

$$P(x \leq x(t) \leq x + \Delta x) = \sum_{i=1}^{N} \frac{\Delta t_i}{T} \tag{4.16}$$

To illustrate the typical changes which can occur, Figure 4.7 shows the normalised probability density function for the vibration signals in Figure 4.4. Note that the raw signal is filtered with a 100 Hz high-pass filter to eliminate the 50 Hz power interfaces. Large differences can be clearly observed in the probability density function from the defect bearings. Firstly, the probability density functions from healthy bearings are much sharper than that from defect ones. Secondly, the signals from defect bearings have a greater span of higher amplitude as shown by the increase in the tails of the probability function.

The probability density function is easy to implement and can result in a smaller data set than that of the original raw data. It can be employed for abnormality detection but has no function for distinguishing the fault components.

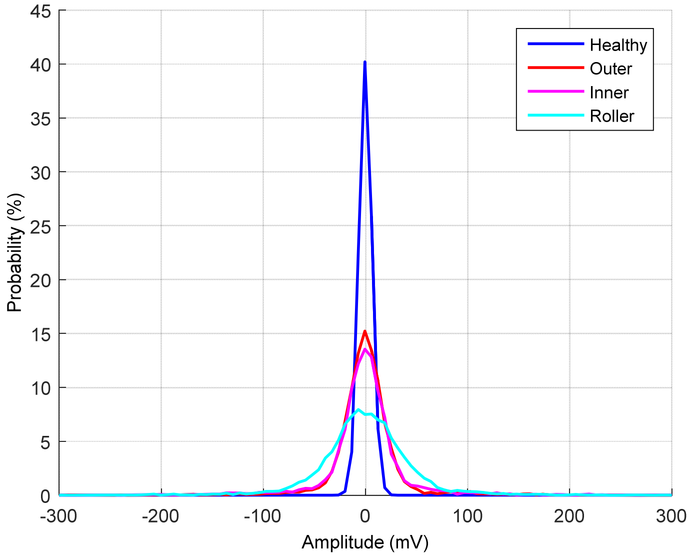


Figure 4.7 Normalised probability density function vibration signal from different bearing conditions

Time synchronous average (TSA) is another promising signal analysis technique in the time domain [138]. Its idea is to "use the ensemble average of the raw signal over a number of revolutions in an attempt to remove or reduce noise and effects from other sources, so as to enhance the signal components of interest" [139]. Since the rotating process of the bearing can be considered as cyclostationary, this method is especially helpful for removing random noises. The TSA $\bar{x}(t)$ for a signal $x(t)$ can be expressed by:

$$\bar{x}(t) = \frac{1}{N}\sum_{n=0}^{N-1} x(t + nT)\,, \qquad 0 \leq t < T \tag{4.17}$$

where T is the averaging period and N is the number of samples for averaging.

For the implementation of TSA, it needs a synchronisation signal usually obtained from an optical encoder to precisely align the signal of different periods. Because the machine running speed is not absolutely stable, the data points for each period might be not exactly the same, which also needs to be taken into consideration during implementation.

4.3 Frequency-domain analysis

The frequency-domain analysis, by its name, analyses spectral information of the signal by transforming a signal to the frequency domain. Its advantage over time-domain analysis is that it can easily identify and isolate certain frequency components of interest [140]. A complicated signal in the time-domain might become simple in the frequency-domain. As vibration signal from rotating components like bearing can be considered cyclostationary [14], this makes frequency-domain analysis a fundamental approach for bearing fault detection. The frequency-domain analysis includes common spectral analysis [141], envelope analysis [14] and cepstrum [142], etc.

4.3.1 Spectral analysis

The idea of spectral analysis is to either observe the whole spectrum or only certain frequency components of interest to extract fault features from the signal [143], [144]. The spectral information $X(m)$ of a discrete time series data $x(n)$ is usually computed through a discrete Fourier transform (DFT):

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N} \tag{4.18}$$

where e is the base of natural logarithm and $j = \sqrt{-1}$.

The most commonly used tool in spectral analysis is power spectrum and can be estimated by:

$$P = E[X(m) * X^{*}(\mathrm{m})] \tag{4.19}$$

where $X^{*}(m)$ represents the complex conjugate of $X(m)$ and E denotes expectation.

Although DFT is the most straightforward mathematical procedure for determining the frequency contents of a time-domain sequence, its poor efficiency has prevented its wide application for a long time until the fast Fourier transform (FFT) is developed, which provides a much faster approach for computing DFT and reduces its computational complexity from the order of N^2 to only $Nlog_2N$. Thereafter, the spectral analysis has achieved wide application for processing various signals. With the fast development of electronics technology, FFT can easily be implemented efficiently on a microcontroller, like the microcontroller TM4C1233H6PM employed in this thesis.

The spectra of the vibration signals in Figure 4.8 are computed and illustrated in Figure 4.8. It can be seen that the vibration signals from healthy bearing contain small high-frequency components while several high-frequency haystacks can be clearly viewed on the spectrum of defect bearings. The resonances mainly appear around three frequencies: 4.5 kHz, 8 kHz, and 10 kHz. The frequency band from 4 kHz to 5 kHz are magnified and illustrated as an inset. From the magnified frequency band, periodical frequency peaks can be observed. These are the harmonics of fault characteristic frequency situated by the resonances. However, it is not easy to identify the fault frequencies from the magnified spectra due to both random fluctuations [49] and noise interferences.

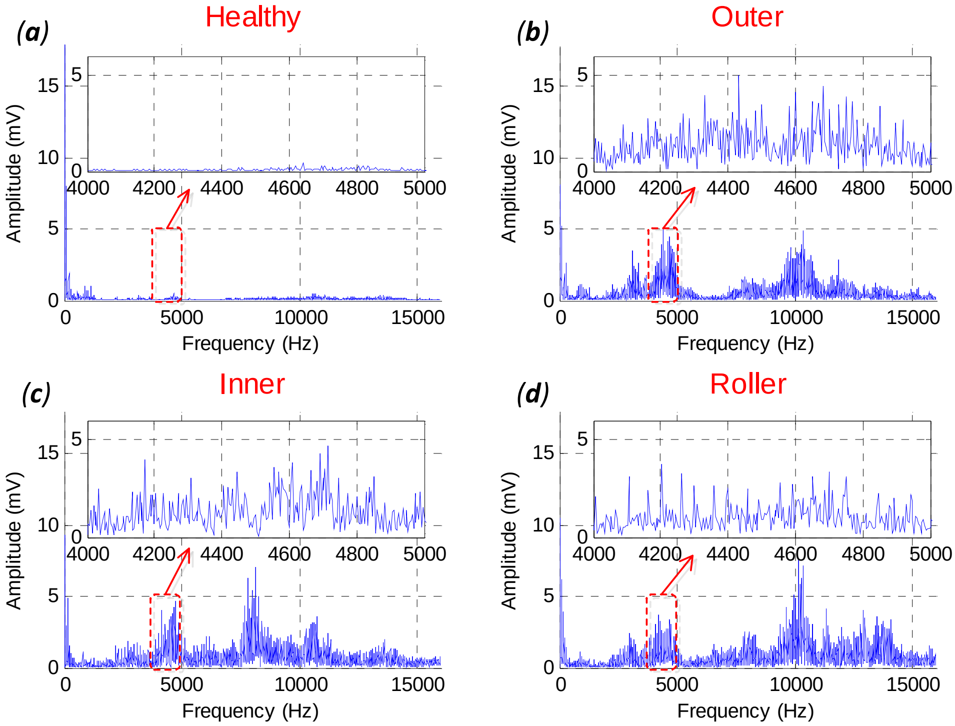


Figure 4.8 Spectra for vibration signal from different bearing conditions with magnified 4 kHz-5 kHz band shown as an inset: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

Comparing current spectrum with the baseline one can help check the fault severities. In [141], it is suggested that an increase of 6-8 dB can be seen as significant and a change over 20 dB is considered as severe. One difficulty with the spectral comparison is that the rotating speeds are not absolutely stable and usually have small fluctuations, which can affect the results of spectral analysis. This problem can be overcome by changing the linear frequency axis to a logarithmic one with constant percentage bandwidth [120].

One purpose of computing spectrum of vibration data is to trend the interested frequency components changes and some spectral parameters has been defined for trending, for example the matched filter root mean square (Mfrms) [120], defined as:

$$Mfrms = 10log\left\{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{A_i}{A_i(ref)}\right)^2\right\} \tag{4.20}$$

where $A_i(ref)$ is the amplitude of the i^{th} spectral line in the reference spectrum and A_i is the amplitude of the i^{th} spectral line in the current spectrum.

Another useful avenue of frequency domain analysis for bearing fault detection and diagnosis is to employ the so-called waterfall of spectra. By observing the frequency components changes along time, abnormal frequency components can be observed in both the low-frequency range and high-frequency range.

Spectral analysis is a fundamental method for bearing fault diagnosis. Because of the cyclostationary feature of bearing signals, the periodical characteristics due to rotating process can be easily observed in the spectrum while it might be not so obvious in the time-domain. However, it surely has several limitations as listed below:

- It only provides the global energy-frequency distributions and fails to reflect the details of a signal, making it hard to analyse a signal when the fault signal is weaker than the background noises [137]. On the occasion that bearing vibration signals are buried in noises or vibrations from other sources, it becomes more difficult to extract useful information from the spectrum.
- It is difficult to identify inner race and roller defects from spectrum because of the presence of additional transfer segments.
- Small random slip often exists in rolling bearing and causes a fundamental change in the character of the signal. According to [14], it is often difficult to extract diagnostic information in the raw spectrum because the higher harmonics smear over one another with even a small amount of slip (0.75%). The spectrum may vary

a lot for bearings located on different machines. So the analysis result is highly dependent on the experience of the end user.

4.3.2 Envelope analysis

Envelope analysis is the FFT frequency spectrum of the envelope of a signal. It is suitable for diagnostics of machinery where faults have periodic excitation or amplitude modulating (AM) effect of the structural resonance [15]. Over many years, it has been recognised as the benchmark method for bearing diagnostics [14], [120].

The procedures of envelope analysis are shown in Figure 4.9, which is composed of three steps, a band-pass filter with the pass band at the fault impulses are amplified by structural resonances, envelope detection and power spectrum calculation. The band-pass filter is applied to reject the low-frequency with high-amplitude signals caused by imbalance or misalignment and to eliminate random noises outside the pass-band for the enhancement of the signal to noise ratio (SNR) [15]. The envelope detection extracts the modulating fault signal and transfers the high-frequency problem to a low-frequency one. Several envelope detections are available, like the square rectifier [49], morphological filter [124], short-time statistical features [129], and the most priciest Hilbert transform [145].

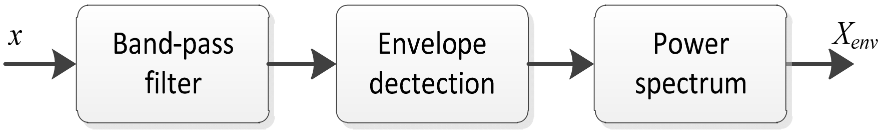


Figure 4.9 Procedures of envelope analysis

The envelope and envelope spectra for the vibration signals in Figure 4.4 are calculated according to the steps in Figure 4.9 and presented in Figure 4.10. Here, a filter with the pass band between 4 kHz and 6 kHz is applied to the raw signal and then the envelope is calculated using the Hilbert transform method which will be explained in Section 5.2. In the envelope signal, the impulses are extracted out from the high-frequency resonances. By calculating its spectrum, the harmonics of the fault frequencies can be clearly observed, which verify the existence of the defect on the corresponding bearings. In addition, it can be observed that the meaningful envelope spectrum that represents bearing fault features only occupy a very small portion of the entire spectrum, which indicates the resultant fault feature set can be very small.

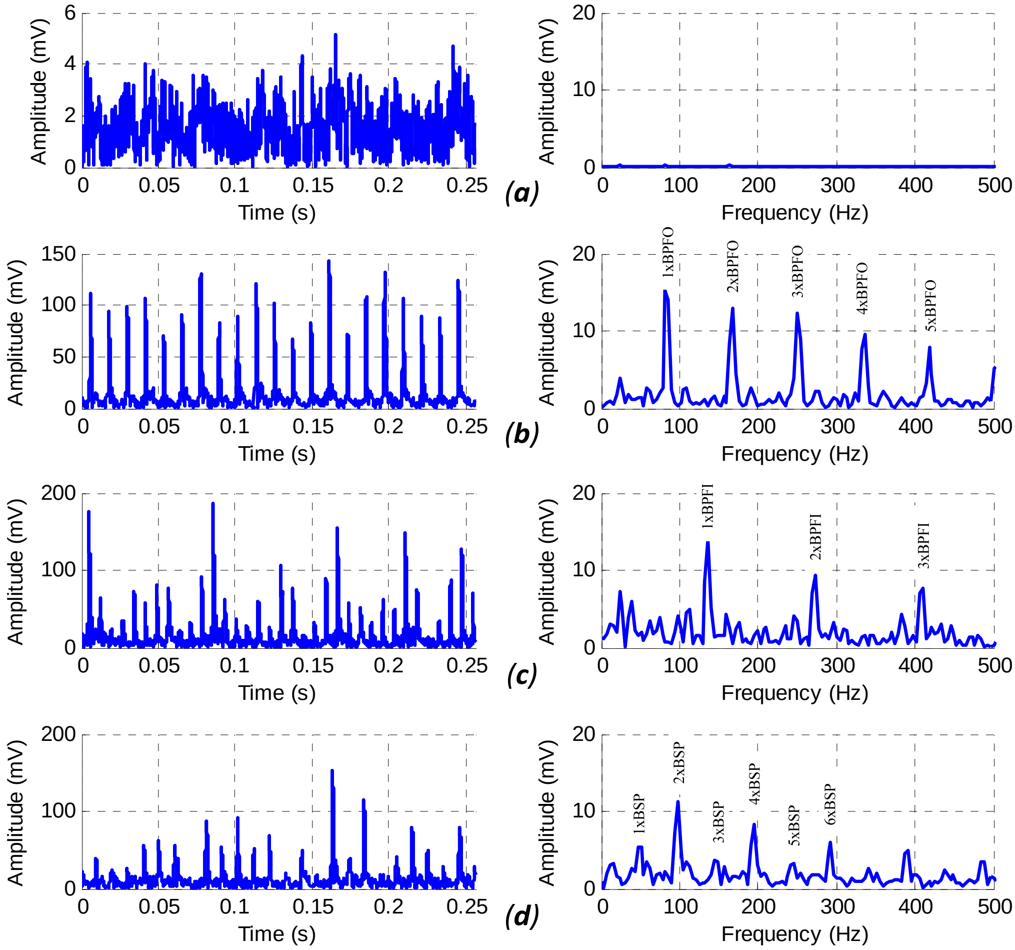


Figure 4.10 Envelope and envelope spectra for vibration signal from different bearing conditions: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

The advantages of envelope analysis are listed below:

- Clear representation of localised bearing faults
- Capability to represent fault features in a small data set
- Robust result even with a small random slip. Proof of this can be found in the literature [14], which compares the performance of spectral analysis and envelope analysis for processing a bearing signal with small random slips.

In terms of computation, the envelope analysis contains a band-pass filter, Hilbert transform, and complex amplitude calculation, which are acceptable for an embedded processor. In fact, the functions for realising envelope analysis, like FIR filter, FFT, and complex amplitude calculation, are already available in the CMSIS DSP library introduced in Section 3.2.

4.3.3 Cepstrum analysis

A cepstrum is the inverse Fourier transformation of the logarithm of a spectrum and can extract periodicity information in the spectrum [146]. It has been found useful in many areas, such as characterising the seismic echoes resulting from earthquakes and bomb explosions, pitch determination of human speech [147] and fault detection in rotating machines [142]. A special investigation for mechanical problems is detailed in [148].

For a discrete time signal $x(n)$, its cepstrum $c(n)$ is defined as:

$$c(n) = \mathcal{F}^{-1}\{log|\mathcal{F}(x(n))|\} \tag{4.21}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the forward and inverse Fourier transform and $|\bullet|$ represents the modulus of a complex signal. The name of cepstrum was derived by reversing the first four letters in "spectrum" [149]. Similarly, the terms in the cepstrum are named from the corresponding spectrum terms, as listed in Table 4.6 .

Table 4.6 Term naming in spectrum and cepstrum [142]

Name in spectrum	Name in cepstrum
Spectrum	cepstrum
Frequency	quefrency
Harmonics	rahmonics
Period	repiod
Low-pass filter	short-pass lifter
High-pass filter	long-pass lifter

The advantages of cepstrum analysis are summarized as below:

- Good extraction of periodicities, or repeated patterns. In the cepstrum, any periodicities, or repeated patterns is sensed as one or two specific components. It can also separate the overlapped sidebands or harmonic series in a spectrum, making it especially good candidates for cepstrum analysis [148].
- Bypass unknown transfer path and reveal local fault directly. According to [142], the localised faults are expressed at long quefrencies while the unknown transfer path is wrapped up at short quefrencies. This indicates cepstrum is insensitive to either the structure changes or the accelerometer mounting positions.

By calculating the cepstrum of the data set in Figure 4.4, the results are shown in Figure 4.11. A series of rahmonic components can be observed in the cepstrum of defect bearings and the peak of the first rahmonic components correspond to their fault frequencies listed in Table 4.3, which can be the evidence for the existence of relevant local bearing faults.

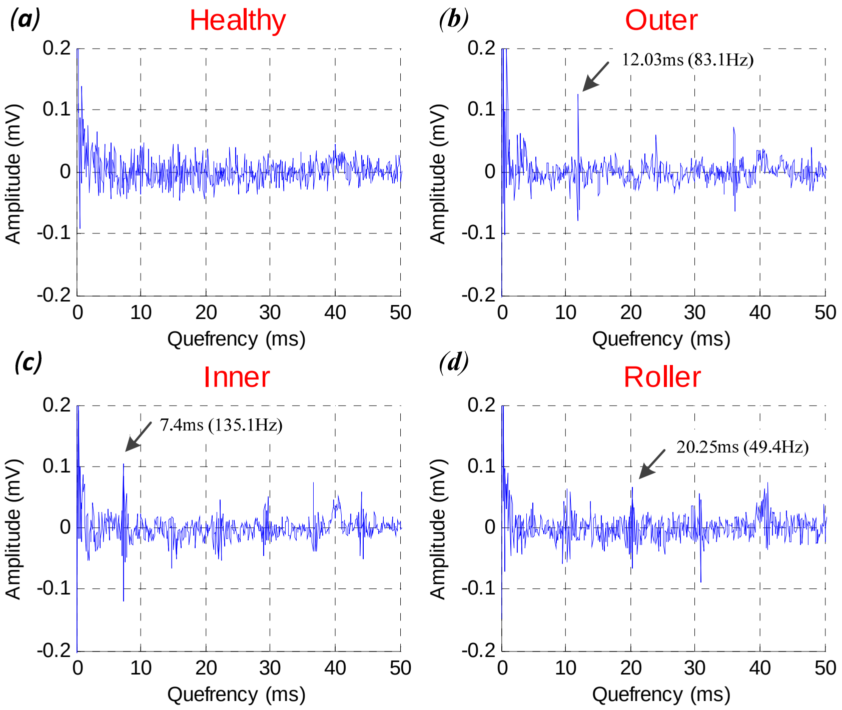


Figure 4.11 Cepstrum analysis result for vibration signal from different bearing conditions: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

Although cepstrum analysis can be employed for bearing fault diagnosis, its processing results are not straightforward to translate and the signal to noise ratio of the result looks low as the processing example shows. However, an improved version of cepstrum, named minimum variance cepstrum, outperforms a complex envelope and continuous wavelet for detecting an inner race fault in noisy environments [150].

In terms of computation, cepstrum analysis requires two FFT calculations a nonlinear logarithm calculation, making its implementation rather time-consuming.

4.4 Time-frequency analysis

The frequency-domain analysis provides an overall frequency distribution for the entire signal with a constant frequency resolution but with no time location information. For solving the decency in the frequency-domain analysis, a number of time-frequency domain techniques have been developed. Such techniques are expected to diagnose faults from a complex signal that has low SNR and contains a large number of frequency components [120]. The popular time-frequency analysis techniques include the short time Fourier transform (STFT), Wavelet transform (WT) and Hilbert-Huang transform (HHT), etc.

4.4.1 Short time Fourier transform

Short-time Fourier transform (STFT) is one of the earliest and also the most widely used time-frequency techniques [120]. It is obtained by sliding a window function over the original time signal and then performing Fourier transform over the windowed signals. The STFT for a discrete time signal $x(n)$ over a time window $w(m)$, is given by:

$$X(m,k) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j2\pi kn/N} \tag{4.22}$$

Take the data set in Figure 4.4 for example, the STFT calculation results are presented as spectrogram shown in Figure 4.12, where a sliding Hanning window of length 256 points was applied over the 4096 points of the signal with a step size of 6 points. By comparison, obvious high-frequency components can be seen in the spectrogram of all three fault bearing vibration signals. The impact mainly triggers the bearing structure to vibrate around three resonances: 4.5 kHz, 8 kHz, and 10 kHz.

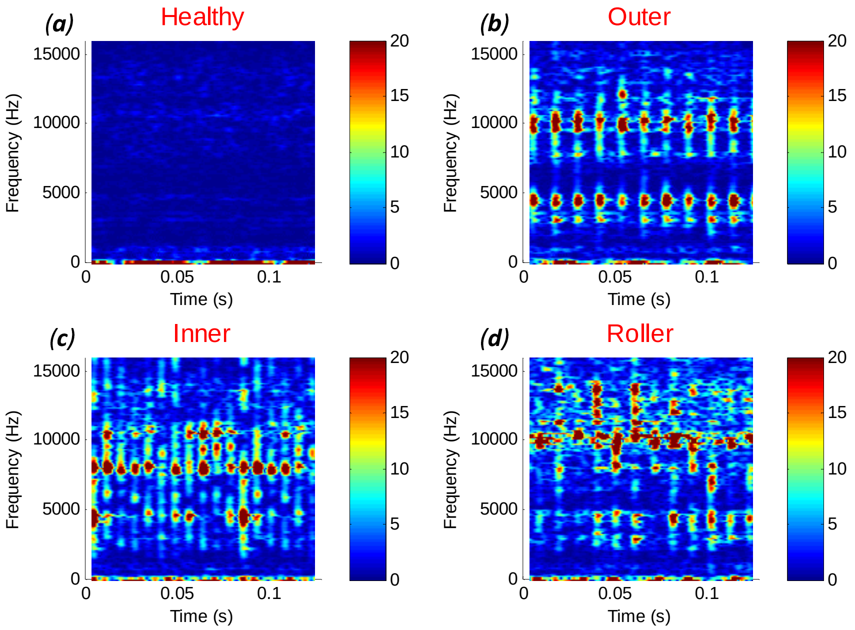


Figure 4.12 Spectrogram with FFT size of 256 and a step size of 6 points for vibration signal from different bearing conditions: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

It can be seen that not only the bearing resonances can be observed but also the impact happening time is extracted as well. Clear periodical impacts can be clearly observed in the spectrogram from bearing with outer race fault and inner race fault while they are not so obvious in the spectrogram from bearing with roller defect. The time period between resonance excitation can be employed for diagnostic purposes by relating it to the various characteristic bearing frequencies; however, it is not easy to identify the local bearing faults as effectively as envelope analysis.

In comparison with the spectrum analysis which only gives an overall frequency distribution of the signal under analysis, STFT provides both frequency and time information. This makes STFT capable of analysing non-stationary signals and uncovers the feature change with time.

One deficiency of STFT is its conflicting relationship between time and frequency resolution [120]. In other words, an STFT with finer frequency resolution usually has a loose time resolution and vice-versa. This means it cannot express a signal feature whose time interval is longer than the window width. In practice, it may need several trials to get a good balance between frequency and time resolution, making its practical usage quite tricky.

In terms of computation, the STFT calculation involves intensive computation and produces a much larger dataset than the raw data. Therefore, it is not a good option for embedded processing method on the wireless sensor node. A practical approach for the application of STFT might be using it to help find the resonances then build a band-pass filter for enhancing the SNR in the envelope analysis. Actually, the spectral kurtosis employs the frequency domain STFT method in combination with kurtosis to select an optimal band-pass filter for envelope analysis [151].

4.4.2 Wavelet

Wavelet transform (WT) is an effective method for processing non-stationary or transient signals. Since the 1990s, WT has been rapidly developed and its successful application can be found in various fields, such as image coding, compressing and edge detection [152], biomedical signal processing [153] and also mechanical fault diagnosis [17], [154], [155].

According to signal decomposition paradigms, the WT can be performed by the continuous WT (CWT) [156], discrete WT (DWT) [157], [158], Wavelet packet transform (WPT) [159], and extended WT with post-processing [160]. In general, CWT provides a thorough exploration on the signal and contains highly redundant information, resulting in it requiring a significant amount of computation time and resources. The WPD and DWT just include a subset of the exploration of the CWT but contain sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time.

4.4.2.1 Continuous Wavelet transform

A continuous Wavelet transform for a time series signal $x(t)$ is defined as [9]:

$$W(a,b)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}x(t)\psi^{*}(\frac{t-b}{a})\,dt \tag{4.23}$$

where a is the scale parameter, b is the time parameter and $\psi(\cdot)$ is a wavelet, which is a zero average oscillatory function centred around with a finite energy, and "*" denotes complex conjugate. A number of wavelets has been developed for processing signals with different features, such as Morlet, Mexican hat, Haar and Daubechies.

A good explanation of CWT is that it "expresses the signal in a series of oscillatory functions with different frequencies at a different time by dilations via the scale parameter a and translations via the time parameter b" [9]. Similar to Fourier transform, CWT has a discrete version by discretizing a, b, and $x(t)$ and also a fast wavelet transform calculation is available for discrete wavelet transform calculation similar to FFT. Like the power spectrum in Fourier analysis, CWT uses a scalogram defined as $|W(a,b)|^2$ to interpret the signal.

Take the data set in Figure 4.4 for example; the scalogram results are given in Figure 4.12, where the scales have been translated to their corresponding frequencies for convenient understanding. Similar to the spectrogram results in Figure 4.12, the resonances and periodical impacts can be clearly observed in the scalogram of the vibration signal from defect bearings.

CWT is similar to STFT in that it provides a time-frequency map of the signal being analysed. The main difference and advantage of WT are its capability to provide multi-resolutions, which indicates it can easily adjust its time and frequency resolution as required. By employing WT, a high-frequency resolution can be produced at low frequencies and a high time resolution can be achieved at high frequencies for signals with long duration low frequencies and short duration high frequencies [9].

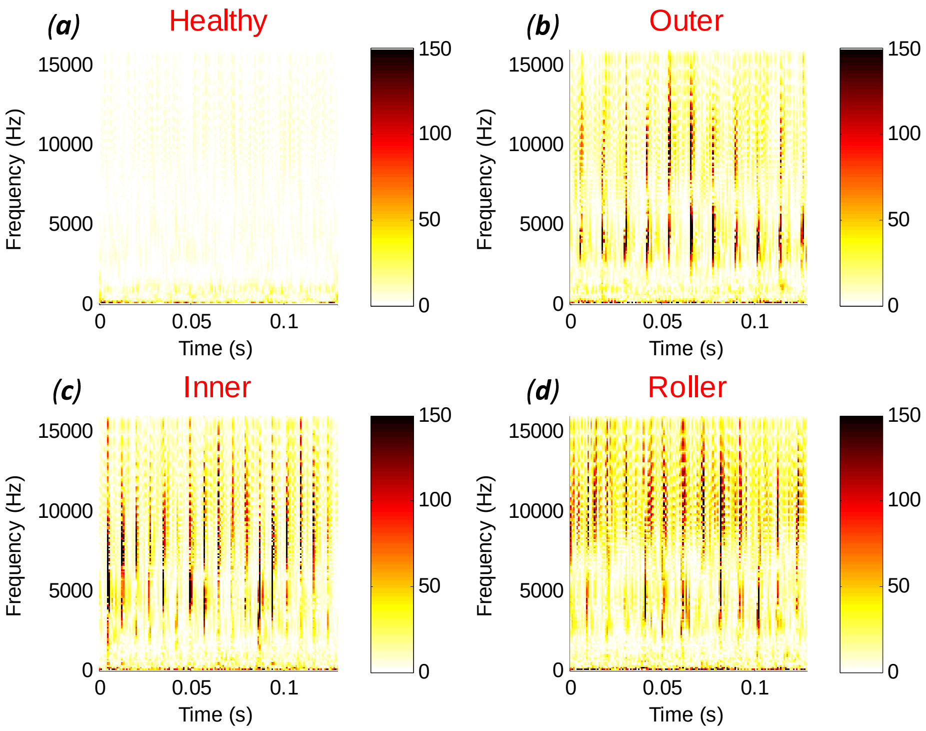


Figure 4.13 Scalogram with scale converted in corresponding frequency for vibration signal from different bearing conditions: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

4.4.2.2 Discrete Wavelet transform

In practice, DWT can provide sufficient resolution for signal analysis and has been more widely employed than CWT because of its much faster computing speed in comparison with CWT. In the area of mechanical fault diagnosis, DWT has mainly been employed for two purposes: de-noising [161], [162] and envelope extraction [17], [163], [163].

Lin and Qu developed a de-noising method based on Morlet wavelet analysis and applied it for rolling bearing signal, the result of which shows much clearer periodic impulses than the raw signal [161]. A de-noising method based on Morlet wavelet filter is developed to obtain optimal time-frequency resolution by applying minimal Shannon entropy criterion for determining shape factor and singular value decomposition (SVD) for determining scale. The experimental results verify its effectiveness for extracting weak periodic impulse signatures [162]. He and Jiang et al combined optimal wavelet filtering and sparse code shrinkage de-noising method to effectively extract the impulsive features hidden in the signals of low SNR [164].

Chen and Li applied one-dimensional DWT on the bearing signal using a db4 wavelet and then calculated the envelope spectrum on the d1 level. The experimental results show the frequency of failure location matches well with the theoretical one using this method [163]. Sheen and Hung proposed a wavelet-based envelope function derived from Morlet wavelet to extract envelope for vibration signal [165]. Yiakopoulos and Antoniadis used a squared rectifier in combination with Wavelet decomposition to get the approximation on the second or third levels as the envelope [17].

The DWT of a discrete time series signal $x(n)$ is calculated by passing it through a series of filters, including a low pass filter with impulse response $g(n)$ and a high pass filter with impulse response $h(n)$, which are related to each other and known as quadrature mirror filters. The outputs of the two filters give the detail coefficients (from $h(n)$) and approximation coefficients (from $g(n)$). After passing through the filters, the outputs can be down sampled by a factor of 2 according to Nyquist sampling theory. This finishes one level of decomposition, as shown in Figure 4.14.

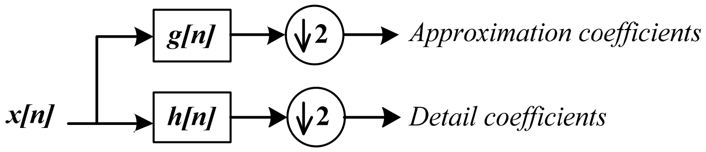


Figure 4.14 One level of decomposition

Both DWT and WPT decomposition are performed by a series of such filters, but a small difference exists between them. As illustrated in Figure 4.15, in the DWT, only approximation coefficients in each level are decomposed while the WPD has a fully explored binary tree with both the detail and approximation coefficients being decomposed.

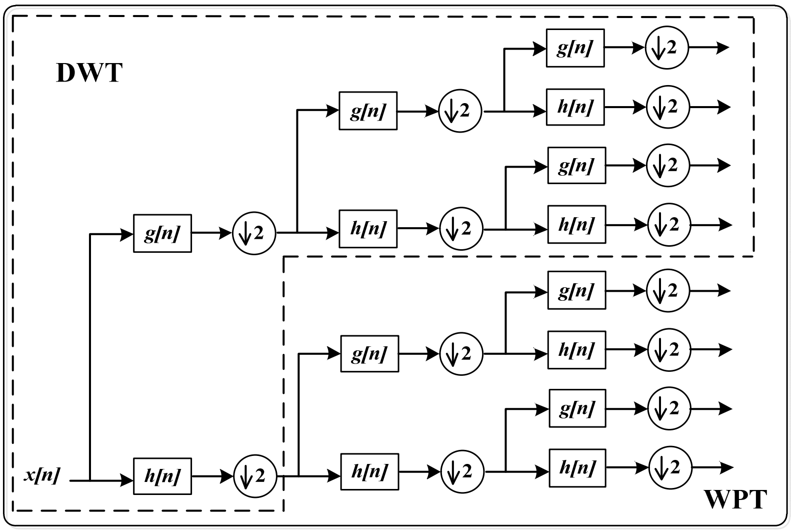


Figure 4.15 WPT and DWT decomposition over 3 levels with $g[n]$ being the low-pass approximation coefficients and $h[n]$ is the high-pass detail coefficients

In terms of computation, DWT only involves a series of convolution and downsampling process and the computation results have the same memory occupation as the original input data. This makes DWT a good candidate for embedded processing on the wireless sensor node.

4.4.3 Hilbert-Huang transform

Hilbert-Huang transform (HHT), is a method developed by Huang et al [166]. As an adaptive data analysis method, it is designed specifically for analysing data from nonlinear and non-stationary processes and has been successfully applied in various areas, such as biomedical signal [167], seismic signal [168] and mechanical fault diagnosis [169]–[171].

HHT are mainly composed of two procedures: empirical mode decomposition (EMD) and Hilbert spectral analysis (HSA). An input dataset is firstly decomposed by EMD method into a finite number of components, named intrinsic mode functions (IMFs). Then, Hilbert transform is employed to the instantaneous frequencies of these IMFs as functions of time. The final presentation of the results is an energy-frequency-time distribution, designated as Hilbert spectrum [172].

Because HHT provides good computation efficiency and does not involve the concept of frequency resolution and time resolution, it has been considered as a powerful method for bearing fault detection and diagnosis [169]. In the past few years, HHT has been explored by several researchers for bearing fault diagnosis. Rai and Mohanty compared the time domain and frequency domain of IMFs and found that the frequency domain approach in HHT is more effective and efficient for bearing fault diagnosis [170]. Li and Zhang used the marginal spectrum obtained from the IMFs to detect the localised fault in a roller bearing and the results show that this method can increased the spectral resolution [173]. Peng et al improved the HHT by pre-processing the signal using wavelet packet and a better resolution is achieved both in the time domain and in the frequency domain than the scalogram [169].

By using the data set in Figure 4.4, the Hilbert spectrum results are presented in Figure 4.16, where the colour represents the strength of the instantaneous frequency. Similar to the spectrogram results in Figure 4.12 or the scalogram results in Figure 4.13, the Hilbert spectrum also clearly shows the resonances and periodical impacts for the signal from defect bearings.

Although HHT can provide very sharp time and frequency resolution, its computation involves the generating of a series of IMFs components, which would occupy a large memory, making it not a good candidate for the implementation on a processor with limited memory. It can be employed in the data processing centre for more detailed and precise analysis.

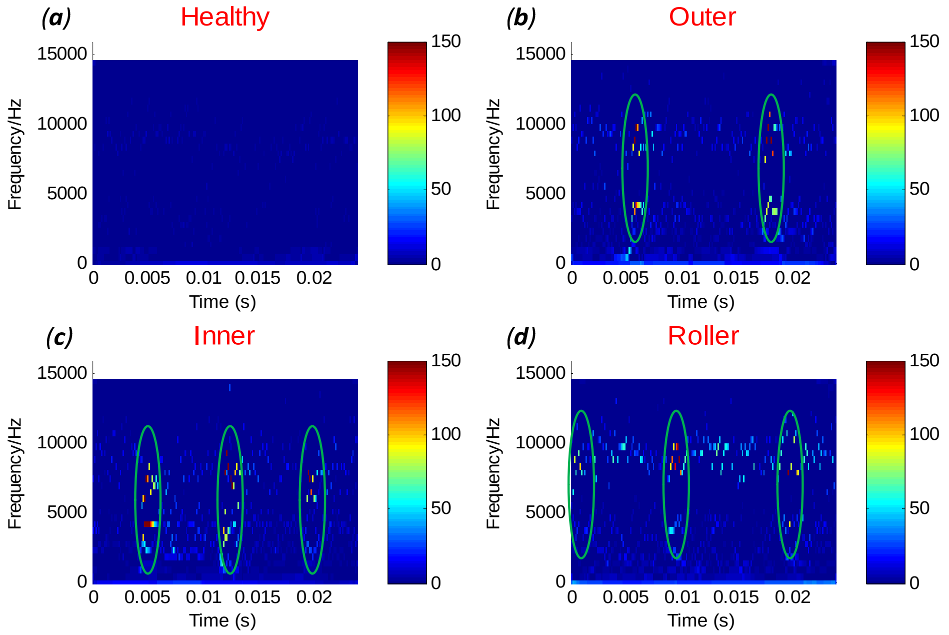


Figure 4.16 Hilbert spectrum for vibration signal from different bearing conditions: (a) healthy, (b) outer race fault (c) inner race fault and (d) roller fault

4.4.4 Other time-frequency analysis

Besides the above discussed three time-frequency methods, Wigner-Ville distribution (WVD) is another common time-frequency analysis method. WVD can provide a sharper time localisation when compared to STFT as the WVD at each time point is relatively independent of the window size [120]. However, WVD contains some cross terms and wavelet analysis can usually provide better processing results [174]. Furthermore, the computation of WVD is rather time-consuming and produces large output data similar to that of the STFT; therefore, it is also not a suitable method for embedded computing.

4.5 Selection of signal processing algorithms

In the above three sections, a number of signal processing algorithms are investigated and discussed for extracting bearing fault features. The advantages and disadvantages of these investigated signal processing algorithms are summarized and shown in Table 4.7. These signal processing methods have their unique advantages in extracting some specific features. Some methods may need to be further processed to get a clearer feature. For example, TSA is more often employed as a de-noising method and it may need an additional spectral analysis or envelope analysis to make fault features easy to interpret.

After the above investigation on the signal processing algorithms, a proper signal processing method for being embedded on sensor nodes should satisfy the following criteria:

- **Detect abnormal conditions.** The embedded algorithm should be able to extract a feature that can indicate the abnormalities of the monitored component. Apparently, all these investigated methods can extract such a feature.
- **Detect faulty components**. The bearing has four types of commonly localised faults. It would be desirable if the method can extract a feature to represent the type of faults, which can be trended for severity monitoring. The statistical parameters and probability density function have no such capability and thus are rejected.

- **Suppress noises**. The vibration signals very likely contain interference noises from nearby components. From Figure 4.8 and Figure 4.11, it can be seen that it is not easy to interpret the fault types from the noisy results of spectral analysis and cepstrum, thus their reliability and robustness are doubted.
- **Smaller results data set.** One main purpose of employing embedded processing on distributed wireless sensor nodes is to extract a small feature set for transmission instead of sending the large raw data set. Otherwise, there would be much less meaningful value for such an operation. Although the time-frequency analysis techniques like STFT, CWT and HHT extracted much information from the signal, they are very computation intensive and produce a much larger data set than the original one, making them not suitable candidates for embedded processing. These techniques can be employed in the data processing centre for more detailed analysis.

The above selections are illustrated in Figure 4.8. After selection by the above algorithms, the proper candidates for embedded processing include the morphological filter, TSA, envelope analysis, and DWT. The main purpose of the morphological filter is to extract envelope detection and can be employed as part of envelope analysis. The TSA method is rejected due to it requiring an additional synchronisation signal which both increases the sensor node complexity and cost. DWT is much more flexible and capable of processing signals for de-noising and envelope analysis. The envelope analysis is more like a combination of several methods: filtering, envelope detection and spectral calculation.

Apart from the above considerations, the efforts for implementing the algorithms should also be taken into consideration. There is usually no library for morphological filter and DWT on embedded processor meanwhile the envelope analysis can utilise the library functions like FIR filter, FFT etc. which are usually available and optimised for embedded processors.

For the above reasons, the envelope analysis is selected as the algorithm for the embedded processing on the distributed wireless sensor nodes for feature extraction. In the following chapter, it will be implemented and evaluated for bearing fault feature extraction. The methods like the morphological filter, DWT, TSA, and cepstrum are not suitable for embedded processing in the current stage. With performance being improved or being combined with proper processing, they could be employed on the distributed wireless sensor nodes for extracting proper features in the future.

Table 4.7 Advantages and disadvantages of investigated signal processing algorithms

Algorithm	Advantage	Disadvantage
Statistical parameters	➢ Computation efficient ➢ Small result in data size	➢ Cannot detect small defects at early stages ➢ Difficulty in the diagnosis of localised fault types
Morphological filter	➢ Good extraction of envelope ➢ Computation efficient	➢ Requires further processing ➢ No available library on embedded processor
Probability density function	➢ Computation efficient ➢ Small result in data size	➢ Cannot detect small defects at early stages ➢ Difficulty in the diagnosis of localised fault types
TSA	➢ Helpful for removing random noises	➢ Requires additional synchronisation signal ➢ No available library on embedded processor
Spectral analysis	➢ Available library on embedded processor ➢ Acceptable computation efficiency	➢ Not easy to observe fault information ➢ Sensitive to small random slip ➢ Difficult to identify inner race and roller defects
Envelope analysis	➢ Clear representation of localised bearing faults ➢ Capability to represent fault features in a small data set ➢ Robust result even with a small random slip. ➢ Acceptable computation complexity ➢ Small result in data size	➢ Requires adjustment to enhance SNR ➢ No available library on embedded processor
Cepstrum	➢ Good extraction of periodicities, or repeated patterns. ➢ Bypass unknown transfer path and reveal local fault directly	➢ Not easy to interpret result ➢ Low SNR ➢ Computation time-consuming

Table 4.7 Advantages and disadvantages of investigated signal processing algorithms (continued)

Algorithm	**Advantage**	**Disadvantage**
STFT	➢ Contains both time and frequency information	➢ Very computation time-consuming ➢ Memory consuming ➢ Conflicted time and frequency resolution
CWT	➢ Contains both time and frequency information ➢ Provide multi-resolution	➢ Very computation time-consuming ➢ Memory consuming
DWT	➢ Contains both time and frequency information ➢ Provide multi-resolution ➢ Acceptable computation time ➢ Good for de-noising and envelope extraction	➢ No available library on embedded processor ➢ Requires adjustment to enhance SNR ➢ Requires further processing
HHT	➢ No concept of frequency resolution and time resolution ➢ Good for nonlinear and non-stationary processing	➢ Memory consuming ➢ No available library on embedded processor

Table 4.8 Selection of signal processing algorithms for embedded processing

Algorithm	Detect abnormal conditions	Detect faulty components	Suppress noises	Smaller results data set	Suitability
Statistical parameters	✓	✗	✓	✓	✗
Morphological filter	-	-	✓	✓	✓
Probability density function	✓	✗	✓	✓	✗
TSA	-	-	✓	✓	✓
Spectral analysis	✓	✓	✗	✓	✗
Envelope analysis	✓	✓	✓	✓	✓
Cepstrum	✓	✓	✗	✓	✗
STFT	✓	✓	✓	✗	✗
CWT	✓	✓	✓	✗	✗
DWT	✓	✓	✓	✓	✓
HHT	✓	✓	✓	✗	✗

Notes: ✓ means the corresponding criteria can be satisfied.

✗ means the corresponding criteria cannot be satisfied.

4.6 Summary

As a typical example of modulating and impulsive signal, the characteristics of bearing vibration signal are studied in this chapter. On this basis, a number of popular signal processing algorithms for bearing fault diagnosis are investigated and evaluated for processing a group of experimental data from a bearing test rig. From the investigations, the envelope analysis is deemed as the most suitable algorithm for being embedded on the wireless sensor node to extract bearing fault features. A detailed implementation of envelope analysis on the proposed vibration sensor node will be explained in Chapter 5.

CHAPTER 5

ENVELOPE ANALYSIS IMPLEMENTATION ON THE WIRELESS SENSOR NODE

In this chapter, the classical and widely accepted envelope analysis method is implemented on the proposed wireless sensor node. The implementation is introduced in three successive procedures: data acquisition, data processing, and data transmission. The program is written with a purpose to optimise the signal processing efficiency and thus reduce the computation time. In this way, the microcontroller can stay in the low power mode longer; thereafter the power consumption can be significantly reduced. For each step, the implementation is validated by processing a typical signal. In the end, its performance for extracting bearing fault features is validated by processing a real bearing signal and its effectiveness on data reduction is analysed.

5.1 Data acquisition

To process the vibration signal on a microcontroller, the analogue signal needs to be firstly digitised. In this design, this procedure is realised by the on-chip analogue to digital converter (ADC) that has 12-bit resolution and a maximum conversion rate of 1 MSPS, which is much higher than our sampling rate requirements. Therefore, the ADC needs to be started periodically to perform the data conversion.

5.1.1 Data acquisition structure

To promise equal sampling interval, a timer event is used to trigger the ADC starting and the converted results are moved to the on-chip RAM through a direct memory access (DMA) channel. By using DMA to move the ADC conversion results instead of the timer interrupt plus CPU moving data way, the required interrupt times can be significantly reduced. In this way, the CPU is released from the trivial data acquisition process and thus can more focus on the intensive signal processing or enter into the low power mode to save power.

The schematic of the data acquisition process by combing DMA and timer is illustrated in Figure 5.1. Here, a double buffer method, usually named *Ping-Pong* structure, is utilised to enable the data acquisition and data processing to operate in parallel. Its mechanism works like this: when DMA moves ADC conversion results to *Ping* buffer, the CPU can process data in *Pong* buffer and vice versa. As long as the time for CPU data processing is shorter than that of DMA moving data, the signal processing can be continuous without losing data points. It can be seen that this structure makes the signal processing much efficient [175].

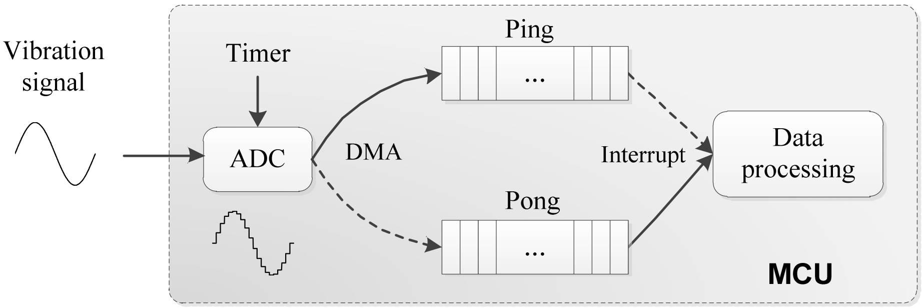


Figure 5.1 Schematic of the double buffer DMA data acquisition

Furthermore, this mechanism enables a long signal to be divided into small segments for real-time processing, which is suitable for a microcontroller with limited memory storage. The benefits of this mechanism can be seen again in Chapter 6, where the signal is processed by segment and concatenated together later.

Another benefit of using Ping-Pong structure is that it can be used to build a buffer larger than the maximum movement data size of the DMA unit without losing data points. For the TIVA-C processor, the maximum moving data point is limited to 1024 but can be extended to much longer one with the *Ping-Pong* structure.

5.1.2 Data acquisition implementation

The construction of this data acquisition structure involves three peripheral modules, which is a bit complicated. A brief program flow chart for building this structure is shown in Figure 5.2, which has been divided into three main stages: initialisation, start and interrupt.

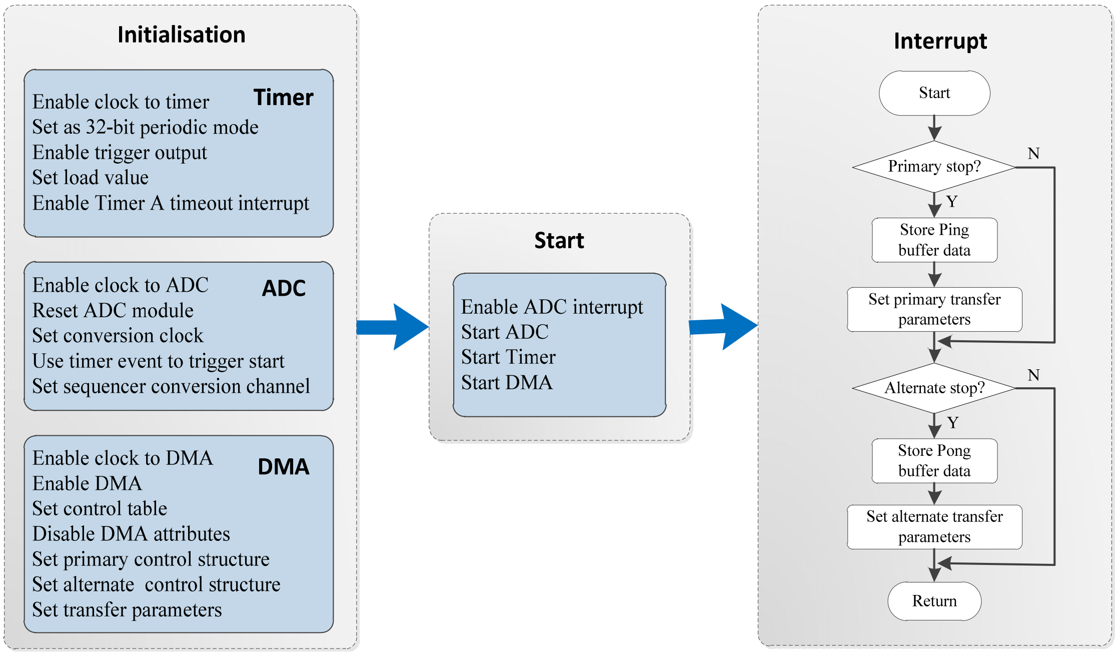


Figure 5.2 Flow charts for data acquisition process

In the initialisation stage, the timer works in 32-bit periodical mode and its load value determines the sampling interval. The ADC works at a high conversion rate and is configured to start conversion by the timer trigger event. For the DMA, both the primary and alternate control structure is configured and its initialisation parameters include source address, destination address, data type and data length, etc. Of note, the first step for configuring all these three peripherals is to enable their corresponding clock. This indicates the power of each peripheral is controllable and the one not in use can be powered off to save power. This is a powerful feature that makes the Cortex-M processor very power efficient.

For the start stage, the ADC interrupt needs to be enabled and then the three peripherals are started. Although the ADC interrupt is enabled, when the ADC conversion finishes, it actually triggers the DMA to move the conversion results to the specified data buffer and only when the specified size of data has been moved, it will notify the CPU to enter the interrupt service routine (ISR).

Inside the ISR, it reloads the parameters for the stopped primary or the alternate control structure and stores the data from *Ping* or *Pong* buffer to a larger buffer. Due to the working mechanism of the DMA unit on this microcontroller, it cannot reload the initial value automatically and thus needs to be reloaded by the program. Of note, when CPU enters the ISR to reload one control structure, the DMA is actually working using the other control structure, so no data will be lost when CPU is inside the ISR.

It can be seen that the programming involves quite a few detailed register levels of operation. Although this can be finished by configuring the relevant registers one by one by consulting the datasheet, it is both time-consuming and error-prone. Fortunately, there is already a convenient library named TivaWare available [176], which includes the driver libraries and example codes for the peripherals and typical applications. With its help, the development period can be significantly shortened.

5.1.3 Validation of data acquisition process

To validate the correctness of the established data acquisition structure, an analogue modulating signal is connected to the ADC input and the *Ping-Pong* structure is employed to acquire 1024 points of data with the *Ping* and *Pong* buffer size of 256 points and sampling frequency of 32 kHz.

The execution graph for this working mode is illustrated in Figure 5.3, which is acquired in the development environment Code Composer Studio (CCS). The program is written on the real-time operating system (RTOS), named TI-RTOS [177]. The execution graph contains abundant RTOS operation information and will be not explained in detail. Here, it is just employed to help understand the working mechanism of the *Ping-Pong* structure.

During the entire data acquisition process, only four ADC interrupts can be observed in Figure 5.3. This means the working buffer for DMA changes four times to finish this long data acquisition process. Besides the ADC interrupt, a series of system tick interrupts can be clearly noticed in the execution graph, which is the fundamental clock for the RTOS and ticks every 1 ms. It can act as the time consultancy for the ADC interrupt. It can count that the interval between two successive ADC interrupt contains about eight system ticks, which means the ADC interrupt interval is about 8 ms. This is in accordance with the time for sampling 256 points of data, i.e. 256x(1/32000)=0.008s.

Besides, a software interrupt (SWI) can be seen for each ADC interrupt, which is employed especially for moving data in *Ping* or *Pong* buffer to the large buffer. The benefits for posting a special SWI to finish the time-consuming data moving instead of directly moving data inside the hardware interrupt are that it reduces the time occupied by the hardware interrupt and thus reduces the conflict between hardware interrupts.

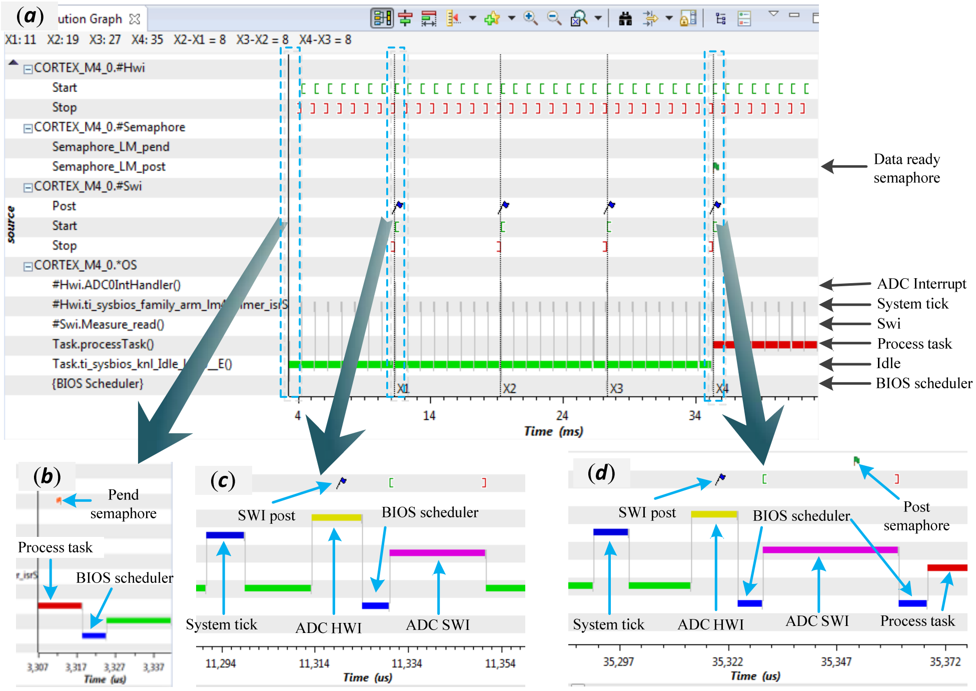


Figure 5.3 Execution graph of data acquisition process: (a) full view, (b) magnified view of process task start point, (c) magnified view of first DMA interrupt and (d) magnified view of last DMA interrupt for the data collection

In the debug mode, by placing a breakpoint in the position where 1024 points of data collection finishes, the microcontroller can be halted. With the help of the graph tool in CCS, the waveform of the collected signal is visualised in Figure 5.4. The waveform is quite smooth, which validates the continuous of the data acquisition process. About 20.5

periods of high-frequency sine wave signal is captured in this frame of data, therefore, its frequency can be calculated as 20.5/(1024/32000) = 640.625 Hz, which is quite close to the carrier frequency of 640 Hz. This confirms the correctness of the defined sampling frequency in the program.

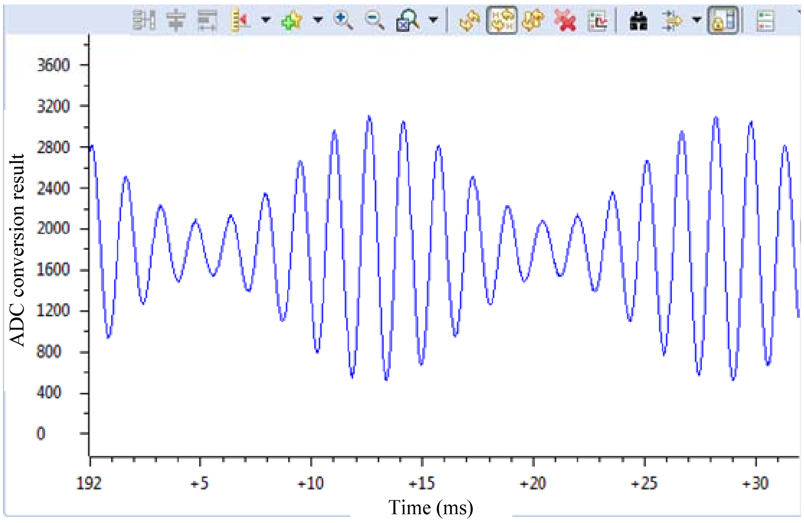


Figure 5.4 Visualised data in CCS of a modulating signal using *Ping-Pong* structure

5.2 Data processing

The data processing is the most time and memory consuming part for this application. As discussed in Chapter 3, the envelope analysis is an algorithm well-recognised by the industrial area and can produce a smaller result dataset compared to the raw data. Recall the procedure of envelope analysis in Figure 3.10, the implementation of envelope analysis is accomplished in three successive steps and explained in the following three subsections.

5.2.1 Band-pass filter

The band-pass filter is employed to keep the structure resonance signal and suppress the noises outside the pass band, in which way the SNR of the amplitude modulated resonance signal can be enhanced.

5.2.1.1 Design

In digital signal processing, FIR or IIR type filters are usually employed to achieve digital filtering. In this application, the FIR type filter is employed due to its simple structure, linear phase response and guaranteed stable performance.

The FIR filter is typically designed offline to get a group of coefficients, which are then stored in the embedded processor for computation during runtime. The design methods for the FIR filter include impulse response truncation method, windowing design method or and Parks-McClellan optimal filter design method [178]. In practice, it can be conveniently designed using well-recognised computing packages, like Matlab from MathWorks.

Apparently, the band-pass filter should keep the modulated characteristic frequencies and their harmonics information. In bearing diagnostics, it is often desirable to be able to detect up to the third harmonic of the bearing defect frequency in the envelope spectrum [49]. Recall the characteristic frequencies of bearing faults in Section 3.1, the maximum one is from the inner race fault at 135.5 Hz, whose third harmonics are inside 500 Hz. Therefore, a bandwidth with 1 kHz can enclosure the first three harmonics of all fault characteristic frequencies.

The *fdesign.band-pass* function in Matlab is employed to design the band-pass filter. An 81-tap FIR type filter with a passband between 1 kHz and 2 kHz is designed using this function. Its impulse response, 512-point magnitude and phase response are presented in Figure 5.5. It can be seen that the band-pass filter has a flat magnitude and linear phase response in the expected pass band (1 kHz to 2 kHz) and the frequency components outside the pass band are attenuated by more than 40 dB.

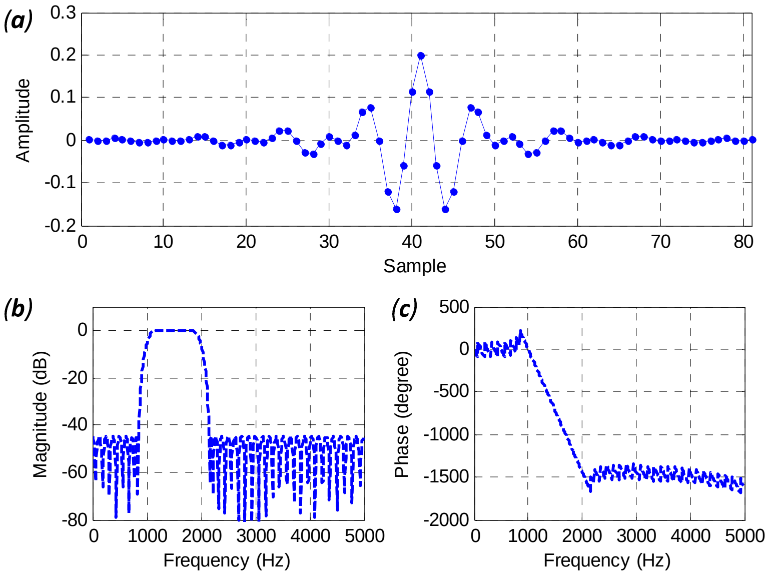


Figure 5.5 Feature of the designed band-pass filter: (a) impulse response, (b) amplitude response and (c) phase response

5.2.1.2 Implementation

The calculated coefficients are stored as a constant table in the Flash type memory on the MCU to save the valuable RAM type memory. For accuracy considerations, the coefficients are stored as *single floating point* format and complex type filter function is employed. As the MCU has an integrated FPU unit, the floating point calculations can be accomplished efficiently. In addition, this has avoided the overflow problem which is quite common and annoying for fixed-point computations.

For the signal processing, CMSIS-DSP library [179] is employed for implementation. By using this optimised DSP library, the program development can be significantly simplified while it still has good computational efficiency. For using this library on TIVA C microcontroller, there needs a migration and guide can be found in [180].

Two functions are used for the FIR filter implementation with *arm_fir_init_f32* for structure initialization and *arm_fir_f32* for data filtering. As shown in Figure 5.6, a computing buffer is required for the FIR filter calculation and its length is determined by the length of the coefficients and input data buffer. To balance between computing memory and efficiency, the large input data set is divided into small datasets with the size of *M* passing through the FIR filter. In the program, *M* is set as 64 and data size number *K=2048/64=32*.

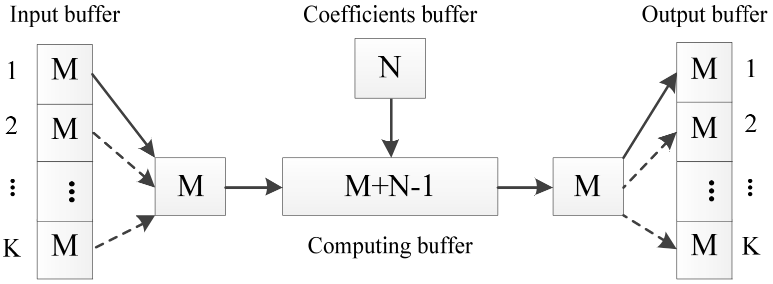


Figure 5.6 Illustration of buffer usage for FIR filter

Of note, the FIR filter produces a constant group delay. For an FIR filter with N tap, its group delay D can be calculated by:

$$D = \frac{N-1}{2} \tag{5-1}$$

For the FIR filter in this design, there exist *(81-1)/2=40* samples delay for the output data compared with the input data. Moreover, the beginning of 40 samples output of FIR filter is invalid.

Supposing the CPU runs at 80 MHz, the average time consumption for processing one data point for FIR filter length ranging from 15 to 250 is shown in Figure 5.7.

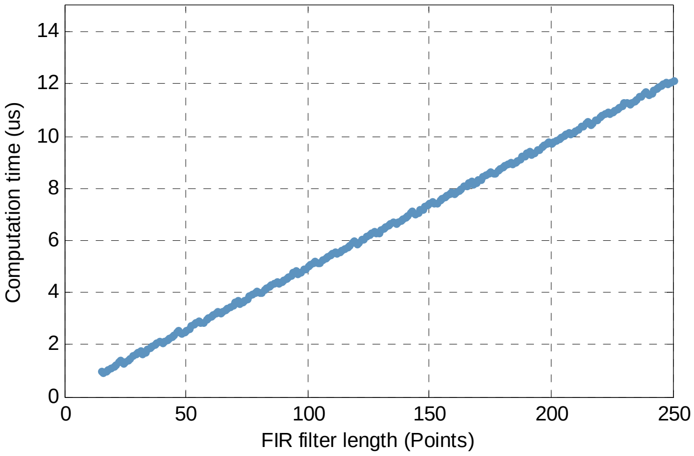


Figure 5.7 Average time consumption for filtering one data point using an FIR filter

5.2.1.3 Verification

To verify the implementation of the band-pass filter, a simulated signal is filtered on the microcontroller and exported in the debug mode for visualisation. From the raw signal and its spectrum in Figure 5.8(a) and (b), the raw signal contains a mixture of frequency components of 500 Hz, 1.5 kHz, and 2.5 kHz. After passing through the band-pass filter, as shown in Figure 5.8(c) and (d), the frequency components of 500 Hz and 2.5 kHz have been significantly attenuated while the amplitude of 1.5 kHz is kept almost the same as the raw signal. It can also be noticed that the beginning of the filtered signal is useless, which is caused by the group delay and will be ignored in the processing of next step.

To verify the preciseness of the calculation on the microcontroller, the raw signal is filtered using the same band-pass filter coefficients by Matlab and is deemed as the baseline signal. The MCU computed results and baseline results are compared in Figure 5.9(a), which shows a slight difference in beginning parts and the two signals start to show good similarity after 60 sample points. It can be seen one point delay exists for the baseline and MCU results, which may be caused by the filter implementation method and does not affect further processing. After compensating the delay point, a scatter plot between baseline and MCU results for the sample points from 80 samples to 150 samples are presented in Figure 5.9(b). It can be seen that the MCU results fit quite precisely with the baseline results.

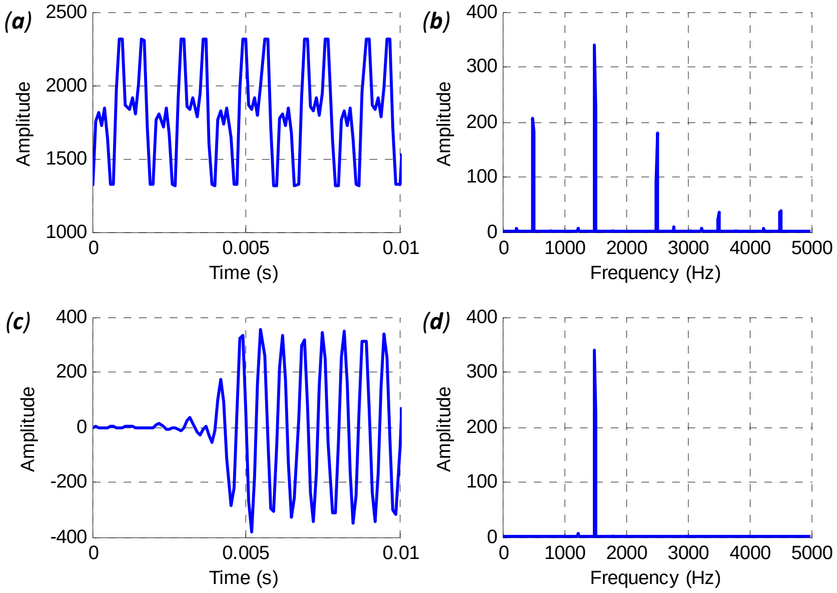


Figure 5.8 Band-pass filter implementation verification: (a) raw signal in time domain, (b) spectrum of raw signal, (c) filtered signal in time domain and (d) spectrum of filtered signal

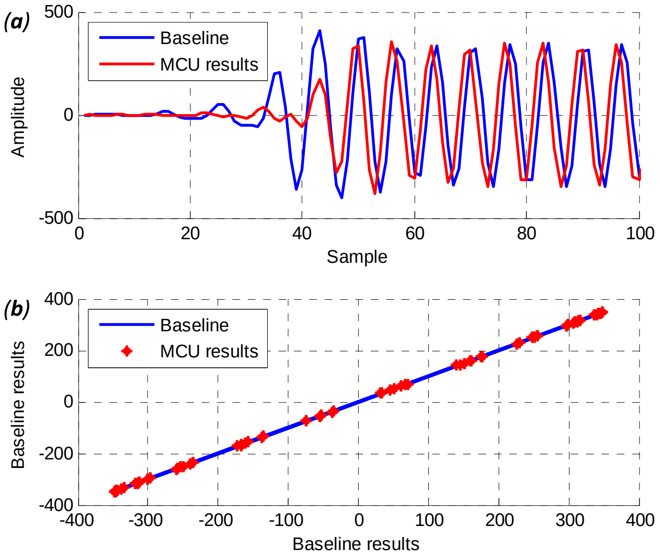


Figure 5.9 (a) MCU filter results in comparison with baseline results and (b) scatter plot between baseline and MCU results with solid line being 1:1 line of baseline results

5.2.2 Envelope detection

After band-pass filter, the signal needs to be demodulated to get its envelope. As a precise detection method [145], Hilbert transform is employed for the envelope detection.

5.2.2.1 Implementation of Hilbert transform

Basically, Hilbert transform produces an analytic signal, whose real part being the original signal and the imaginary part being its Hilbert transform [181]. Then, the envelope of the original signal can be obtained by calculating the modulus of the analytic signal.

Hilbert transform can be computed either in the time domain or in the frequency domain. Here the frequency domain method is employed because it is more straightforward to understand and a comparative study of the time domain and frequency domain method will be presented in Chapter 8. As shown in Figure 5.10, the analytic signal x_a is obtained by a forward FFT, zero those coefficients in the negative frequency band and double those coefficients in the positive frequency band, and then the inverse FFT yields the analytic signal. A more accurate result is obtained by weighting the coefficients via a windowing function shown in Eq. (5.2) [182], [183].

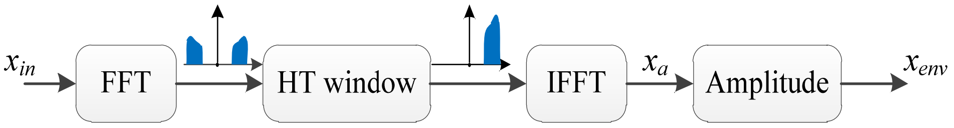


Figure 5.10 Schematic of the Hilbert transform based envelope detection in the frequency domain

$$w[n] = \begin{cases} 1, & n = 0, \frac{N}{2} \\ 2, & n = 1, 2, \cdots, \frac{N}{2} - 1 \\ 0, & n = \frac{N}{2} + 1, \cdots, N - 1 \end{cases} \tag{5.2}$$

Thereafter, the envelope x_{env} of a signal x_{in} can be calculated using:

$$X_{in} = fft(x_{in}) \tag{5.3}$$

$$X_a(\mathrm{n}) = \begin{cases} X_{in}(n), & n = 0, \frac{N}{2} \\ 2X_{in}(n), & 1 \leq n \leq \frac{N}{2} - 1 \\ 0, & \frac{N}{2} + 1 \leq n \leq N - 1 \end{cases} \tag{5.4}$$

$$x_a = ifft(X_a) \tag{5.5}$$

$$x_{env} = \sqrt{x_a .* \, x_a^*} \tag{5.6}$$

where X_{in} is the FFT of x_{in}, X_a is the FFT of the analytic signal x_a and x_a^* is the conjugate of x_a.

It can be seen that Hilbert transform involves one forward and one inverse FFT calculation, making it the most time-consuming part of the implementation of entire envelope analysis. Thus, it is worth optimising its calculation. In the CMSIS DSP library, four computation methods are available for the FFT calculation, as shown in Table 5.1. In specific, *arm_cfft_f32* and *arm_cfft_radix2_f32* can operate on complex type data while *arm_rfft_f32* and *arm_rfft_fast_f32* can operate on real-valued type data. In practice, the measured signal only contains the real-valued part and thus all these four methods can be used for FFT calculation.

Table 5.1 FFT calculation functions in CMSIS DSP library

Function name	**Description**
arm_cfft_radix2_f32	This FFT function operates on floating complex type data and the radix2 algorithm is employed for calculation.
arm_cfft_f32	This FFT function operates on floating complex type data and a mixed-radix algorithm is employed for fast calculation.
arm_rfft_f32	This FFT function can only operate on floating real-valued type data.
arm_rfft_fast_f32	This FFT function can only operate on floating real-valued type data. This function is expected to be faster than *arm_rfft_f32*.

For memory usage, the real-valued data consumes half of that for complex-typed data. Based on the test, it is found that the two complex type FFT functions can reuse the calculation buffer whilst the two real-valued FFT functions must employ different input and output buffers. Thus, the complex and real-valued type FFT calculation consume the same size of computation buffer. This also indicates the buffer for FFT calculation is at least twice of that in the input data.

Recall that the computations are achieved in single floating point format, which indicates each data point requires a four-byte memory to store. Therefore, the computation buffer for an FFT calculation with N points should be larger than $4N \times 2 = 8N$ bytes. Remember in Chapter 3 that the SRAM of the employed MCU is 32 kB. The maximum FFT size N can be calculated as 32k/8=4k. Note that MCU also requires SRAM memory for the other operations, the maximum FFT size becomes 2048 points.

According to their descriptions, *arm_rfft_fast_f32* should be the fastest one and *arm_cfft_radix2_f32* the slowest one. To get a more convincible conclusion, the methods are benchmarked in CCS by using the timestamp function provided in TI-RTOS and the results are presented in Figure 5.11. It can be seen that the fastest *arm_rfft_fast_f32* function is more than two times faster than the slowest *arm_cfft_radix2_f32* method for the 2048 points calculation. It needs to be mentioned that the computation speed improvement is achieved with the price of storing large constant tables in the flash memory.

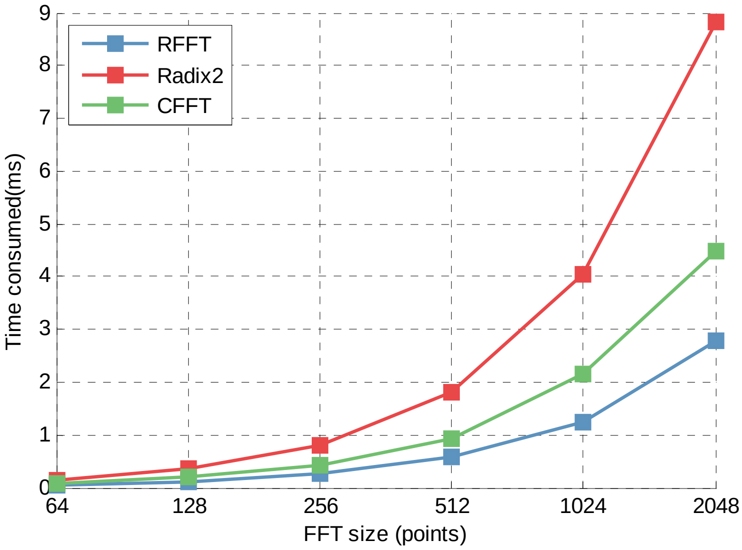


Figure 5.11 FFT computation speed comparison for *arm_cfft_f32* (CFFT), *arm_cfft_radix2_f32* (Radix2) and *arm_rfft_fast_f32* (RFFT)

For Hilbert transform, it involves one forward and one inverse FFT calculation. For the real-valued FFT calculation, its spectrum is symmetric and the function *arm_rfft_fast_f32* is designed for this structure. However, the analytic signal does not have this property anymore, which means the real-valued FFT calculation is not suitable for the inverse FFT calculation in Hilbert transform. Thus, the Hilbert transform calculation employed a combined version with forward FFT using *arm_rfft_fast_f32* and inverse FFT using *arm_cfft_f32.*

A computation speed comparison between a pure CFFT and a combined version of RFFT + CFFT is presented in Figure 5.12. An obvious speed improvement can be noticed in the combined version, especially when the FFT calculation size is large. It shows the combined version has a time reduction of approximately 20 % in comparison with that of the pure CFFT version.

After the inverse FFT operation, the analytic signal is obtained and the envelope can be acquired by computing the magnitude of the analytic signal. Such a calculation can be finished by the function *arm_cmplx_mag_f32*. For this function, the input and output buffer can be reused. Finally, the total time consumption for calculating the envelope using RFFT + CFFT method is presented in Figure 5.13.

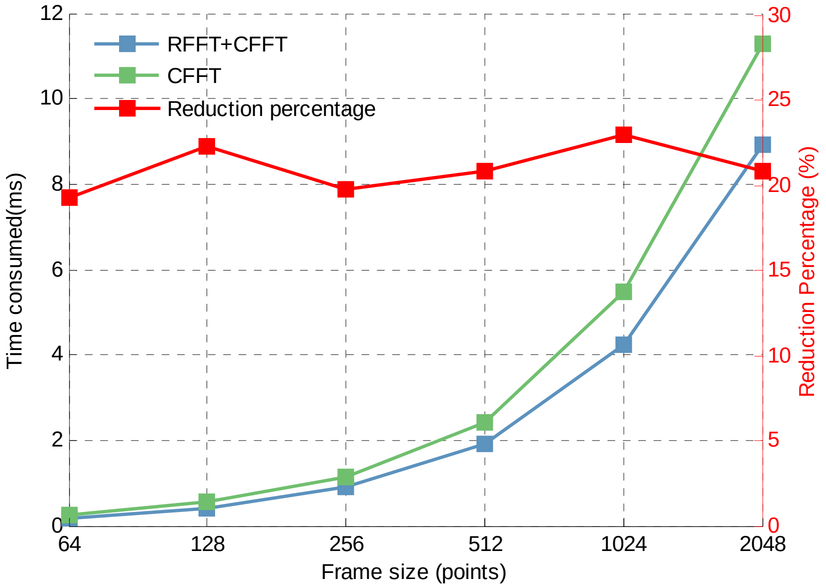


Figure 5.12 Hilbert transform computation speed comparison between pure CFFT and a combined version of RFFT + CFFT

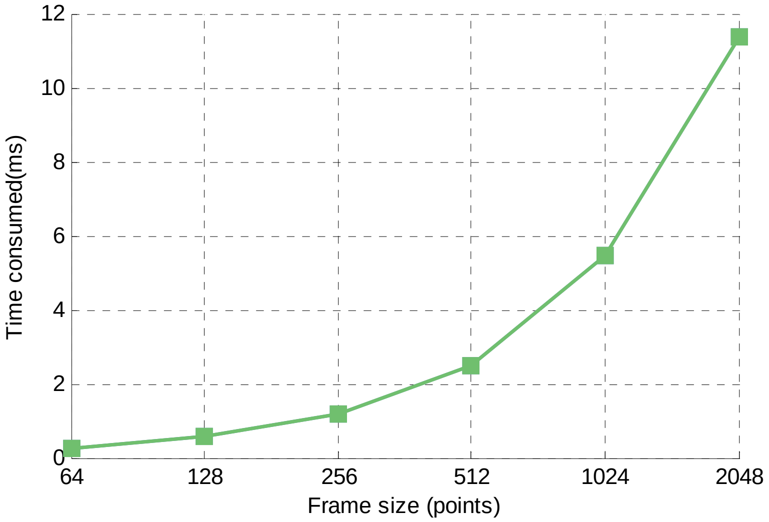


Figure 5.13 Total time consumption for computing envelope using RFFT + CFFT

5.2.2.2 Verification

To verify the implementation of envelope detection, a modulating signal is stored in the memory, processed on the MCU and exported in the debug mode for visualisation. The calculation of Hilbert transform and envelope are verified separately.

The modulating signal and its calculated Hilbert transform on the MCU are compared in Figure 5.14. In the time domain, the Hilbert transform signal shows a fixed delay compared to the raw carrier signal. In the frequency domain, they show the same amplitude while there has a fixed 90° phase difference for the majority of frequency components. This is just the feature of Hilbert transform should look like.

To verify the preciseness of the calculation on the MCU, the Hilbert transform of the same modulating signal is calculated in Matlab and considered as the baseline signal. The comparison between MCU computed results and baseline are shown Figure 5.15(a), which shows a perfect match. This can also be verified by the scatter plot in Figure 5.15(b), where the MCU results fit quite well with the 1:1 line of baseline.

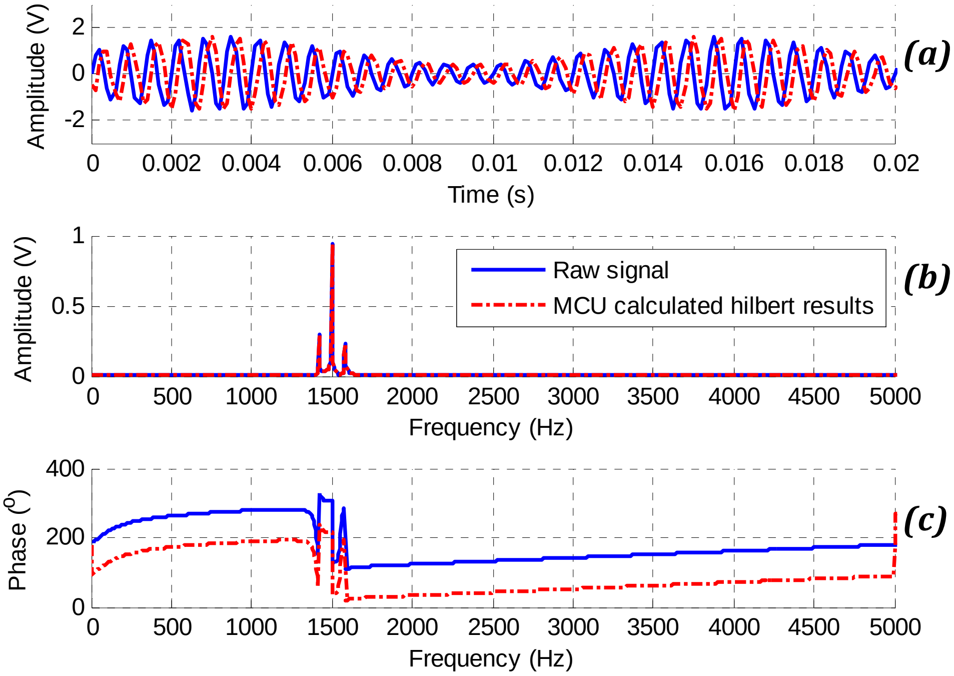


Figure 5.14 Raw signal and MCU calculated Hilbert transform results in comparison: (a) time domain (b) spectrum and (c) phase

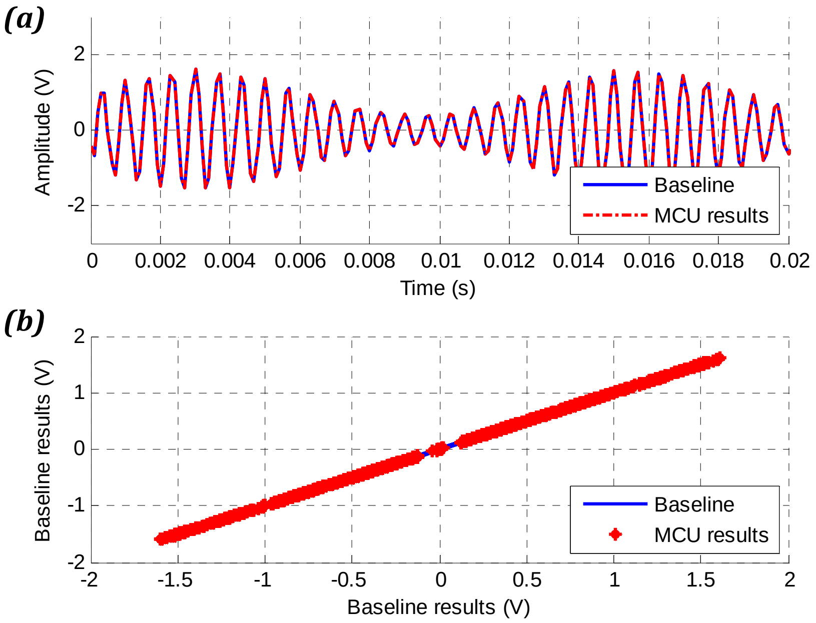


Figure 5.15 (a) Baseline and MCU calculated Hilbert transform results and (b) scatter plot between baseline and MCU results with solid line being 1:1 line of baseline results

The calculated envelope of the modulating signal is shown in Figure 5.16. Through visual observation, the calculated envelope fits well with peaks of the modulating signal. Similar as the preceding verification of Hilbert transform calculation, the envelope computed in Matlab is considered as the baseline signal. A comparison plot between the MCU calculated envelope and the baseline is presented in Figure 5.17(a) and a scatter plot in Figure 5.15(b). Both comparison results show a good match between the MCU result and the baseline one. This verifies the envelope calculation implementation on the MCU.

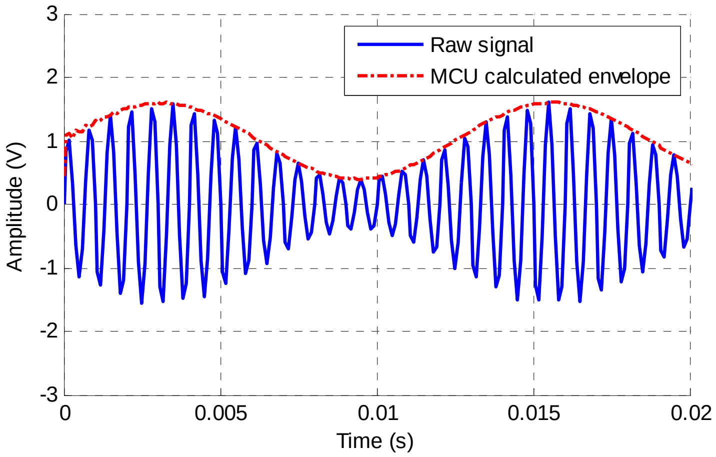


Figure 5.16 Raw signal and MCU calculated envelope

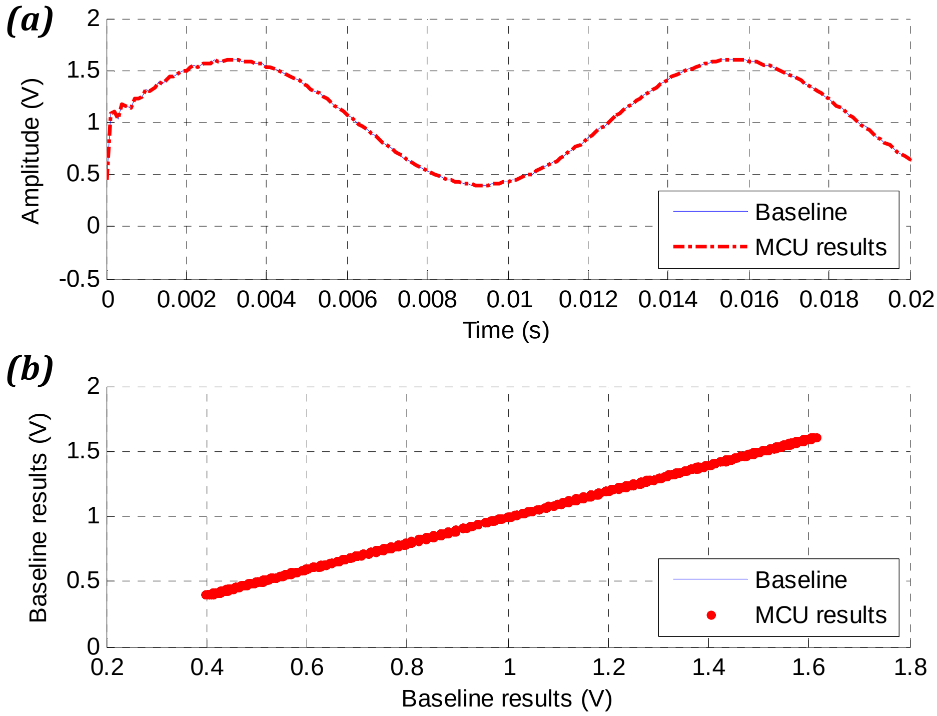


Figure 5.17 (a) Baseline and MCU calculated envelope and (b) scatter plot between baseline and MCU calculated envelope with solid line being 1:1 line of baseline results

5.2.3 Spectrum calculation

5.2.3.1 Implementation and spectrum leakage consideration

The spectrum calculation involves one forward real-valued FFT and one magnitude calculation. By consulting the discussion in Section 5.2.2, the real-valued FFT function *arm_rfft_fast_f32* is used for FFT calculation. For the amplitude calculation, the function *arm_cmplx_mag_f32* is employed.

Special attention needs to be paid for the spectrum calculation is the spectrum leakage, which is unavoidable for the FFT calculation in practice. One approach of reducing leakage is to increase the length of data for FFT calculation. For the bearing vibration signal, with the sampling rate of 10 kHz, the periods of fault signals contained in one maximum FFT computation frame, i.e. 2048 points, is given in Table 5.2. It can be observed that less than two periods of the FTF signal is enclosed in one frame, in which case the spectrum calculation result would be untrustworthy. For the other three signals, the spectrum leakage problem is relatively smaller. However, it is still suggested to apply a Hanning window to suppress the spectrum leakages.

Table 5.2 Periods contain in one frame

Fault type	Fault characteristic frequency (Hz)	Periods in one frame (2048 points)
BPFI	135.5	27.7
BPFO	83.5	17.1
BSF	48.4	9.9
FTF	9.3	1.9

Moreover, the obtained envelope signal contains a direct current (DC) offset and should be removed before FFT calculation. Otherwise, the large DC frequency components will also leak to the nearby low-frequency components, making their amplitudes incorrect.

The time consumption for spectrum calculation for different frame sizes is presented in Figure 5.18.

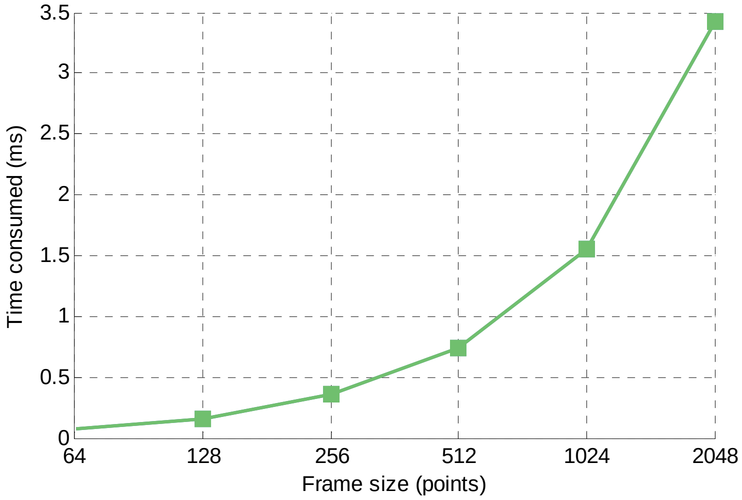


Figure 5.18 Time consumption for spectrum calculation

5.2.3.2 Verification

To verify the implementation of spectrum calculation, the same modulating signal in Section 5.2.2.2 is employed as the sample signal and its spectrum is calculated on the MCU and visualised in Figure 5.19. From the magnified spectrum, the two side bands of the modulating signal can be clearly observed. The spectrum is in accordance with the modulating signal, whose carrier frequency is 1500 Hz and modulation signal is 80 Hz.

For computation preciseness verification, the MCU spectrum is compared with the baseline result from Matlab calculation and shown in Figure 5.20(a), which shows a perfect match. The scatter plot in Figure 5.20(b) also verifies the preciseness of the MCU calculated spectrum.

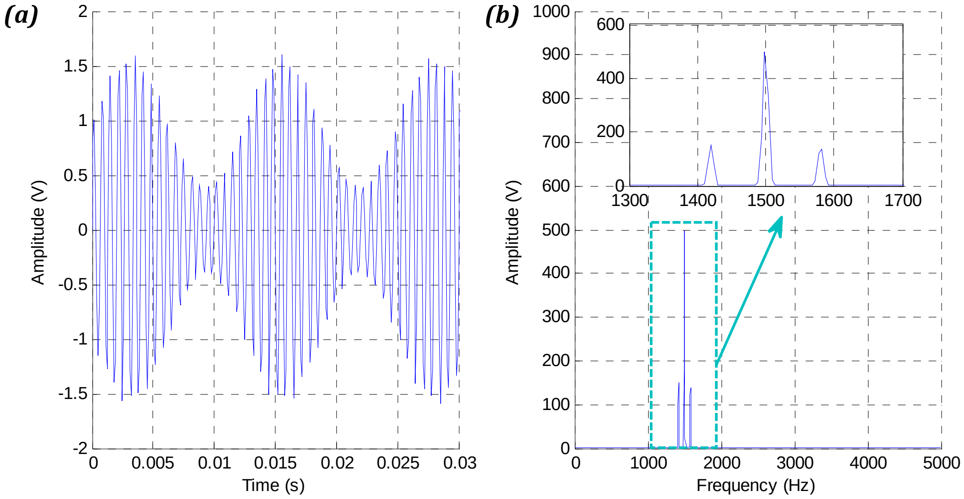


Figure 5.19 (a) Raw signal and (b) MCU calculated spectrum with magnified portion shown as an inset

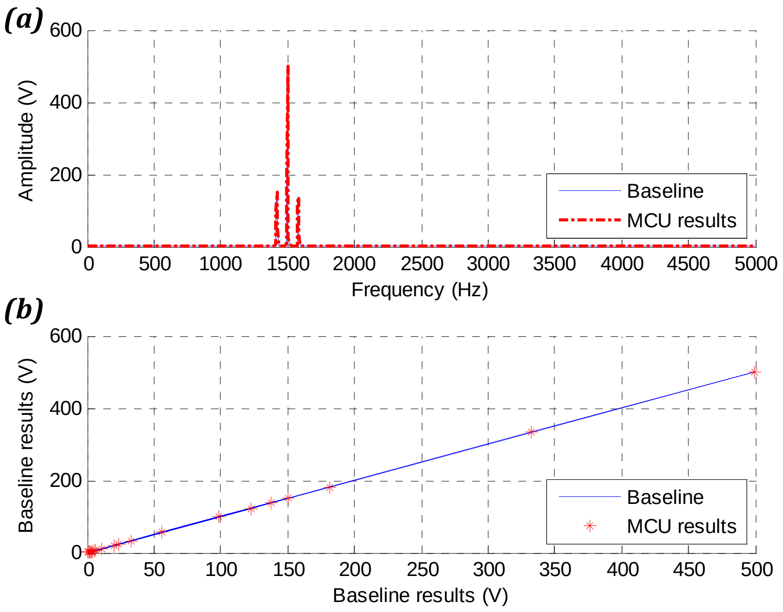


Figure 5.20 (a) Comparison between baseline and MCU calculated spectrum and (b) scatter plot between baseline and MCU calculated spectrum with solid line being 1:1 line of baseline results

5.2.3.3 Spectrum average

As the bearing's state is relatively stable in a short running period, averaging can be employed to suppress random noises and enhance the true vibration signals that are related to fault information [184]. The averaging can be implemented either in the time domain or in the frequency domain. For the time domain average, the synchronisation method is called TSA as introduced in Section 4.2.3. This operation typically requires an additional synchronisation signal from an encoder, which increases both the system complexity and cost. For the frequency domain average, it just needs the vibration signal itself. Therefore, the frequency domain average method is employed for improving the SNR of the spectrum.

For a series of FFT results Y_m, their averaged results $\bar{Y}$ can be obtained by:

$$\bar{Y}(n) = \frac{1}{M} \times \sum_{m=1}^{M} Y_m(n)\, , n = 0, 1, 2, \dots\ N - 1 \tag{5-7}$$

where N is the size of FFT and M is the number of FFT calculations.

For an FFT average operation with M frames, the output noise variance is reduced to $1/M$ of that in a single FFT result [178], resulting a higher sensitivity in the averaged FFT results. In addition, the averaging process not only brings the benefits of better SNR of the envelope spectrum but also reduces the redundancy in the transmission data set and thus saves the occupation of the valuable bandwidth.

5.3 Data transmission

After data processing, the obtained envelope spectrum needs to be transmitted to the host data processing centre for further analysis. Note that there is no need to send the entire envelope spectrum data via the wireless sensor network. Only the band that contains the first three harmonics of the fault characteristic frequencies is useful for diagnosis. Recall the discussion in Section 5.2.1, the first three harmonics of characteristic frequencies for the three common faults are well within 500 Hz. Thereafter, only the spectrum below 500 Hz needs to be transmitted over the network, i.e. 103 data points with the sampling frequency at 10 kHz. In addition, this frequency bandwidth can meet the needs for a wide range of machines driving with popular 4 pole-induction motors.

To help the host extract the effective data in the streaming data, the transmitted data need to be structured properly, as shown in Figure 5.21. The header can help the host identify the starting of the streaming data and the data type can be a raw signal, filtered signal, spectrum, envelope or the envelope spectrum, etc. During debugging stage, these middle processing results can also be transmitted to the host side for verification. As the

ZigBee network has included robust checksum and retransmission mechanism, here we just use the data length as a simple method for checking the integrity of current frame of data.

To reduce bandwidth occupation, the computed envelope spectrum data are converted back to 16-bit *unsigned integer* format. For easier coding and decoding, the data are currently coded in American standard code for information interchange (ASCII) during wireless transmission but can be transmitted as raw data or lossless compression like Huffman to further save bandwidth.

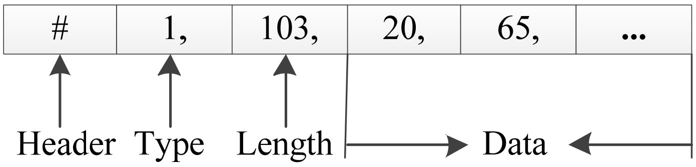


Figure 5.21 Structure of the transmission data

5.4 Performance analysis

5.4.1 Processing of a bearing fault signal

To validate the performance of the implemented envelope analysis for extracting bearing fault features, a typical vibration signal from a bearing with outer race fault is measured and processed on the proposed wireless sensor node. The middle processing results on the MCU are observed in the debug mode by setting breakpoints at the proper position and viewed using the graph tool and FFT magnitude graph tool provided in CCS environment [185]. The appearance of the visualisation tool is not neat but very helpful in the debugging process.

The processing results of the roller bearing on the sensor node are illustrated in Figure 5.22, which includes the raw signal, filtered signal and analysed envelope. Due to function limitations of the graph tool in CCS, the unit of *x* axis is given in samples, which is the index of the corresponding signal array. As the sampling rate of the ADC is at 10 kHz and the size of the FFT frame is 2048 points, the resolution of the frequency spectrum is about 4.9 Hz per bin.

As shown in Figure 5.22 (a), a direct current (DC) offset exists in the raw vibration signal, and periodical spikes are observable which are caused by the defect on the outer race. From its spectrum, it can be seen that the signal spans a wide frequency range and has many discrete components, making it difficult to identify the fault types. Hence, the DC offset is removed from the raw signal in order to highlight the alternating current (AC) spectrum. A large frequency component appears around 1500 Hz, which should be one of the resonance signals of the bearing. Therefore, an 80-tap FIR type band-pass filter is applied on the raw signal to extract frequency components between 1 kHz and 2 kHz.

The filtered signal and its spectrum are shown in Figure 5.22 (b). The signal becomes much smoother in the time domain in comparison with the raw signal and only the band between 1 kHz and 2 kHz are kept in the frequency domain. The analysed envelope and its spectrum are presented in Figure 5.22 (c). The envelope roughly matches the outline of the filtered signal. The three low frequency components can be clearly observed as distinctive peaks in the spectrum, also any frequency components above 500 Hz are significantly attenuated.

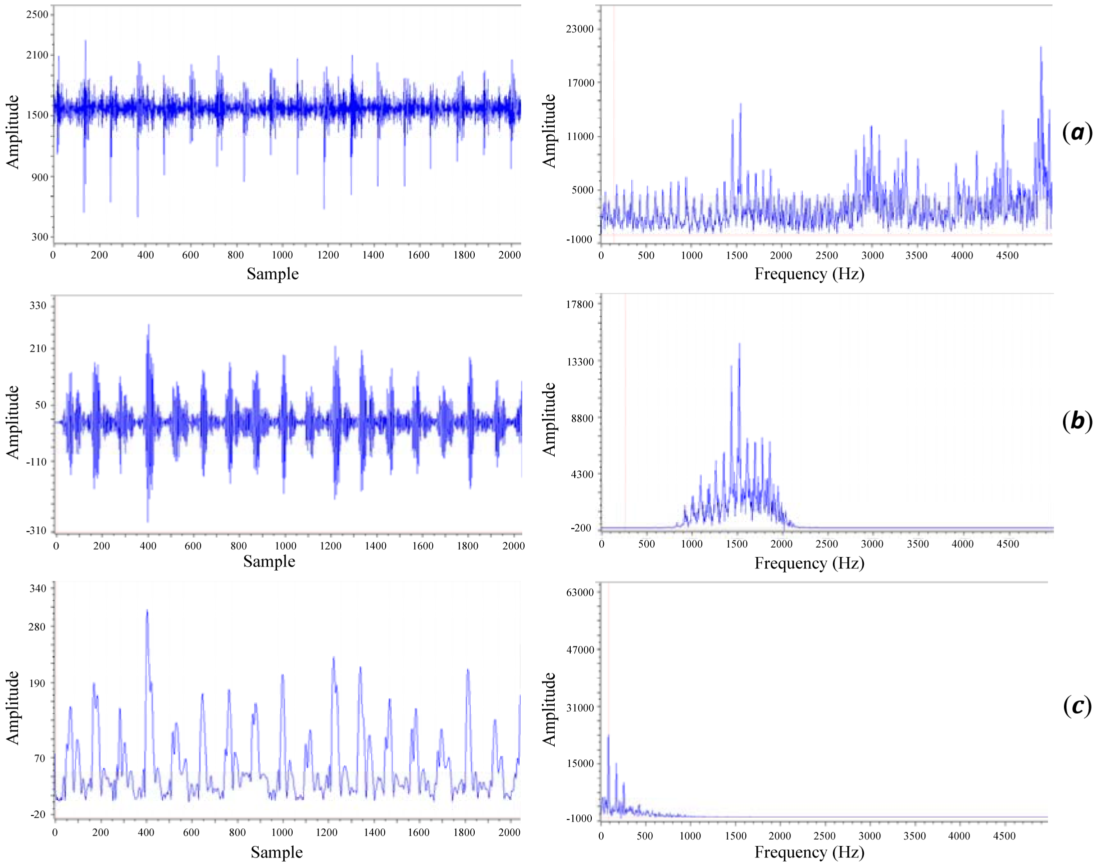


Figure 5.22 Bearing signal and spectrum: (a) measured signal, (b) band-pass filtered signal and (c) envelope signal

Figure 5.23 shows the envelope spectrum magnified at the low frequency range. Three distinctive spectral peaks can be observed at sample point 18, 36 and 54, whose corresponding frequencies are 87.9 Hz, 170.9 Hz and 258.8 Hz, respectively. Recall the fault frequencies listed in Table 4.2, these frequencies agrees with the first three harmonics of the outer race fault frequency. Therefore, the spectrum feature can verify the existence of an outer race fault on the roller bearing.

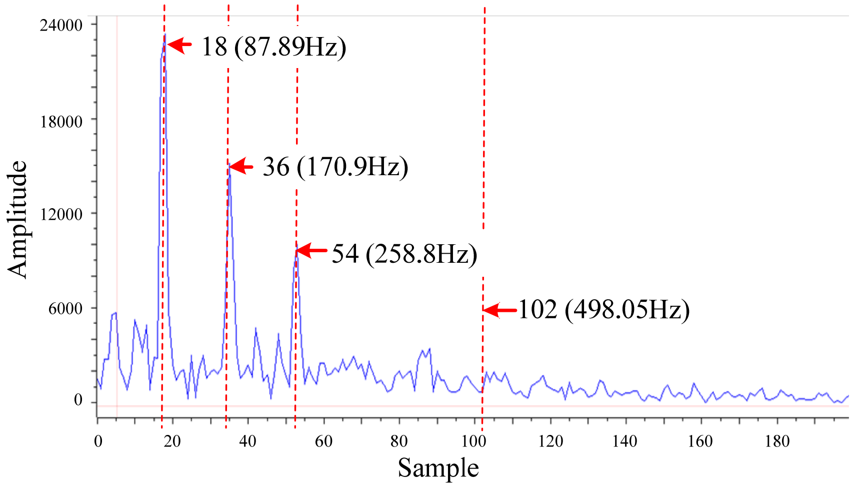


Figure 5.23 Envelope spectrum of roller bearing with area zoom in

In addition, the spectrum above 500 Hz (equivalent data sample is 102) is nearly flat due to the effect of the band-pass filter. Therefore, only data in the low frequency range (less than 103 points) needs to be averaged and transmitted to the remote host computer. Figure 5.24 shows the roller bearing remote display results obtained with four times of average. As it shows, the spectral peaks of interest become more distinctive and the background random components are suppressed effectively by the averaging process compared with those in Figure 5.23.

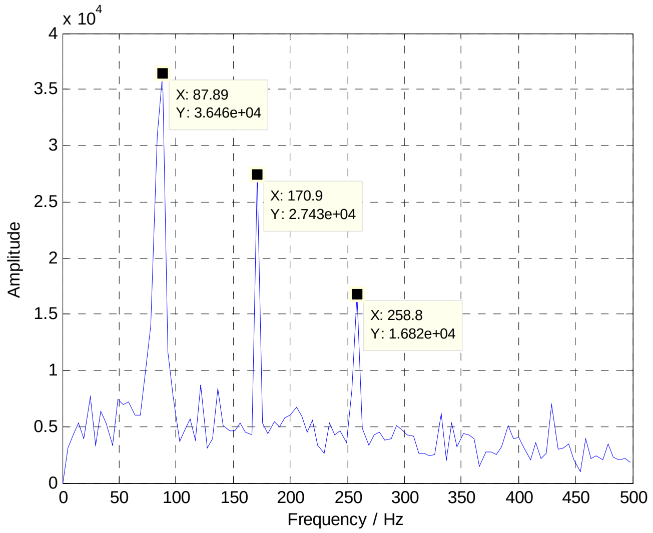


Figure 5.24 Bearing remote display results averaged by 4 times

5.4.2 Time consumption and data throughput analysis

In the last section, the envelope analysis includes an 80-tap FIR filter, 2048-point Hilbert transform, and spectrum calculation. The total time consumption is measured as 28.4 ms for the MCU running at 80 MHz. Recall the sampling frequency is 10 kHz and one frame of data includes 2048 points of data, this allows the time for data processing to be about 204.8 ms. This indicates the MCU can decrease running speed to save power or increase the sampling frequency to allow a wider frequency band to be analysed.

For data throughput, only the size of effective data for transmission is counted and the extra data cost due to packaging is taken into consideration as this is not the focus of this thesis. A comparison of data amount that needs to be transmitted in different processing stages is presented in Figure 5.25. Provided the data are stored with 16-bit resolution, the raw data per frame (2048 points) occupy as much as 4096 bytes, and the envelope spectrum needs to send half of that since the spectrum is symmetric, a reduction of 50%. With the application of the band-pass filter in the envelope analysis, only 103 points of spectrum data (206 bytes) need to be transmitted, contributing a significant reduction of nearly 95%. In this case, the data output rate is reduced to approximately 8 kbps. Furthermore, if the resultant envelope spectrum is averaged by four times, the output rate will be reduced to only 2 kbps, which will be a much lighter load for the ZigBee network and the real-time transmission could be possibly achieved.

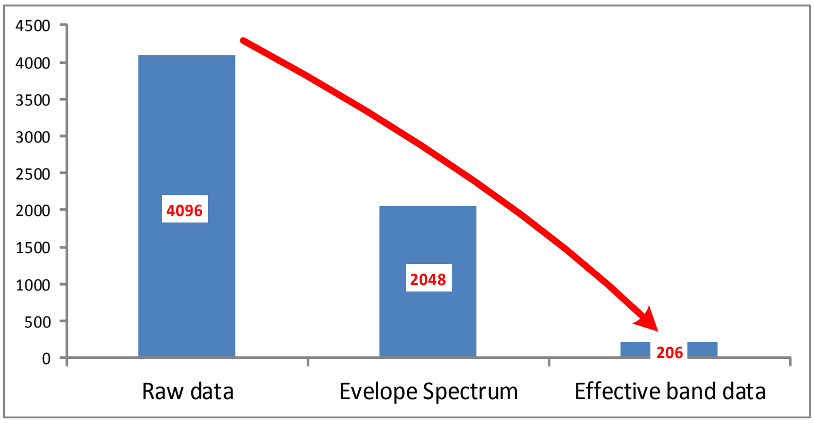


Figure 5.25 Comparison of the data amount

5.5 Summary

In this chapter, an envelope analysis with a frame length of 2048 points is successfully implemented on the designed wireless sensor node. The implementation process is explained in detail and validated carefully. From the experimental analysis, it can be seen that the computed envelope spectrum under a sampling frequency of 10 kHz is capable of expressing bearing fault features and data transmission requirement is reduced by nearly 95% in comparison with the raw data set. This shows envelope analysis is a good candidate algorithm for being embedded on the wireless sensor nodes for bearing fault diagnosis and hence will be explored in more detail to improve its accuracy, robustness and computing efficiency in later chapters.

CHAPTER 6

IMPROVEMENT OF THE FREQUENCY RESOLUTION OF ENVELOPE SPECTRUM

In the last chapter, the envelope analysis is implemented on the proposed wireless sensor node and is proved to be effective for both extracting bearing fault features and reducing data transmission requirements. In this chapter, a novel signal processing procedure is developed to achieve the finest frequency resolution of envelope spectrum under constraints of limited computing and memory resources. The implementation of this proposed method on the embedded processor is explained in detail and the processing results are compared with that of the previous one. By using this method, a more accurate fault diagnosis can be achieved.

6.1 Theoretical background

6.1.1 Frequency resolution analysis

For a discrete signal, its frequency resolution Δf is determined as:

$$\Delta f = \frac{1}{T} = \frac{F_s}{N} \tag{6.1}$$

where T is the time window length for FFT analysis and N is the number of samples inside the time window length T under sampling rate F_s.

Apparently, to increase the frequency resolution (i.e. reduce Δf), the time window length T for FFT analysis must be increased. This can be achieved by either reducing F_s or increasing the number of sampling points N. Because the sampling frequency is usually fixed and restricted by the frequency range in the bearing vibration signal, in practice, the frequency resolution is dependent primarily on the number of points N.

Recall the analysis in the last chapter, due to memory limitations, the maximum size for FFT calculation in *single floating point* format is limited to 2048 points. In other words, the maximum frequency resolution achievable is about 10000/2048≈4.88 Hz, which is just less than half of the FTF fault characteristic frequency. This means it would be very difficult to distinguish FTF and its harmonics.

Apart from the FTF fault characteristic frequency, the harmonics of the fault characteristic frequencies in Table 4.2 are shown in Figure 6.1 (a). It can be seen that several frequency components are rather close to each other. From Figure 6.1 (b) which shows the frequency difference between two successive frequency components in Figure 6.1 (a), it can be seen the minimal frequency difference is less than 10 Hz, which is corresponding to one or two frequency bins as shown by the right Y-axis in Figure 6.1 (b). This indicates that should there be a slight slip during bearing running, these frequency components would be very likely mixed together. Thereafter, the frequency resolution needs to be increased for more accurate fault diagnosis purposes.

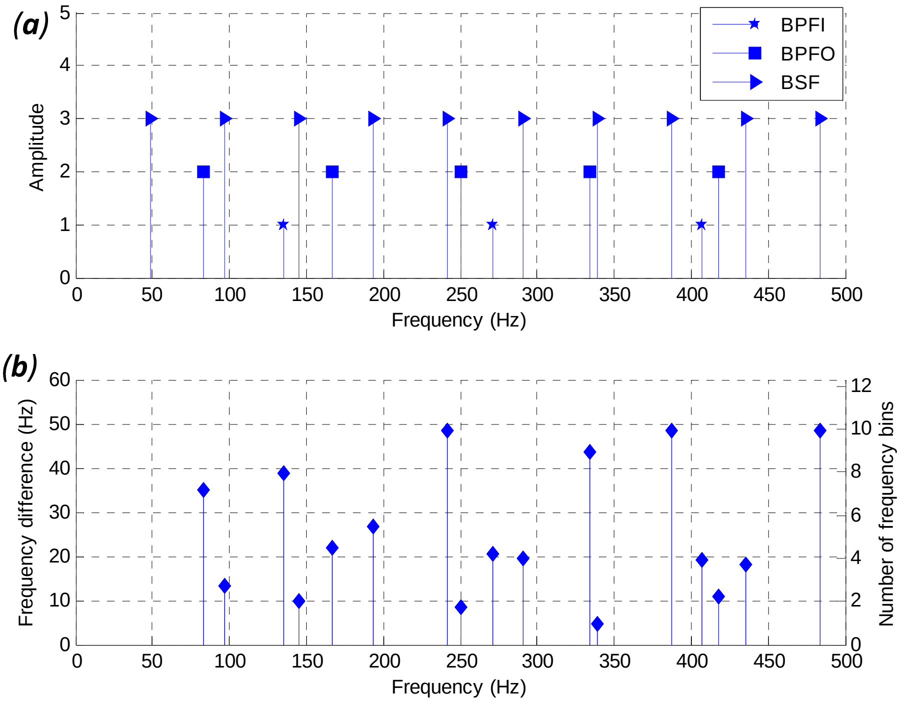


Figure 6.1 (a) Harmonics of fault characteristic frequencies and (b) frequency difference between two successive frequencies

6.1.2 Down-sampling, overlap processing and cascading

Although the frequency resolution in the spectrum of the original signal is difficult to be improved on current wireless sensor node, it is possible to increase the frequency resolution of the spectrum of the envelope. Figure 6.2 shows the envelope analysis to a simple modulation signal, in which, a 1500 Hz carrier signal is modulated by an 80 Hz sine wave. In the spectrum of the modulation signal, three frequency components are observable and the 80 Hz frequency component is carried to the high-frequency band centred at the carrier frequency. After envelope analysis, the low-frequency component is extracted out and the high-frequency problem is transferred to a low-frequency one.

In comparison, the sampling frequency should be higher than 3160 Hz according to Shannon-Nyquist sampling theorem whilst the sampling frequency of the extracted envelope does not need to be so high. After envelope analysis, the interested frequency components of the envelope signal are shifted to the low-frequency band. Thus, it is possible to improve the frequency resolution of envelope spectrum by down-sampling and then cascading the computed envelope data.

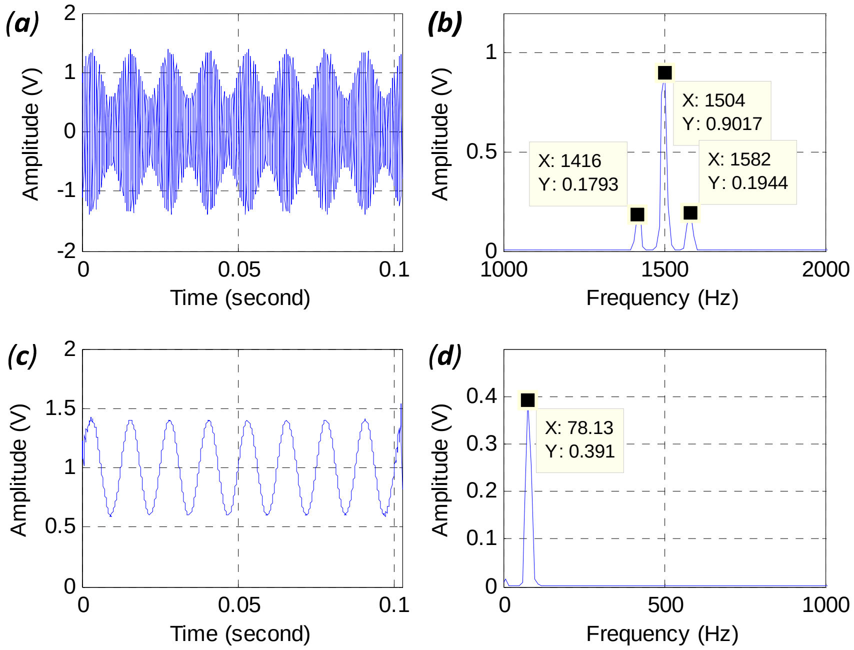


Figure 6.2 Application of envelope analysis to a time waveform: (a) modulation signal, (b) spectrum of modulation signal, (c) analysed envelope and (d) envelope spectrum

Meanwhile, it cannot simply concatenate the down-sample envelope signal. As shown in the analysed envelope in Figure 6.2(c), obvious distortions appear at the front and end edge of the analysed envelope signal. These distortions are called edge distortions and are caused by the window effect when implementing Hilbert transform [145]. This means simply cascading the down-sampled signal may bring undesirable distortions to the spectrum. To exclude these unwanted distortions, an overlap processing scheme can be employed according to [186].

On this basis, a processing scheme that combines the overlap processing, down-sampling and cascading is proposed and illustrated in Figure 6.3. The collected vibration data are processed frame by frame with the size of 512 points and a 50% overlap is employed to keep the consecutiveness of the analysed envelope signal. As shown in the figure, the scheme consists of two main steps: envelope calculation and envelope down-sampling. In the first step, a data frame first passes through an FIR band-pass filter and then combines with the last filtered data frame to form a new data frame of 1024 data points. Then the envelope of this 1024 data is computed using Hilbert transform, as described in Section 5.2.2.

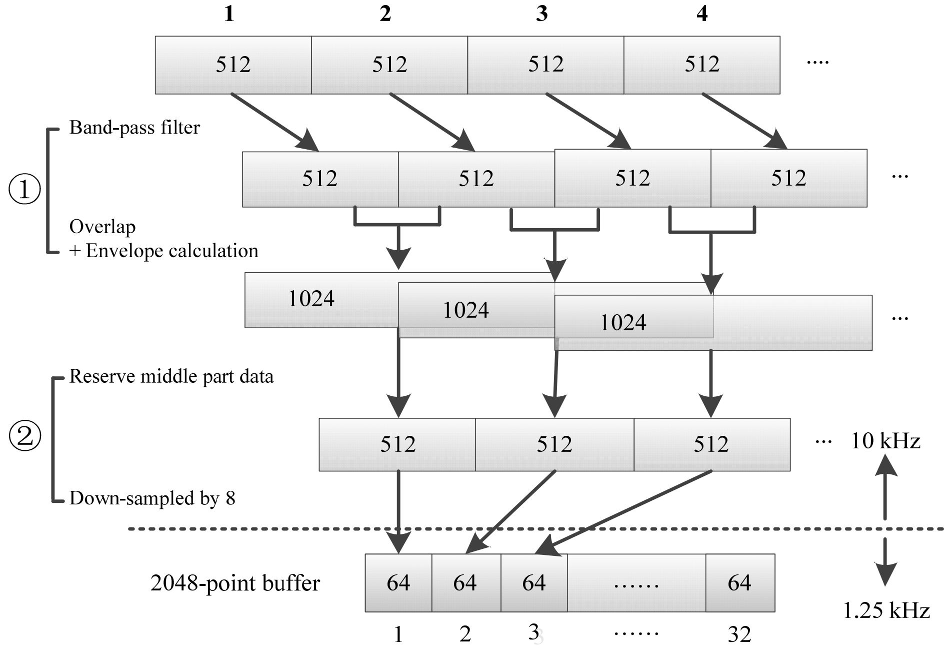


Figure 6.3 Diagram for the overlap, down-sampling and cascading scheme

In the second step, the middle half of the computed envelope data is kept and down-sampled. The down-sampling rate is determined by the interested frequency band of the envelope signal. Recall the discussion in Section 5.3, the interested frequency band of the envelope signal is within 500 Hz. This indicates the sampling frequency of the envelope signal should be higher than 1 kHz and hence a down-sampling rate of 8 is employed, i.e. the sampling frequency of the envelope is 1.25 kHz. To avoid aliasing, a low-pass FIR filter with cut-off frequency at 1 kHz is applied to filter the envelope signal before down-sampling. As shown in the fourth and the fifth row of the data flow diagram in Figure 6.3, the original 512 points of data shrink to only 64 points.

The repeat of the above two steps allows an envelope of 2048 points to be obtained. As the construction of the envelope data is based on the middle part of each frame, the edge distortions are avoided. By computing FFT of the down-sampled envelope data set, a spectrum with improved frequency resolution can be obtained. Although the size of FFT calculation stays the same, the above procedure actually increased the time window length for FFT calculation to improve the frequency resolution. For this application, the sampling rate for the down-sampled envelope becomes 1250 Hz, therefore, the resolution of the envelope spectrum is about 1250/2048 ≈ 0.61 Hz. In other words, the frequency resolution of the envelope spectrum is effectively increased by 8 times in comparison with that in Chapter 5.

6.2 Algorithm implementation

With an observation of the diagram in Figure 6.3, it can be seen that the proposed algorithm has the following two important features:

- The long raw signal is continuously collected but should be divided into small frames of 512 points for processing due to memory size limitations;
- The buffer for envelope calculation is reused and hence the envelope calculation should be finished by the next frame of data ready.

These two features bring real-time signal processing requirement for the envelope calculations. The *Ping-Pong* structure discussed in Section 5.1.1 can take its advantage to help implement this algorithm.

6.2.1 Overall data flow

By utilising the *Ping-Pong* structure, data processing flow inside the processor is presented in Figure 6.4. The vibration signal is firstly sampled by the on-chip ADC, which is triggered by a timer with overflow rate of 10 kHz. Then, converted results are moved from ADC register to internal buffer *Ping* or *Pong* alternatively through a DMA unit. When a buffer is full, an interrupt is generated to inform the CPU to process the newly collected frame data (512 points) and DMA will move new data to the other buffer.

For accuracy considerations, the collected data of 16-bit *unsigned integer* format is converted to 32-bit *single floating* format. The overlap, down-sampling and cascading scheme described in Section 6.1.2 is performed on the upper half of buffer *fBuf* and the buffer *lastFrame* is used to temporarily store the band-pass filtered results. The down-sampled envelope is stored in the lower half of buffer *fBuf* and the FFT of 2048-points is performed on the entire buffer *fBuf* to get the spectrum of the envelope. Finally, the computed envelope spectrum is averaged by four times in the buffer *envOut* and then sent to the ZigBee network through UART.

From Figure 6.4, it can be seen that the memory consumption for the computation buffer is up to 24 kB, which is more than half the size of the total random access memory (RAM), i.e. 32 kB. These buffers are used for high efficiency, especially for the main buffer *fBuf*, which is used for several calculations, such as envelope computation, down-sampling, cascading and FFT.

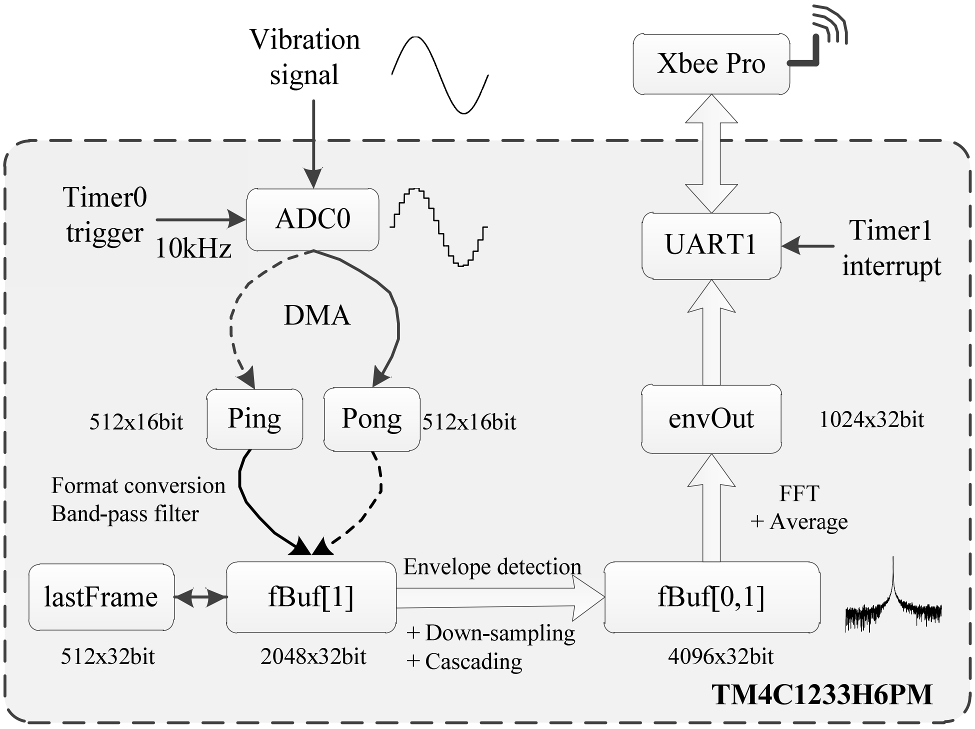


Figure 6.4 Data flow inside the processor

6.2.2 Flowchart of the signal processing

To explain the implementation of the signal processing algorithm more clearly, the flow charts for the overall signal processing and the critical sub-function for envelope detection and down-sampling are illustrated in Figure 6.5(a) and (b), separately.

Apparently, the sub-function for envelope detection and down-sampling needs real-time performance, in other words, the calculation of this function should be finished before a deadline, otherwise, conflicts between data collection and data processing will occur and thus the calculated envelope spectrum will be incorrect. The deadline for this function is the time for collecting 512 points of data, i.e. 512/10000=51.2 ms.

To measure the computation time for the signal processing, two GPIO signals are set in the program and measured through a data acquisition system, as shown in Figure 6.6. The blue signal is set high when entering the DMA ISR and set low when exiting the ISR, thus, it indicates frame data ready and its interval is the deadline for frame data processing, which can be measured as 51.175 ms. The red signal is set high when starting frame signal processing and set low when the processing finishes, hence, the time length for the red signal being high is the signal processing time, which can be measured as 10.875 ms. This is smaller than the deadline requirement, i.e. the real-time computing requirement is satisfied.

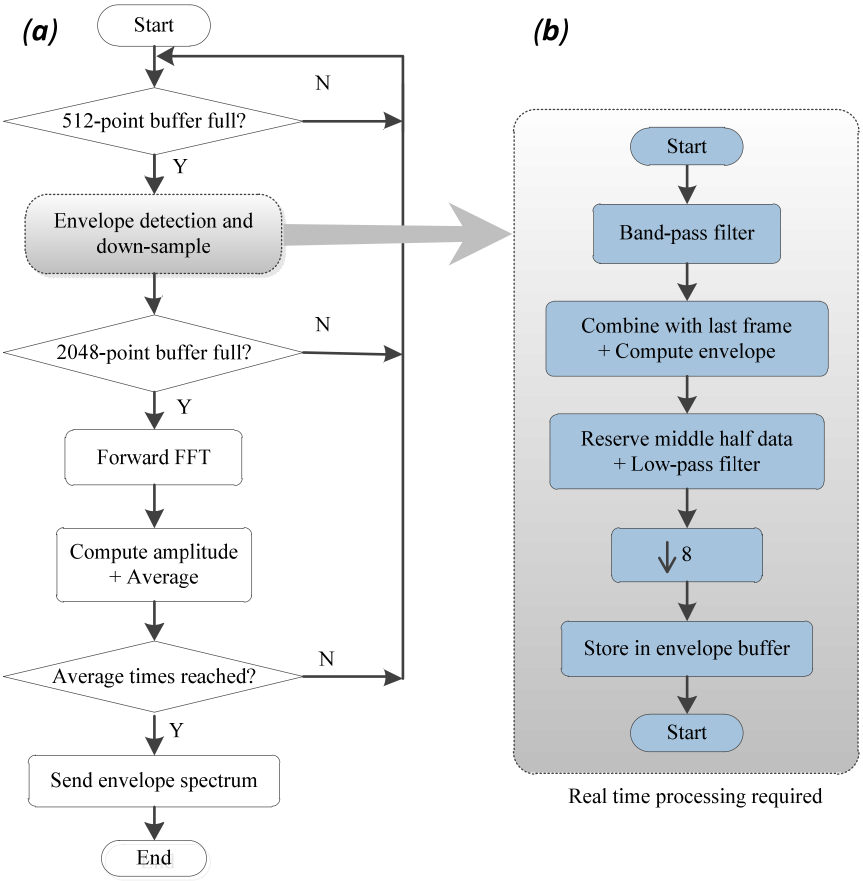

(a)
Start
512-point buffer full?
N
Y
Envelope detection and down-sample
2048-point buffer full?
N
Y
Forward FFT
Compute amplitude + Average
Average times reached?
N
Y
Send envelope spectrum
End
(b)
Start
Band-pass filter
Combine with last frame + Compute envelope
Reserve middle half data + Low-pass filter
↓ 8
Store in envelope buffer
Start
Real time processing required

Figure 6.5 (a) Flowchart of overall signal processing program and (b) flow chart of the sub-function for envelope detection and down-sample

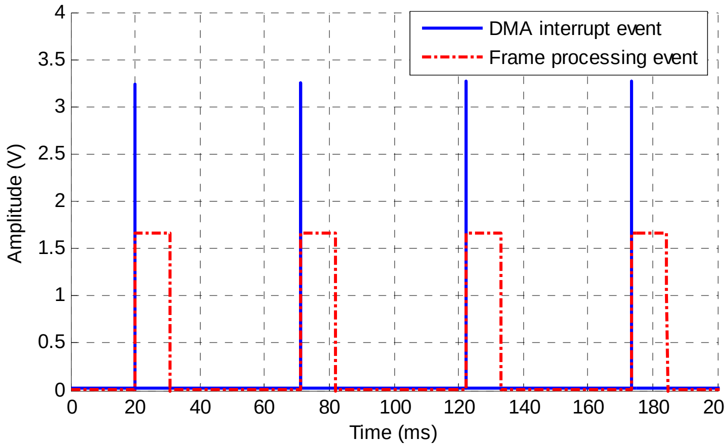


Figure 6.6 DMA interrupt and frame processing event measured through GPIO

6.2.3 Verification on a simulated signal

To verify the correctness of the implementation of the proposed algorithm, a simulated modulating signal is generated and processed on the MCU.

$$x(t) = (1 + 0.6 \times \sin(2\pi \times 80t)) \times \sin(2\pi \times 1500t) \tag{6.2}$$

By setting breakpoints on relevant program lines in the debug mode, the data is exported for visualisation. The key signals for the first frame of envelope signal processing are presented in Figure 6.7, including the raw signal, band-pass filtered signal, calculated envelope, low-pass filtered envelope and down-sampled envelope. It can be seen that the first frame of the envelope calculation (1024 points) processes the first and second frame of raw data (512 points) and only the middle half part of the envelope signal is kept in the down-sampled envelope signal. Moreover, a delay exists in the band-pass filtered signal and the low-pass filtered envelope. This is caused by the group delay discussed in Section 5.2. For this application, both the band-pass filter and the low-pass filter has 81-tap, thus, they caused 40 points of delay separately.

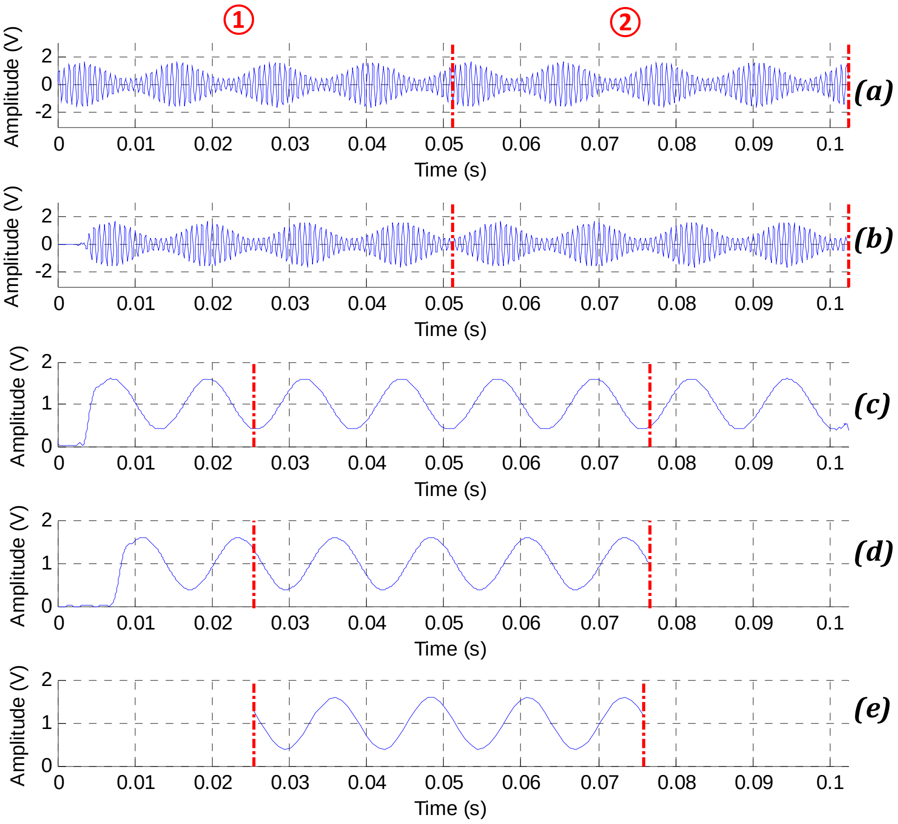


Figure 6.7 First frame of envelope data processing: (a) raw signal, (b) band-pass filtered signal, (c) envelope signal, (d) low-pass filtered envelope and (e) down-sampled envelope

Similarly, the key signals for the second frame of envelope signal processing are presented in Figure 6.8, which processes the second and third frame of the raw signal. After processing, it also only keeps the middle half of the calculated envelope. Of note, the filtering of the first frame envelope starts from the first point of the calculated envelope whilst that of the second frame envelope starts from the 257th point of the calculated envelope. This operation for the first frame envelope is to avoid the incorrectness of the beginning of the filtered signal caused by group delay.

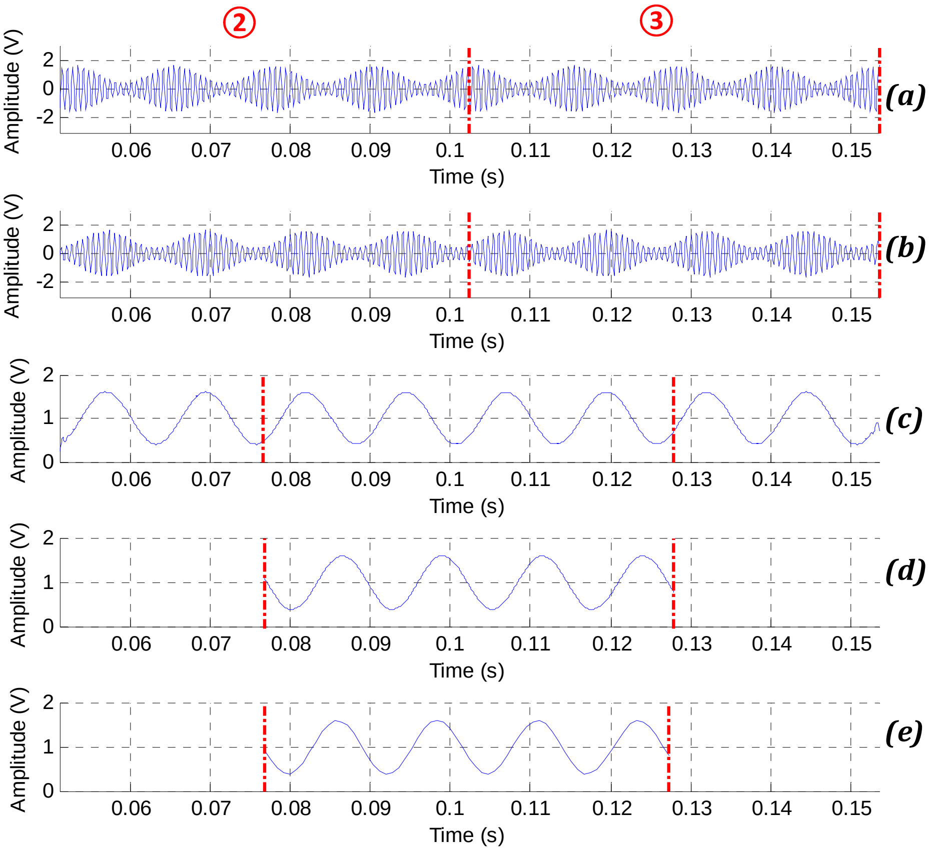


Figure 6.8 Second frame of envelope data processing: (a) raw signal, (b) band-pass filtered signal, (c) envelope signal, (d) low-pass filtered envelope and (e) down-sampled envelope

The cascaded envelope of the first and second frame envelope calculation results are illustrated in Figure 6.9. It can be seen the joining between this two frame envelops are continuous and smooth, which verifies the correctness of the implementation. The raw signal is also presented in Figure 6.9 but with 80 points of delay compensated. It can be noticed that the calculated envelope fits well with the upper outline of the signal, which verifies the correctness of envelope calculation.

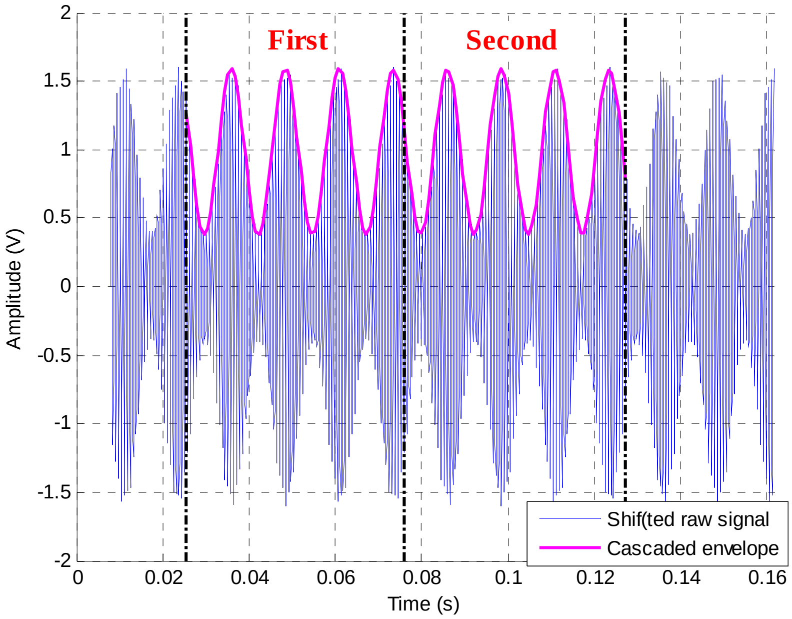


Figure 6.9 Cascaded envelope of first and second frame calculation results

6.3 Results and discussion

6.3.1 Processing results on bearing vibration signal

To further verify the performance of the proposed algorithm, a vibration signal from a bearing with outer race defect is processed on the wireless sensor node. Several important internal processing results are visualised via the graph tools in the debugging mode, as shown Figure 6.10. The processing results include raw signal, calculated envelope, down-sampled envelope and cascaded envelope.

Figure 6.10(a) shows one frame of the raw signal with 512 points of data, from which short period impact pulses can be observed whilst high-level noises are also noticeable. Figure 6.10(b) shows one frame of extracted envelope with 1024 points, which shows that high-frequency components have been removed whilst the period pulses are still clear. In the down-sampled envelope in Figure 6.10 (c), the pulses are still obvious while the down-sampled envelope only contains one-eighth of that in Figure 6.10 (b).

To avoid edge distortions, only the middle half of the down-sampled envelope is kept and cascaded to compose the longer envelope signal in Figure 6.10 (d), which shows quite a number of pulses.

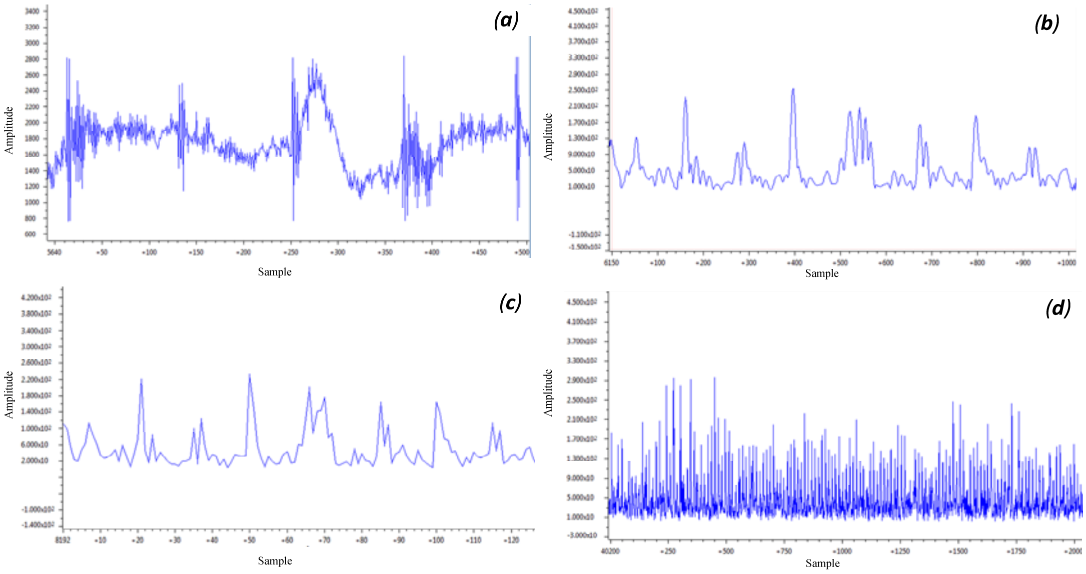


Figure 6.10 Bearing vibration signal processing results: (a) raw data (512 points), (b) envelope (1024 points), (c) down-sampled envelope (128 points) and (d) cascaded envelope (2048 points)

The spectrum of the cascaded envelope is presented in Figure 6.11. In the test, the sampling rate of the AD converter is set at 10 kHz and after 8 times' down-sampling, the true sampling rate of the cascaded envelope is reduced to 1250 Hz. Therefore, the frequency resolution after 2048 points' FFT calculation is about 1250/2048≈0.61 Hz. The corresponding frequencies for the peaks are illustrated in Figure 6.11 to get a better understanding of the spectrum. Recall the fault characteristic frequencies in Table 4.2, the peak frequencies in Figure 6.11 well matches the harmonics of the outer race fault frequency, which verifies the existence of the outer race fault on the rolling bearing.

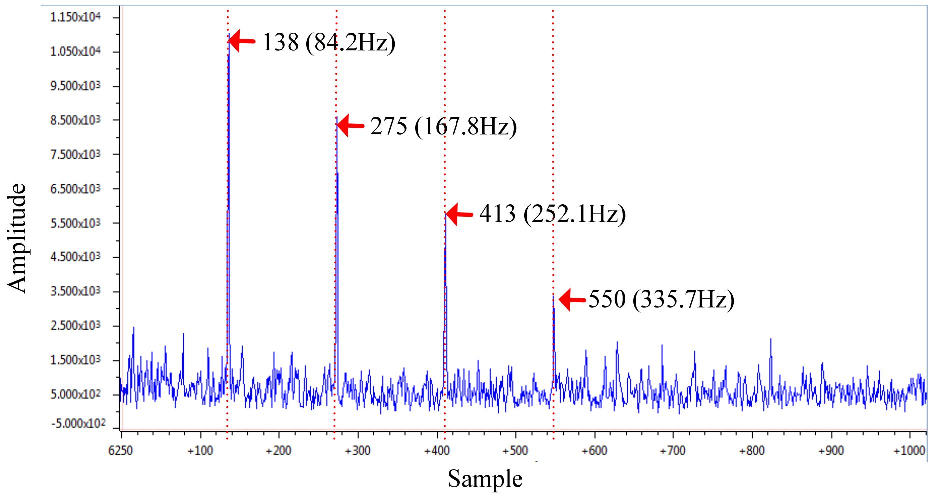


Figure 6.11 Spectrum of the cascaded envelope

To highlight the benefits of the improved algorithm, the normal envelope spectrum discussed in Chapter 5 is also employed to analyse the signal from the same defect bearing and shown in Figure 6.12 in comparison with the proposed method in this Chapter. Apparently, the outer race fault can be identified by both algorithms. However, the fault frequency identified from the improved algorithm is 84.23 Hz, which is much closer to the expected characteristic frequency, i.e. 84.5 Hz, whereas that from normal envelope algorithm is 83.01 Hz, more deviated from the expected one.

This shows that the improved algorithm produces more accurate and reliable diagnostic results. In addition, it can be seen that the ratio from the spectrum of the improved algorithm between the amplitude at the fault frequency and the background noise level is more than 6 while that from the normal envelope spectrum is less than 5. This means that the improved algorithm provides a higher SNR for better detection and diagnosis performance.

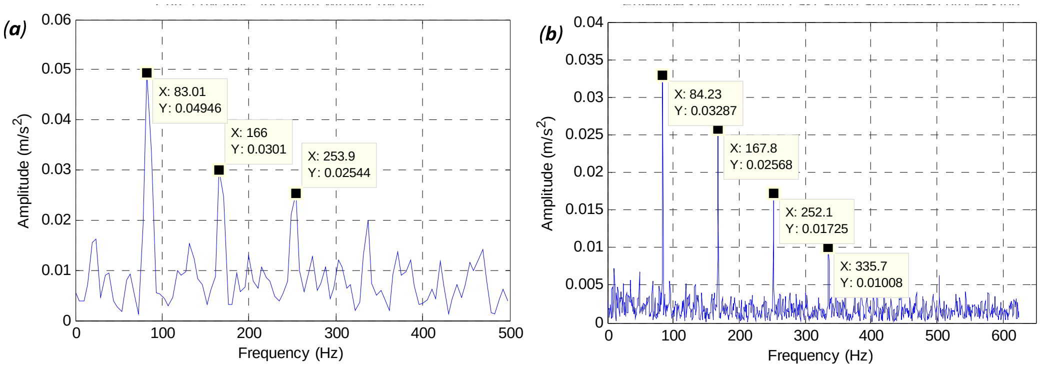


Figure 6.12 (a) Normal envelope spectrum and (b) envelope spectrum using proposed method

6.3.2 Data throughput requirement analysis

In the proposed method, much longer raw data (32*512=16384 points) are analysed than that in Chapter 5 (2048 points). By analysing the data with a longer time span, a higher frequency resolution is thereafter achieved in the resultant envelope spectrum. With an improved resolution, more data points in the envelope spectrum are required to be transmitted. The number of data points under 500 Hz can be calculated as 500/0.61≈820. In comparison with the raw data set, the reduction ratio in data size is still kept as 95%.

Table 6.1 summarises the effective data rate per second for different processing stages on the sensor node. All the data are supposed to be stored with 16-bit resolution. It can be seen the improved method has similar data throughput requirement as the former one obtained in Chapter 5. Furthermore, it should be noted that the improved envelope spectrum has higher SNR, which allows a higher compression ratio and hence a lower data throughput requirement to be achieved.

Table 6.1 Required effective data rate comparison

Data type	Data rate (kbps)
Raw data	160
FFT of raw data	80
Envelope analysis without cascading	8
Envelope analysis with cascading (entire spectrum)	9.7
Envelope analysis with cascading (within 500Hz)	8

6.4 Summary

In this chapter, an envelope analysis algorithm with accurate frequency resolution is developed on a wireless sensor node for bearing fault diagnosis. The key processing steps of the algorithm include down-sampling, data frame overlapping and cascading. Through these steps, the resultant envelope spectrum can be obtained on the same wireless sensor node in Chapter 5 but with increased frequency resolution and amplitude accuracy for reliable diagnostic results.

Experimental evaluation shows that after down-sampling and cascading, the frequency resolution is increased by eight times in comparison with the normal envelope analysis algorithm in Chapter 5. The edge distortions caused by Hilbert transform is minimised by a 50% overlap scheme, which requires real-time signal processing performance. With these improvements, the tested bearing fault can be identified reliably in the envelope spectrum.

CHAPTER 7

OPTIMAL BAND-PASS FILTER SELECTION BASED ON FAST KURTOGRAM

With the improved envelope analysis in the last chapter, a more accurate fault feature can be obtained for bearing fault diagnosis. Meanwhile, it can be noticed that the fault features can be successfully extracted out only when a proper band-pass filter is applied to the band that includes resonant frequency components. This indicates that band-pass filter plays a crucial role in the envelope analysis.

In this chapter, the methods for selecting an optimal band-pass filter for envelope analysis are explored and the fast kurtogram algorithm is chosen for selecting the optimal band-pass filter. Due to its computation complexity, the fast kurtogram is implemented on the host computer and the band-pass parameters are updated periodically in the raw data backup procedure or when required.

7.1 Investigation on optimal band-pass filter selection

In the last chapter, a method is proposed to improve the frequency resolution of the envelope spectrum, thus more accurate and reliable diagnosis results can be obtained. However, the performance of envelope analysis is highly affected by the selected band-pass filter. Due to the wide varieties of the bearing resonances, a fixed frequency band may or may not encompass the structural resonances excited by the bearing damage [9]. Thus, there has been an active study on how to choose the most suitable band to perform envelope analysis [14], [120].

Traditionally, two methods are usually employed to choose a suitable band-pass filter for envelope analysis. One method is searching for a peak at a high frequency in response spectra, on the assumption that the peak frequency would be excited by bearing faults [14]. In this method, a number of trials may be needed before the correct band is selected due to the noise interferences, which obviously is not suitable for automatic applications. Another method is using a hammer impact test to find bearing housing resonances when installing the sensor [14] and a band-pass filter encompasses the system resonance is thereafter designed. However, this test usually needs an expert to use professional equipment to perform. It is therefore not convenient for practical factory cases where there are quite a number of bearing systems. In addition, the system responses may change with bearing conditions, which may cause the fixed filter parameters to be no longer suitable.

This problem has now largely been solved by the use of spectral kurtosis (SK) and the kurtogram to find the most impulsive frequency band [14]. In this method, an optimal band-pass filter can be precisely selected automatically. This means that the filter can be updated in an adaptive way and maintained optimum even if the system responses have changed with the bearing condition. However, the computation of the kurtogram is quite costly and not convenient for online industrial purposes, thus, in 2007, a fast kurtogram algorithm is proposed by Antoni to accelerate the computation [187].

In this chapter, a scheme is proposed to bring the fast kurtogram to the wireless CM system for automatic bearing fault diagnosis. The fast kurtogram is implemented on the host computer for selecting an optimal frequency band where to filter and demodulate the signal, while the envelope analysis is embedded on the wireless sensor node for calculating the envelope spectrum. With this scheme, the implementation of envelope analysis on the wireless sensor node can be made more meaningful and practical.

7.2 Introduction to fast kurtogram

7.2.1 Spectral kurtosis and kurtogram

The objective of bearing fault detection is to test if the vibration signal contains the faulty bearing signal. For a healthy bearing, its vibration signal is considered to follow the Gaussian distribution while, for a bearing with defect, impulsive series will be added to the measured vibration signal, resulting in the signal not following Gaussian distribution any more. The Gaussian distribution can usually be checked using the global kurtosis. If the signal follows a Gaussian distribution, its kurtosis is almost equal to 0, otherwise, the value will be higher than 0. However, the impulses from early stage faults are likely to be very weak and can be easily buried by the large background noises, making the fault difficult to be detected using the global kurtosis.

Therefore, SK was proposed to solve this problem, which is a function of frequency that can indicate the impulsiveness of a signal distributing in the frequency domain [14]. Its usage can be traced back to 1980s when it was firstly employed for detecting impulsive events in sonar signals [188]. The SK of a signal $x(t)$ can be computed using Eq. (7.1) based on STFT $X(t,f)$, which is local Fourier transform at time t obtained by moving a window along the signal.

$$K(f) = \frac{< |X(t,f)|^4 >}{< |X(t,f)|^2 >^2} - 2 \tag{7.1}$$

where <•> denotes the time-averaging operator. The subtraction of 2 is used to enforce $K(f) = 0$ since $X(t,f)$ here is complex Gaussian (instead of 3 for real signals) [189].

According to the definition of SK, the frequency band dominated by the bearing fault signal will incur a high value of SK, otherwise, the value of SK will be small. Therefore, the SK can be used as a filter function to choose the frequency band that has the highest level of impulsiveness, as shown in Eq. (7.2).

$$K_y(f) = \frac{K_x(f)}{[1+\rho(f)]^2} \tag{7.2}$$

where $\rho(f)$ is the noise-to-signal ratio function of frequency.

Because the computation of SK is based on STFT, its value is also critically related to the choice of the STFT window length. Therefore, the two-dimensional kurtogram is raised to improve the filter function, which is a function of both centre frequency f and the frequency resolution Δf. A combination of f and Δf determines a value of SK and is named a dyad $\{f; \Delta f\}$. The dyad $\{f; \Delta f\}$ with the maximum value is considered as the optimum band-pass filter parameters.

7.2.2 Fast kurtogram

As explained above, the kurtogram contains a host of dyads $\{f; \Delta f\}$ and a thorough computation of all possible dyads is quite costly and not practical for on-line industrial applications. Therefore, Antoni proposed a fast kurtogram algorithm to solve this problem [187]. Using this algorithm, an equivalent dyad can be computed but with much better speed performance. As an example in [187], on the same computer, for the computation of the same signal with 1.37s, the fast kurtogram took a fraction of a second while the full kurtogram took more than 10 minutes.

The principle of fast kurtogram is based on an arborescent multi-rate filter bank structure using a quasi-analytic filter [187]. Let $x(n)$ be a non-stationary process and $h(n)$ a low-pass prototype filter used to construct two quasi-analytic low-pass and high-pass analysis filters $h_0(n)$ and $h_1(n)$, in the frequency bands [0; 1/4] and [1/4; 1/2], respectively.

$$\begin{gathered} h_0(n) = h(n)e^{j\pi n/4}, \\ h_1(n) = h(n)e^{j3\pi n/4}(j^2 = -1) \end{gathered} \tag{7.3}$$

Then, these two filters are used to perform the elementary low-pass/high-pass decomposition illustrated in Figure 7.1. As it shows, the sequence number is doubled after the decomposition, but their respective length is also halved so that the total amount of data remains the same. By using this decomposition, a tree of filter banks can be built, as shown in Figure 7.2 (a). In the filter-bank tree, there are 2^k bands in the corresponding level and the coefficients $c_k^j(n)$ can be interpreted as the complex envelope of signal $x(n)$.

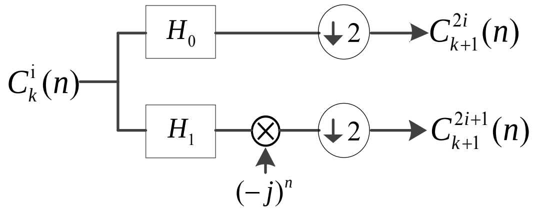


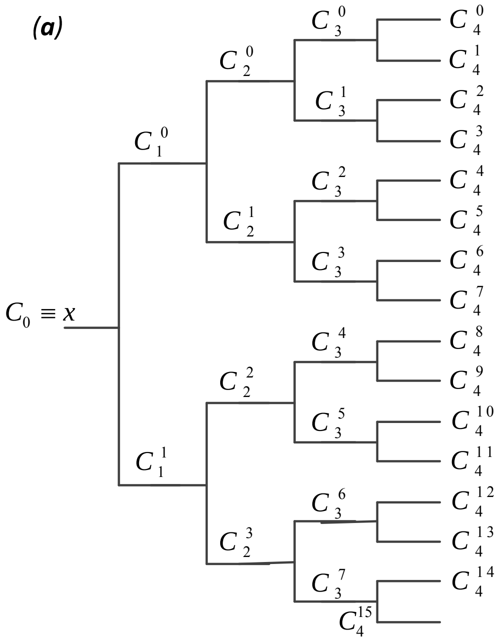
Figure 7.1 Basic decomposition

Then, the kurtogram K_k^i is estimated using Eq. (4) by computing the kurtosis of all sequences $c_k^i(n),\ j = 0, \dots, 2^k - 1,\ k = 0, \dots, K - 1.$

$$K_k^i = \frac{< |c_k^i(n)|^4 >}{< |c_k^i(n)|^2 >^2} - 2 \tag{7.4}$$

To achieve a finer sampling of the dyad plane, a 1/3-binary tree of filter-banks is extended on the binary tree structure shown in Figure 7.2 (a). Three additional quasi-analytic band-pass filters $g_j(n), j = 0,1,2$ are defined with pass-bands [0; 1/6], [1/6; 1/3] and [1/3; 1/2], respectively. These filters are then used to further decompose each sequence $c_k^i(n)$ into three sub-sequences $c_{(k+1).6}^{3*i+j}(n), j = 0, 1, 2.$

Finally, the kurtogram of the corresponding 1/3-binary tree are also calculated using Eq. (4) and the complete fast kurtogram paving of the $(f_i, (\Delta f)_k)$ plane is illustrated in Figure 7.2 (b). For each dyad $\{f_i, (\Delta f)_k\}$, a kurtosis value is calculated. The optimum band-pass filter is expected to have the maximum kurtosis value.



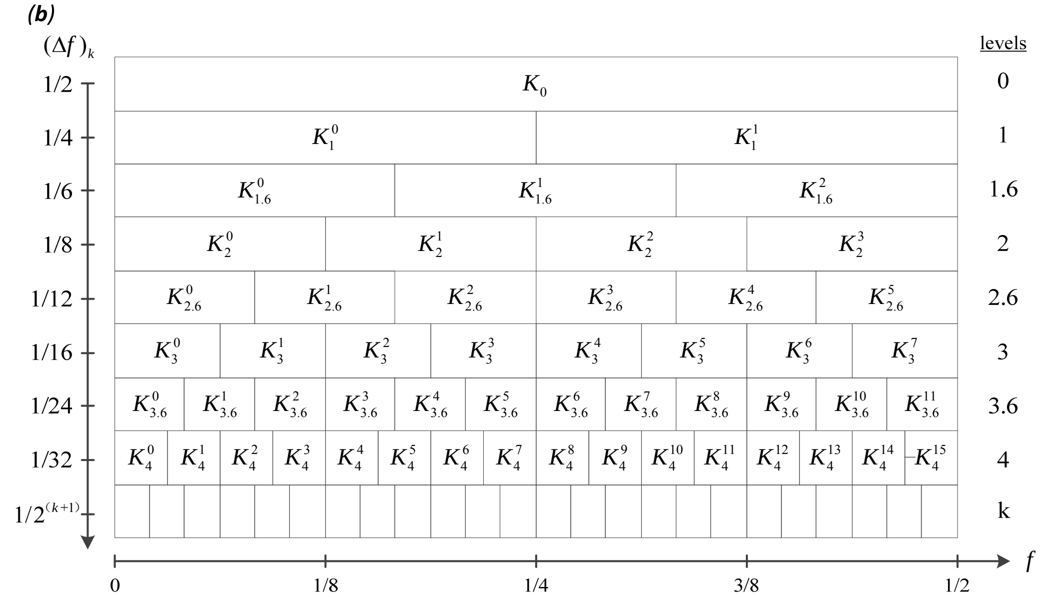


Figure 7.2 (a) Fast calculation of the kurtogram by using an arborescent filter-bank structure and (b) the complete paving of the (frequency / frequency resolution) plane

7.3 Implementation of fast kurtogram in the wireless sensor network

7.3.1 Implementation scheme

To accomplish the task of band-pass filter automatic selection and localised envelope analysis in the proposed wireless CM system, the system works in two modes which are named as configuration mode and monitoring mode respectively, as shown in Figure 7.3. The configuration mode is used for updating the optimal band-pass filter coefficients. It operates at the installation of the system and during a period when a significant change appears in the monitoring mode or at a given long time interval. Most of the time, the system works in the monitoring mode, in which the wireless sensor node uses the optimum band-pass filter calculated in the configuration mode to analyse the vibration signal and transmit the envelope spectrum to the host computer for fault diagnosis. Thus, this system can automatically finish the optimum band-pass filter selection and perform effective condition monitoring.

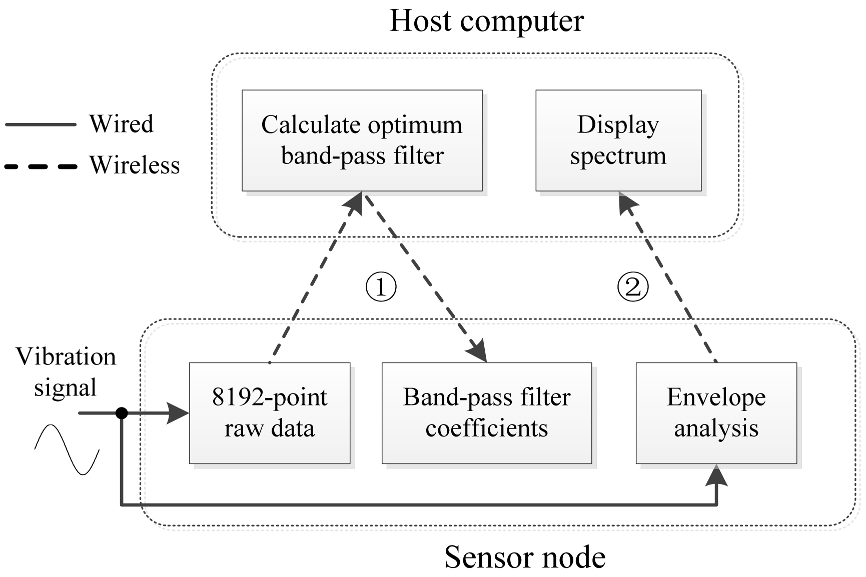


Figure 7.3 System working scheme

7.3.2 Configuration mode

In this mode, a dataset containing 8192 points of data is firstly collected on the sensor node and transmitted to the host computer via the wireless network. Then, the fast kurtogram is computed on the host computer to obtain the optimal band-pass filter. As discussed in Section 7.2.2, the filters are designed as a complex type, which is quite challenging and time-consuming when implementing on the memory limited microcontroller. Thereafter, an equivalent FIR filter with only real part coefficients is generated based on the parameters of the calculated optimum band-pass filter. At last, the coefficients of the FIR filter are sent back to the sensor node where the coefficients will be renewed for the next coming computation cycle.

In this application, the sampling rate of the ADC is raised to 32 kHz and 8192 points of data is collected. That’s a signal length of about 0.256 seconds, which is fairly enough for performing fast kurtogram. As discussed in Chapter 5, the frequency range of interest in the bearing system is less than 500 Hz. According to Figure 7.2 (b), the bandwidth at level 5 has reached 500 Hz with the sampling rate at 32 kHz, therefore, 5 levels’ analysis is sufficient for this system.

7.3.3 Monitoring mode

Most of the time, the system works in the monitoring mode, in which the envelope analysis is carried out at the sensor node and the calculated envelope spectrum is transmitted wirelessly to the host computer for display and further analysis. Due to limited computing memory size on the sensor node, only small frames of data can be calculated at one time. To obtain a high-frequency resolution, the algorithm proposed in Chapter 6 is employed. For this method, a large frame of data is divided into several smaller frames and processed in different time divisions.

To explore the optimal band-pass filter in a wider frequency band, the sampling rate is increased to 32 kHz and the diagram for the data processing scheme is updated and illustrated in Figure 7.4, which is upgraded from Figure 6.3. As shown in Figure 7.4, the collected data are also processed frame by frame with a size of 512 points. The frame first passes through a band-pass filter and combines with the last filtered frame to produce a new frame of 1024 data points. Then the envelope of the newly produced frame is computed using Hilbert transform. To avoid distortions on the edge of the extracted envelope using Hilbert transform, only the middle half of the computed envelope data are reserved. Then, the data are filtered using a low-pass filter with the cut-off frequency at 500 Hz and down-sampled by a ratio of 1/32 to save computation memory and reduce the calculation amount. After the process, the original 512 points of data shrink to only 16 points while the envelope information of the raw data is still reserved.

As shown in Figure 7.4, from the second frame, every frame of data will create 16 points of envelope data and after the 129th frame; an envelope signal with 2048 points can be obtained. The consistency of envelope signal is guaranteed by a 50% overlap processing when calculating the envelope.

In the above process, there are two kinds of sampling rate: 32 kHz and 1 kHz. Data with high sampling rate share the same computing space and are computed in different time divisions. The high sampling rate of the raw data allows high-frequency components to be analysed and the low sampling rate of the envelope signal allows longer analysis time length and thus higher frequency resolution can be achieved.

For the down-sampled envelope signal, its equivalent sampling rate becomes 1 kHz; therefore, the resolution of the envelope spectrum is about 0.49 Hz. This means the frequency band of interest has about 1024 points of data, which are the data needs to be transmitted to the host computer.

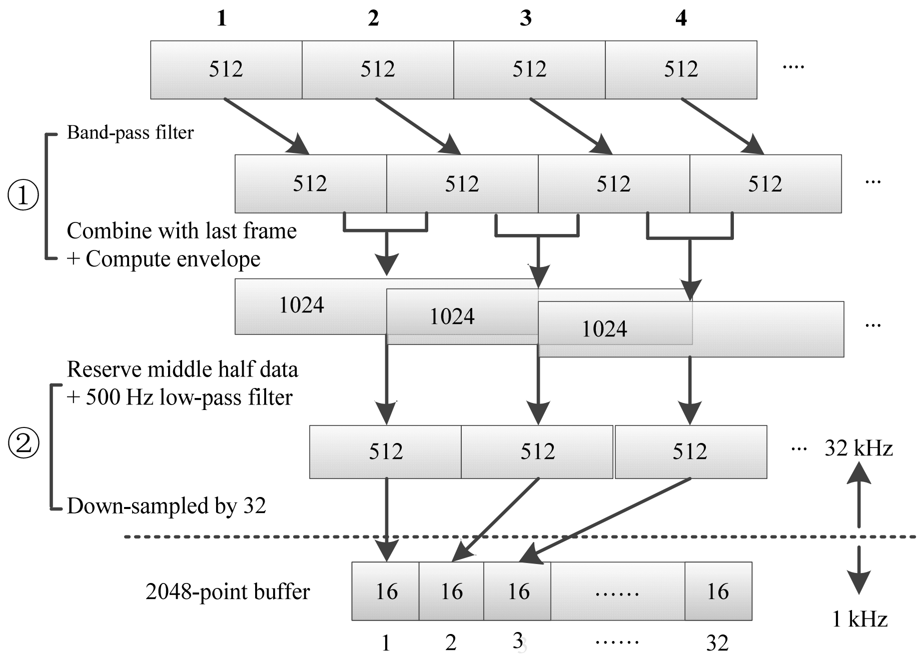


Figure 7.4 Diagram for the overlap, down-sampling and cascading scheme

Recall the signal processing diagram in Figure 6.5, where the calculation for one frame of envelope signal requires real-time computation performance. Because of the increase in sampling rate, it brings more strict real-time computation performance. The deadline for processing one frame data of 512 points sampled at 32 kHz can be calculated as 512/32=16 ms.

Because the sampling rate is increased, the length of the FIR filter needs to be increased as well in order to achieve the same frequency response performance for the same pass band. Figure 7.6 shows the 512-point magnitude and phase response of a series of FIR low pass filter with different lengths, which are designed using the same pass band and stop band criteria. It can be seen that all filters have a linear phase response but the ones with longer filter coefficients have a flatter passband response and a greater attenuation ratio in the stop band.

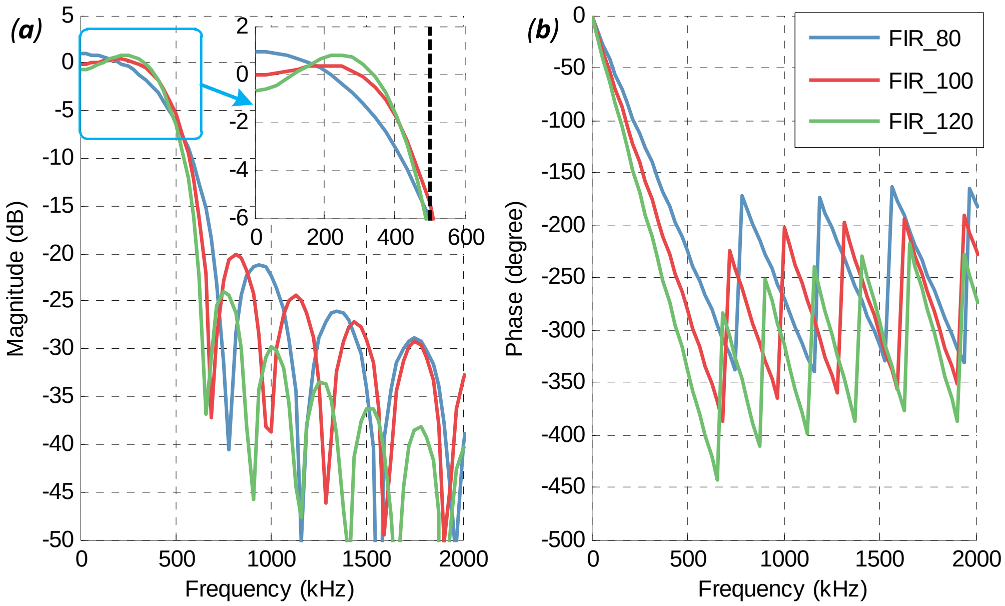


Figure 7.5 512-point filter response of a series of FIR low-pass filter with different lengths: (a) magnitude response with magnified portion shown as an inset and (b) phase response

Similarly, a series of FIR band-pass filters are designed using the same pass band and stop band criteria and their 512-point magnitude and phase response are presented in Figure 7.6. The similar phenomenon as Figure 7.5 can be observed, with the increase of filter length, the magnitude response within pass band becomes flatter and the attenuation ratio of the stop band becomes greater.

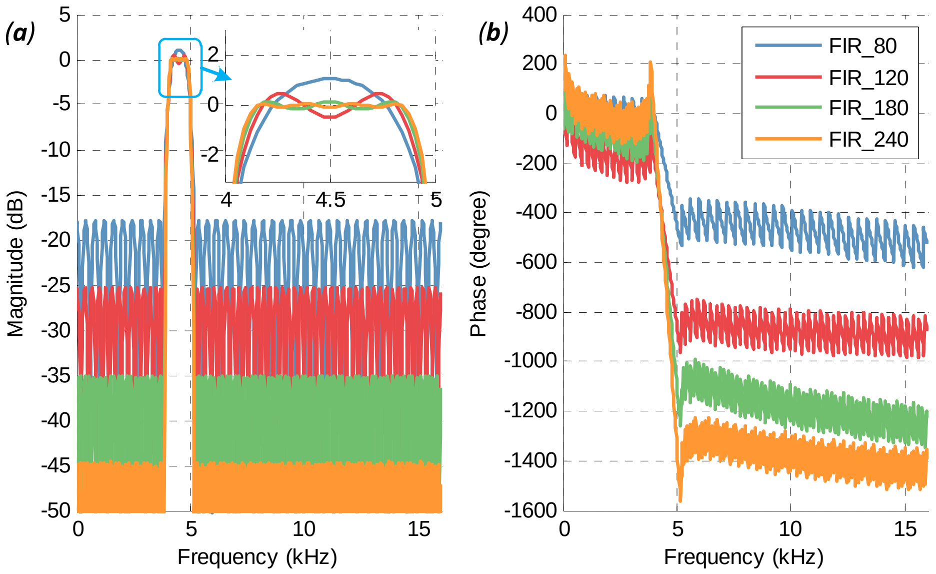


Figure 7.6 512-point filter response of a series of FIR band-pass filter with different lengths: (a) magnitude response with magnified portion shown as an inset and (b) phase response

Therefore, to get better filtering results, there needs a longer length of the low-pass filter and a band-pass filter. However, the increase of filter length also increases the computation time significantly. Thus, there should be a balance between filtering performance and computation time. The computation time of one frame of the envelope for different band-pass and low-pass filter lengths is presented in Figure 7.7, which are calculated for the MCU running at 80 MHz.

It can be observed that the real-time requirement cannot be achieved for longer length low-pass and band-pass filter combinations. Together with the filter response in Figure 7.5 and Figure 7.6, it can be observed that low-pass filter length of 100-point and a band-pass filter of 160-point or 180-point can be a good balance between filtering performance and computation time.

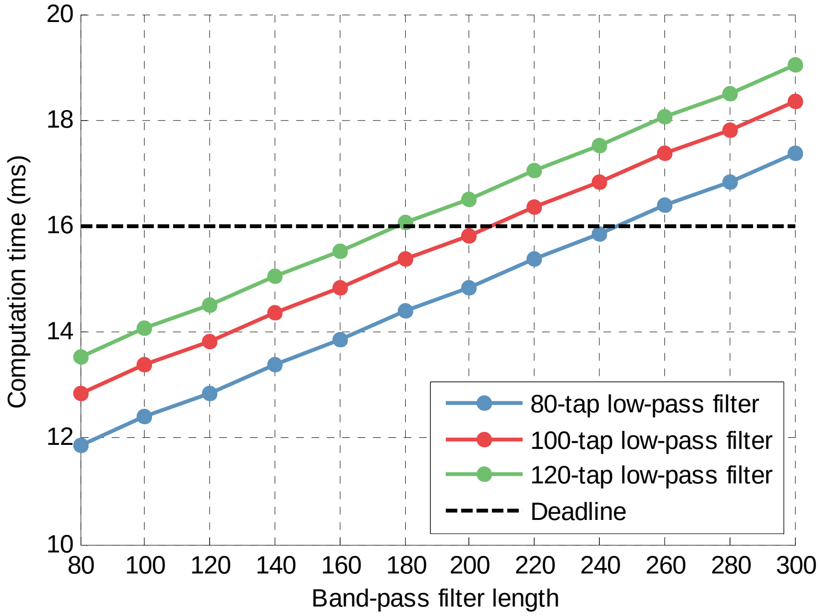


Figure 7.7 Computation time of one frame of envelope for different band-pass and low-pass filter lengths

7.3.4 Band-pass filter implementation in the frequency domain

Although the real-time performance for the calculation of one frame of the envelope can be satisfied with a proper combination of low-pass and a band-pass filter, the computation time is rather close to the deadline requirement, with just about 1 ms' headroom. If other operations are conducted in the meantime, the calculation might not be able to be finished in time. For this consideration, an improvement is applied on the implementation of diagram for the overlap, down-sampling and cascading scheme, as shown in Figure 7.8. The idea of this improvement is to remove the FIR band-pass filter calculation in the time domain and apply directly a band-pass window on the calculated FFT results in the frequency domain. In this way, the former convolution operation of the FIR filter becomes a multiplication in the frequency domain. Thus, the calculation can be much faster.

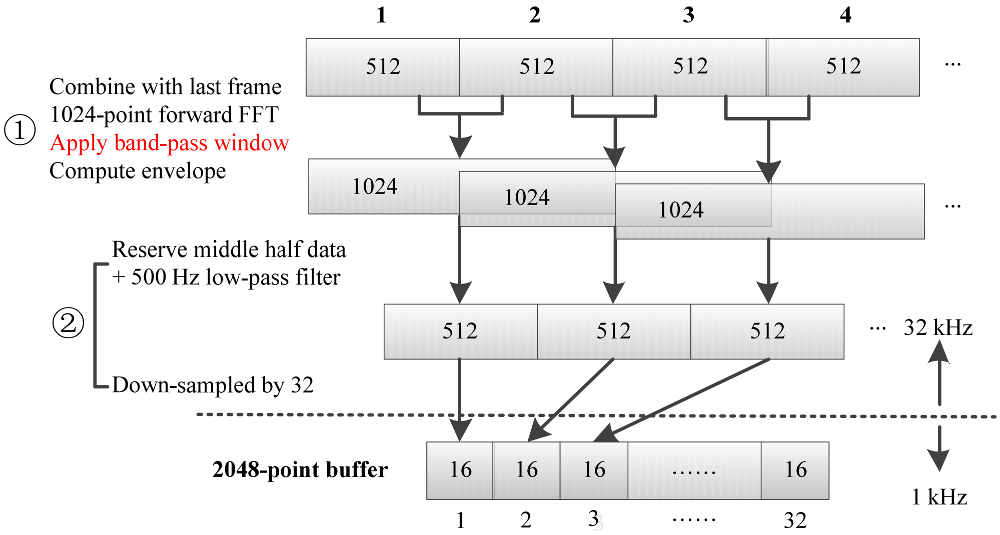


Figure 7.8 Improved diagram for the overlap, down-sampling and cascading scheme

Of note, applying a rectangle window in the frequency domain usually, introduces distortions to the corresponding signal in the time domain. Because an overlapping mechanism of up to 50% has already been applied in the diagram, the end distortions due to the window effect can be effectively avoided. Another benefit of applying band-pass window in the frequency domain is that there is no need to transmit long FIR coefficients, instead only two boundary frequencies, i.e. the lower and upper cut-off frequency need to be updated in the configuration mode.

The computation time of one frame of the envelope by applying different band-pass window widths and different low-pass filter lengths is presented in Figure 7.9. It can be seen the computation time is significantly reduced and all the combinations in the figure can satisfy the real-time computing requirement, with the headroom for more than 5 ms. For the same low-pass filter, the increase of band-pass window width only brings a slight increase in the computation time. Thereby, applying a band-pass filter in the frequency domain has better real-time performance than the FIR method.

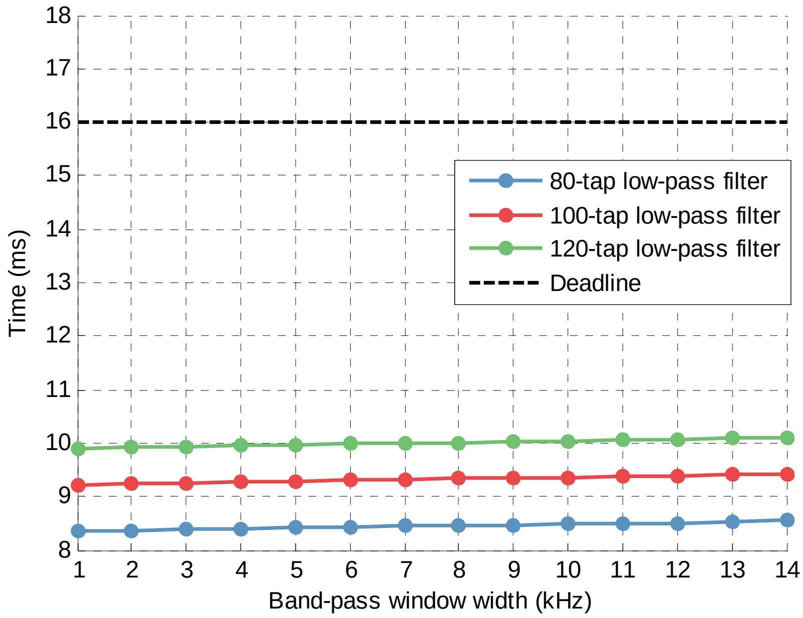


Figure 7.9 Computation time of one frame of envelope by applying different band-pass window widths

7.4 Results and discussion

To evaluate the performance of the fast kurtogram for selecting optimal band-pass filter and the effectiveness of the selected band-pass filter for extracting bearing fault features, the proposed condition monitoring scheme is employed for processing vibration signals from three types of defect bearing, including outer race fault, inner race fault, and roller fault.

7.4.1 Outer race fault detection results

A vibration signal (8192 points, approximately 0.25s) measured on the bearing with outer race fault is shown in Figure 7.10 (a) and its spectrum in Figure 7.10 (b). This data set is received from the host computer in the configuration mode and calibrated on the host computer before displaying. As shown in Figure 7.10 (a), periodical spikes are observable which are caused by the defect on the outer race. From its spectrum, it can be seen that the signal has a wide frequency range which makes it difficult to identify the fault types. Several peaks exist in the spectrum and it’s not easy to tell which one is suitable for implementing the envelope analysis.

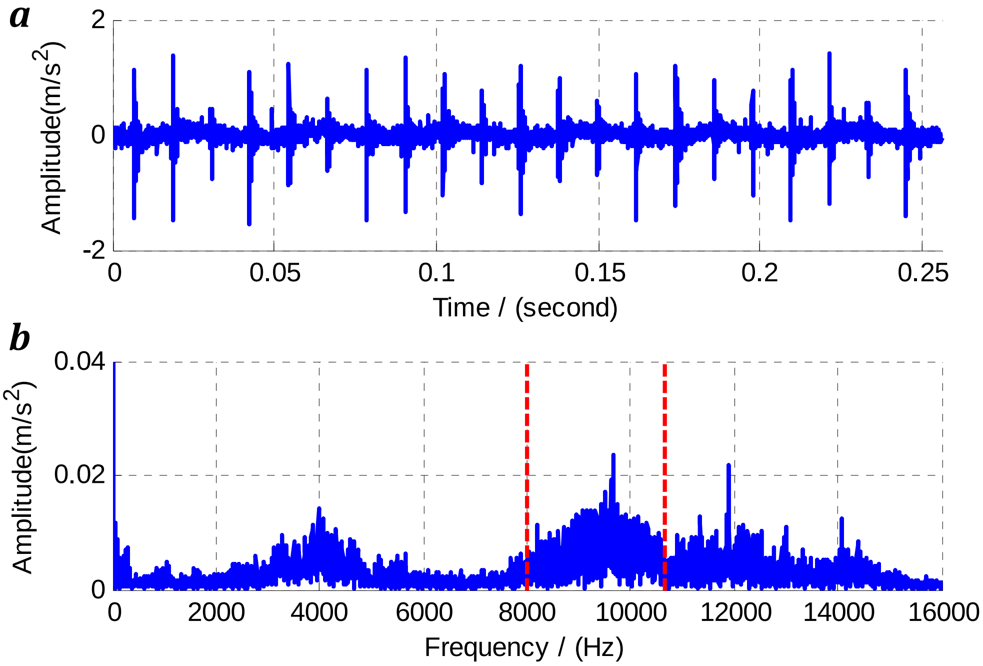


Figure 7.10 Outer race raw dataset: (a) time domain and (b) frequency domain

The corresponding fast kurtogram of this data set is displayed in Figure 7.11, which shows a high non-stationary activity around the carrier frequency f = 9.33 kHz. The dyad which maximises the kurtosis is $\{f_4; (\Delta f)_{2.6}\} = \{9.33\ kHz;\ 2.67\ kHz\}$, that is the frequency band $\{8\ kHz;\ 10.67\ kHz\}$. According to Figure 7.10 (b), high peaks are observable in this frequency band. Thus, band-pass filter window with pass band between 8 kHz and 10.67 kHz is applied on the sensor node for implementing envelope analysis.

The envelope spectrum computed on the sensor node is given in Figure 7.12. It clearly shows the spectrum lines at the characteristic frequency of outer race fault. Without question, it can be based on to diagnose the outer race fault on the bearing.

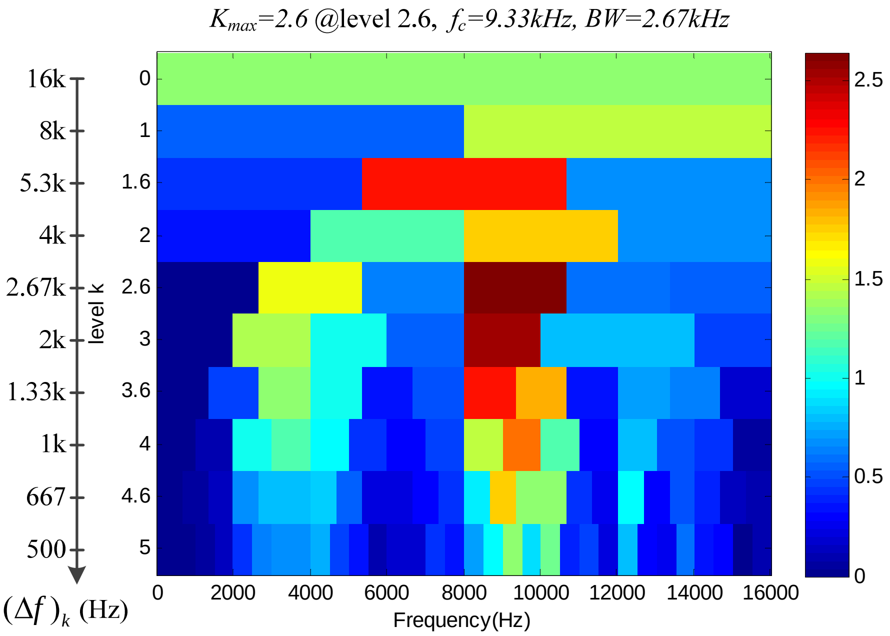


Figure 7.11 Fast kurtogram of the outer race raw data set, maximum kurtosis is 2.6 at level 2.6. The optimum band-pass filter is centred at 9.33 kHz with 2.67 kHz bandwidth.

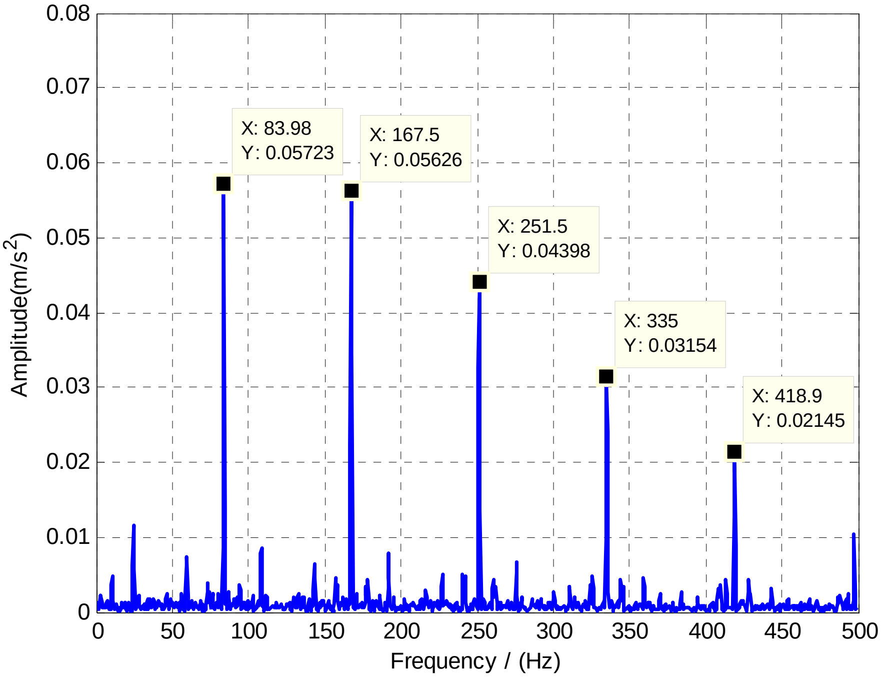


Figure 7.12 Envelope spectrum for bearing with outer race fault

7.4.2 Inner race fault detection results

Similarly, a dataset of vibration signal was collected on the bearing with inner race fault, as shown in Figure 7.13 (a), and its spectrum in Figure 7.13 (b). The periodical spikes are observable but not as obvious as those in Figure 7.10 (a). Its spectrum also spread over a wide frequency range and the peaks are a bit different from those in Figure 7.10 (b), which indicates the resonances of the bearing system have changed slightly with the change of the installed bearing.

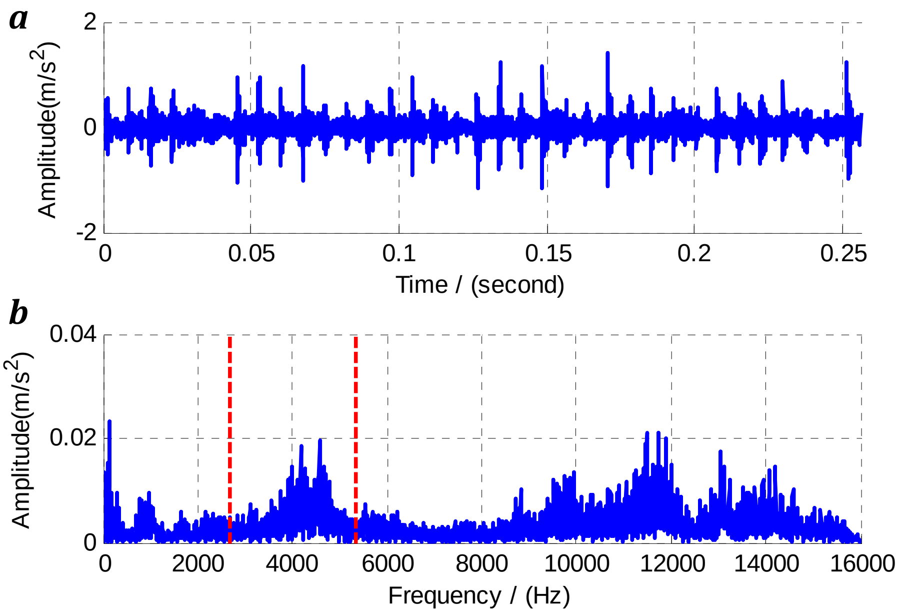


Figure 7.13 Inner race raw dataset: (a) time domain and (b) frequency domain

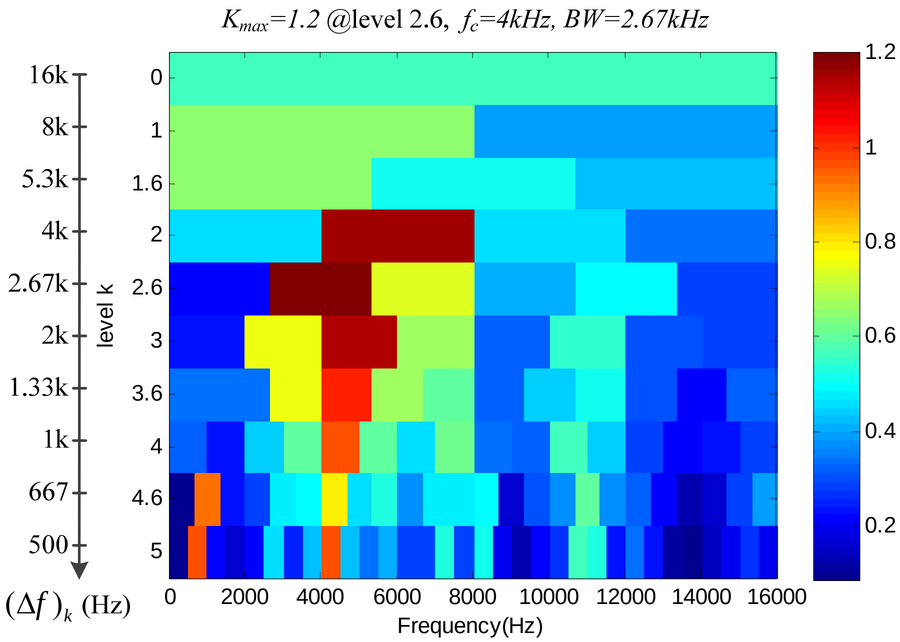


Figure 7.14 Fast kurtogram of the inner race raw data set, maximum kurtosis is 1.1 at level 2.6. The optimum band-pass filter is centred at 4 kHz with 2.67 kHz bandwidth.

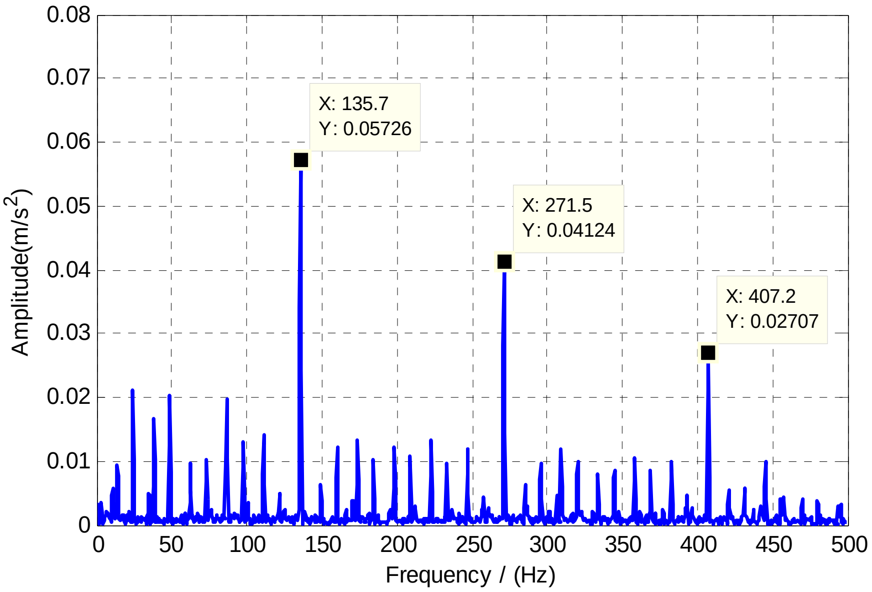


Figure 7.15 Envelope spectrum for bearing with inner race fault

The corresponding fast kurtogram is obtained as shown in Figure 7.14. Its maximum kurtosis is reached at dyad $\{f_2; (\Delta f)_{2.6}\} = \{4\ kHz;\ 2.67\ kHz\}$. As shown in Figure 7.13 (b), the frequency bins around 4 kHz have relative higher amplitudes. Therefore, an 80-order FIR band-pass filter with the frequency band from 2.67 kHz to 5.33 kHz is designed accordingly.

The envelope spectrum calculated using the above band-pass filter is shown in Figure 7.15, which clearly reveals the existence of the inner race defect on the bearing.

7.4.3 Roller fault detection results

A dataset of vibration signal was collected on the bearing with roller fault and presented in Figure 7.16(a), and its spectrum in Figure 7.16(b). The periodical spikes are observable but not as obvious as those in Figure 7.10 (a). Its spectrum also spread over a wide frequency range and the peaks are a bit different from those in Figure 7.10 (b), which indicates the resonances of the bearing system have changed slightly with the change of the installed bearing.

The corresponding fast kurtogram is obtained as shown in Figure 7.17. Its maximum kurtosis is reached at dyad $\{f_5; (\Delta f)_4\} = \{4.5\ kHz;\ 1\ kHz\}$. As shown in Figure 7.17(b), the frequency bins around 4 kHz have relative higher amplitudes. Therefore, an 80-order FIR band-pass filter with the frequency band from 4 kHz to 5 kHz is designed accordingly.

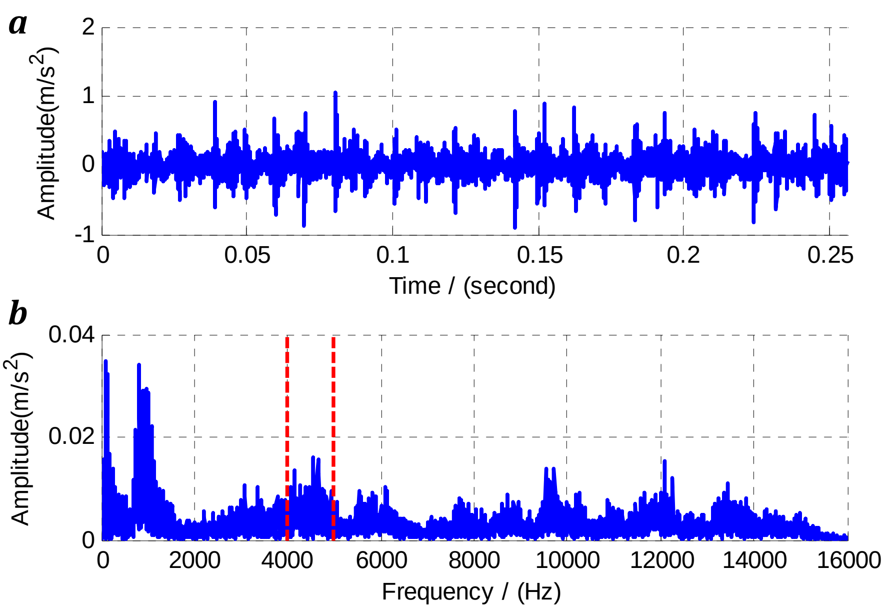


Figure 7.16 Roller raw dataset: (a) time domain and (b) frequency domain

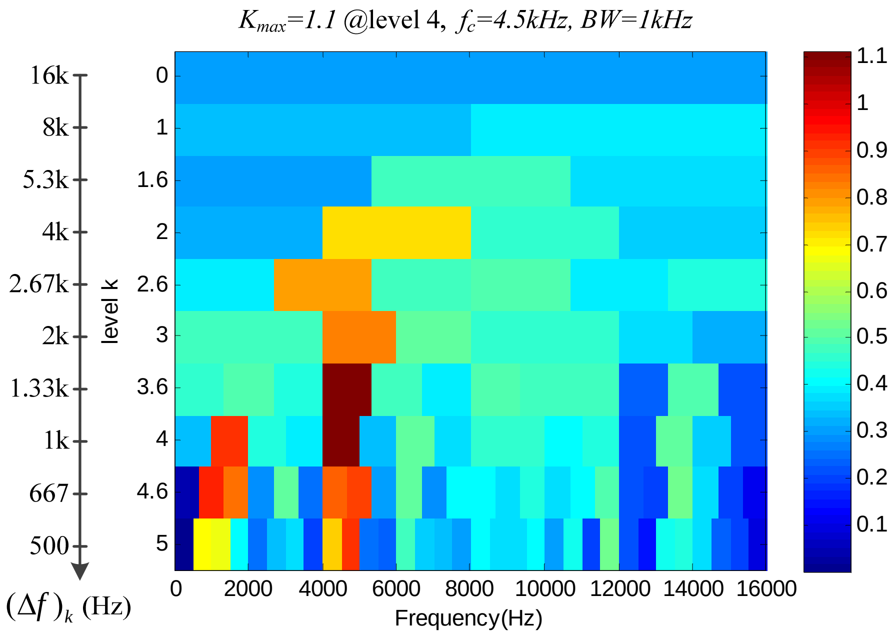


Figure 7.17 Fast kurtogram of the roller raw data set, maximum kurtosis is 1.1 at level 4. The optimum band-pass filter is centred at 4.5 kHz with 1 kHz bandwidth.

The envelope spectrum calculated using the above band-pass filter is shown in Figure 7.18, which clearly reveals the existence of the roller defect on the bearing.

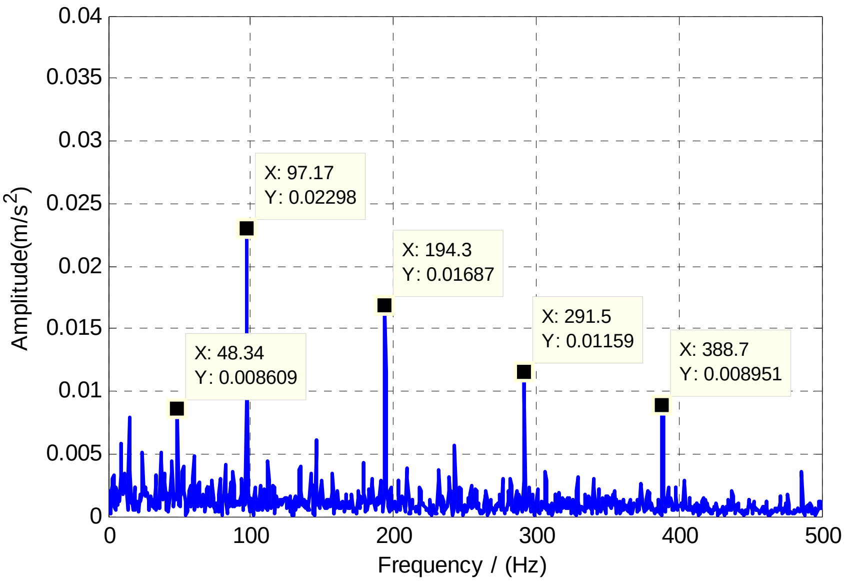


Figure 7.18 Envelope spectrum for bearing with roller defect

7.4.4 Data throughput requirement analysis

In the configuration mode, a large dataset needs to be transmitted over the wireless network for kurtogram calculation and filter optimisation. However, considering that the fault progression is a slow and gradual process, the filter, once optimised, can be used for a long time period. This means that this mode does not need to run frequently; instead, it may be activated once a week or a month when the raw data backing up is carried out. In addition, the large raw dataset does not need to be transmitted in real-time. So it can be organised into small packages for transmission when the wireless network is not busy.

On the other hand, the monitoring mode requires a high-speed data flow so that the monitoring can be achieved in real time. As discussed in Section 3.3, 2048 points of envelope data are produced for every 129 frames of raw data (each frame is composed of 512 points of data) and the envelope spectrum only contains 1024 points of effective data. In order to evaluate the performance of data reduction in this mode, the data throughput requirements in different processing stages are summarised in Table 7.1. All the data are supposed to be transmitted at a 16-bit resolution occupying two bytes' space.

Table 7.1 Data throughput requirement comparison

Processing stages	Data points per computation cycle	Data throughput requirement (kbps)
Raw dataset	66048	512
Envelope	2048	15.88
Envelope spectrum	1024	7.94
Averaged spectrum	256	1.99

As shown in Table 7.1, the data throughput requirement for the raw data is as high as 512 kbps which are even higher than the maximum transmission speed of ZigBee, i.e. 250 kbps. However, after the processing on the sensor node, the data throughput of the envelope and its spectrum are significantly reduced to about 16 kbps and 8 kbps, respectively. Supposing a four-time average is carried out on the envelope spectrum, the data throughput requirement can be further decreased to less than 2 kbps. Therefore, multiple such sensor nodes would be allowed to coexist in the same wireless network for bearing fault diagnosis.

7.5 Summary

In this chapter, an adaptive envelope analysis scheme is proposed for automatically selecting the optimal band-pass filter in the wireless CM system. The fast kurtogram is realised on the host computer due to its computation complexity and envelope analysis is embedded in the wireless sensor for extracting envelope spectrum and reducing the data throughput requirement. By using this scheme, the vibration signal can be monitored effectively over the bandwidth limited wireless sensor network for bearing fault diagnosis. Furthermore, frequency domain filtering is implemented instead of the FIR filter to reduce data size for filter parameters updating and the sampling frequency is increased to 32 kHz to allow a wide frequency band being analysed while similar level frequency resolution is still kept as that in Chapter 6.

CHAPTER 8

IMPROVEMENT OF COMPUTATION SPEED OF ENVELOPE DETECTION METHODS

From the work in the last three chapters, it can be noticed that envelope detection is a most time-consuming part of the envelope analysis process. For this reason, this chapter explores several envelope detection methods, including the simple envelope detectors, a recent method based on short-time statistical features and the precise Hilbert transform.

The simple envelope detectors, especially the squared rectifier, have fast computation speed but also suffer from accuracy problems. With a further study, it is found that it can get an envelope with satisfactory accuracy as Hilbert transform when processing bearing vibration signal. Meanwhile, Hilbert transform is fully explored in both time domain and frequency domain and a fast implementation method is proposed in the frequency domain for getting the squared envelope spectrum.

At the end, two fast implementation methods are proposed and employed for calculating the spectrum of high resolution and adaptive filtering. The experimental results show satisfactory accuracy and computation improvement in comparison with the algorithm in Chapter 7.

8.1 Simple envelope detectors

Envelope detection has played an important role not only in extracting fault features in vibration signals from rotating machinery components but also in many other areas, such as ECG signal processing in the medical area [190] and envelope tracking to improve the power efficiency of the transmission amplifier in the communication area [191].

In radio signal processing, a simple rectifier circuit shown in Figure 8.1 can be employed for the demodulation of an amplitude modulated (AM) analogue signal. This method can also be utilised to process a vibration signal from rotating components (like bearings, gears or valves) with AM effect. However, it should be noted that the carrier frequency in radio (several MHz) are usually much higher than that of modulating signals (several kHz). For a bearing vibration signal with AM effect, the carrier signal comes from resonances of the bearing housings, whose frequency (several kHz) are close to that of the modulating faulty signals (dozens to hundreds Hz). In addition, vibration signals usually contain several resonances, which are also varied for different bearing systems. Thus, it is difficult to find proper analogue filtering appropriate circuits for getting the faulty signal from an AM vibration signal by using a simple rectifier shown in Figure 8.1. In comparison, the digital signal processing can also fulfil the rectifiers in analogue circuits and more flexible filtering options can be applied to get a better processing result.

8.1.1 Half-wave, full-wave and squared rectifier

Three simple envelope detectors are presented in Figure 8.2, which can be used to simulate the half-wave, full-wave and squared rectifier methods. Comparatively, when a sinusoidal signal passes through a half-wave or full-wave rectifier, the harmonics extend to infinity in the output spectra while the output of the squared signal only contains DC component plus double frequency component of the original signal. The harmonics in the half-wave and full wave rectifier method will inevitably alias into the measurement range due to the fact that the sampling rate is limited.

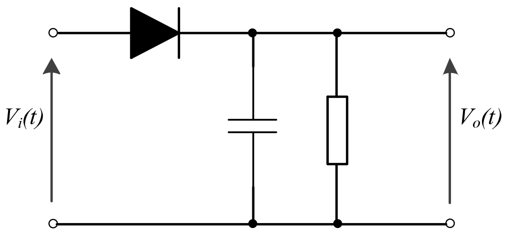


Figure 8.1 Half-wave rectifier

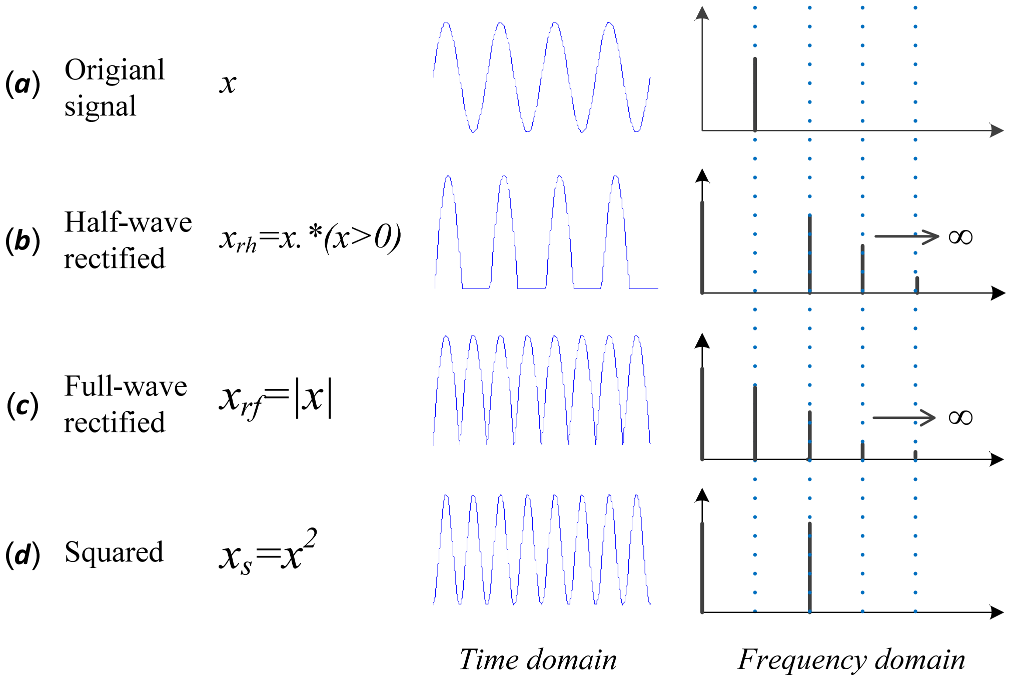


Figure 8.2 Illustration of a signal passing through different rectifiers: (a) original signal, (b) half-wave rectified signal, (c) full-wave rectified signal and (d) squared signal

Figure 8.3 shows a modulation signal passes through the above three rectifiers, where the modulation signal is centred at 1500 Hz with two frequency components 80 Hz and 120 Hz situated at its both sides. As shown in Figure 8.2, after passing through the three types of rectifiers, harmonics of the half-wave and full wave rectifier extend further and aliases back to the low-frequency band as highlighted in the red circle in Figure 8.2(b) and (c). Meanwhile, in the spectra of the squared signal, there is no such distortion. However, it can be expected that if the carrier frequency of the modulation signal is near or even higher than the ¼ of the sampling frequency, similar aliasing problems could also happen in the spectra of the squared signal.

It can be observed that low-frequency components are demodulated in the spectra of rectified signals in Figure 8.3 (b), (c) and (d). With a closer look at this frequency band of these rectified signals as shown in Figure 8.4. The modulating frequency components

80 Hz and 120 Hz can be found in all three spectra while in the spectra of the squared signal, frequency components of 160 Hz, 240 Hz, and 200 Hz can also be observed. These extra frequency components are actually the 2nd harmonics and sum of 80 Hz and 120 Hz, which result from the squared effect and can be removed by the following square-root method.

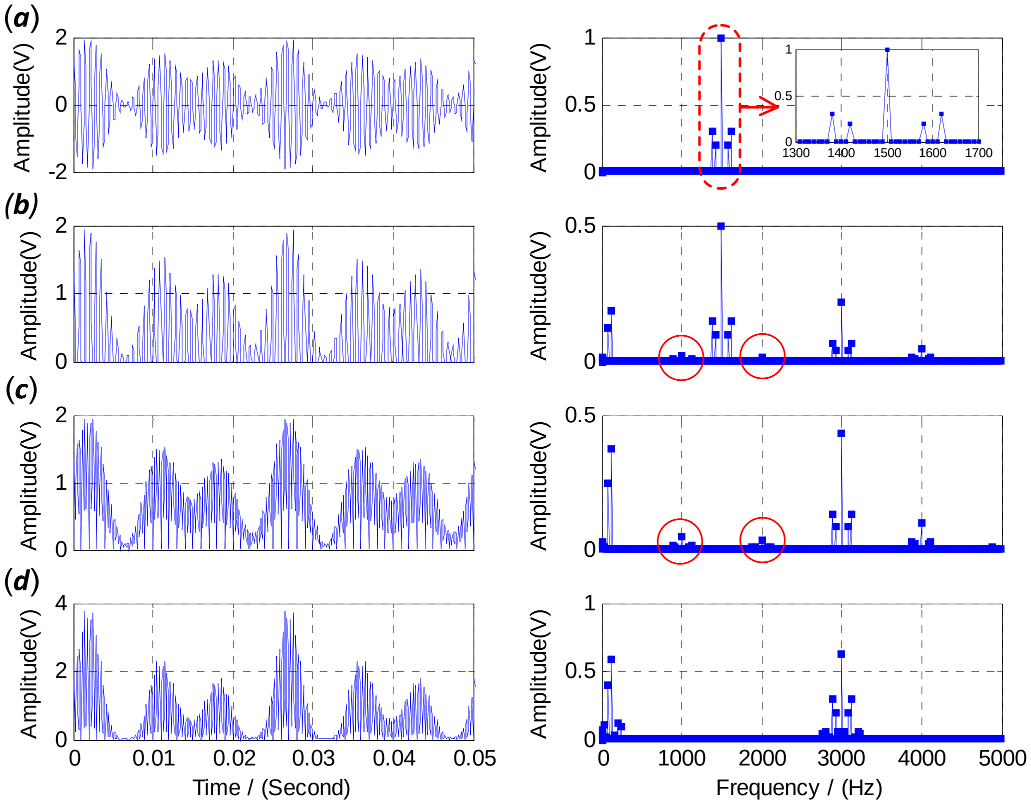


Figure 8.3 A modulation signal centred at 1500 Hz: (a) original signal, (b) half-wave rectified signal, (c) full-wave rectified signal and (d) squared signal

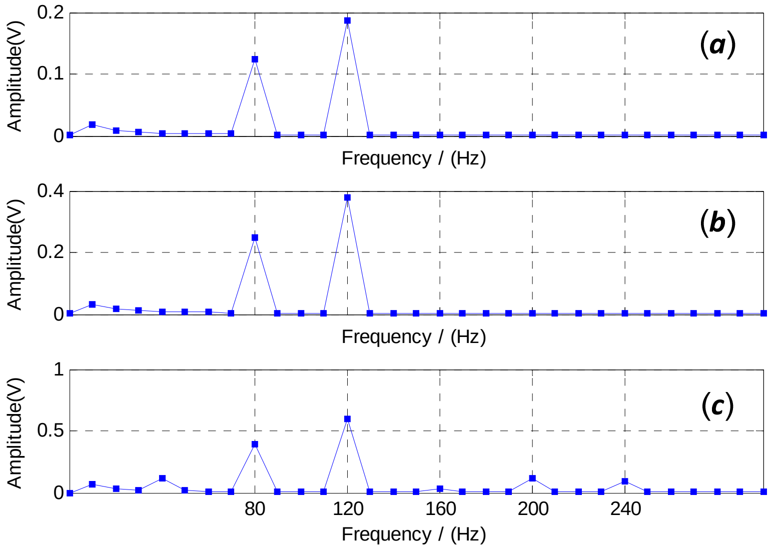


Figure 8.4 Partially magnified view of the spectra: (a) half-wave rectified signal, (b) full-wave rectified signal and (c) squared signal

In summary, distortions will be inevitably brought in with half-wave and full wave rectifier method due to infinite extended harmonics. Distortions can be attenuated with an increase of the sampling frequency, in which case, fewer harmonics will alias back into the measurement range. Meanwhile, distortions in squared rectifier can be compensated with a following square root method but also requires a higher sampling frequency to avoid aliasing. Note that in some cases, the squared envelope spectrum might be more preferred as it can provide higher SNR in comparison with the envelope spectrum [49].

In practice, vibration signals are usually band-pass filtered before the squared operation to find an optimal band and enhance SNR as discussed in Chapter 7. Later in Section 8.3.2, it is compared with the Hilbert transform method and the reason for its capability for demodulation will be explained.

8.1.2 Peak detector

In the above three rectifier methods, new frequency components are brought in, which may cause an aliasing problem if they are not properly handled. With the increase of sampling frequency, distortions can be reduced, which consequently will increase the computation load. The peak detector is another simple but effective envelope detector, in which no new frequency components are generated. Figure 8.5 highlights that the peak-detector is an approximation of the envelope signal. When the sampling frequency is much higher than the carrier frequency, the results would be more precise. Otherwise, when the carrier frequency is close to the sampling frequency, the sampling points may not appear near the carrier peaks and thus, the results may not be accurate. Therefore, the peak detector method also requires a much higher sampling frequency of the carrier signal.

It should be noted that the peak detector can be extremely sensitive to noises and hence the detection result of peak-detector has low repeatability problem sometimes and can lead to a false reading.

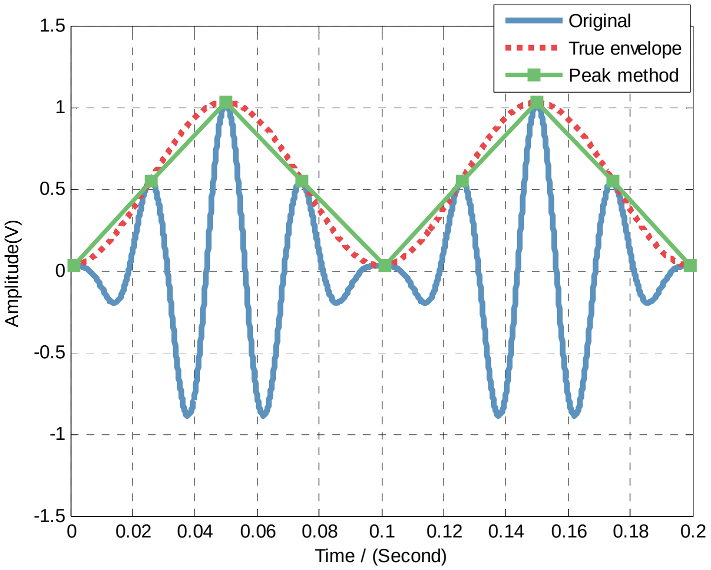


Figure 8.5 Envelope detected by peak method in comparison with the true envelope

8.1.3 Short-time statistical features

Short-time statistical features are another simple yet effective method for extracting bearing fault features, which were based on the concept of cyclostationary [129], [192]. The aim of this method is to extract the fault characteristic information by finding the cyclic information from the short-time features. The statistical features for calculation can be second and fourth statistical moments, peak value and kurtosis.

The short-time statistical feature of signal x by using a statistical operator P, can be written as:

$$ST_x^P(t,d) = P\{x(\tau) \times w_d(\tau - t)\} \tag{8.1}$$

Where w is the window function, t is the origin of the window, τ is the time variable, and d is the window size. The window function may be different types, such as uniform, triangle, Hanning, or Hamming. The window function slides over the signal with a constant step size, which is suggested to be smaller than 25 percent of the window size to avoid aliasing [129]. After statistical features at all windows are calculated, an FFT operation is performed on the extracted statistical features to find periodic patterns.

For better understanding, the above computation process for short-time statistical features is illustrated in Figure 8.6, where a window of size M slides over the signal x with a step size of S. At each step, a statistical feature P is computed for the signal extracted by the sliding window.

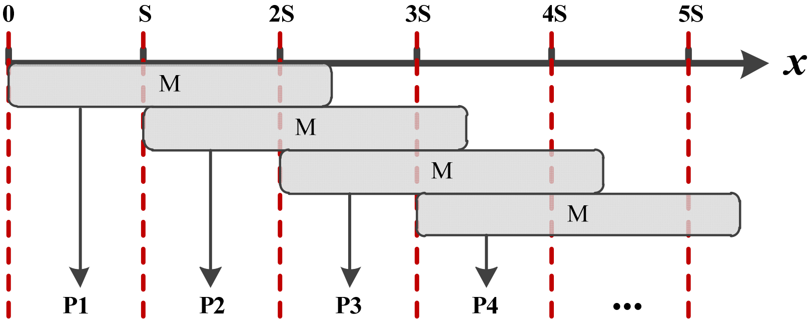


Figure 8.6 Illustration of short-time statistical feature calculation with step size of *S* and window size of *M*

In [129], Behzad et al. proved the best statistical feature for detecting localised bearing fault is the square root of the second moment, i.e. RMS. For the statistical feature RMS, this process actually calculates a series of instant power for signal x. Recall the definition of RMS in Eq. (4.6), short-time RMS can be considered as a simplified squared rectifier with the low-pass filter and downsample operation being implemented by an overlapped average. The sliding window together with RMS calculation acts like a low-pass filter to omit high-frequency fluctuations in the squared signal and the step size is the down-sampling ratio.

The window length and step size are critical parameters for short-time RMS calculation. The larger the step size is, the lower of computation complexity will be. Meanwhile, the window length plays a significant role in the success of this method. A window time length T_w for detecting all bearing fault features is recommended in [129] as:

$$T_w = \frac{T_r}{N} = \frac{1}{NF_r} \tag{8.2}$$

in which, T_r is the shaft rotation time, F_r is the shaft rotation frequency, and N is the number of rolling elements. In [129], Behzad et al selected the windows according to a so-called peak clarity ratio, which only takes the fundamental fault frequency into consideration. From the view of low-pass filter and down-sampling discussed above, the window size can be determined by the maximum interested frequency component of the short-time features and the step size should be less than half of the window size.

Figure 8.7 shows the short-time RMS and its spectrum of a vibration signal from a bearing with an outer race defect. The sampling rate of the signal is 32 kHz and a uniform type window of 64 points is applied for the calculation with a step size of 32 points. Before short-time RMS calculation, the signal is filtered by a high-pass filter with cut-off frequency at 500 Hz for removing the influence of low-frequency vibration components.

From Figure 8.7(a), it can be seen that the periodic impulses in the vibration signal can be well represented by the extracted short-time RMS feature, although this is not the envelope of the original signal. From its spectrum in Figure 8.7(b), a series of harmonics of the outer race fault frequency can be clearly observed with good SNR, which verifies the existence of the outer race defect on the bearing.

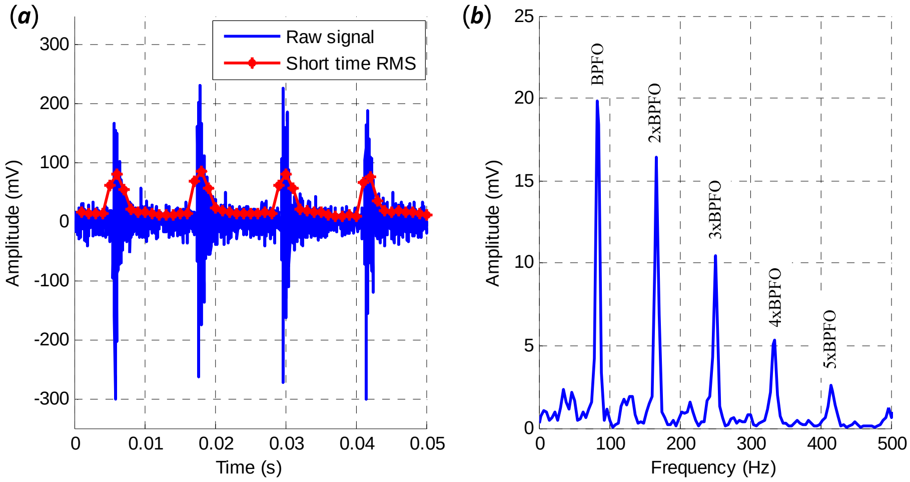


Figure 8.7 Short-time RMS of a bearing vibration signal with outer race fault
(a) raw signal and short-time RMS and (b) spectrum of short-time RMS

The schematic for a short-time RMS based envelope analysis is presented in Figure 8.8. Note that the sampling frequency of the obtained x_{env} is $1/S$ of that in the original input signal x, therefore, for the same size of FFT calculation, the envelope spectrum frequency resolution is S times of the original signal spectrum.

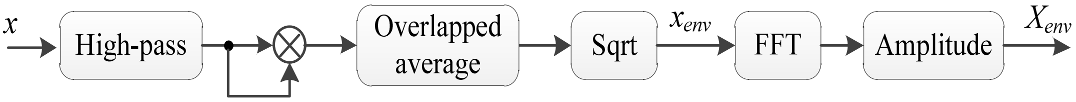


Figure 8.8 Schematic of short-time RMS based envelope analysis

8.2 Hilbert transform based envelope detection

The simple envelope detectors are easy to implement, but their performance might be unreliable with respect to the signal. In comparison, Hilbert transform based envelope detector is more precise and is not sensitive to the carrier [193]. In this section, Hilbert transform will be studied in detail and the reason for its preciseness will be explained in comparison with the squared rectifier.

8.2.1 Definition of Hilbert transform and analytic signal

Hilbert transform (HT) is named after David Hilbert and was firstly introduced to solve a special case of integral equations in the area of mathematical physics. The HT of a real-valued function $x(t)$ extending from $-\infty$ to $+\infty$ is a real-valued function $\tilde{x}(t)$ defined as [193]:

$$H[x(t)] = \tilde{x}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau \tag{8.3}$$

A complex signal whose imaginary part $\tilde{x}(t)$ is the HT of the real component $x(t)$ is called an analytic or quadrature signal [193], [194]:

$$x_a(t) = x(t) + j\tilde{x}(t) \tag{8.4}$$

An example of the analytic signal is presented in Figure 8.9. The analytic signal only contains positive or negative frequency components. The HT is usually employed either to generate or to measure complex time-domain signals, where the HT's power lies. It delivers another dimension of signal processing capabilities as we move from two-dimensional real signals to three-dimensional complex signals.

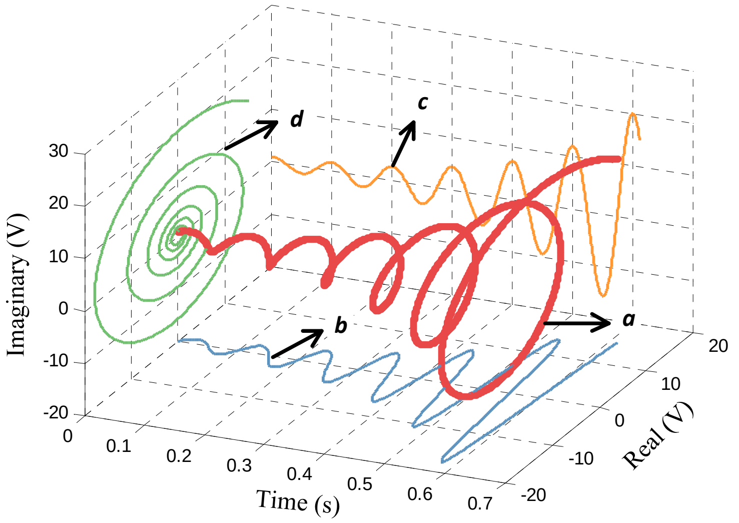


Figure 8.9 Illustration of an analytic signal in three dimensions: (a) analytic signal, (b) real signal, (c) HT signal and (d) phasor in complex plane [193]

By calculating the amplitude of an analytic signal, the envelope of a modulating signal can be obtained, as illustrated in Figure 8.10. For the implementation of Hilbert transform, it can be achieved either in the frequency domain, which is more straight forward as explained in Section 5.2.2, or can be implemented in the time domain using a Hilbert transformer, which will be explained in Section 8.2.3.

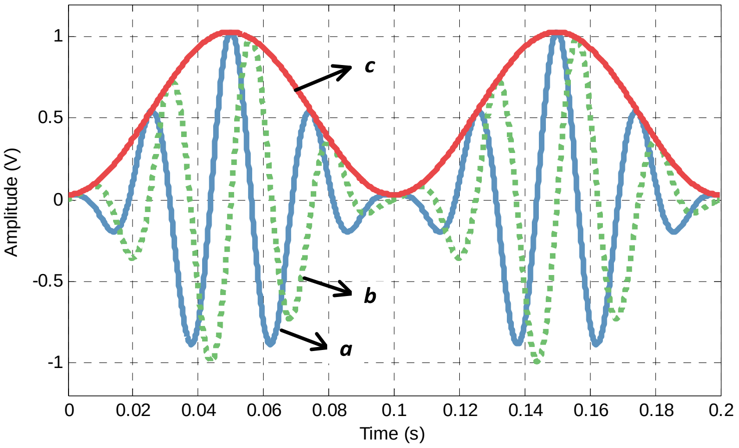


Figure 8.10 Illustration of HT based envelope detection: (a) modulating signal, (b) HT of modulating signal and (c) envelope calculated using HT

8.2.2 Frequency domain implementation and its problems

The Fourier transform of the analytic signal $x_a(t)$ can be expressed as:

$$X_a(f) = \begin{cases} 2X(f), & f > 0, \\ X(f), & f = 0, \\ 0, & f < 0 \end{cases} \qquad (8.5)$$
$$= 2u(f)X(f)$$

where $X(f)$ is the Fourier transform of $x(t)$ and $u(f)$ is the step function.

In the Hilbert implementation in Section 5.2.2, a schematic for the frequency domain HT based envelope analysis is presented in Figure 8.11. The HT window function applied in the frequency domain actually forces the spectrum of $x(t)$ to become the spectrum of $x_a(t)$.

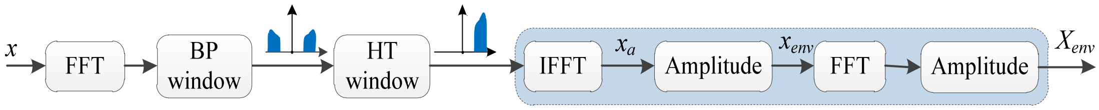


Figure 8.11 Schematic of frequency domain HT based envelope analysis

This operation is straightforward and easy to understand, but it brings edge distortions to the result. Figure 8.12(a) shows an envelope detection result for applying a 128-point window (see Eq. (5.2)) in the frequency domain. Obvious distortions can be clearly witnessed in the beginning and end of the calculated envelope signal, which is called the end effect. This problem appears in a digital filtering or discrete Fourier transform due to an incomplete data periodicity when the waveform has not completed a full cycle within its period of the analysis [193]. With the increase of FFT size, the influence of this effect can be reduced. However, the size of FFT calculation on an MCU is limited as aforementioned.

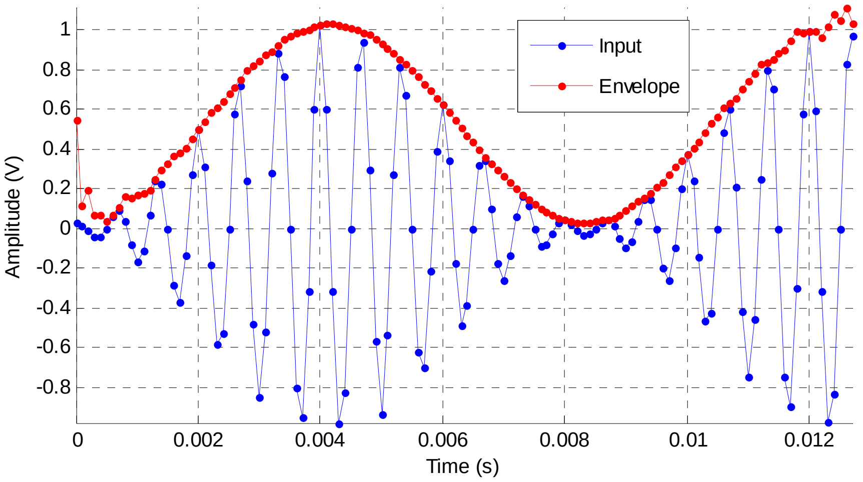


Figure 8.12 Envelope detection results for a 128-point signal in the frequency domain

According to the convolution theorem [178], multiplying the window function in the frequency domain equals to circular convolving the time domain signal with the inverse FFT of the window function. The impulse response of the 128-point window function is shown in Figure 8.13 by presenting its real and imaginary part separately. The real part of the impulse response has a single value of 1 in the middle with the other values being zero, thus, it results in itself after passing through the filter. The imaginary part of the impulse response is the actual HT kernel, which produces the HT of the input signal. As can be seen in Figure 8.13(b), the imaginary component extends its value until both ends, in which case it wraps around itself in the convolution and time-domain aliasing problem comes in [183].

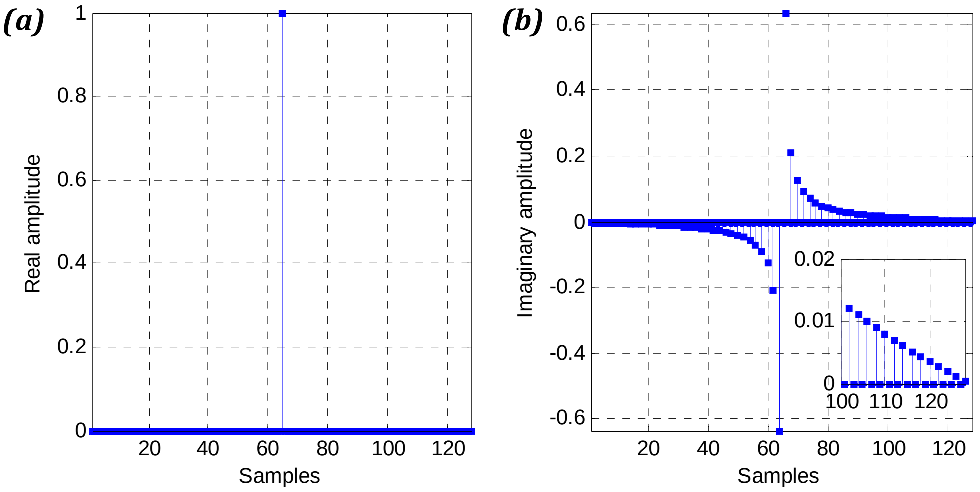


Figure 8.13 The impulse response of the 128-point window function: (a) real part and (b) imaginary part with partially magnified as an inset

8.2.3 Time domain implementation

The overlap processing method proposed in Chapter 6 is one effective method to solve the aliasing problem and another method is to design a Hilbert transformer with finite length while it still approaches the ideal HT transfer function. The designed HT transformer can be either FIR or IIR type and the FIR type filter is preferred due to its linear phase response performance.

HT introduces a 90-degree phase shift to all sinusoidal components. In the discrete-time periodic-frequency domain, the transfer function of HT is specified as follows:

$$H[j\omega] = \begin{cases} -j, & 0 < \omega < \pi \\ j, & -\pi < \omega < 0 \end{cases} \tag{8.6}$$

Thus, the impulse response of the transfer function can be calculated through inverse Fourier transform:

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(j\omega)e^{j\omega n}d\omega = \begin{cases} \frac{2sin^2(\pi n)}{\pi n}, & n \neq 0 \\ 0, & n = 0 \end{cases} \tag{8.7}$$

It can be seen that the impulse response is with infinite length, which is the cause of the time-domain aliasing problem. By multiplying the ideal impulse response $h[n]$ with a window function $w[n]$ having a finite support (let's say from $n = -N\ to\ N$), then the resulting function $w[n]h[n]$ yields an approximate magnitude response $H_\omega(j\omega)$ that has a smooth transition between negative and positive frequencies as well as ripples in both regions. The height of the ripples can be reduced by selecting the window function wisely, while the transition bandwidth is inversely proportional to the window length.

As an example, an FIR type HT transformer with 31 points is designed in Figure 8.14(a). Suppose it is applied to process an array of data sampled at 10 kHz, the amplitude response is presented in Figure 8.14(b), which shows the filter produces a flat pass-band from 0.5 kHz to 4.5 kHz and the pass-band ripple is measured as less than 0.05 dB. Suppose a carrier signal of 1500 Hz is modulated by a sinusoidal signal of 80 Hz, by passing through this designed 31-tap FIR filter, its HT output, and the computed envelope is shown in Figure 8.15. As it shows, the obtained envelope signal is smooth even at both ends.

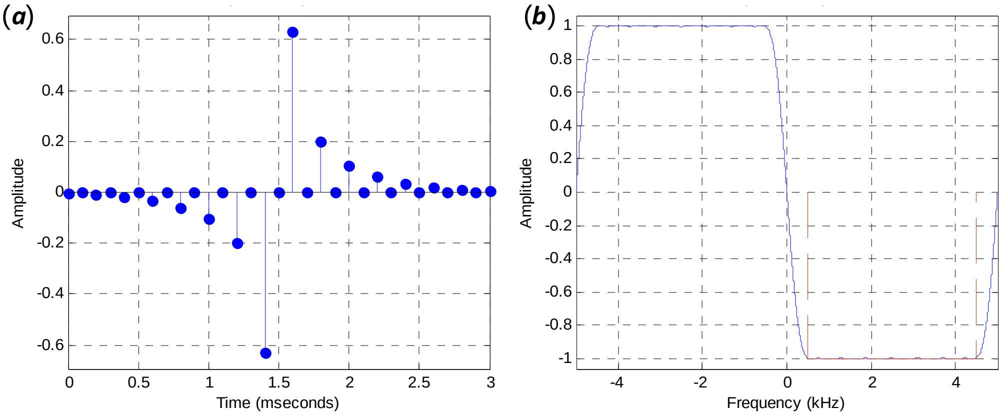


Figure 8.14 An FIR Hilbert transformer with 31-tap: (a) impulse response and (b) zero-phase response

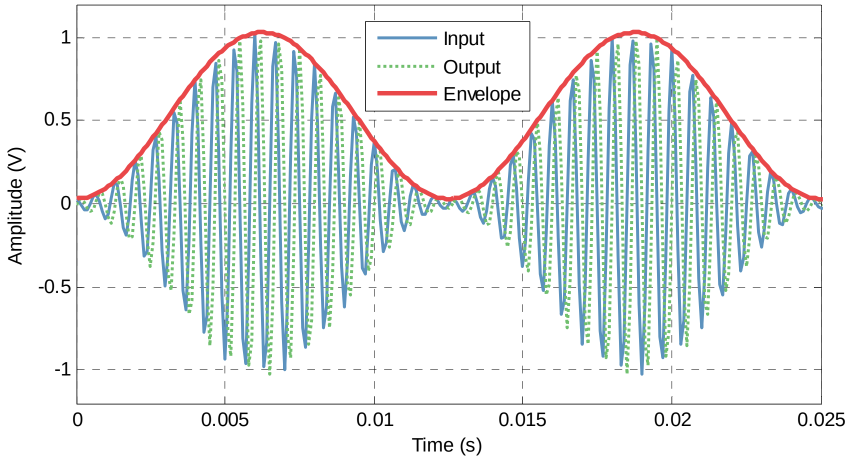


Figure 8.15 The envelope computed using a 31-tap FIR Hilbert transformer

Although Hilbert transformer minimises edge effects, it has some restrictions. The performance of a Hilbert transformer, including transition region and ripples in the passband, is affected by the filter length. With the increase of filter length, the transition region becomes narrower and the ripples in the passband are smaller, however, the computation time for applying the transformer increases as well. Thereby, a good balance between transformer performance and computation time is required.

A schematic of time domain HT based envelope analysis is illustrated in Figure 8.16. It shows two FIR filters are required for this implementation. The pass band of the band-pass filter needs to be updated for different bearing tests and a wider passband of the Hilbert transformer is also required for accommodating various band-pass filter conditions.

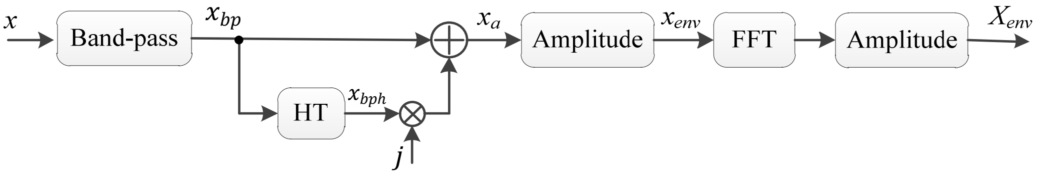


Figure 8.16 Schematic of time domain HT based envelope analysis

In a comparison of applying a band-pass filter and Hilbert transform directly in the frequency domain as discussed in Section 7.3.4, the FIR based band-pass filter and Hilbert transformer can be more efficient in terms of computation time for short filters. However, it can be even more time consuming on the condition long filters are required.

8.2.4 FFT convolution

When long filters are required, the time domain implementation method in Section 8.2.4 can be time-consuming. On this occasion, this operation can be accelerated by implementing the FIR filter in the frequency domain, which is called FFT convolution (also fast convolution) based on the convolution theorem [178].

A diagram for FFT convolution is presented in Figure 8.17, where the input signal $x[n]$ and the filter coefficient $h[k]$ are firstly converted to the frequency domain via N-point FFT operations. Then, the obtained $X[m]$ and $H[m]$ are multiplied and their results are converted back to time domain through an N-point IFFT operation. Note that FFT of the FIR filter only need to be computed once in advance and stored in the memory to save computation time.

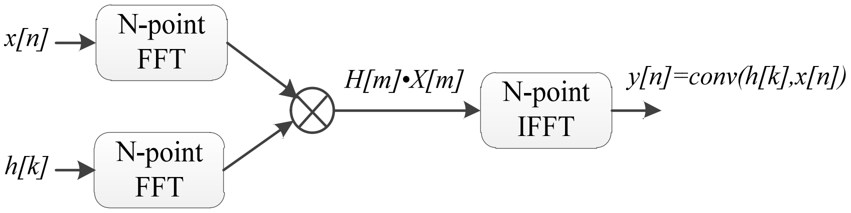


Figure 8.17 Schematic of fast convolution process [178]

The computational complexity of FFT convolution is approximate $O(Nlog_2N)$ instead of $O(N^2)$ in direct FIR filter [195]. Clearly, very small FIR filters are more efficient while fast convolution will win as the FIR filters get longer. A typical efficiency crossover point is at 25–30 filter coefficients, depending on the CPU structure and FFT efficiency, etc. [195]

A benchmark for the MCU employed in this thesis is presented in Figure 8.18, which shows the average time consumption for one effective data point by the FIR and FFT convolution methods by only taking the valid output data number into consideration, illustrating that the crossover point is at around 60. Furthermore, it can observe that the time consumption for the FIR filter method increases faster than the FFT convolution method, which confirms their computational complexity discussed above.

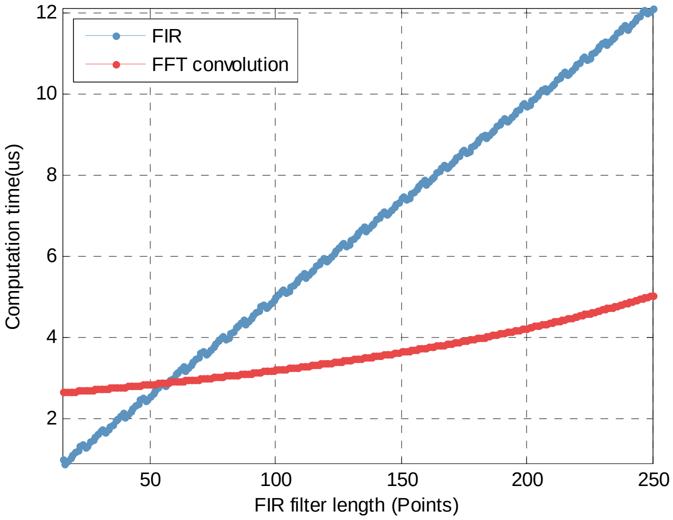


Figure 8.18 Average time consumption for one effective data point by FIR and FFT convolution

A schematic for the FFT convolution HT based envelope analysis is presented in Figure 8.19. Its only difference with frequency domain HT envelope analysis in Figure 8.11 is the HT window applied. For the FFT convolution based method, the HT window can more effectively minimise the aliasing problem in the obtained x_a. Similarly, the band-pass window can also cause aliasing problem and a similar window function can be produced from band-pass FIR filter coefficients.

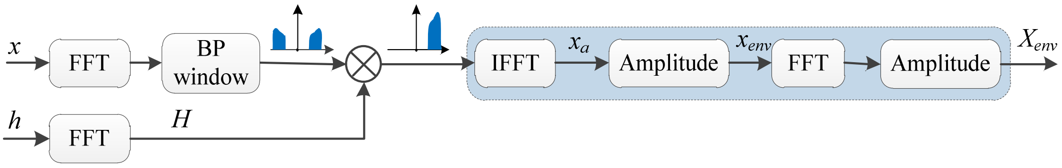


Figure 8.19 Schematic of FFT convolution HT based envelope analysis

Note that the implementation of FFT convolution is restricted by memory size on the processor. For the MCU employed in this thesis, the maximum size for FFT calculation is restricted to 2048 point as discussed in Section 5.2.2. In this way, the input sequence $x(n)$ can be partitioned into multiple frames, which are then processed individually. Of note, part of the results yielded have time-domain aliasing errors due to the circular convolution in the FFT computation, which can be minimised by using two techniques: overlap-save or overlap-add method [178], [195].

8.3 Spectral convolution and spectral correlation

Recall the implementation of HT in the frequency domain for envelope analysis in Figure 8.11. It can be observed that two forward FFT and one inverse FFT are involved during the implementation. Recall the convolution theorem that the production in one domain equals to the convolution in the other domain, it reminds us of one question: is it possible to calculate the envelope spectrum using one single FFT without the inverse FFT and the second forward FFT? If this can be achieved, the computation time can be significantly reduced since the FFT and IFFT operation is rather expensive in terms of computation time.

8.3.1 Spectral correlation for squared envelope spectrum

In Figure 8.11, the analytic signal x_a is obtained after the IFFT operation. By calculating the amplitude of the analytic signal x_a, the envelope of the input signal x can be obtained. Here, the amplitude is computed by the square root of the product of x_a with its conjugate x_a^* in the time domain. In the frequency domain, as shown in Figure 8.20, the product between x_a and x_a^* is equivalent to the convolution of X_a and G_a, where X_a and G_a are the FFT of x_a and x_a^*, respectively; $\bullet$ is the inner product operator and $*$ is convolution operator. The result of convolution of X_a and G_a produces X_{envs}, which is the FFT of squared envelope x_{env}^2. In [49], the authors show that analysing the squared envelope can improve the SNR ratio when SNR is greater than unity.

$$\begin{tikzcd}
x_a \arrow[d, "\text{FFT}"] & \bullet & x_a^* \arrow[r] \arrow[d, "\text{FFT}"] & x_{env}^2 \arrow[r, "\sqrt{}"] & x_{env} \arrow[d, "\text{FFT}"] \\
X_a & * & G_a \arrow[r] & X_{envs} \arrow[u, "\text{IFFT}"] & X_{env}
\end{tikzcd}$$

Figure 8.20 The amplitude calculation of analytic signal x_a in time domain and frequency domain

For X_a and G_a, they only contain either the positive or negative frequency components. According to the definition of $X_a(f)$ in Eq. (8.5), the Fourier transform of $x_a^*(f)$ can be expressed as:

$$G_a(f) = 2u(-f)X^*(f) = X_a^*(-f) \tag{8.8}$$

This shows that the Fourier transform $G_a(f)$ of $x_a^*(t)$ can be easily obtained from the spectrum of $X(f)$. Thereafter, a convolution operation between $G_a(f)$ and $X_a(f)$ gives the spectrum of the envelope directly instead of using one additional forward FFT and inverse FFT in Figure 8.11. We can look at this operation in more detail. Recall the definition of cross-correlation and convolution:

$$f[n] \otimes g[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m]g[m-n] = \sum_{m=-\infty}^{\infty} f^*[m-n]g[m] \tag{8.9}$$

$$f[n] * g[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m] \tag{8.10}$$

where $\otimes$ and $*$ are cross-correlation operator and convolution operator, respectively. Suppose $s[n] = f^*[-n]$, the convolution between $s[n]$ and $g[n]$ can be expressed as:

$$\begin{aligned} s[n] * g[n] &= \sum_{m=-\infty}^{\infty} s[n-m]g[m] \\ &= \sum_{m=-\infty}^{\infty} f^*[m-n]g[m] \\ &= f[n] \otimes g[n] \end{aligned} \tag{8.11}$$

Thereby, the convolution between $G_a[n]$ and $X_a[n]$ become:

$$\begin{aligned} G_a[n] * X_a[n] &= X_a^*[-n] * X_a[n] \\ &= X_a[n] \otimes X_a[n] \end{aligned} \tag{8.12}$$

This result is useful, which means the squared envelope spectrum can be simply computed from the cross-correlation of X_a with itself, i.e. the correlation of X_a. Thereafter, the convolution and correlation operation on the spectrum of the analytic signal are equivalent, as shown in Figure 8.21. In comparison, the conjugate operation in correlation is more straightforward to understand than the folding operation in convolution. Usually, the correlation is used to represent similarity of a signal to a delayed version of itself. Here, the correlation of X_a can be interpreted as finding the difference frequency components in X_a.

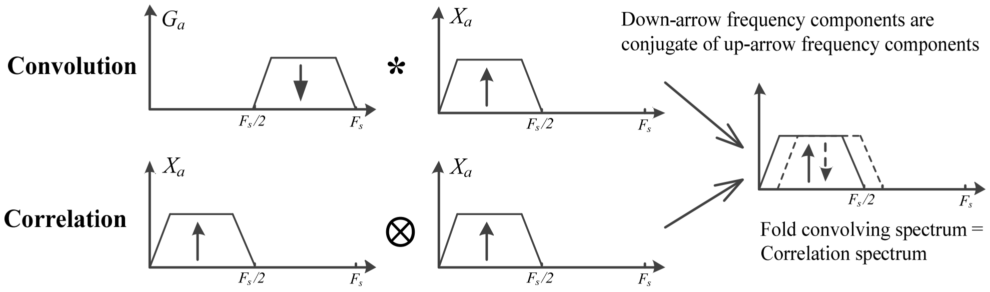

Convolution
Correlation
G_a
X_a
$F_s/2$
F_s
*
⊗
Down-arrow frequency components are conjugate of up-arrow frequency components
Fold convolving spectrum = Correlation spectrum

Figure 8.21 Illustration of convolution and correlation operation of the spectrum of the analytic signal

On this basis, the diagram in Figure 8.11 can be simplified as Figure 8.22, which obtains the spectrum of the squared envelope instead of the spectrum of the envelope in Figure 8.11. As usually only a partial band in the spectrum of an analytic signal is selected for correlation, the selected band can be shifted to the low-frequency band and perform correlation with a lower sampling frequency [49]. Thereafter, the correlation of X_a can be updated as:

$$\begin{aligned} X_a[n] \otimes X_a[n] &= \sum_{m=0}^{N} X_a^*[m] X_a[m-n] \\ &= \sum_{m=nfL}^{nfH} X_a^*[m] X_a[m-n] \end{aligned} \tag{8.13}$$

where nfL and nfH is the index for low and high cut-off frequency in the band-pass filter, respectively.

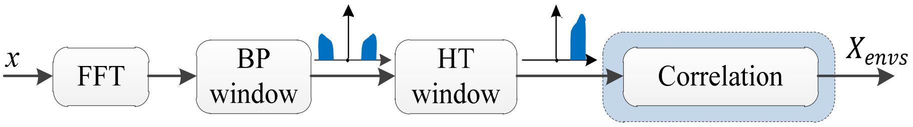


Figure 8.22 Schematic of spectral correlation based envelope analysis

Furthermore, for bearing fault diagnosis, only a small portion of the envelope spectrum is of interest, hence there does not need a through computation of correlation. The data points effective for squared envelope spectrum analysis (N_{envs}) can be calculated as:

$$N_{envs} = N f_{em} / F_s \tag{8.14}$$

where N is the FFT size, f_{em} is the interested maximum envelope frequency and F_s is the sampling frequency.

These indicate the method in Figure 8.22 can be very efficient in terms of computation time. To demonstrate the spectral correlation based method, the modulation signal in Figure 8.3 is also processed by this method. Below are the processing steps:

1) Perform a 2048-points FFT on the modulation signal.
2) Extract the frequency components from 1000 Hz to 2000 Hz (index from 64 to 129) and double its amplitude to get the analytic spectrum.
3) Perform a correlation operation on the obtained analytic spectrum to obtain the squared envelope spectrum.

The calculated squared envelope spectrum through spectral correlation is shown in Figure 8.23. This highlights that the results are a good match with the convolution for the full analytic spectrum, verifying the equality of spectral correlation and spectral convolution on the analytic spectrum.

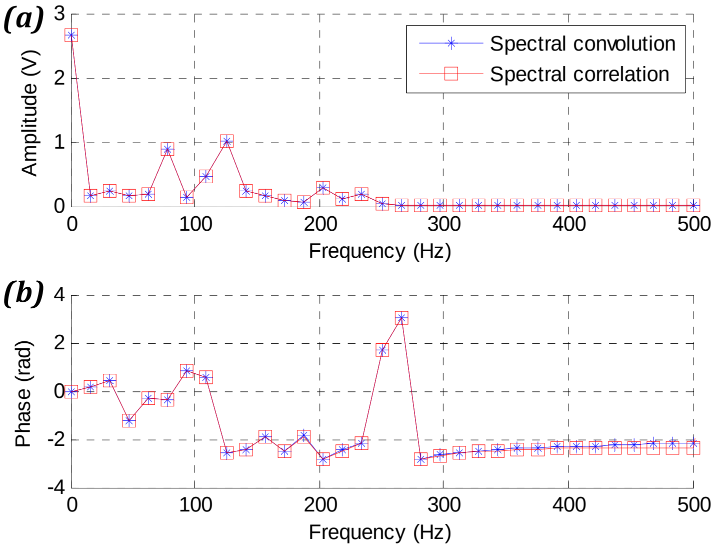


Figure 8.23 Comparison between spectral convolution and spectral correlation: (a) amplitude and (b) phase

Moreover, the envelope signal is obtained by performing an inverse FFT and a following square root operation on the spectrum of spectral correlation, which is presented in Figure 8.24 in comparison with the raw modulation signal. It shows the computed envelope from spectral correlation has a good match with the upper outline of the modulation signal.

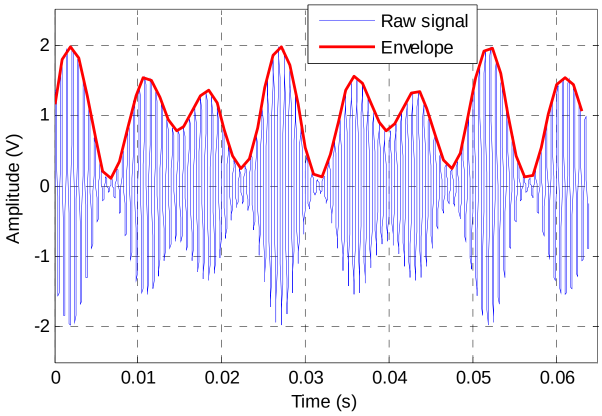


Figure 8.24 Raw modulation signal and envelope obtained from spectral correlation

8.3.2 Squared rectifier in the frequency domain

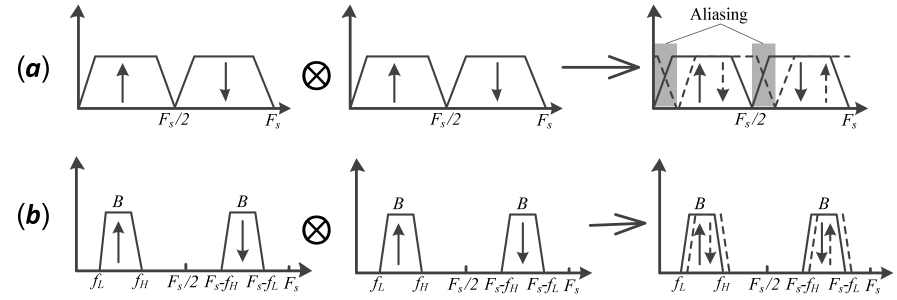
Since the inner production of an analytic signal with its conjugate is equivalent to the spectral correlation of the analytic signal, another question is presented: how to understand the inner production of a real-valued signal, i.e. square operation, in the frequency domain?

According to convolution theorem, a square process in the time domain is equivalent to the spectrum being convolved with itself in the frequency domain. For the spectrum $X[n]$ of a real-valued signal x, it has the property of conjugate symmetric [178], i.e. $X[n] = X^*[-n]$, thereby, the convolution of $X[n]$ with itself is equivalent to its correlation:

$$X[n] * X[n] = X[n] \otimes X[n] \tag{8.15}$$

The convolution of a full baseband signal is presented in Figure 8.25(a) using an equivalent correlation. It can be seen that aliasing is brought in and the envelope spectrum includes both sum and difference frequency components. Note that, the correlation is actually a circular one due to the circular convolution in Fourier transform and the first aliasing area in Figure 8.25(a) is caused by this phenomenon. Furthermore, it can be observed that difference frequency components come from the correlation of positive or negative frequency components while sum frequency components are contributed by the cross-correlation between positive and negative ones.

On the occasion when a signal is band-pass filtered (band-pass filtered squared rectifier), as shown in Figure 8.25(b), the aliasing problem can be effectively avoided. It has been proved that through proper zero-padding on both lower and the upper side of the passband in the frequency domain, the band-pass squared rectifier can achieve the same results as Hilbert transform [49]. In practice, there is no need to transform signals to the frequency domain, zero-pad the spectrum and then perform correlation, as this operation will have the same result apart from there being more computation requirement than frequency domain HT.



Spectral correlation.
Down-arrow frequency components are
conjugate of up-arrow frequency components

Correlation spectrum

Figure 8.25 Equivalent spectral correlation of time domain squared operation: (a) full baseband signal and (b) band-pass filter signal

For a band-pass filtered signal, its square operation in the time domain can avoid the aliasing problem on the condition that its spectrum is already zero-padded as discussed above, i.e. its lower pass band f_L and upper pass band f_H can satisfy the following criteria:

$$\begin{gathered} f_{\mathrm{L}} \geq (f_{\mathrm{H}} - f_{\mathrm{L}})/2 \\ F_s/2 - f_H \geq (f_{\mathrm{H}} - f_{\mathrm{L}})/2 \end{gathered} \tag{8.16}$$

where F_s is the sampling frequency. Recall the squared rectifier in Section 8.1, the spectrum of the simulated signal only occupies a limited band, which can satisfy Eq. (8.16). Thus, its sum and difference frequency components are well separated in the spectrum of the squared signal.

In bearing fault diagnosis, only low-frequency components in the envelope spectrum are of interest as discussed in Section 5.3. This indicates aliasing in the high-frequency range of a squared signal is acceptable as they are ineffective for bearing fault diagnosis and can be easily eliminated with a high-pass filter.

Supposing the maximum interested frequency component in the envelope spectrum is f_{em}, the band-pass filter restriction in Eq. (8.16) becomes:

$$\begin{gathered} f_{\mathrm{L}} \geq f_{em}/2 \\ F_s/2 - f_H \geq f_{em}/2 \end{gathered} \tag{8.17}$$

As for a bearing vibration with the sampling frequency at 32 kHz, discussed in this thesis, the frequency range of interest for an envelope spectrum is well within 500 Hz. According to Eq. (8.16), a band-pass filter whose pass band falls inside the range between 250 Hz and 15,750 Hz can easily avoid aliasing problem within the interested envelope spectrum.

Thereby, a flow chart can be designed as Figure 8.26 for implementing band-pass squared rectifier based envelope analysis.

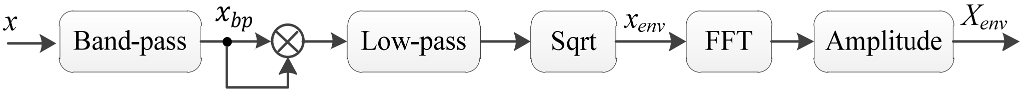


Figure 8.26 Schematic of band-pass squared rectifier based envelope analysis

8.4 Comparison and benchmark

In the previous three sections, different envelope detection methods are studied and discussed. These methods have their own advantages, disadvantages and also there are some connections between some of them. It has shown the band-pass squared rectifier is equivalent to the Hilbert transform in some conditions and the short-time RMS can be considered as a simplified squared rectifier. In this section, these methods will be compared by dividing them into two categories: time domain and frequency domain.

8.4.1 Time domain methods comparison

The time domain envelope analysis methods include a band-pass squared rectifier, short-time RMS, and time domain HT, as illustrated in Figure 8.27. For these methods, a raw vibration signal firstly passes through a band-pass filter to select a demodulation band that has high SNR; then envelope signal is computed, which mainly differentiates these methods, and finally the envelope spectrum is obtained with a forward FFT and an amplitude computation. Note that the high-pass filter of short-time RMS method in Figure 8.8 is changed to a band-pass filter, which can provide better SNR. Similar to the short-time method, a down-sampling process can be added in the band-pass squared rectifier and timed domain HT to speed up the FFT calculation.

The main difference for these three methods lies in the envelope detection part. Time domain HT is the most accurate envelope detection method and also the most expensive one in terms of computation time. The band-pass squared rectifier can achieve very much similar results with time domain HT through a proper selection of band-pass filter and low-pass filter as discussed in Section 8.3.2. The short-time RMS can be considered as a simplified band-pass squared rectifier with faster speed.

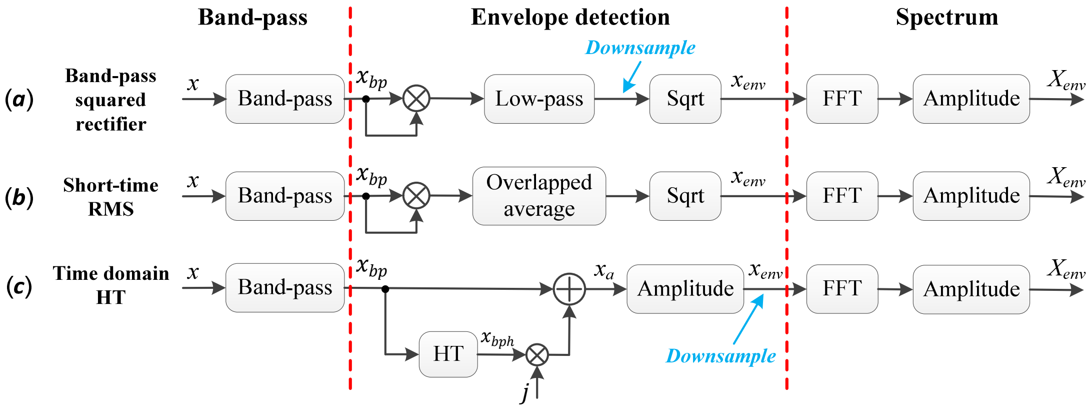


Figure 8.27 Comparison of time domain envelope analysis methods

To evaluate the performance of these three methods, they are employed to process the same vibration signal from bearing with inner race fault. The sampling rate of the signal is at 10 kHz and 2048-points of data are processed. The raw vibration signal and its spectrum are shown in Figure 8.28(a). An 81-tap FIR filter with a pass band from 250 Hz to 4750 Hz is utilised to filter the raw signal. The filtered signal and its spectrum are presented in Figure 8.28, which shows the high amplitude low-frequency components in the raw spectrum are filtered out.

To illustrate the interference of low and high-frequency components on the envelope spectrum, the squared rectifier is performed on both the raw signal (full band) and band-pass rectifier and the computed envelope and spectrum are compared in Figure 8.29. Obvious aliasing components can be clearly observed in the envelope spectrum of the raw signal, which has buried the inner race fault frequency and its harmonics, making it not reliable for fault diagnosis. Meanwhile, in the spectrum of the band-pass filtered one, the first three harmonics of inner race fault can be clearly observed, which validates the existence of inner race fault. This shows the effectiveness of the band-pass filter for minimising aliasing problems.

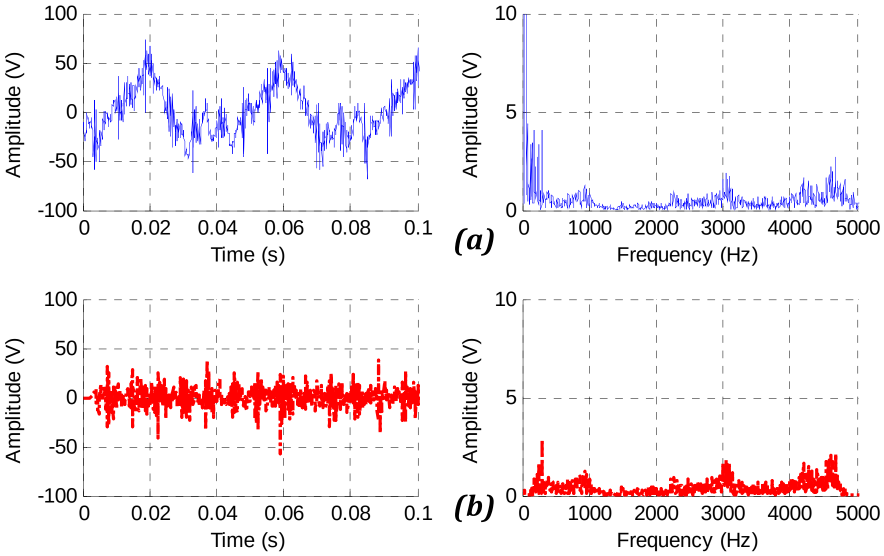


Figure 8.28 A vibration from bearing with inner race fault in both time and frequency domain: (a) raw signal (b) band-pass filtered signal

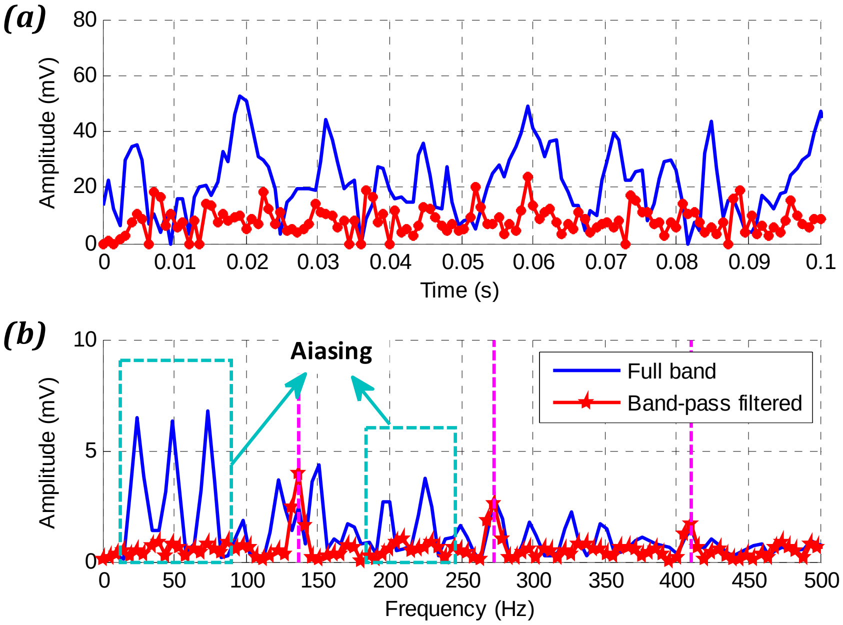


Figure 8.29 Comparison of squared rectifier on raw signal (full band) and band-pass filtered signal: (a) envelope and (b) envelope spectrum

Furthermore, the band-pass filtered signal is also processed by the short-time RMS and time domain HT methods. A 63-tap Hilbert transformer with a transition zone of 500 Hz is applied in the time domain HT method. The processed results of these three methods are compared in Figure 8.30, including both envelope and its corresponding spectrum. The first three harmonics of the inner fault can be clearly observed in all three methods.

It can be observed that the amplitude for band-pass squared rectifier and time domain HT are very close, which further verifies the effectiveness of minimising aliasing for band-pass squared rectifier just as the time domain HT one. Also, it shows the amplitude of the harmonics for short-time RMS method is lower than those from the other two methods, which can be explained by the strong filtering effect of the overlapped average in the short-time RMS.

In terms of computation speed, their difference mainly lies in the envelope detection part as illustrated in Figure 8.31, which is measured on the condition of MCU running at 80 MHz. It shows short-time RMS method is the fastest, followed by band-pass squared rectifier and time domain HT is the most time consuming. Note that the low-pass filter and down-sampling process in band-pass squared rectifier are simplified as an N-tap average operation in short-time RMS. Hence, short-time RMS requires much less computation time than the corresponding operation in the band-pass squared rectifier.

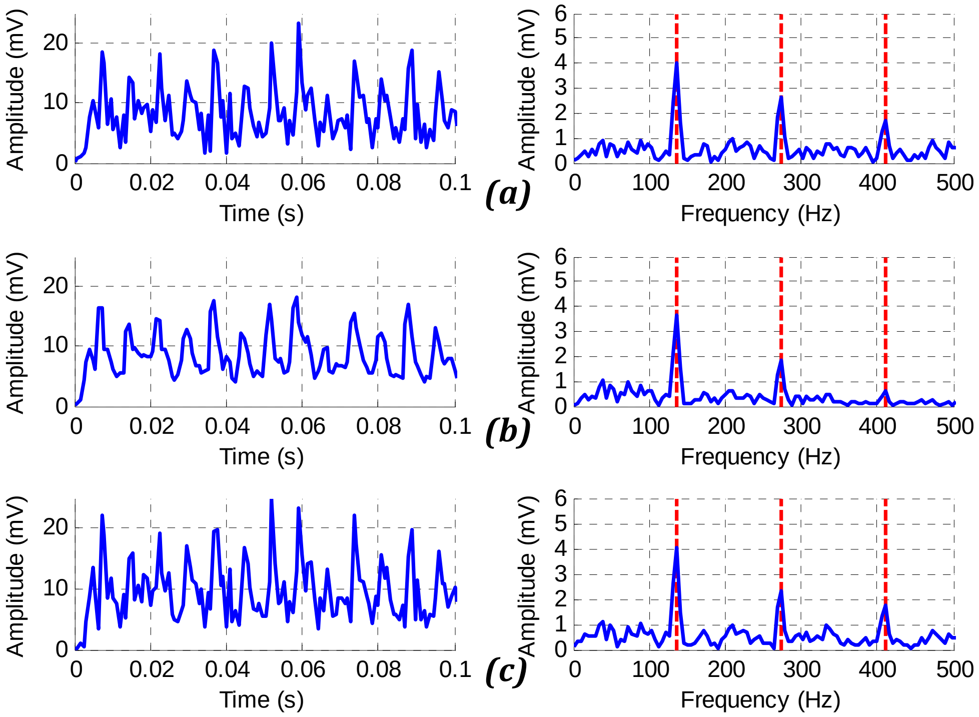


Figure 8.30 Calculated envelope and its spectrum by time domain methods: (a) band-pass squared rectifier (b) short-time RMS and (c) time domain HT

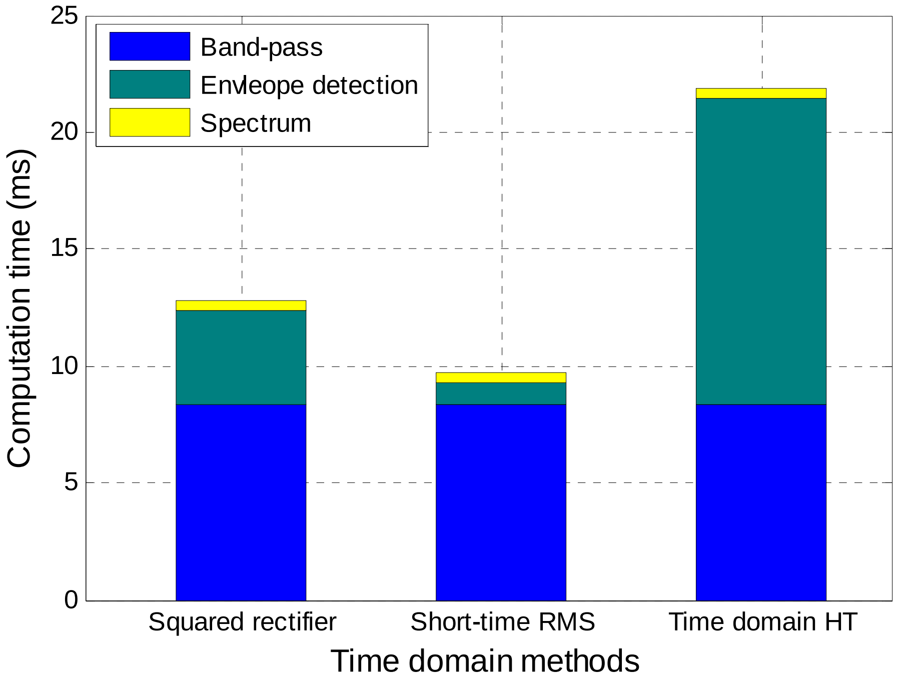


Figure 8.31 Time consumption for three time-domain methods with frame size of 2048 points

In this section, it has shown that a band-pass squared rectifier can well minimise the aliasing influence similar as the time domain HT. In practice, a high-pass filter can be used instead of a band-pass filter in band-pass squared rectifier as the high-frequency components that might cause aliasing problems have usually been attenuated before fed into ADC for satisfying Nyquist sampling theorem.

8.4.2 Frequency domain methods comparison

The frequency domain envelope analysis methods include frequency domain HT, FFT convolution HT, and spectral correlation, as illustrated in Figure 8.32. For these methods, a forward FFT operation is firstly conducted on the raw vibration signal and both band-pass selection and Hilbert transform are realised in the frequency domain. It can be seen that the frequency domain HT and FFT convolution HT are very similar and their only difference is the applied HT window. As for the envelope spectrum calculation, the spectral correlation obtains the squared envelope spectrum directly in the frequency domain while the other two methods convert spectrum back to time domain for envelope spectrum calculation.

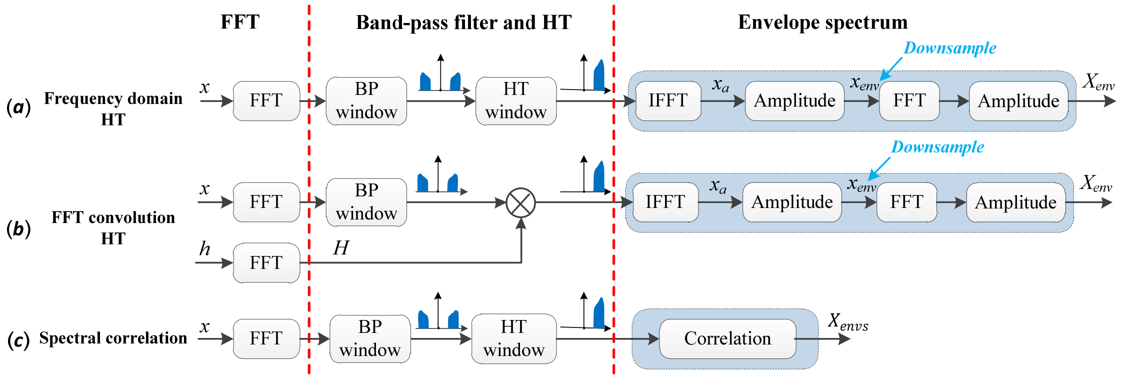


Figure 8.32 Comparison of frequency domain envelope analysis methods

To compare their performance, the three frequency domain methods are employed to process the same vibration signal in Section 8.4.1. The same Hilbert transformer in time domain HT method is used in the FFT convolution HT and its response is compared with the ideal HT response in Figure 8.33. It can be observed that the ideal HT has a sharper transition zone than that of the Hilbert transformer.

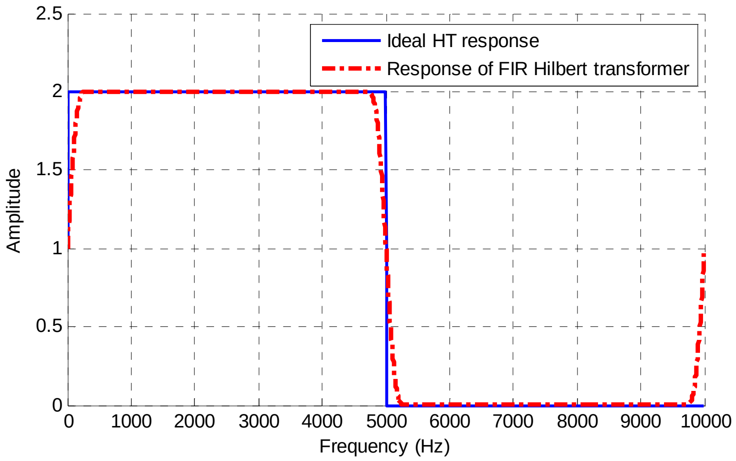


Figure 8.33 Amplitude response comparison between an ideal HT and an FIR Hilbert transformer (63-tap and 1 kHz transition band)

The computed envelope and corresponding spectrum for these methods are presented in Figure 8.34. Note that results for spectral correlation in Figure 8.34(c), the squared envelope is obtained through an inverse FFT on the squared envelope spectrum. It can be seen that the first three harmonics of the inner fault can be clearly observed in all three methods. The spectrum from frequency domain HT and FFT convolution HT are similar and also very close to the spectrum from time domain HT in Figure 8.30(c). This shows the conformity for time domain and frequency domain implementation. In comparison, the amplitude from squared envelope spectrum in Figure 8.34(c) is higher than that in Figure 8.34(a) and Figure 8.34(b) but the peakness of their three harmonics are very similar. Furthermore, the shape of the squared envelope signal from spectral correlation is very similar to the envelope in frequency domain HT and FFT convolution HT, which again verifies the correctness for spectral correlation.

For their computation speed, frequency domain HT and FFT convolution HT are expected to be very similar, as shown in Figure 8.35, which shows their time consumption for processing a frame of data with 2048 points when the MCU running at 80 MHz. Their main difference lies in applying the HT window. Applying the HT window in FFT convolution HT is a bit more time consuming as it involves complex type multiplication while in frequency domain HT, this can be easily implemented by filling zeros and scaling operations. Note that the FFT convolution HT needs a very large buffer to store the FFT of the Hilbert transformer. Furthermore, it can be noticed that the computation time for frequency domain HT and FFT convolution HT is less than the time domain HT in Figure 8.31. This shows better computation performance in the frequency domain over time domain when longer filter length is required.

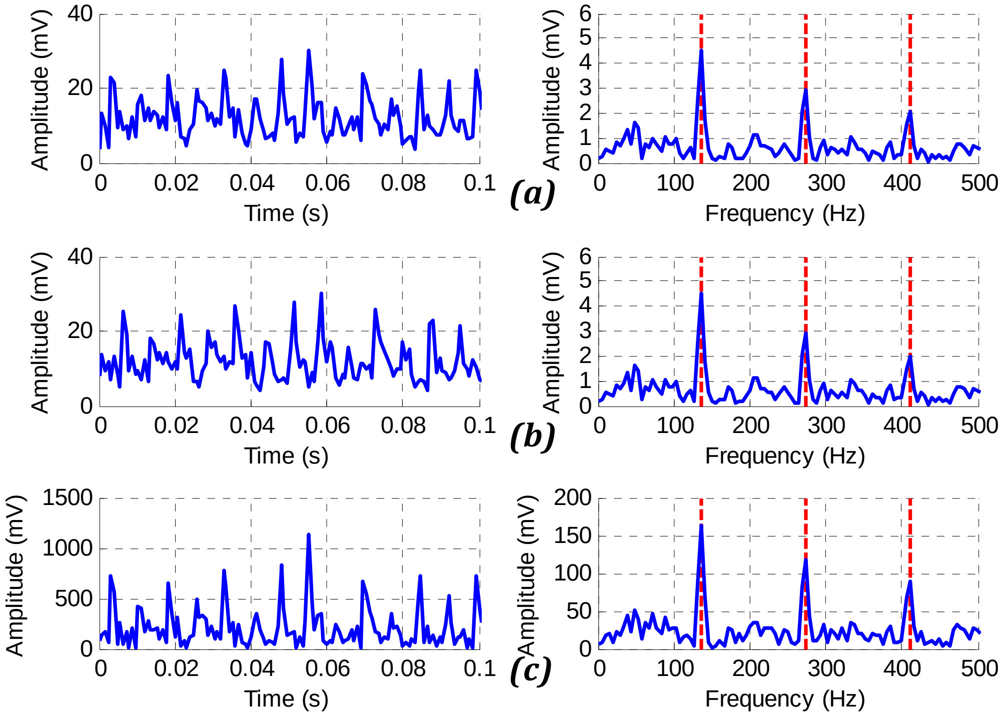


Figure 8.34 Calculated envelope and its spectrum by frequency domain methods: (a) frequency domain HT, (b) FFT convolution HT and (c) squared envelope and its spectrum by spectral correlation

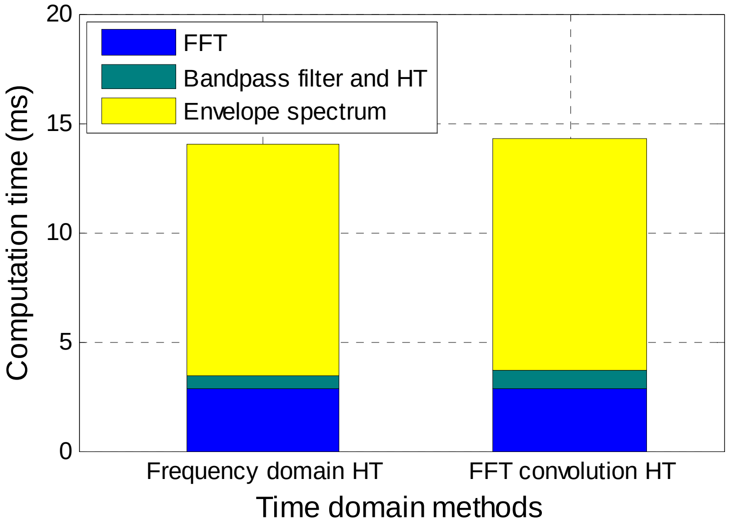


Figure 8.35 Computation time comparison between frequency domain HT and FFT convolution HT for a frame of data with 2048 points

For computation speed of spectral correlation, it is affected by the data points involved with spectral correlation, i.e. data length from nfH to nfL in Eq. (8.13), and data points useful for squared envelope spectrum analysis N_{envs} in Eq. (8.14). This means the pass band width, FFT size and sampling frequency have an influence on the computation time of spectral correlation. The computation time for spectral correlation for various bandwidths, FFT size with sampling frequency at 10 kHz and 32 kHz are presented in Figure 8.36. It can be seen that the computation time increases with FFT size and pass band width. This can be explained by that larger FFT size and wider pass band allow more data points involved for spectral correlation than shorter FFT size and narrower pass band do. For the same pass band width and FFT size, the computation for sampling frequency at 10 kHz is much higher than that for 32 kHz, which is due to more data points being involved for lower sampling frequency condition.

In comparison with frequency domain HT and FFT convolution HT, the computation time of spectral correlation is more in some conditions but can be very efficient on the condition of a narrow pass bandwidth. Recall the band-pass filter selection results in Section 7.4, the optimum band-pass filter usually has a bandwidth of only 1 kHz or 2.67 kHz. This is due to that the fault information is modulated on the resonances of bearing structure. According to Figure 8.36(b), their computation time are less than 10 ms for 2048-point FFT with the sampling frequency at 32 kHz, which is much less than the computation time of frequency domain HT and FFT convolution HT in Figure 8.35.

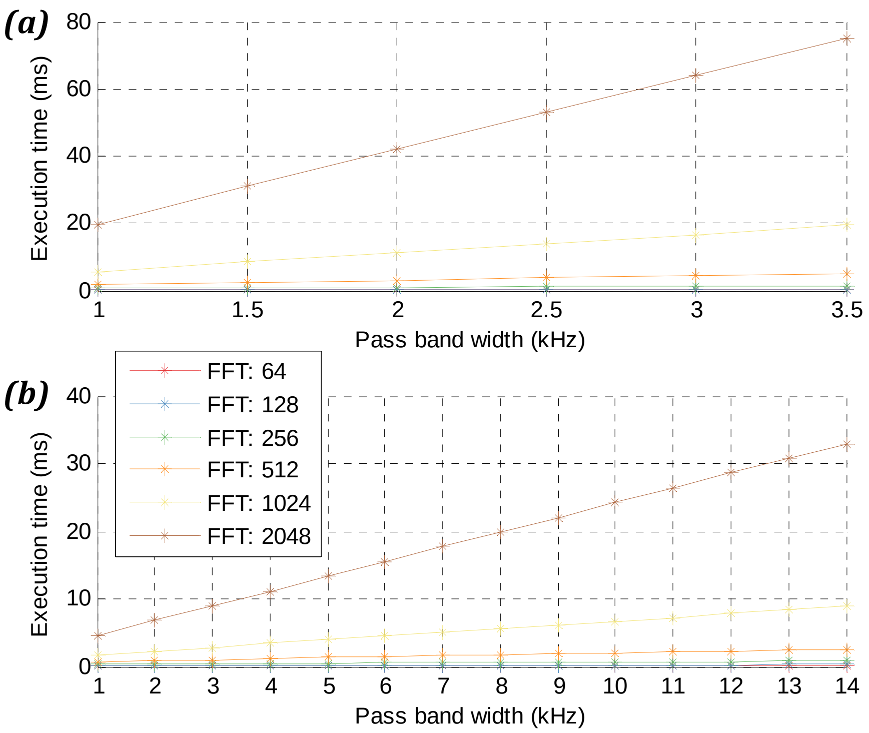


Figure 8.36 Computation time of spectral correlation for different FFT size with fixed pass band of 1 kHz: (a) sampling frequency at 10 kHz and (b) sampling frequency at 32 kHz

8.5 Implementation of short-Time RMS and spectral correlation for high spectrum resolution

From the comparison in Section 8.3, a better understanding of the demodulation process is gained both in time domain and frequency domain. It shows all the methods can successfully extract the simulated features and there is no obvious accuracy difference on these methods, but their computation speed varies a lot. In this section, the fastest methods in the time domain and frequency domain are employed for cascaded frame processing to achieve a high-resolution envelope spectrum, respectively.

8.5.1 Implementation of cascaded short-time RMS

As discussed in Section 8.4.1, the short-time RMS is the fastest method in the time domain and it can clearly reveal the simulated faults although higher harmonics in short-time RMS are attenuated slightly more than the other two methods in the time domain. Thereafter, it is employed for cascaded processing to obtain an envelope spectrum with high resolution.

The data processing diagram for cascaded short-time RMS is presented in Figure 8.37, in which the collected data are processed frame by frame with the size of 512 points. The data is firstly filtered by an 81-tap FIR type band-pass filter. Note that the coefficients for the band-pass filter in Figure 8.37 can be updated from the host computer in the configuration mode introduced in Section 7.3.2. Then, the short-time RMS of the filtered signal is computed and stored in a buffer. Note that for the first frame of data, 13 effective feature data are produced due to the first 40 data of the FIR filter being invalid and thus the short RMS calculation actually starts from the third step.

From the second frame until the second last frame, last 32 points from the previous frame are concatenated in front of the current frame of data for the short-time RMS calculation and hence they produce 16 points of effective feature data. For the last frame, only 3 points of feature data are required for filling the 2048-point buffer. After the 2048-point buffer is full, the spectrum of short-time RMS features is computed by FFT calculations.

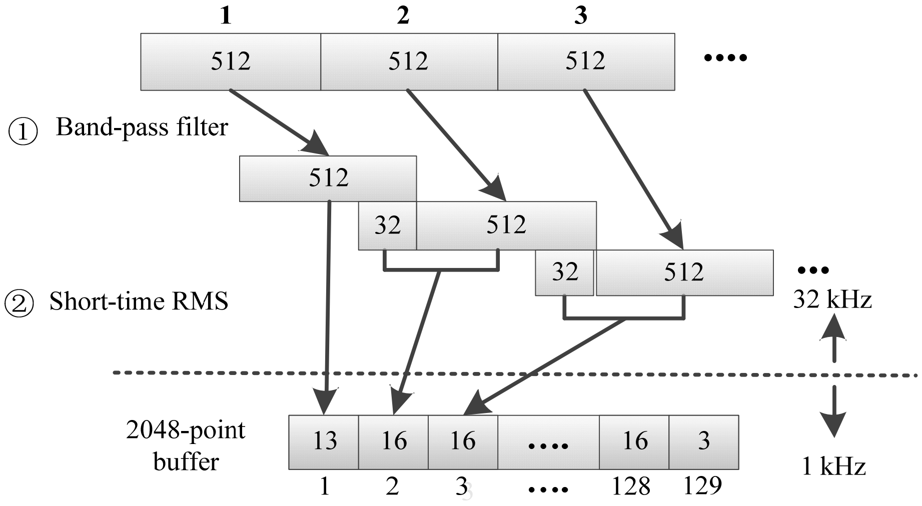


Figure 8.37 Diagram for cascaded short-time RMS

8.5.2 Verification of short-time RMS implementation

To verify the implementation of cascaded short-time RMS, a modulating signal is processed by both MCU and Matlab and their results are compared in Figure 8.38. It shows a good match between the MCU result and the Matlab result, which hence verifies the short-time RMS implementation on the MCU.

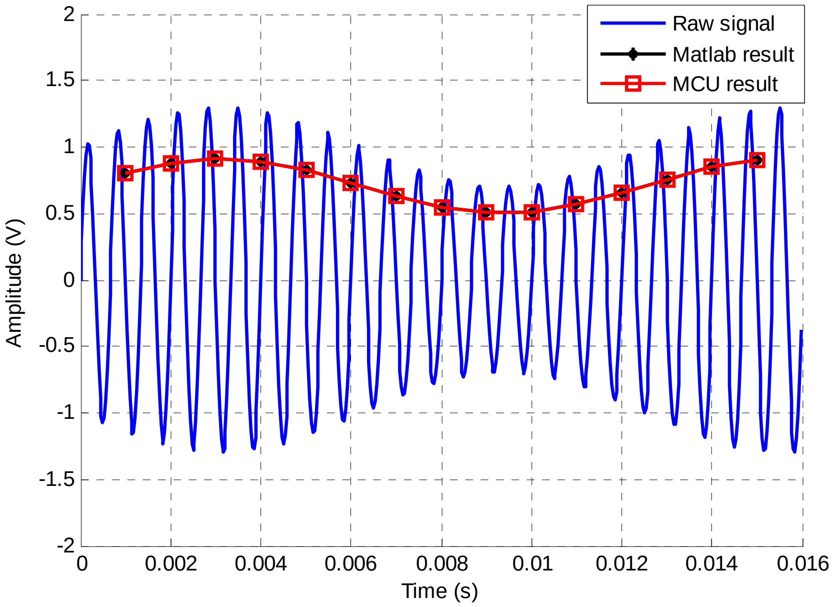


Figure 8.38 Comparison of short-time RMS processing results by Matlab and MCU

To check the real-time performance of short-time RMS method, the frame data ready event (DMA interrupt event) and time for frame data processing (frame processing event) are measured with two GPIO outputs and shown in Figure 6.6, similar to the method in Section 6.2.2. It can be observed that the real-time requirement can be satisfied more than sufficient and the CPU usage time for one frame is about 2.2 ms, which is reduced by approximately three times than that in Section 7.3, which ranges from 8 ms to 10 ms. This allows the CPU to run at only 11.11 MHz, in which case, the processing time for one frame is about 15.84 ms and the real-time performance can still be satisfied.

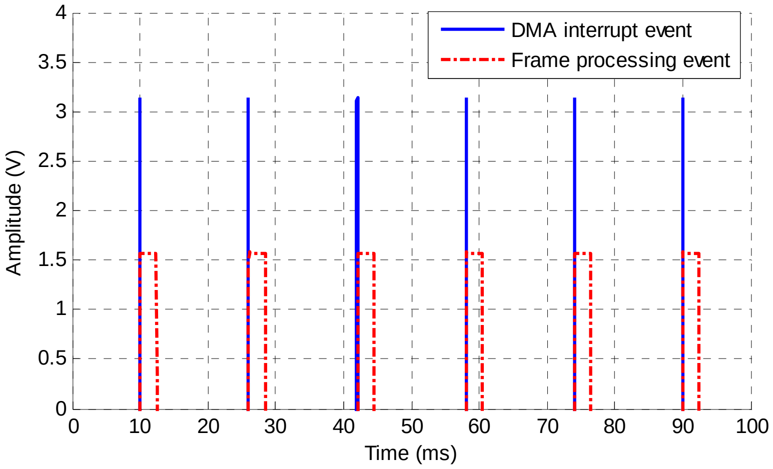


Figure 8.39 DMA interrupt and frame processing event measured through GPIO for short-time RMS

8.5.3 Implementation of cascaded spectral correlation

As discussed in Section 8.4.2, the spectral correlation can achieve similar diagnosis results as the other two frequency domain methods but can be much faster for processing vibration signal from bearing. Thereby, it is employed for cascaded processing to obtain an envelope spectrum with high resolution.

To achieve cascaded processing, the envelope signal is obtained from the square envelope spectrum. As shown in Figure 8.40, the envelope signal is acquired by performing an IFFT operation on the squared spectrum X_{envs} and a following square root operation. Note that the IFFT computation has much less points than that in FFT calculation, which indicates the appended IFFT and square root operation consume very little time.

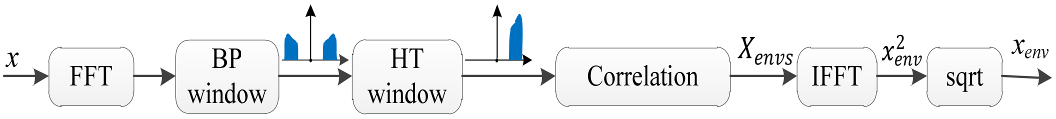


Figure 8.40 Schematic of spectral correlation based envelope detection

Similar as the cascaded frequency domain HT in Figure 7.8, the cascaded processing for spectral correlation processed frame with a size of 1024 points and overlap ratio of 50%, as shown in Figure 7.8. For each frame, a 1024-point forward FFT is firstly performed and then the frequency components in the pass-band are extracted and amplitude doubled to obtain the analytic spectrum. Then, a correlation is performed to obtain the squared envelope spectrum. Note that, with sampling rate at 32 kHz, there only need to be 16 points of squared envelope spectrum data to cover the 500 Hz envelope frequency range.

Then, a 32-point reverse RFFT operation is performed to get the squared envelope and only the middle 16 points of data are kept as a valid squared envelope for scale and square root operation to obtain the envelope signal.

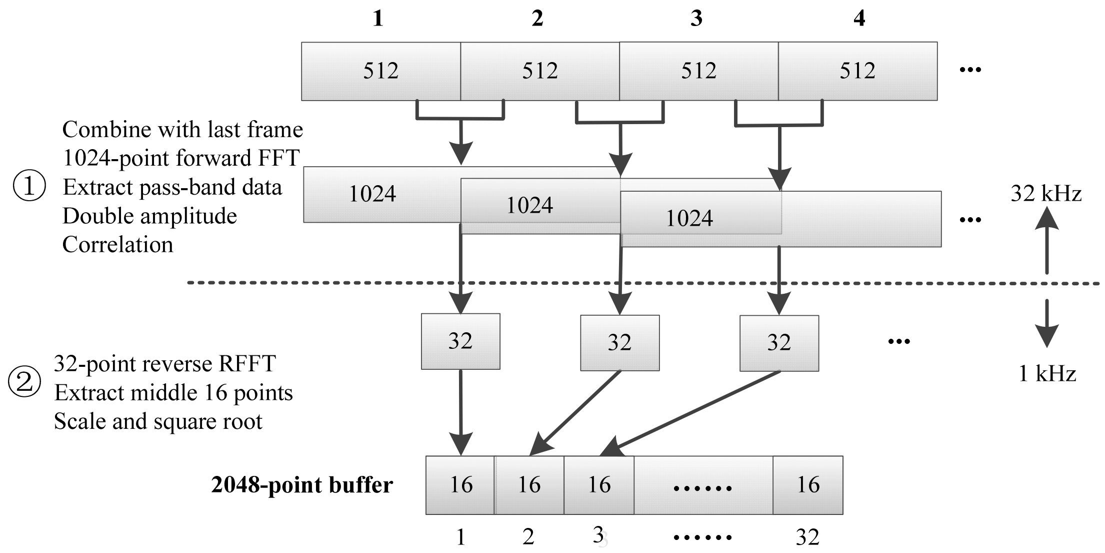


Figure 8.41 Diagram for cascaded spectral correlation

8.5.4 Verification of spectral correlation implementation

To verify the implementation of spectral correlation and the consecutiveness of the cascaded envelope, a modulating signal is processed on the MCU and the detected envelope from first three frames are extracted and illustrated in Figure 8.42. The cascading procedure from successive overlapped frames is visualised in the figure. It can be observed that the obtained envelope has a good match with the upper outline of the modulating signal in Figure 8.42(d). Furthermore, small distortions can be seen on the edge of the obtained envelope in Figure 8.42(b) and (c). These parts are not employed for cascaded processing.

Similar to the verification with short-time RMS in the last section, the frame data ready event (DMA interrupt event) and time for frame data processing (frame processing event) are measured and presented in Figure 8.43. It shows the CPU usage time for one frame is about 3.12 ms, which is just slightly more than that in short-time RMS and two times reduction in comparison with that in Section 7.3. This allows the CPU running speed to only 16 MHz, in which case the processing time for one frame is approximately 15.6 ms. Note that it is easier to update the band-pass window in spectral correlation than to update the band-pass filter coefficients in short-time RMS.

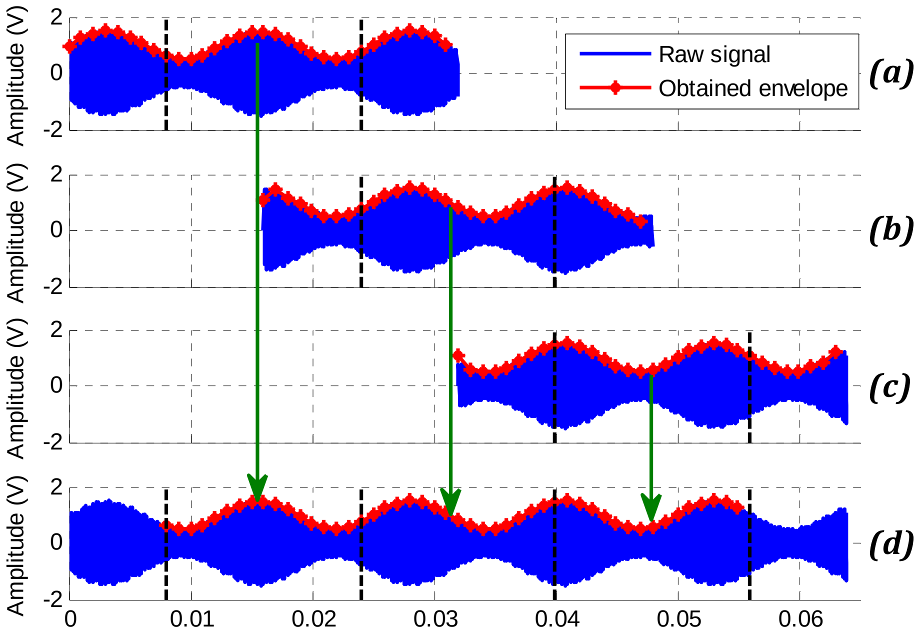


Figure 8.42 Illustration of cascaded spectral correlation for envelope calculation:
(a) first frame result, (b) second frame result, (c) third frame result and (d) cascaded envelope

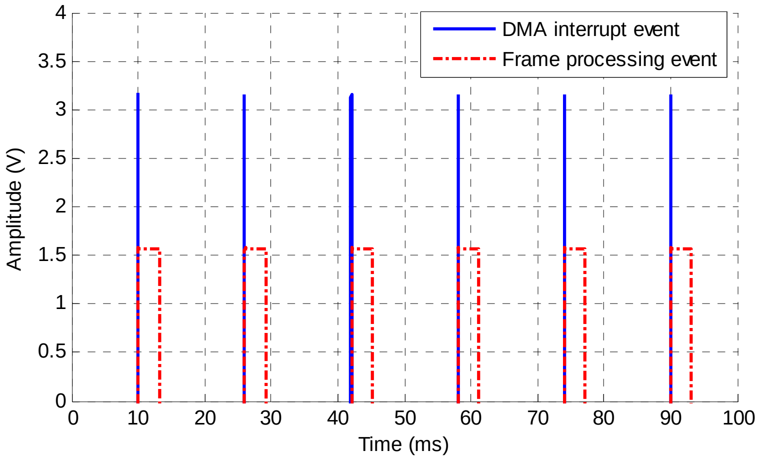


Figure 8.43 DMA interrupt and frame processing event measured through GPIO
for spectral correlation

8.5.5 Results and discussion

To demonstrate the effectiveness of the implemented cascaded short-time RMS and spectral correlation, the same three types of vibration signals described in Chapter 7 are used for processing. The processing results from frequency domain HT, short-time RMS, and spectral correlation is then compared in Figure 8.44, Figure 8.45 and Figure 8.46. Note that the same band-pass filter is employed for three algorithms for three types of signals, that's 8 kHz to 10.667 kHz for outer race fault signal, 2.667 kHz to 5.333 kHz for inner race fault signal and 4 kHz to 5 kHz for roller fault signal.

It can be observed that the three faults can be clearly identified in the envelope spectrum of all three algorithms. In comparison, it is not easy to observe the higher harmonics in the envelope spectrum of short-time RMS in Figure 8.44(c), Figure 8.45(c) and Figure 8.46(c). This can be explained in the high attenuation effect of the overlapped average process in short-time RMS. For frequency domain HT and spectral correlation, they can reveal the same count of harmonics for all three faults and the amplitudes of the corresponding harmonics are very similar. However, more noise can be observed in the low-frequency range of spectral correlation than those in frequency domain HT or short-time RMS.

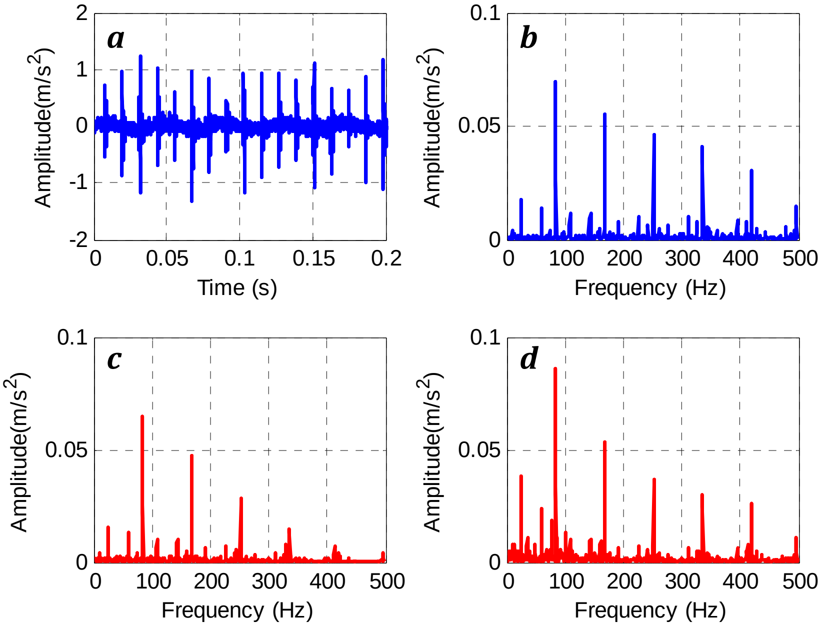


Figure 8.44 Detection results for a bearing vibration signal with outer race fault: (a) raw signal, (b) envelope spectrum from frequency domain HT, (c) envelope spectrum from short-time RMS and (d) envelope spectrum from spectral correlation

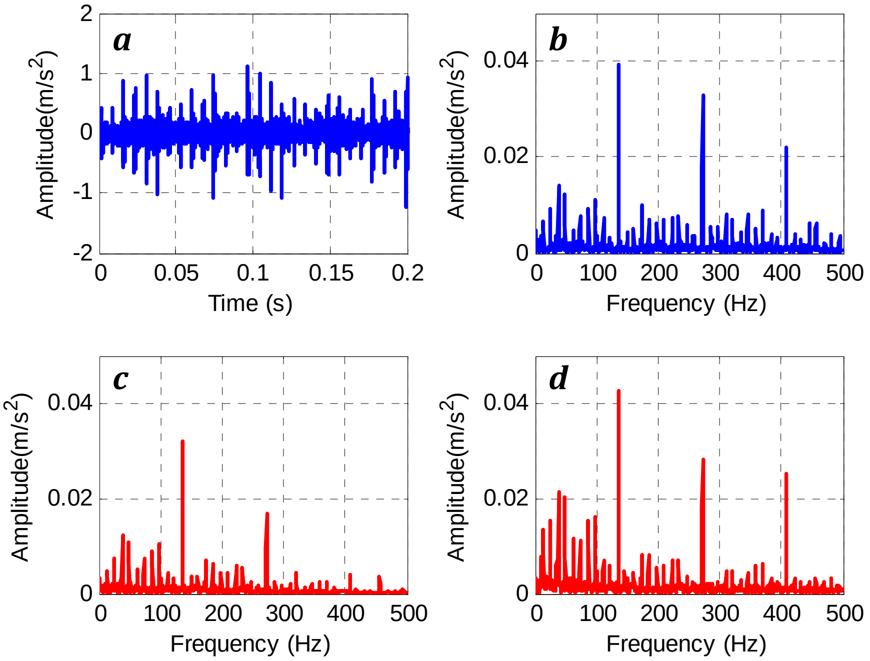


Figure 8.45 Detection results for a bearing vibration signal with inner race fault: (a) raw signal, (b) envelope spectrum from frequency domain HT, (c) envelope spectrum from short-time RMS and (d) envelope spectrum from spectral correlation

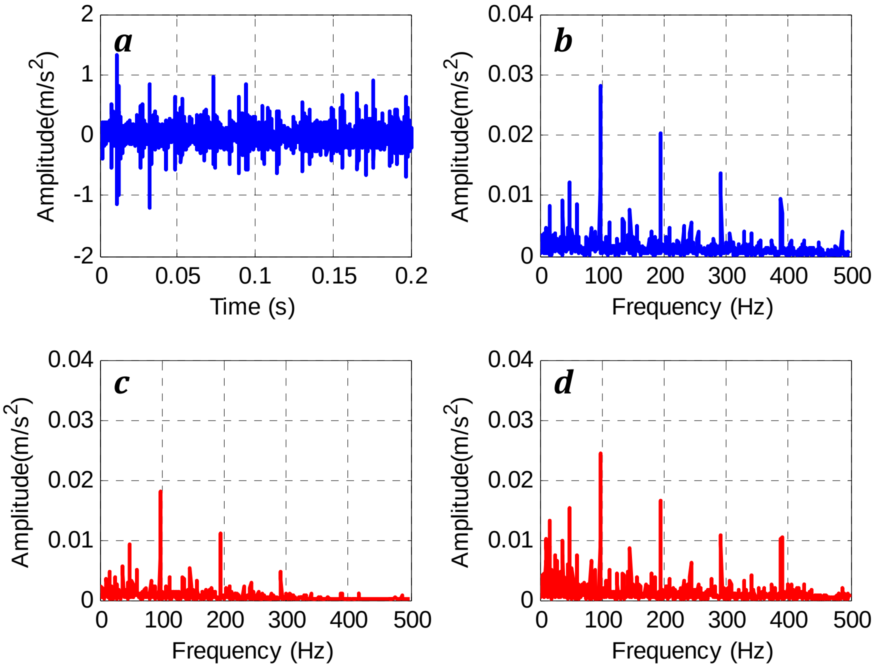


Figure 8.46 Detection results for a bearing vibration signal with roller fault: (a) raw signal, (b) envelope spectrum from frequency domain HT, (c) envelope spectrum from short-time RMS and (d) envelope spectrum from spectral correlation

8.6 Summary

In this chapter, several envelope detection methods are explored, including the simple rectifier, short-time statistical characteristics, and the Hilbert transform implementation both in the time domain and frequency domain. From the analysis and discussion, it is found faster implementation can be achieved both in time domain and frequency domain. This allows the CPU usage to be decreased by about three times for time domain method and two times for frequency domain method in comparison with the implementation in Chapter 7. Furthermore, the analysis results on the vibration signals also show that the proposed methods have good accuracy for bearing fault diagnosis.

CHAPTER 9

CONCLUSIONS AND FUTURE RESEARCH PLAN

In this chapter, the research objectives are reviewed and the corresponding achievements are summarised in detail. On this basis, the conclusions for the entire thesis are drawn and a summary of the contributions of this research is listed. Finally, several suggestions are given for future work.

9.1 Review of research objectives and achievements

This research aims to develop a wireless vibration CM node which allows an optimal implementation of advanced signal processing algorithms. The main achievements of this work are presented below in the same order as they appear in Section 1.5.

Objective one: Review popular wireless techniques and compare their advantages and disadvantages for being applied for industrial CM applications. Investigate research works performed and current industrial solutions for wireless condition monitoring. On this basis, select one wireless protocol to prototype a wireless CM system for evaluation.

Achievement one: The popular wireless protocols are investigated and compared in Section 2.2. It shows WirelessHART and ISA100.11a can be employed for industrial applications requiring low time latency and high data delivery reliability, whereas CM applications normally do not need such high performance. The protocols such as ZigBee, BLE, WiFi HaLow and Thread are suitable candidates for wireless CM.

The research work and products for industrial condition monitoring are investigated in Section 2.2, which shows the distributed computation by embedding intelligent data processing on wireless sensor nodes are getting popular in the wireless CM area. This can not only solve the bandwidth limitations but also brings the benefit of lowering the power consumption of sensor nodes.

In addition, the popular wireless sensor platforms are investigated which shows the Cortex-M series processor are becoming popular among the WSN solutions. Based on the investigation, an integrated wireless module specially for handling wireless network task is selected for establishing the wireless network.

Objective two: Prototype a wireless CM system with a focus on the design of the vibration based wireless sensor node for implementing and evaluating signal processing algorithms that are commonly used in condition monitoring field. Consideration of its power consumption should be made in order to maintain a service span as long as possible when it is powered by batteries.

Achievement two: A wireless CM system structure is proposed in Section 3.1, based on the Zigbee wireless technique. It composes a number of vibration sensor nodes for data collection and embedded processing to extract fault feature and one sink node for organising the wireless network and relaying data packets between sensor nodes and the host computer.

In Section 3.2, the main baseboard is introduced. A state-of-the-art Cortex-M4F processor is selected as the core processor for the wireless sensor node instead of the traditional DSP processors. The reason is that the fast development in Cortex-M based processor and their integration of DSP and FPU unit has made an additional DSP processor redundant and hence reduced the system cost. A LaunchPad board with Cortex-M4F processor is employed as the main baseboard due to its powerful performance and easily extensible structure.

In Section 3.3, the vibration conditioning circuit is designed, including a charge amplifier, low-pass filter and voltage amplifier. For low power considerations, low power operational amplifiers are employed for building the circuit and load switches are added to power off some circuits when they are not in use. In addition, an external ADC with 16-bit resolution and a MicroSD card are also integrated for future upgrading considerations.

In Section 3.4, an integrated wireless module is employed for establishing the wireless network. The XBee module is selected due to its well-recognised effectiveness, compatibility, and reliability. The configurations for the XBee modules are detailed in Section 3.4.3. For the power management on the wireless sensor node, a commercially available battery booster pack is employed in Section 3.5. The information like temperature, charging state, capacity can be accessed by the main baseboard through a simulated I2C communication port.

Objective three: Study the fault generation mechanism and fault features of rolling element bearings and then investigate popular signal processing algorithms for their fault detection and diagnosis. Discuss the requirements for an algorithm to be implemented on processors with restricted computational capability and limited memory size. On this basis, select one signal processing algorithm for being embedded on the sensor node for bearing fault feature extraction.

Achievement three: Through the study in Section 4.1, it is found that the existence of a defect on a bearing’s subcomponent (e.g. outer race, inner race, roller or cage) results in the bearing to periodically vibrate at its resonance frequencies as the bearing rotates at a steady speed. The frequency of this periodical vibrating response is component specific and can be used as evidence for fault existence on corresponding components. This is a typical impulsive and modulating vibration signature, which can also be found in other common rotating components, such as gears, valves, and turbines.

Various signal processing algorithms are then explored for analysing vibration signals from rolling element bearings, including time domain, frequency domain and time-frequency domain methods in Section 0, 4.3, 4.4, respectively. The time domain methods like statistical parameters and probability density function are computation efficient for detecting abnormal conditions in bearing but they cannot be relied on for diagnosing the specific fault component. TSA is helpful for de-noising but requires an additional synchronisation signal. The morphological filter provides an efficient way for envelope detection and thus can be employed as part of envelope analysis.

The spectral analysis can visualise the vibrating frequency distribution in the spectrum but it is quite challenging to figure out the localised fault types. The envelope analysis and cepstrum extract the component fault information from the modulated vibration signal and can directly tell the localised fault type. In comparison, the result from cepstrum is noisier and more difficult to interpret than that from envelope analysis.

The time-frequency domain analysis methods provide informative details about the vibration signal in both time domain and frequency domain. The methods like STFT, CWT and HHT are too computation complicated or memory consuming for being implemented on an embedded processor. In the meantime, DWT provides sufficient resolution for analysis and has good computation performance for being a good candidate for de-noising or envelope detection on an embedded processor.

With further consideration of the efforts in implementing these algorithms, envelope analysis is deemed as the most appreciative one to be embedded on wireless sensor nodes for feature extraction. The advantages of envelope analysis are summarised below:

- Capable of detecting the abnormality and localised fault components.
- Potential to produce a small resultant feature set.
- Suppressing interference noises from nearby components.

- ➢ Acceptable computation complexity and capability to well utilise the library functions optimised on the embedded processors.

Objective four: Implement the selected envelope analysis algorithm on the proposed wireless sensor nodes for extracting the fault features of rolling bearings. Optimise and validate each implementation step including data acquisition, data processing, and data transmission. Utilise the implemented algorithm for bearing fault feature extraction and evaluate its performance in terms of computation speed, memory consumption and data reduction in comparison with the raw data transmission.

Achievement four: In Chapter 5, by implementing Hilbert transform with an efficient complex FFT algorithm, an envelope analysis of 2048 points is successfully implemented on the proposed wireless sensor node.

A double buffer structure combining ADC, timer, and DMA is established in Section 5.1 for efficient data acquisition and thus reducing the interferences to the computation-intensive signal processing. The implementation is achieved on top of a real-time operating system and validated by collecting an amplitude modulated signal and confirming its frequency components.

The three steps for envelope analysis, including band-pass filter, envelope detection, and spectrum calculation are implemented and validated in Section 5.2. An 81-tap FIR type filter with a passband between 1 kHz and 2 kHz is designed and validated by filtering a signal with several frequency components. Its preciseness is also confirmed with a comparison with the processing results in Matlab. The frequency domain Hilbert transform is utilised for envelope detection and its implementation is optimised by combining a real-valued forward FFT and a complex inverse FFT calculation. Its implementation is validated by locally processing a simulated modulating signal and comparing the results with that in Matlab. The spectrum is calculated with a real-valued forward FFT and a partial complex magnitude computation. Similarly, its implementation is validated by comparing the results from embedded processing and Matlab. Furthermore, the spectrum average is suggested for reducing noise and enhancing sensitivity.

The implementation of data transmission is explained in Section 5.3. Through analysis, it shows only 103 data points in the calculated envelope spectrum are useful for transmission and further analysis. The performance of the overall implementation is evaluated in Section 5.4 by processing a true vibration signal from bearing with an outer race fault. Experimental results show that outer race fault frequency can be clearly observed in the envelope spectrum, validating the existence of outer race fault, also correctness and effectiveness of the implemented envelope analysis on the embedded processor. The total time consumption for the whole processing is about 28.4 ms with the processor running at 80 MHz, which is less than the data collection time (204.8 ms) for a frame with 2048 points of data. The memory consumption is less than the maximum 32 kB RAM on the MCU through a large portion of memory being reused during calculations. A significant data reduction of more than 95% in comparison with the raw data set is achieved by just sending the useful results in the envelope spectrum.

Objective five: Discuss the necessity and possibility for improving the spectrum resolution in the envelope spectrum. Note that this improvement should be accomplished on the same processor with the idea of frame data processing and down-sampling. Explain the implementation procedure and validate the correctness of the implementation. Compare the performance of the improved method on bearing fault feature extraction with that in the initial implementation.

Achievement five: From the analysis in Section 6.1.1, it shows the spectrum resolution achieved in Chapter 5 is only about 4.88 Hz, which can very likely cause incorrect interpretation on fault frequencies and their harmonics. For accurate fault diagnosis, there is a need to increase the resolution of the envelope spectrum.

A data processing scheme that combines down-sampling, overlap processing and cascading is proposed in Section 6.1.2, through which a spectrum resolution of 0.61 Hz can be achieved, which is eight times of that in Chapter 5. An overlap of 50% between successive frames is employed to avoid edge distortions and keep the consecutiveness of the computed envelope signal.

In Section 6.2, the proposed data processing scheme is implemented by processing 32 successive data frames with 512 points. Furthermore, the double buffer mechanism is utilised for efficient data acquisition. The processing time for one frame data is measured as 10.875 ms, much less than the deadline requirement of 51.2 ms, i.e. the time for collecting one frame of data. Then, the implemented method is validated by processing an amplitude modulated signal, showing correct demodulation of the envelope and no jumping points in the joint of successive data frames.

In Section 6.3, the improved method is employed for processing the same bearing fault signal in Chapter 5 and a much finer frequency resolution is observed in the envelope spectrum of the improved method, hence allowing a more accurate and reliable diagnosis result to be achieved. Furthermore, it shows the proposed method not only improves the frequency resolution by eight times on the same processor but also has no increment on data throughput requirement.

Objective six: Investigate the methods for selecting an optimal band-pass filter for envelope analysis. On this basis, utilise one algorithm in the prototype wireless CM system to achieve automatic band-pass filter selection and parameters updating. Explain and optimise the implementation procedure. Evaluate the implemented scheme on different bearing fault signals to confirm its effectiveness.

Achievement six: In Section 7.1, it is found that the traditional methods, like searching for peaks in the spectrum or hammer tests for finding system resonances, are not suitable for automated band-pass filter. The spectral kurtosis and the improved fast kurtogram are good candidates for optimal band-pass filter selection. Thus, their theories are studied in Section 7.2. On this basis, the fast kurtogram is employed for optimum band-pass filter selection in the proposed wireless sensor network due to its relatively faster computation speed than spectral kurtosis and acceptable preciseness.

In Section 7.3, the implementation of fast kurtogram is explained. The sensor node is enabled to work in two modes: configuration mode and monitoring mode. In the configuration mode, a data set with 8192 points is collected and transmitted to the host computer for selecting an optimal band-pass filter through fast kurtogram. Then, the band-pass filter parameters are generated and sent back to the sensor node for use in the envelope analysis. In the monitoring mode, the sensor node collects vibration data, computes the envelope spectrum and just transmits the useful results back to the host computer. Note that the sampling rate is increased to 32 kHz to allow features being extracted in a wide frequency band with a frequency resolution similar to that in Chapter 6. Furthermore, a frequency domain band-pass filter is employed instead of the FIR filter in Chapter 6 to accelerate the calculation that requires real-time performance and also reduces the parameter size for updating in configuration mode.

In Section 7.4, the implemented scheme is evaluated by processing three types of bearing fault signals, including outer race fault, inner race fault, and roller fault. The experimental results show that the resonance frequency band can be correctly selected by the fast kurtogram method and the bearing fault types can be clearly identified from the calculated envelope spectrum. In addition, it shows in Section 7.4.4 that the data throughput requirement is very similar to that in Chapter 6 for the monitoring mode. As the configuration mode can be performed in data backup stage, which is unusual, it is not expected to cause much load in the wireless network.

Objective seven: Investigate and compare common envelope detection methods, including the simple squared rectifier and the precise Hilbert transform based method. On this basis, find the possibility to optimise and speed up the implementation process of envelope detection. Implement, validate and evaluate the proposed methods for processing bearing fault signals.

Achievement seven: The simple envelope detectors are investigated in Section 8.1, including the half-wave, full-wave and squared rectifier, peak detector and short-time statistical features. The half-wave and full-wave inevitably bring new frequency component distortions due to infinite extended harmonics while squared rectifier has much fewer such distortions but also requires a higher sampling rate to avoid aliasing. The peak detector does not bring in new frequency components while it is quite sensitive to noise and also requires a higher sampling frequency to obtain an accurate envelope. Among the various short-time statistical features, the short-time RMS is the best for envelope detection. In addition, the short-time RMS can be regarded as a simplified squared rectifier with the low-pass filter and downsample operation being implemented by an overlapped average. Moreover, it also shows that the short-time RMS is sufficiently sensitive to bearing defects.

The Hilbert transform based envelope detection is fully explored in Section 8.2. The time domain Hilbert transformer can solve the aliasing problems existing in the frequency domain method, which can be more efficiently computed by the FFT convolution, especially, for the case when a long transformer is required.

In Section 8.3.1, it is shown that the multiplication of analytic signal with its conjugate in the time domain is equivalent to its spectral correlation in the frequency domain. This indicates the squared envelope spectrum of a signal can be directly computed by the spectral correlation of its analytic signal, without the need for the inverse FFT and the second forward FFT in the frequency domain HT. The correctness of this finding is verified by calculating the envelope of a modulated signal using the spectral correlation method. Similarly, it is shown that the square operation of a real-valued signal in the time domain is equivalent to its cyclic spectral correlation in the frequency domain. With proper zero padding in the spectrum, the square operation can obtain the same envelope results as that in the Hilbert transform based method. Furthermore, a squared rectifier with the band-pass filter of much loose restriction can obtain a precise envelope for bearing fault feature extraction.

On this basis, the time domain methods, including band-pass squared rectifier, time domain HT and short-time RMS are compared in Section 8.4.1, which shows very similar performance for detecting an outer race fault while the short-time RMS method has the fastest computation speed. In Section 8.4.2, the frequency domain methods, including frequency domain HT, FFT convolution HT, and spectral correlation, are compared, which also shows a very similar detection result for detecting an outer race fault. In terms of computation speed, the FFT convolution HT consumes 2% more time than that the frequency domain HT while the spectral correlation has the potential to be faster than the other two methods.

In Section 8.4.1, the short-time RMS and spectral correlation methods are implemented for achieving higher resolution in the envelope spectrum by adopting the scheme proposed in Chapter 6. In terms of computation time, the short-time RMS consumes about 2.2 ms and spectral correlation consumes about 3.12 ms, being about three times and two times less than that achieved in Section 7.3.4 (8 – 10 ms), respectively. Comparative studies show that the short-time RMS and spectral correlation can achieve the same accuracy as the Hilbert transform in detecting and quantifying different bearing faults.

9.2 Conclusions

Based on the investigations and evaluations described in the previous chapters, the key findings covered in this thesis are concluded as follows:

Conclusion 1: The investigation on popular wireless protocols shows that WirelessHART and ISA100.11a can be employed for industrial applications requiring low time latency and high data delivery reliability, whereas CM applications normally do not need such high performance. The protocols such as ZigBee, BLE, WiFi HaLow and Thread are suitable candidates for CM. The distributed computation by embedding intelligent data processing on wireless sensor nodes is getting popular in the wireless CM area, which can not only solve the bandwidth limitations but also bring the benefit of lowering its power consumption.

Conclusion 2: Among the various signal processing algorithms, it is found envelope analysis is the most suitable algorithm for being embedded on wireless sensor nodes for extracting fault features from impulsive and modulating signals, especially the vibration signals from rolling element bearings. It has the following advantages:

- Capable of detecting the abnormality and localised fault components.
- Potential to produce a small resultant feature set.
- Suppressing interference noises from nearby components.
- Acceptable computation complexity and capability to well utilize the library functions optimised on the embedded processors.

Conclusion 3: An envelope analysis of 2048 points is successfully implemented on a state-of-the-art Cortex-M4F processor with only 32 kB memory. Experimental results show that the computed envelope spectrum feature can well identify the simulated bearing fault with a significant data reduction of more than 95% in comparison with the raw data set. The effective optimisation for the embedded computation includes:

- A double buffer structure combining ADC, timer and DMA for efficient data acquisition and thus reducing the interferences to the computation-intensive signal processing
- A combination of real-valued forward FFT and complex type inverse FFT for accelerating the computation of Hilbert transform
- The spectrum average for reducing noise, enhancing sensitivity and also decreasing the transmission of redundant information

Conclusion 4: A data processing scheme that combines down-sampling, overlap processing and cascading is proposed for getting a high resolution in the envelope spectrum. The proposed method is successfully implemented on the same wireless sensor node with a spectrum resolution of 0.61 Hz being achieved, which is eight times of that in the initial implementation. Experimental results show extracted feature from the improved method can provide more accurate and reliable diagnosis results without increment on data throughput requirement. One key implementation is the employment of a 50% overlap between successive frames, which has effectively avoided edge distortions and kept the consecutiveness of the computed envelope signal.

Conclusion 5: An automatic updating scheme is proposed for selecting an optimal band-pass filter for the envelope analysis in the vibration based wireless CM. In the scheme, the complicated fast kurtogram is implemented on the host computer for optimal band-pass filter selection and the envelope analysis which requires real-time requirement is embedded on the distributed sensor nodes for extracting fault features. The optimal filter parameters of the envelope analysis are updated only at data backup stage or required by the user and thus it is not expected to cause much load in the wireless network. Experimental results show the implemented scheme can successfully select the resonance frequency band for the evaluated three types of fault signals. Two main efforts have been implemented for the optimisation:

- The sampling rate is increased to 32 kHz to allow features being extracted in a wide frequency band with a frequency resolution of 0.49 Hz.
- A frequency domain band-pass filter is employed instead of the FIR filter to accelerate the calculation that requires real-time performance and also reduces the parameter size for updating in configuration mode.

Conclusion 6: A fast and accurate time domain method by cascading short-time RMS features is proposed, implemented and validated to be effective for bearing fault feature extraction on the proposed embedded processor. This method shows a three times reduction in CPU usage in comparison with that for the frequency domain HT. The idea of employing short-time RMS is based on the following three findings:

- The short-time RMS can be considered as a simplified squared rectifier with the low-pass filter and downsample operation being implemented by an overlapped average.
- A square operation of a real-valued signal in the time domain is equivalent to its spectral correlation in the frequency domain.
- A band-pass squared rectifier can obtain a precise envelope for bearing fault feature extraction. There is a very loose restriction on the band-pass filter

parameters as aliasing in the high-frequency range of the calculated envelope spectrum is allowed for bearing fault feature extraction.

Conclusion 7: It is found and validated that the multiplication of analytic signal with its conjugate in the time domain is equivalent to its spectral correlation in the frequency domain. This indicates the squared envelope spectrum of a signal can be directly computed by the spectral correlation of its analytic signal, without the need for the inverse FFT and the second forward FFT in the frequency domain HT. On this basis, it is implemented with the scheme of down-sampling, data frame overlapping and cascading to achieve a high-frequency resolution in the envelope spectrum. Experimental results show its effectiveness and accurate for bearing fault diagnosis with a two times reduction in CPU usage in comparison with that for the frequency domain HT.

9.3 Contributions to knowledge

A number of practical works have been performed in this thesis with a purpose to optimise the signal processing embedded on vibration based wireless CM nodes for bearing fault diagnosis. The contributions of this research are summarised as follows:

First contribution: It is the first time to propose envelope analysis as a rolling element bearing fault feature extraction algorithm for distributed computation on wireless sensor nodes. Through investigation on various signal processing algorithms, envelope analysis is shown as the best algorithm for being embedded on wireless sensor nodes to extract fault features from impulsive and modulating signals, especially the vibration signals from rolling element bearings. Its advantages for distributed computation on embedded processor include:

- ➢ Capable of detecting the abnormality and localised fault components.
- ➢ Potential to produce a small resultant feature set.
- ➢ Suppressing interference noises from nearby components.
- ➢ Acceptable computation complexity and capability to well utilize the library functions optimised on the embedded processors.

Second contribution: An efficient envelope analysis of 2048 points is the first time implemented and optimised on a state-of-the-art Cortex-M4F processor with only 32 kB memory. Experimental results show that the computed envelope spectrum feature can well identify the simulated bearing fault with a significant data reduction of more than 95% in comparison with the raw data set. This shows the great benefits of employing envelope analysis on the distributed wireless sensor nodes for condition monitoring.

Third contribution: A data processing scheme that combines down-sampling, overlap processing and cascading is firstly proposed for getting a high resolution in the envelope spectrum on a processor with limited computation memory (32 kB). A spectrum resolution of 0.61 Hz is achieved, which is eight times of that in the initial implementation. Note that this scheme is implemented on the same wireless sensor node without extra requirement on computation memory. This scheme makes envelope analysis more attractive for accurate bearing fault diagnosis on resource constraint wireless sensor nodes.

Fourth contribution: An automatic band-pass filter updating scheme is firstly proposed for selecting an optimal band-pass filter for the envelope analysis in the vibration based wireless CM. In the scheme, the complicated fast kurtogram is implemented on the host computer for optimal band-pass filter selection and the envelope analysis which requires real-time requirement is embedded on the distributed sensor nodes for extracting fault features. The optimal filter parameters of the envelope analysis are updated only at data backup stage or required by the user and thus it is not expected to cause much load in the wireless network. This shows a more meaningful and practical way of applying distributed computation in the wireless CM.

Fifth contribution: A fast and accurate time domain method based on short-time RMS is firstly embedded on distributed wireless sensor nodes for bearing fault feature extraction. Experimental results show short-time RMS provides sufficient accuracy to various bearing defects with three times reduction in CPU usage in comparison with that for the frequency domain HT.

Sixth contribution: The spectral correlation is firstly proposed for speeding up the calculation of envelope analysis on an embedded processor. Through the usage of the proposed scheme in Chapter 6, an accurate envelope spectrum is computed with sufficient accuracy for detecting various bearing faults. This algorithm shows a two times reduction in CPU usage in comparison with that for the frequency domain HT.

9.4 Future work

Recommendation 1: Evaluate and optimise the power management scheme of the sensor node. Although many efforts have been performed in this thesis to optimise and speed up the implemented algorithms which provide the potential to reduce power consumption, the resources on the wireless sensor node should be utilised wisely to achieve real minimal power consumption.

Recommendation 2: Explore other signal processing algorithms for embedded feature extraction, such as compressive sensing, DWT, and morphological filter. Such algorithms can provide more options for extracting different fault features.

Recommendation 3: Achieve reliable firmware upgrading over the air. In this way, the signal processing algorithms can be conveniently adjusted for satisfying different application requirements.

Recommendation 4: Fabricate the wireless sensor node into a more compact design and build up suitable package so that to can be applied to a more hash industrial environment.

Recommendation 5: Add multiple wireless sensor nodes in the established wireless network to evaluate the performance of the distributed wireless CM system.

Recommendation 6: Develop a graphical user interface that is more friendly and convenient for visualisation and control using more efficient development tools like Visual C++ or Python.

REFERENCES

[1] Clarence W. de Silva, *Vibration Monitoring, Testing, and Instrumentation*. Taylor & Francis Group, LLC, pp. 6.1-6.34, 2007.

[2] G. O. Allgood, W. W. Manges, and S. F. Smith, ‘It’s time for sensors to go wireless. Part 2: Take a good technology and make it an economic success’, *Sens. Peterb. NH*, vol. 16, no. 5, 1999.

[3] P. J. Tavner, ‘Review of condition monitoring of rotating electrical machines’, *IET Electr. Power Appl.*, vol. 2, no. 4, pp. 215–247, 2008.

[4] A. Kelly, *Maintenance Strategy*. Elsevier, pp. 79-124, 1997.

[5] W. Zhou, T. G. Habetler, and R. G. Harley, ‘Bearing Condition Monitoring Methods for Electric Machines: A General Review’, in *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2007. SDEMPED 2007*, 2007, pp. 3–6.

[6] E. P. Carden, P. Fanning, E. P. Carden, and P. Fanning, ‘Vibration based condition monitoring: A review’, *J. Struct. Health Monit.*, vol. 3, pp. 355–377, 2004.

[7] C. R. Farrar and S. W. Doebling, ‘damage detection ii: field applications to large structures’, *Modal Anlysis Test.*, 1999.

[8] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & systems (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., pp. 160, 1996.

[9] A. K. S. Jardine, D. Lin, and D. Banjevic, ‘A review on machinery diagnostics and prognostics implementing condition-based maintenance’, *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006.

[10] M. Elhaj, F. Gu, A. D. Ball, A. Albarbar, M. Al-Qattan, and A. Naid, ‘Numerical simulation and experimental study of a two-stage reciprocating compressor for condition monitoring’, *Mech. Syst. Signal Process.*, vol. 22, no. 2, pp. 374–389, Feb. 2008.

[11] F. Immovilli, C. Bianchini, M. Cocconcelli, A. Bellini, and R. Rubini, ‘Bearing Fault Model for Induction Motor With Externally Induced Vibration’, *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3408–3418, Aug. 2013.

[12] J. Urbanek, T. Barszcz, and T. Uhl, ‘Comparison of advanced signal-processing methods for roller bearing faults detection’, *Metrol. Meas. Syst.*, vol. 19, no. 4, pp. 715–726, 2012.

[13] S. Abdusslam, P. Raharjo, F. Gu, and A. Ball, ‘Bearing defect detection and diagnosis using a time encoded signal processing and pattern recognition method’, *J. Phys. Conf. Ser.*, vol. 364, no. 1, p. 12036, May 2012.

[14] R. B. Randall and J. Antoni, ‘Rolling element bearing diagnostics—A tutorial’, *Mech. Syst. Signal Process.*, vol. 25, no. 2, pp. 485–520, Feb. 2011.

[15] S. A. McInerny and Y. Dai, ‘Basic vibration signal processing for bearing fault detection’, *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 149–156, 2003.

[16] Y.-C. Choi and Y.-H. Kim, ‘Fault detection in a ball bearing system using minimum variance cepstrum’, *Meas. Sci. Technol.*, vol. 18, no. 5, p. 1433, May 2007.

[17] C. T. Yiakopoulos and I. A. Antoniadis, ‘Wavelet based demodulation of vibration signals

generated by defects in rolling element bearings', *Shock Vib.*, vol. 9, no. 6, pp. 293–306, 2002.

[18] U.S. Department of Energy, 'Industrial wireless technology for the 21st century', 2002.

[19] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, 'Wireless sensor networks for habitat monitoring', in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, New York, NY, USA, 2002, pp. 88–97.

[20] J. P. Lynch, 'An overview of wireless structural health monitoring for civil structures', *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 365, no. 1851, pp. 345–372, Feb. 2007.

[21] A. Carullo, S. Corbellini, M. Parvis, and A. Vallan, 'A Wireless Sensor Network for Cold-Chain Monitoring', *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1405–1411, May 2009.

[22] V. C. Gungor and G. P. Hancke, 'Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches', *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[23] L. Hou and N. W. Bergmann, 'System requirements for industrial wireless sensor networks', in *2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010, pp. 1–8.

[24] B. Lu and V. C. Gungor, 'Online and Remote Motor Energy Monitoring and Fault Diagnostics Using Wireless Sensor Networks', *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4651–4659, Nov. 2009.

[25] L. Hou and N. W. Bergmann, 'Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis', *IEEE Trans. Instrum. Meas.*, vol. 61, no. 10, pp. 2787–2798, Oct. 2012.

[26] ADI, 'Analog Devices' MEMS Wireless Vibration Sensing System Enables Remote Monitoring of Industrial Machine Health | Press Releases | Analog Devices', 04-Jun-2013. [Online]. Available: http://www.analog.com/en/press-release/06_04_13_ADI_MEMS_Wireless_Vibration_Sensing/press.html. [Accessed: 16-Aug-2013].

[27] ABB, 'Condition monitoring- Wireless vibration monitoring system'. 2010.

[28] Emerson Process management, 'Emerson Process Management - CSI 9420 Wireless Vibration Transmitter'. [Online]. Available: http://www2.emersonprocess.com/en-us/brands/csitechnologies/vt/csi9420/pages/csi9420wirelessvibrationtransmitter.aspx. [Accessed: 11-Aug-2013].

[29] GE measurement & Control, 'Essential Insight.mesh', 2013. [Online]. Available: http://www.ge-mcs.com/en/bently-nevada-monitoring/wireless-surveillance-scanning/essential-insight.html. [Accessed: 09-Aug-2013].

[30] A. Prijic, L. Vracar, D. Vuckovic, D. Milic, and Z. Prijic, 'Thermal Energy Harvesting Wireless Sensor Node in Aluminum Core PCB Technology', *IEEE Sens. J.*, vol. 15, no. 1, pp. 337–345, Jan. 2015.

[31] S. Dalola, M. Ferrari, V. Ferrari, M. Guizzetti, D. Marioli, and A. Taroni, 'Characterization of Thermoelectric Modules for Powering Autonomous Sensors', *IEEE Trans. Instrum. Meas.*, vol. 58, no. 1, pp. 99–107, Jan. 2009.

[32] N. Harid, A. C. Bogias, and A. Haddad, 'A solar-powered wireless data acquisition system for monitoring electrical substations', in *Proceedings of the 11th WSEAS international conference on Electronics, Hardware, Wireless and Optical Communications, and proceedings of the 11th WSEAS international conference on Signal Processing, Robotics and Automation, and proceedings of the 4th WSEAS international conference on Nanotechnology*, Stevens Point, Wisconsin, USA, 2012, pp. 181–186.

[33] Y. K. Tan and S. K. Panda, 'Energy Harvesting From Hybrid Indoor Ambient Light and Thermal Energy Sources for Enhanced Performance of Wireless Sensor Nodes', *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4424–4435, Sep. 2011.

[34] J. H. Jang, D. F. Berdy, J. Lee, D. Peroulis, and B. Jung, ‘A wireless sensor node for condition monitoring powered by a vibration energy harvester’, in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.

[35] J. H. Jang, D. F. Berdy, J. Lee, D. Peroulis, and B. Jung, ‘A Wireless Condition Monitoring System Powered by a Sub-100 W Vibration Energy Harvester’, *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 60, no. 4, pp. 1082–1093, 2013.

[36] P. Radmand, A. Talevski, S. Petersen, and S. Carlsen, ‘Comparison of industrial WSN standards’, in *2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2010, pp. 632–637.

[37] T. Sreenuch, A. Tsourdos, and I. K. Jennions, ‘Distributed embedded condition monitoring systems based on OSA-CBM standard’, *Comput. Stand. Interfaces*, vol. 35, no. 2, pp. 238–246, 2013.

[38] K. K. Tan, S. N. Huang, Y. Zhang, and T. H. Lee, ‘Distributed fault detection in industrial system based on sensor wireless network’, *Comput. Stand. Interfaces*, vol. 31, no. 3, pp. 573–578, Mar. 2009.

[39] Y. Kaicheng and Z. Chaosheng, ‘Vibration data fusion algorithm of auxiliaries in power plants based on wireless sensor networks’, in *2011 International Conference on Computer Science and Service System (CSSS)*, 2011, pp. 935–938.

[40] T. Li and M. Fei, ‘Fault diagnosis of auxiliaries in power plants based on wireless sensor networks with vibration transducer’, in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, 2010, pp. 732–736.

[41] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, ‘IMOTE2: Serious Computation at the Edge’, in *Wireless Communications and Mobile Computing Conference, 2008. IWCMC ’08. International*, 2008, pp. 1118–1123.

[42] Y. Bao, H. Li, X. Sun, Y. Yu, and J. Ou, ‘Compressive sampling based data loss recovery for wireless sensor networks used in civil’, *Struct. Health Monit.*, p. 1475921712462936, Nov. 2012.

[43] X. Zhang, N. Hu, L. Hu, L. Chen, and Z. Cheng, ‘A bearing fault diagnosis method based on the low-dimensional compressed vibration signal’, *Adv. Mech. Eng.*, vol. 7, no. 7, p. 1687814015593442, Jul. 2015.

[44] Y. Bao, Z. Zou, and H. Li, ‘Compressive sensing based wireless sensor for structural health monitoring’, 2014, p. 90611W.

[45] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza, ‘A real-time compressed sensing-based personal electrocardiogram monitoring system’, in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, 2011, pp. 1–6.

[46] E. J. Candes and M. B. Wakin, ‘An Introduction To Compressive Sampling’, *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[47] O. M. Bouzid, ‘In-situ health monitoring for wind turbine blade using acoustic wireless sensor networks at low sampling rates’, 2013.

[48] O. M. Bouzid, G. Y. Tian, J. Neasham, and B. Sharif, ‘Envelope and Wavelet Transform for Sound Localisation at Low Sampling Rates in Wireless Sensor Networks’, *J. Sens.*, vol. 2012, p. e680383, Mar. 2012.

[49] D. Ho and R. B. Randall, ‘Optimisation Of Bearing Diagnostic Techniques Using Simulated And Actual Bearing Fault Signals’, *Mech. Syst. Signal Process.*, vol. 14, no. 5, pp. 763–788, Sep. 2000.

[50] H. Konstantin-Hansen, ‘Envelope analysis for diagnostics of local faults in rolling element bearings’, *Brüel Kja Er Den.*, 2003.

[51] H. Li, X. Zhang, and F. Xu, ‘Experimental Investigation on Centrifugal Compressor Blade Crack Classification Using the Squared Envelope Spectrum’, *Sensors*, vol. 13, no. 9, pp. 12548–12563, Sep. 2013.

[52] M. Ahmed, F. Gu, and A. D. Ball, ‘Fault Detection of Reciprocating Compressors using a Model from Principles Component Analysis of Vibrations’, *J. Phys. Conf. Ser.*, vol. 364, no. 1, p. 12133, May 2012.

[53] K. Jiang, G. Xu, L. Liang, T. Tao, and F. Gu, ‘The Recovery of Weak Impulsive Signals Based on Stochastic Resonance and Moving Least Squares Fitting’, *Sensors*, vol. 14, no. 8, pp. 13692–13707, Jul. 2014.

[54] K. R. Rao, Z. S. Bojkovic, and D. A. Milovanovic, *Wireless Multimedia Communications: Convergence, DSP, QoS, and Security*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2008.

[55] J. Cheng, J. Jin, and L. Kong, ‘Wireless distributed monitoring and centralized controlling system for prefabricated substations in China’, in *IEEE International Conference on Industrial Technology, 2005. ICIT 2005*, 2005, pp. 45–50.

[56] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ‘A survey on sensor networks’, *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[57] L. Mainetti, L. Patrono, and A. Vilei, ‘Evolution of wireless sensor networks towards the Internet of Things: A survey’, in *2011 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2011, pp. 1–6.

[58] V. Beal, ‘What is Wi-Fi (IEEE 802.11x)? Webopedia Definition’. [Online]. Available: http://www.webopedia.com/TERM/W/Wi_Fi.html. [Accessed: 24-Jul-2016].

[59] H. Wu, L. Liu, and X. Yuan, ‘Remote Monitoring System of Mine Vehicle Based on Wireless Sensor Network’, in *2010 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2010, vol. 2, pp. 1015–1019.

[60] Wi-Fi Alliance, ‘Wi-Fi HaLow’. [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wi-fi-halow. [Accessed: 23-May-2016].

[61] M. Hamblen, ‘Wi-Fi for the Internet of Things gets a name: “Wi-Fi HaLow”’, *Computerworld*, 04-Jan-2016. [Online]. Available: http://www.computerworld.com/article/3018510/mobile-wireless/wi-fi-for-the-internet-of-things-gets-a-name-wi-fi-halow.html. [Accessed: 23-May-2016].

[62] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel, ‘The Intel reg; mote platform: a Bluetooth-based sensor network for industrial monitoring’, in *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005*, 2005, pp. 437–442.

[63] N. Aakvaag, M. Mathiesen, and G. Thonet, ‘Timing and power issues in wireless sensor networks - an industrial test case’, in *International Conference Workshops on Parallel Processing, 2005. ICPP 2005 Workshops*, 2005, pp. 419–426.

[64] C. Gomez, J. Oller, and J. Paradells, ‘Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology’, *Sensors*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012.

[65] S. Sirur, P. Juturu, H. P. Gupta, P. R. Serikar, Y. K. Reddy, S. Barak, and B. Kim, ‘A mesh network for mobile devices using Bluetooth low energy’, in *2015 IEEE SENSORS*, 2015, pp. 1–4.

[66] J.-S. Lee, Y.-W. Su, and C.-C. Shen, ‘A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi’, in *33rd Annual Conference of the IEEE Industrial Electronics Society, 2007. IECON 2007*, 2007, pp. 46–51.

[67] K. Khakpour and M. H. Shenassa, ‘Industrial Control using Wireless Sensor Networks’, in *3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008*, 2008, pp. 1–5.

[68] M. F. Cabanas, F. Pedrayes, C. H. Rojas, M. G. Melero, J. G. Norniella, G. . Orcajo, J. M. Cano, F. Nuno, and D. R. Fuentes, ‘A New Portable, Self-Powered, and Wireless Instrument for the Early Detection of Broken Rotor Bars in Induction Motors’, *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4917–4930, Oct. 2011.

[69] T. Lennvall, S. Svensson, and F. Hekland, ‘A comparison of WirelessHART and ZigBee for industrial applications’, in *IEEE International Workshop on Factory Communication Systems, 2008. WFCS 2008*, 2008, pp. 85–88.

[70] HART communication foundation, ‘HART Communication Protocol -- News & Events’, 2010. [Online]. Available: http://hartcomm.org/hcf/news/pr2010/WirelessHART_approved_by_IEC.html. [Accessed: 27-Mar-2013].

[71] N. Q. Dinh, S.-W. Kim, and D.-S. Kim, ‘Performance evaluation of priority CSMA-CA mechanism on ISA100.11a wireless network’, in *2010 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, 2010, pp. 991–996.

[72] H. Hayashi, T. Hasegawa, and K. Demachi, ‘Wireless technology for process automation’, in *ICCAS-SICE, 2009*, 2009, pp. 4591–4594.

[73] K. Al Agha, M.-H. Bertin, T. Dang, A. Guitton, P. Minet, T. Val, and J.-B. Viollet, ‘Which Wireless Technology for Industrial Wireless Sensor Networks? The Development of OCARI Technology’, *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4266–4278, Oct. 2009.

[74] Montenegro, N. Kushalnagar, J. Hui, and D. Culler, ‘Transmission of IPv6 Packets over IEEE 802.15.4 Networks’, vol. RFC4944, no. 4944, pp. 1–30, 2007.

[75] Thread Group, ‘What is Thread?’ [Online]. Available: https://threadgroup.org/what-Is-thread. [Accessed: 24-May-2016].

[76] P. Zand, A. Dilo, and P. Havinga, ‘D-MSR: A Distributed Network Management Scheme for Real-Time Monitoring and Process Control Applications in Wireless Industrial Automation’, *Sensors*, vol. 13, no. 7, pp. 8239–8284, Jun. 2013.

[77] F. Salvadori, M. De Campos, R. de Figueiredo, C. Gehrke, C. Rech, P. S. Sausen, M. A. Spohn, and A. Oliveira, ‘Monitoring and Diagnosis in Industrial Systems Using Wireless Sensor Networks’, in *IEEE International Symposium on Intelligent Signal Processing, 2007. WISP 2007*, 2007, pp. 1–6.

[78] F. Salvadori, M. De Campos, P. S. Sausen, R. F. de Camargo, C. Gehrke, C. Rech, M. A. Spohn, and A. C. Oliveira, ‘Monitoring in Industrial Systems Using Wireless Sensor Network With Dynamic Power Management’, *IEEE Trans. Instrum. Meas.*, vol. 58, no. 9, pp. 3104–3111, Sep. 2009.

[79] X. Xue, V. Sundararajan, and W. P. Brithinee, ‘The application of wireless sensor networks for condition monitoring in three-phase induction motors’, in *Electrical Insulation Conference and Electrical Manufacturing Expo, 2007*, 2007, pp. 445–448.

[80] B. Lu, L. Wu, T. G. Habetler, R. G. Harley, and J. A. Gutierrez, ‘On the application of wireless sensor networks in condition monitoring and energy usage evaluation for electric machines’, in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005*, 2005, pp. 2674–2679.

[81] B. Lu, T. G. Habetler, R. G. Harley, and J. A. Gutierrez, ‘Applying wireless sensor networks in industrial plant energy management systems. Part I. A closed-loop scheme’, in *2005 IEEE Sensors*, 2005, p. 6 pp.-pp.

[82] B. Lu, T. G. Habetler, R. G. Harley, and J. A. Gutierrez, ‘Applying wireless sensor networks in industrial plant energy management systems. Part II. Design of sensor devices’, in *2005 IEEE Sensors*, 2005, p. 6 pp.-pp.

[83] J. A. Gutierrez, D. B. Durocher, B. Lu, and T. G. Habetler, ‘Applying Wireless Sensor Networks in Industrial Plant Energy Evaluation and Planning Systems’, in *Pulp and Paper Industry Technical Conference, 2006. Conference Record of Annual*, 2006, pp. 1–7.

[84] P. S. Sausen, J. R. Sousa, M. A. Spohn, A. Perkusich, and A. M. N. Lima, ‘Dynamic Power Management with Scheduled Switching Modes in Wireless Sensor Networks’, in *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS ’07*, 2007, pp. 1–8.

[85] J. P. Amaro, F. J. T. E. Ferreira, R. Cortesao, N. Vinagre, and R. P. Bras, ‘Low cost wireless sensor network for in-field operation monitoring of induction motors’, in *2010 IEEE International Conference on Industrial Technology (ICIT)*, 2010, pp. 1044–1049.

[86] Y. Hong-fang, W. Hua-qing, Q. He, and K. Xi-yong, ‘A ZigBee wireless sensor network for fault diagnosis of pump’, in *2010 Sixth International Conference on Natural Computation (ICNC)*, 2010, vol. 7, pp. 3778–3782.

[87] GST, ‘Wireless Sensors & Measurement Systems - GST’, 2012. [Online]. Available: http://www.globalsensortech.com/wireless-sensors-measurement-systems. [Accessed: 21-Aug-2013].

[88] MicroStrain, ‘IEPE-Link^{TM} -LXRS® Wireless IEPE Sensor Node’, 2014. [Online]. Available: http://www.microstrain.com/wireless/iepe-link. [Accessed: 28-Sep-2014].

[89] Siemens AG, ‘WirelessHART - Automation Technology - Siemens’, 2013. [Online]. Available: http://www.automation.siemens.com/w1/automation-technology-wirelesshart-18957.htm#. [Accessed: 11-Aug-2013].

[90] Honeywell Inc., ‘XYR 6000 Wireless Transmitters - Accurate, Cost-Effective Process Monitoring’, 2013. [Online]. Available: https://www.honeywellprocess.com/en-US/explore/products/wireless/input-output-devices/xyr-6000/Pages/default.aspx. [Accessed: 11-Aug-2013].

[91] PCB piezoelectronics, ‘Echo® Wireless Vibration Monitoring System’. [Online]. Available: https://www.imi-sensors.com/Echo_Wireless.aspx. [Accessed: 11-Aug-2013].

[92] Texas Instruments, ‘Personal Area Networks - ZigBee RF4CE - CC2530 - TI.com’, 2013. [Online]. Available: http://www.ti.com/product/cc2530. [Accessed: 29-May-2013].

[93] NXP, ‘JN5148 :: NXP Semiconductors’, 2013. [Online]. Available: http://www.nxp.com/products/microcontrollers/wireless_microcontrollers/JN5148.html. [Accessed: 21-Aug-2013].

[94] J. L. Hill and D. E. Culler, ‘Mica: a wireless platform for deeply embedded networks’, *IEEE Micro*, vol. 22, no. 6, pp. 12–24, 2002.

[95] D. Lymberopoulos and A. Savvides, ‘XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications’, in *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005*, 2005, pp. 449–454.

[96] A. El Kouche, A. Alma’aitah, H. Hassanein, and K. Obaia, ‘Monitoring operational mining equipment using Sprouts Wireless Sensor Network platform’, in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, 2013, pp. 1388–1393.

[97] Digi International Inc., ‘XBee® ZB - Digi International’, 28-Mar-2013. [Online]. Available:

http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module#overview. [Accessed: 28-Mar-2013].

[98] S. M. Kuo, B. H. Lee, and W. Tian, *Real-time digital signal processing: fundamentals, implementations and applications*, Third edition. Chichester, West Sussex: Wiley, 2013.

[99] T. B. Welch, C. H. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing from MATLAB® to C with the TMS320C6x DSPs*. CRC Press, 2011.

[100] W.-S. Gan and S. M. Kuo, *Embedded signal processing with the Micro Signal Architecture*. Hoboken, N.J: Wiley-Interscience : IEEE Press, 2007.

[101] R. Reinhard Keil, 'Digital Signal Processing with Cortex-M Microcontrollers'. 2011.

[102] BDTI, 'ARM Cortex-M7: Digital Signal Processing Drives Family Evolution'. [Online]. Available: http://www.bdti.com/InsideDSP/2014/09/25/ARM. [Accessed: 29-May-2016].

[103] Texas Instruments, 'ARM® Cortex®-M4F Based MCU TM4C123G LaunchPad^{TM} Evaluation Kit - EK-TM4C123GXL'. [Online]. Available: http://www.ti.com/tool/ek-tm4c123gxl. [Accessed: 10-Jul-2016].

[104] Texas Instruments, 'Tiva^{TM} C Series TM4C123GH6PM Microcontroller Data Sheet'. Texas Instruments, 20-Jun-2014.

[105] Texas Instruments, 'TI LaunchPad - BoosterPacks', 2014. [Online]. Available: http://www.ti.com/ww/en/launchpad/boosterpacks.html. [Accessed: 16-Jul-2014].

[106] PCB piezoelectronics, 'Introduction to Piezoelectric Accelerometers', 01-Jun-2013. [Online]. Available: http://www.pcb.com/TechSupport/Tech_Accel#.UanZSpO1GrI. [Accessed: 01-Jun-2013].

[107] A. Albarbar, S. Mekid, A. Starr, and R. Pietruszkiewicz, 'Suitability of MEMS accelerometers for condition monitoring: An experimental study', *Sensors*, vol. 8, no. 2, pp. 784–799, 2008.

[108] E. Bartolome, 'Signal conditioning for piezoelectric sensors'. Texas Instruments Incorporated, 2010.

[109] M. Oljaca and H. Surtihadi, 'Operational amplifier gain stability, Part 1: General system analysis', *Analog Appl.*, 2010.

[110] Linear Technology, 'LTC6078 - Micropower Precision, Dual CMOS Rail-to-Rail Input/Output Amplifiers'. [Online]. Available: http://www.linear.com/product/LTC6078. [Accessed: 17-Jul-2014].

[111] Linear Technology, 'LTC2051 - Dual Zero-Drift Operational Amplifiers'. [Online]. Available: http://www.linear.com/product/LTC2051. [Accessed: 17-Jul-2014].

[112] C. Siegl, 'Power electronics - Load switches get smart', *Power Eng.*, vol. 20, no. 6, pp. 42–45, Dec. 2006.

[113] Texas Instruments, 'TPS22929D, Integrated Load Switch'. [Online]. Available: http://www.ti.com/product/tps22929d. [Accessed: 17-Jul-2014].

[114] Texas Instruments, 'REF5025 datasheet'. [Online]. Available: http://www.ti.com/product/ref5025. [Accessed: 17-Jul-2014].

[115] Linear Technology, 'Datasheet LTC1864'. [Online]. Available: http://www.linear.com/product/LTC1864. [Accessed: 17-Jul-2014].

[116] X. Liu, H. Chen, M. Wang, and S. Chen, 'An XBee-Pro based energy monitoring system', in *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*, 2012, pp. 1–6.

[117] Digi International Inc., 'XBee®/XBee-PRO® ZB RF Modules'. Digi International Inc., 08-May-2013.

[118] HobbyTronics, 'Xbee USB Adapter'. [Online]. Available:

http://www.hobbytronics.co.uk/xbee-usb-adapter. [Accessed: 21-Jul-2014].

[119] Element14, 'Fuel Tank BoosterPack'. [Online]. Available: http://www.element14.com/community/docs/DOC-55618/l/fuel-tank-boosterpack. [Accessed: 17-Jul-2014].

[120] I. Howard, 'A Review of Rolling Element Bearing Vibration'Detection, Diagnosis and Prognosis'', DTIC Document, 1994.

[121] B. Zhang, G. Georgoulas, M. Orchard, A. Saxena, D. Brown, G. Vachtsevanos, and S. Liang, 'Rolling element bearing feature extraction and anomaly detection based on vibration monitoring', in *2008 16th Mediterranean Conference on Control and Automation*, 2008, pp. 1792–1797.

[122] O. V. Thorsen and M. Dalva, 'A survey of faults on induction motors in offshore oil industry, petrochemical industry, gas terminals, and oil refineries', *IEEE Trans. Ind. Appl.*, vol. 31, no. 5, pp. 1186–1196, Sep. 1995.

[123] N. Tandon and A. Choudhury, 'A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings', *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, Aug. 1999.

[124] N. G. Nikolaou and I. A. Antoniadis, 'Application of morphological operators as envelope extractors for impulsive-type periodic signals', *Mech. Syst. Signal Process.*, vol. 17, no. 6, pp. 1147–1162, Nov. 2003.

[125] Mitchell Lebold , Katherine McClintic, Robert Campbell, and Carl Byington, and Kenneth Maynard, 'Review of Vibration Analysis Methods for Gearbox Diagnostics and Prognostics', presented at the Proceedings of the 54th Meeting of the Society for Machinery Failure Prevention Technology, Virginia Beach, VA, 2000, pp. 623–634.

[126] R. B. W. Heng and M. J. M. Nor, 'Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition', *Appl. Acoust.*, vol. 53, no. 1–3, pp. 211–226, Jan. 1998.

[127] 01dB technologies, 'Vibration monitoring of Rolling element bearings'. .

[128] W. Wang and O. A. Jianu, 'A Smart Sensing Unit for Vibration Measurement and Monitoring', *IEEEASME Trans. Mechatron.*, vol. 15, no. 1, pp. 70–78, 2010.

[129] M. Behzad, A. R. Bastami, and D. Mba, 'Rolling bearing fault detection by short-time statistical features', *Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng.*, vol. 226, no. 3, pp. 229–237, Aug. 2012.

[130] B. Kilundu, P. Dehombreux, C. Letot, and X. Chiementin, 'Early detection of bearing damage by means of decision trees', *J. Autom. Mob. Robot. Intell. Syst.*, vol. 3, pp. 70–74, 2009.

[131] P. Maragos, 'Tutorial On Advances In Morphological Image Processing And Analysis', *Opt. Eng.*, vol. 26, no. 7, p. 267623-267623-, 1987.

[132] P. Maragos and R. W. Schafer, 'Morphological filters–Part I: Their set-theoretic analysis and relations to linear shift-invariant filters', *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 8, pp. 1153–1169, Aug. 1987.

[133] S. Lou, X. Jiang, and P. J. Scott, 'Morphological filters for functional assessment of roundness profiles', *Meas. Sci. Technol.*, vol. 25, no. 6, p. 65005, Jun. 2014.

[134] Z. Liu, J. Wang, and B. Liu, 'ECG Signal Denoising Based on Morphological Filtering', in *(iCBBE) 2011 5th International Conference on Bioinformatics and Biomedical Engineering*, 2011, pp. 1–4.

[135] J. Wang, G. Xu, Q. Zhang, and L. Liang, 'Application of improved morphological filter to the extraction of impulsive attenuation signals', *Mech. Syst. Signal Process.*, vol. 23, no. 1, pp. 236–245, Jan. 2009.

[136] K. Jiang, G. Xu, T. Tao, and L. Liang, 'Rolling bearing quality evaluation based on a morphological

filter and a Kolmogorov complexity measure', *Int. J. Precis. Eng. Manuf.*, vol. 16, no. 3, pp. 459–464, Mar. 2015.

[137] W. Sui and D. Zhang, 'Research on envelope analysis for bearings fault detection', in *2010 5th International Conference on Computer Science and Education (ICCSE)*, 2010, pp. 973–976.

[138] P. D. MCFADDEN and M. M. TOOZHY, 'Application of synchronous averaging to vibration monitoring of rolling element bearings', *Mech. Syst. Signal Process.*, vol. 14, no. 6, pp. 891–906, Nov. 2000.

[139] Y. Qu, D. He, J. Yoon, B. Van Hecke, E. Bechhoefer, and J. Zhu, 'Gearbox Tooth Cut Fault Diagnostics Using Acoustic Emission and Vibration Sensors — A Comparative Study', *Sensors*, vol. 14, no. 1, pp. 1372–1393, Jan. 2014.

[140] J.-B. Yu, 'Bearing performance degradation assessment using locality preserving projections', *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7440–7450, Jun. 2011.

[141] M. Angelo, 'Technical Review No. 1 1987 Vibration Monitoring of Machines'. Brüel & Kjær, 1987.

[142] H. Konstantin-Hansen and H. Herlufsen, 'Envelope and cepstrum analyses for machinery fault identification', *Sound Vib.*, vol. 44, no. 5, p. 10, 2010.

[143] Z. Liu, X. Yin, Z. Zhang, D. Chen, and W. Chen, 'Online rotor mixed fault diagnosis way based on spectrum analysis of instantaneous power in squirrel cage induction motors', *IEEE Trans. Energy Convers.*, vol. 19, no. 3, pp. 485–490, 2004.

[144] R. R. Schoen and T. G. Habetler, 'Effects of time-varying loads on rotor fault detection in induction machines', *IEEE Trans. Ind. Appl.*, vol. 31, no. 4, pp. 900–906, 1995.

[145] M. Feldman, *Hilbert transform applications in mechanical vibration*. Chichester: Wiley, pp. 9-32, 2011.

[146] R. R.B., 'Cepstrum Analysis and Gearbox Fault Diagnosis'. Bruel & Kjaer, 1973.

[147] A. M. Noll, 'Cepstrum Pitch Determination', *J. Acoust. Soc. Am.*, vol. 41, no. 2, pp. 293–309, Feb. 1967.

[148] R. B. Randall, 'A history of cepstrum analysis and its application to mechanical problems', in *International Conference*, 2013, pp. 29–30.

[149] B. Bogert, M. Healy, and J. Tukey, 'The quefrency alanysis of time series for echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking', presented at the Proc. Symp. on Time Series Analysis, 1963, pp. 209–243.

[150] C.-S. Park, Y.-C. Choi, and Y.-H. Kim, 'Early fault detection in automotive ball bearings using the minimum variance cepstrum', *Mech. Syst. Signal Process.*, vol. 38, no. 2, pp. 534–548, Jul. 2013.

[151] J. Antoni and R. B. Randall, 'The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines', *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 308–331, Feb. 2006.

[152] S. G. Mallat, *A wavelet tour of signal processing: the sparse way*, 3rd ed. Amsterdam ; Boston: Elsevier/Academic Press, 2009.

[153] M. Unser and A. Aldroubi, 'A review of wavelets in biomedical applications', *Proc. IEEE*, vol. 84, no. 4, pp. 626–638, Apr. 1996.

[154] N. G. Nikolaou and I. A. Antoniadis, 'Demodulation of vibration signals generated by defects in rolling element bearings using complex shifted morlet wavelets', *Mech. Syst. Signal Process.*, vol. 16, no. 4, pp. 677–694, Jul. 2002.

[155] W. J. Wang and P. D. McFadden, 'Application of wavelets to gearbox vibration signals for fault

detection', *J. Sound Vib.*, vol. 192, no. 5, pp. 927–939, May 1996.

[156] W. Q. Wang, F. Ismail, and M. Farid Golnarachi, 'Assessment of gear damage monitoring techniques using vibration measurements', *Mech. Syst. Signal Process.*, vol. 15, no. 5, pp. 905–922, Sep. 2001.

[157] X. Li and X. Yao, 'Multi-scale statistical process monitoring in machining', *IEEE Trans. Ind. Electron.*, vol. 52, no. 3, pp. 924–927, Jun. 2005.

[158] I. Cade, P. S. Keogh, and M. N. Sahinkaya, 'Fault identification in rotor/magnetic bearing systems using discrete time wavelet coefficients', *IEEEASME Trans. Mechatron.*, vol. 10, no. 6, pp. 648–657, Dec. 2005.

[159] L. Eren and M. J. Devaney, 'Bearing damage detection via wavelet packet decomposition of the stator current', *IEEE Trans. Instrum. Meas.*, vol. 53, no. 2, pp. 431–436, Apr. 2004.

[160] Q. Sun and Y. Tang, 'Singularity analysis using continuous wavelet transform for bearing fault diagnosis', *Mech. Syst. Signal Process.*, vol. 16, no. 6, pp. 1025–1041, Nov. 2002.

[161] J. Lin and L. Qu, 'Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis', *J. Sound Vib.*, vol. 234, no. 1, pp. 135–148, Jun. 2000.

[162] H. Qiu, J. Lee, J. Lin, and G. Yu, 'Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics', *J. Sound Vib.*, vol. 289, no. 4–5, pp. 1066–1090, Feb. 2006.

[163] K. Chen, X. Li, F. Wang, T. Wang, and C. Wu, 'Bearing fault diagnosis using Wavelet analysis', in *2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE)*, 2012, pp. 699–702.

[164] W. He, Z.-N. Jiang, and K. Feng, 'Bearing fault detection based on optimal wavelet filter and sparse code shrinkage', *Measurement*, vol. 42, no. 7, pp. 1092–1102, Aug. 2009.

[165] Y.-T. Sheen and C.-K. Hung, 'Constructing a wavelet-based envelope function for vibration signal analysis', *Mech. Syst. Signal Process.*, vol. 18, no. 1, pp. 119–126, Jan. 2004.

[166] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, 'The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis', *Proc. R. Soc. Lond. Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[167] J. Yan and L. Lu, 'Improved Hilbert–Huang transform based weak signal detection methodology and its application on incipient fault diagnosis and ECG signal analysis', *Signal Process.*, vol. 98, pp. 74–87, May 2014.

[168] B. M. Battista, C. Knapp, T. McGee, and V. Goebel, 'Application of the empirical mode decomposition and Hilbert-Huang transform to seismic reflection data', *GEOPHYSICS*, vol. 72, no. 2, pp. H29–H37, Mar. 2007.

[169] Z. K. Peng, P. W. Tse, and F. L. Chu, 'A comparison study of improved Hilbert–Huang transform and wavelet transform: Application to fault diagnosis for rolling bearing', *Mech. Syst. Signal Process.*, vol. 19, no. 5, pp. 974–988, Sep. 2005.

[170] V. K. Rai and A. R. Mohanty, 'Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform', *Mech. Syst. Signal Process.*, vol. 21, no. 6, pp. 2607–2615, Aug. 2007.

[171] R. Yan and R. X. Gao, 'Huang Transform-Based Vibration Signal Analysis for Machine Health Monitoring', *IEEE Trans. Instrum. Meas.*, vol. 55, no. 6, pp. 2320–2329, Dec. 2006.

[172] N. E. Huang, Z. Shen, and S. R. Long, 'A NEW VIEW OF NONLINEAR WATER WAVES: The Hilbert Spectrum ^{1}', *Annu. Rev. Fluid Mech.*, vol. 31, no. 1, pp. 417–457, Jan. 1999.

[173] H. Li, Y. Zhang, and H. Zheng, ‘Hilbert-Huang transform and marginal spectrum for detection and diagnosis of localized defects in roller bearings’, *J. Mech. Sci. Technol.*, vol. 23, no. 2, pp. 291–301, Apr. 2010.

[174] B. S. Kim, S. H. Lee, M. G. Lee, J. Ni, J. Y. Song, and C. W. Lee, ‘A comparative study on damage detection in speed-up and coast-down process of grinding spindle-typed rotor-bearing system’, *J. Mater. Process. Technol.*, vol. 187–188, pp. 30–36, Jun. 2007.

[175] C. Zinner and W. Kubinger, ‘ROS-DMA: A DMA Double Buffering Method for Embedded Image Processing with Resource Optimized Slicing’, in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium, 2006*, 2006, pp. 361–372.

[176] Texas Instruments, ‘TivaWare Peripheral Driver Library user’s guide’. .

[177] Texas Instruments, ‘TI-RTOS: A real-time operating system for TI devices’. 2015.

[178] R. G. Lyons, *Understanding digital signal processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, pp. 59-160, 2011.

[179] ARM Limited, ‘CMSIS DSP Software Library’. [Online]. Available: http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html.

[180] Texas Instruments, ‘Using the CMSIS DSP Library in Code Composer Studio for TM4C MCUs’. Oct-2015.

[181] X. X. Zheng, H. S. Xu, and others, ‘Fault diagnosis of wind turbine rolling bearing based on wavelet and Hilbert transforms’, in *Control Conference (CCC)*, 2012, pp. 24–26.

[182] S. Holm, ‘FFT pruning applied to time domain interpolation and peak localization’, *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 12, pp. 1776–1778, 1987.

[183] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3 edition. Upper Saddle River: Prentice Hall, 2009.

[184] D. H. Shreve, ‘Signal processing for effective vibration analysis’, *IRD Mechanalysis Inc Columb. OH USA*, 1995.

[185] Texas Instruments, ‘Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v5 - CCSTUDIO - TI Tool Folder’, 2013. [Online]. Available: http://www.ti.com/tool/ccstudio#Technical Documents. [Accessed: 12-Dec-2013].

[186] S. K. Basha and P. C. Pandey, ‘Real-time enhancement of electrolaryngeal speech by spectral subtraction’, in *2012 National Conference on Communications (NCC)*, 2012, pp. 1–5.

[187] J. Antoni, ‘Fast computation of the kurtogram for the detection of transient faults’, *Mech. Syst. Signal Process.*, vol. 21, no. 1, pp. 108–124, Jan. 2007.

[188] R. Dwyer, ‘Detection of non-Gaussian signals by frequency domain Kurtosis estimation’, in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’83.*, 1983, vol. 8, pp. 607–610.

[189] J. Antoni, ‘The spectral kurtosis: a useful tool for characterising non-stationary signals’, *Mech. Syst. Signal Process.*, vol. 20, no. 2, pp. 282–307, Feb. 2006.

[190] D. Benitez, P. A. Gaydecki, A. Zaidi, and A. P. Fitzpatrick, ‘The use of the Hilbert transform in ECG signal analysis’, *Comput. Biol. Med.*, vol. 31, no. 5, pp. 399–406, Sep. 2001.

[191] F. Wang, A. Yang, D. F. Kimball, L. E. Larson, and P. M. Asbeck, ‘Design of wide-bandwidth envelope-tracking power amplifiers for OFDM applications’, *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 4, pp. 1244–1255, Apr. 2005.

[192] A. C. McCormick and A. K. Nandi, ‘Cyclostationarity in rotating machine vibrations’, *Mech. Syst.*

Signal Process., vol. 12, no. 2, pp. 225–242, 1998.

[193] M. Feldman, 'Hilbert transform in vibration analysis', *Mech. Syst. Signal Process.*, vol. 25, no. 3, pp. 735–802, Apr. 2011.

[194] R. Lyons, 'Quadrature Signals: Complex, But Not Complicated'. 2008.

[195] M. Borgerding, 'Turning overlap-save into a multiband mixing, downsampling filter bank', *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 158–161, Mar. 2006.