Graduation Thesis Academic Year 2012

Pattern-based Matrix-size Optimization Algorithm for Compressive Sensing in Real-world Body Sensor Networks

Advisors:
Professor Hideyuki Tokuda
Professor Jun Murai
Professor Osamu Nakamura
Associate Professor Hiroyuki Kusumoto
Associate Professor Kazunori Takashio
Associate Professor Rodney D. Van Meter III
Associate Professor Noriyuki Shigechika
Associate Professor Keisuke Uehara
Associate Professor Jin Mitsugi
Assitant Professor Jin Nakazawa
Professor Keiji Takeda

Keio University Faculty of Policy Management
Akito Ito

aqram@ht.sfc.keio.ac.jp

Keio University Academic Year 2012

Pattern-based Matrix-size Optimization Algorithm for Compressive Sensing in Real-world Body Sensor Networks

Keio University Faculty of Policy Management

Akito Ito

Abstract

Pattern-based Matrix-size Optimization Algorithm for Compressive Sensing in Real-world Body Sensor Networks

Summary

Compressive Sensing (CS) is a novel approach for data representation, which can represent signals at a rate below the Nyquist rate with low computation costs on encoder. For these characteristics, CS is very suitable for low power sensor nodes to save power consumption that is a primary problem in Wireless Sensor Networks (WSN). But there are many problems when using CS in a real environment, especially in Body Sensor Network which aims to monitor human health and detect context. One of these is that pattern of sensor values change dynamically. It decreases the efficiency of power consumption and accuracy of recovery. To solve the problem, we propose Pattern-based Matrix-size Optimization Algorithm (PMOA), which aims to improve the accuracy of exact recovery and power consumption. We performed experiments both in real world and simulation and the result show our approach is effective in energy consumption and reliable. The result shows our approach can achive the improvement of lifetime by 11.7%.

Keywords:

1 Wireless Sensor Networks　2 Body Sensor Networks　3 Compressive Sensing
4 System Architecture

Keio University Faculty of Policy Management
Akito Ito

卒業論文要旨　　2012 年度 (平成 24 年度)

パターン情報を用いたボディセンサネットワークにおける圧縮センシングのための効率化手法

論文要旨

圧縮センシングは、低いエンコードコストでより高い圧縮率を得られるデータ圧縮手法である。その特徴より、無線センサネットワークでの応用に適しているが、実世界で運用するためには様々な課題がある。センサデータは動的に値が変化するので、正しい設定をエンコードで行うことが出来ず、復元率が下がったり、より多くの電力を消費してしまう。特に、人のモニタリングやコンテキスト検知を行うことを目的としたボディセンサネットワーク環境では、　顕著にその影響が出てしまう可能性がある。本研究では、あらかじめ設定されたパターンにより、動的に圧縮センシングのパラメータを変化させる、Pattern-based Matrix-size Optimization Algorithm (PMOA) を提案し、より高い復元率と低消費電力を実現する。実験の結果、PMOA は 11.7%の消費電力の改善を可能にした。

キーワード：
1 ボディセンサネットワーク　2 圧縮センシング　3 無線センサネットワーク
4 システム設計

慶應義塾大学総合政策学部
伊藤　瑛

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, Wireless Sensor Networks (WSN) are becoming important many applications such as environmental monitoring, target tracking and health monitoring. In this chapter, we will show the background, motivation and objective of the research, followed by the organization of the thesis.

1.1 Background

By the development of Micro-Electro-Mechanical Systems (MEMS) and wireless technology, sensor nodes have been more smaller and cheaper than previous one. As a result, various kinds of applications such as environmental monitoring, target tracking are proposed by many researchers. Among these, one of the most attractive applications is Body Sensor Network (BSN) that aims to monitor human health or detect context by putting sensors on human body and processing these data. Although these developments are remarkable, there are still problems in practical use of it; Since tiny sensor nodes are designed cheap and small, they have a few resources, such as CPU, storage, memory and energy sources. Especially, compared to normal WSN applications, WeSN handle with large and complex data, including multi-axis acceleration and pressure, in high frequency. Therefore, the WeSN node is under harsh conditions in terms of data processing and energy consumption. To solve this, many developer and researcher propose protocols and algorithms, including data compression and aggregation.

On the other hand, Compressive Sensing (CS) is emerging as a novel data sensing paradigm that overturn the previous data representation theory; CS can express certain data under the Nyquist rate. Since CS encodes the data with a few extra computation costs, it is very suitable for tiny sensor nodes with low power CPU. Hence, many research apply CS theory to WSN in order to reduce the data and energy consumption that is one of the most important problem in WSN research.

1.2 Motivations and Objective

This thesis proposes a system architecture that allows tiny sensor nodes to apply CS in real-world WeSN. This system architecture aims to maximize compression ratio and recovery reliability of CS along specific WeSN application. It is important to achieve high compression ratio when compressing and reducing the sensing data because sensor nodes have severe constraint on resources. Although sensor nodes utilize low-energy, low-power radio chips, wireless communication consumes large amount of energy compared to other operations, such as calculation, sensing and data storing. Therefore, it is important to maximize the CS efficiency and minimize the length of payload. Besides, the recovery reliability of CS depends on signal characteristics. Since, sensor data changes dynamically and also changes their state especially in WeSN that generally uses sensor which collect continous data such as acceleration, pressure and gyro. encoding and decoding strategy of normal CS is unstable and will cause a reduction in the application service.

In order to solve these problems, we propose Pattern-based Matrix-size Optimization Algorithm (PMOA) which configures some CS parameters along application state. This thesis describes the design and implementation of this architecture and evaluates this architecture in some real world environment and in simulation.

1.3 Organization

This thesis is organized as follows. In Chapter 2, we introduce the background of WSNs and WeSNs. We then present the importance of data compression in wireless sensor networks, and describe algorithms and related works in Chapter 3. In Chapter 5, we present our approach on utilizing PMOA, an algorithm that improves efficiency and reliability in CS and explain its design. Chapter 6 describes our implementation of PMOA. We present the method and results of evaluation in Chapter 7. Finally, in Chapter 8, we conclude this thesis, and discuss our future work.

Chapter 2

Body Sensor Networks

In this Chapter, we discuss wireless/body sensor networks and energy consumption of tiny sensor nodes. First, we discuss traditional WSNs consisted of static sensor nodes. We then explain the energy consumption of tiny sensor nodes and finally, we describe our target environment, Body Sensor Networks (BSN), issues and requirements.

2.1 Wireless Sensor Networks

For the past few decades, the advances in technologies that relate to MEMS, various kinds of sensors, and batteries have lead to the development of tiny device equipped with sensors called sensor nodes. The improvement in these technologies have made sensor nodes smaller, cheaper and smarter. In addition, progress in wireless technologies provided these the ability to sensor nodes in order to communicate with other ones. These sensor nodes with wireless communication abilities are called wireless sensor nodes, whose examples are mote, SunSpot (also called OracleSpot), and μpart. Firstly, we will show various kind of applications and beneits of WSN. We then introduce the application area of WSN.

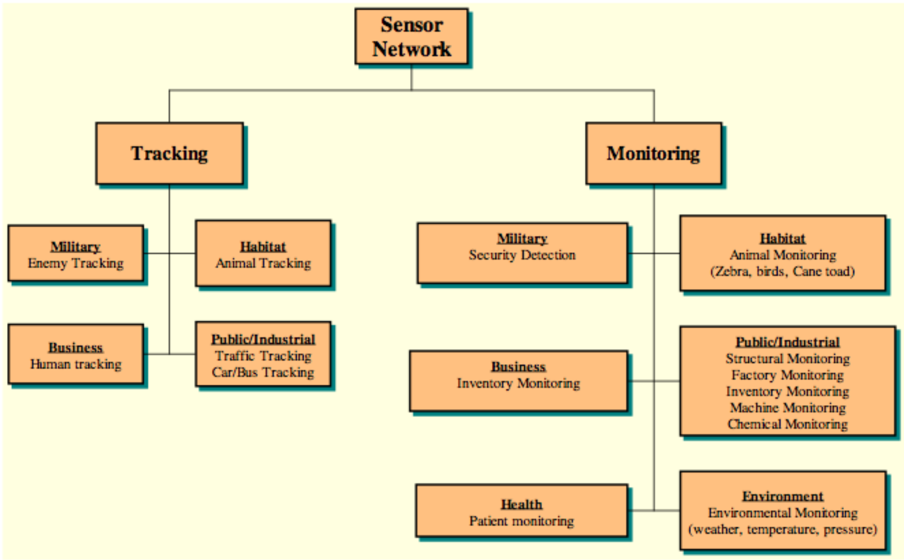

Figure 2.1: Overview of WSN applications [1]

2.1.1 WSN applications

WSN applications can be divided into two categories; environmental monitoring and target tracking (Fig .2.1). Monitoring applications include indoor/outdoor environmental monitoring, power monitoring, factory automation and etc. Tracking applications include tracking or counting objects, animals, humans and vehicles [1]. We will introduce examples of each application.

Macroscope of redwood [4] is a typical environmental monitoring application. Using WSN, this work monitors and records environmental data including temperature, humidity and solar radiation at the redwood trees in Sonoma, California up to 44days. Since the requirements of local biologist, including the dispersal patterns of wind-borne seeds, the water profiles, insect densities, the micro climate of meadow and etc, can be acquired by processing these data, which was previously achieved by walking across the forest and colleting these, benefits are significantly meaningful.

Design and deployment of industrial sensor networks [5] focuses on preventive equipment maintenance and it's cost. Preventive equipment maintenance is the idea that maintenance is needed only when equipment failure is predicted; regular check is not needed to reduce maintenance cost. This work achieved it by using WSN and gathering vibration data and also shows that the deployment cost of wireless sensor network system is cheaper than wired sensor network system.

ZebraNet [6] aims to track animal migrations by using mobile wireless sensor network. This system used GPS technology to record high-accurate position data. From this position data, the biologists can better understand the animal movements throughout the day.

2.1.2 WSN research area

To enable WSN application, described as above, there are sill many tasks and problems. These can be roughly classified to three areas (Fig .2.2);

(1) System area, including platform, OS, storage and perform evaluation.

(2) Communication protocol area which deal with data link, network and transport layer protocol,

(3) Service area includes localization, data aggregation, compression, security and etc.

Since wireless communication is the main factor of energy drain which is fatal problem for tiny sensor node which a few resources, communication protocol research is prosperous particularly. New protocols are needed because that sensor nodes have to self-organize, self-manage and self-control networks in high efficiency in order to reduce communication cost. However traditional networking protocols are not designed to meet these requirements. We will describe specifically about the relation between wireless communication and energy consumption at section 2.3.

In this point of view, our work addresses on service area. This area does not lead to improvement on these problems directly as communication

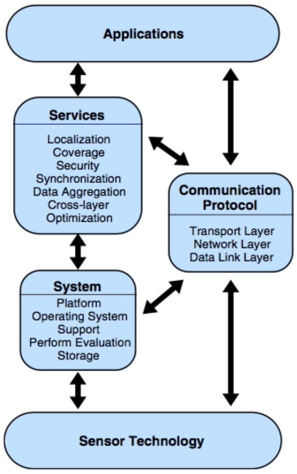

Figure 2.2: Classification of WSN research issues [1]

protocol research, but it can enhance these communication protocols and application service. As a result, a significant improvement can be seen on WSN world. In general since a certain service, such as data processing and compression, is independent from network protocols, the benefit of these research is acquired without any interference between communication protocol and service. Therefore, in recent years, service area research is becoming also important issue in WSN.

2.2 Body Sensor Networks

As described at 2.1.1, WSN provides various kinds of applications such as monitoring and tracking. Our research targets on certain kind of sensor network and application; Body Sensor Networks (BSN) as known as Body Sensor Networks (BSN), generally aims to monitor human health or detect context of user. According to Wireless Body Area Sensor Networks Signal Processing and Communication Framework [7], the home health monitoring market in 2010 was about $10 billion, and this market is growing about 10 percent at annual. Therefore, it can be said that it is more practical and attractive in this area. In this section, we show some examples of BSN firstly. We then discuss issues and problems of this area.

2.2.1 Applications

Many application of this area is health monitoring. FASH (Fatigue Alerting SHoes) [8] is a typical one. This system aims to prevent elders from falling accidents by detecting their fatigue. To achieve this, FASH uses a pressure sensor embedded in a pair of shoes and notifies the use of "Tiredness Scale", a scale defined in order to show the degree of tiredness. The result shows that pressure sensor has an ability to detect tiredness of user and this notification is siginificantly important to prevent falling accidents. They discovered three differences in pressure value before and after fatigued; (1) decreases in maximum pressure, (2) increases time depending on foot on the ground, (3) transition of pressure in between feet. Based on this observation, FASH detects the fatigue.

MARS: A Muscle Activity Recognition System [9] is also typical system, but MARS is designed as more general than FASH. MARS aims to uniquely distinguish and identify muscle activity. This data is important for many reasons. For instance, the fatigue level of muscle can be calculated based on sports medicine theory and this, which prevents injuries and accidents. Besides, it is useful for precision training and allows athletes to monitor the progress. MARS achieved this by using inertial sensors and gathering the vibration of active muscles. To evaluate MARS, they conduct the experiment; three isolation exercise, including leg extension, standing leg curl and calf raises work, is selected and these exercise use 3 leg muscle groups. Hence, MARS is tring to distinguish these muscles. Then, the result shows that MARS can distinguish muscles with greater than 85% precision.

BodyScope [10] is a platform for activity recognition that enables various kind of applications, such as context-aware and life-logging. Although the implementation cost of wearable sensors on user effects on usability, it is needed that user need to waear multiple sensors for high accurate recognition.

BodyScope focuses on this issue and provides wearable acoustic sensor designed to meet this requirements.This sensor records the sounds in the user's throat area and classify them into user activities, such as eating, drinking, speaking, laughing and coughing. In a small-scale real world experiment, BodyScope was able to identify four activities (eating, drinking, speaking and laughing) at 71.5%.

2.2.2 Issues and Problems

Although many protocols are proposed for traditional WSN (described in 2.1.2, they do not well work in BSN due to the unique features and application requirements. The main two difference between BSN and WSN are listed as follows [11]:

1. Density
   The density of sensor network is an important factor. Since, BSN nodes are typically placed on the human body, or under clothe, the network consists of 10 nodes at most within body area. On the other hand, traditional WSN uses more nodes and deploys wider area. For example, in A macroscope in the redwoods [4], they deployed 40 nodes in whole redwood which height is 70m. However, FASH [8] only uses 2 nodes in users shoes. This makes a difference in the policy of protocol and application design.

2. Data Type
   Most BSN collect continuous data such as acceleration, pressure, vibration and sounds to meet their requirements. Compared to data that is collected by traditional WSN systems, these need to be gathered in high frequency. Traditional WSN collect the data per few seconds-minutes (ex, in A macroscope in the redwoods [4], each data is gathered per 5 minutes). However, BSN need to do per at least tens of milliseconds. As a result, in short term, there is a considerable difference in the amount of data made by each node between WSN and BSN.

There is, of course, more difference between them; such as latency, mobility, flexibility and cost-effeciency. But especially *density* and *data type* are the most important. As described in sec .2.1.2, communication protocol is the main area of WSN research. Since wireless communication is the main factor of energy drain (we will discuss at 2.3), deciding efficient path to host and minimizing number of hop is needed to reduce energy consumption of nodes. Therefore, routing protocol is the key issue and nearly decides the lifetime of network in traditional WSN. However, BSN isn't under this condition; routing is not needed because sensor nodes and the host deploy within human body area and can deliver their packet to the host by one hop in the most case. This means that the demands of efficient routing protocol are not required as the WSN.

We show that the energy consumption problem is hard to be solved by only improving communication protocols. Although BSN nodes force to handle with bigger and more complex data than WSN ones, traditional stategy to reduce energy consumption doesn't work well. To break this condition, another energy saving strategy is urgently required.

One of the efficient strategy is data compression. As described 2.3, the enegy consumption of CPU is much smaller than wireless communication cost. This suggests that it is effective to compress that data and reduce the number of transmission or the length of payload. Although There are some works which focus on this, such as [12], [13], some problems still remain especially in BSN.

2.3 Energy Consumption

In this section, we review the energy consumption problem that is the main issue of WSN research. Firstly, we study the hardware of tiny sensor node and enegy consumption of each part. We then show more specific energy consumption when node sends a packet.

2.3.1 Energy Consumption Model

There are many works which address energy consumption modeling and estimation, such as [14], [3], [15], [16]. It is difficult to estimate or precisely obeserve energy consumption because they have unstable energy sources and each node has a individual difference at level that can't be ignored. In addition, since energy consumption is the most considerable problem of WSN, the result of these works allow researcher and designer to decide a policy of protocol, algorithm and application. Hence, these works are significantly meaningful.

Experimental Evaluation of a WSN Platform Power Consumption [2] is one of them. They conduct the experiments in real WSN platform, such as TMote Sky [17] and telosB [18] mote that are widely used in WSN research, in order to observe energy consumption exactly. According to this work, in general, sensor node has five energy drain modules; CPU Module, Flash Module, Timer Module, LEDs, Wireless Module and they estimate specific energy consumption for each module. Typically, CPU module and Wireless module have low power mode in order to control their energy consumption. They also confirm each mode and result is described at Fig .2.4 and Fig .2.3. Compared to CPU module whose current is 2.6 mA at highest when it commands WHILE loop without using any low power mode, wireless module is outstanding; The current is 8.5 mA at least even when the module uses the lowest power mode. Since other modules are much less than this value, it can be said that wireless module is the main factor of energy drain of wireless sensor nodes. Hence, to reduce wireless module energy consumption, the research of an efficient routing protocol which decreases the number of data transmission has been actively conducted.

2.3.2 Wireless Communication Cost

We then study the specific energy consumption of data transmission in tiny sensor nodes. An Energy Model for Transmission in Telos-based Wireless

Comand	CPU clock 1 MHz	CPU clock 4 MHz
AND	598 µA	2.33 mA
OR	600 µA	2.33 mA
XOR	596 µA	2.33 mA
ADD	600 µA	2.33 mA
MUL	598 µA	2.33 mA
DIV	598 µA	2.33 mA
FOR loop	592 µA	2.32 mA
WHILE loop	660 µA	2.6 mA

Figure 2.3: Instruction execution power consumption [2]

	Consumed Current [mA] (This work)	Consumed Current [mA] (Data sheet [14])	
Radio mode	Radio +CPU	Radio +CPU	Only Radio
Receive/Listen	22.8 mA	21.8mA ÷ 23mA	19.7 mA
TX = 0dBm (31)	21.7 mA	19.5mA ÷ 21mA	17.4 mA
TX = -1dBm (27)	20.3 mA	N/A	16.5 mA
TX = -3dBm (23)	19.0 mA	N/A	15.2 mA
TX = -5dBm (19)	17.3 mA	N/A	13.9 mA
TX = -7dBm (15)	16.3 mA	N/A	12.5 mA
TX = -10dBm (11)	14.9 mA	N/A	11.2 mA
TX = -15dBm (7)	13.6 mA	N/A	9.9 mA
TX = -25dBm (3)	12.1 mA	N/A	8.5 mA
MCU on, Radio off	2.4mA (idle)	1.8mA ÷2.4mA	N/A

Figure 2.4: Radio power consumption [2]

T x power	1 byte	2 byte	4 byte	8 byte	16 byte	28 byte
0dBm	40.6	42.5	46.5	54.2	68.6	90.9
-5dBm	34.0	35.7	38.4	44.5	56.2	73.7
-10dBm	29.1	30.5	32.9	37.7	47.3	61.6
-15dBm	26.2	27.3	29.5	33.7	42.0	54.5
-25dBm	24.1	25.1	27.0	30.8	38.2	49.4

Figure 2.5: The results of energy calculation (μJ) [3]

Sensor Networks [3] estimate the energy consumption when sensor nodes send only 1 byte. The result is shown in Fig .2.5.

According to this, there is a clear relation between packet size (or length) and energy consumption. Therefore, it is effective not only reducing the number of data transmission but also shortening packet payload.

2.4 Summary

In this Chapter, we studied the application and research area of Wireleess Sensor Networks and Body Sensor Networks. Although WSN has significant benefit for monitoring and tracking application, there are a lot of problems especially energy consumption caused by resource constrains of sensor node. Body Sensor Network is one of WSN types and applications, which monitor or detect user's health and context in most case. Because of the density and the type of sensing data, traditional energy efficient strategy, such as routing protocol and time synchronization, is not suitable for BSN. The efficient strategy is data compression to solve the energy consumption problem and to be suitable for BSN.

Chapter 3

Compressive Sensing

In this Chapter, we discuss Compressive Sensing (CS) that is a novel data compression scheme. We firstly introduce theorical basics of CS and then show related works and discuss their problems.

3.1 Compressive Sensing

In this section, we will describe the CS basics. We firstly introduce the abstract idea of CS. The CS basic theory can be divided into encoding and decoding [19], [20], [21], [22] and we then study each theory.

3.1.1 Abstract Idea

It is well known that human looks at a part of the object when he/she recognize it; human has an ability in their brain, which can recover the correct image from incomplete information. For instance, look at following sentence which is passage at the beginning of *Lewis Carroll's Alice's Adventures in Wonderland* [23]:

> The ra*bit-h*le went strai**t on lik* a tu*nel for some way, and then dipped suddenly do*n, so suddenly th*t Al**e had not a

mom*nt to think ab**t sto*pin* hers**f before sh* found herself fal*in* down a very de*p we*l.

Although some characters are replaced intentionally, you could read and understand this sentence. It is because that you (may) know the story of *Alice's Adventures in Wonderland* and you have the common knowledge about English. We call this condition *a priori* knowledge.

And then, we look the same sentence once again.

> The *****-h*** *en* straight on like a t**n*l for some w*y, and then d***ed suddenly down, so suddenly that Alice had not a mom*** to t**k about sto****g herself before she fo**d herself falling down a very deep well.

You may not read or understand this sentence completely even you have *a priori* knowledge. This is simply why that the position of characters which replaced are too wrong. It is almost impossible that recover *rabbit-hole* from ******-h**** even if the human brain is clever enough. We call this constrain *appropriate selection.*

As shown above, human can acquire the whole data from incomplete information based on their *a priori* knowledge and appropriate selection. In CS, each is equivalent to the following things: *a priori* knowledge is *Sparsity* and appropriate selection is *Random Sensing.*

3.1.2 Compressive Sensing Encoding

Sparsity

Encoding depends on sparsity of signals. Sparsity is the idea that when signals are expressed in a convenient basis (such as a fourier or a wavelet basis), nearly all bases have zero or very small values; a few basis have nonzero values.

For example, sounds are sparse signal. Needless to say, sounds are vibration of the air and can be expressed into waveform in time domain such

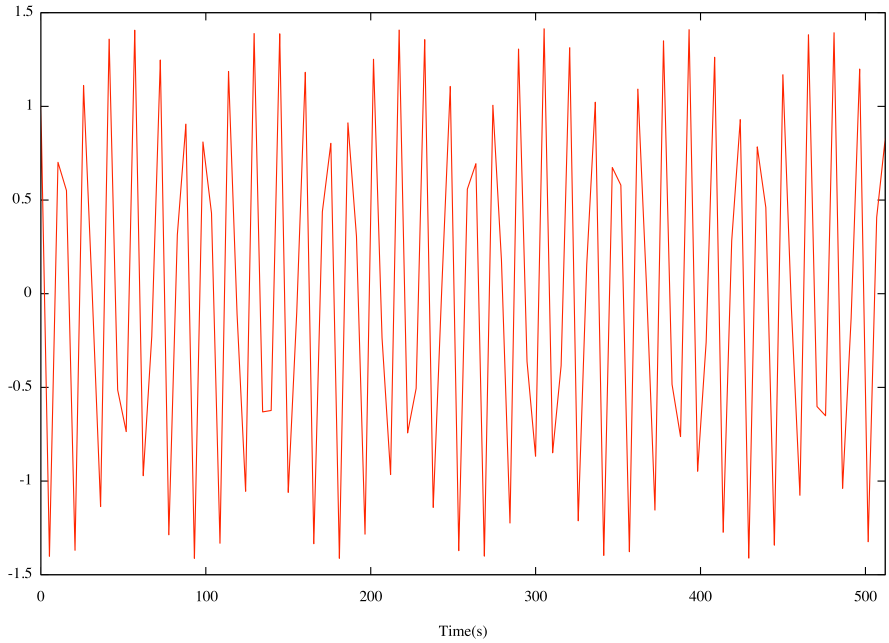

Figure 3.1: Sound waveform

as Fig .3.1. Applying fourier transform sounds also can be expressed in frequency domain and show result in Fig .3.2. It is easy to understand that the sparsity of sound signal. And next example is acceleration of human movement. Fig .3.3 and Fig .3.4 shows that the acceleration of human walking and apply discrete cosine transform in order to analyze spectrum. This also shows the acceleration data is sparse; most coefficient have 0 or nearly 0 value and a few one has meaningful value. As shown in examples, many natural signal or data, such as sounds, acceleration, pressure and image, can be transformed to sparse signal.

We then study the mathematical definition of sparsity. Let $S = [s_1, s_2.....N]$ denotes the N-length original signal, and $\Psi = [\psi_1, \psi_2....\psi_N]$ denotes the convenient basis. We acquire the following equation:

$$S = \sum_{i=1}^{N} \psi_i x_i = \Psi x \tag{3.1}$$

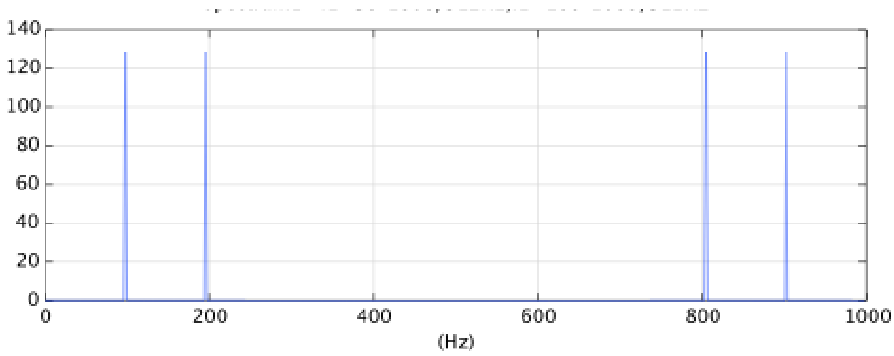

Figure 3.2: Sound representation in frequency domain

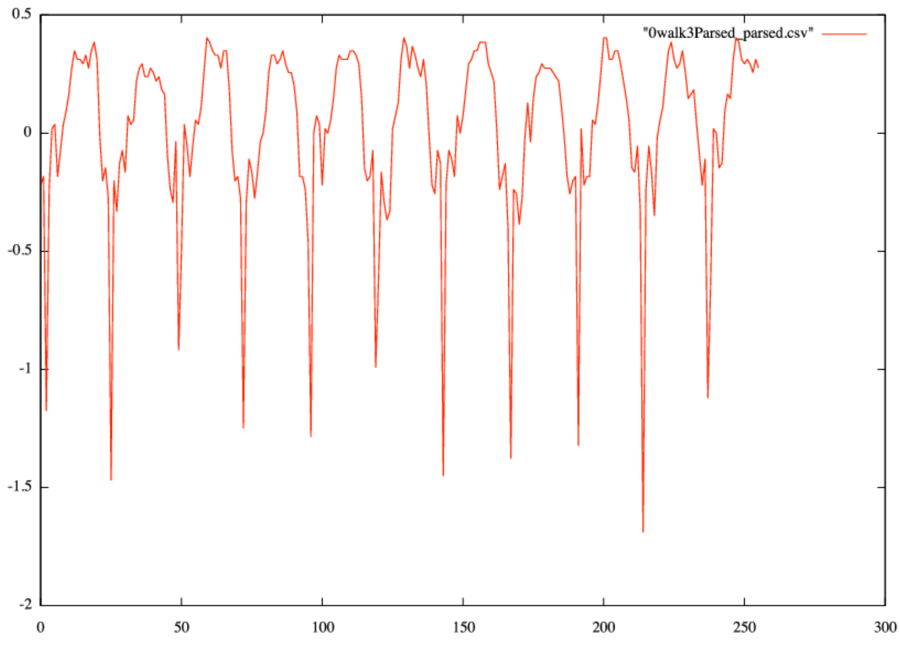

Figure 3.3: Acceleration data of human walking

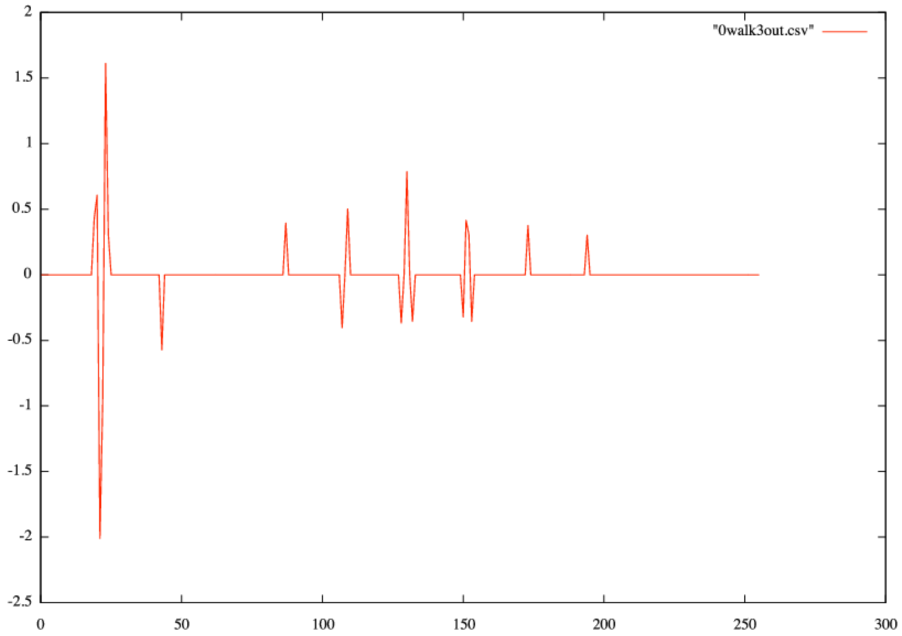

Figure 3.4: Acceleration data representation in frequency domain

where x_i is the coefficient sequence of S (Fig .3.5) when there are $k-$nonzero entries of x_i, S is called $k-$sparse.

Random Sensing Problem

We then study how to compress and reduce the sparse signal. We consider a general linear measurement process between sparse signal x (N-length) and $M \times N$ ($M < N$) measurement matrix $\Phi = \{\phi_j\}_{j=1}^{M}$ defined with independent identically distributed ($i.i.d.$), such as Gaussian or Bernoulli distribution.

$$y = \Phi x = \Phi \Psi S \tag{3.2}$$

y is $M-$length vector, thus, we succeeded in reducing $N - M$ values (Fig .3.6).

Considering two matrices Φ and Ψ which are acquired from these formulas (3.1 and 3.2), we introduce the condition to improve exact recovery

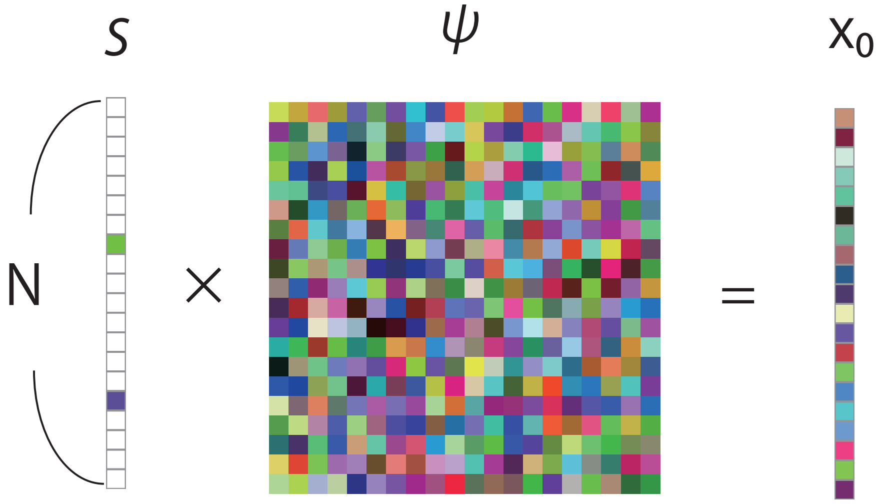

Figure 3.5: Sparse Representation

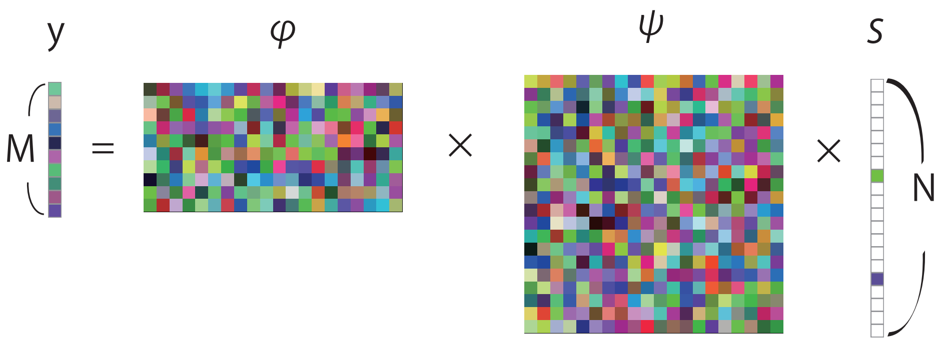

Figure 3.6: Random Sensing

accuracy, called *incoherence*. Comparing the convenient basis Ψ and the measurement matrix Φ, if the relation between these is low, the condition is called *incoherence*. Mathematically speaking, this relation is calculated from the correlation of each matrix elements; when Ψ and Φ contain correlated elements, the coherence is large.

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k,j \leq n} \| \langle \phi_k, \psi_j \rangle \| \tag{3.3}$$

To minimize the above formula, CS can achive high reliability.

3.1.3 Compressive Sensing Decoding

$L1-$norm minimization

Next, we study how to recover original signal S from compressed signal y and measurement matrix Φ. We can recover S by solving $L0-$ norm minimization, which minimizing $0-$norm subject to linear equality constrains ($y = \Phi x$):

$$\hat{x_0} = \arg\min_x \|x\|_0 \text{ subj. to } y = \Phi x \tag{3.4}$$

Superior point of this idea is that if $M > k$m, $L0-$norm minimization can always recover S without the condition of Φ. Although $L0-$norm minimization can achieve high reliability, this is $NP - hard$ problem [24]. Thus, we hove to consider realistic way. In general CS, $L1-$norm minimization is used to solve this problem. Formula is as follow:

$$\hat{x_1} = \arg\min_x \|x\|_1 \text{ subj. to } y = \Phi x \tag{3.5}$$

Since $L1-$norm minimization can be easily solved as liner programing, this is served as alternative. Needless to say, it is nonsense that $L1-$norm minimization can't recover original signal S. However, surprisingly, $L1-$norm recover original signal S with high probability when S is sparse [19], [20]. Therefore, $L1-$norm minimization is generally used in CS.

Restricted Isometry Property (RIP)

$\hat{x_1}$ equals to x with high probability under particular circumstance, which is called *restricted isometry property*(RIP) [24]. For $1 \leq K \leq N$ define the isometry constant δ_S of Φ

$$(1 - \delta_N)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_N)\|x\|_2^2 \tag{3.6}$$

holds for all $K-$sparse vector x. If K exists when $\delta_2 K < \sqrt{2} - 1$ holds for Φ, $L1-$norm minimization can recover for all $K-$sparse vector x with high probability.

3.2 Compressive Sensing in WSN

In this section, we review related works that use CS in WSN and discuss their problems.

3.2.1 Related Works

Since CS can encode and reduce the data with extra low computation costs, It is very suitable for tiny sensor node which has a few resources. Thus, there are many works that tries to apply CS to WSN to reduce energy consumption, signal processing and bandwidth.

In-situ Soil Moisture Sensing [25] focuses on monitoring soil moisture using a wireless underground sensor network. To reduce cost and prolong lifetime, this work applies CS based sensing scheduling algorithm. They designed two matrices, measurement matrix Φ and representation basis Ψ, to address these issues. The design policy of these is following:

1. Measurement Matrix
   Measurement matrix corresponds to sensing scheduling; if the column which position is (m, n) $(1 \leq m \leq M, 1 \leq n \leq N)$ has 1, the $m-$th measurement is taken at $n-$time. Periodic and random policy was used

for definition of this matrix. This is called Measurement scheduling Matrix.

2. Representation Basis
   Representation basis is designed to meet two requirements: (1) It guarantees to convert natural signal to sparse representation. (2) It meets the incoherence between Measurement matrix Ψ. To achieve these requirements, they used the characteristics of soil moisture signals; The soil moisture change relatively smooth except at the onset of a rainfall. Based on this observation, they designed representation basis Ψ called *different matrix*.

Numerical experiments are performed on both real and simulated data. The result shows that these approach and design is extremely effective.

Compressive Oversampling [26] is also one of CS applications, which use a randomness of CS and add a redundancy and robustness to data transmission. Since measurement matrix is normaly, except for a certain application such as above, defined with *i,i,d*, the worth of each measurement value is equal. Focusing on this point, this work designed a measurement matrix Φ to afford a redundancy to wireless data transmission by adding some extra rows a to Φ, which is called Compressive Sensing Erasure Coding (CSEC). It is possible to recover exactly on decoder even if some packets are lost via wireless transmission within *RIP* hold. Simulation based experiments are performed and results shows that CSEC can achieve robust data transmission. This work suggests that CS has applicability not only to energy consumption but also to data transmission.

Compressive Sensing Method for Human Activity Sensing [27] proposes the CS-based framework for human activity monitoring in mobile devices, such as mobile phone and smartphone. Since mobile device, suceas mobile phone and smartphone, has resource constrains as same as wireless sensor nodes, it is significantly meaningful to reduce communication costs and power consumption. Normally, 3-axis accelerations are used for human activity recognition application; the Euclidean norm defined as $\sqrt{Acc_x^2 + Acc_y^2 + Acc_z^2}$ is more convenient for this kind of applications. Thus, original signal x is set to the Euclidean norm values. The experiment is performed to evaluate the effect of CS on energy consumption. The result shows that CS can reduce energy consumption by 16% compared to ZIP scheme.

3.2.2 Issues

These related works are remarkable and interesting. However, it is insufficient or do not work well in our target environment, Body Sensor Networks. As studied in Chap .2, there are some charateristics in BSN: density and data type. Especially, data type is very important in CS encoding because of it depends on signal sparsity and *a priori* knowledge. In BSN, sensor data changes dynamically in many application, such as human movement detection. Existing works use only static property about CS. For instance, *In-situ Soil Moisture Sensing* [25] adapt CS parameters to meet their requirements. However, it is adequate to set static value because the signal characteristics of soil moisture is smooth, much differs from BSN data. Hence, these static design will not work well in BSN. We proof this issue in next chapter 4.

3.3 Summary

In this section, we study the basics of Compressive Sensing (CS), a novel data compression/representation scheme which can express the signal under the Nyquist. CS encoding depends on a convenient basis Ψ and measurement matrix Φ. We then surveyed the related works which focus on CS applications in WSN.

Chapter 4

Case Study

In this chapter, we performed some experiments in real and mathematical environment to confirm the efficiency of CS in terms of energy consumption and decoding reliability. We then discuss the suggestion from the result of these experiments and the problems to address in this work.

4.1 Energy Consumption Analysis

In this section, we analyze theoretical and actual lower bound of compression ratio that is defined by measurement matrix size Φ. Firstly, we study theoretical lower bound and then show real world experiments results.

4.1.1 Theoritical Analysis

As described at 2.3, the main factor of energy drain in tiny sensor node is radio communication module. However, there is a fact that CPU also consumes a little energy. Thus, too much computation also causes shortening of lifetime. Compared to previous data compression scheme, CS has weird characteristics about compression ratio. Since encoding process is executed as a multiplying between vector (sparse signal S) and matrix (measurement matrix Φ) and larger matrix size M means that lower compression efficiency and ratio, makes larger computation and transmission costs in terms of energy consumption. Addressing this point, we study the numerical lower bound.

Let E_p denotes the power consumption of a sigle multiplying operation and E_t denotes the power consumption of transmitting one packet. This time, we choose a fourie transform to represent sparse signals. Generally $O(NlogN)$-time is needed to compute the Fast Fourier Transform(FFT) and that of a general linear measurement process between $n-$length vector and $M \times N$matrix is $O(MlogN)$. Thus, An extra computation energy to execute CS is $E_p \times (NlogN + logMN)$ Many techniques can be used to reduce computation time, but to simplify the problem, we consider as above to simplify the problem even there are many techniques that can be used to reduce computation time. Since $M - N$ length data can be reduced using CS, a sensor node can save $E_t \times (M - N)$ energy. Therefore we obtain the following formula:

$$E_p \times \{O(NlogN) + O(logMN)\} < E_t \times (M - N) \tag{4.1}$$

CS can save energy if the formula4.1 is true. Since $E_p \ll E_t$ in general case [2], depending on the configuration (especially M or N), CS will far be effective in many cases.

4.1.2 Real World Experiment

Next, we perform the real-world experiment to observe the impact of CS on energy consumption. We implement CS encoding algorithm on 4 SunSPOTs which are equipped with temperature, light and 3-axis acceleration sensor and controlled by Java progming [28]. They sense and send or compress acceleration data at every $30ms$.

The results of each node are shown at 4.1-4.4. CS can achieve the improvement of lifetime by 38.8% in the average case. It is clear that CS significantly improves their lifetime despite of making extra computation costs.

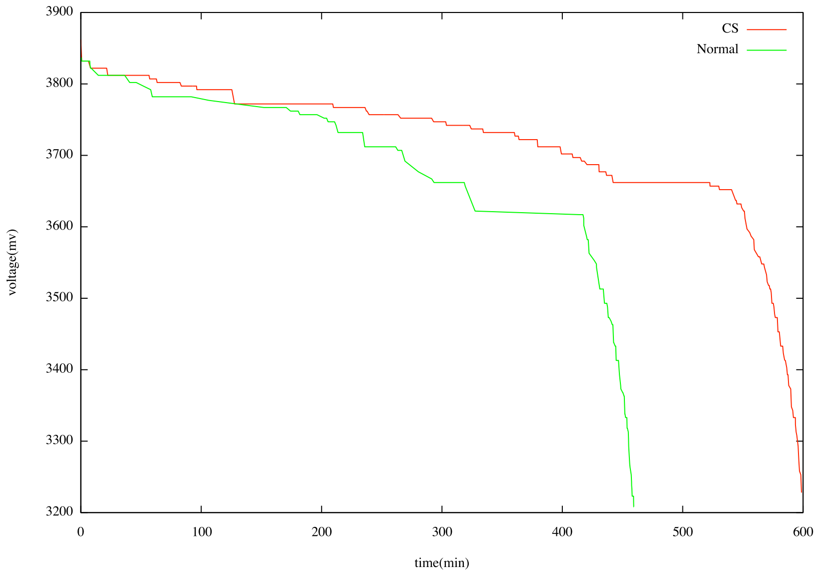

Figure 4.1: Experiments result1: comparison between CS and Normal in terms of energy consumption

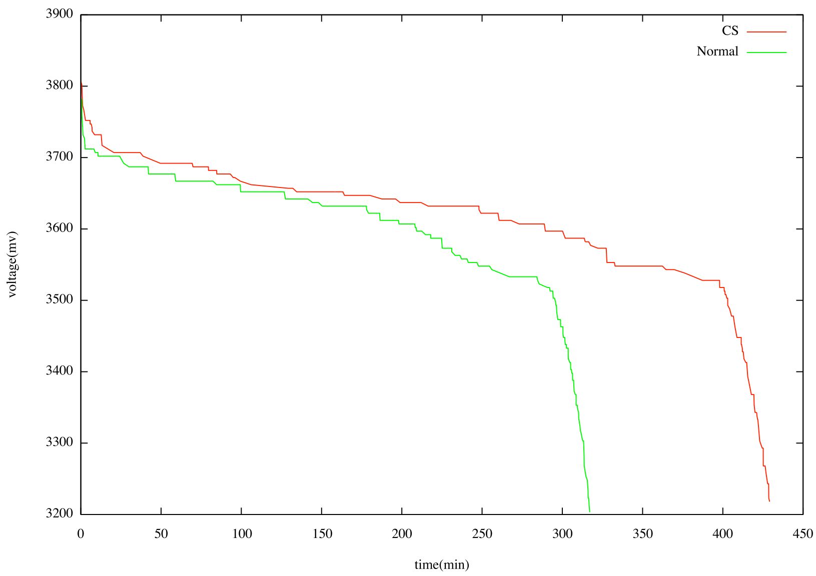

Figure 4.2: Experiments result2: comparison between CS and Normal in terms of energy consumption

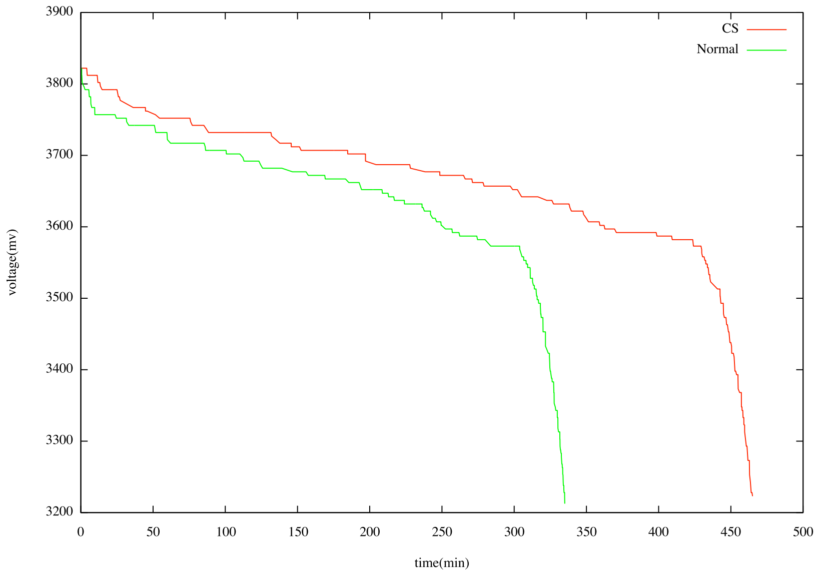

Figure 4.3: Experiments result3: comparison between CS and Normal in terms of energy consumption

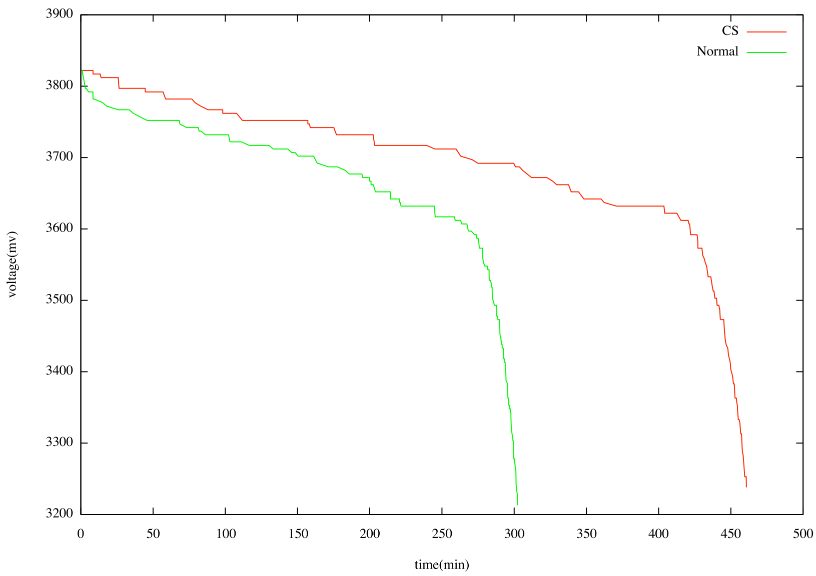

Figure 4.4: Experiments result4: comparison between CS and Normal in terms of energy consumption

Figure 4.5: Experiments environment

4.2 Decoding Reliability

In this section, we study the reliability of CS in real BSN application. As described at 3.1.3, sparse signal S can be always recovered within each condition meeting the RIP. We confirm this theory is also available in real sensor data and, if true, observe the behavior of CS in real environment.

4.2.1 Environment

We decided the experiments environment assuming human movement detection application. We put iris mote [29], which is also widely used in WSN reasearch, on user's back as shown in picture 4.5. Then, they walk and run. The sensor nodes sense and send these acceleration data at every 10 ms. We choose Discrete Cosine domain as a convenient basis Ψ, gaussian distributed matrix as a measurement matrix and CoSAMP [30] as a decoding algorithm, which is pupularly used some CS applications.

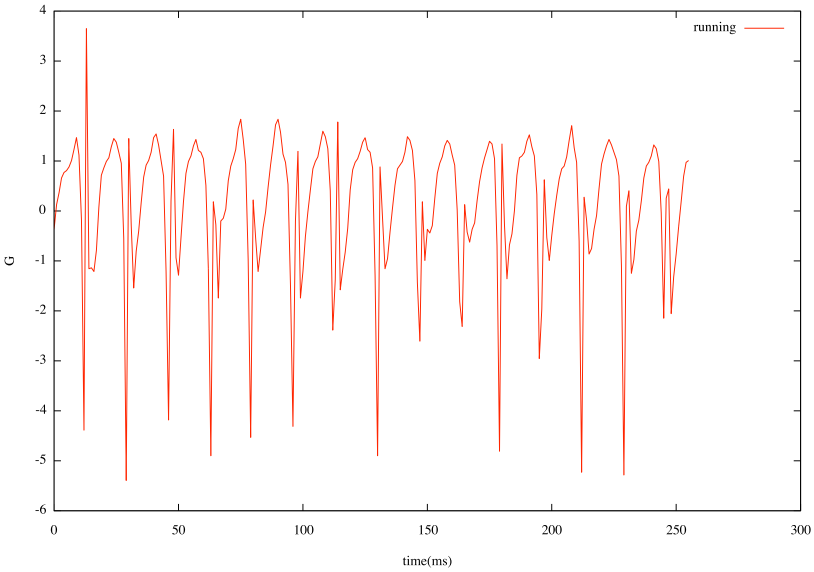

Figure 4.6: Experiments Sample data: the raw acceleration data of running

4.2.2 Result

The result is shown in 4.6 - 4.10. Fig .4.8 and Fig .4.6 illustrate the example data, including raw acceleration data and sparse frequency data before applying DCT. It is clear that there are big differences between these values on both the acceleration and the frequency. For instance, the sparsity of human running acceleration data k, shown at Fig .4.7, is 45. On the other hand, the it of human walking acceleration data k, shown at Fig .4.9, is 11. This difference affects CS parameters such as matrix length M in terms of reliability of decoding. Fig .4.10 shows the relation between exact recovery ratio (ER) and measurement matrix size M. It is simply caused by the difference of signal characteristics and sparsity.

4.3 Matrix Collision Problem

We then discuss the suggestion obtained by the result. To maximize the performance, it is desirable to optimize measurement matrix size for the signals.

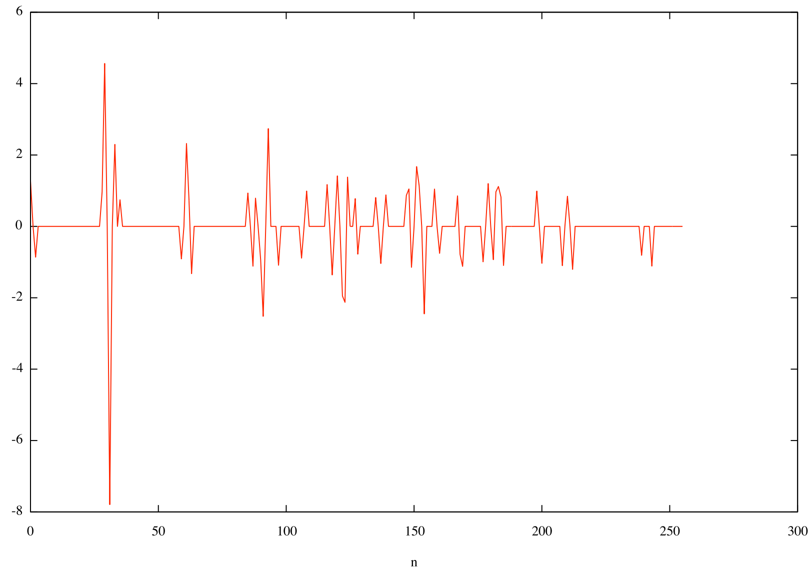

Figure 4.7: Experiments Sample Data: running data in DCT domain

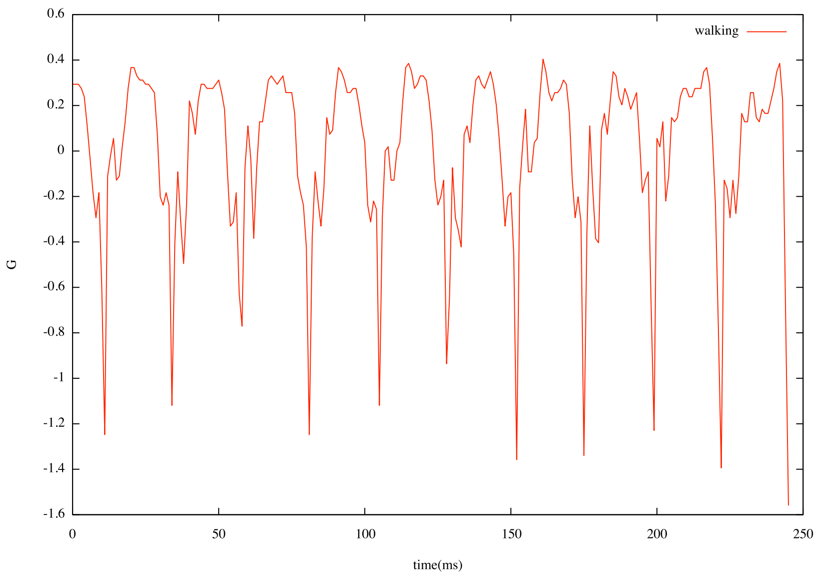

Figure 4.8: Experiments Sample data: the raw acceleration data of walking

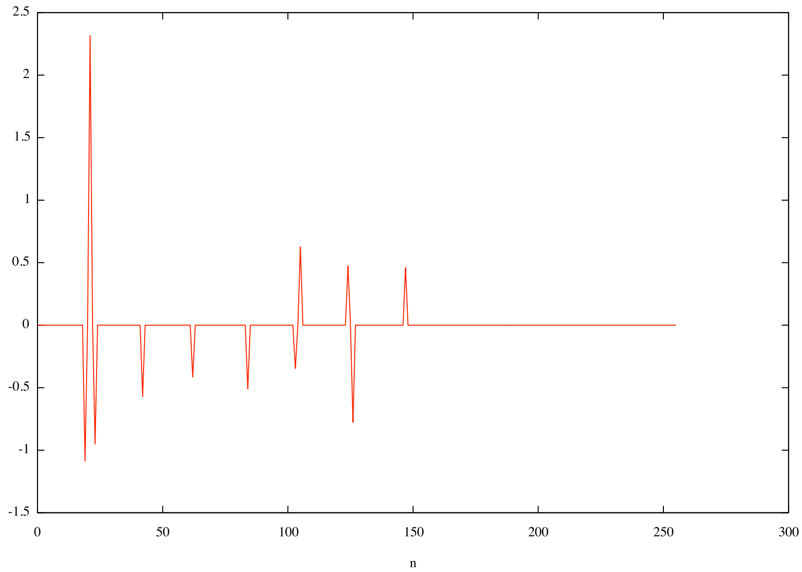

Figure 4.9: Experiments Sample Data: walking data in DCT domain

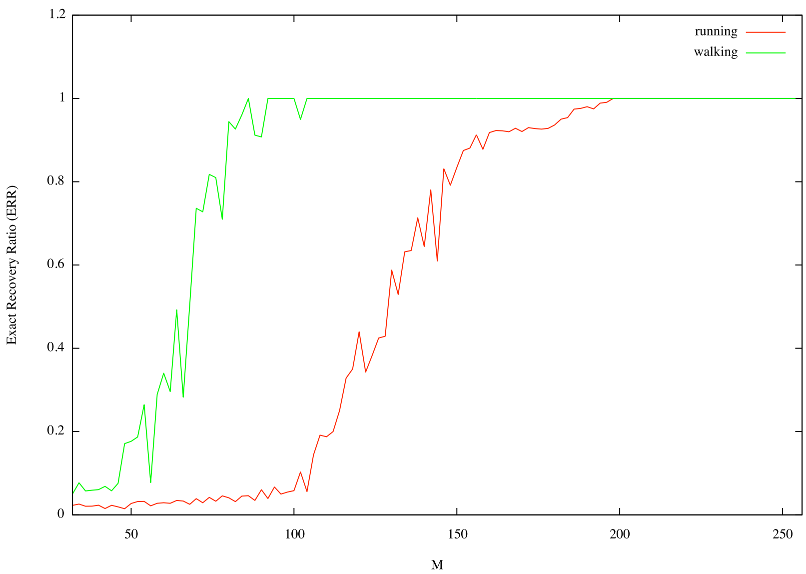

Figure 4.10: Experiments result: Comparison ERR between walking and running

Since larger M length of measurement matrix Φ makes lower compression ratio and high computation costs, it is desirable to set appropriate M length. When M length is minimum, the performance is maximum in terms of energy consumption.

Described in chapter 3.1, when M length of measurement matrix Φ is minimum in a range of meeting RIP [19], CS gets the highest compression ratio and most effective. However, real sensor value changes dynamically, which mean the sparsity of signals and M that depends on it are not static values. As shown section 4.2, there are big difference between the acceleration values and the sparsity of each state. This difference affects CS encoding and decoding. Fig .4.10 shows it; When user is walking, measument matrix size M is adequate to 100 in this case, otherwise, it is needed to set 200. We call this situation *Matrix Collision Problem*. This makes it difficult to set the length of M uniquely. Much larger M makes inefficiency and much smaller M makes inaccuracy, both cases are not desirable. We address this *dilemma* as a problem to solve in this research.

4.4 Summary

In this chapter, we performed some experiments in real and athematical or situmulation environment to confirm and study the performance of CS in terms of energy consumption. These results suggest that the *Matrix-size Collision Problem* decreases CS performance both in terms of energy consumption and reliability. In this thesis, to solve this problem, we propose Pattern-based Optimization Algorithm (PMOA).

Chapter 5

Approach and Design

In this Chapter, we firstly propose our approach to attack matrix size collision problem defined in chapter 4. We then propose Pattern-based Matrix size Optimization Algorithm (PMOA) that provides high efficiency and reliability in CS.

5.1 Approach

This thesis proposes Pattern-based Matrix size Optimization Algorithm (PMOA) that optimizes measurement matrix size M dynamically. This algorithm achieves high efficiency in compression data and high reliability in CS decoding. PMOA optimizes CS parameter, measurement matrix size M, along application state in order to improve the energy consumption and decoding reliability. Appropriate measurement matrix size M_{opt} leverage to achieve high efficiency in data compression. As shown in chapter 4, there is a big difference between sensor value and these sparsity of each application state especially in BSN applications. Fig 4.10 shows the impact of this difference on CS. Although, in this case, M size is sufficient at 100 to achieve almost 100% exact recovery ratio when user is walking, it is insufficient when user is running. In order to solve this *matrix size collision problem* defined in 4.3, PMOA optimize the measurement matrix size M and achieve high

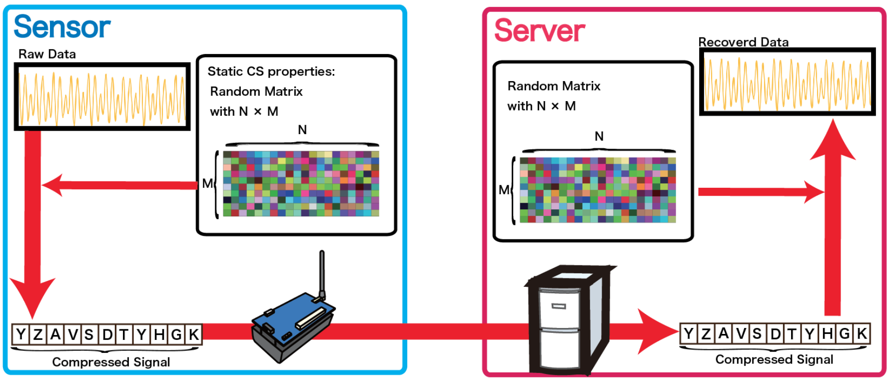

Figure 5.1: Overview of classic CS process

efficiency in data transmission and high reliability in recovering signal.

Existing CS encoding algorithm uses static properties including measurent matrix and convenient basis. Fig. 5.1 illustrates how sensor nodes encode data to CS measurement. Whenever user state changes, such as walking, running and jumping, it uses the static measurement matrix with fixed size M. Hence, measurement matrix size M is forced to fix to apply the highest sparsity state; if the matrix size M is 200 along the highest sparsity state, encoded value length is always 200 despite of it is adequate that matrix size is 100 at other lower sparsity state.

Unlike existing CS encoding scheme, in PMOA, encoder can selects some kind of matrix size M_{opt} that is calculated beforehand. This approach suits our target environment, where sensor data forms waveform and patterns. Fig. 5.2 illustrates a preparatory process of PMOA. At first, PMOA collects sensor data patterns along application scenario and calculate appropriate matrix size M_{opt}.While application is running, CS encoder detect application state based on the sparsity and select corresponding matrix size (Fig. 5.3).

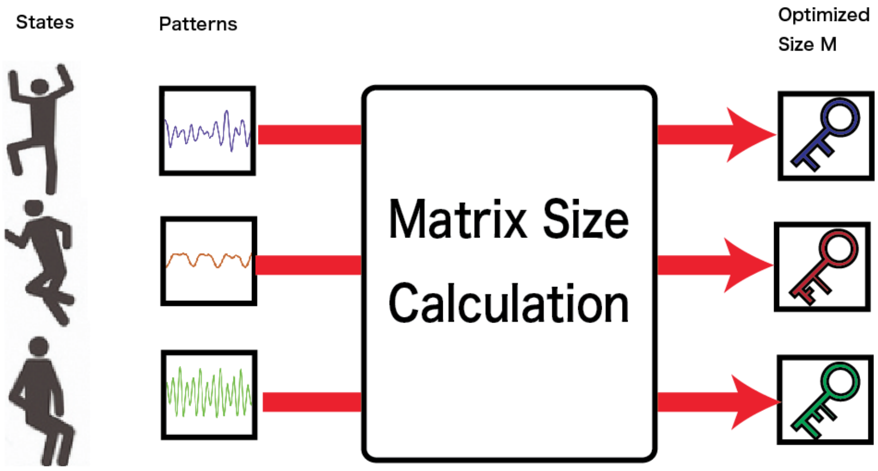

Figure 5.2: Preparatory process of PMOA

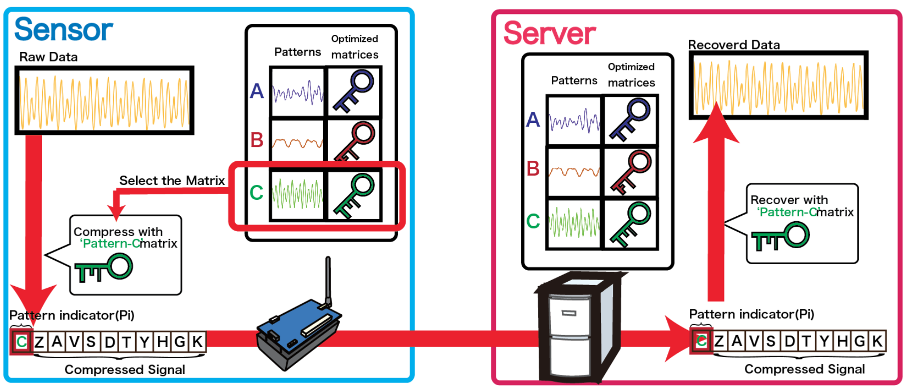

Figure 5.3: Overview of PMOA

5.2 Design of PMOA

In this section, we show the specific algorithm of PMOA. We firstly describe how to calculate appropriate measurement matrix size M and then show the design of it. Finally, we discuss the validity and effectiveness of our approach, PMOA.

5.2.1 Matrix Size Calculation

PMOA collects sensor data along specific application scenario, including human movement detection and health monitoring, and calculate appropriate matrix size M for each pattern beforehand. This calculation phase decides the balance between efficiency and reliability. For instance, we assume *human movement detection* application which detects human movement from 3 states: walking, running and jumping. Firstly, we put sensors on user's body and collect the acceleration data for each pattern We then divide each raw data into *convenient time slot* N (corresponds to original signal length) and apply transformation algorithm which expresses in frequency domain, such as fourier transform and discrete cosine transform, to it. After this process, we get some sample sparse signals for each pattern. We then apply CS encoding, random matrix projection, and decoding, $L-1$ norm minimization to these data with various value of M, which range is within $1-N$. Thus, we can acquire the relation between matrix size M and exact recovery ratio (ERR) for each pattern. User set the appropriate matrix size M for each pattern based on this result; if application needs high reliability, M is set to the value which meets the 100% ERR, otherwise, application does not require highest reliability, almost 70%, it is set to the value which adequate to reach this ERR. Since higher reliability needs larger M_{opt} and this makes more extra data transmission and processing costs, there is a trade-off between reliability and efficiency. Thus, application designer can modify this value to meet their requirements.

5.2.2 Measurement Matrix Design

We discuss a measurement matrix design in order to suit for our architecture. In previous CS scheme, Gaussian or Bernoulli distribution is used for defining of the measurement matrix Φ. Although these distribution guarantee the randomness of matrix, which is needed to achieve high incoherent between a convenient basis, there are deviation of each entries. This is caused by the lack of random processes. We perform a simple experiment to proof this. We conducted a *Bernoulli process* 100 - 100000 times using Mersenne Twister which is popular random generator [31] and observe the correlativity between the number of process and the probability of deviation. *Bernoulli process* provides 2 states, 0 or 1, in a certain probability, thus, it is easily calculated the theoretical value and the error ratio. Fig .5.4 shows that the result and it is clear that less processes make high error ratio. Since the number of measurement matrix entries is at most 256×256, this means that it is difficult to achieve highly precise randomness. Besides, in our approach, it is rare to use whole matrix; CS encoder uses a measurement matrix with various optimized size M_{opt}, which means CS encoder uses a part of the matrix. Therefore, there are no guarantee that each part of matrix have same randomness.

To achieve high incoherent between a measurement matrix Φ and a convenient basis Ψ with high reliability in decoding, it is needed to adjust this randomness in order to keep fairness among each part of matrix. We select the *Balanced Bernoulli Matrix* to address this issue. This matrix is adjusted that the sum of each row is 0. Since each row has the same value and same pseudo randomness, fairness between each part of matrix is maintained in any M_{opt} numbers. Specific implementation of this is described at Chap. 6.

5.2.3 Discussion

We discuss about the validity and effectiveness of our approach. At first glance, our approach is too simple and not smart. But we believe that this

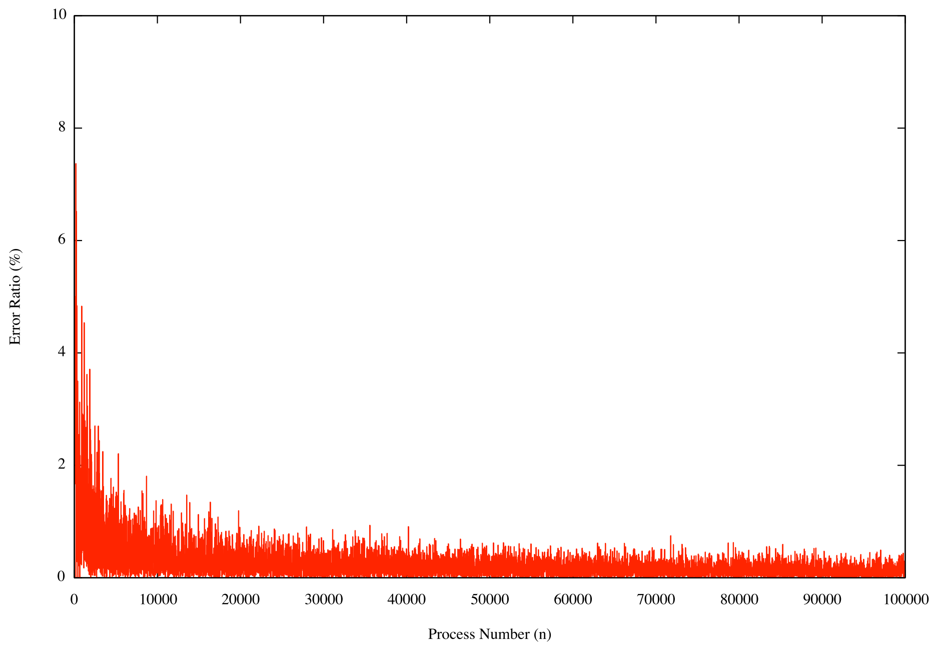

Figure 5.4: Performance of Mersenne Twister

is the best compromise between CS effectiveness and resource constrain of tiny sensor nodes. If there are no constrains about sensor node resources, we could improve our approach and effectiveness. Using existing works and algorithms, such as [32], we can calculate and modify measurement matrices M to achieve more incoherence for each pattern. However, this calculation and implementation makes huge overhead both on memory and CPU. Firstly, it is impossible to calculate this optimization for a sensing signal in sensor node because they have only low power CPU and this calculation is very complex process. Even if pattern classification and calculation this for each of it is performed beforehand, like our approach, implementation of these matrices makes huge burden on their little memory.

On the otherhand, our approach uses only single matrix and optimize the size of it. Since calculation of this size is performed beforehand, sensor nodes only detect the sparsity of sensed signal and select from static values, matrix size M_{opt}. Compared to previous CS scheme, our approach make very little overhead on sensor nodes, however, our approach can improve CS efficiency and reliability.

5.3 Summary

In this chapter, we show and discuss the approach and design of PMOA. PMOA optimize measurement matrix size M along application state based on the sensor data pattern which is collected beforehand. We also study that the measurement matrix design in order to keep the fair randomness among optimized matrix size M_{opt}. Since there are resource constrains in tiny sensor nodes, complex algorithm isn't suit for it. Although PMOA is too simple, therefore, our approach is the best compromise.

Chapter 6

Implementation

In this Chapter, we describe the implementation of PMOA. First, we discuss the platform we used to implement the proposed algorithm on. Second, we explain the system overview. Finally, we describe the implementation of PMOA including measurement matrix, convenient basis and decoding algorithms.

6.1 Implementation Platform

We choose to implement proposed algorithm on SunSPOT [28], [33] which are widely used in the area of WSNs. Table 6.1 shows the specification

Table 6.1: SunSpot spec description

	SunSpot
CPU	ARM AT91SAM9G20
CPU Type	32bit
Clock Speed	400MHz
Flash Memory	8MB
RAM	1MB
Radio Chip	TI CC2420
Battery	770 mAh

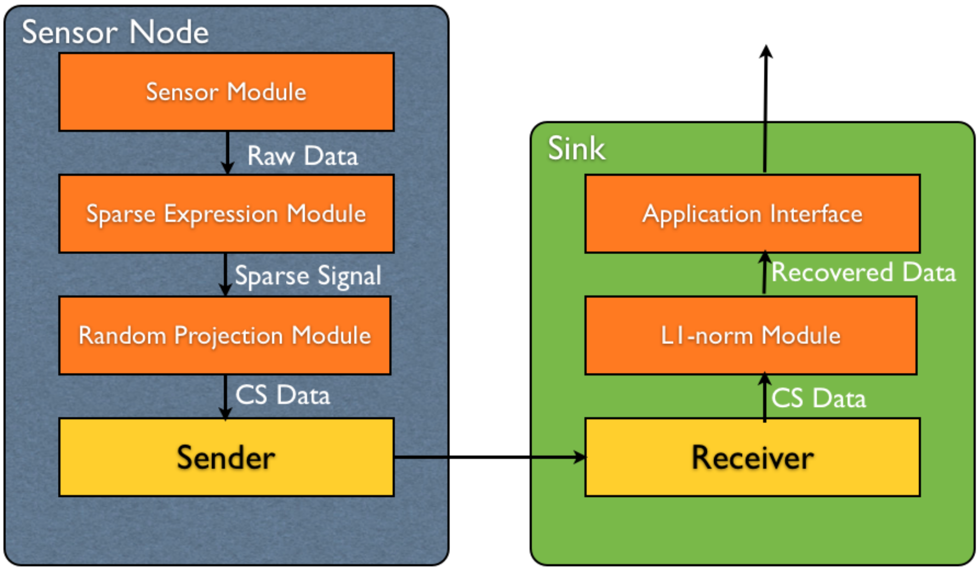

Figure 6.1: System Architecture of PMOA

of SunSPOT. We use Java language for sensor node programming because SunSPOT is controlled by java and C++ for decoding and matrix calculation.

6.2 System Overview

Fig. 6.1 illustrates the overall system architecture of proposed algorithm.

We show the description of PMOA as follows: Encoder consists of 4 modules, including Sensor, Sparse expression, Random projection and Send module, and 3 queues connect between each modules and absorb processing time difference of each module (Fig .6.2). Sensor module such as acceleration, gyro or pressure sensor, collects a data and pushes it to the FIFO queue. When the number of storing data is equal to original signal length N, sparse expression module that convert raw signal to sparse signal and count sparsity k, calls the queue to pop data. Sparse expression module get these data corresponding to x and apply the sparse represent algorithm. The result

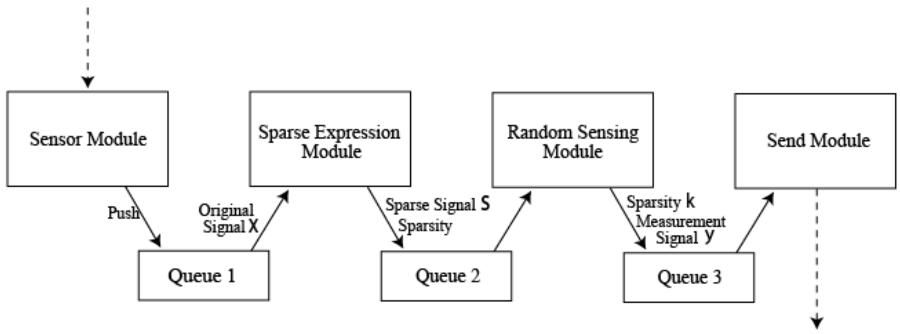

Figure 6.2: Encoder Implementation

of this, including sparse signal S and sparsity data k, is also pushed into second FIFO queue. Random Sensing Module pop the sparse signal S from this queue and sparsity data k. Based on sparsity k, this module select the matrix size M_{opt} which is calculated beforehand and apply random projection to sparse signal S. Then, it pushes third queue. Finally, send module pop the data from third queue and send it to host with sparsity data k.

Decoder has almost same architecture design as encoder but this side is more simple than encoder; it consists of 3 modules, including Receive module, $L-1$ norm minimization module and application interface, and 2 queues (Fig .6.3). Receive module gets the measurement signal y and sparsity k from each sensor nodes and checks these data. If there are no errors, this module pushes the data into the queue, otherwise, drop it. Solver module is a decompression module that applies $l-1$ norm minimization to sparse signal S. This module has also installed *a pattern information*, M_{opt} length, and based on this information, solves *linear programming* problem, $l-1$ norm minimization. Finally, the result is pushed into the second queue and pop to application interface that sends it to application via UDP.

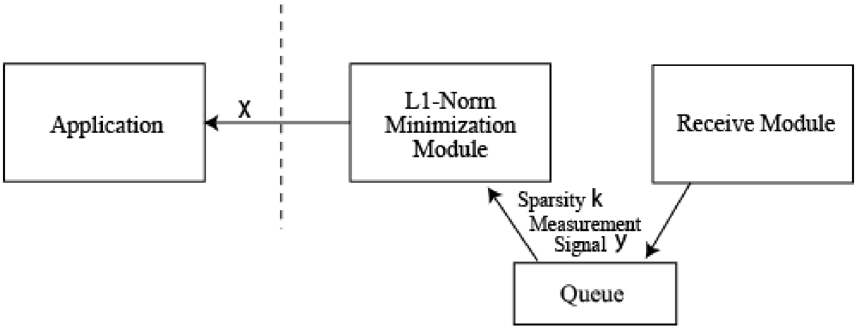

Figure 6.3: Decoder Implementation

6.3 Implementation of PMOA

In this section, we show the specific implementation of PMOA both side, encoder and decoder.

6.3.1 Encoder

We show the implementation of an important property of CS encoding; Convenient Basis and Measurement Matrix.

Convenient Basis

In this work, in order to express original raw signal to sparse signal, we select 1-dimension discrete cosine transform (DCT):

$$X_k = \frac{1}{2}x_0 + \sum_{n=0}^{N-1} \cos \frac{\pi}{N}(n + \frac{1}{2})k \tag{6.1}$$

DCT is known as the algorithm that transforms a signal to *a linear combination* with the cos basis, which means that the sum of a frequency and a amplitude. Unlike the Fast Fourier Transform (FFT), also well known as a signal transform algorithm and it express the signal in frequency domain with real and complex number, DCT can express the signal in frequency domain with only real number. In programming language, complex number is treated as the array of two number; imaginary part and real part. Thus, handling with complex number makes unnecessary costs both on CPU and memory. From this point of view, DCT is a reasonable selection as a convenient basis.

In general, it is needed to apply *window function* to a raw signal before performing such algorithms, FFT and DCT, in order to remove *side-lobe*. We select the *Hamming Window* which is one of the most widely used window function:

$$\omega(x) = 0.54 - 0.46 \cos 2\pi x, \text{if } 0 \leq x \leq 1 \tag{6.2}$$

These two algorithms are implemented on sparse expression module.

Measurement Matrix

As described 5.2.2, we select the *Balanced Bernoulli Matrix* as a measurement matrix. Entries of this matrix is, basically, defined as the result of *Bernoulli process*, a stochastic process that takes only two values with *independent probability*, like coin-tos. Bernoulli process is formalized in the language of probability space. Formal definition is following as [34]:

Probability space (Ω, P_r) is given as a random sequence in $0, 1$.

$$P_r = p, 1 - p^{\mathbb{N}} \tag{6.3}$$

$$P_r([\omega_1, \omega_2, ...\omega_n]) = p^k(1-p)^{n-k} \tag{6.4}$$

where k is the number of times 0 *appears.*

$$P_r(X_1 = \omega_1, X_2 = \omega_2, ...X_n = \omega_n) = p^k(1-p)^{n-k} \tag{6.5}$$

where X_i is a random variable.

Based on process above, we implemented the *Balanced Bernoulli Matrix*. We used the Boost C++ libraries [35], which are popular and powerful libraries, in order to use more accurate random generator, *Mersenne Twister*. Fig 6.4 is the abstract code of this algorithm. Firstly, we repeat the Bernoulli process N times, where N is original signal length, and store each binary value, as a result, to double array. We then check the number of each value in the array and take the balance so that the sum of each row to be 0. This balance algorithm is simple; Using Mersenne Twister, we get the random value r in the range of 0 and N-1, and check the r-th value of the array. If this value is a larger one, we replace it to a smaller one and repeat this until the sum of row is 0. This balance algorithm is performed for each row.

6.3.2 Decoder

On decoder side, the main module is Solver module which performs $l-1$ norm minimization algorithm to compressed measurement y and recover sparse signal S from it. In general, this module is the key of CS because the reliability of it depends on this algorithm. However, our work addresses on encoder problem in order to reduce cost both on computation and data transmission from tiny sensor nodes. Thus, we used existing algorithm, except for dynamic setting of measurement matrix measurement.

$L-1$ norm minimization

We used 4 algorithms, which are provided by KL1p a portable C++ library for Compressed Sensing [36], for signal recovery in order to verify the effectiveness in many algorithms. 4 algorithms are following:

1. Compressive Sampling Matching Pursuit (CoSaMP) [30]
   CoSaMP is a fast signal recovery algorithm where the running time is just $O(N \log^2(N))$, where N is the length of the signal, for many cases

```
#define N 256

double balanced_bernoulli_matrix[N][N];
random_generator MT;
int balance = 0;

for(int i=0; i++; i<N){
  for(int j=0; j++; j<N){
      if(MT.getRandomValue(distribution=bernoulli, probability=0.5))
        balanced_bernoulli_matrix = 1;
      else
        balanced_bernoulli_matrix = −1
  }

  while(sum(balanced_bernoulli_matrix[i]!=0)){

    if(sum(balanced_bernoulli_matrix[i] > 0))
        balance = 1;
    else
        balance = −1;

    int r = MT.getRandomValue(distribution=uniform_int, from=0, to=N);

    if(balanced_bernoulli_matrix[i][r]==balance)
      balanced_bernoulli_matrix *= −1;

  }

}
```

Figure 6.4: Balanced Bernoulli Matrix Algorithm

of interest with as the same guarantees as the best optimization based approaches.

2. Regularized Orthogonal Matching Pursuit (ROMP) [37]
   ROMP is also a signal recovery algorithm with the strong guarantees of the convex programming method. This method recovers a signal x with k nonzero values from its inaccurate measurements y in at most k iterations, where each iteration amounts to solve a least squares problem. The noise level of the recovery is proportional to $\sqrt{\log n}||e||_2$, where e is error vector.

3. Subspace Pursuit [38]
   Subspace Pursuit is a signal recover algorithm that has two characteristics: low computational complexity, comparable to orthogonal matching pursuit techniques, and reconstruction accuracy of the same order as that of LP optimization method.

4. Smoothed L0 (SL0) [39]
   Unlike other algorithms, this algorithm minimizes $L0$ norm directly. SL0 is about two to three orders of magnitude faster than the state-of-art interior-point LP solvers with providing the same (or better) accuracy.

We use these algorithms and evaluate our approach in Chapter 7.

6.4 Summary

In this chapter, we describe the implementation of PMOA. We firstly show the platform and then the description of each module. We select DCT domain as a convenient basis, Balanced Bernoulli Matrix as a measurement matrix and some $l-1$ norm minimization algorithm which are produced by KL1p library in order to clarify the effectiveness of our approach.

Chapter 7

Evaluation

In this Chapter, we present the evaluation of proposed algorithms. We conduct evaluations both in the real world environment and in simulation. For each of them, we provide the evaluation methodology, including evaluation environment, metrics and comparison targets, and then present its results and discussions.

7.1 Application Scenario

In this section, we define the application scenario which is used for experiments. Firstly, we show the scenario and then we perform preparatory experiment. We illustrate the result and calculate the optimized matrix size M_{opt}.

7.1.1 Scenario

In this experiment, we assume the human movement detection application, and based on this scenario, we firstly collect the raw data and calculate optimized matrix size M_{opt} as described in Chap .5. Using these data, we conducted simulation evaluation to evaluate the impact of *Balanced Bernoulli Matrix* in some solver algorithms.

Table 7.1: Pre-experiemtns environment

Parameter	Values
The number of test subjects	3
The number of data-sets for each pattern	20 (at minimum)
Sensing Rate	30 mili seconds
Data set size N	256
Data Type	Acceleration
Convenient Basis Ψ	DCT
Measurement Matrix Φ	Gaussian
Decoder algorithm	CoSaMP [30]

The description of scenario is following: This application detects 3 human state, walking, running and skipping, whose patterns occur with same probability, from acceleration data that is collected from the sensor attached to user's back. Thus, the application state is 3 patterns: walking, running and skipping and we collect these accelerations and calculate optimized matrix size M for each pattern.

7.1.2 Pre-experiments environment

Table 7.1 shows the description of pre-experiments which collects sensor data patterns and calculate optimized matrix size M. Based on the scenario (described as above 7.1.1), we attached sensors to 3 users and collect acceleration data for each pattern. The number of data-sets which is fixed to CS process for each pattern is at least 20. We select the CoSaMP [30] to define the optimized matrix size M.

7.1.3 Result and properties

Fig .7.1 - Fig .7.6 shows example acceleration and sparse representation data for each pattern.

Based on above, we calculate appropriate matrix size M_{opt}: this result is shown in Fig 7.7.

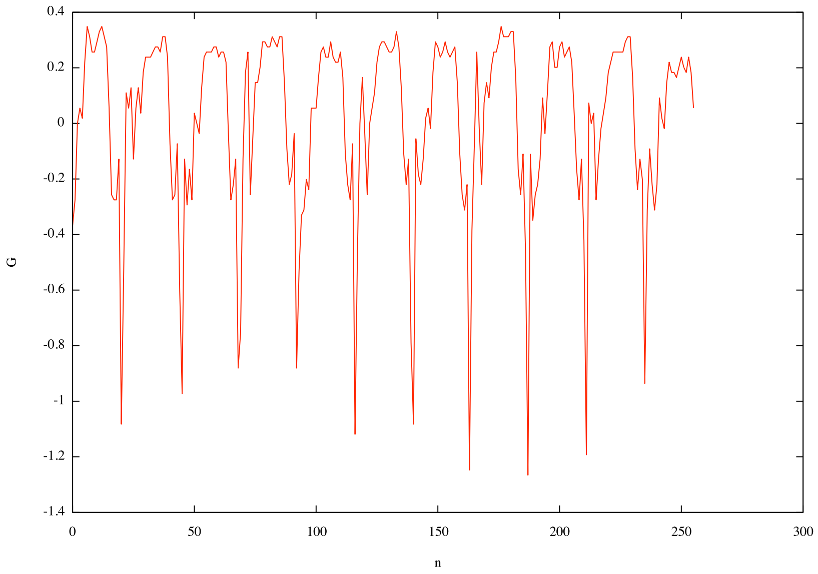

Figure 7.1: Experiments result1-1: Acceleration data of walking

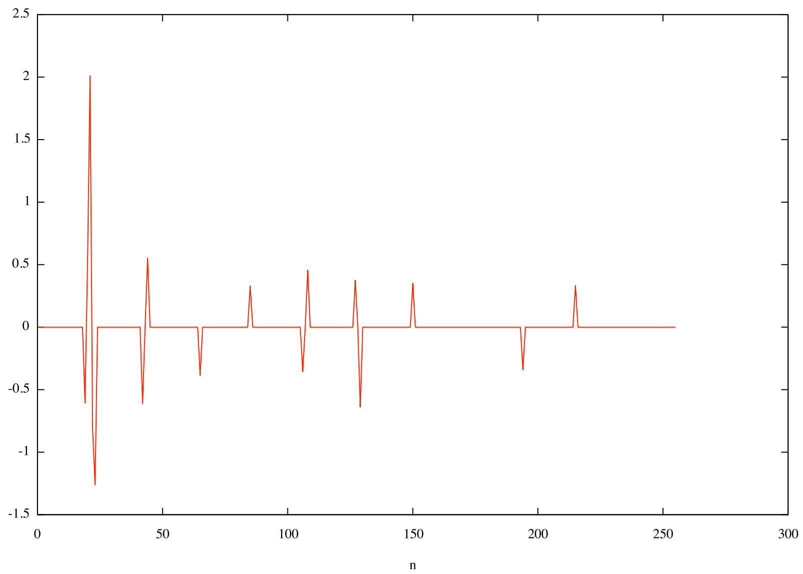

Figure 7.2: Experiments result1-2: Walking data in DCT domain

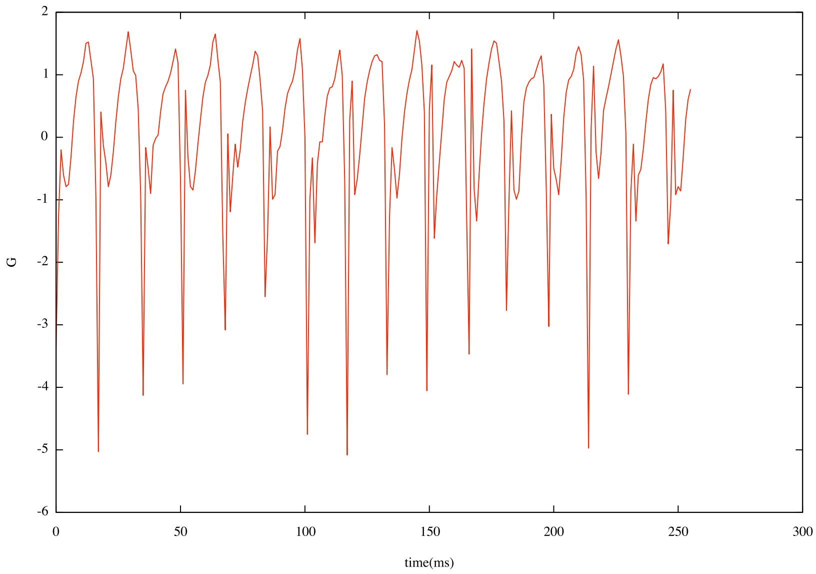

Figure 7.3: Experiments result2-1: Acceleration data of running

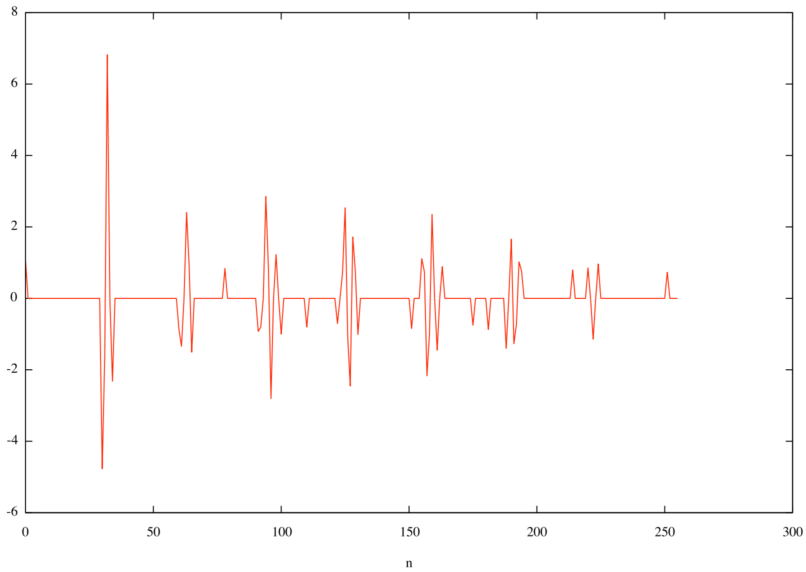

Figure 7.4: Experiments result2-2: Running data in DCT domain

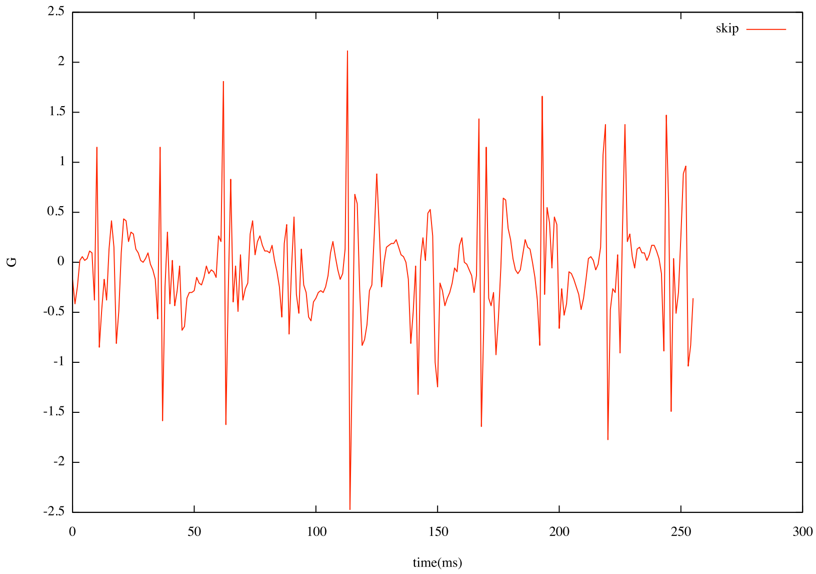

Figure 7.5: Experiments result3-1: Acceleration data of skipping

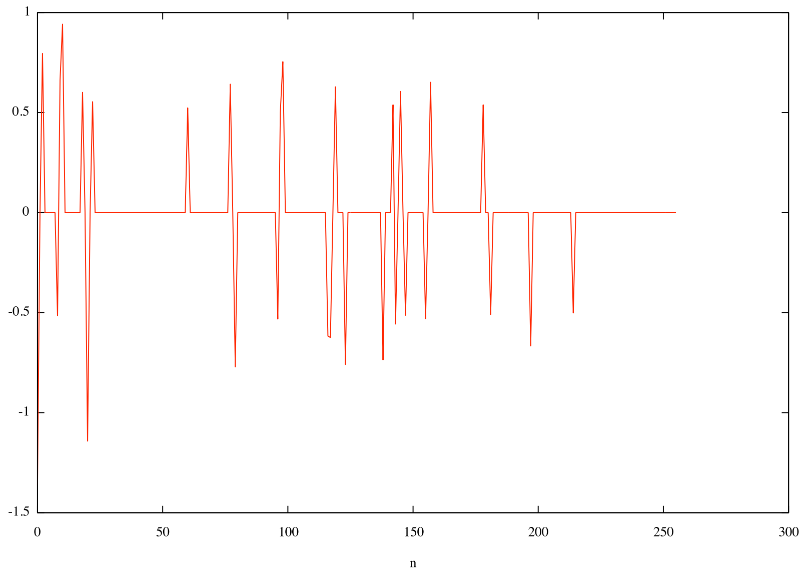

Figure 7.6: Experiments result3-2: Skipping data in DCT domain

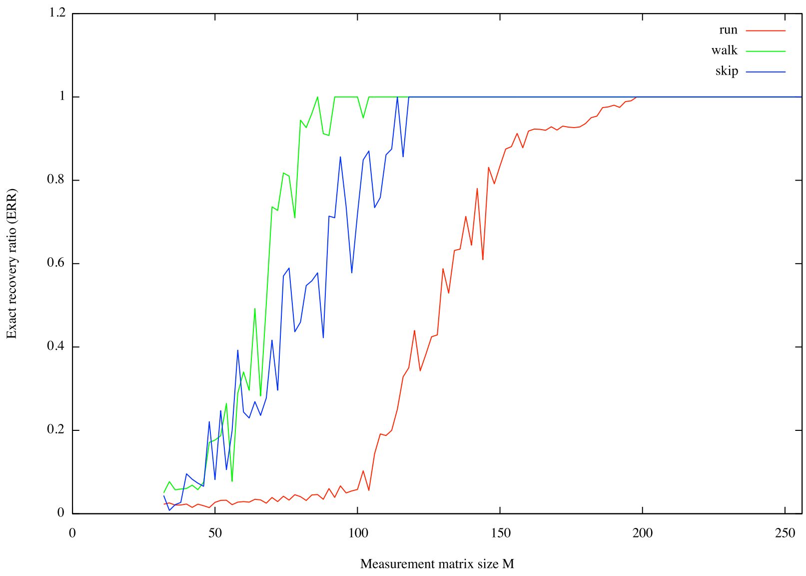

Figure 7.7: Experiments result: Comparison ERR

Each optimized matrix size M_{opt} is following as:

- Running ($M_{opt:run}$): 192
- Walking ($M_{opt:walk}$): 96
- Skipping ($M_{opt:skip}$): 124

7.2 Evaluation of Energy Consumption

We conduct a real world evaluation to clarify the impact of PMOA on energy consumption. Unless otherwise noted, values expressed in this Section are rounded to four significant figures.

7.2.1 Evaluation Methodology

The impact of normal CS on energy consumption of sensor nodes is shown at Chap .4. PMOA aims to maximize the efficiency of CS in terms of energy consumption, therefore, we compare it against existing CS algorithm. We use same CS encoding parameter, original signal length N, measurement matrix Φ and convenient basis Ψ, both on normal CS and PMOA, except for matrix size M. We evaluate PMOA and normal CS in perspective of A) energy consumption and B) the number of transmission. Unless otherwise noted, we evaluate each item by deriving the average of all data.

A. Energy Consumption

The purpose of PMOA is to minimize energy consumption of sensor nodes, therefore, this is the most important perspective in our work. Energy consumption is calculated by subtracting the voltage of batteries at the end of the experiment from that at the beginning of it. It is affected by the characteristics of both sensor nodes and batteries, therefore, we use them as pairs throughout the evaluation.

B. The number of transmission

The number of transmission also indicates energy consumption because data transmission is the main factor of energy drain in tiny sensor nodes. Since our approach can reduce transmission data, we evaluate this to verify the our approach. The number of transmission is counted at sink by checking the sequence number of packets.

Table 7.2 shows the parameters used in the evaluation. Since there are no combination of measurement data in this experiments, every measurements are sent alone; M data can be generated in a single CS process and these are sent one by one, which means that M packets are sent to host in a single process. Of course, it is more effective to store 2 or 3 generated data into 1 packet, however, we select this method in order to clarify the performance of PMOA. We use DCT and *Balanced Bernoulli Matrix* as described in Chap .6 and the scenario shown in sec .7.1 based matrix size M_{opt} and static matrix size M. Each pattern switching was carried out at random. We let the evaluation run until battery expires.

Table 7.2: Energy Consumption Evaluation Environment

Parameter	Values
The number of sensor nodes	5
Duration	Over 9 hours
Sensing Rate	30 mili seconds
Data Type	Acceleration
Scenario	human movement detection, shown in sec .7.1
Convenient Basis Ψ	DCT
Measurement Matrix Φ	Balanced Bernoulli
Original Signal Length N	256
Optimized Matrix Size M_{opt}	96, 124, 192

Table 7.3: Experiments Result3: The number of packet transmission

Parameter	Normal CS	PMOA
The average of transmission	765168 times	632772 times
The average of packet reduction	–	132396 times
The reduction ratio of transmission	–	15%

7.2.2 Evaluation Results and Discussions

Some example of results are shown in Fig .7.8 - 7.9. It is clear that our approach is effective on energy consumption. Compared to normal CS, PMOA improves their lifetime by 11.7% in average case. Table 7.3 shows the result of the reduction of packet transmission. PMOA can reduce the number of data transmission by 15% in average case.

7.3 Evaluation of Reliability

This section describes the simulation evaluation conducted to clarify the impact of PMOA and *Balanced Bernoulli Matrix* on the CS decoding reliability.

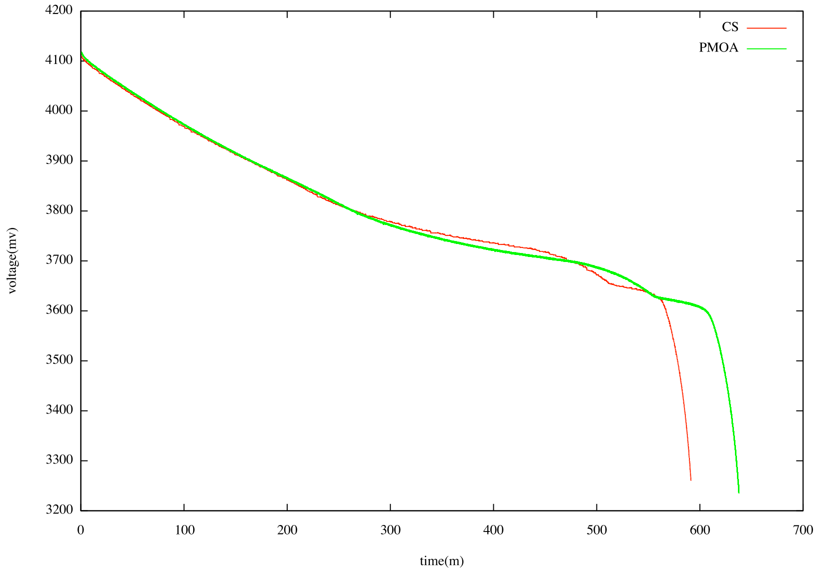

Figure 7.8: Experiments Result1: Comparison of Energy Consumption between Normal CS and PMOA

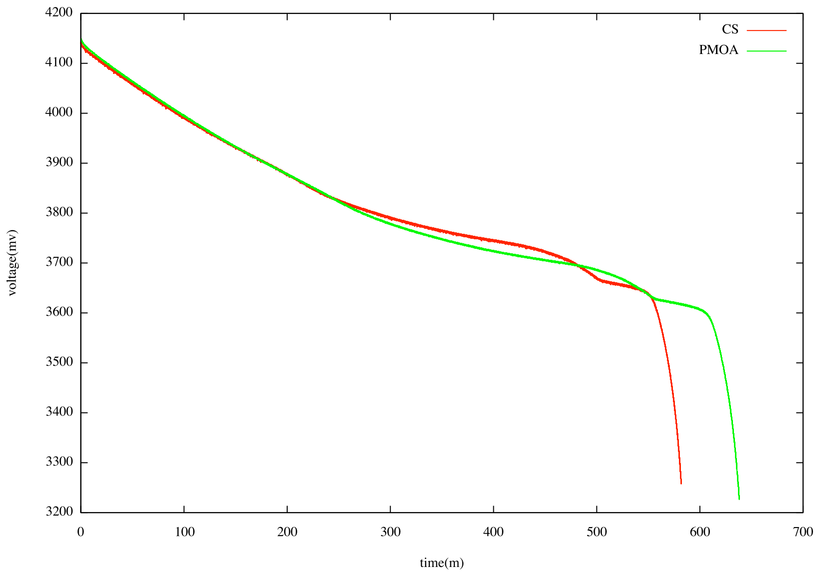

Figure 7.9: Experiments Result2: Comparison of Energy Consumption between Normal CS and PMOA

Table 7.4: Reliability Experiments Environment

Data Type	Acceleration
Scenario	human movement detection
Convenient Basis Ψ	DCT
Measurement Matrix Φ	Balanced Bernoulli, Gaussian and Bernoulli
Original Signal Length N	256
Optimized Matrix Size M_{opt}	96, 124, 192
Decoding algorithm	CoSaMP, ROMP, SL0 and Subspace Pursuit

Table 7.5: Experiments Result8: Optimized Matrix Size M_{opt} on Gaussian Matrix

Data-sets	CoSaMP	SL0	SP
Run	192	154	222
Skip	124	124	138
Walk	96	78	128

7.3.1 Evaluation Methodology

Though this experiment, we observe the impact of proposed algorithm on CS decoding reliability. In this experiment, we observe the reliability of PMOA and *Balanced Bernoulli Matrix* as a measurement matrix on various decoding algorithm described in 6.3.2. Using same data-sets in 7.2, we measured the relation between exact recovery ratio (ERR) and the kind of measurement matrix. Table .7.2 shows the parameters.

7.3.2 Evaluation Results and Discussions

Some example of results are shown in Fig .7.10 - 7.13. The almost of all graphs show *Balanced Bernoulli Matrix* slightly improves ERR on various decoding algorithms except for ROMP. ROMP didn't work well on any matrices or data-sets. Thus, we exclude ROMP from this evaluation.

Table 7.5 and 7.6 show optimized matrix size M_opt for each matrix.

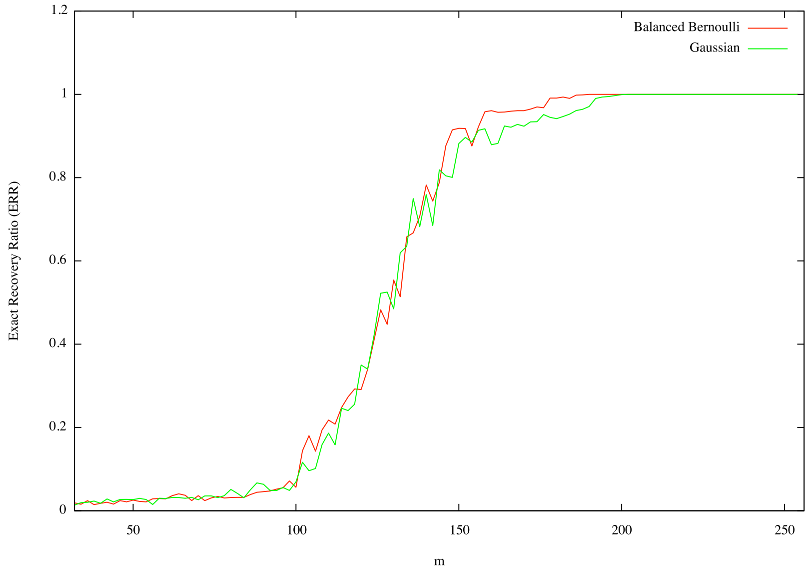

Figure 7.10: Experiments Result4: The Exact Recovery Ratio of Balanced Bernoulli Matrix and Gaussian Matrix on CoSaMP

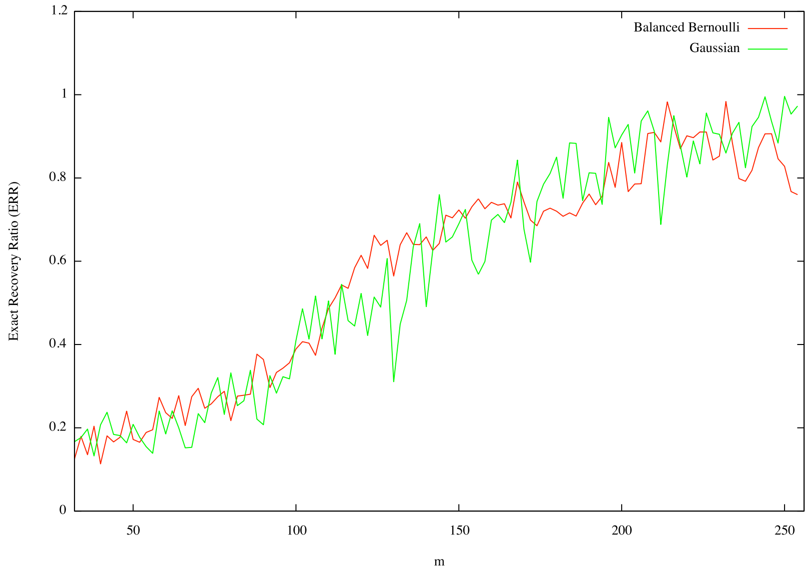

Figure 7.11: Experiments Result5: The Exact Recovery Ratio of Balanced Bernoulli Matrix and Gaussian Matrix on ROMP

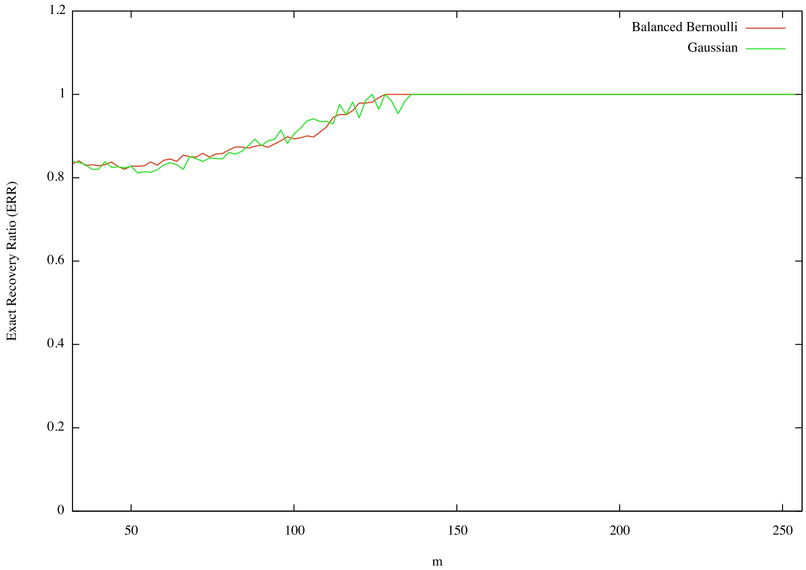

Figure 7.12: Experiments Result6: The Exact Recovery Ratio of Balanced Bernoulli Matrix and Gaussian Matrix on SL0

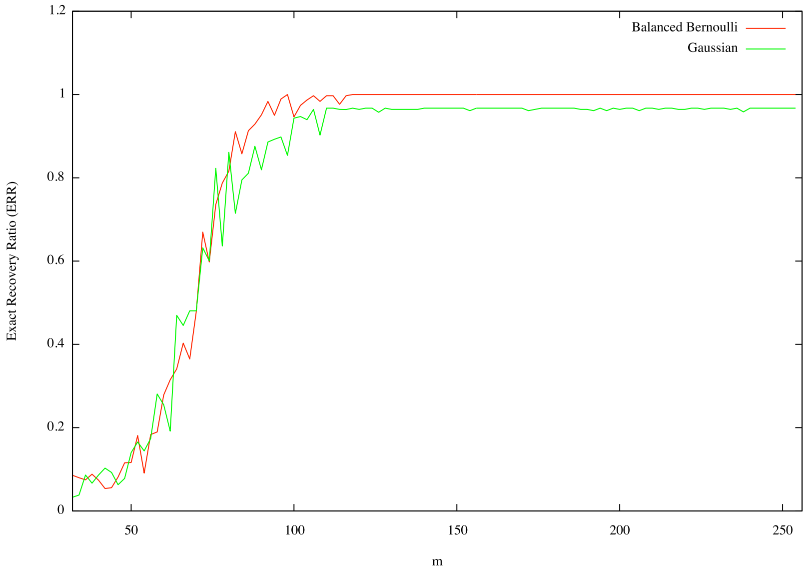

Figure 7.13: Experiments Result7: The Exact Recovery Ratio of Balanced Bernoulli Matrix and Gaussian Matrix on Subspace Pursuit

Table 7.6: Experiments Result9: Optimized Matrix Size M_{opt} on Balanced Bernoulli Matrix

Data-sets	CoSaMP	SL0	SP
Run	190	150	216
Skip	112	118	142
Walk	92	80	118

Compared to Gaussian matrix, *Balanced Bernoulli Matrix* can reduce 4.22 matrix length in average case.

7.4 Summary

In this section, we performed real-world and stimulation experiments. It turned out that our approach, PMOA, can improves the lifetime of sensor nodes by 11.7% in average case and reliability on many data-sets, measurements matrix and decoding algorithms.

Chapter 8

Conclusion and Future Work

In this Chapter, we summarize this thesis, and discuss future direction of our research.

8.1 Conclusion

In this thesis, we proposed Pattern-based Matrix-size Optimization Algorithm (PMOA), an algorithm which improve Compressive Sensing (CS) efficiency and reliability. In Body Sensor Network applications, sensor value changes dynamically and also appropriate matrix size changes. This leads inefficiency and inaccuracy on Compressive Sensing. In order to improve this, PMOA optimizes the *measurement matrix* size M along specific Body Sensor Network application scenario. This approach and design enable them to efficiently encode sensor data with high reliability in decoding.

We conducted real-world and stimulation experiments in order to evaluate PMOA. The result shows our approach can achieve the improvement of lifetime by 11.7%.

8.2 Future Work

In this section, we describe our future work: convenient basis optimization and more sophisticated measurement matrix design.

As described in Chap .5, it is hard to optimize a structure of measurement matrix for each pattern due to resource constrains of sensor nodes. However, an additional implementation of sparse representation algorithm is low complexity, 100 - 200 lines at most. If, therefore, we optimize not only measurement matrix size but also a convenient basis for each pattern, we will get more sparsity and this improves CS efficiency and reliability. For instance, DCT can achieve high sparsity for human walking. When human is running, however, DCT don't work well as walking and make low sparsity. So, we select alternative algorithm, such as FFT, to improve sparsity for running. This optimization may improve our approach.

Second is sophistication of measurement matrix design, which is used *Balanced Bernoulli Matrix* in our work, to make more incoherence. Many existing works attack this issue and improve CS efficiency and reliability. Thus, it is reasonable to consider another approach.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Hideyuki Tokuda for his technical and professional advice, encouragement, and guidance throughout my years in bachelor's and master's degrees. I am deeply grateful to Lecturer Jin Nakazawa for their valuable and technical comments on this thesis.

I would like to thank the members of the lab, especially, Associate Professor Kazunori Takashio, Research Assistant Tomotaka Ito, Dr. Michio Honda, and Dr. Naoya Naotame for their comments, advices and encouragement. I would like to thank fellows of LINK research group for their valuable discussions, supports and friendships. I also thank Kenji Yonekawa, Kazuhiro Imura, Junki Terayama, Kodai Suzuki, Ayaka Yagi, Shoko Minagawa, Kenichi Yasukata, Mina Sakamura, Shinosuke Usami and Shinsuke Takagi for their daily support and encouragement. Finally, I wish to thank numerous members of Tokuda and Murai lab for their great support.

Akito Ito
January 22, 2013

Bibliography

[1] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[2] Ch Antonopoulos, A Prayati, T Stoyanova, C Koulamas, and G Papadopoulos. Experimental evaluation of a wsn platform power consumption. *Structure*, 2009.

[3] Phongsak Keeratiwintakorn Yuthapong Pathachai. An energy model for transmission in telos-based wireless sensor networks. 2007.

[4] Gilman Tolle, Joseph Polastre, Robert Szewczyk, David Culler, Neil Turner, Kevin Tu, Stephen Burgess, Todd Dawson, Phil Buonadonna, David Gay, and Wei Hong. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 51–63, New York, NY, USA, 2005. ACM.

[5] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 64–75, New York, NY, USA, 2005. ACM.

[6] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware design experiences in zebranet. In *Proceedings*

of the 2nd international conference on Embedded networked sensor systems, SenSys '04, pages 227–238, New York, NY, USA, 2004. ACM.

[7] Massudi Mahmuddin Khalid A. Al-Saud and Amr Mohamed. Wireless body area sensor networks signal processing and communication framework : Survey on sensing, communication technologies, delivery and feedback. pages 121–132.

[8] Kenji Yonekawa, Takuro Yonezawa, Jin Nakazawa, and Hideyuki Tokuda. Fash: Detecting tiredness of walking people using pressure sensors. IEEE, 11 2009.

[9] Frank Mokaya, Cynthia Kuo, and Pei Zhang. Mars: a muscle activity recognition system using inertial sensors. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, IPSN '12, pages 97–98, New York, NY, USA, 2012. ACM.

[10] Koji Yatani and Khai N. Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 341–350, New York, NY, USA, 2012. ACM.

[11] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C. Leung. Body area networks: A survey. *Mob. Netw. Appl.*, 16(2):171–193, April 2011.

[12] Hyejung Kim, R.F. Yazicioglu, P. Merken, C. Van Hoof, and Hoi-Jun Yoo. Ecg signal compression and classification algorithm with quad level vector for ecg holter system. *Information Technology in Biomedicine, IEEE Transactions on*, 14(1):93 –100, jan. 2010.

[13] S.S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *Signal Processing Magazine, IEEE*, 19(2):51 –60, mar 2002.

[14] L. Barboni and M. Valle. Experimental analysis of wireless sensor nodes current consumption. In *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, pages 401 –406, aug. 2008.

[15] M. Casares, A. Pinto, Youlu Wang, and S. Velipasalar. Power consumption and performance analysis of object tracking and event detection with wireless embedded smart cameras. In *Signal Processing and Communication Systems, 2009. ICSPCS 2009. 3rd International Conference on*, pages 1 –8, sept. 2009.

[16] Rachit Agarwal, Rafael V. Martinez-Catala, Sean Harte, Cedric Segard, and Brendan O'Flynn. Modeling power in multi-functionality sensor network applications. In *Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications*, SENSORCOMM '08, pages 507–512, Washington, DC, USA, 2008. IEEE Computer Society.

[17] MEMSIC. Tmote sky. , Dec 2011.

[18] MEMSIC. Telosb mote. `http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152%3Atelosb`, Dec 2011.

[19] M. B. Wakin E. J. Candes. An introduction to compressive sampling. *IEEE, Signal Processing Magazine, Vol. 25*, pages 21–30, 2008.

[20] Richard G. Baraniuk. Compressive sensing. *IEEE, Signal Processing Magazine, Vol. 24*, pages 118–121, 2007.

[21] Toshiyuki TANAKA. Mathematics of compressed sensing. *IEICE Fundamental Review Vol.4 No.1*, pages 39–47, 2010.

[22] ASHINO Ryuichi. Basics and research trends on compressive sensing. *Systems, control and information*, 55(3):88–93, 2011-03-15.

[23] Lewis Caroll. Alice's adventures in wonderland. http://www.cs.cmu.edu/~rgs/alice-table.html.

[24] Emmanuel J. Cands. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9—10):589 – 592, 2008.

[25] Xiaopei Wu and Mingyan Liu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, IPSN '12, pages 1–12, New York, NY, USA, 2012. ACM.

[26] Z. Charbiwala, S. Chakraborty, S. Zahedi, Younghun Kim, M. B. Srivastava, Ting He, and C. Bisdikian. Compressive Oversampling for Robust Data Transmission in Sensor Networks. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, March 2010.

[27] D. Akimura, Y. Kawahara, and T. Asami. Compressed sensing method for human activity sensing using mobile phone accelerometers. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, pages 1 –4, june 2012.

[28] Sun. Sunspot. http://www.sunspotworld.com/docs/Red/SunSPOT-TheoryOfOperation.pdf, Dec 2009.

[29] MEMSIC. Iris mote. http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis, Dec 2011.

[30] Deanna Needell and Joel A. Tropp. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM*, 53(12):93–100, December 2010.

[31] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, January 1998.

[32] Jianping Xu, Yiming Pi, and Zongjie Cao. Optimized projection matrix for compressive sensing. *EURASIP Journal on Advances in Signal Processing*, 2010(1):560349, 2010.

[33] Oracle. Oraclespot. `http://www.sunspotworld.com/docs/Yellow/eSPOT8ds.pdf`, Dec 2011.

[34] Wikipedia. Bernoulli process. `http://en.wikipedia.org/wiki/Bernoulli_process`, Dec.

[35] Boost c++ libraries. `http://www.boost.org/`, Dec.

[36] Kl1p a portable c++ library for compressed sensing. `http://kl1p.sourceforge.net/home.html`, Dec.

[37] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):310 –316, april 2010.

[38] Wei Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *Information Theory, IEEE Transactions on*, 55(5):2230 –2249, may 2009.

[39] G. Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. A fast approach for overcomplete sparse decomposition based on smoothed l0 norm. *CoRR*, abs/0809.2508, 2008.