LOCALIZED PIPELINE ENCROACHMENT DETECTOR SYSTEM USING SENSOR NETWORK

A Thesis

by

XIAOXI OU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2011

Major Subject: Electrical Engineering

Localized Pipeline Encroachment Detector System

Using Sensor Network

Copyright 2011 Xiaoxi Ou

LOCALIZED PIPELINE ENCROACHMENT DETECTOR SYSTEM

USING SENSOR NETWORK

A Thesis

by

XIAOXI OU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Jim X. Ji
	Mi Lu
Committee Members,	Andrew Chan
	Tiffani Williams
Head of Department,	Costas Georghiades

August 2011

Major Subject: Electrical Engineering

ABSTRACT

Localized Pipeline Encroachment Detector System Using Sensor Network.

(August 2011)

Xiaoxi Ou, B.S., Nanjing University of Aeronautics and Astronautics

Co-Chairs of Advisory Committee: Dr. Jim X. Ji
Dr. Mi Lu

Detection of encroachment on pipeline right-of-way is important for pipeline safety. An effective system can provide on-time warning while reducing the probability of false alarms. There are a number of industry and academic developments to tackle this problem. This thesis is the first to study the use of a wireless sensor network for pipeline right-of-way encroachment detection. In the proposed method, each sensor node in the network is responsible for detecting and transmitting vibration signals caused by encroachment activities to a base station (computer center). The base station monitors and analyzes the signals. If an encroachment activity is detected, the base station will send a warning signal. We describe such a platform with hardware configuration and software controls, and the results demonstrate that the platform is able to report our preliminary experiments in detecting digging activities by a tiller in the natural and automotive noise.

DEDICATION

To my parents for their love and support

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Ji, and co-chair, Dr. Lu, for their guidance and assistance throughout the course of this research. I would also like to thank my committee members, Dr. Chan and Dr. Williams, for their support throughout the course of my study.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my parents for their love and support.

NOMENCLATURE

WSN	Wireless Sensor Network
ROW	Right of Way
WIFI	Wireless Fidelity
PC	Personal Computer
LED	Light Emitting Diode
USB	Universal Serial Bus
IEEE	Institute of Electrical and Electronics Engineers
A/D	Analog to Digital
WPAN	Wireless Personal Area Network

TABLE OF CONTENTS

Page

ABSTRACT iii

DEDICATION iv

ACKNOWLEDGEMENTS v

NOMENCLATURE vi

TABLE OF CONTENTS vii

LIST OF FIGURES ix

LIST OF TABLES xi

1. INTRODUCTION 1

   1.1 Problem Statement 1
   1.2 Current Solutions 2
   1.3 Summary of Thesis Work 3

2. PROPOSED DETECTOR SYSTEM 5

   2.1 Wireless Sensor Network-imote2 8
      2.1.1 ITS400 Basic Sensor Board 9
      2.1.2 Imote2 Processor Board (IPR 2400) 11
      2.1.3 Software Environment 12
   2.2 WSN Algorithm Design 12
      2.2.1 Wireless Sensor Node Algorithm 14
      2.2.2 Gateway Function 19
   2.3 Base Station and Detection Algorithm 20

3. EXPERIMENTAL RESULTS 22

   3.1 Lab Test 23
   3.2 Field Test 30
      3.2.1 Experiment Setup 30
      3.2.2 Results 33
         3.2.2.1 Results on Grass Surface 33

Page

3.2.2.2 Results on Sand/Gravel Surface 39

4. DISCUSSIONS AND FUTURE WORK 48

5. CONCLUSIONS 50

REFERENCES 51

APPENDIX 54

VITA 59

LIST OF FIGURES

FIGURE Page

1 Pipeline Damaged by an Excavator 2

2 The Proposed System Configurations (top) System Using Phone Link to Send Warnings; (bottom) System Using WIFI to Send Warnings............ 5

3 Components Focused in Research.. 6

4 An Illustration of Wireless Sensor Network 7

5 Imote2 Sensor Node 8

6 Top and Bottom View of ITS400 Sensor Board 9

7 Top and Bottom View of IPR 2400 Processor Board 11

8 WSN Data Flow Design 13

9 Initial Design of Package Content... 15

10 Message Sent 3 Times 16

11 Final Version of Message/Package Content... 18

12 Received Messages 19

13 Base Station Algorithm Design.. 21

14 Procedure of Lab Test 23

15 200Hz Linear Interpolated Signals.. 25

16 Spectrum of 1Hz/0.5Hz Hitting Signals.. 26

17 Spectrum of Autocorrelation Signals 27

18 Spectrum for Different Number of Packages of 1Hz Signal 29

FIGURE Page

19 Two Types of Ground Surface:

Sand/Gravel Land (left) and Grassland (right)... 30

20 Two Equipments Used in Experiment:

Tiller (left) and Car (right) 32

21 Collection of Vehicle Data on Grass Surface... 34

22 Data Collected on Grass Surface.. 35

23 Mean Values of Every 2 Packages of Data on Grass Surface.................... 36

24 Spectrum of Data Collected on Grass Surface 37

25 All Data Collected on Grass Surface: 2 Package Mean 38

26 Collection of Tiller Signals on Sand/Gravel Surface 39

27 All Data Collected on Sand/Gravel Surface... 40

28 Spectrum of Data Collected on Sand/Gravel Surface 47

LIST OF TABLES

TABLE		Page
1	Magnitude Range Compared to Tiller Data at Critical Location	41
2	Data with Range Overlaps That of Tiller Data on Critical Location	42
3	Magnitude Distribution Compared to Tiller Data at 0.5m and 1.0m	43
4	Magnitude Distribution Compared to Tiller Data at 1.5m	44
5	Magnitude Distribution Compared to Tiller Data at 2.0m and 2.5m	44
6	Magnitude Distribution Compared to Tiller Data at 3.0m	45

1. INTRODUCTION

1.1 Problem Statement

Pipelines are normally buried in utility right-of-ways and are well maintained or monitored by machines or technique staff. Theoretically, it is quite unlikely that the pipelines are endangered. Nevertheless, they are sometimes damaged by construction equipment not owned by the pipeline company [1]. Referred to as third-party damage, it is a major cause of damage to natural gas and oil pipelines [2] which causes millions of dollars of loss each year.

As a matter of fact, of the many pipeline accident causes that occur to oil and gas pipelines, approximately 80% of all the accidents are caused by impacts (mechanical damage) to the pipeline. The pipeline industry indicates that clearly half of the accidents are caused due to incursion of excavating activities into the buried pipeline right-of-ways. To make things worse, the excavators usually do not notify the pipeline company or the One Call System of the intent to conduct the excavation activity. That is to say, the pipeline companies do not even receive a warning before their properties are impaired due to third-party damage.

In addition, the potential for mechanical damage to be inflicted to pipelines located in or near urban expansion areas has become a major concern for the pipeline owner/operators and government regulators.

This thesis follows the style of Computer Networks.

Figure 1 is a photo which shows how the pipeline looks after it is damaged by an excavator.

Figure 1: Pipeline Damaged by an Excavator [3]

1.2 Current Solutions

Right now there are several methods available for detecting or preventing third-party damages. These include acoustic approach, fiber-optic system, and aerial/satellite surveillance, among others.

The acoustic approach to pipeline intrusion monitoring is based on the detection of sounds from impacts against the pipeline. Such impacts include backhoe strikes [4]. According to a Battelle chronology of developments [5], British Gas first used the pipe wall as an acoustic signal carrier with a detector on the pipe wall. The acoustic way can detect other significant conditions such as product theft but it is susceptible to confusion from benign acoustic sources such as valve closures and routine maintenance operations.

Another patented solution is the Fiber-Optic System. The principle of its operation is that optical fibers are sensitive to stress applied to the fiber and changes in the fiber's light transmission may be detected and located. The major work in fiber-optic detection has been carried out by the Gas Research Institute [6]. This method has the ability of detecting and locating encroachment at different locations along the pipeline. However, sophisticated instrumentation and signal processing are required and installation and maintenance of such systems are expensive.

Commercial satellites can now monitor pipeline rights-of-way for encroachment detection. For example, synthetic aperture radar (SAR) can be used to provide RADARSAT images [7] that can be processed to reveal the presence of trucks or earth-moving equipment in proximity to the pipelines [4]. The remote sensing technique can cover the rights-of-way of an entire pipeline quickly and efficiently. However, this method does not have real time ability. Other security surveillance technologies can also be applied but they often share part of some of the disadvantages of the aforementioned methods, for example, cost effectiveness, ease of maintenance, and/or high probability of false alarms.

1.3 Summary of Thesis Work

In this research, we aim to develop a pipeline encroachment detector system that poses the following features:

1. Cost effectiveness

2. Ease of installation and maintenance

3. Real time processing

4. Accurate locating ability

The operating principle of the proposed system is detection and analysis of vibration/acceleration data from distributed wireless sensor readings. The signals are analyzed in a base station (field computer) that is powered by a battery pack or other power source. If a potential encroachment activity is detected, the system can report it to a pipeline company designated operating center. There are two ways to achieve this: calling a staff phone to notify the impact, or sending an email to a remote computer center if WIFI is available at this location.

In order to develop such a system, the work is generally comprised of following subtasks:

1. Build system software for WSN to obtain and transmit acceleration data.

2. Develop pattern recognition algorithms to detect encroachment activities.

3. Perform experiments to obtain real data to test the developed system.

In this thesis, we take advantage of the recent advances in sensor network technology and develop platform for pipeline encroachment system from commercially available wireless sensors. Due to the time limit, the work focuses on WSN implementation and signal analysis. Preliminary results in lab tests and a field test are presented. To our best knowledge, the thesis is the first to apply wireless sensor network (WSN) to pipeline safety. The method is low cost, man portable and can be deployed by a pipeline field maintenance employee.

2. PROPOSED DETECTOR SYSTEM

Figure 2 depicts the system design. The dashed line arrows indicate wireless connection.

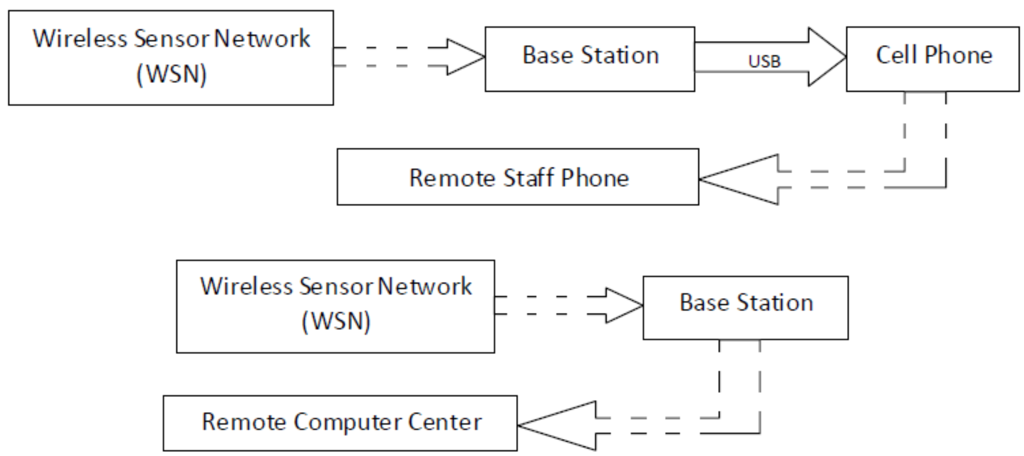


Figure 2: The proposed system configurations (top) system using phone link to send warnings; (bottom) system using WIFI to send warnings

The base station receives and analyzes the incoming data while gives an alarm when encroachment activity is detected. Besides, the base station also points out the accident location according to different signals obtained from different sensors.

To report encroachment activities to a pipeline designated operating center, there are two ways: calling a staff phone to notify the impact, or sending an email to a remote computer center if WIFI is available at this location. In either case we would need battery pack to maintain the life of a base station. The prototype platform consists of hardware components and software programs.

In this project, work is focused on the wireless sensor network and encroachment signal detection, as is shown in Figure 3.

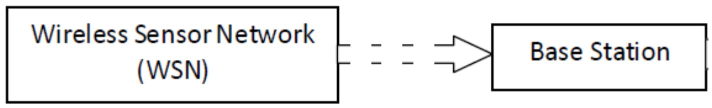


Figure 3: Components Focused in Research

Building prototypes for wireless applications used to be a time-consuming work. Thanks to technology advances and commercialization, a variety of well-developed devices can now be purchased at a reasonable price [8]. For instance, based on our requirements and constraints, a type of wireless node, also known as “mote”, which itself includes a processor, power supply and radio/antenna, can be employed in this project. Those motes are often supported by ready-to-use operation systems, which can further facilitate the software development. On top of the hardware, we also need software interfaces and applications so that user-defined taskscan be accomplished.

Several sensor/mote brands are commercially available, each with its special configuration. Intel Mote2, also known as Imote2 [9], stands out in its category because of its excellent computational capability, wide variety of I/O ports and OS supports. Since introduced in 2007, Imote2 has quickly drawn attentions from various application fields, such as structural health monitoring [10], volcano monitoring [11], and human behavior interpretation [12].

Wireless sensor system consists of two types of sensors:

1. Sensor node/sender

2. Gateway sensor/receiver

Each node in the WSN contains a sensor (accelerometer) to detect the vibration signals caused by encroachment activities. The sensor network system has multiple point sensors which use wireless network technology to communicate through a radio link to report to the receiving sensor/gateway sensor. The gateway sensor then relays the collected packages to base station. The entire wireless sensor network is illustrated in Figure 4.

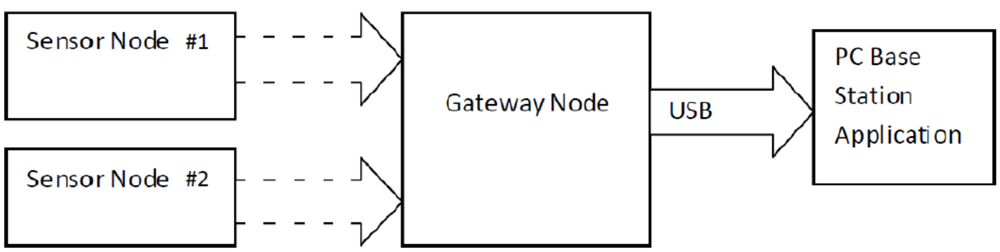


Figure 4: An illustration of Wireless Sensor Network

Senders run identical program implemented in C# to acquire acceleration data. The acceleration data is then packed and transmitted to the gateway node through wireless communication. The receiver node then relays the incoming packages to the PC base station (computer center) via a USB cable. The PC base station completes the rest of the entire work by processing the result.

2.1 Wireless Sensor Network—imote2

Imote2.BuilderKit is designed to be a complete development environment for high performance wireless sensor networking (WSN) applications leveraging the Microsoft .NET Framework. Code on the “mote tier” uses the .NET Micro Framework [13]. Each sensor node contains three parts:

1. Sensor board
2. Processor board
3. Battery board

These boards are plugged together to function as an independent wireless sensor node. Figure 5 shows how it looks like and compares its size with a quarter.

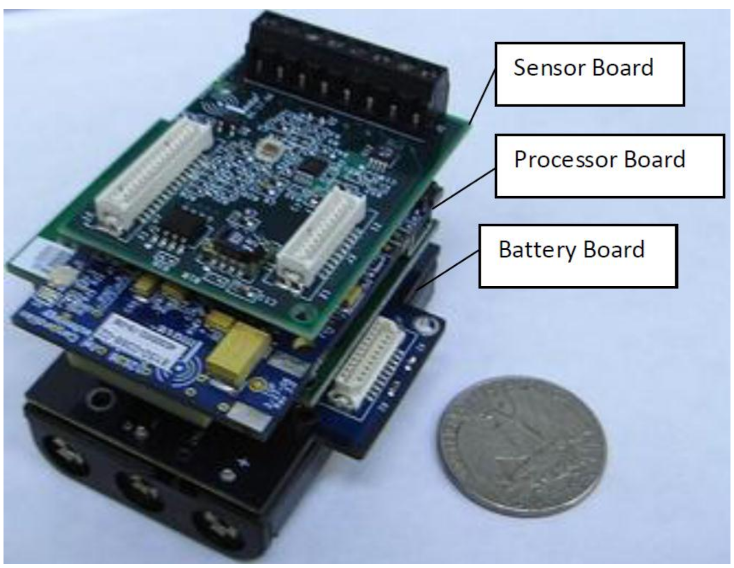


Figure 5: Imote2 Sensor Node

The gateway sensor is just the processor board loaded with a different program. It is connected to PC base station using USB cable and relays data packages to base station. It can either go with a battery board or using power from PC.

2.1.1 ITS400 Basic Sensor Board

The basic sensor board is designed to connect to the basic connectors on the Imote2. It contains a 3d Accelerometer, advanced temp/humidity sensor, light sensor and 4-channel A/D. It is a pass through board to allow stacking with another sensor/communication board. The top and bottom view of ITS400 sensor board is shown in Figure 6.

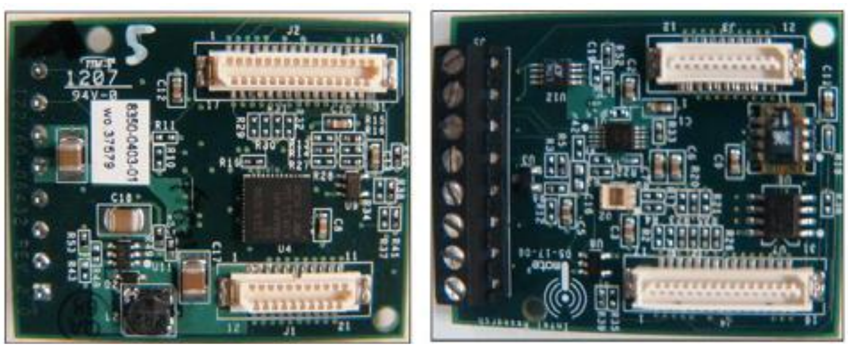


Figure 6: Top and Bottom View of ITS400 Sensor Board

The ITS400 sensor board is multi-sensor board that combines a popular set of sensors for wireless sensor network applications, including:

- ST Micro LIS3LV02DQ 3d 12 bit ±2g accelerometer
- High Accuracy, ±0.3 ℃ Sensirion SHT15 temperature/humidity sensor

• TAOS TSL2651Light Sensor

• Maxim MAX1363 4 Channel General Purpose A/D for quick prototyping

• TI Tmp175 Digital Temperature Sensor with two-wire interface

In this project, only the accelerometers are to be employed to collect ground vibration data. The LIS3LV02DQ is a three axes digital output linear accelerometer that includes a sensing element and an IC interface able to take the information from the sensing element and to provide the measured acceleration signals to the external world through an I2C/SPI serial interface [14].

The accelerometer has 4 options for data rate selection: 40Hz, 160Hz, 640Hz, and 2560Hz [14]. Since the accelerometer is part of the sensor board, there is no way to alter this parameter as an application developer: it has been pre-selected by Crossbow as the default 40Hz data rate.

The LIS3LV02DQ accelerometer has a 2-bit control register called DF1, DF0. It allows selecting the data rate at which acceleration samples are produced. The default value is 00 which corresponds to a data-rate of 40Hz. By changing the content of DF1, DF0 to “01”, “10” and “11”, the selected data-rate will be set respectively equal to 160Hz, 640Hz and to 2560Hz [14]. Therefore, in order to have a higher sampling frequency in the future, we should either purchase a new sensor board with a faster accelerometer or alter the register value in the LIS3LV02DQ accelerometer.

2.1.2 Imote2 Processor Board (IPR 2400)

Figure 7 depicts the top and bottom view of IPR 2400 processor board.

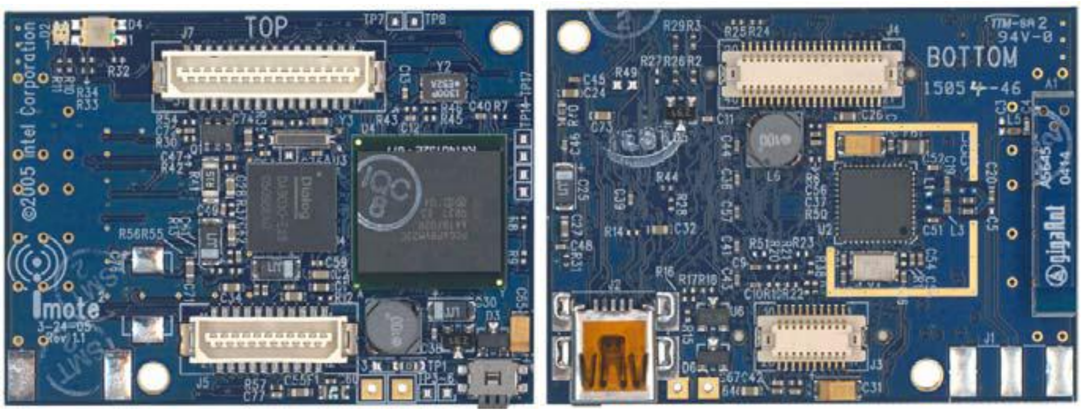


Figure 7: Top and Bottom View of IPR 2400 Processor Board

The Crossbow Imote2 is an advanced sensor network node platform designed for demanding wireless sensor network applications requiring high CPU/DSP and wireless link performance and reliability. The platform is built around Intel's XScale processor: PXA271. It integrates an 802.15.4 radio (TI CC2420) with an on-board antenna. The Imote2 is a modular stackable platform and can be stacked with sensor boards to customize the system to a specific application, along with a "battery board" to supply power to the system. The Imote2 integrates an 802.15.4 radio transceiver from ChipCon (CC2420). 802.15.4 is an IEEE standard describing the physical & MAC layers of a low power low range radio, aimed at control and monitoring applications. The CC2420 supports a 250 kb/s data rate with 16 channels in the 2.4 GHz band [9]. Its features are summarized as below:

- PXA271 XScale® processor @ [13–416] MHz
- Wireless MMX coprocessor
- Integrated 802.15.4 radio, support for external radios through SDIO and UART
- Integrated 2.4GHz antenna

2.1.3 Software Environment

For the development environment requires:

- Microsoft Visual Studio 2005
- Microsoft .NET Micro 2.0

All development in this project is done in C# and requires Visual Studio to be installed. The Microsoft .NET Micro Framework provides extensions for doing development on small embedded platforms such as the Imote2. In our experience, this combination must be running under Windows 2003 with SP1. Under other operating systems, the C# project files cannot be opened and executed.

2.2 WSN Algorithm Design

The WSN algorithm consists of two parts:

1. Sensor node algorithm
2. Gateway algorithm

The data flow and system chart is depicted in Figure 8.

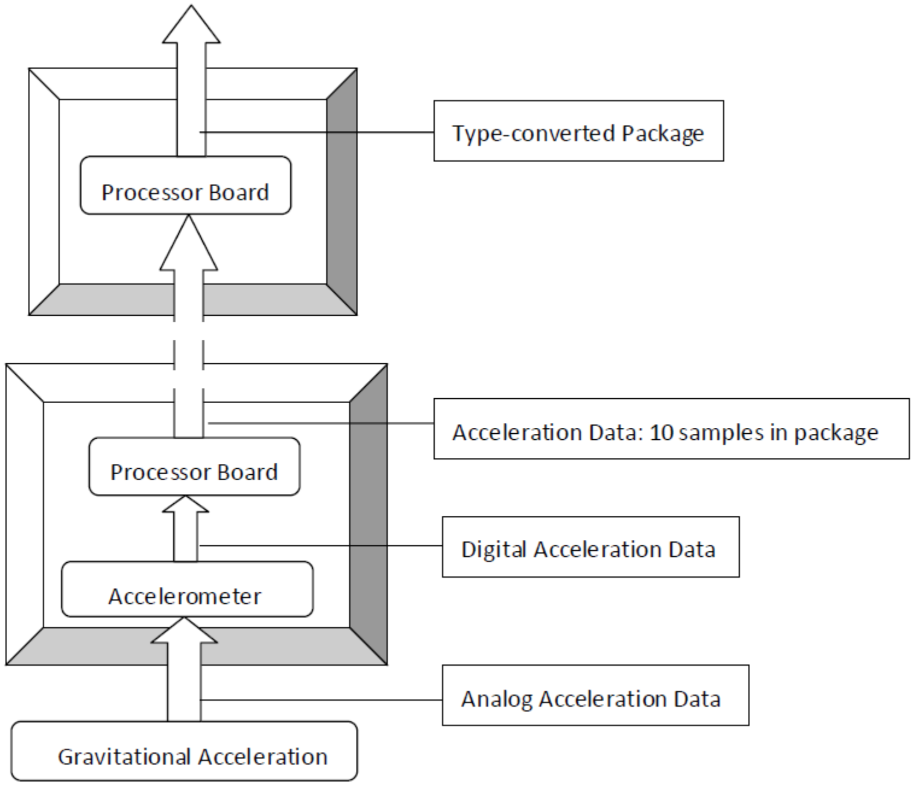


Figure 8: WSN Data Flow Design

The two framed blocks represent sender and receiver respectively. The upper framed block is the gateway sensor and the one below represents sensor node. Starting from the bottom of the chart, analog gravitational acceleration data becomes digital at the output of accelerometers; the processor board then packs a message of 10 readings and sends it out to gateway sensor via wireless communication; the gateway sensor converts the message format then relays it to PC base station. The entire WSN algorithm will be illustrated in details in the following two sections.

2.2.1 Wireless Sensor Node Algorithm

The job of the sensor node is to obtain local gravitational acceleration from embedded accelerometers and sends them out in a package with proper headers. Ideally the gravitational acceleration is a constant in the same area. But working equipment on the ground may alter its value and the change depends on both the equipment type and distance. In this project, horizontal acceleration data is not our concern and only the vertical direction is of interest.

Since the accelerometers have a fixed A/D precision of 12 bits, the unsigned short data type (two bytes) would be enough to store the value. Each package consists of three parts: node ID, time stamp and accelerations readings.

Node ID is necessary in a multi-sensor system for base station to know where the present package is sent from. The user should keep a table of the identity number of each sensor node and the corresponding location where this sensor is placed. In this way, the base station would know where a package is sent from and thus the exact location is obtained by looking up the table. Time stamp is also important in real time plotting implemented by base station application and is also useful as a debugging tool.

As is mentioned in the previous section, the IPR 2400 processor board integrates the IEEE standard 802.15.4. This standard intends to offer the fundamental lower network layers of a type of wireless personal area network (WPAN) which focuses on low-cost, low-speed ubiquitous communication between devices (in contrast with other, more end user-oriented approaches, such as WiFi). The emphasis is on very low cost communication of nearby devices with little to no underlying infrastructure, intending to exploit this to lower power consumption even more. The basic framework conceives a 10-meter communications range with a transfer rate of 250 kbit/s. Lower transfer rates of 20 and 40 kbit/s were initially defined, with the 100 kbit/s rate being added in the current revision [15].

In order to employ this short-distance and low-speed communication channel to the greatest extent, a test is to be carried on to clarify channel capacity regarding maximal message size.

Originally, the package is designed to contain 10 samples of three-dimensional acceleration data or 30 samples of Z-axis/vertical data. Thus each package has a length of 64 bytes, which is shown in Figure 9 as the initial design:

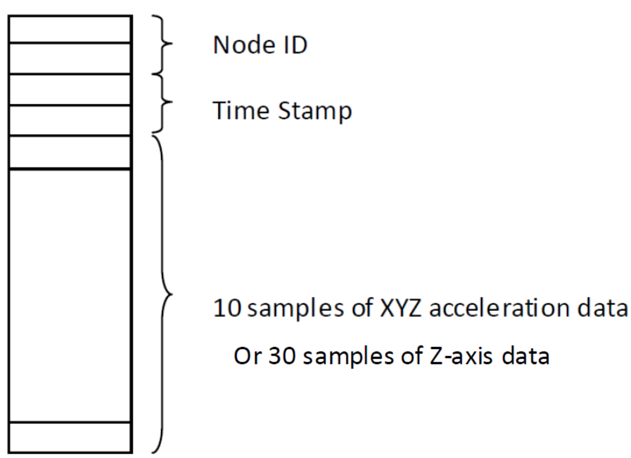


Figure 9: Initial Design of Package Content

The algorithm of the sensor node program is developed as follows:

1. Allocate 64 bytes of memory buffer for message.

2. Pack node ID into the message buffer.
3. Enter a *for* loop and set on board LED color to be purple indicating the start of reading.
4. In the *for* loop, 30 samples of acceleration data are obtained, which occupies 60 bytes
5. Pack time stamp into buffer.
6. Send out the package through wireless communication.
7. Set the on board LED color to be yellow indicating end of work.
8. End of *for* loop, which is also end of *main* function.

The program coding style is similar to any type of embedded code: work is done repeatedly in an infinite loop. In short, the processor board is used to accomplish on-board data aggregation. The test of sending 64-byte-long message reveals that the message has to be sent three times. Figure 10 shows that the message is not separated as three equal parts in the sending process. The first part has the most information.

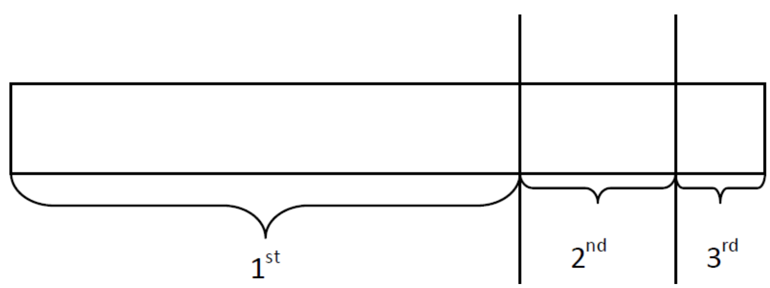


Figure 10: Message Sent 3 Times

In a one-sensor system, this underlying mechanism is fine for message of any length. However, in a multi-sensor system, the second and third pieces would otherwise be discarded as they have no header that indicates where they are from. This means the base station can only accept the first piece of each message if package length exceeds some upper limit.

To discover this upper limit, numerous messages are examined in the hope of finding out a general rule. As a result, the first truncation point is not a specific number of bytes. Instead, it is enforced by number of characters. In other words, the number of characters that can be sent at once has an upper limit: 111. That is to say, we cannot pack more than 111 characters in a package, including the space sign.

In the worst case, each byte must be represented by three characters (100~255) plus one space character, the total number of bytes that can be included in a package is (111-3)/4 +1 = 28. The 3 indicates last byte does not require extra space to separate it from the next byte and 4 here means each byte requires 4 character positions, one of them is the space.

In this design, all data is two bytes long so in the worst case, 14 unsigned short type numbers can be sent at one time. By this standard, each package can hold at most 12 acceleration data plus one node ID and the time stamp. In order to make calculation easier, it contains 10 samples of data. The redundant 4 bytes can be used in future development to store new data. This designed is clearly showed in Figure 11.

Node ID	Time Stamp	10 samples of Z-axis data	trailing 0s

Figure 11: Final version of message/package content

Since the readings are codes in two's complement, a small negative number (its higher byte is 255) and a very large number take the most space. Gravitational acceleration is a large number and on rare occasions an activity could make it negative. The goal of our project is to detect anomalies so we must have enough space for a stream of numbers which rarely shows up in normal conditions. In future development where horizontal acceleration is involved, this bandwidth sacrifice is compulsory.

The sensor node algorithm can achieve a data rate as high as 300Hz by repeatedly accessing the accelerometer. However, the mechanical characteristic of LIS3LV02DQ accelerometer embedded in the sensor node has a bandwidth limit of 40Hz. Therefore, any sampling frequency higher than 40Hz is a waste of power and cause a "staircase-like" effect in the readings. In real time testing, the actual sampling rate is between 39.52Hz and 39.84Hz.

A small program is written to record the content of each package in a local text file and Figure 12 shows the bandwidth constraint pattern.

```
216 2 101 2 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 158 2 166 5 166 5 166 5 166 5 166 5 166 5 166 5 166 5 166 5 166 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 177 2 166 5 166 5 166 5 166 5 222 5 222 2 222 2 222 2 222 2 222 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 233 2 181 251 181 251 181 251 181 251 181 251 181 251 181 251 181 251 181 251 181 251 0 0 0 0 0 0 0 0 0 0
216 2 252 2 181 251 181 251 181 251 0 248 0 248 0 248 0 248 0 248 0 248 0 248 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 53 3 0 248 0 248 0 248 0 248 0 248 0 248 0 248 0 248 0 248 0 248 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 72 3 0 248 205 249 205 249 205 249 205 249 205 249 205 249 205 249 205 249 205 249 0 0 0 0 0 0 0 0 0 0 0
216 2 129 3 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 148 3 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 205 3 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 255 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216 2 224 3 70 5 70 5 70 5 70 5 70 5 70 5 70 5 70 5 70 5 70 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 12: Received Messages

The figure above is a screen shot of packages. Each row is a message with node ID, time stamp and 10 samples of vertical acceleration data. All messages without a valid sensor node ID as header are discarded. The fourth, fifth and seventh line take a lot of space since there are negative readings. Other lines have more zeros padding due to the fact that the bandwidth is not fully used.

2.2.2 Gateway Function

The gateway sensor is just an IPR 2400 processor board running a different program. It has two functions:

1. Receive wireless data packages.
2. Send the package to PC base station

The first function means the program should always listen to incoming packages and receiver them at proper time.

The second function is trickier since the gateway sensor cannot simply dumps what it receives to computer center. The PC only recognizes *mica2* message format defined by Crossbow but the sensor node can only pack a byte array. Therefore, the program must perform format conversion before it throws the package to PC.

This part of work is handled in software packages provided by Crossbow and can be simply implemented by an API called *ConvertToMica2Msg()*. The software package provided by Crossbow defines APIs to facilitate the work of application developers. Although the Crossbow software package expedites high-level application developing, it provides few interfaces for developers to alter hardware configuration such as altering accelerometer's sampling frequency.

2.3 Base Station and Detection Algorithm

The key role of base station application is to detect excavation activities using pattern recognition algorithm. The decision is based on the detection of unusual vertical vibration data. The gravitational acceleration at a specific location is a constant number most of the time. But this value may be modified when there is a piece of working equipment working nearby. There is a possibility that the change in local gravitational acceleration may reveal certain properties of the activity. In order to identify excavation activities, the base station algorithm use certain data features that can help recognize excavation activities.

The features can be any type of signal properties: magnitude, mean values, standard deviation, frequency (spectrum), etc. The flow chart of a simple algorithm based on signal magnitude is shown in Figure 13.

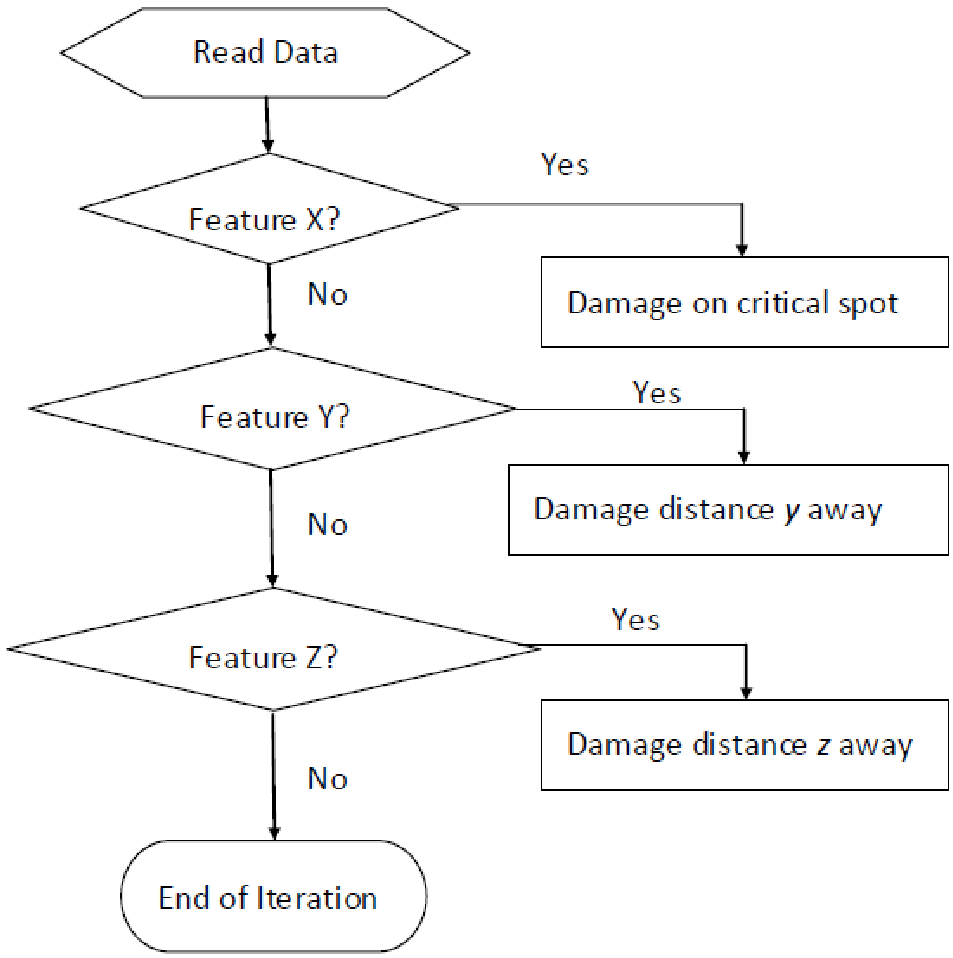


Figure 13: Base station algorithm design

The algorithm should not only detect damage right above the critical spot (with pipelines buried underneath), but also tell how far away the activity is from the critical spot. In either case the base station must notify the pipeline company for safety concerns.

The base station on board processing ability involves feature matching to check if the present situation meets any of the conditions that require the pipeline company's attention. To achieve this, features be obtained beforehand through experiments.

The so-called "feature" in this thesis refers to any signal anomalies caused by excavation activities that can be interpreted mathematically in base station application. If the anomaly is interpreted in terms of mean value or standard deviation, the base station would calculate these parameters for received packages and compare them with empirical data. If the anomaly is interpreted in the frequency domain, for example, spikes at certain frequency points, the base station would work out the spectrum of received package using Fourier Transform. The obtained spectrum is then examined to see if it has dominant frequency components that fit the description of an "anomaly".

To extract a feature, two types of signals are needed: background signal used as a reference and signal resulting from digging activities. To prevent false alarms, the background signal requires two facets of data: natural ground vibration without disturbance and signals collected while non-harmful situations are present. In reality, any significant activities can cause disturbance and make ground vibration different from natural values. Our work is to decide if the disturbance is from a benign activity, such as a vehicle driving nearby or from harmful excavation equipment. Therefore, the "features" we are looking for should belong to excavation activities only. For example, if a feature can be observed in the presence of both cases, it cannot by applied in the base station.

3. EXPERIMENTAL RESULTS

3.1 Lab Test

Before the sensor network is sent to collect real data, some tests are to be performed in an indoor lab setting to ensure that the system is able to pick up signal features. In the test, the sensor is placed on a desk and a person taps the desk at a certain rate of 0.5Hz or 1Hz. The post-processing is described in Figure 14.

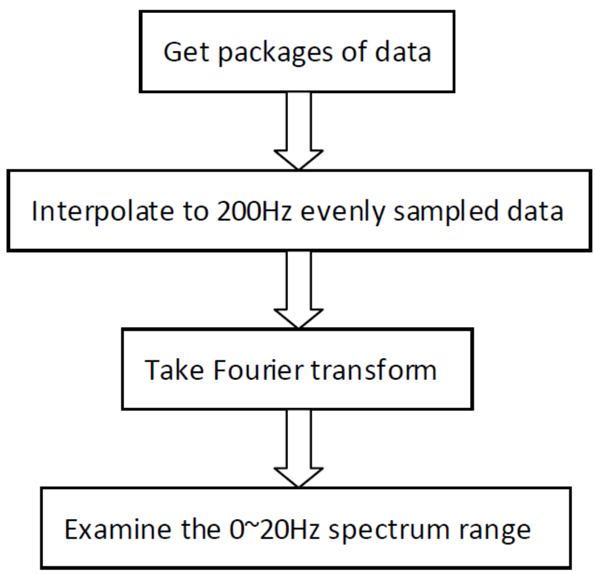


Figure 14: Procedure of Lab Test

According the Nyquist sampling theorem, spectrum range beyond half of the sampling frequency is aliased. As is mentioned before, the wireless sensor node has a 40Hz sampling rate constraint, so our focus is in the 0~20Hz range. The 200Hz interpolation is employed simply for the purpose of obtaining finer spectrum. In the ideal case, the 0~20Hz spectrum should have a clear 1Hz/0.5Hz component and harmonics. The Fourier transform in this project is called Discrete Fourier Transform (DFT), which decomposes a sequence of values into components of discrete frequencies. It can be described in the following formula.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \dots, N-1.$$

The formula takes an input time domain signal x_n with length N and calculates its corresponding N-point DFT. In the DFT sequence, first value X_0 represents the 0Hz component and the last value X_{N-1} corresponds to half the sampling frequency Fs/2. In this way, it can be deduced that X_k is the $Fs*k/2(N-1)$ frequency point. Figure 15 displays time domain signals for 1Hz and 0.5Hz desk hitting respectively.

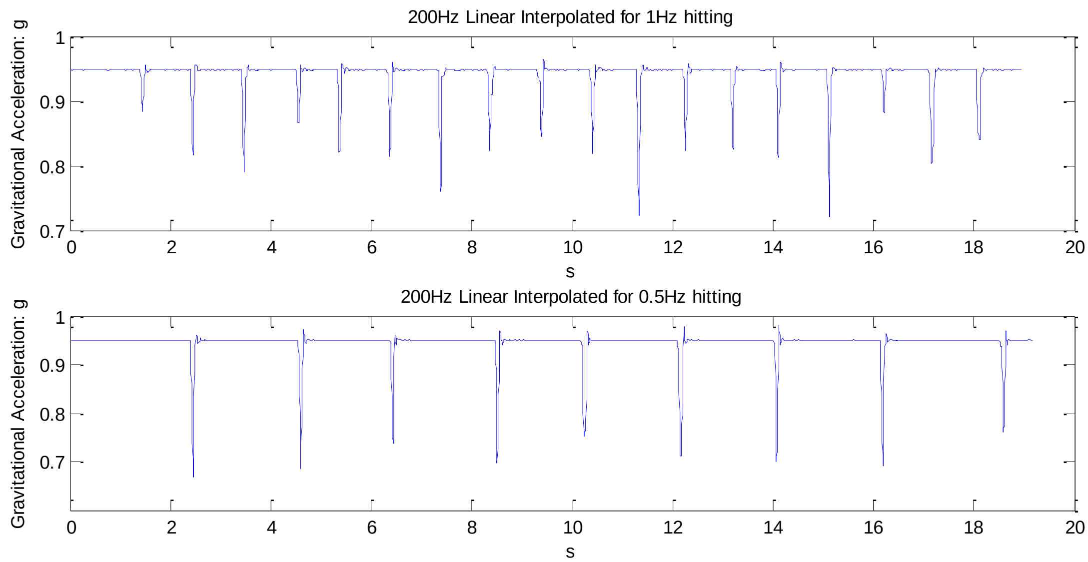


Figure 15: 200Hz Linear Interpolated Signals

In Figure 15, the horizontal axis is time and both signals have duration of 19 seconds. The vertical axis is the multiple of the standard gravitational acceleration converted from raw sensor readings. In real cases, objects with low densities do not accelerate as rapidly as the standard acceleration indicates due to buoyancy and air resistance [16].

Figure 16 shows the 0~20Hz spectrum for the signals depicted in Figure 15.

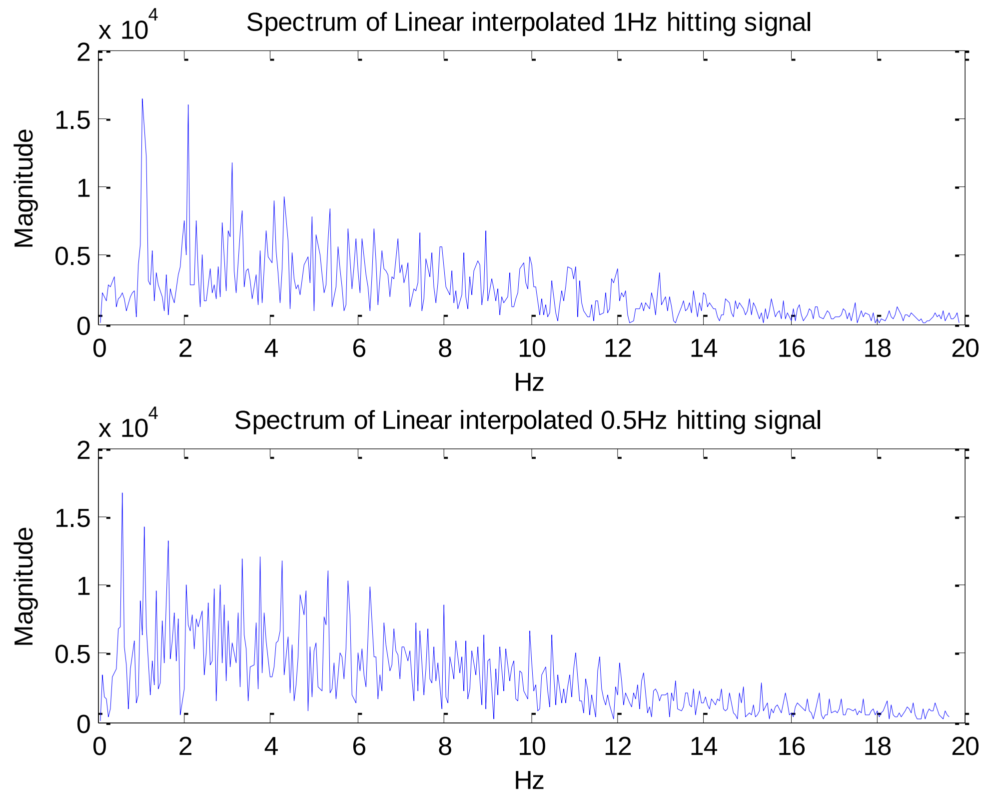
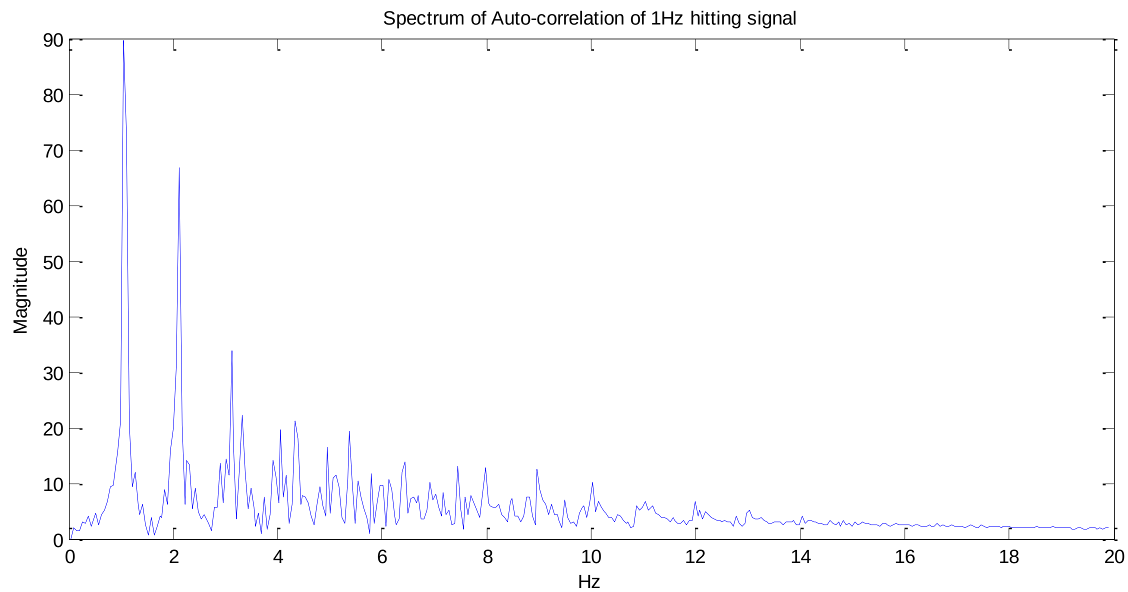


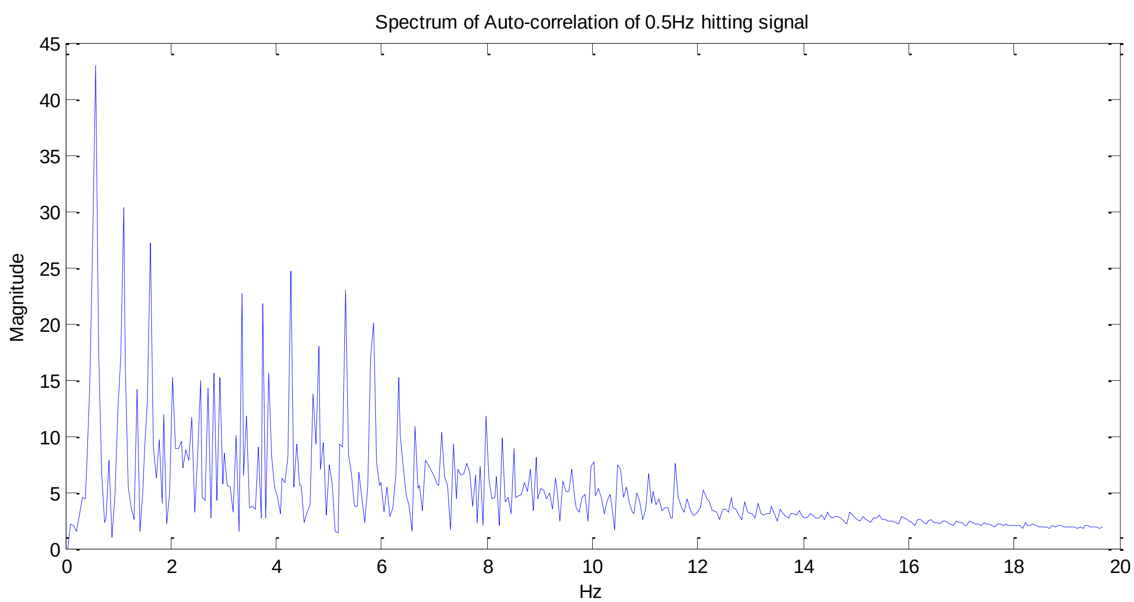
Figure 16: Spectrum of 1Hz/0.5Hz Hitting Signals

In Figure 16, spectra of 1Hz/0.5Hz hitting signals are illustrated. Since the original signals are not precisely sampled, they are processed using linear interpolation to make sure the spectrum is obtained from evenly-sampled time domain signals. In Figure 16, it is clear that the base frequency component and other harmonics are well kept in the spectrum.

In order to display more periodicity information, Figure 17 is obtained from taking the Fourier transform of autocorrelation.



(a) Spectrum of 1Hz Auto-correlation



(b) Spectrum of 0.5Hz Auto-correlation

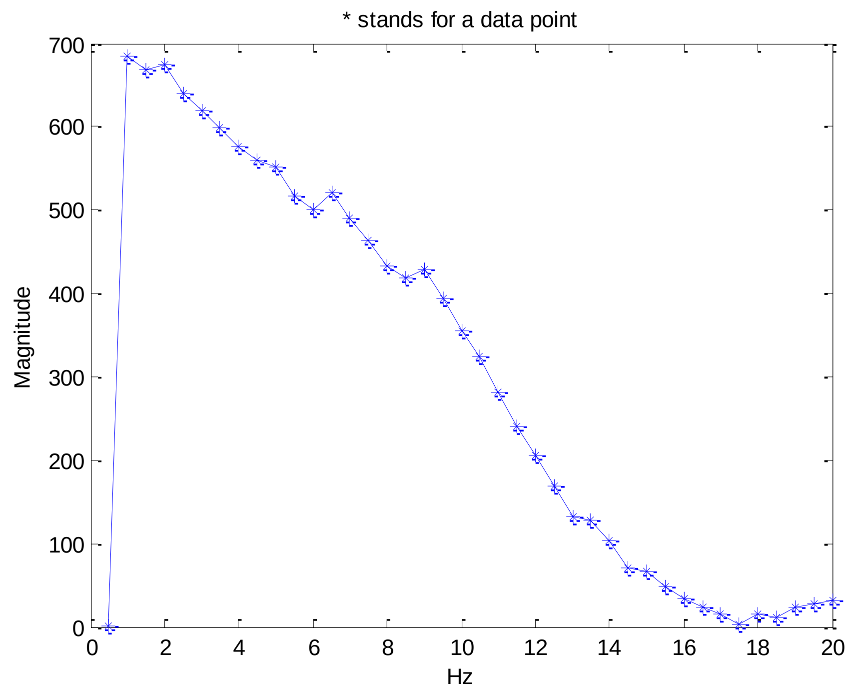
Figure 17: Spectrum of Autocorrelation Signals

In the 1Hz case, the 1Hz spike is clear and dominant. The 2Hz and 3Hz harmonics can also be observed. In the 0.5Hz case, more harmonics can be detected as well as the 0.5Hz spike.

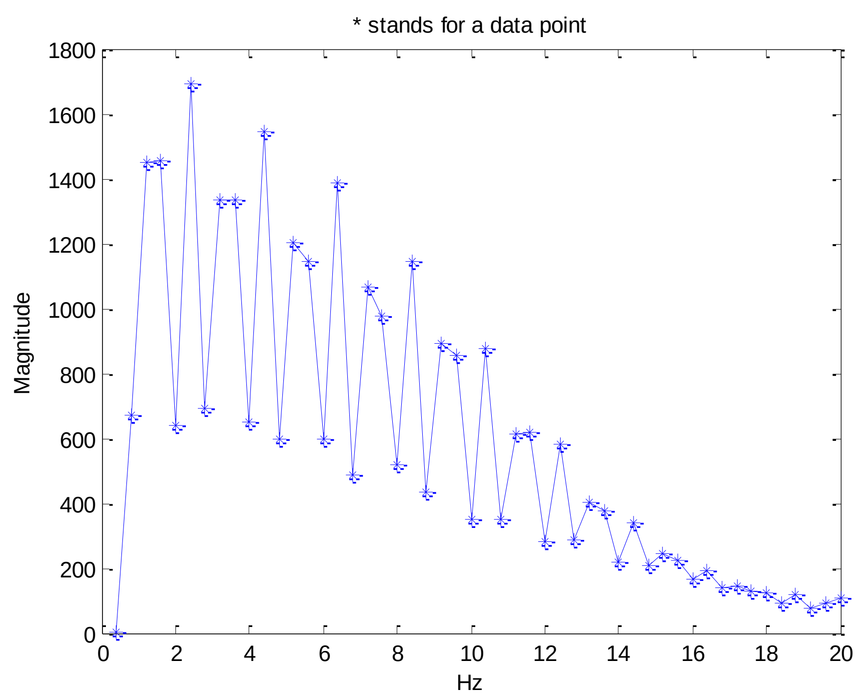
In real time processing, the program cannot wait for long before it processed the data. The delay depends on what frequency component we are aiming at. According to Nyquist theory, to identify the 1Hz frequency, at least 2 seconds of data is required. Experiments have shown that, in order to detect the 1Hz component, at least 10 consecutive packages are required, which takes about 2.5 seconds.

According to the mechanical characteristics of the LIS3LV02DQ accelerometer, the data rate has a limitation of 40Hz. Therefore, the processor board in the sensor node cannot tickle faster than that and instead, it accesses the sensor at a rate between 39.52Hz and 39.84Hz. That is to say, in this sensor system, only the 0~20Hz frequency range is examined and anything beyond that is aliased.

Figure 18 shows two different spectra taking from the 1Hz hitting signal. The first spectrum is obtained from 2 seconds of the signal, which is aliased. The second one is obtained from 2.5 seconds long signal. It picks up the 1Hz frequency and all harmonics.



(a) Spectrum of 8 Packages: Aliased



(b) Spectrum of 10 Packages

Figure 18: Spectrum for Different Number of Packages of 1Hz Signal

Lab test in this section demonstrates that the sensor network is able to acquire features from both time and frequency domain. Now it is ready to pick up real data and develop base station application for encroachment detection.

3.2 Field Test

In real application, sensors are only placed in urban expansion areas where there is a high probability of excavation activities. Therefore, local geological surface plays an important role in ground vibration. Some surface may transmit ground vibration better than the others. Moreover, the gravitational acceleration varies from place to place on the earth, which causes different comparison benchmarks as the critical location changes.

3.2.1 Experiment Setup

In this experiment, two commonly seen rural area ground surfaces are to be tested on: sand/gravel land and grassland, as is shown in Figure 19. The experiment is held in front of the Magnetic Resonance Systems Lab located in College Station, TX.

Figure 19: Two Types of Ground Surface: Sand/Gravel Land (left) and Grassland (right)

Monitoring pipeline safety using a small sensor network is a challenging task. In real environment, almost all types of activities can inflict a change in the local gravitational acceleration right on the critical spot. The activity may be from a person walking by, a sedan driving through, a farming machine or even an earthquake. Among all possible activities, only the excavation is of our great concern. The challenging part is that the small wireless sensor network relies a lot on one factor: gravitational acceleration. It is the goal to find out whether or not this only factor is able to tell the truth.

Theoretically, all possible equipment should be tested on all types of ground surfaces to ensure the accuracy of the proposed pattern recognition algorithm. In other words, the extracted features should belong to one particular digging activity only and the extracted features should be able to tell digging activities from all kinds of other situations. However, real experiment cannot achieve this. Therefore, the most commonly seen vehicle activity is chosen in this experiment to represent the “other activities” as a comparison to harmful excavation actions for the purpose of reducing false alarms in order to make algorithm more reasonable.

Each of the wireless sensor nodes is placed right on the ground to collect vertical acceleration data throughout the test. In real deployment, they need wrapper cases as a protection and they should be fixed on the ground surface to prevent the sensor node leaving the critical location. However, previous experiments did not find the suitable materials for wrapper case or fixture. They all reduce the sensitivity of the sensor and make sensor readings almost a constant number, which is of no help in feature extraction.

In the future we will continue trying different materials and also different shapes of wrapper cases. In this experiment, sensors are put on the ground without wrapper case or fixture.

In this experiment, two types of equipments are tested: rotary tiller (left) and Toyota Camry car (right), which is shown in Figure 20. The rotary tiller is a motorized cultivator that works the ground by means of rotating tines or blades. Since pipelines are usually buried shallow right beneath the ground, these tines/blades could cause severe damage.

Figure 20: Two Equipments Used in Experiment: Tiller (left) and Car (right)

In total, 4 sets of data are collected: signals from two equipments on two ground surfaces. Moreover, each set of data contains signals collected at seven different distances from working equipment: 0m, 0.5m, 1.0m, 1.5m, 2.0m, 2.5m, and 3.0m.

In total, for each ground surface, 15 set of signals are collected: one background signal, seven vehicle signals and seven tiller signals. In order to reduce loss of generality, each signal is obtained on different critical spots chosen randomly on the ground.

3.2.2 Results

In this section, data obtained on both sand/gravel land and grassland is examined to extract the features. Ideally, the background signal should be the local gravitational acceleration. However, experiments demonstrate that gravitational acceleration is slightly different in the two ground surfaces. The fundamental of this experiment is that working equipment alters the local gravitational acceleration slightly and the amount of change depends on both the equipment type and distance from the equipment. Therefore, the objective of result analysis it to find out this “changes of acceleration” and proves a one-to-one mapping between each change and its corresponding activity. In other words, only when a change belongs to a particular case (equipment type and the distance from it) alone, it is valid feature that can be employed in detection algorithm.

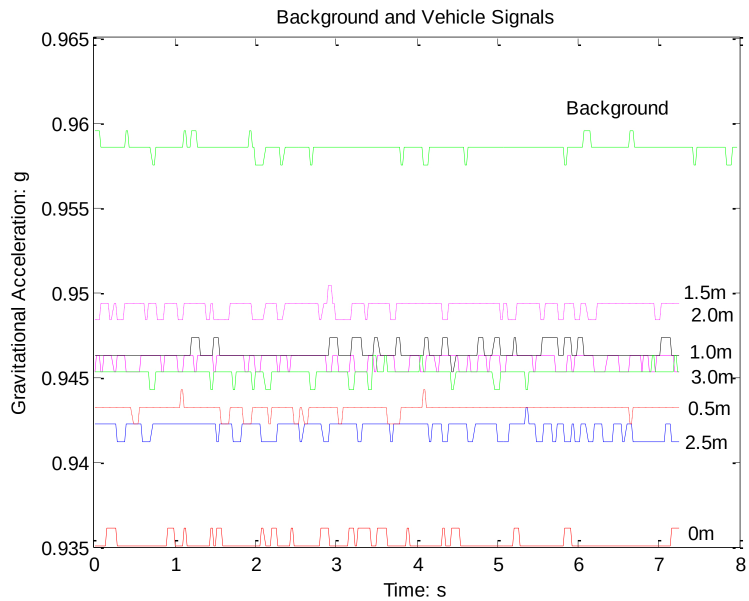
3.2.2.1 Results on Grass Surface

Figure 21 shows that the sensor network is collecting vehicle signals. The left sensor is the sensor node used for acquiring gravitational acceleration data. The right one linked with a USB cable is the gateway sensor. The gateway sensor is deployed with such a program that people can tell the working status of this sensor from the on board LED light. When the sensor is up and running its program, the LED light turns blue. When the sensor receives a package of data, the LED light turns white for quite a short moment than goes back to blue again. When the sensor network is working continuously, the LED light on the receiver sensor shines with alternative blue and white color. In case of package loss, the sensor stays in blue for a longer time.

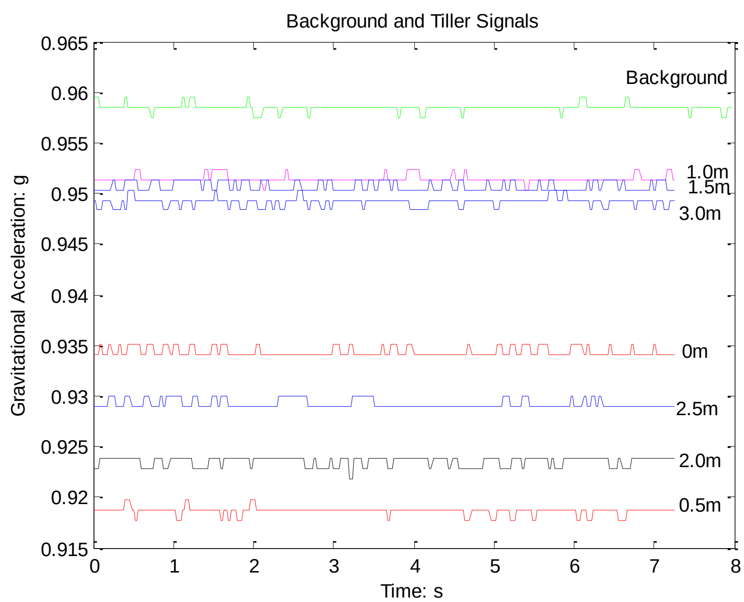
To avoid package loss and also to ensure data points have a fixed interval, sensor node and gateway sensor are placed close to each other, as is shown in Figure 21.

Figure 21: Collection of Vehicle Data on Grass Surface

Figure 22 contains two charts showing tiller and vehicles signals. The result shows that even at a sampling rate of 40Hz, the signals do not exhibit any obvious features. Instead, the signal is very stable and mostly flat. Since the ST Micro LIS3LV02DQ 3d 12 bit $\pm 2g$ accelerometer has a $\pm 2g$ error range, it is possible that these up-and-downs in the data may be caused by mechanical characteristics of the sensors, i.e., internal noise, not the ground vibration.



(a) Background and Vehicles Data on Grass Surface



(b) Background and Tiller Data on Grass Surface

Figure 22: Data Collected on Grass Surface

To compensate for the impact of sensor mechanical characteristic, Figure 23 displays grassland data obtained from taking the average of every 2 packages.

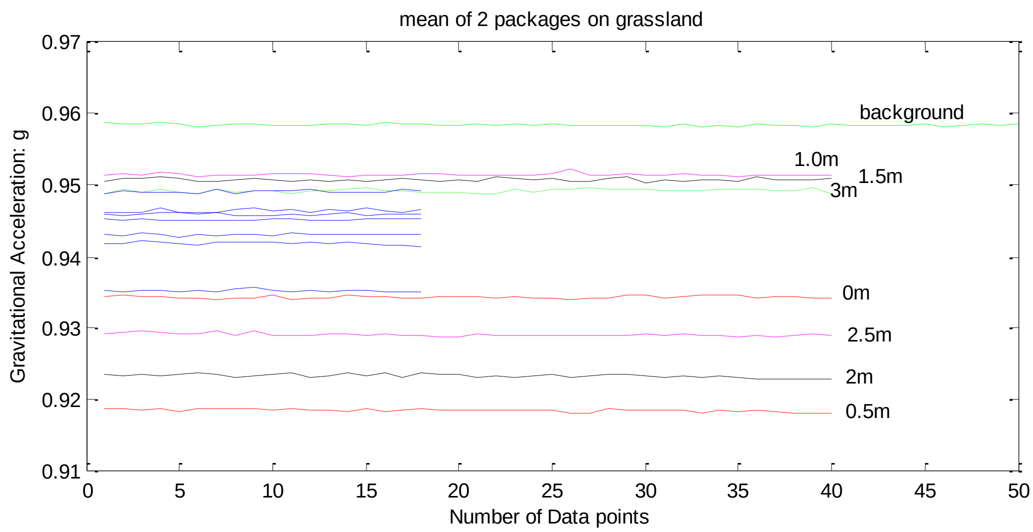
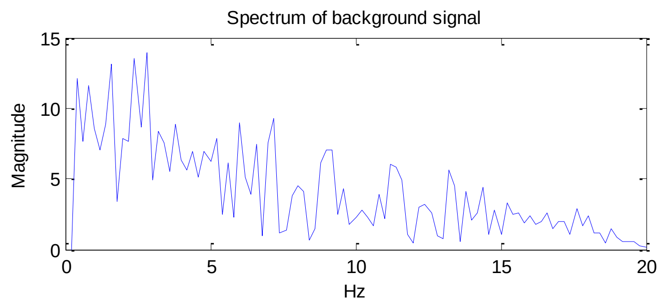


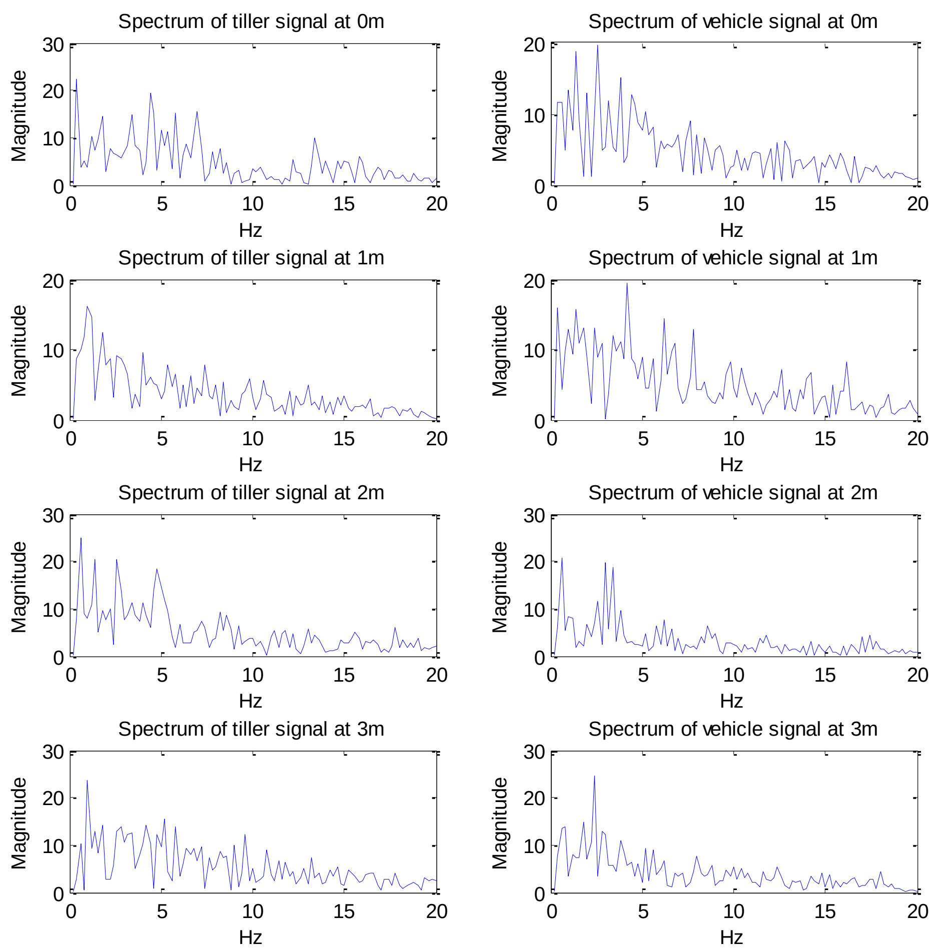
Figure 23: Mean Values of Every 2 Packages of Data on Grass Surface

It can be observed from the figure above that all vehicle data fit in the range between 3m signal and 0m signal, though one vehicle data overlap the 3m tiller signal. After taking the average, data becomes even more flat and stable. The figure proves that 6 sets of tiller signals can be identified by their magnitude alone. In real time processing, 2 packages is enough for the base station application to make a decision and it takes only 0.5 second.

The next step is to examine the spectrum of these data sets and to see if more features can be observed. Figure 24 displays 9 sets of data. There is no obvious feature to separate any set of signal from the others, not to mention in real time processing.



(a) Spectrum of Background Data



(b) Spectrum of Tiller and Vehicle Data

Figure24: Spectrum of Data Collected on Grass Surface

Given this fact, frequency domain analysis is not able to perform the feature extraction work.

Since the signal fluctuation range is the only factor that is of great concern, how it looks or how it oscillates no longer matters. Therefore, the data can be plot in the way shown by Figure 25.

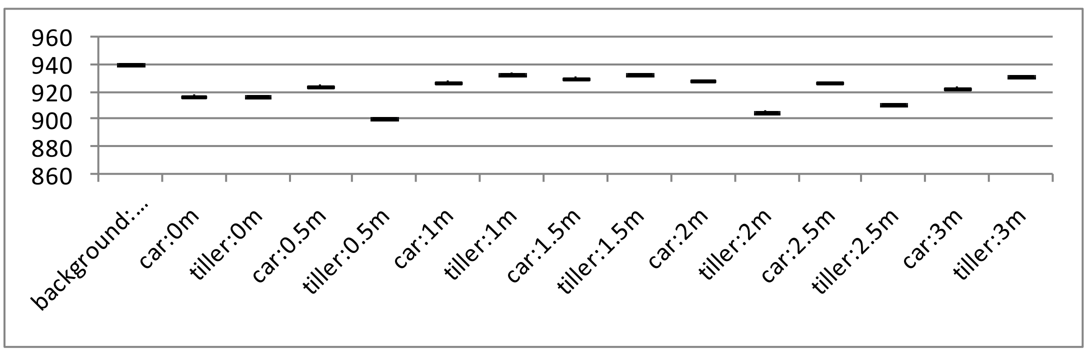


Figure 25: All Data Collected on Grass Surface: 2 Package Mean

The figure above shows the ranges of 2-package-average-data collected on grassland. In this figure, each signal is represented by a small rectangle (block). The vertical axis becomes raw accelerometer reading. The height and vertical position of each rectangle illustrate its fluctuation range. The width of each rectangle does not matter. The leftmost block represents background signal. In this figure, all 15 set of signals on the grassland are clearly shown. The non-monotonic feature is obvious in this figure.

In conclusion, if the tiller is working at a distance of 0m, 0.5m, 1.0m, 1.5m, 2.0m or 2.5m away from the critical location, it can be detected immediately judging from the mean value of two packages of data.

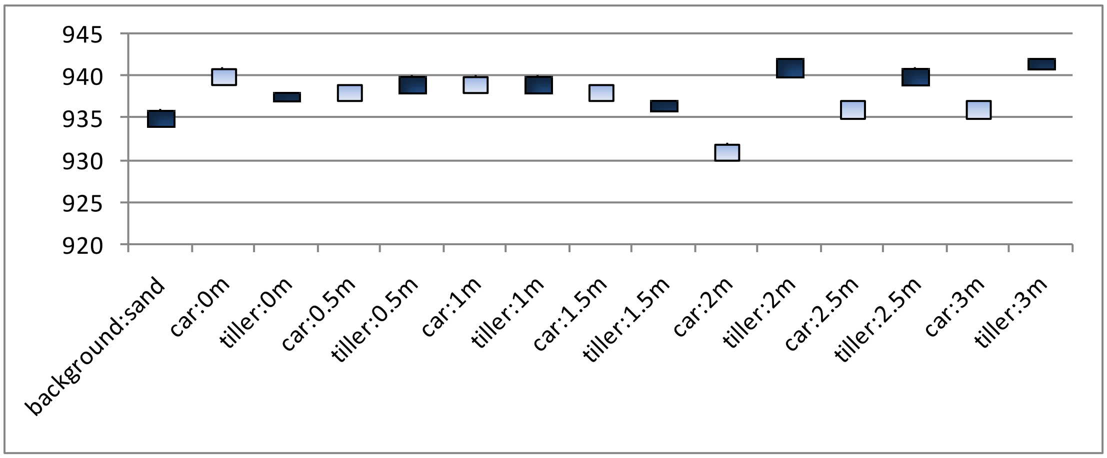
3.2.2.2 Results on Sand/Gravel Surface

Figure 26 shows that the sensor is collecting gravitational acceleration data impacted by a working tiller.

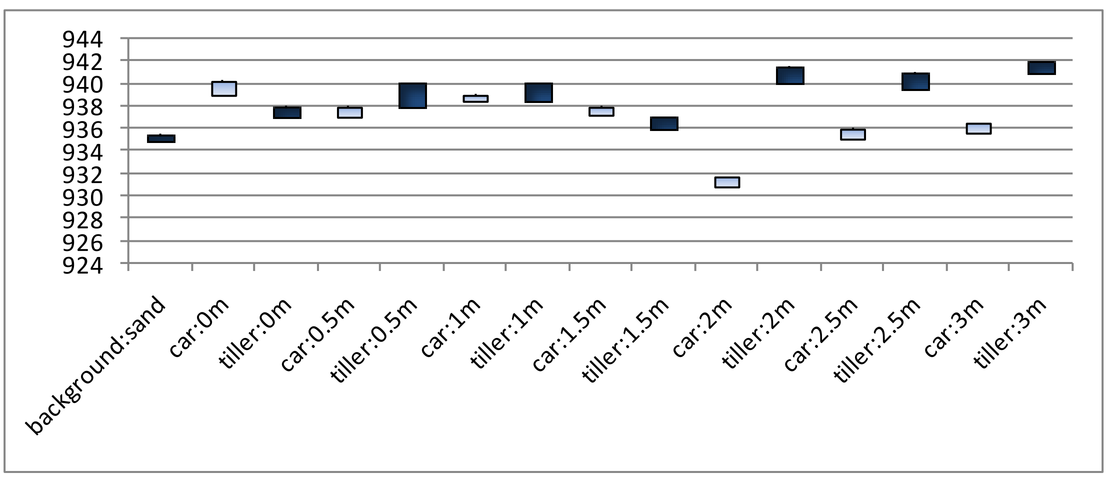
Figure 26: Collection of Tiller Signals on Sand/Gravel Surface

Compared to signals collected on grassland, the problem of range overlapping is much more severe on the sand/gravel surface. Therefore, in this section data are not plot with regard to the time axis. There is one thing in common between the data collected on two different ground surfaces: signals are stable with fluctuation in a small range.

First, feature extraction work relies on the fluctuation range and value distribution. The range plotting is shown in Figure 27.



(a) Original Data



(b) 2-Package Average Data

Figure 27: All Data Collected on Sand/Gravel Surface

The signals collected on sand have more overlaps compared to the grassland case. Figure 27 shows that taking the average does not solve the overlapping problem of sand/gravel data.

Since the critical location case is of the greatest concern, the next step is to look into its distribution. Table 1 lists all the signals on sand/gravel land whose magnitude range overlap that of the critical location data. The first column gives all possible values these signals may take: from 935 to 938. The rest of the table lists the distribution: probability of occurrence for different values. Since vehicles signals collected at 2.5m and 3m have similar distribution, they are combined into one column and the same reason is with tiller signals at 0.5m and 1m.

Table 1: Magnitude Range Compared to Tiller Data at Critical Location

Magnitude on sand	tiller:0m	car: 0.5m	tiller: 0.5m,1m	car: 1m	car: 1.5m	tiller: 1.5m	car: 2.5m,3m
935							10.6%
936						52.7%	83.5%
937	45.0%	48.7%			36.8%	47.3%	5.9%
938	55.0%	51.0%	1.9%	17.4%	64.2%		
939		0.3%	80.5%	81.4%	0.8%		
940			17.6%	1.2%			

Some of the overlap is benign and does not affect the decision. For example, in the fourth and fifth column, once 939 is detected, which is quite possible, they are out; in the last column, once a 936 is detected, it is also certain there is no tiller working at the critical location. Both signals have a high probability of falling out of the range that belongs to the critical location signal. Therefore, so long as magnitude ranges are not exactly the same, overlapping may not be a big problem. By the same standard, the table needs to be simplified and the result is shown in Table 2.

Table 2: Data with Range Overlaps That of Tiller Data on Critical Location

Magnitude on sand	tiller:0m	car:0.5m	car:1.5m
937	**45.0%**	**48.7%**	**36.8%**
938	**55.0%**	**51.0%**	**64.2%**
939		0.3%	0.8%

Table 2 shows a real problem in feature extraction. Overlap itself may not pose as a huge problem per se. But if the range of one signal is a subset of that of another, separation between the two is no longer possible in our situation. It is true that if a 939 is detected, the situation would be very clear.

The problem is: it is not wise to rely on a small probability event for anomaly detection so long as real time processing is still our goal. In other words, there is no criterion to detect a tiller working at the critical location.

Table 3 shows it is also not feasible to detect a tiller 0.5m or 1.0m way since they have exactly the same range as vehicle signal 1.0m away.

Table 3: Magnitude Distribution Compared to Tiller Data at 0.5m and 1.0m

Magnitude on sand	tiller:0.5m,1m	car:1m
938	1.9%	17.4%
939	80.5%	81.4%
940	17.6%	1.2%

As we move on to tiller signals further away from the critical location, the meaning of detection drops. However, we would still check all tiller signals to find out if any of them can be detected.

Table 4 lists all signals with overlapping magnitude range with tiller signal collected 1.5m away.

It can be observed that the range of tiller signals at 1.5m away is the subset of two vehicle signals' range, which is marked in bold in the table. What is worse, the two vehicle signals shown in the last column only have a small probability of 10.6% to fall out of the tiller signal range.

Table 4: Magnitude Distribution Compared to Tiller Data at 1.5m

Magnitude on sand	tiller:1.5m	tiller:0m	car:0.5m	car:1.5m	car:2.5m,3m
935					10.6%
936	**52.7%**				**83.5%**
937	**47.3%**	45.0%	48.7%	36.8%	**5.9%**
938		55.0%	51.0%	64.2%	
939			0.3%	0.8%	

Table 5 shows the distribution of all signals whose range overlaps that of tiller signal collected at 2.0m and 2.5m away.

Table 5: Magnitude Distribution Compared to Tiller Data at 2.0m and 2.5m

Magnitude on sand	tiller:2m	tiller:0.5m,1m	tiller:2.5m	tiller:3m	car:0m	car:1m
938		1.9%				17.4%
939		80.5%	0.4%		19.0%	81.4%
940	**48.7%**	17.6%	**36.1%**		**77.4%**	1.2%
941	**50.9%**		**63.5%**	47.6%	**3.6%**	
942	0.4%			52.4%		

Table 5 shows that both tiller signals have a high possibility of 99.6% of taking the value 940 or 941. In other words, if either of the two readings is obtained, the sensor will not be able to provide a definite judgment. However, vehicle signals collected at the critical location has a high probability 81.0% of taking these two values.

Then it moves on to whether a working tiller 3m away can be identified. Till now, it can already be concluded that the detection work on sand/gravel surface not efficient enough. If an excavator is working on the ground three meters from the critical location, it is quite possible that the person working on the excavator already sees the warning signs. However, the result is still shown in Table 6 and the detection result can be processed as a “might-be-dangerous” situation, not an emergency.

Table 6: Magnitude Distribution Compared to Tiller Data at 3.0m

Magnitude on sand	tiller:3m	tiller:2m	tiller:2.5m	car:0m
939			0.4%	19.0%
940		48.7%	36.1%	77.4%
941	47.6%	50.9%	63.5%	3.6%
942	52.4%	0.4%		

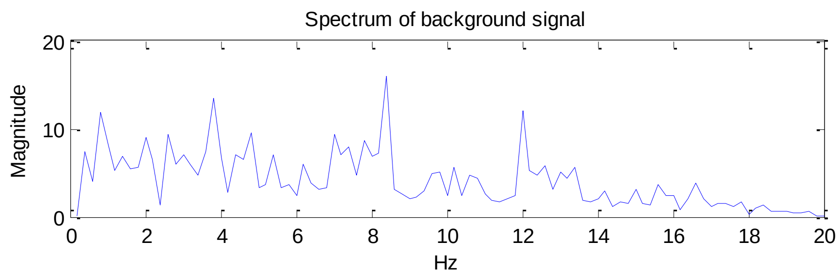
The table above gives 3 tiller signals and 1 vehicle signal. The tiller signal in the second column can take two values: 941 and 942 with almost a half-half chance. If we can separate this vehicle signal from tiller signal collected at 3.0m away from critical location, the detection is valid.

It is not a big issue that the range of tiller at 3.0m is the subset of range of tiller at 2.0m, as long as they are both tiller signals. It can be observed that the tiller working at 3m away from the critical spot takes the value 942 with probability 52.4%. In other words, if a 942 is detected, it is certain that a tiller is working nearby, more likely it is at 3.0m away.

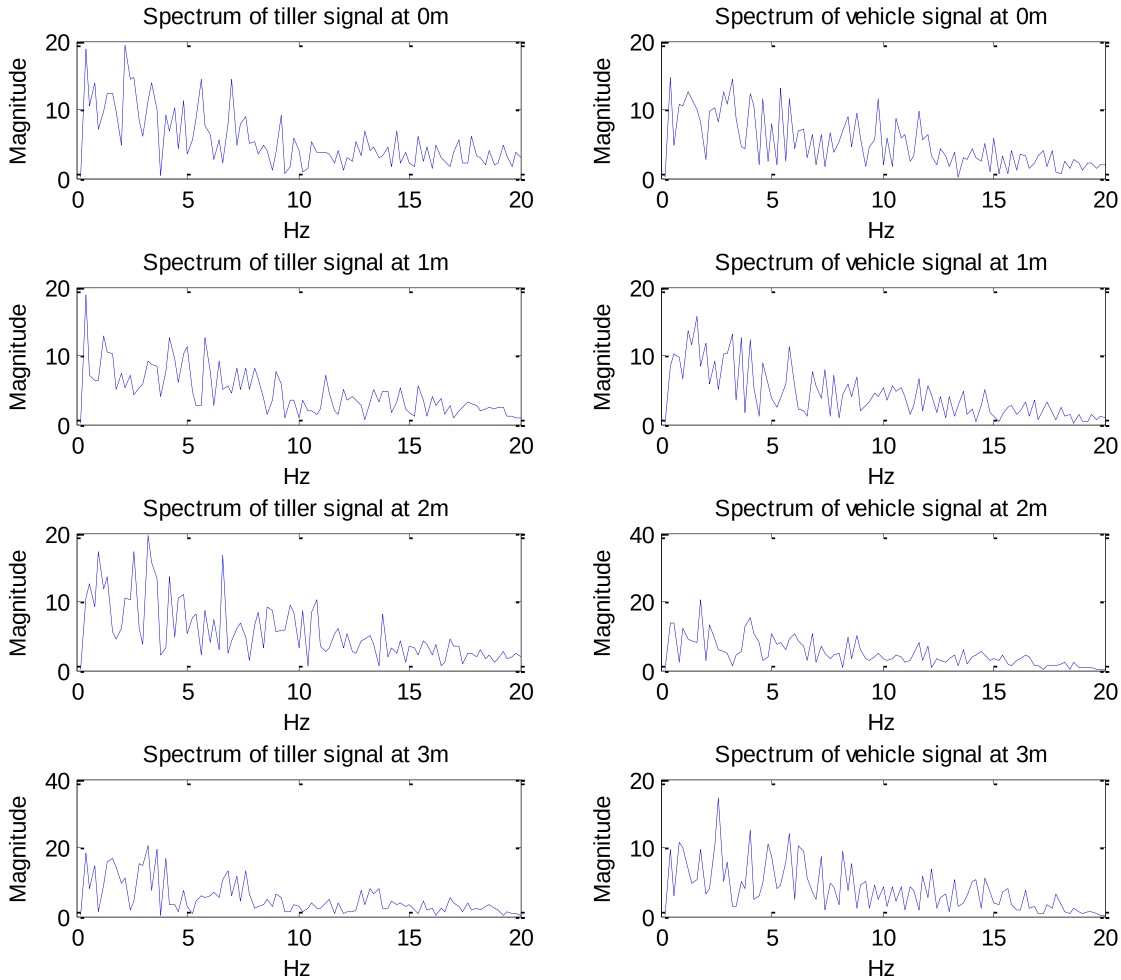
These results show that magnitude features can be potentially used to detect activities close to the sensor location (<2 meters on grass surface). However, they are not sufficient for encroachment activity detection, i.e., differentiate activities caused by different equipment.

In the bandwidth of the current accelerometer/sensor board (40 Hz), it is not possible to detect the presence of an evacuation equipment (a tiller or a car in this example), or to identify the type and distance of such equipment.

Figure 28 illustrates the spectrum of 7 sets of data collected on sand/gravel surface. The signals were processed in the same way as in the lab test to extract the spectrum feature.



(a) Spectrum of Background Data



(b) Spectrum of Tiller and Vehicle Data

Figure28: Spectrum of Data Collected on Sand/Gravel Surface

4. DISCUSSIONS AND FUTURE WORK

Several technical and practical issues must be addressed in order to make this work useful for pipeline encroachment detection.

First and foremost, the sensors need to be packaged and mounted or a solid shell. This will make it easier to maintain the sensors and make it work under all weather conditions. For example, how it works if it is raining or snowing at the critical location.

External antenna might also be included to extend the wireless range. The packaging should be carefully designed to minimize its effects on the sensor performance or wireless communication. In the future, wrapper cases and fixtures are to be tested and see if they can improve the sensor's sensitivity. Also, we will look for sensors with better sensitivity and explore packaging methods that can boost the coupling of signal from the ground to the sensors.

In addition, it is also necessary to modify the hardware (write value to certain registers) configuration of the accelerometer in order to improve the bandwidth. The linear accelerometer used in the work (LIS3LV02DQ) can output vibration data at a rate as high as 2650 Hz. However, the sensor board does provided by the vendor is fixed at 40 Hz output rate. Means to overwrite this limit will enable faster signal sampling to provide more reliable spectrum information that might be critical for encroachment detection.

As more sensors are used in the critical location, the quality and accuracy of the encroachment detection increases. Then an issue arises: if many sensor nodes send data back to the same gateway sensor, how the bandwidth would be shared among each sensor? Either the bandwidth is equally shared or certain motes get advantage in transmission. Despite significant technological advances, constraints in WSN functionalities still demand further investigation [17]. New techniques in wireless network protocols, time synchronization among sensors and data aggregation must be implemented for WSN to achieve reliable operation and expand their applicability [18].

In the project, experiments were performed on horizontal and flat ground surface. In the tests, sensor system was deployed in flat rural areas with few obstacles. This ensured reliable communication between sensor nodes and gateway sensor. However, practical development in other terrains may present problems for reliable wireless communications. Various obstacles such as a metal plate or giant rock between the gateway sensor and wireless sensor nodes may negatively affect the performance.

In real applications, the parameters obtained in our experiment may not apply to other situations since gravitational acceleration varies in different locations and on different ground surfaces. Moreover, test results from other excavation equipments such as backhoe, trencher or ditch witch are also needed.

In short, further research, testing and optimization are required to achieve the new and improved levels of detection accuracy that can potentially make the WSN pipeline encroachment detection a reliable technique.

5. CONCLUSIONS

This thesis proposes using WSN to detect pipeline encroachments due to excavation activities. We have presented a platform including vibration sensors, wireless communication, detection algorithm (on a field computer), and how the warning can be communicated with pipeline company. Preliminary lab and field tests were performed on a prototype system developed based imote2 WSN. The result shows that this system can detect vibration signals due to close excavation activities (represented by a tiller in the experiment) on and near the critical locations (< 2 meters on grass surface). The results from sand/gravel surface do not seem very effective but it can still provide some information about the distance of a working tiller.

Several significant issues and challenges were identified in the studies. First, the current WSN accelerometer cannot provide data sampling beyond 40 Hz. This is largely due to a vendor restriction. Such sampling rate significantly limits the ability of the WSN to identify useful spectrum features for encroachment detection. Second, the sensitivity of the vibration sensor needs further improvement so activities at longer distance can be detected.

This thesis provides experiment methodology and the results can be used as a reference. If a light excavation activity caused by the tiller can be detected, the detection of other heavier equipments is expected to be more straightforward.

REFERENCES

[1] J. E. Huebler, State-of-the-Art in detection of unauthorized construction equipment in pipeline Right-of-Ways, http://www.netl.doe.gov/technologies/oil-gas/publications/Status_Assessments/state-of-the-art.pdf, last accessed: May 2011.

[2] Pipeline Safety Community, Analysis of DOT Office of Pipeline Safety natural gas pipeline operators' incident summary statistics for 01/01/86 –06/21/1999, http://phmsa.dot.gov/pipeline, last accessed: May 2005.

[3] J. Stancliffe, Third party damage to major accident hazard pipelines, http://www.hse.gov.uk/pipelines/ukopa.htm, last accessed: February 2011.

[4] G. Burkhardt, Real-time monitoring of pipelines for third party contact, http://www.netl.doe.gov/technologies/oilgas/publications/Status_Assessments/10211TechStatusAssess-Final.pdf, last accessed: March 2011.

[5] B. Leis, Real time monitoring to detect contact, product loss, and encroachment on transmission pipelines, in: 54th American Petroleum Institute Pipeline Conference, April 2003, pp.1-6.

[6] R. H. Doctor, N. Dunker, Field evaluation of a fiber-optic intrusion detection system, in: 9th Global Reporting Initiative Final Report, Tulsa, Oklahoma, December 1995.

[7] K. B. Fung, D. W. Fraser, R. P. Gauthier, Application of remote sensing data for monitoring of gas pipeline right-of-way, in: Proceedings of 17th Geographic Information System, April 1998, pp. 401-404.

[8] Y. Lu, J. Tang, Prototyping a wireless sensing platform for acoustic emission signals collected by microphone arrays, in: Proceedings of 24th Society of Photo-Optical Instrumentation Engineers, 2009, pp. 477-483.

[9] Crossbow Technology, Inc., Imote2 Hardware Reference Manual, http://www.xbow.com.cn/LinkClick.aspx?fileticket=i7FK6%2B1oodU%3D&tabid=121, last accessed: May 2011.

[10] J. Rice, B. Spencer, Structural health monitoring sensor development for the Imote2 platform, in: Proceedings of 23th Society of Photo-Optical Instrumentation Engineers, 2008, pp. 2165-2185.

[11] Y. Peng, R. Lahusen, B. Shirazi, W. Song, Design of smart sensing component for volcano monitoring, in: The Institution of Engineering and Technology 4th International Conference on Intelligent Environments, 2008, pp.1-7.

[12] D. Lymberopoulos, T. Teixeira, A. Savvides, Macroscopic human behavior interpretation using distributed imager and other sensors, in: 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2007, pp. 821-827.

[13] Crossbow Technology, Inc., Imote2.Builder Kit Manual, http://www.hoskin.qc.ca/uploadpdf/Instrumentation/divers/CrossBow/divers_Imote2.Builder_kit_48ea1c4265c6e.pdf, last accessed: June 2011.

[14] STMicroelectronics, Inc., LIS3LV02DQ MEMS Inertial Sensor, http://www.sparkfun.com/datasheets/IC/LIS3LV02DQ.pdf, last accessed: May 2011.

[15] http://en.wikipedia.org/wiki/IEEE_802.15.4, last accessed: March 2011.

[16] http://en.wikipedia.org/wiki/Gravitational_acceleration, last accessed: March 2011.

[17] N. Castaneda, F. Sun, S. Dyke, C. Lu, A. Hope, T. Nagayama, Implementation of a correlation-based decentralized damage detection method using wireless sensors, in: 11th American Society of Engineering Management Conference, 2008, pp. 47-56.

[18] T. Nagayama, Structural Health Monitoring Using Smart Sensors, Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2007.

APPENDIX

Sensor Node

The program running on the wireless sensor node is responsible for accessing the vertical axis accelerometer at a certain rate and packs a message after obtaining 10 readings. Part of the code is shown below.

```
int sampling_interval = 23; // The sampling interval 0.023 seconds ensures that
                            //  the data rate is below but
                            // close to the accelerometer's 40Hz bandwidth.
for (; ; )
{
   accelZ = _sensor.AccelZRaw ; // access the vertical axis acceleromter
   packet[index++] = (byte)(accelZ & 0xFF);
   packet[index++] = (byte)((accelZ >> 8) & 0xFF);
   System.Threading.Thread.Sleep(sampling_interval);

   accelZ = _sensor.AccelZRaw;
   packet[index++] = (byte)(accelZ & 0xFF);
   packet[index++] = (byte)((accelZ >> 8) & 0xFF);
   System.Threading.Thread.Sleep(sampling_interval);

   accelZ = _sensor.AccelZRaw;
```

```
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);

accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);

accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);

accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);

accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
```

```
System.Threading.Thread.Sleep(sampling_interval);


accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);


accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);


accelZ = _sensor.AccelZRaw;
packet[index++] = (byte)(accelZ & 0xFF);
packet[index++] = (byte)((accelZ >> 8) & 0xFF);
System.Threading.Thread.Sleep(sampling_interval);
  int t = System.DateTime.Now.Millisecond;
  packet[2] = (byte)(t & 0xFF);
  packet[3] = (byte)((t >> 8) & 0xFF);
  if (_radio != null)
  {
    // send it to the radio
```

```
            _radio.SetRadioOption(RadioOption.LocalAddress, (ushort)_nodeid);

            _radio.amType = _amType;

            _radio.Send((ushort)0xFFFF, (ushort)0xFFFF, packet);

        }

        //send it to uart

        TosSerialDump.printWithTosHeader(packet);

        index = 4;

    }
```

Real Time Plot Algorithm

```
      tmp[i] = int.Parse(arr[2 * i + 4]) + (int.Parse(arr[2 * i + 5])) * 256;
       if (tmp[i] > 32768)
           tmp[i] = tmp[i] - 65536;

    if((int.Parse(arr[0]) == 4))
                //first sensor with node id 4 14, which is the new sensor
    {

       newTime1 = int.Parse(arr[2]) + (int.Parse(arr[3])) * 256;
                  //  the new time stamp

       int itv1 = newTime1 - oldTime1;
       if (itv1 <= 0)
       itv1 = 999 - oldTime1 + newTime1;

       double ttmp = itv1/10000.0;

       oldTime1 = newTime1;
                // new intervals for new 10 data
       demo.Myplot(0,1, ttmp, (MWNumericArray)tmp);

    }
```

Matlab Code for Myplot

```
function Myplot(node,t,tmp)
```

```
    load t1.mat
    load d1.mat
    load t2.mat
    load d2.mat
if(node == 1)
    d1(1:4990) = d1(11:5000);
    d1(4991:5000) = tmp(1:10);
    t1(1:4990) = t1(11:5000);
    t1(4991:5000) = t1(4990:4999) + t;
    save d1.mat d1
    save t1.mat t1
end
if(node == 2)
    d2(1:4990) = d2(11:5000);
    d2(4991:5000) = tmp(1:10);
    t2(1:4990) = t2(11:5000);
    t2(4991:5000) = t2(4990:4999) + t;
    save d2.mat d2
    save t2.mat t2
end
end
```

VITA

Name: Xiaoxi Ou

Address: Department of Electrical and Computer Engineering,

Texas A&M University,

214 Zachry Engineering Center,

TAMU 3128

College Station, TX 77843-3128

Email Address: xiaoxi.ou86@gmail.com

Education: B.S., Electrical Engineering,

Nanjing University of Aeronautics and Astronautics, 2008

M.S., Electrical Engineering,

Texas A&M University, 2011