Data Processing Algorithms in Wireless Sensor Networks for Structural Health Monitoring

DANNA NIGATU MITIKU
MEKONNEN ESAYAS GETACHEW

Master of Science Thesis
Stockholm, Sweden 2011
Data Processing Algorithms in Wireless Sensor Networks for Structural Health Monitoring

DANNA NIGATU MITIKU
MEKONNEN ESAYAS GETACHEW

December 2011
TRITA-BKN. Master Thesis 335, 2011
ISSN 1103-4297
ISRN KTH/BKN/EX-335-SE
Preface

This master thesis was carried out by students from Blekinge Institute of Technology (BTH) at the Swedish Institute of Computer Science (SICS), the division of Networked Embedded Systems (NES), and the Department of Civil and Architectural Engineering, the division of Structural Engineering and Bridges, at the Royal Institute of Technology (KTH), in Stockholm on the VINNOVA STRUCT project.

We would like to express our gratitude to Thiemo Voigt for giving us the opportunity to be part of the Network Embedded Systems (NES) group of SICS at Kista, Stockholm. His fast response and guidance each time with a positive feedback to get sensible results were so tremendous. We also would like to acknowledge our supervisor Luca Mottola for his valuable comments on every way and for guiding us to concentrate on our work. Our supervisor and examiner, Prof. Raid Karoumi is a well dedicated person giving us time, patience, understanding and helping even with administration issues. Once again we would like to thank our supervisor, Ignacio Gonzales Silva (PhD student) for his encouragement that made the interdisciplinary work more interesting by sharing his experiences.

Thanks to Claes Kullberg, the lab assistant at KTH who facilitated the instrumentation and tests. Many thanks to SICS staffs, specially Simon Duquennoy, Marcus Lundén, Nicklas Wirström and Joakim Eriksson, for their technical support.

The most special thanks to our families who gave unlimited support, care and love.

The last, but not the least, thanks to all BTH staffs and friends.

Stockholm, January 23, 2012
Nigatu Mitiku and Esayas Getachew
Abstract

The gradual deterioration and failure of old buildings, bridges and other civil engineering structures invoked the need for Structural Health Monitoring (SHM) systems to develop a means to monitor the health of structures. Dozens of sensing, processing and monitoring mechanisms have been implemented and widely deployed with wired sensors.

Wireless sensor networks (WSNs), on the other hand, are networks of large numbers of low cost wireless sensor nodes that communicate through a wireless media. The complexity nature and high cost demand of the highly used wired traditional SHM systems have posed the need for replacement with WSNs. However, the major fact that wireless sensor nodes have memory and power supply limitations has been an issue and many efficient options have been proposed to solve this problem and preserve the long life of the network. This is the reason why data processing algorithms in WSNs focus mainly on the accomplishment of efficient utilization of these scarce resources.

In this thesis, we design a low-power and memory efficient data processing algorithm using in-place radix-2 integer Fast Fourier Transform (FFT). This algorithm requires inputs with integer values; hence, increases the memory efficiency by more than 40% and highly saves processor power consumption over the traditional floating-point implementation. A standard-deviation-based peak picking algorithm is next applied to measure the natural frequency of the structure.

The algorithms together with Contiki, a lightweight open source operating system for networked embedded systems, are loaded on Z1 Zolertia sensor node. Analogue Device’s ADXL345 digital accelerometer on board is used to collect vibration data. The bridge model used to test the target algorithm is a simply supported beam in the lab.

Keywords: Bridge, FFT, Modal Analysis, Monitoring, Natural Frequency, Peak Picking, Structure, Wireless Sensor.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analogue to Digital Converter</td>
</tr>
<tr>
<td>ANPSD</td>
<td>Averaged Normalized Power Spectral Densities</td>
</tr>
<tr>
<td>AR</td>
<td>Auto Regressive</td>
</tr>
<tr>
<td>ASICS</td>
<td>Application Specific Integrated Circuits</td>
</tr>
<tr>
<td>BSL</td>
<td>BootStrap Loader</td>
</tr>
<tr>
<td>CFDD</td>
<td>Curve-fit Frequency Domain Decomposition</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DAQ</td>
<td>Data Acquisition System</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processors</td>
</tr>
<tr>
<td>EFDD</td>
<td>Enhanced Frequency Domain Decomposition</td>
</tr>
<tr>
<td>ERA</td>
<td>Eigensystem Realization Algorithm</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Domain Decomposition</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Arrays</td>
</tr>
<tr>
<td>FRF</td>
<td>Frequency Response Functions</td>
</tr>
<tr>
<td>HPBW</td>
<td>Half-Power BandWidth</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol Version 6</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial Scientific and Medical</td>
</tr>
<tr>
<td>ITD</td>
<td>Ibrahim Time Domain</td>
</tr>
<tr>
<td>KTH</td>
<td>Kungliga Tekniska Högskola</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bits</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro Electro-Mechanical Systems</td>
</tr>
<tr>
<td>MIPS</td>
<td>Million Instructions per Second</td>
</tr>
<tr>
<td>MSP</td>
<td>Mixed Signal Processor</td>
</tr>
<tr>
<td>ODS</td>
<td>Operation Deflection Shapes</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PP</td>
<td>Peak Peaking</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>RDT</td>
<td>Random Decrement Technique</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RISC</td>
<td>Reduced Instruction Set Computer</td>
</tr>
<tr>
<td>ROM</td>
<td>Read-Only Memory</td>
</tr>
<tr>
<td>SDOF</td>
<td>Single Degree-of-Freedom</td>
</tr>
<tr>
<td>SEK</td>
<td>Swedish Kronor</td>
</tr>
<tr>
<td>SHM</td>
<td>Structural Health Monitoring</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>SSI</td>
<td>Stochastic Subspace Identification</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transport and Communication Protocol</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Table of Contents

Preface iii

Abstract v

List of Abbreviations vii

List of Figures xiii

List of Tables xv

1 Introduction 1

1.1 Problem Statement and Research Contribution . . . . . . . . . . . . 2

1.2 Aim and Scope of the Study . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 Survey of Selected Related Works . . . . . . . . . . . . . . . . . . . 3

1.4 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2 Structural Health Monitoring 5

2.1 Trends in SHM Systems . . . . . . . . . . . . . . . . . . . . . . . . 5

2.2 Advanced Approaches in SHM . . . . . . . . . . . . . . . . . . . . 5

2.3 Signal Analysis in Structural Health Monitoring . . . . . . . . . . . 6

2.3.1 Sampling Theorem . . . . . . . . . . . . . . . . . . . . . . . . 6

2.3.2 Fourier Transform . . . . . . . . . . . . . . . . . . . . . . . . 7

2.3.3 Discrete Fourier Transform (DFT) . . . . . . . . . . . . . . . 7

2.3.4 Fast Fourier Transform (FFT) . . . . . . . . . . . . . . . . . 8

2.4 Categories of Structural Health Monitoring . . . . . . . . . . . . . 8

2.4.1 Feature Extraction . . . . . . . . . . . . . . . . . . . . . . . . 9

2.4.2 Modal Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.5 Input-Output Modal Analysis . . . . . . . . . . . . . . . . . . . . . 13

2.6 Output-Only Modal Identification . . . . . . . . . . . . . . . . . . . 13

2.6.1 Peak Picking Method (PP) . . . . . . . . . . . . . . . . . . . . 15

2.6.2 Frequency Domain Decomposition (FDD) . . . . . . . . . . . . 15

2.6.3 Enhanced Frequency Domain Decomposition . . . . . . . . . . 16

2.6.4 Curve Fitting Frequency Domain Decomposition . . . . . . . 16

ix
# TABLE OF CONTENTS

## 3 Wireless Sensor Networks for SHM

3.1 Applications of WSNs .................................................. 20
3.2 Wireless Sensor Nodes .................................................. 20
3.3 Hardware Components of a Wireless Sensor Node ..................... 21
  3.3.1 Microcontroller .................................................. 22
  3.3.2 Transceiver .......................................................... 22
  3.3.3 Memory ............................................................... 23
  3.3.4 Power source ....................................................... 23
  3.3.5 Sensors .............................................................. 23
  3.3.6 Energy harvester (Optional) ..................................... 25
3.4 Software Components of a Wireless Sensor Node ...................... 26
  3.4.1 Operating systems .................................................. 26
  3.4.2 Application programs ............................................. 27
3.5 Challenges in Using WSNs .............................................. 27
  3.5.1 Memory constraints .............................................. 28
  3.5.2 Power consumption constraints ................................. 28

## 4 Design and Methodology .............................................. 29

4.1 Hardware Used ............................................................ 29
  4.1.1 Wireless system .................................................... 29
  4.1.2 Wired system ....................................................... 33
4.2 Software Used ........................................................... 33
  4.2.1 Realtime software tools .......................................... 33
  4.2.2 Offline software tools ........................................... 38

## 5 Accelerometers .......................................................... 39

5.1 Introduction .............................................................. 39
5.2 What Accelerometers Are? ............................................. 39
5.3 Selection Criteria ....................................................... 39
  5.3.1 Analogue or digital ................................................ 40
  5.3.2 Sensitivity ............................................................ 40
  5.3.3 Full-Scale range .................................................... 40
  5.3.4 Number of axes measured ....................................... 40
  5.3.5 Bandwidth response ................................................ 40
  5.3.6 Noise level .......................................................... 40
  5.3.7 Power consumption ................................................ 41
  5.3.8 Extra features ....................................................... 41
5.4 Which Ones Did We Use? ............................................... 41
  5.4.1 Analogue devices ADXL345 ....................................... 41
  5.4.2 Si-Flex™ accelerometer .......................................... 43
  5.4.3 Other accelerometers ............................................. 43
5.5 Validating the Accelerometer ......................................... 43
5.6 Cares to Be Taken ....................................................... 44
  5.6.1 Calibration .......................................................... 44
# List of Figures

1.1  The catastrophic failure of I-35W and Point Pleasant Bridges  ............. 1

2.1  Frequency response function. ........................................ 10
2.2  Single degree of freedom model. .................................... 11
2.3  Half power bandwidth method. ....................................... 12
2.4  An Impact hammer ..................................................... 13

3.1  Example application of SHM using WSN. From [23]. ....................... 19
3.2  The Z1 Zolertia wireless sensor unit .................................. 21
3.3  Hardware architecture of a wireless sensor node. ....................... 21
3.4  Internal components of a microcontroller. ............................ 22
3.5  The 3-axis ADXL MEM accelerometer mounted on a circuit board. .... 24
3.6  An example sensor board with accelerometer sensor on board. ......... 24
3.7  Solar panel and rechargeable battery for Jindo Bridge SHM [41] ....... 25
3.8  Position of the operating system ...................................... 26

4.1  Functional block diagram of Z1 WSN module. .......................... 30
4.2  The service architecture of Contiki. (From [2]) ........................ 34
4.3  A frequency spectrum plot divided into non-overlapping windows. .... 37

6.1  Part of the 3.5 m long test beam ..................................... 47
6.2  A simple beam model of experiment 1 showing the parts 1 to 4 for each attachment (all measurements are in cm). ...................... 48
6.3  The wooden support at the ends of the experimental beam .......... 48
6.4  The front end of the instrumentation on the experimental beam ...... 49
6.5  Attachment of the sensor units on the experimental beam .......... 50
6.6  Exciting the experimental beam with an impulse hammer .......... 50

7.1  Time and frequency domain data from iPhone ............................ 54
7.2  Time and frequency domain data from the wireless sensor .......... 54
7.3  Comparison between Floating point and Integer point FFT ........... 55
7.4  Time domain data from wired and wireless sensors .................. 56
7.5  Measurement from the wired sensor ................................... 57
7.6  Node 1 experiment 1 .................................................... 58
List of Figures

7.7  Node 2 experiment 1 ................................................................. 58
7.8  Node 3 experiment 1 ................................................................. 59
7.9  Node 4 experiment 1 ................................................................. 59
7.10 Experiment 2 wireless sensor data ............................................. 60
7.11 Experiment 3 wireless sensor data ............................................. 60
7.12 Experiment 3 wired sensor data .................................................. 60
List of Tables

4.1 Summary of parametric features of MSP430F2617. . . . . . . . . . . . . 31
4.2 Summary of CC2420 features. . . . . . . . . . . . . . . . . . . . . . . . . 31
4.3 Approximate current consumption of Z1 components. . . . . . . . . . . 32
5.1 Summary of the ADXL345 accelerometer features. . . . . . . . . . . . . 41
5.2 Summary of the Si-Flex™ accelerometer features. . . . . . . . . . . . . 43
5.3 Summary of the most common accelerometers in SHM. . . . . . . . . . 44
7.1 Comparison of floating point and fixed point FFT on the node . . . . . 55
Chapter 1

Introduction

The process of implementing a damage characterization and detection method for engineering structures is referred to as Structural Health Monitoring (SHM). Although it had been quite a while since the science of SHM was introduced, its use was confined to mechanical structures like airplanes, ships, machineries, etc. It had never been applied to civil engineering structures until its significance was noticed in the frequent deterioration and collapse of large and prestigious structures.

For example, the catastrophic failure of the I-35W Bridge in Minneapolis, Minnesota (Figure 1.1 left) and the Point Pleasant Bridge (Figure 1.1 right) were among the failures that alerted the need to devise some means to tell the status of structures before anything worse happens. Consequently, a continuous health monitoring of structures is important and a mechanism should be developed by which efficient and accurate information could be obtained.
1.1. PROBLEM STATEMENT AND RESEARCH CONTRIBUTION

Researchers, hence, gave special attention to this discipline and proposed their own customized solutions in the last couple of decades which eventually gave birth to the science of SHM. SHM is thus one of the multidisciplinary fields that integrates the contribution of researchers from mechanical, electrical, civil and architecture engineering. Due to the easy access to, the wide availability and reliability of wired systems, many solutions have been implemented using wired sensor networks. High installation cost, need for specially trained professionals for setting up and maintain and their bulky nature made the research community to divert its attention towards WSNs. In addition, the advance of low cost wireless sensors has given a new dimension to the field.

Wireless Sensor Networks (WSN) is an upcoming technology which has a wide range of applications including infrastructure protection, industrial sensing and diagnostics, environment monitoring, context-aware computing (for example intelligent home and responsive environment) and so on. This kind of network usually consists of a large number of nodes that communicate together to form a wireless network. It is however essential to improve the energy efficiency for WSNs as the energy designated for sensor nodes is usually extremely limited.

The important thing one should have in mind, while considering the design of applications on wireless sensor networks, is that WSNs have the constraints of memory and energy consumption. Thus, designing customized systems involves trade-offs.

1.1 Problem Statement and Research Contribution

The use of wired sensors and data acquisition systems (DAQs) in SHM incur a high cost to purchase and setup the instruments on structures. The one which is in use at the Division of Structural Engineering in KTH, for example, is in the order of more than 300,000 SEK per unit. Densely instrumenting structures with these systems is thus economically infeasible. Hence, low-cost options are worthy to consider.

However, the use of WSNs for SHM systems is a significant task as there are some limitations that need to be addressed when it comes to utilizing reliable and robust monitoring systems. The fact that the process of SHM systems involves collecting huge amounts of data, applying complex data processing algorithms and extracting significant information, requires the use of high memory and processing power nodes. This poses an issue to the trend of currently available wireless sensor nodes with small memory size and processor power. Fully autonomous, fast and reliable systems to-date use wired systems. To completely shift the realm of SHM to a fully WSN-based solution, we need to tackle the challenges.

In this thesis, we explore the use of WSNs for SHM and implement a low-power and memory efficient data processing algorithm thereby paving the way for promising
1.2 Aim and Scope of the Study

The thesis work aims at understanding to what extent existing algorithm run on WSN devices in face of computing and memory limitations, studying the trade-off in terms of quality of the output versus resource consumption, and implementing a customized algorithm to better fit the characteristic constraints of WSN devices.

1.3 Survey of Selected Related Works

In this section, a review of some selected important works in the field of SHM is given. The review follows chronological order from the earliest to the latest with some jumps.

In 1998, Straser and Kiremidjian [44] were the first to develop a method for determining the health of a civil structure using wireless sensors on Alamosa Canyon Bridge, NM, USA. The method detected the general state of a structural immediately following a seismic event. They used a program at the core of their embedded analysis to indirectly measure the kinetic energy of a structure to detect when energy is dissipated during damage. It is better conceived as a decentralized damage detection method than a monitoring system as it measures the after-effects of a structural damage.

The Cooley-Tukey [20] implementation of FFT was successfully embedded in the computational core of a wireless sensing unit developed by Lynch and his research team [36]. The FFT embedded in the wireless sensing unit was utilized during field deployments of the wireless monitoring system to provide the Frequency Response Functions (FRFs) of instrumented structures. The accuracy of the complex-valued Fourier amplitude spectra computed by the wireless sensor node was shown to provide identical results to those generated by Matlab using the same time-history data.

While many researchers proposed the use of modal frequencies as a primary damage indicator, the method lacks the sensitivity to capture ambient vibrations in structures where environmental factors also contribute to modal frequency shifts [22]. To fully account for the environmental and operational variability of structures, a damage detection methodology based upon a pattern recognition framework was proposed by Sohn and Farrar [42]. An autoregressive (AR) time series model is fit to the stationary response time history of the structure.

Hackmann et al. [24] presented a way of structural monitoring using a holistic
1.4. THESIS OUTLINE

distributed damage localization technique where each sensor node calculates the partial curve fitting factor and then sends its data to the central station for final integration and correlation based damage localization.

The effect of time synchronization on the performance of WSNs for SHM that affects the quality and accuracy of mode shapes was addressed by V. Krishnamurthy et al. [33] using the popular frequency domain decomposition (FDD) technique.

Musiani et al. [40] described the design of an active sensor platform - Shimmer-that uses super capacitor to store solar energy harvesting that serves for twenty years once charged.

Recently, Ghuaco Feltrin et al. [23] used accelerometers and strain gauge sensors to measure vibration data. They built a separate sensor board supplied with separate power source from the host node to avoid errors that maybe introduced due to the initial sensor warm up phase while powered on during duty cycling.

1.4 Thesis Outline

As the work is multidisciplinary and readers from both the electrical engineering side and structural engineering side should fully be able to understand the concept, the thesis is organized in such a way that detailed analysis of the subjects is given to make the report as self-complete as possible.

- Chapter 2 presents a detailed background on SHM.
- Chapter 3 introduces WSNs and their use in SHM.
- Chapter 4 addresses the design and methodology that we used while doing experiments.
- Chapter 5 gives a detailed look into accelerometer selection.
- Chapter 6 explains the experimental evaluation to validate the designed SHM system.
- Chapter 7 Results and discussion are presents and analyzed as well.
- In Chapter 8 conclusions are drawn along with recommendations and insights on future works.
Chapter 2

Structural Health Monitoring

It has been mentioned in the previous chapter that the process of implementing a damage detection and characterization strategy for engineering structures is referred to as SHM. Damage may refer to all changes to the properties of a structure. It is an interdisciplinary study that involves identifying structural changes and effects on the overall integrity of the structure. In this chapter, we discuss the trends and practices of SHM system by analyzing frequently used approaches.

2.1 Trends in SHM Systems

There are several methods that are practiced among structural engineers in the study of changes in the health of civil engineering structures. Visual inspection is one of the predominant methods used quite often in studying changes and effects on structures. It is labor intensive and is only capable of observing changes on the surface of the structures.

The modern practice towards SHM is designing a continuous methodology for the study of structural changes. This has a wide benefit over traditional visual inspection methods by providing real-time, continuous analysis and detection of damage on civil structures. Thus, in order to avoid the failure of bridges and buildings, implementing SHM system to monitor the safety of structures is important.

2.2 Advanced Approaches in SHM

An advanced and alternative approach to visual structural inspections, SHM steps have been modified by Sohn [42]. These steps are:
2.3 SIGNAL ANALYSIS IN STRUCTURAL HEALTH MONITORING

1. Operational evaluation of damage identification process to features which are unique to the system being monitored.

2. Data acquisition, fusion, and cleansing which are used to collect sensor measurements data and to store in a centralized location.

3. Feature extraction and information condensation in identification of data features that allow one to distinguish between healthy and damaged part.

4. Statistical model development, concerned with the implementation of algorithms that operate on the extracted features to quantify the damage state of the structure.

2.3 Signal Analysis in Structural Health Monitoring

In bridge monitoring, accelerations, strains, deflections, temperature and applied forces, etc., are the important signals interesting to measure for analysis. These measurements are recorded for a given period which shows the time flow of a parameter under the study. It is difficult to see directly from the time domain data representation with what frequencies the bridge vibrates. The characteristics of a signal can be better understood in a frequency domain representation rather than the intuitive time domain representation. Therefore, it’s better to use a Fourier transform, a mathematical operation that decomposes a signal into its constituent frequencies. Currently, most SHM systems depend on measuring structural dynamics characteristics and analyzing these data in the frequency domain by performing modal analysis which directly involves Fast Fourier Transform (FFT).

2.3.1 Sampling Theorem

The Nyquist theorem states that any continuous baseband signal (signal extending down to zero frequency) may be similarly reconstructed if the signal is bandwidth limited and the sampling frequency is at least twice the bandwidth of the signal. If a time-domain signal is sampled uniformly, then the frequency corresponding to one-half that rate is called the Nyquist frequency, $N_1$

$$N_1 = \frac{1}{2\Delta t}$$  \hspace{1cm} (2.1)

where $\Delta t$ is the time between successive samples. Appropriate choice of sampling interval is necessary for obtaining correct representation of the original signal.
2.3. SIGNAL ANALYSIS IN STRUCTURAL HEALTH MONITORING

2.3.2 Fourier Transform

The Fourier transform is a mathematical transform that converts the time domain waveform into the frequency domain. Hence, Fourier Transform produces a complex number valued sinusoid which can be displayed with two parts, either with the real and imaginary part or with magnitude and phase. When this sinusoid of different frequency components is summed, it would give the original waveform. The Fourier transforms $F(w)$ of a function can be mathematically expressed from the time domain data $f(t)$ as:

$$F(w) = \int_{-\infty}^{+\infty} f(t)e^{-iwt} dt. \quad (2.2)$$

and the inverse Fourier transform as:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(w)e^{iwt} dw. \quad (2.3)$$

where $i$ is the imaginary value and in both cases, $i = \sqrt{-1}$. The complex valued output can be expressed as the sum of two terms

$$F(k) = R(k) + iI(k) \quad (2.4)$$

where $R(k)$ is the real component and $I(k)$ is the imaginary component. The magnitude is:

$$|F(k)| = \sqrt{R(k) \ast R(k) + I(k) \ast I(k)} \quad (2.5)$$

and the phase can be calculated as:

$$\angle f(x) = tan^{-1} \left( \frac{I(k)}{R(k)} \right) \quad (2.6)$$

2.3.3 Discrete Fourier Transform (DFT)

DFT is essential because it takes a discrete signal in the time domain and transforms that signal into its discrete frequency domain representation while the inverse DFT performs the reverse operation from frequency domain to an equivalent time domain sequence. It transforms a discrete signal to a sum of sinusoidal-shaped signals, all with their different amplitudes, frequencies and phases. The DFT of $N$ uniformly sampled data points $x_j$ (where $j = 0, \ldots, N-1$) and its inverse are defined in mathematical representation as:

$$X_k = \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N} \quad (2.7)$$

and

$$x_j = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i2\pi jk/N} \quad (2.8)$$
2.4 Categories of Structural Health Monitoring

2.3.4 Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is an efficient mechanism for transforming time domain data into frequency domain. This fast implementation of discrete Fourier transform reduces the number of computations needed for N-point DFT from $O(N^2)$ to $O(N \log_2 N)$ in (Radix-2 Cooley-Tukey) [22]. The only requirement of the most popular implementation of this algorithm is that the number of points in the series has to be a power of 2. The computing time for the radix-2 FFT is proportional to $N$. The FFT considerably reduces the computational requirements of the DFT. For example, if we take a transform on 1024 points using the FFT it is about 100 times faster than using the DFT, which is a significant speed increase.

The frequency resolution is given as:

$$f_r = \frac{f_{Nyquist}}{N/2} = \frac{2f_{Nyquist}}{N}$$  \hspace{1cm} (2.9)

Where:

- $f_r = \text{frequency resolution}$
- $f_{Nyquist} = \text{Nyquist frequency}$
- $f_{max} = \text{maximum frequency}$

Hence, to determine the corresponding frequency, $f_N$, at particular sample point. It can be written as:

$$f_N = f_r \times i$$  \hspace{1cm} (2.10)

Where $i = \text{instantaneous sample number point}$

2.4 Categories of Structural Health Monitoring

SHM process involves the study of a system over time by collecting measurements via data acquisition systems, extraction of damage-sensitive features from these measurements, and making statistical analysis of these features to determine the current state of system health. In order to make all these detailed analysis we need to implement data processing algorithms in efficient manner. Until recently qualitative and non-continuous methods have been used to evaluate structures for their structural health to provide their intended use.

In the last few years SHM technologies have developed well due to the advancement in the other related science and technology fields. SHM system can be wired or wireless. Due to the technology development and the benefits that wireless systems have over the wired system, wireless SHM became an interesting research area. The wired system is robust and reliable, but it is
costly and less flexible. On the other hand, the wireless systems are cheap, flexible and easy for deployment. The drawbacks with the wireless systems are problems with lifetime, memory, reliability and also having energy efficient data processing algorithms [36]. Taking the advantage of those wired system and customizing to wireless system is a major development to SHM in the wireless network environment. For the appropriate choice of efficient data processing algorithm having the background of these data processing algorithms is important. The goal of this chapter is to give a brief background of data processing algorithms in relation to wireless sensor networks ability to process huge volume of data, reliability and flexibility. We will give brief description of existing methodologies for structural health monitoring.

A large number of sensors are used in most SHM system to study the structural elements of which include accelerometers, strain gauges, displacement transducers, temperature sensors, etc. After collection of these measurements, data has to be extracted and appropriate analysis should be done to investigate the general condition of the system. There are broadly two categories of SHM: Feature extraction and modal analysis.

### 2.4.1 Feature Extraction

Feature extraction algorithm depends on features or signatures gathered from the recorded structural response signals such as acceleration, strain or other data that change with the occurrence of damage. Then, the measured vibration responses are used to identify damage-sensitive properties which allow one to distinguish between the damaged and the healthy structure. Such algorithms mainly involve the following steps:

i. The evaluation of a structure’s operational environment which include loading conditions, temperature and humidity.

ii. The acquisition of structural response measurements and data preprocessing.

iii. The extraction of features that are sensitive to damage and

iv. The development of statistical models for feature discrimination

The downfall of this method is that it needs previous record of damage features and development of statistical models in order to monitor structural changes. Therefore, this process is computation intensive and requires more application of artificial intelligence at the sensor nodes which lead to the design of power-hungry algorithms [32].
2.4. CATEGORIES OF STRUCTURAL HEALTH MONITORING

2.4.2 Modal Analysis

Modal analysis has been widely applied and commonly used approach in vibration based structural health monitoring in aerospace, mechanical and civil engineering to investigate the integrity of structures based on natural frequency, modal damping, and a mode shape. Modal analysis describes a structure in terms of its natural characteristics which are frequency, damping ratio and mode shapes. The objective is the evaluation of modal parameters of a structure under ambient vibrations and dynamic loads. Modal parameters provide information that help to detect changes in structure.

In modal analysis, we use Frequency response function (FRF) as a mathematical representation to describe the relationship between the input and the output of a system. FRF shows how much displacement, velocity, or acceleration response a structure has at the output point with respect to input force; if we take measurement of the time data and transform it to the frequency domain using FFT. Then, looking at the FRF we will see some very interesting characteristics of the structure as we can see, e.g., in Figure 2.1. We will be able to identify that there are peaks in this function which occur at the resonant frequencies of the system. Modal analysis can further be classified as output only and input/output based on the excitation [6]. This section gives a brief introduction to Modal parameters.

![Figure 2.1: Frequency response function.](image)
2.4. CATEGORIES OF STRUCTURAL HEALTH MONITORING

**Single degree-of-freedom**

The simplest vibratory system that can be described by a single mass connected to a spring. The mass is allowed to travel only along the spring elongation direction. Such systems are called Single Degree-of-Freedom (SDOF) systems and are shown in Figure 2.2, where $m$ is the mass, $c_v$ is the viscous damping coefficient, $k$ is the stiffness, $x$ is the absolute displacement of the mass, $l_o$ is the base input displacement.

![Figure 2.2: Single degree of freedom model.](image)

**Natural frequency**

When an object is hit and the external force is removed it will vibrate at its natural frequency. The natural frequency is defined as the number of times a system will oscillate (move back and forth) between its original position and its displaced position assuming there is no outside interference. The natural frequency can be calculated by the formula

$$ f = \frac{1}{2\pi} \sqrt{\frac{k}{m}} $$

where $k$ is the stiffness and $m$ is the mass of the structure.

**Damping Ratio**

Damping ratio, a dimensionless measure that tells us how oscillations in a system behave after an excitation or due to environmental loadings. For determining damping ratio from frequency domain data, half-power bandwidth (HPD) method can be used. In this method, for each natural frequency there
2.4. CATEGORIES OF STRUCTURAL HEALTH MONITORING

is a peak in FRF amplitude and 3 dB down from the peak there are two points corresponding to half power points which are depicted in the Figure 2.3. HBD is defined as the ratio of the frequency range between the two half power points to the natural frequency at this mode where $f_a$ and $f_b$ are the frequencies associated with the half power points on either side of the peak as shown in Figure 2.3. $A_1$ is the amplitude at the peak $A_2$ can calculated as

$$A_2 = \frac{A_1}{\sqrt{2}}$$ (2.12)

Thus damping ratio associated with each natural frequency can be mathematically calculated as obtained using the formula

$$\zeta = \frac{f_b - f_a}{f_b + f_a}$$ (2.13)

**Mode Shape**

Modes are defined as inherent properties of a structure, and are determined by the material properties (mass, damping, and stiffness), and boundary conditions of the structure. Different mode shapes will be related with different frequencies and each mode has a natural frequency associated with it. Basically, there are characteristics that depend on the weight and stiffness of the structure that determine where these natural frequencies and mode shapes will exist. Mode shapes are helpful because they represent the shape that the structure will vibrate and simplifies the vibration response of a complex structure into a set of modal parameters that can be easily analyzed. Experimentally to determine the mode of a structure, modal testing is done, e.g, by measuring operational deflection shapes (ODS) which are deflections of a
structure at a specific particular frequency then post processing them in a specific manner to define mode shapes \[13\]. Mode shapes do not have unique value but they are unique in shape. These mode shapes can be arbitrarily scaled to any set of values and the relationship of one shape component to the other is distinct. One of the common ways to scale mode shapes is to scale them so that the modal masses are one (unity).

Modal Testing is based on the estimation of a set of FRFs relating the applied force and the corresponding response at several pairs of points along the structure, with enough high spatial and frequency resolution \[6\].

\subsection*{2.5 Input-Output Modal Analysis}

Input-Output modal analysis is based on the estimation of a set of FRFs relating an applied force to the corresponding response at several points along the structure. In small and medium-sized structures, the excitation can be induced by an impact hammer (Figure \ref{fig:impact-hammer}) or special impulse devices specifically designed to excite bridge and large buildings. The main problem in input output modal analysis associated to the performance of forced vibration tests in large engineering structures like bridges, dams and high-rise buildings, is the difficulty to excite with sufficient energy and in controlled manner, which needs a sophisticated testing instrument. Thus, it would not be a good option for using input-output modal analysis in WSN for structural health monitoring.

\subsection*{2.6 Output-Only Modal Identification}

Output only modal identification is the study of a response under ambient excitations which usually includes temperature, humidity, wind or under traffic loadings. Random forces such as vehicle trafficking, wind excitation, etc.
2.6. OUTPUT-ONLY MODAL IDENTIFICATION

may act on bridges and excite the structure together so it’s almost impossible to measure all these forces simultaneously. If the forces are not measured correctly, then input-output modal analysis cannot give accurate estimates of the modal parameters. Natural conditions under which structure operates are very difficult to be produced in the laboratory. So, output-only modal analysis is the best option available in such conditions as it depends on the responses only and the responses could be measured with high accuracy both on wired and wireless environments. One assumption made in output only modal analysis is the excitation input is taken as zero mean Gaussian white noise. The advantage to measure the natural (or ambient) response and then estimate the modal parameters by performing an output-only modal identification includes.

- The test is inexpensive and fast; since special equipment for excitation is not needed. Thus, there is no need for vibration shake or impacts hammer
- The test can be performed with out creating obstacle for normal operation of the structure.
- The real operating conditions of the structure are represented by the output response of FRF.

The great advancement in sensing technologies such as MEMS sensors, wireless sensors, very high computational efficiency and communication technologies allows to record ambient response of a structure accurately. This development makes output modal identification method preferable for monitoring structural changes. With output-only modal identification it has been realized the instrumentation of different bridges and buildings with wireless monitoring systems\[27\]. Mathematical models on the output only identification technique can roughly be classified as:

I. Parametric methods: These are methods that involve time domain analysis.
   - Ibrahim Time Domain (ITD)
   - Eigensystem Realization Algorithm (ERA)
   - Random Decrement Technique (RDT)
   - Stochastic Subspace Identification (SSI)

II. Non-parametric methods: These are methods that involve frequency domain analysis.
   - Peak Picking (PP)
   - Frequency Domain Decomposition (FDD)
   - Enhanced Frequency Domain Decomposition (EFDD)
   - Curve-Fitting Frequency Domain Decomposition (CFDD)
These methods are advantageous on natural frequency and mode shape extraction. There is uncertainty in damping ratio estimation but the EFDD and CFDD perform better than FDD in this regard. In the next subsections we will see in depth non parametric methods for structural health monitoring for their implementation in wireless sensor networks.

### 2.6.1 Peak Picking Method (PP)

Peak picking (PP) method is an output-only modal analysis technique for estimating the modal properties of a structure. The method is based on the values of PSD spectrum that show changes in extreme values around the natural frequencies. PSD shows the strength of the variations (energy) as a function of frequency. Thus, it shows at which frequencies variations are strong and at which frequencies variations are weak. The frequency at which this extreme value occurs considered a good estimate for natural frequency of the system. In this method the natural frequencies are determined from the observation of the changes in peaks on the graphs of the averaged normalized power spectral densities (ANPSDs)\(^\text{10}\). The ANPSDs are basically obtained by converting time data to the frequency domain by performing DFT. The main drawback of this method is it is difficulty in identifying closely spaced modes and results are highly biased in modal frequencies values and mode shapes in case of closely spaced modes. Even though PP has this weak point, the method is strong and useful since the identification is very fast.

In structural health monitoring, using WSN the implementation of peak picking algorithm starts with a user setting the maximum number of peaks which will be determined by the algorithm. Next, using data acquisition system, a set of acceleration time history data will be collected at each sensor node and converted to a PSD using an embedded FFT algorithm. Each node picks largest peaks from its PSD function by searching for frequencies at which the value of the PSD found to be greater than some threshold level set by the user. If less than a preset number of peaks are found, zeros will be returned in place of the missing peaks \(^\text{50}\). This implementation as a method is strong, fast and useful for indication of changes in structures \(^\text{34}\). As we have constraints with memory and power in the wireless sensor network this algorithm greatly reduces the amount of data to be transmitted by the wireless sensing network which will result in energy efficient algorithm.

### 2.6.2 Frequency Domain Decomposition (FDD)

FDD is a popular technique based on decomposing the power spectral density functions matrix at each discrete frequency using the Singular Value Decomposition (SVD) algorithm, for more understanding of this method refer a paper by Brincker \(^\text{15}\). It decomposes the spectral matrix response into a set of...
2.6. OUTPUT-ONLY MODAL IDENTIFICATION

single degree-of-freedom systems, each corresponding to one individual mode. In FDD method, the mode shapes are estimated as the singular vectors at the peak of each auto power spectral density function corresponding to each mode. The results can be accurate if the structure assumed is lightly damped.

In wireless sensor network using FDD the method is to have each sensor send its vibration data to a central sensor node which computes the SVD of the PSD power spectral density matrix and then distributes the mode shapes back to each sensor \[29\]. This would require significant computational power and memory both at the sensor node and at the central node.

FDD method can only be applied in the frequency domain if the dominant frequencies are determined at a priori. This method has played a major role in WSN SHM and has been implemented for output-only system identification through the ambient vibration measurements due to its reliability, straightforwardness and effectiveness \[15\]. The drawback of this method is with uncertainty in damping ratio estimation. Other methods like EFDD and CFDD which are the extension of the classical FFD would perform better with determining damping ration than FDD.

2.6.3 Enhanced Frequency Domain Decomposition

EFDD technique is an extension to the FDD technique. In this method, the auto spectral density functions are transformed into the time domain using IFFT, resulting in autocorrelation functions for each mode of a system \[17\]. Extraction of modal parameters are done from the inspection of the decay of auto correlation functions.

2.6.4 Curve Fitting Frequency Domain Decomposition

This method is an extended new version of the Enhanced Frequency Domain Decomposition (EFDD). The auto-spectral function is curve-fitted directly in to the frequency domain using SDOF curve-fitter to produce high-quality estimates of the natural frequency and damping ratio. There is no need of converting spectral density functions in to time domain using Inverse Fourier transform as needed in EFDD method. \[26\].

In summary, for effective determination and study of overall integrity of a structure accurate data processing algorithms in SHM are required. The design of continuous real time monitoring of changes in structural elements is needed to avoid failures in bridges, dams and civil structures. Thus, identifying changes by using output-only modal analysis without caring about input excitation and determining modal parameters is important for clearly obtaining changes in structures. There exist two categories within the output-only identification method, namely as time domain and frequency domain analysis.
Due to their computation intensiveness, the need of prior knowledge of the mathematical model and the need of database in feature extraction we prefer frequency domain analysis. These methods are mainly on PSD response, which are advantageous on natural frequency and mode shape determination. SHM using WSN has the advantage of flexibility, low cost and easy deployment as compared with wired system, but it suffers from memory and energy constraints. Therefore, appropriate choice of and implementation of data processing algorithms on WSN is a good step moving to an advanced WSN SHM system. Parametric methods like ITD, ERA or RDT are advantageous for estimating modal damping ratio, but have difficulty in determining natural frequencies and mode shapes extraction. On the other hand, nonparametric methods such as PP, FDD or EFDD are better on determining natural frequencies. Among the non-parametric methods, PP is selected for our work in this thesis because of its fast and less energy demanding requirement.
Chapter 3

Wireless Sensor Networks for SHM

In the science of instrumentation, there is a trend of adopting more recent achievements of sensing technologies into the traditional ones. In the latter, wireless sensors are one of the emerging technologies that lessen the tedious works imposed while using wired sensors. They take the sensing technology into a new realm giving new ways of sensing paradigm. Mass production

Figure 3.1: Example application of SHM using WSN. From [23].

and wide utilization of wireless sensors is made possible due to the extreme progress of micro electro-mechanical systems (MEMS) technology in the production of very small counterparts of huge mechanical tools. Thanks to this technology, today almost all sensors are being replaced for special applications that demand the integration of very small components into a system yielding a great reduction of cost, size and also ease of deployment [8].

As it is the case in other fields, it has been quite a while since the adoption of wired sensor applications in structural health monitoring to WSNs paved its
way into the field of structural engineering. A reliable SHM requires a dense instrumentation of structures which is infeasible deploying such structures with tethered sensors [18]. Fortunately, it is possible to densely deploy wireless sensors with comparatively reduced costs and this gives wireless sensors advantage over the traditional wired sensors.

Beyond just as a simple replacement of wired sensor applications due to their high savings on budget, WSNs have gained also a great attention due to their great distributed signal processing capability. They come up with analogue to digital converter (ADC) and microcontrollers integrated in one unit. The presence of a microcontroller helps the sensor node to apply some simple signal processing algorithms. This capability gives the whole system an additional feature to locally process collected data than sending the whole bulk of data to the central base station.

In this chapter, the overall features of WSNs for structural health monitoring is presented. Hardware and software architectures are reviewed.

### 3.1 Applications of WSNs

Inspired by the developments of battle field surveillance applications in military [37], their application has shown much more progress that such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, health care applications, home automation, and traffic control [8]. Figure 3.1 shows an application of WSNs for bridge monitoring.

### 3.2 Wireless Sensor Nodes

A sensor node, also known as a mote is the basic building block of a wireless sensor node [40]. Every node in a WSN is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. The size of a sensor node might vary depending upon the number of components integrated on the unit [31]. According to Moore’s law and the fast progress of nanotechnology, the size is even expected to diminish extensively.

Today, a wide variety of commercial and academic wireless sensor node prototypes have been developed and validated. A detailed summary of academic wireless sensing unit prototypes is found in [37].
3.3 Hardware Components of a Wireless Sensor Node

Each sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery and/or an embedded form of energy harvesting [21]. Many off-the-shelf sensor nodes on the market come up with all of the above components integrated with one or more sensors. Nevertheless, there comes a situation in...
3.3. HARDWARE COMPONENTS OF A WIRELESS SENSOR NODE

which the hardware components that are pre-built on the node do not suffice for a specific project at hand while designing a system that works with a wireless sensor node. In such cases, the engineers are required to assemble hardware components together to form the customized wireless sensor node. Attaching a separate sensor board is quite a common practice [37] and seldom independent power supply is provided for such boards while considering duty cycling to avoid some measurement errors due to the time and power needed to warm up those external devices.

A detailed explanation on each hardware component follows.

3.3.1 Microcontroller

The microcontroller is the core component of a sensor node which performs several tasks, processes data and controls the functionality of other components in the sensor node. It is the most common controller type in embedded systems. It has a set of hardware components built around a central processing unit (CPU), Figure 3.4.

![Internal components of a microcontroller](image)

Figure 3.4: Internal components of a microcontroller.

The CPU controls a range of peripherals with both digital and analogue functions such as timers and analogue to digital converters. Small microcontrollers usually incorporate volatile and/or nonvolatile memory. They can easily be programmed with C or C++. They are cheaper, flexible to connect to other devices, easier to program and consume a very low power when compared to digital signal processors (DSPs), Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs) [30].

3.3.2 Transceiver

Transceivers designate the communication interface of the sensor node with the functionality of both transmitter and receiver combined into a single device. Communication is the most energy demanding process in wireless sensor
3.3. HARDWARE COMPONENTS OF A WIRELESS SENSOR NODE

nodes and care must be taken while selecting a communication means. Sensor nodes often make use of the industrial, scientific and medical (ISM) radio band which gives free radio, spectrum access and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication, Bluetooth and infrared. Infrared and optical communication media need no antenna and require less energy, but need line-of-sight for communication and are sensitive to atmospheric conditions. Bluetooth has a small range. RF based communication is the most common means that fits most of the WSN applications. WSNs tend to use license-free communication frequencies: 173Hz, 433Hz, 868Hz, and 915M Hz; and 2.4G Hz. The operational states are transmit, receive, idle, and sleep.

3.3.3 Memory

When it comes to processing large amounts of data for quite long time, we need at least a temporary storage to store a transitional bulk of data before, during and after processing. In such cases, the built in memory on the microcontroller hardly suffices. However, from an energy efficiency point of view and speed of operation, the on-chip memory of a microcontroller and flash memory are most relevant [8]. Flash memories are typically used due to their low cost and better storage capacity. Two categories of memory are used based on the purpose of storage: RAM, random access user memory, which is used for storing application related or personal data, and ROM, read only program memory, that is dedicatedly used for storing data used for programming the device. The MSP430F2617 has 92KB of Flash memory for program and data and 8KB SRAM, a static random access memory, which is common in microcontrollers [39], that retains its data even when the clock is stopped provided that power is maintained.

3.3.4 Power source

The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process. Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. As wireless sensor nodes are typically very small electronic devices, they can only be equipped with a limited power source of less than 0.5-2 AH and 1.2-3.7 V [8].

3.3.5 Sensors

Sensors are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure. They can be analogue
3.3. HARDWARE COMPONENTS OF A WIRELESS SENSOR NODE

or digital sensors based on the nature of the output they give to the external world. In case of Analogue sensors, an external ADC is required to convert the analogue signals to digital. The ADCs can be integrated in the microcontrollers or they can be integrated into the same circuit board with the microcontroller and/or the sensor board. Digital sensors have built in configurable ADCs hence we can bypass the external ADCs of the microcontroller.

Figure 3.5: The 3-axis ADXL MEM accelerometer mounted on a circuit board.

Sensors are commonly built on a separate sensor board as can be seen in Figure 3.6 and are interfaced with the main sensor node externally. In such cases, the host node should contain an interface to which sensing boards can be plugged. The sensing interface is largely responsible for converting analog output of sensors into a digital representation that can be understood and processed by digital systems.

Figure 3.6: An example sensor board with accelerometer sensor on board.
3.3. HARDWARE COMPONENTS OF A WIRELESS SENSOR NODE

The quality of the sensor interface is a function of the conversion resolution, sampling rate, and number of channels available on its ADC. For most structural monitoring applications, an ADC resolution of 16-bits or higher is preferred. Ordinarily, low sampling rates of less than 500 Hz are adequate for structural monitoring [23].

3.3.6 Energy harvester (Optional)

Energy harvesting (also known as power harvesting or energy scavenging) is the process by which energy is derived from external sources (e.g., solar power, thermal energy, wind energy, salinity gradients, and kinetic energy), captured, and stored. This is an optional component which gives a supplementary source of power to the battery.

Figure 3.7: Solar panel and rechargeable battery for Jindo Bridge SHM [41]

Energy harvesters provide very small amount of power for energy constrained systems such as wireless networks to increase their life time by extracting energy from their environment [7]. Energy harvesting mechanisms based on solar cells are the ones that are widely used and there are some wireless sensor units that are incorporated in a box with solar cells on it cover, Figure 3.7. But, for places like Sweden, where the sun light is not such a reliable source of energy, other energy harvesting options such as vibration based piezoelectric energy harvesters would be more feasible. To the best of our knowledge, although few successful products are available with such features, they are not yet applicable for small devices like wireless sensor units. But, we hope that they would be available soon and will help in the long time deployment of WSNs.
3.4 Software Components of a Wireless Sensor Node

The CPU spends much of its processing power performing some calculations on the data collected by sensors and interacting with peripherals. Both the algorithms to do the computations on sensor nodes and the policy to interact and coordinate all the operation functions should be defined and loaded to the node. These are software components of a wireless sensor node loaded into the microcontroller for debugging through bootstrap loader (BSL) and are used to drive the hardware components and the CPU. These are broadly classified into operating system and application software.

3.4.1 Operating systems

An operating system (OS) is software that manages hardware resources and provides common services for execution of various application software [43]. It acts as an intermediary between application programs and the hardware for input output and memory allocation. Operating systems are found on almost any device that contains a microprocessor or a microcontroller, from tiny to much complex systems. Unlike other large systems with larger processors, operating systems for WSN nodes are typically less complex and take kilobytes of memory rather than megabytes [3]. They are written with the memory and power consumption constraints of the node in mind. Operating systems on general purpose computers like the desktop are complex with multitasking features in which several applications and tasks run concurrently on a device. This requires the availability of large amounts of memory and power on board.
3.5 CHALLENGES IN USING WSNS

to run all those processes. In contrast, OS in wireless sensor nodes are basically of event driven systems to load and unload applications on the encounter of interrupts from hardware or software. Moreover, the operating systems for wireless sensor networks should be portable across a range of hardware platforms.

Based on the above requirements, several operating systems for WSNs have been proposed, written and deployed so far with a focus of one or more functionality. To name some: TinyOS, Contiki, a virtual Java machine based MagnetOS [9], SensorWare with an abstract scripting language for programming sensors [12], the Mantis system that used a traditional preemptive multi-threading model of operation [1], etc.

TinyOS is perhaps the earliest operating system that addresses the specific constraints and limitations of motes [35]. It is a tiny (the core is less than 400 B), flexible, low power, application specific operating system for sensor networks. It is written in NesC, a dialect of C, programming language and developers also use this language to write applications.

As this thesis is done with Conitiki, we give a detailed explanation of the Contiki operating system latter in section 4.2.1.

3.4.2 Application programs

Application programs are any software in the mote other than the operating system that determine the way the node accomplishes specifically tailored applications. These programs should be developed with algorithms which are simple and efficient that ensure effective utilization of memory and power consumption of the node. The number of instruction cycles that each piece of program runs should also be limited so that the flow of the control would not end up in a long loop that eventually triggers the watchdog timers to reset the hardware. In our case, the FFT algorithm and the peak picking algorithms are carefully selected in accordance to the requirements.

3.5 Challenges in Using WSNs

The small physical size and simplicity nature of the components that make up the motes results in corresponding constraints on resources such as memory, computational speed, communications bandwidth and power supply [5]. Hence, selection of an appropriate wireless sensor node is the first important step towards the design of applications, operating systems, and even other plug-in components. Because, the performance of the entire system is dependent upon the individual node capacity and applications should be developed with those constraints in mind. We discuss the major constraints as follows.
3.5. CHALLENGES IN USING WSNS

3.5.1 Memory constraints

Most sensor nodes make use of only the memory available in the microcontroller which is not large enough to give room for memory demanding operating systems, processes and data. The MSP430F2617 microcontroller, for example, has 92 KB of flash memory for program and data and 4 KB of RAM for data. All software and data should fit into this small compartment. Thus, proper selection of small size operating system, less bit representation of data like data type integer shorts instead of floating points, developing applications with algorithms that efficiently utilize the memory would help to tackle this problem.

3.5.2 Power consumption constraints

When it comes to power consumption, sensor nodes operate with limited-power-supplied batteries. This is a decisive thing that determines the life span of the sensor network. Hence, power should be redeemed from being depleted for long term deployments. Changing the batteries frequently would augment the maintenance costs severely putting one of the main assets of the WSN monitoring systems into jeopardy; not forgetting the fact that the WSN monitoring systems could be deployed in remote and/or risky places like high buildings and bridges.

Possible ways of power consumption reduction are:

a) **Low-power components**: building the mote from low power components (sensors, microcontrollers, radio transceivers) that need less input power and expend less amount of power [11].

b) **Duty cycling**: keeping the node switched-on for a fraction of time within a given period of time. Care should be taken in such cases that some devices, e.g. the radio unit, draw quite a significant amount of power when they are powered on; and others to warm-up introducing error in data collection. This corruption, however, can be eliminated by supplying a separate power source to those components continuously supplying low power [23].

c) **Node-level data processing**: since the energy cost of transmitting 1 Kb data a distance of 100 m, e.g., is approximately the same as that used for the execution of 3 million instructions by a 100 million instructions per second (MIPS) processor [21], sending information instead of raw data is beneficial when monitoring vibration based processes which produce large samples of raw data. This strategy is the most powerful energy saving method for a longer system lifetime.
Chapter 4

Design and Methodology

The design and methodology that we followed is based on the resources that we have at hand. As the aim of the thesis is to implement a data processing algorithm on WSNs, the environment on which the designed target algorithm would be implemented should be considered carefully. In this chapter, we address the detailed explanation of the methodology and tools that we used throughout our experiments. We look into the hardware and the software sections for both the wired and wireless measurement systems used from the beginning of the thesis work.

4.1 Hardware Used

We used two sets of hardware for measurement. Wireless system and wired system.

4.1.1 Wireless system

We used different wireless sensor nodes during our thesis work. A t-mote sky, a Sentilla Jcreate node and the Z1 Zolertia node. The first two were not suitable for our thesis as a t-mote sky does not have accelerometer while the accelerometer in Sentilla node was of low quality and unreliable. Thus, the primary hardware that we used in the wireless system is the Z1 Zolertia sensor node. Its functional block diagram is given in Figure 4.1. It is one of the recently developed sensor nodes to test applications and prototypes with the best trade off between time of development and hardware flexibility. It is a general purpose low-power WSN module that can be used as a development platform for WSN applications. Its core architecture is based upon the MSP430+CC2420 family of microcontrollers and radio transceivers respectively by Texas Instruments, which makes it compatible with other motes,
4.1. HARDWARE USED

like Crossbow’s TelosB, Moteiv’s Tmote sky, and alike. Immersing devices in the Internet of Things (IoT), Personal health care monitoring, environmental monitoring, emergency detectors, safe and rescue devices, long-term unattended monitoring, energy metering, agricultural monitoring are some of the applications that the mote is entitled for. It has a temperature and accelerometer sensors already included on board.

![Functional block diagram of Z1 WSN module.](image)

Based on the hardware components of a sensor node from Section 3.3, a detailed explanation on hardware component which are relevant to this thesis follows.

**Microcontroller**

The Z1 Zolertia mote, see Figure 3.2, is based upon the MSP430 family of microcontrollers from Texas instruments, specifically a 64 pin MSP430F2617 [51]. MSP stands for mixed signal processor (analogue or digital) [39] and the processor has ADC12, a 12-bit 8 channel analogue to digital converter for mixed signal use. The F indicates that the program memory in the microcontroller is of flash memory, the most common type of memory that can be both programmed and erased electrically. It is a 16-bit processor, both the address bus, the data bus and all the 16 registers in the CPU are 16-bits, with a Von Neumann architecture, designed for low-power applications. Its CPU is often described as a reduced instruction set computer (RISC) in which simplified sets of instructions are provided for faster execution.

In addition, other feature of the MSP430 that makes it preferable for low power applications is that the device can easily be put into a low-power mode. The
4.1. HARDWARE USED

Table 4.1: Summary of parametric features of MSP430F2617.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>support for a 32768 Hz</td>
</tr>
<tr>
<td>Flash</td>
<td>92 KB</td>
</tr>
<tr>
<td>Static RAM</td>
<td>8192 B</td>
</tr>
<tr>
<td>Timers - 16-bit</td>
<td>2</td>
</tr>
<tr>
<td>Multiplier</td>
<td>16x16</td>
</tr>
<tr>
<td>ADC</td>
<td>12bit 8 channels</td>
</tr>
</tbody>
</table>

mode can be controlled by bits in the status registers without the need for special instructions. Many of the peripherals can run autonomously without the CPU for most of the time and the processor can be awakened from a standby mode rapidly by an interrupt and returns automatically to its low-power mode after handling the interrupt. This feature is extremely important in structural health monitoring as the state of the device is controlled based on the traffic load on the structure. Table 4.1 summarizes some parametric features of MSP430F2617.

Transceiver

The communication module in Z1 Zolertia is a Texas Instruments CC2420 \cite{51}, a 48 pins, low cost 2.4 GHz IEEE 802.15.4 compliant RF transceiver, Table 4.2.

Table 4.2: Summary of CC2420 features.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX Current Consumption</td>
<td>19.7 mA</td>
</tr>
<tr>
<td>TX Current Consumption</td>
<td>17.4 mA</td>
</tr>
<tr>
<td>Data Rate(Max)</td>
<td>250 Kbps</td>
</tr>
<tr>
<td>Frequency Range</td>
<td>2400 Hz - 2483.5 Hz</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-40℃ to 85℃</td>
</tr>
<tr>
<td>Operating voltage range</td>
<td>2.1 V - 3.6 V</td>
</tr>
<tr>
<td>Output Power Range</td>
<td>-25 dBm to 0 dBm</td>
</tr>
<tr>
<td>RSSI Output</td>
<td>Digital</td>
</tr>
</tbody>
</table>

It is connected with an integrated Yageo/Phycomp ceramic antenna \cite{51}. External antennas can be used when large range communication is needed.
4.1. HARDWARE USED

Memory

The MSP430F2617 has 92KB of Flash memory for program and data and 8KB RAM for storing temporary data. In addition to these on chip memories, it is integrated with an external flash memory. This can be used as extension to store data while programming or collecting data; however, care should be taken while doing so as writing and reading operations from this memory take extra time and CPU cycles which add unnecessary cost to the entire system.

Power supply

The Z1 Zolertia mote should nominally be powered at 3V although it may work partially or totally from 1.8V, without radio, to 2.1V, with radio. The absolute maximum rating for the whole component is, Table 4.3, in the range -0.3V to +3.6V. No battery is needed if it is connected to the mini-USB port for programming or communication.

<table>
<thead>
<tr>
<th>Component</th>
<th>Operating Range</th>
<th>Current Consumption</th>
<th>Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430F2617</td>
<td>1.8 V to 3.6 V</td>
<td>0.1 µA</td>
<td>OFF Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5 µ A</td>
<td>Standby Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5 mA</td>
<td>Active Mode @ 1 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;10 mA</td>
<td>Active Mode @ 16 MHz</td>
</tr>
<tr>
<td>CC2420</td>
<td>2.1 V to 3.6 V</td>
<td>&lt;0.1 µA</td>
<td>OFF Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 µA</td>
<td>Power Down</td>
</tr>
<tr>
<td></td>
<td></td>
<td>426 µA</td>
<td>IDLE Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18.8 mA</td>
<td>Rx Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17.4 mA</td>
<td>Tx Mode @ 0 dBm</td>
</tr>
<tr>
<td>ADXL345</td>
<td>1.8 V to 3.6 V</td>
<td>0.1 µA</td>
<td>Standby</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40 µA to 145 µA</td>
<td>Active Mode</td>
</tr>
<tr>
<td>M25P16</td>
<td>2.7 V to 3.6 V</td>
<td>1 µA</td>
<td>Deep Power Down</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40 mA to 15 mA</td>
<td>Active Mode</td>
</tr>
<tr>
<td>TMP102</td>
<td>1.4 V to 3.6 V</td>
<td>1 µA</td>
<td>Shutdown Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 µA</td>
<td>Active Mode</td>
</tr>
</tbody>
</table>

Sensors

The Z1 Zolertia mote comes with TMP102 Thermometer from Texas Instruments and ADXL345 digital accelerometer from Analogue Devices. We used the accelerometer to measure vibration data from our lab beam model.
4.2. SOFTWARE USED

4.1.2 Wired system

The wired system was used to validate the instruments and the algorithms used in the wireless system. The system available in the division of structural engineering and bridges, KTH consists of a high resolution 24-bit data acquisition system HBM MGC Plus\(^1\), and a very sensitive, extremely low-noise level measuring Si-Flex™ MEMS accelerometer and a laptop computer loaded with Windows XP operating system. Although high quality data is collected using this system, it incorporates a bulky set of cables and huge high cost hardware equipments depicting the wireless system replacement of the overall system quite a big relief.

4.2 Software Used

In this section, we discuss the software tools that we used. They are subcategorized as realtime and offline software tools.

4.2.1 Realtime software tools

By realtime software tools, we mean software components embedded into the mote to run while processing the entire monitoring. According to section 3.4, these are the operating system and the application software. The operating system that we loaded onto the sensor nodes is the Contiki operating system and the application program is fixed-point Real-valued FFT, RFFT.

Contiki

Contiki is a lightweight, highly portable, small open source operating system primarily developed for networked embedded systems. Contiki is written with the C programming language and applications are also written using the same language. It is portable on a number of memory-constrained systems with processors ranging from 8-bit to high level microcontrollers. A typical full functioning system on a device with a graphical user interface, for instance, requires just about 30 KB of memory. It is highly flexible whereby memory efficiency can be increased by reconfiguring it to as low as tens of bytes with a code footprint on the order of kilobytes. It gives a built-in IPV6 compatible TCP/IP connected into the Internet to reap the benefit of the Internet of things.

A Contiki system is partitioned into the core and the loaded programs, see Figure 4.2 at compile time depending on the specific deployment.

\(^1\)http://www.hbm.com
4.2. SOFTWARE USED

**Embedded Fast Fourier algorithm**

Some embedded microprocessors may have an external unit capable of performing floating point arithmetic, but most low-end embedded systems like MSP430 ultra low power microcontrollers have no floating point arithmetic operation. Thus, MSP430s do not apply floating point rule on their program; hence, fixed point arithmetic operation is required to obtain proper results.

The main reasons why we use fixed point arithmetic operations in our implementation for emulating floating point calculations are:

- Limited memory of The microcontroller which has 8 KB of RAM and 92 KB of flash memory. The algorithm requires 2N 16-bit variables for FFT data, and our microcontroller can’t perform FFTs for values of N greater than 1024. Even this is difficult because other parts of the firmware will also require a few bytes of RAM. For our implementation, we therefore limit N to 512 due to overflow problem. Having small number of samples would have effect on frequency resolution; thus having large number of sample for FFT is recommended.

- Limited speed: low power microcontrollers have low speed for performing FFT which is significantly slower.

We used 16-bits variables in our implementation of fixed point FFT to represent the real and imaginary parts of every value. In doing so, for a 16 bit digit to represent a number:

- Bits 0 to 6 represent the fractional part
4.2. SOFTWARE USED

- Bits 7 to 14 represent the integer part
- Bit 15 holds the sign bit

This format has no effect on adding and subtracting but we must be careful when performing multiplication that would cause overflow. For the forward FFT (time to frequency domain), fixed scaling is performed to prevent arithmetic overflow while multiplying two 16 bit numbers. It’s important to store the intermediate product in 32 bit representation and perform shifting by 15 bit to get the 16 bit result. If we want the format of the output to be the same as the format of the input, we must restrict the range of the inputs to prevent overflow.

**Lookup Tables**

The twiddle factors calculations in FFT are one of the processes that consume too much processing time of the CPU. In each and every instance of their occurrence, the processor is required to compute the values of the trigonometric sine and cosine functions. The number of this computation is directly proportional to the size of the FFT. Therefore, by using a lookup table of these trigonometric functions, we can extremely reduce the extra computation time claimed. In doing so, we load pre-calculated values of those trigonometric functions into an array of memory to be called with array indexing. This is based on the fact that it is much economical and faster to load data from memory than do computations on the run.

One decisive thing we should note is that the range of sine and cosine functions lies in the interval \([-1,1]\). For any angle \(\theta\) the following condition should be fulfilled: \(|\sin(\theta)| \leq 1\) and similarly \(|\cos(\theta)| \leq 1\). Which means that the magnitude of most of the values are less than 1 and floating point. Hence proper scaling is important here as well. Keeping in mind the 16-bits microprocessor, we scaled the trigonometric table multiplying by \(2^{15} = 32768\) and to compensate for the truncation error introduced while representing the floating numbers with integers, we added 0.5 to each value. Of course, we need to scale back this result latter dividing by the same number somewhere in the computation.

**Peak Picking**

The peak picking is the next experiment that is done after the FFT. To do the peak picking, the frequency spectrum is needed to be calculated before hand. The standard way of doing the peak picking is by calculating the magnitude of the FFT output samples. This is done with the Euclidean norm as:

\[
|X| = \sqrt{|Re\{X\}|^2 + |Im\{X\}|^2}
\]  

(4.1)
4.2. SOFTWARE USED

But, to implement the standard square root function we need to include the math C library header in our code and this would add to the memory consumption and processing cycle. To avoid this loss, we approximate the magnitude simply as:

\[ |X| = |Re\{X\}| + |Im\{X\}| \]  
(4.2)

Although this is not 100% accurate, it better approximates the magnitude with so much save on the processing resources.

Set Threshold

The crucial part in applying a peak picking algorithm is to set the threshold so that a limit should be given to pick the peak. This is the lower threshold to discard the data below the limit and concentrate on the ones above the limit.

We used a standard deviation and mean based limit. Setting the lower limit just from the mean value would end up in losing lots of significant peak values at higher frequencies. Thus, we added the standard deviation into the arithmetic so that the limit should be set fairly in an appropriate way. In probability and statistics, the standard deviation, \( \sigma \), is calculated as:

\[ \sigma(X) = \sqrt{(X - \bar{X})^2} \]  
(4.3)

Where \( X \) is sample and \( \bar{X} \) is mean of the samples. It is a widely used stochastic index to show how much dispersion there is from the average (mean).

The equation that we used to set the lower limit, \( X_{\bar{X}} \), is:

\[ X_{\bar{X}} = \bar{X} + 2\sigma(X) \]  
(4.4)

This gives a peak picking limit with 95% confidence.

Sample Window Size

After setting the limit, care should be taken while setting the sample size to compare the peaks within. Fortunately, it was not a problem in the experiments we did, but there are cases where the mode frequencies are large or closely spaced. In those situations, setting the windows we would lose those closely spaced modes next to the dominant peak in the range. Observing the nature of the frequency spectrum would help to set a good approximation.
4.2. SOFTWARE USED

Figure 4.3: A frequency spectrum plot divided into non-overlapping windows.

In our experiments, we used a non-overlapping moving window size of 50 samples sizes.

How the Algorithm Works

Having set the lower limit, the algorithm picks a sample from the beginning of a window and compares it against the next two consecutive samples. If the sample is found to be the peak value, it sets it as a local maximum value and then moves to the next sample and does the same thing, but in this and the next cases, it does one more comparison with the immediately previously found local maximum value. If the new sample is found to have higher peak than the previous local maximum value, it is set to be the new local maximum value and the next comparison is done against this new value and it goes on like this with in the window and the whole process is repeated in the following windows up to the end.

Computing the Natural Frequency

After the peak values are found, the next step is to calculate the natural frequencies. From the theory of vibration analysis, the resonance natural frequencies are believed to occur at those peak amplitude values. Multiplying the instant sample number with the frequency resolution gives the natural frequency.

After all the above procedures are performed, the resonance frequencies are ready to be transmitted to the sink station where final modal analysis are done and the overall work is finalized.
4.2. SOFTWARE USED

4.2.2 Offline software tools

These software tools are signal processing tools that we used to analyze the data collected by the wireless and wired sensor systems and hence validate the algorithms implemented and the instruments used in the entire process. Matlab, a high level language with advanced signal processing toolkits, is used on Windows based computers. SciLab\(^2\), which is an open source free software, was used for Unix based machines. Catman\(^{TM}3\) a powerful data acquisition software for configuring, visualizing and analyzing measurements, was integrated with the wired measurement system. It has interactive user interface and provides a wealth of mathematical and graphical functions for analyzing and evaluating measurement data and data can be exported into all common standard formats (for example Excel or ASCII).

On the wireless side, most of the experiments were latter done by collecting data with Matlab through serial communication. We wrote a code so that four nodes are connected to our laptop through a USB hub. After the data are collected, they are post processed and plotted in a time and frequency domain subplots.

\(^2\)http://www.scilab.org
\(^3\)http://www.hbm.com
Chapter 5

Accelerometers

5.1 Introduction

Accelerometers are the major instruments used for vibration measurement and data collection in this thesis. We used different accelerometers for wireless and wired systems. While exploring which accelerometer to use, we went through different sets of choices and in this section, we will present a detailed walk through of the ways we chose the ones that we ended up working with. After a generic explanation of accelerometers, a tabular summary of the features of some of them that we encountered is given latter in the section.

5.2 What Accelerometers Are?

An accelerometer is an electromechanical device that is useful to measure static or dynamic acceleration forces in units of meters per second squared \( (m/s^2) \) or G-force (g). By static force, it is meant that the accelerometer measures constant force of gravity pulling. This helps, e.g., to find out the angle by which the device attached to the accelerometer is tilted at with respect to the earth. And by dynamic force, it means that it measures force caused by moving or vibrating the accelerometer as in SHM.

5.3 Selection Criteria

The following characteristics should be considered while selecting accelerometers \( [49] \) for SHM systems:
5.3. SELECTION CRITERIA

5.3.1 Analogue or digital

Analogue output accelerometers output a continuous voltage that is proportional to the sensed acceleration. As most microcontrollers come up with a built in analog to digital converter, these accelerometers are the easiest to interface with. Where as in microcontrollers with purely digital inputs a digital output accelerometer is preferable. Digital accelerometers usually incorporate a serial interface, SPI or I^2C to communicate with the microcontroller.

5.3.2 Sensitivity

It is the rate at which the signal reading changes with the acceleration. The higher the sensitivity of the accelerometer the better is the measurement as ambient vibrations can accurately be measured.

5.3.3 Full-Scale range

This determines the range of readings one does. The full-scale range of an accelerometer and its sensitivity are inversely proportional in that a smaller full-scale range means a more sensitive output. A ±2 g range accelerometer should suffice to measure vibration of a structure.

5.3.4 Number of axes measured

This is the number of possible axes (x, y, and z) out of which the accelerometer senses. 1- and 2-axis accelerometers are the most common ones. Nonetheless, if there is no other limitation that hinders from doing so, the 3-axis accelerometers are more preferable as they give a surplus of information against the 2- and 1- axis accelerometers. Besides, they come up with affordable prices with negligible difference.

5.3.5 Bandwidth response

This is the frequency range of vibration data that the accelerometer can measure.

5.3.6 Noise level

This is the lower limit of the measurement below which is considered to be unnecessary noise. It is the smallest amount of vibration an accelerometer can detect and is an important factor to properly measure the ambient vibration data. The lower is better as small vibrations are considered as noise if this floor is higher.
5.4. WHICH ONES DID WE USE?

5.3.7 Power consumption

Accelerometers in wireless sensor nodes should not consume too much power. It should be in the 100 µA range. Often, some accelerometers feature a sleep functionality to conserve energy when not needed.

5.3.8 Extra features

Modern accelerometers may include features like sleep control, tap sensing, free fall (0 g) detection and selectable measurement ranges. These can be used to trigger certain functionalities.

5.4 Which Ones Did We Use?

In the next sections we are going to give a detailed explanation of the accelerometers that we used.

5.4.1 Analogue devices ADXL345

The ADXL345 [4] is a digital accelerometer, Figure 3.5, found in Z1 Zolertia node. It is a 14-pin, ultra-low power, accelerometer with 10- to 13-bit resolution. Although it is a 3-axial accelerometer, we used a single axis measurement in the vertical direction as most of the properties of a structure can be identified in that dimension. With a 10000 g absolute shock rating, it is well suitable for SHM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Output Range</td>
<td>±2 - ±16</td>
<td>g peak</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>29 - 282</td>
<td>LSB/g</td>
</tr>
<tr>
<td>Bandwidth Response</td>
<td>0.1 to 3200</td>
<td>Hz</td>
</tr>
<tr>
<td>Dynamic Range</td>
<td>120</td>
<td>dB</td>
</tr>
<tr>
<td>Noise</td>
<td>0.75 to 1.1</td>
<td>LSB rms</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-40 to +85</td>
<td>°C</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>2 to 3.6</td>
<td>Volts DC</td>
</tr>
</tbody>
</table>

Several extra features are also provided. Activity and inactivity sensing features detect the presence or absence of motion by comparing user set thresholds with the acceleration in any axis. These features help the ADC to continuously sample input data and check the amount to determine if the whole system should be actively responding to the change in acceleration. Hence the
5.4. WHICH ONES DID WE USE?

system remains in sleep mode unless otherwise woke up by a rise of acceleration above the threshold saving a tremendous amount of power. On the other hand, tap sensing detects double taps and single taps in any direction while free fall sensing detects whether the device is falling. Some more important specifications are summarized in Table 5.1.

Data collection with ADXL345

ADXL345 provides two output pins for driving interrupts which can be enabled by setting the appropriate bit in the INT_ENABLE register. Different set of actions are activated with these interrupts like Single_Tap, Double_Tap, Free_Fall detection and advanced features such as Activity and Inactivity detection when they pass some threshold values set by the user. ADXL345 contains an integrated memory management system with a 32-level first in, first out (FIFO) buffer. This buffer has four modes: bypass, FIFO, stream, and trigger.

We first save memory space for buffer size of 512 samples to collect data. In order to obtain a synchronized data as possible from the wireless sensor nodes simple broadcasting is implemented on the nodes. The message sent is a simple 'hello' text to trigger other nearby sensors to simultaneously start sampling after receiving the broadcast message. In this case, one of the wireless sensor nodes act as an event triggering node. Then when the other sensor nodes receive the broadcast message, the appropriate bit register will be enabled with either of the pins mapped to the interrupts on the ADXL345 accelerometer.

In order to collect the X, Y, and Z-axes vibration data, set FIFO mode and data are stored in FIFO. When the number of samples in FIFO equals the level specified in the samples bits of the FIFO_CTL register, the WATERMARK interrupt is set. WATERMARK is an interrupt and a bit is set when the number of samples in FIFO equals the value stored in the samples bits (Register FIFO_CTL)[4]. The WATERMARK bit is cleared automatically when FIFO is read, and the content returns to a value below the values stored in the samples bits. The accelerometer does not stop working while FIFO continues collecting samples until it is full (32 samples from measurements of the X-, Y-, and Z-axes). WATERMARK interrupt still continues to occur until the number of samples in FIFO is less than the value stored in the samples bits of the FIFO_CTL register.

when the memory space of the buffer for the set sample size is full it shows warning by printing 'WARNING: FIFO OVERRUN'. This happens when new data replaces unread data and it gives indication of data collection is overrun. The actual overrun bit is set when FIFO is filled. The overrun bit is automatically cleared when the contents of FIFO are read. Hence, the registers will fill the buffer until it reaches the maximum buffer size and anymore data

42
5.5. VALIDATING THE ACCELEROMETER

will not be written to the accelerometer then finally it will print out the values collected by the accelerometer.

5.4.2 Si-Flex™ accelerometer

This is a relatively low-noise 3-axial analogue accelerometer that is used in the wired system. It is among a line of sensors produced by Si-Flex™ and the sensors are characterized by features such as wide dynamic range, excellent bandwidth, low distortion, high shock tolerance, and thermal stability.

Table 5.2: Summary of the Si-Flex™ accelerometer features.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Output Range</td>
<td>±3 g</td>
<td>peak</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>1.2 V/g</td>
<td></td>
</tr>
<tr>
<td>Bandwidth Response</td>
<td>DC to 1000 Hz</td>
<td></td>
</tr>
<tr>
<td>Dynamic Range</td>
<td>120 dB</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>300 to 500 ng rms / √ Hz</td>
<td></td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-40 to +85 °C</td>
<td></td>
</tr>
<tr>
<td>Input Voltage</td>
<td>6 to 15 Volts DC</td>
<td></td>
</tr>
</tbody>
</table>

Being the best accelerometer available, this accelerometer would have been the right one to be used with wireless systems. However, the fact that it needs much operating power input and its bulkiness are the factors that we didn’t consider it in our selection.

5.4.3 Other accelerometers

We also encountered other dozens of accelerometers and in Table 5.3 we give the overall summary of their characteristic features. Most of the accelerometers in the list are also among the ones used by most other wireless SHM system projects: SMB380 from Bosch Sensortec, LIS302DL from ST Microelectronics and MMA7361L. The LIS302DL accelerometer is the one similar to the accelerometer in iPhones and MMA7361L is the accelerometer in Senlilla node, the node that we were practicing with during the beginning of the thesis.

5.5 Validating the Accelerometer

The validation of the accelerometer selected is made by comparing the measurements against a well known accelerometer that measures accurate values. In our case, we did this by comparing the measurement from our wireless accelerometer against the wired one.
Table 5.3: Summary of the most common accelerometers in SHM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SMB380</th>
<th>LIS302DL</th>
<th>MMA7361L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Output Range</td>
<td>±2 - ±8 g</td>
<td>±2 - ±8 g</td>
<td>±1.5 - ±6 g</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>64 - 256 LSB/g</td>
<td>16.2 - 79.2 mg/digit</td>
<td>206 - 800 mv/g</td>
</tr>
<tr>
<td>Bandwidth Response</td>
<td>25 - 1500 Hz</td>
<td>20 - 4000 Hz</td>
<td>400 Hz</td>
</tr>
<tr>
<td>Noise</td>
<td>500 µg/√Hz</td>
<td>50 µg/√Hz</td>
<td>350 µg/√Hz</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>-40 to 85 °C</td>
<td>-40 to 85 °C</td>
<td>-40 to 85 °C</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>2.4 - 3.6 V</td>
<td>2.16 - 3.6 V</td>
<td>2.2 - 3.6 V</td>
</tr>
</tbody>
</table>

5.6 Cares to Be Taken

Cares should be taken while using an accelerometer so that the measurement should not go erroneous. Some of the most important factors that need be considered before using accelerometers are:

5.6.1 Calibration

Accelerometers measure the acceleration of a structure by detecting the rate of the relative change velocity. This relativity is taken by setting a reference level. Hence, setting reference is the first thing to do while using accelerometers. This is termed as calibration and in doing so accurate vibrations can be acquired. This is generally done by taking random measurements by setting a structure in its initial position. Those random measurements are averaged and are considered as 0 g from which the next measurements are contrasted within the range of the accelerometer. In case of measuring the vertical acceleration, the random average value is rather considered as 1 g as it is in the direction of the earth’s gravitation.

In our case, the calibration is done by measuring the ambient vibration of a structure and is considered as 1 g and is taken as the reference level of all coming measurements. Thus, this ambient measurement is averaged and the average is subtracted from any further measurements.

5.6.2 Attachment

The way the accelerometer is attached determines the quality and accuracy of the readings. The accelerometer should be attached as firmly as possible to the structure so that the whole integrated structure should be stiff enough. If the attachment is loose, this gives unnecessary extra acceleration due to the free movement of the accelerometer from the main structure.
5.6. CARES TO BE TAKEN

5.6.3 Axis alignment

The right axis orientation should be done in the driver code at the time of collecting data. For example, while the accelerometer is attached with the vertical orientation bound towards the Z-axis, assigning X-axis in the driver code to be measured will account for some unexpected results.

5.6.4 Sampling rate

In case of using a digital accelerometer, special care should be taken while selecting the sampling rate. As described in section 5.3.1, digital accelerometers are interfaced with the host device with serial interface. The quality of this connection affects the rate at which data can be retrieved. In the ADXL345 accelerometer case, using higher sampling rates would lead to improper data collection. Hence, using sampling rates of 100 Hz is what we did. Data collected by an accelerometer may vary depending on the sensitivity, resolution, sampling rate and range of the accelerometer. Thus literal comparison on the time domain is not the best option. However, no matter what the quality of the time domain data from different accelerometers vary, they all should measure the same frequency of the material.
Chapter 6

Experimental Setup

We used a simple steel beam, Figure 6.1, supported by wooden blocks from two ends to experiment our laboratory measurement works. The length of the beam was 3.5 m and the wooden supports, Figure 6.3, at the end points take a total of 18 cm thus the whole length excluding the wooden support becomes 3.32 m. This span was further divided into five subparts of each 66.4 cm long. By placing our sensor nodes on these subparts, we collected our measurements for realtime and offline analysis. A simple illustration of the steel beam is given in Figure 6.2 as a simple bar supported at both ends.

![Figure 6.1: Part of the 3.5 m long test beam](image)

The motivation behind the selection of this beam structure is that, it is very simple in form, easily available to do repeated experiments and move around. Since all we needed was a structure which can give a good output for a light excitation to verify our algorithm, this was the simplest and best available in the lab that can fulfill our requirements.
The steel beam is considered as simply supported beam structure thus we can theoretically determine the frequency, $\omega_n$, by a simple equation from the strength of materials [6].

$$\omega_n = (n\pi)^2 \sqrt{\frac{EI}{pAL^4}}$$  \hspace{0.5cm} (6.1)

Where:

- $n$, Mode number
- $A$, Cross-sectional area of the beam ($m^2$)
- $p$, Density ($kg/m^3$)
- $E$, Young’s modulus ($N/m^2$)
6.1 INSTRUMENTATION

- I, Area moment of inertia (m^4)
- L, Effective length of simply supported beam (m)

The second modal frequency is four times the first modal frequency.

6.1 Instrumentation

Both the wired and wireless sensor systems were used to instrument the beam. For the wired system, HBM MGC Plus data acquisition system (DAQ) was used.

This system encompasses, a Si-Flex™ MEMS sensor to be firmly attached to the beam with a heavy electromagnet attachment, a multichannel ADC and a Catman® DAQ software (see section 4.2.2) installed on a laptop computer.

Our ZI Zolertia wireless sensor nodes are placed at four points on the beam while the Si-Flex™, see Figure 6.5 accelerometer is placed at one fixed position on the beam and data is obtained on our laptop terminal through the mini-USB cable while the data from the Si-Flex™ accelerometer are acquired to the Catman® DAQ software.

We need to make sure not to excite above the ±1g range so that not to harm the instruments and the setup.
6.2 Experimental Procedures

In order to obtain as synchronized data as possible from the wireless sensor nodes in the lab test, we used simple broadcasting message on the sensor nodes. We used 4 wireless sensor nodes to collect vibration measurements from the simple beam on the experimental lab test. Additional sensor node is used to send a broadcast message to all the sensors to trigger the accelerometers to collect the vibration at the same time. When the sensor nodes receive the broadcast message, they automatically start sampling simultaneously for a sample size of 512 samples.

There exists an average time drift of 0.22s between measurements among the wireless sensors due to lack of applying synchronization protocol. This time
6.3. EXPERIMENTS

Drift is not constant and has effect on determining the correct mode shape of the structure because it is impossible to determine the relative phase difference between the sensor nodes and the imaginary components of the FFT data.

When a time synchronization algorithm is applied, the time-delay occurs is less than a few milliseconds [38].

6.3 Experiments

We divided the experiments into three sets as:

- Experiment 1: Sensor nodes are placed at 66.4 cm between each other. This is a setup where the sensors divide the beam into five equal subparts. In doing so, it is possible to capture more vibrations using the sensor array.
- Experiment 2: Two sensor nodes are placed at the two wooden supports. In this case, we can see if no or zero vibrations are measured at the supports as they are defined to be fixed from our definition.
- Experiment 3: The sensors are placed 66.4 cm apart from each other as in experiment 1 but here we measure ambient vibrations. This setup helps us see the sensitivity of the accelerometers to measure ambient vibrations in the absence of external impact.
Chapter 7

Results and Discussion

In this chapter, the results of the measurement setups are analyzed and presented. The validation of the accelerometers and the FFT algorithm, time domain and the frequency domains of both the wired and wireless systems are addressed and relevant discussions are made.

7.1 Validation

We divided the validation into categories:

- Validating the accelerometer
- Validating the FFT

7.1.1 Validating the accelerometer

The accelerometer validation was done at our table before we continue the final test in the lab. The validation was performed by comparing its measured frequency components against an iPhone Vibration measurement.

Vibration is a spectrum analyzer in iPhone using the built in accelerometers. It acquires and displays time series data, optionally removes DC bias, applies a Hamming window and performs an FFT on each channel to produce frequency spectra. The 3-channel accelerometer in the iPhone has a sensitivity of approximately 0.02 g and a range of ±2 g. Other features of the application includes:

- Variable start delay from 0 to 20 s
- Selectable data length from 128 to 1024 samples
- Adjustable sample rate from 10 Hz to 100 Hz
7.1. VALIDATION

The time domain and frequency plots obtained from the iPhone measurements are given in Figure 7.1.

Figure 7.1: Time and frequency domain data from iPhone

Figure 7.2 shows the same measurement done with our wireless sensor node. The frequency of the vibration from the iPhone, 19.34 Hz, is almost the same as the one from the wireless node, 19.26 Hz.

7.1.2 Validating the FFT

The second step is validating our FFT. After collecting a time domain data with the wireless sensors, we perform floating point FFT and compare the result with floating Matlab FFT. Figure 7.2 shows the floating point FFT result.

We used 512 samples to perform FFT. The total amount of samples that can be collected with Zolerita node is 1024 samples with data type 16 bit integer, but for performing FFT we have to first declare two variables with
7.1. VALIDATION

buffer size of 1024 for the real and imaginary components which would be 2048 samples and this, together with the program data, can not be supported with the 8KB Zolertia node flash memory. Thus, we used 512 data samples. Figure 7.3 compares the frequency spectrum computed with a 16 bit integer approximation FFT with the one obtained by the standard 32 bit floating point offline FFT with Matlab/Scilab.

Figure 7.3: Comparison between Floating point and Integer point FFT

As we can see from Figure 7.3 the integer FFT gives a good approximation of the floating FFT on top of saving memory space and reduce energy consumption.

We also implemented floating point FFT on the nodes and we put the comparison in Table 7.1.

Table 7.1: Comparison of floating point and fixed point FFT on the node

<table>
<thead>
<tr>
<th></th>
<th>Floating point FFT</th>
<th>Integer point FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of samples</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Size on memory</td>
<td>41 KB</td>
<td>24 KB</td>
</tr>
<tr>
<td>Running time</td>
<td>6 s</td>
<td>0.6 s</td>
</tr>
</tbody>
</table>

One big issue while using floating point FFT on the nodes is that due to the large memory size consumed during collecting the time domain data and because of the large number of cycles in the FFT, the node resets prematurely before printing out the FFT output values. Therefore the result obtained from the floating point FFT is not reliable.

Given the low resolution data from the ADXL35 accelerometer, the most important thing is noting the existence of the resonant peak frequencies. These points are what we want to know to go further to extract the important information to feature the behavior on the model bridge.
7.2 Time Domain Data Analysis

A time domain plot of the data from the wired and wireless systems is given in Figure 7.4.

As can be seen from the Figure, the time domain data plot from the wired system is relatively of better quality than the one from the wireless system. The data obtained from the wired and wireless look similar to some extent which indicates that both are measuring same vibration data from the bridge.

![Time domain data from wired and wireless sensors](image)

Figure 7.4: Time domain data from wired and wireless sensors

The close up look into those two measurements exposes some of the flaws in the wireless sensing system. At low amplitudes, they introduce noise due to the low resolution (10 bits per sample) of the ADC used in the wireless system in contrast with the 24-bits high resolution one used in the wired system. Here, the 14-bits difference in the fidelity of both data is considered as one of the trade-offs in using wireless sensor system.

The difference for the signal quality can be attributed to:

a) **Noise level**: The Si-Flex™ accelerometer has lower noise level (300 $ng_{rms}/\sqrt{Hz}$) than that of the ADXL345 accelerometer (<1.5 LSB rms). Hence less amount of noise is introduced into the measurements of the wired system.

b) **ADC resolution**: The MGC plus DAQ system has a 24-bit ADC which is of much resolution than the ADC found in the ADXL345 accelerometer (10 bits of resolution for a g-range of $\pm 2g$). The more the bit resolution of the ADC, the more the quantization level of the converter and eventually, this yields digital samples of more fidelity.

c) **Attachment**: The Si-Flex™ accelerometer is attached so firmly to the structure with an electromagnetic attachment which gives it’s system a great stiffness to freely vibrate with the whole structure giving a good signal quality.
7.3. FREQUENCY DOMAIN ANALYSIS

The data from the wireless sensor also has higher amplitude fringes beyond \( \pm 1 \text{g} \) at the beginning of the vibration as a result of imperfect attachment between the two sensors that make the wireless sensor resonate during the initial conditions. Using stronger attachments like magnetic ones can improve signal quality.

7.3 Frequency Domain Analysis

We made three experiments with different sensor placements and the results are put accordingly.

7.3.1 Experiment 1

In experiment 1, Figures 7.6 - 7.9, peak frequencies can be seen at sensors 1 to 4 as 12.76 Hz, 13.1 Hz, 13.29 Hz and 13.1 Hz respectively. Whereas in the wired system in Figure 7.5 the peak is at around 13 Hz. These peaks are the first modal frequencies of the model beam.

These peak frequencies are also the same with the ones that we found using our pick peaking algorithm.

7.3.2 Experiment 2

Placing the sensors just at the two end points at the wooden support would give measurements only at sensor positions 2 and 3 while sensors 1 and 4 give no frequency readings as they are placed right on the firm supports where no vibration is expected.
Since the wired sensor placement is not changed, giving the same excitation, the result obtained is similar to experiment 1 as shown in Figure 7.5. The wireless sensors, Figure 7.10, provide the expected.

### 7.3.3 Experiment 3

In this experiment, since the ambient vibrations were so small, a vibration can barely be detected on the wireless system.

The wired sensor and the wireless sensor positions are not changed as shown in Figure 6.2, but the input excitation here is ambient vibration just walking near the support of beam. The wireless sensors are not able to measure this small vibration and provide zero natural frequency, Figure 7.11. In the wired system; however, there exist very small vibration measurement, depicted in Figure 7.12, and the natural frequency of the beam is obtained even if the magnitude is 140 times smaller than the one obtained in experiment 1.
In general, the number of peak frequencies observed are similar on the lab experiments. However, the values of the frequencies at which these peaks are found show a slight difference from the wired measurement and among the experiments as well.

### 7.4 The Cause of the Frequency Differences

Before our final experiment, we had encountered high frequency measurement differences between wired and wireless sensors upto 5 Hz.
7.4. THE CAUSE OF THE FREQUENCY DIFFERENCES

Figure 7.10: Experiment 2 wireless sensor data

Figure 7.11: Experiment 3 wireless sensor data

Figure 7.12: Experiment 3 wired sensor data
Getting the same readings from both the wired and the wireless systems took too much of our thesis time. We tried to figure out the possible sources of the frequency differences between the readings from the two systems. As the development of FFT or PP algorithms depend on the similarity of the results of the measurements, we should make the readings similar enough to proceed with the rest of the thesis. The readings from the wired system were reliable as we verified them with the iPhone application called Vibration. The two readings match well. Thus we deduced that the problem is with the accelerometer on the WSN node.

We tried every possible way, from varying the placements and attachments of the node on the beam to changing axes, to find the problem with the accelerometer.

We divided the probably sources of the error into:

1. **Programming Bugs** - We used the ADXL345 driver code that came up with the Contiki OS. The algorithm uses I²C connection to stream two bytes of accelerometer readings from the registers.

2. **Circuitry** - The placement of the ADXL345 accelerometer in the PCB of the WSN node and the electronic components around it also alter the output of the readings of the accelerometer.

3. **Mishandling** - Extensive and high vibrations that result from inconvenient environment and mishandling of the equipment would as well alter the calibrated device. Since WSN nodes are such tiny devices, they often fall on floors and this might be one of the reasons that affected the readings.

After careful investigation of the problem behind the errors, we found out that it was a programming bug in the driver code that is included in the Contiki OS. Together with the SICS developers, we debugged and rewrote the code.

The problem is described as follows:

The driver code for ADXL345 in the old Contiki OS works in such a way that after the accelerometer function is initialized, it collects a preset amount of samples of two bytes and when the buffer is full, it collects the next set of samples without clearing the buffer. This causes some residual sample bits to accumulate and results in the difference in the frequency content of the overall sample.

Hence we debugged the codes so that the memory buffers are cleared before the next set of samples are collected. This was possible to do by using the advanced features of the ADXL345 accelerometer: FIFO and WATERMARK.
Chapter 8

Conclusion, Recommendations and Future Works

8.1 Conclusion

In this paper, we used Zolertia node as a WSN platform for performing output only modal analysis studies in laboratory test on a simple 3.5 m long beam structure for analyzing how data processing algorithms can be implemented with limitations of memory and power. This work would help us to know to what extent we can trade off memory limitations and energy efficient data processing algorithms on the implementation of less energy demanding SHM system. We developed our programs on a platform with Contiki OS and cheap wireless sensor node with built in ADXL345 accelerometer attached to it and carried out the tests.

We did a series of measurements where time history data both from wired and wireless system were collected and modal parameters were compared. We then implemented fixed point integer FFT on 512 samples followed by embedded peak picking algorithm. We evaluated this by off-line processing of the time domain data collected from the wired system and got similar results. Implementing fixed point FFT, we were able to manage the memory effectively by reducing the size of the program from 41 KB to 24 KB.

8.2 Recommendation

After all the works we have done, we give the following recommendations for whomsoever wants to continue the project and get a better result in less time and efficiency:

- Building a sensor board with a better accelerometer.
8.3 Future Works

This project mainly focused on data processing algorithms and SHM. The project can be extended to a more and advanced system so that it could encompass design and implementation of less energy consuming communication protocols. Some complex algorithms could also be implemented to perform damage detection as well. The experimentation and the test could also be taken to the real bridges in the field like the Södeström bridge. Implementation of a time synchronization would also increase the accuracy of the determination of the mode shapes.
Bibliography


BIBLIOGRAPHY


