공학박사 학위논문

Aerial Acoustic Communication Using Chirp Signal

– System Design and Applications –

처프 신호를 이용한 음파 통신 기법 연구

– 시스템 설계와 응용서비스 –

2014년 8월

서울대학교 대학원

전기컴퓨터공학부

이 혜 원

공학박사 학위논문

Aerial Acoustic Communication Using Chirp Signal

– System Design and Applications –

처프 신호를 이용한 음파 통신 기법 연구

– 시스템 설계와 응용서비스 –

2014년 8월

서울대학교 대학원

전기컴퓨터공학부

이 혜 원

Aerial Acoustic Communication Using Chirp Signal

– System Design and Applications –

지도교수 최성현

이 논문을 공학박사 학위논문으로 제출함
2014년 7월

서울대학교 대학원

전기정보공학부

이혜원

이혜원의 공학박사 학위 논문을 인준함

2014년 6월

위 원 장: 이병기 (인)

부위원장: 최성현 (인)

위 원: 박세웅 (인)

위 원: 진성근 (인)

위 원: 최준원 (인)

Abstract

Aerial Acoustic Communication Using Chirp Signal

– System Design and Applications –

Today's smart devices such as smartphones and tablet/wearable PCs are equipped with voice user interface (UI) in order to support intuitive command input from users. Speakers and microphones of the voice UI are generally used to play and record human voice and/or environmental sound, respectively. Accordingly, various *aerial acoustic communication* techniques have been introduced to utilize the voice UI as an additional communication interface beyond WiFi and/or Bluetooth. Smart devices are especially suitable for the aerial acoustic communication since the application processor (AP) of smart devices can process the sound to embed or fetch information in it. That is, smart devices work similar to *software defined radio* platform. The aerial acoustic communication is also very versatile as any audio interface can be utilized as a communication interface. In this dissertation, we propose an aerial acoustic communication technique using inaudible chirp signal as well as corresponding receiver architecture for smart devices. We additionally introduce the applications of the proposed communication technique in indoor environments.

We begin the receiver design for aerial acoustic communication by measuring the characteristics of indoor acoustic channel, composed of speaker, air-medium, and microphone. Our experimental research reveals that the indoor acoustic channel typically has long delay spread (approximately 40 msec), and it is very frequency-selective due to the frequency response of audio interfaces. We also show that legacy physical layer (PHY) modulation schemes such as phase/frequency shift keying (PSK/FSK) are likely to fail in this indoor acoustic channel, especially in long communication scenarios, due mainly to the instability of local oscillator and frequency selectivity of audio interfaces.

In order to resolve the above-mentioned problems, we use *chirp signals* for the aerial acoustic communication. The proposed acoustic receiver supports long-range communication independent of the device characteristics over the severely frequency-selective acoustic channel with large delay spread. The *chirp signal* has time-varying frequency with a specific *frequency sweeping rate*. The chirp signal was widely used for radar applications due to its capability of resolving multi-path propagation. However, this dissertation is the first study of adopting chirp signal in aerial acoustic communications for smart devices. The proposed receiver architecture of chirp binary orthogonal keying (BOK) can be easily implemented via fast Fourier transform (FFT) in smart devices' application layer. Via extensive experimental results, we verify that the proposed chirp signal can deliver data at 16 bps up to 25 m distance in typical indoor environments, which is drastically extended compared to the few meters of previous research.

The data rate of 16 bps is enough to deliver short identification (ID) in indoor environments. The exemplary applications with this short ID can be multimedia content recognition and indoor location tracking. The low data rate, however, might be a huddle of the proposed system to be applied to the services that require high data rate. We design a backend server architecture in order to compensate for the low data rate and widen the application extent of the proposed receiver. The smart devices can send queries in order to refer to the backend server for additional information that is related with the received ID. We also propose an energy-efficient recording and processing method for the acoustic signal detection. Note that it would consume huge amount of energy if the smart devices contiguously sensed the acoustic signal for 24 hours. The smart devices instead control the sensing (i.e., recording) timing so that it is activated only when there exists chirp signal. This can drastically extend the battery lifetime by removing unnecessary signal processing. We also present two application examples of the proposed receiver, namely, (1) TV content recognition, and (2) indoor location tracking, including technical discussions on their implementations. Experiments and field tests validate the feasibility of the proposed aerial acoustic communication in practical environments.

Keywords: Chirp signal, aerial acoustic communication, smart devices, software-based digital modem, application design.

Student Number: 2006-21267

Contents

Abstract i

Contents iv

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Acoustic communication 1

1.1.1 Underwater acoustic communication 2

1.1.2 Aerial acoustic communication 3

1.2 Overview of Existing Approaches 5

1.2.1 Indoor Location Tracking 5

1.2.2 Data Communication using Acoustic Signal 7

1.2.3 Commercial Services 9

1.2.4 Limitations of Previous Work 10

1.3 Main Contributions 11

1.3.1 Acoustic Channel and PHY Analysis 12

1.3.2 Receiver Design for Acoustic Chirp BOK 12

1.3.3 Applications of Chirp BOK Receiver 13

1.4 Organization of the Dissertation 13

2 Acoustic Channel and PHY Analysis 15

2.1 Introduction 15

2.2 Characteristics of Indoor Acoustic Channel 18

2.2.1 Hearing Threshold of Human 18

2.2.2 Frequency Response of Various Audio Interfaces 21

2.2.3 Delay Spread of Acoustic Channel 25

2.3 Revisit of Existing Modulation Schemes 26

2.3.1 Case Study: Phase Shift Keying 28

2.3.2 Case Study: Frequency Shift Keying 35

2.3.3 Chirp Binary Orthogonal Keying (BOK) 40

2.4 Performance Evaluation of PHY Modulation Schemes 42

2.4.1 Experimental Environment 44

2.4.2 PSK Demodulator 44

2.4.3 FSK Demodulator 45

2.4.4 BER of PHY Modulation Schemes 46

2.5 Summary 47

3 Receiver Design for Acoustic Chirp BOK 49

3.1 Introduction 49

3.2 Chirp Signals and Matched Filter Receiver 51

3.2.1 Notation of Chirp Signals 51

3.2.2 Matched Filter and FFT 53

3.2.3 Envelope Detection of Chirp Auto Correlation 55

3.3 System Design and Receiver Architecture 59

3.3.1 Frame and Symbol Design 60

3.3.2 Signal Reception Process 63

3.3.3 Receiver Architecture 65

3.3.4 Symbol combining for BER enhancement 68

3.4 Performance Evaluation of Chirp BOK Receiver 73

3.4.1 Experimental Environment 74

3.4.2 Transmission Range in Indoor Environment 74

3.4.3 Multi-path Resolution Capability of Chirp Signal 75

3.4.4 Symbol Sampling and Doppler Shift 82

3.4.5 Selective combining 85

3.5 Summary 87

4 Applications of Chirp BOK Receiver **90**

4.1 Introduction 90

4.2 Backend Server Architecture 93

4.2.1 Implementation of Backend Server 93

4.2.2 Operation of Backend Server 95

4.3 Low Power Operation for Smart Devices 98

4.3.1 Reception Process of Chirp BOK receiver 98

4.3.2 Revisit of Signal Detection in Wireless Communications . . . 100

4.3.3 Chirp Signal Detection using PSD 102

4.3.4 Performance Evaluation of Signal Detection Algorithm 105

4.4 Applications of Chirp BOK Receiver and Feasibility Test 110

4.4.1 TV Content Recognition 111

4.4.2 Indoor Location Tracking in Seoul Subway 114

4.4.3 Device to Device Communication 118

4.5 Summary 120

5 Conclusion and Future Work 123

5.1 Research Contributions 123

5.2 Future Work and Concluding Remark 125

Abstract (In Korean) 136

감사의 글 **139**

List of Tables

1.1 Various Modulation Schemes for Underwater Communication [1,2] . 4

2.1 Typical Audio Sampling Frequencies and Applications 21

3.1 System Parameters of Chirp BOK Receiver 63

4.1 False alarm and miss detection of chirp signals 103

4.2 Measured BER of Testing Points 117

List of Figures

1.1 Analogy of antennas and speakers/microphones for wireless communication. 2

1.2 Frequency band of acoustic waves. 5

2.1 Power spectral density of environmental background noise in various routine activities. 18

2.2 Theoretic and empirical hearing threshold of human ears. 20

2.3 Frequency response of speakers of various smart devices and off-the-shelf speakers. 23

2.4 Frequency response of microphones of various smart devices. 24

2.5 Impulse responses of aerial acoustic channels in indoor office environment, measured using LFS signal. 27

2.6 Estimated phase of recorded audio using various smart devices in 5 m LOS channel. 29

2.7 Estimated phase of recorded audio using various smart devices in 5 m NLOS channel. 31

2.8 Frame structure of BPSK signal. 32

2.9 Demodulation results of BPSK signals in Galaxy Note10.1. 33

2.10 Demodulation results of BPSK signals in Galaxy Note3. 34

2.11 Demodulation results of BPSK signals in Vega Iron. 35
2.12 Empirical constellation point of non-coherent BFSK receivers for four smart devices in 5 m LOS channel. 37
2.13 Adaptive ecision bound of non-coherent BFSK receivers for four smart devices in 5 m LOS channel. 39
2.14 Wave form and frequency changes of up chirp and down chirp. 40
2.15 Auto correlation and cross correlation of chirp symbols. 42
2.16 An example of chirp BOK: modulation and demodulation. 43
2.17 Block diagram of non-coherent FSK receiver. 46
2.18 Empirical BER of PSK, FSK, and chirp BOK modulations in indoor acoustic channel. 47

3.1 Binary receiver for signal set $\phi_1(t)$ and $\phi_2(t)$ using matched filter. . . 52
3.2 Binary receiver for signal set $\phi_1(t)$ and $\phi_2(t)$ using matched filter, implemented by FFT and IFFT. 54
3.3 An example of auto correlation for chirp signals sweeping from 19.5 to 22 kHz with 20 msec symbol duration. 56
3.4 A block diagram of envelope detector. 59
3.5 Comparison of correlation output and its envelope. 60
3.6 Design of chirp signals and corresponding frame structure with preamble and guard interval. 61
3.7 Reception process of the chirp BOK receiver. 64
3.8 Block diagram of the chirp BOK receiver. 66
3.9 Decoding example of the proposed modem. 67
3.10 An snapshot of symbol combining. 71
3.11 An snapshot of symbol combining with a severely corrupted symbol. . 72
3.12 Algorithm flow chart of selective combining. 73

3.13 Success probability of signal reception at 10 m and 25 m distances for various smart devices. 74

3.14 Signal constellation points of chirp BOK modem. 76

3.15 Chirp envelopes of Galaxy Note2 in single-path environment. 77

3.16 Chirp envelopes of Galaxy Note2 in multi-path environment. 79

3.17 Chirp envelopes of two-ray propagation model obtained from MATLAB. 80

3.18 Experiment topology and results in stereo chirp transmission. 81

3.19 Shift of peaks due to Doppler shift, assuming 20 msec symbol is transmitted. 83

3.20 Experimental results of Doppler shift and shifts of envelope peaks. . . 84

3.21 Histogram of BER in function of soft value. 85

3.22 BER of selective combining emulated by the acoustic signal trace of various smart devices. 86

3.23 Average recording overhead of selective combining emulated by the acoustic signal trace of various smart devices. 86

4.1 Message flow of smart device and backend server. 95

4.2 An example of JSON response format. 96

4.3 Two-step recording for signal detection and demodulation. 99

4.4 PSD of recorded audio at three smart devices. 100

4.5 PSD of different types of audio at Vega N6. 101

4.6 An example of calculating the peak PSD ratio. 102

4.7 Receiver operating characteristic curve and the effect of FFT size. . . 105

4.8 Peak PSD ratio values in indoor experiment for 65,536- point FFT. . . 107

4.9 Peak PSD ratio values in indoor experiment for 8,192-point FFT. . . . 107

4.10 Peak PSD ratio values in indoor experiment for 4,096-point FFT. . . . 108

4.11 False alarm and miss detection probability of the propose signal detection algorithm. 109

4.12 Application examples of chirp signal receiver. 111

4.13 Frequency response of AAC and AC3 encoders. 114

4.14 Experiments on audio sources in AAC, AC3, and WAV audio formats. 115

4.15 PSD of Chirp signal passed through the PA system of Seoul Metro. . . 116

4.16 BER testing points in a subway car. 117

4.17 An exemplary application of D2D connection and backend server framework to authenticate the chirp signal and D2D connection. 120

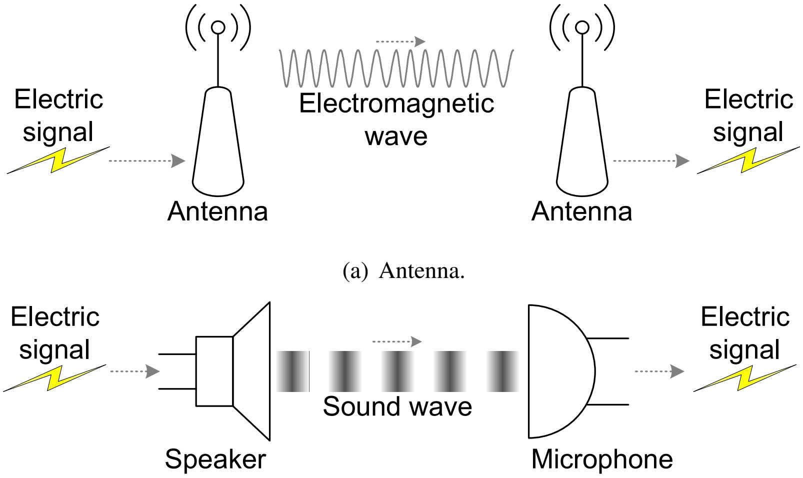
Chapter 1

Introduction

1.1 Acoustic communication

Today's various communication technologies can be categorized by the medium that a specific technology uses to deliver information from transmitter to receiver. Common wireless communication technologies such as WiFi [3] and Bluetooth [4] deliver information through electromagnetic waves, generated by moving electrons of conducting material. Antennas are used to transmit and receive the electromagnetic wave, where the size of antenna depends on the center frequency and wavelength of the electromagnetic wave.

Separate from the wireless communications based on electromagnetic waves, *acoustic communication* refers to the communication technique that uses sound waves. Sound waves propagate through the mechanical vibration of a medium (e.g., water or air), and it is a type of pressure wave. Unlike electromagnetic waves that use antennas, sound waves can be transmitted and received by speakers and microphones. Speakers generally converts electric signals into mechanical vibration of membrane using electric magnetics, which results in a pressure wave of medium. Microphones work in a re-



(a) Antenna.

(b) Speaker and microphone.

Figure 1.1: Analogy of antennas and speakers/microphones for wireless communication.

verse manner: a thin membrane reacts to the vibration of medium so that condensers connected to the membrane can convert the mechanical energy to electric signals. Figures 1.1(a) and 1.1(b) present the analogy of antennas and speaker/microphones for wireless communications using electromagnetic and sound waves, respectively.

1.1.1 Underwater acoustic communication

Acoustic communication has been developed mainly for the underwater communication of sensor networks to monitor water temperature, marine ecosystems, and the status of underwater platforms like oil rigs [1, 2, 5, 6]. One of the most well-known applications of the underwater acoustic communication is sound navigation and ranging, also known as sonar. An active sonar transducer broadcasts a reference signal, and then analyze the received echoes to range the distance between the transducer and target object. As one can imagine in the example of sonar, acoustic sound wave in underwater can reach up to a small km range. This is because the sound wave in dense medium propagates faster and farther than in sparse medium.

Traditional physical layer (PHY) digital modulation schemes that are used for electromagnetic wave communications, e.g., phase/frequency shift keying (PSK/FSK), can also be applied to acoustic communications to modulate sound signals. Table 1.1 summarizes the PHY modulation schemes as well as the data rate and communication range of representative underwater communication studies [1, 2]. Due to the lack of bandwidth as well as time-varying and multi-path propagating acoustic channels [2], the data rate of underwater acoustic communication is limited to a few tens of kbps in a small km range, depending on the water depth of operation [5]. Underwater acoustic communication also requires transceivers to operate in water, which typically means the use of special speakers and microphone modules, i.e., hydrophones.

1.1.2 Aerial acoustic communication

In this dissertation, we specifically focus on *aerial acoustic communication* that delivers information through airborne sound waves [17, 18]. The transmission range of aerial acoustic communication is much shorter (less than few tens of meters) than that of underwater acoustic communication due to the propagation loss in air medium. Depending on the operating frequency of aerial acoustic communication, the transceiver can either be an ultrasound transducer (for high frequencies over 22 kHz) using piezoelectric crystals or condensers, or an off-the-shelf audio interface that plays/records audio (under 22 kHz in general).[1] Figure 1.2 presents the frequency band of acoustic

[1] Note that the sampling rate of compact discs (CDs) is 44.1 kHz, and most of audio devices (i.e., speakers and microphones) support this sampling rate. The maximum playable and recordable frequency of 44.1 kHz is 22.05 kHz according to the Nyquist sampling theory.

Table 1.1: Various Modulation Schemes for Underwater Communication [1,2]

Investigator	Year	Modulation Type	Rate (kbps)
Catipovic [7]	1984	FSK	1.2
Kaya [8]	1989	16 QAM	500
Mackelburg [9]	1991	FSK	1.25
Howe [10]	1992	DPSK	1.6
Suzuki [11]	1992	DPSK	16
Stojanovic [12]	1993	8 QAM	3
Scussel [13]	1997	FSK	2.4
Jones [14]	1997	DPSK	20
Freitag [15]	1998	QPSK	6.7
Proakis [16]	2001	16 QAM	40

waves for infrasound, sound, and ultrasound.

Aerial acoustic communication under 22 kHz has recently been studied in depth [19–33]. Especially, smart devices such as smartphones, wearable computers, and tablet PCs are basically equipped with voice user interface (UI). The smart devices can play and record sound signals through their voice user interface, and then process through application processor (AP). Therefore the smart devices can be utilized as a software-defined radio (SDR) platform for aerial acoustic communications. It should be noted that the aerial acoustic communication is versatile because any audio interface-equipped device can be utilized as a communication device. In other words, it is hardware and operating system (OS) independent. For instance, a TV can be connected with a smartphone without using Bluetooth either WiFi. Numerous speakers in indoor buildings and/or public transportation systems (buses, subways) for public announcement (PA)

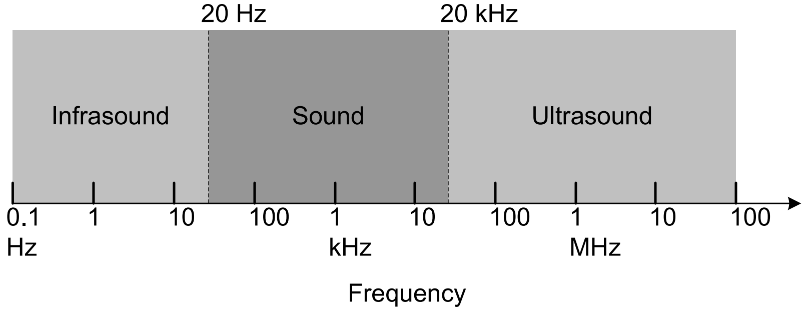


Figure 1.2: Frequency band of acoustic waves.

can be utilized as acoustic beacons to broadcast location information to nearby smartphones. We review the existing studies on aerial acoustic communications in academia and industry in the following section.

1.2 Overview of Existing Approaches

1.2.1 Indoor Location Tracking

One application of aerial acoustic communication is indoor location tracking [19–23]. Global positioning system (GPS) and WiFi positioning system (WPS) are widely used for location tracking. Note that, however, GPS signals from satellites are hardly received in indoors due to the severe penetration loss through walls and/or ceilings. In case of WPS, it usually relies on off-line table of signal strength of the beacons transmitted by nearby WiFi access points. The signal strength of WiFi beacons, however, severely varies over time due to the fluctuation of wireless channel. Moreover, the availability of WiFi access points frequently changes as the access points are turned on and off in an on-demand manner of operators and/or owners. Due to the above reasons, GPS and WPS are likely to fail in indoor environments.

Instead of GPS or WPS, acoustic signals can be used to calculate or identify the location of smart devices in indoor environments. In [19, 20], the authors propose indoor location tracking framework, referred to as Beep. A mobile device plays a beacon sound signal when it needs to identify its current location. Then multiple microphones attached at indoor walls record the beacon sound simultaneously, and the recorded sound is gathered at a central server through WiFi. The central server calculates the location of the mobile device from the time-of-arrival differences of recorded beacons, assuming that the relative locations of the microphones are known *a priori* and recordings are synchronized. The proposed Beep achieves 40 cm estimation error in a 20 m-by-9 m room.

The acoustic signals that can also enhance the accuracy of WPS [21,22], as the location tracking capability of the acoustic signal outperforms that of WiFi. [21] claims that the typical estimation error of WPS systems are 6∼8 m. Acoustic ranging is adopted to measure the relative distance among smart devices, assuming that the areal density of smart devices are high enough. The relative distance (and eventually, the topology) of neighboring smart devices can augment the WPS so that the estimation error decreases to 1∼2 meters. The authors in [22] similarly try to combine acoustic ranging and WPS, and introduce Centaur framework. Centaur adaptively weights acoustic ranging and WPS based on Bayesian inference model depending on the estimation error in indoor environments. The proposed framework removes the localization errors of WPS greater than 10 m, so that the accuracy of WPS is enhanced in an average sense.

Sound signals can also be used to measure the distance between two audio devices in centimeter level [23], thanks to the slow propagation speed of sound waves compared to electromagnetic waves. This highly precise distance measuring is far more accurate than the triangulation of wireless signals. BeepBeep of [23] adopts bi-directional ranging as well as self-ranging of two smart devices. This relaxes the requirement of time synchronization of two devices so that the time-of-arrival (which is equivalent to the distance between two devices) can easily be calculated without complex synchronization process.

1.2.2 Data Communication using Acoustic Signal

Aerial acoustic communication can be used for data transmission among audio-enabled devices [24–30]. [24] supports two different communication modes: audible mode and inaudible mode. Audible mode (735$\sim$4,410 Hz) uses M-ary FSK (MFSK) modulation. Multiple tones are used to transmit data at 5.6 kbps, where the signal of MFSK sounds like dial tones of phone call. Inaudible mode (18.4 kHz) is based on on-off keying. A single tone alternates on and off to represent binary bit 1 and 0, and the achievable data rate is 1.4 kbps. Both modes work up to 2 m distance at maximum in line-of-sight (LOS) communication.

A similar MFSK-based acoustic communication scheme, referred to as Digital Voice, is proposed in [25]. The number of tones and symbol durations are adaptively chosen depending on the acoustic environment and system target performance. The achievable data rate varies from tens of bps to thousands of bps accordingly. The tones with the frequency band under 12 kHz (i.e., audible band) are selected, where the tone spacing is controlled so that the played signal resembles the sounds of clarinets, crickets, robots (R2D2 of Star Wars), and birds.

The authors of [26] implement orthogonal frequency division multiplex (OFDM) with binary and quadrature PSK modulation schemes in the 6$\sim$7 kHz band. The proposed system, Dhwani, aims to substitute near field communication (NFC) by acoustic communication with a data rate of up to 800 bps in a very short range (less than 20 cm). Dhwani includes self-jamming feature (JamSecure) for security provisioning.

A receiver transmits a jamming signal while receiving an acoustic signal. Then the receiver can differentiate the jamming signal via successive interference cancellation (SIC) as it knows the waveform of jamming signal, while nearby eavesdroppers cannot decode the received acoustic signal due to the interference of jamming.

NTT DoCoMo introduces Acoustic OFDM in [27]. The motivation of Acoustic OFDM comes from the characteristics of human hearing system. Considering the frequency domain representation of sound, the human hearing system is more sensitive to the amplitude changes than the phase changes. Therefore two different sounds with the same frequency-domain amplitude but different phases are heard the same for human. Based on this fact, Acoustic OFDM hides digital data into an audio source, e.g., mp3 music file. Approximately 240 bps of information is modulated using OFDM in 6.4$\sim$8 kHz band, where the phase of each subcarrier embeds digital data. Then the amplitude of the modulated signal is shaped equal to that of the audio source in the same frequency band. Substituting the corresponding frequency components yields audio signal with embedded information, while it sounds the same to human. [28] similiarly embeds 600 bps data by modulating the phase of 6.4$\sim$8 kHz band, where the modulation is conducted via modulated complex lapped transform (MCLT) instead of fast Fourier transform (FFT) of OFDM.

Above studies are using audible band signal under 18 kHz. This is due mainly to the frequency response of audio interfaces; high-frequency sounds are hard to be played and/or recorded in some of the off-the-shelf smart devices.[2] The authors in [29, 30] attempt to use inaudible band over 18 kHz by using specific audio interfaces. More into detail, [29] implements a testbed using 200 ultrasound tags (transmitters) and 31 detectors (receivers) in a national hospital, where the detailed frequency band is

[2] The audible frequency range of human and frequency response of various smart devices will be discussed in detail in Chapter 2.

not specified. The authors theoretically shows that one can achieve a data rate of up to 100 bps at 10 m distance using the ultrasound transducers in indoor environments. In [30], commercial off-the-shelf laptops are used to build a mesh network via aerial acoustic communications. The built-in audio interface of the laptops are shown to have flat frequency response up to 21 kHz. This enables the laptops to play and record high frequency sounds in 16$\sim$21 kHz band without loss of signal. The proposed acoustic mesh network achieves 20 bps data rate at 20 m maximum distance.

1.2.3 Commercial Services

Commercial services using aerial acoustic communication technologies have also been introduced by a number of start-up companies [31–33]. SonicNotify has developed small beacon devices that repeatedly transmit high frequency inaudible tones [31]. The inaudible signal of SonicNotify is using 18.5$\sim$20.5 kHz band, and it has similar waveforms to BFSK modulated signals, but the details are not open in public. SonicNotify also provides a software development toolkit (SDK) for mobile app developers to build their own application that detects the inaudible signal of their beacons. For example, an owner of a grocery market can install a number of SonicNotify beacons on the rack to broadcast product information and/or discount events. The owner then builds an application to be installed in customers' smartphones using the SDK of SonicNotify. The customers with the application can easily find a product and/or discount coupons through the inaudible acoustic signal.

ShopKick is another start-up that is collaborating with retail stores such as Macy's and BestBuy [32]. The service of ShopKick (with the same name of the company) aims to provide special offers to shoppers via inaudible acoustic signals. ShopKick installs ultrasound transmitters at the entrance of stores, and the transmitter repeatedly broadcast a unique ID of the store. Customers can download ShopKick application through public application markets such as Google Play Market or AppStore in their smartphones. The customers with ShopKick application can receive the inaudible acoustic signal at the entrance when they visit retail stores; they are noticed by coupons and other store information in their smartphones.

Naratte is a company that introduce Zoosh as a substitute to NFC. This technology is currently being used by Verifone as a mobile payment service for taxis, referred to as Way2Ride [33]. Verifone installs small tablet PCs at the back seats of taxis. The display of the table PC presents usage of Zoosh service, and the speaker of it plays inaudible acoustic signal. A customer can pay the taxi fee using his/her smartphone with Zoosh application by tagging the smartphone at the tablet PC - the smartphone that received the acoustic signal notices that the customer arrived at the destination, and the taxi fee is payed from electric wallet.

1.2.4 Limitations of Previous Work

The aforementioned previous work has a few limitations. In [24–26], the sound signals of aerial acoustic communication is within the audible band. This has two main drawbacks: (1) the sound signals heard by human beings can be disruptive, and (2) the background noise in a typical environment deteriorates signal performance. Acoustic signals designed in [27, 28] require an existing audible sound source to embed the signal, and hence, are not stand-alone techniques. That is, the acoustic signal cannot be used in silent environments such as libraries and museums. Also, the inaudible sound signals generated in [29,32,33] require a special transmitter which limits the versatility of the aerial acoustic communication. The designed acoustic signal cannot be played unless the there are ultrasound transmitters provided by specific companies.

In addition, the modulation schemes in many of these examples have a limited transmission range of up to a few meters. In the worst case, the signal can reach only a few tens of centimeters [26, 33]. The only exception is [30] that can be delivered up to 20 m. However, this is available since the speaker and microphone of laptops used in the testbed have a very flat frequency response in the high-frequency band. It should be noted that it is not the case in most audio interfaces as presented in Section 2.2.2. Using audio interfaces with severely selective frequency response can partly lose the acoustic signal, and hence, the performance eventually degrades.

1.3 Main Contributions

In this dissertation, we focus on inaudible aerial acoustic communication for off-the-shelf audio interfaces in long-range indoor environments. Specifically, we consider the scenario that a speaker repeatedly broadcasts short-length ID, and a smart device decodes the received acoustic signal. By adopting *chirp signal* that was originally used in radar applications [34], we drastically extend the communication range as well as support most speakers and smart devices with very selective frequency responses. We also propose a software digital modem architecture for smartphones that can efficiently demodulate the chirp signal. The proposed receiver uses FFT and Hilbert transform in order to efficiently find the envelope peak of chirp auto correlation. Auxiliary features for the chirp signal receiver are introduced in the dissertation. Backend query makes up the low data rate of the proposed receiver, and signal detection algorithm supports low power reception of signals. Finally, the proposed chirp BOK receiver is applied to two representative applications, namely, (1) content recognition of TV programs, and (2) indoor location tracking in subway. More into detail, we summarize the contributions of the dissertation as follows.

1.3.1 Acoustic Channel and PHY Analysis

The characteristics of aerial acoustic channel in inaudible band and performance of PHY modulation schemes are presented based on measurement results. Experimental results show that the indoor acoustic channel has large delay spread, up to 40 msec for -6 dB threshold. We investigate the frequency response of smart devices' audio interfaces in an anechoic chamber, and discover that their frequency response are severely selective especially over 20 kHz band. We also evaluate the bit error rate (BER) of PSK and FSK modulation schemes in long-range indoor environment. Based on the analysis of the experimental results, we claim that the PSK and FSK modulation schemes are hard to work in indoor acoustic channel. Specifically, PSK signals suffer from delay spread and drifting of local oscillator, and FSK does from the frequency selectivity of speaker and microphones.

1.3.2 Receiver Design for Acoustic Chirp BOK

We find that *chirp signal* can be a solution to cope with the large delay spread and severe frequency selectivity of acoustic channel. In this dissertation, chirp binary orthogonal keying (BOK) scheme is adopted for binary modulation of digital signal. The demodulation of chirp BOK signal is relatively simple and does not require tight synchronization of transmitter and receiver. We design a receiver architecture including envelope detector for efficient demodulation of matched filter for the chirp BOK modem in smart devices. The proposed envelope detector is equivalent to matched filter, while convolution of matched filter is substituted by FFT in order to reduce the computing complexity. Extensive experimental results in indoor environments present that the proposed chirp BOK modem can receive inaudible acoustic signal up to 25 m distance for various smart devices.

1.3.3 Applications of Chirp BOK Receiver

The proposed chirp BOK signal achieves approximately 16 bps data rate in practice. This is enough to deliver short ID information to neighboring smart devices through acoustic signal, but not for the applications that require higher data rate. We implement a backend server to compensate for the low bit rate. Smart devices connected to the Internet can fetch additional information related with the received 16 bit ID. In order to support low power operation of the chirp signal receiver, we additionally introduce a signal detection algorithm. When the receiver judges that there is no signal in the recorded audio, it can skip the following signal processing, and hence, computing power is saved. We also present two representative application examples of the proposed chirp BOK receiver: (1) content recognition of TV programs, and (2) indoor location tracking in subway. Our in-lab tests prove that the chirp BOK signal can be delivered through TV network which uses lossy audio encoding such as Dolby Digital (AC3) codec. Field test results in Seoul subway trams and stations also show that the legacy PA systems composed of old audio amplifiers and loud speakers can effectively deliver location information to the smart devices with an acceptable margin of bit error.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we present measurement results of the characteristics of indoor acoustic channel and audio interfaces of smart devices. As the BER performance of PSK and FSK modulation schemes degrades due to the measurement results above, we adopt *chirp signal* to overcome the delay spread and frequency selectivity.

Chapter 3 presents the software modem architecture specifically designed for efficient demodulation of chirp signal. System parameters of the proposed modem is based on the measurement results, and hence, the propose receiver works in practice. We evaluate the reception probability of the proposed receiver, and validate its transmission range in indoor acoustic channel.

Chapter 4 introduces the backend server architecture to compensate for the chirp BOK receiver, as well as signal detection algorithm to save energy consumption for signal recording in smart devices. We present lab test and field test results of application examples for TV content recognition and indoor location tracking, respectively.

Finally, Chapter 5 concludes the dissertation with the summary of contributions and discussion on the future work.

Chapter 2

Acoustic Channel and PHY Analysis

2.1 Introduction

Typical audible frequency range of human is said to be from 20 Hz to 20 kHz, where the detailed hearing capability is differ by person (Fig. 1.2). In this dissertation, we aim to design *inaudible* acoustic signal. Note that acoustic communication transmits information through sound wave. Hence audible signal bothers the people nearby the transmitter as its sound can be heard. The environmental background noise even deteriorates the performance of acoustic communication.

In case of the indoor acoustic channel, we experience echoes during conversation in closed rooms. Intuitively, one can expect severe multi-path propagation of sound waves in indoor acoustic channel. Numerous researchers in acoustics have been studying the characterization of room impulse responses [35–37], and report the delay spread of various room impulse response. The acoustic signal therefore suffers from this multi-path propagation in indoor environments,

Another challenge of indoor acoustic communication is the frequency selectivity of audio interfaces. Various speakers and microphones used in acoustic communication can have different frequency responses that varies by models. It should be noted that the audio interface with flat frequency response up to 22 kHz are rare and very expensive. Therefore smart devices cannot afford such hi-fidelity (HiFi) audio interfaces. As to be shown in Section 2.2.2, speakers and microphones of smart devices have severely selective frequency response. This also distorts the transmission and reception of the acoustic signal.

As reported in [1, 2], PSK and FSK modulation schemes have been used for the underwater acoustic communications. This is the same for the aerial acoustic communications; the major modulation schemes in the previous study were PSK and FSK. PSK demodulations require precise synchronizations at the receiver by implementing components for carrier synchronization and phase estimation using phase locked loop (PLL). These components might fail in the aerial acoustic communication due to the large delay spread. The sampling frequency offset caused by the instability of local oscillator further degrades the receiver performance, especially for PSK that demands coherent demodulation.

FSK with non-coherent demodulation can be another option for aerial acoustic communication. Note that, however, the frequency tones of M-ary FSK can partly be lost due to the frequency selectivity of speaker, air-medium, and microphone. For example, if a smart device's microphone cannot record above 21 kHz band, then the frequency tones above 21 kHz band is not available. Moreover, one cannot find commonly-available frequency band for all types of devices as the frequency selectivity of various smart devices are too diverse as shown in Section 2.2.2.

We adopt *chirp signal* to resolve the challenges above in indoor aerial acoustic communication. Different from PSK and FSK signals with constant frequency, a chirp signal has time-varying frequency. Therefore the transmission and reception of chirp signal is available for all types of devices in spite of their selective frequency response.

The demodulation of chirp signal can easily be implemented via *matched filter* (or equivalently, *correlator*), and it can asynchronously detect the signal. So the receiver does not require precise synchronization like PLL in PSK, and drifting of local oscillator rarely degrades the receiver performance.

In this chapter, we theoretically and empirically define the frequency band for inaudible aerial acoustic communication. Given the operational frequency band, we measure the frequency responses of speakers and microphones of various smart devices in an anechoic chamber. We also measure the impulse response of typical indoor acoustic channel to verify its delay spread. Based on these observations, we evaluate and analyze the PHY modulation schemes, PSK and FSK. The performance of PSK degrades due to the delay spread of acoustic channel and unstable local oscillator of smart devices. FSK signals are hard to be delivered over severely frequency-selective acoustic channel due to the frequency response of speakers and microphones. In order to cope with the severe frequency selectivity and phase distortion of the indoor acoustic channel composed of speaker, air medium, and microphone, we adopt chirp signal for indoor aerial acoustic communication. Experimental results presents that chirp BOK modulation outperforms PSK and FSK in a long-range indoor acoustic channel.

The rest of this chapter is organized as follows. Section 2.2 studies the audible range of human hearing system, and measure the delay spread of the indoor acoustic channel as well as the frequency response of speakers and microphones. In Section 2.3, we evaluate the performance of PSK and FSK modulation schemes in indoor acoustic communications through experiments. Discussions on the reasons of performance degradation of PSK and FSK leads us to the adoption of chirp BOK modulation. Section 2.4 presents experimental results on the BER of PSK, FSK, and chirp BOK modulations. Finally Section 2.5 summarizes the chapter with short conclusion.

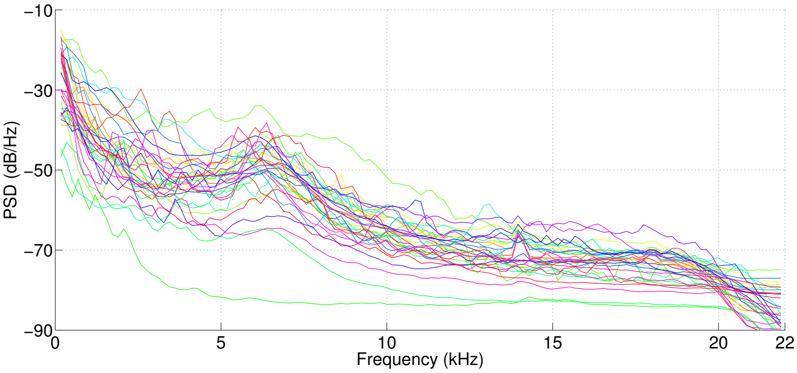


Figure 2.1: Power spectral density of environmental background noise in various routine activities.

2.2 Characteristics of Indoor Acoustic Channel

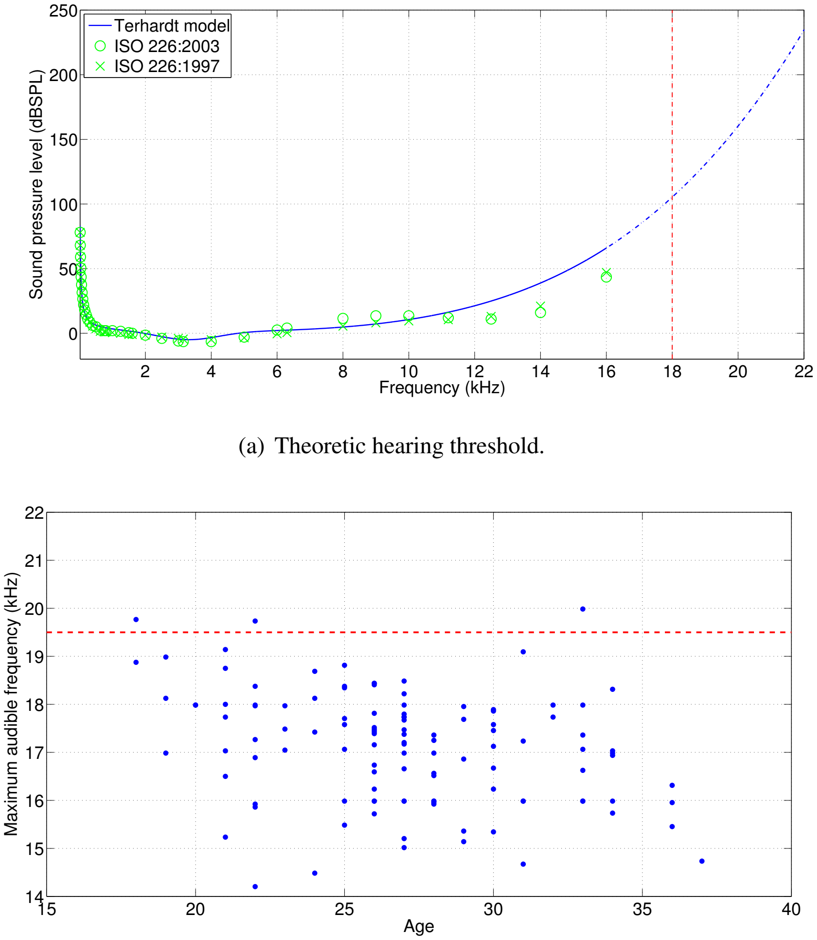
2.2.1 Hearing Threshold of Human

As we aim to design *inaudible* acoustic signal, we theoretically and empirically evaluate the maximum audible frequency of human ears. Designing an inaudible acoustic signal is important not to bother humans around transmitters. Moreover, inaudible band over 20 kHz (to be specified in the followings) is relatively free from environmental background noise. Figure 2.1 present the power spectral density (PSD) of environmental background noise measured in 37 routine activities such as watching TV at home, driving a car, working at office, shopping in a retail store, etc. We observe that the environmental background noise is not significant over 20 kHz.

Figure 2.2(a) presents the hearing threshold in literature, measured at 1 m distance [38–40]. ISO 226 presents equal-loudness contour, for which a listener perceives the same loudness level, up to 16 kHz [41, 42]. The values of ISO 226 in Fig. 2.2(a) depict 0 phon values, i.e., human can barely perceive the existence of sound. The difference of 1997 and 2003 models is the measurement condition, i.e., under free-field and diffuse-field test [41, 42], where the difference is not significant. Terhardt model is a fitted curve for the ISO 226 recommendations, where dashed lines over 16 kHz shows estimated values [43]. From Fig. 2.2(a), we can find that the sound pressure level that is required to stimulate human cochlear varies over frequency. At 18 kHz (vertical dashed line) for instance, the sound pressure level should be at least 100 dB-SPL to be heard by human. Note that this is equivalent to the loudness of a heavy metal concert, and hence, we claim that the signal over 18 kHz is mostly inaudible in typical environments.

To be more specific, we conduct an experiment with 134 people to determine the maximum frequency they can hear through earphones. We build a test application to measure the maximum audible frequency. The application tries to find the maximum audible frequency in a binary-searching manner. A testee listens to 14∼22 kHz tones through earphones, where the tones are sequentially played with a specific order. We need to prevent false positive, i.e., a testee claims that he/she perceived a high frequency tone that is actually not played. For the audibility test of a specific tone, we play the tone for randomly multiple times and the testee checks how many times he/she listens to the tone. From Fig. 2.2(b), it is shown the maximum audible frequency generally decreases as the age of testee increases. We also see that the most people (131 out of 134) cannot hear over 19.5 kHz. Note that the tones are played through earphones that is very close to cochlear. If the tone were played through loud speakers in distance, then the three exceptions would not happen. In this dissertation, we choose 19.5 kHz as the lower frequency bound of inaudible sound signal.

Now we define the upper bound of inaudible signal. The maximum playable frequency of an audio interface is limited by its sampling rate F_s; the maximum playable frequency is $F_s/2$ considering the Nyquist theory. Table 2.1 summaries the typical



(a) Theoretic hearing threshold.

(b) Empirical hearing threshold.

Figure 2.2: Theoretic and empirical hearing threshold of human ears.

Table 2.1: Typical Audio Sampling Frequencies and Applications

Sampling Frequency	Maximum Playable Frequency	Applications
8,000 Hz	4,000 Hz	Phone calls, Walkie-talkies
16,000 Hz	8,000 Hz	Wide-band vocoders, VoIP
44,100 Hz	22,050 Hz	CDs, MPEG audios
48,000 Hz	24,000 Hz	Professional digital video equipments
96,000 Hz	48,000 Hz	DVD audios, Blu-ray Discs

sampling rates of audio interfaces. Note that the sampling frequency of most audio interfaces is 44.1 or 48 kHz.[1] Accordingly we set the upper frequency bound as 22 kHz.

2.2.2 Frequency Response of Various Audio Interfaces

We measure the frequency responses of various audio interfaces in an anechoic chamber, located at Institute of New Media and Communications, Seoul National University. The measurement is based on linear frequency sweep (LFS) signal method proposed in [44, 45]. Note that the acoustic channel is composed of speaker, air medium, and microphone. We can ignore the frequency selectivity of the propagation of sound wave through the air medium, as the measurement is conducted in an anechoic chamber. However, the recorded sound passes through the frequency response of two different devices; speaker and microphone. Hence we need a reference speaker and reference microphone that has flat frequency response. This can minimize the effect of the frequency distortion of the opposite side. We use Genelec 6010A active speaker [46]

[1] In case of smart devices with Android and iOS operating systems (OSs), one can set the sampling frequency of speaker and microphone on initialization process.

and Audix TM1 condenser microphone [47] as the reference speaker and microphone, respectively.[2]

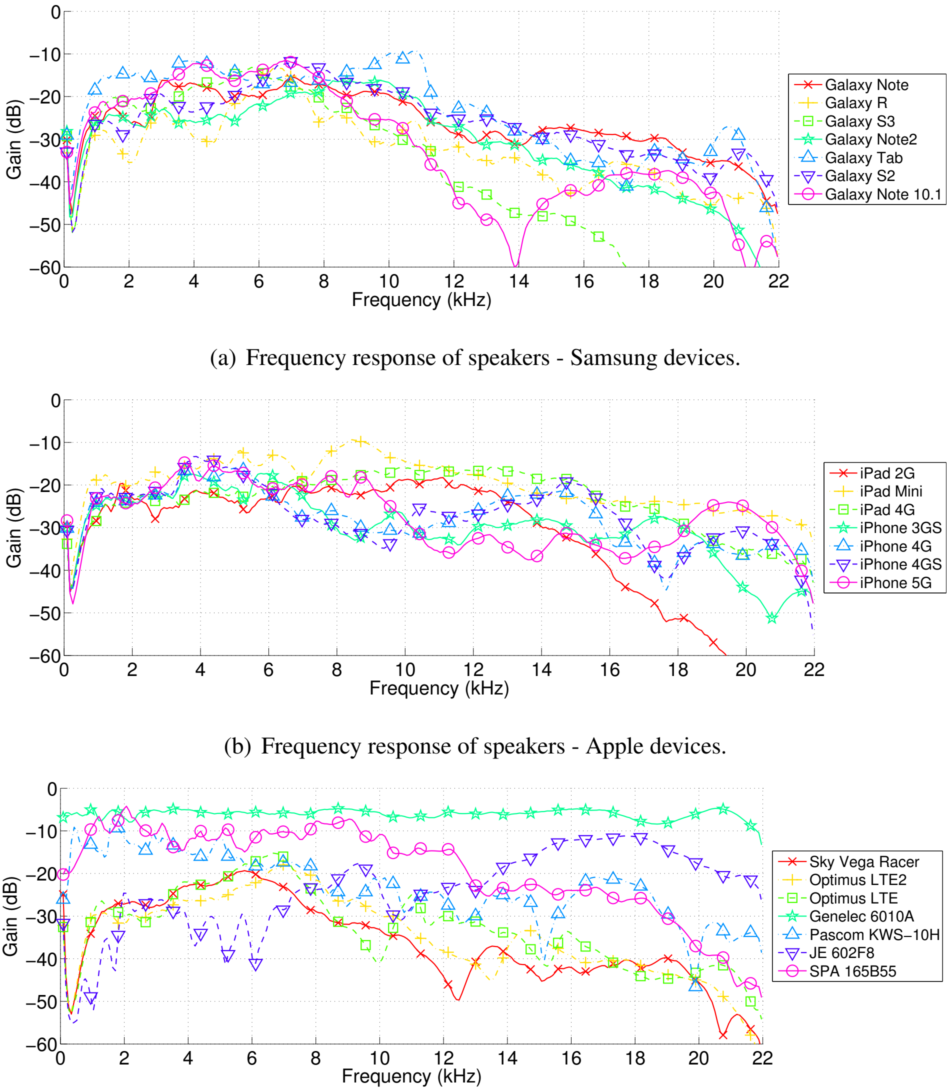
The process of measurement is as follows [44, 45]. Suppose that we measure the frequency response of a smart device's microphone. We first place the reference speaker and smartphone in the anechoic chamber with 1 m distance. The membranes of speaker and microphone are set to a coaxial line. Then we play 50 sec sine signal linearly sweeping from 20 to 22,000 Hz using a reference speaker, and record the sound using smart device in 16 bit pulse coded modulation (PCM) format with 44.1 kHz sampling rate.[3] After the play ends, we read the recorded wave file into a laptop, and post-process it using MATLAB to calculate the frequency response of microphone. As the sweep signal has all frequency components in 20 to 22,000 Hz, PSD of the wave file is equal to the frequency response of the microphone. Measuring the frequency response of a speaker is the same process, except that the signal is played and recorded using the testing speaker and reference microphone, respectively.

We present the measurement results of various smart devices and off-the-shelf loudspeakers in Figs. 2.3 and 2.4. In Fig. 2.3(c), we verify that the reference speaker (Genelec 6010A) has flat frequency response as in the data sheet. The main observations of Figs. 2.3 and 2.4 are as follows.

Frequency selectivity of speakers. We observe that the frequency selectivity is severe for speakers rather than microphones. Infrasound (under 20 Hz) are rarely played in smart devices due to the size of the vibrating membrane is too small — an woofer with large and thick cone is needed to play the infrasound. The frequency selectivity is the same for commercial loud speakers (Fig. 2.3(c)).

[2] One can find the frequency response of these audio interfaces in the data sheet of the corresponding webpage [46, 47].

[3] This is a typical wave file format.

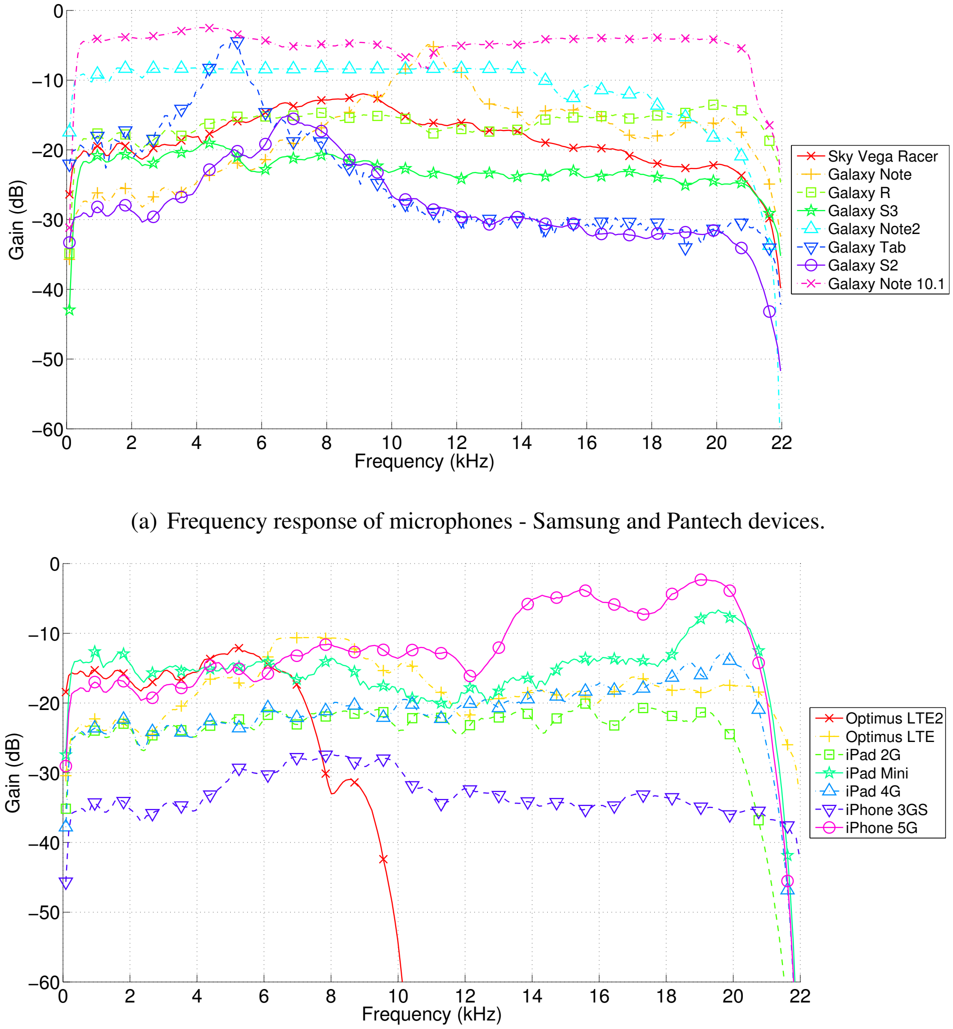


(a) Frequency response of speakers - Samsung devices.

(b) Frequency response of speakers - Apple devices.

(c) Frequency response of speakers - Other devices.

Figure 2.3: Frequency response of speakers of various smart devices and off-the-shelf speakers.



(a) Frequency response of microphones - Samsung and Pantech devices.

(b) Frequency response of microphones - Apple and LG devices.

Figure 2.4: Frequency response of microphones of various smart devices.

Nonetheless the frequency band from 19.5∼22 kHz can be played through commercial loud speakers with 20∼30 dB loss compared with audible band.

Frequency selectivity of microphones. The microphones of smart devices has relatively flat frequency response up to 20 kHz. However, the frequency selectivity increases in 20∼22 kHz band. Therefore the maximum recordable frequency significantly varies over device. The inaudible signal needs to cope with the loss of microphones.

Exceptional cases In case of Optimus LTE2 of LG Electronics, the microphone cannot record the sound over 10 kHz band. LG Electronics answers that the OS of Optimus LTE2 cuts off the sound over 10 kHz of the microphone input in order to reduce the howling effect.

From the observations, we conclude that inaudible acoustic communication in 19.5∼22 kHz band is mostly feasible, though it has to overcome the severe frequency selectivity of audio interfaces.

2.2.3 Delay Spread of Acoustic Channel

Now we measure the impulse responses of aerial acoustic channels in a typical indoor office environment. We again adopt LFS signal method in [44,45] in order to measure the indoor acoustic impulse response. We assume the transmission from a loud speaker to a smart device. Two different communication environments are supposed in this experiment. One is a good case; loud speaker has flat frequency response, and there exists an line-of-sight (LOS) channel between the speaker and smart device. The other is a bad case; loud speaker has selective frequency response, and the acoustic channel does not have an LOS link (NLOS).

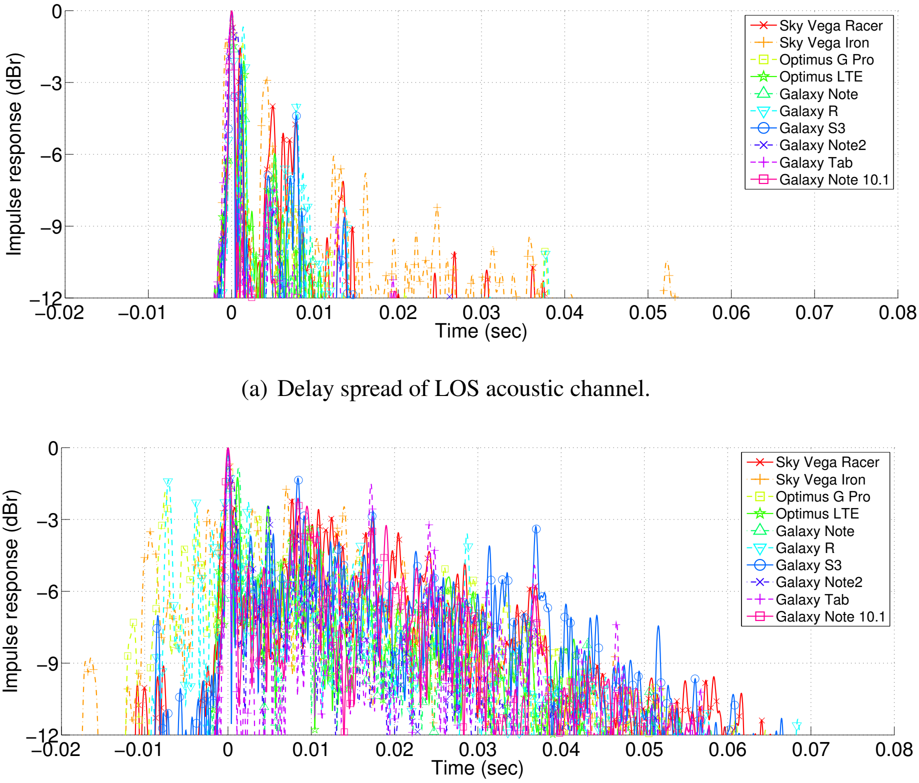
We emulate the communication scenarios above as follows. We play LFS signal using Genelec 6010A and JE602F8 speakers for LOS and non-LOS (NLOS) acoustic channel, respectively, in a typical indoor office environment. The frequency responses of the loud speakers are shown in Fig 2.3(c). Then we record the LFS signal using various smart devices which are randomly located within 5 m distance from the speaker. The LOS link between the speaker and smart device is controlled by placing partition screens in between them. After reading the recorded audio (wave) file, MATLAB is used to calculate the auto-correlation of the received audio and transmitted LFS signal. Note that this is the same process that radar uses to measure the distance between the transmitter and an object.

Figure 2.5 depicts the obtained auto-correlation plots in for 10 different smart devices, where auto-correlation is normalized to the maximum peak of each experiment. We observe that the indoor aerial acoustic channel suffers from severe multi-path propagation. In case of LOS scenario (Fig. 2.5(a), delay spread spans approximately 8 msec for -6 dBr threshold. In NLOS case (Fig. 2.5(b), however, the multi-path delay extends up to 40 msec for the same -6 dBr threshold.

This is significantly large delay spread, compared to the few μsec delay spread of wireless communications. Therefore one needs guard interval between symbols or a channel equalizer in order to avoid inter-symbol interference (ISI).

2.3 Revisit of Existing Modulation Schemes

In the previous section, we investigated the characteristics of indoor aerial acoustic channel. The inaudible signal suffers from the frequency selectivity of speaker and microphones, as well as the large delay spread of indoor acoustic channel. Now we revisit two widely used digital signal modulation schemes, PSK and FSK, through exper-



(a) Delay spread of LOS acoustic channel.

(b) Delay spread of NLOS acoustic channel.

Figure 2.5: Impulse responses of aerial acoustic channels in indoor office environment, measured using LFS signal.

imental study. As those PHY modulation schemes are hard to work in the indoor aerial acoustic channel, we introduce chirp signal-based binary orthogonal keying scheme in this section.

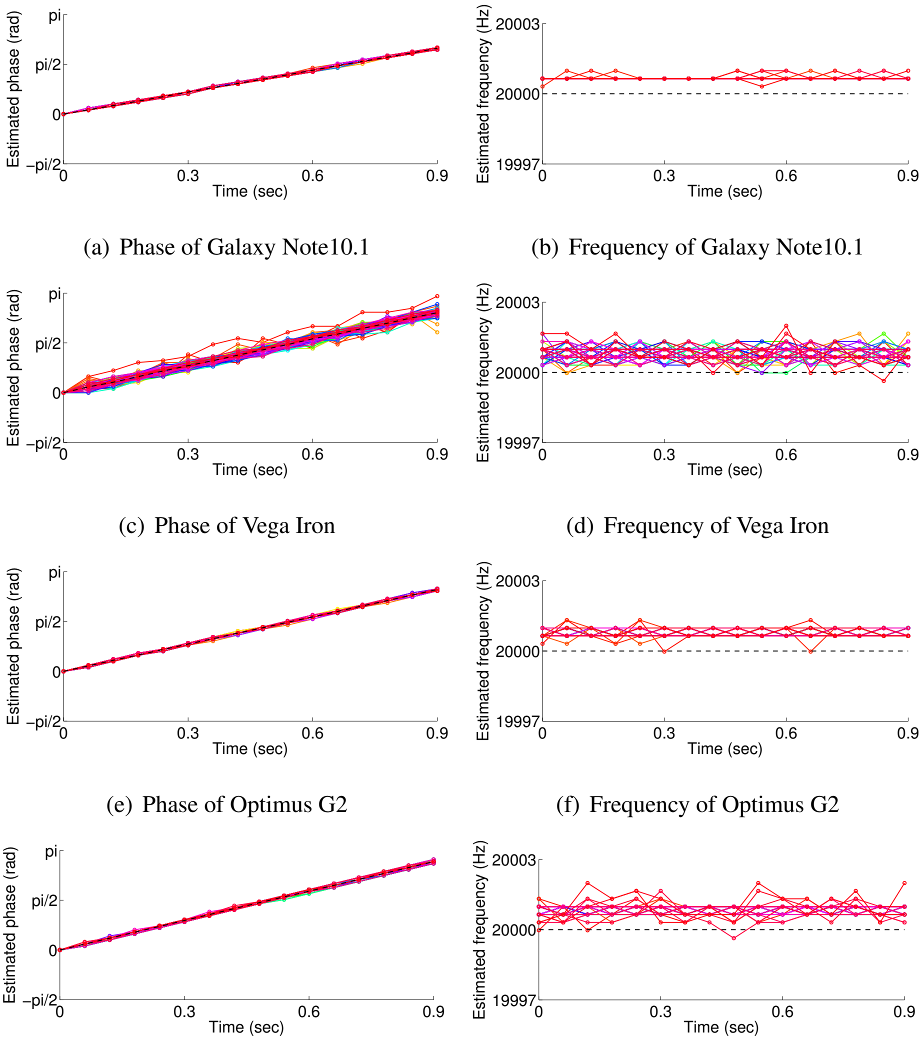
2.3.1 Case Study: Phase Shift Keying

It is well known that coherent demodulation of PSK signals requires exact carrier, sampling, and symbol timing recovery [48–50]. This is the same for the PSK modulation in aerial acoustic communications. We study the performance of PSK modulation in indoor acoustic channel through experimental study as follows.

Phase Drift in Recording

We first check the phase drift at the receiver side due to the *frequency offset*. Genelec 6010A reference speaker plays a constant cosine tone with 20 kHz center frequency. Then we record the tone for 1 second using various smart devices in 5 m distance LOS and NLOS links. From the recorded audio, we sample 20 msec symbols every 60 msec interval and estimate its phase. In an ideal situation, i.e., there is no phase distortion, the estimated phase should remain constant regardless of the sampling timing.

In practice, however, the estimated phase varies over time. We repeat the experiment above for 100 times, and plot the estimated phase and frequency of each 20 msec samples in Fig. 2.6. We observe that the estimated phase changes over time (figures in the left column of Fig. 2.6). This is due to the sampling offset of transmitter and receiver. The speaker plays 20 kHz frequency tone, but the sampling frequency of the transmitter and receiver is not precisely synchronized. Hence the estimated frequency of the received audio is not exactly 20 kHz, as shown in the figures in the right column of Fig. 2.6.



(a) Phase of Galaxy Note10.1

(b) Frequency of Galaxy Note10.1

(c) Phase of Vega Iron

(d) Frequency of Vega Iron

(e) Phase of Optimus G2

(f) Frequency of Optimus G2

(g) Phase of Galaxy Note3

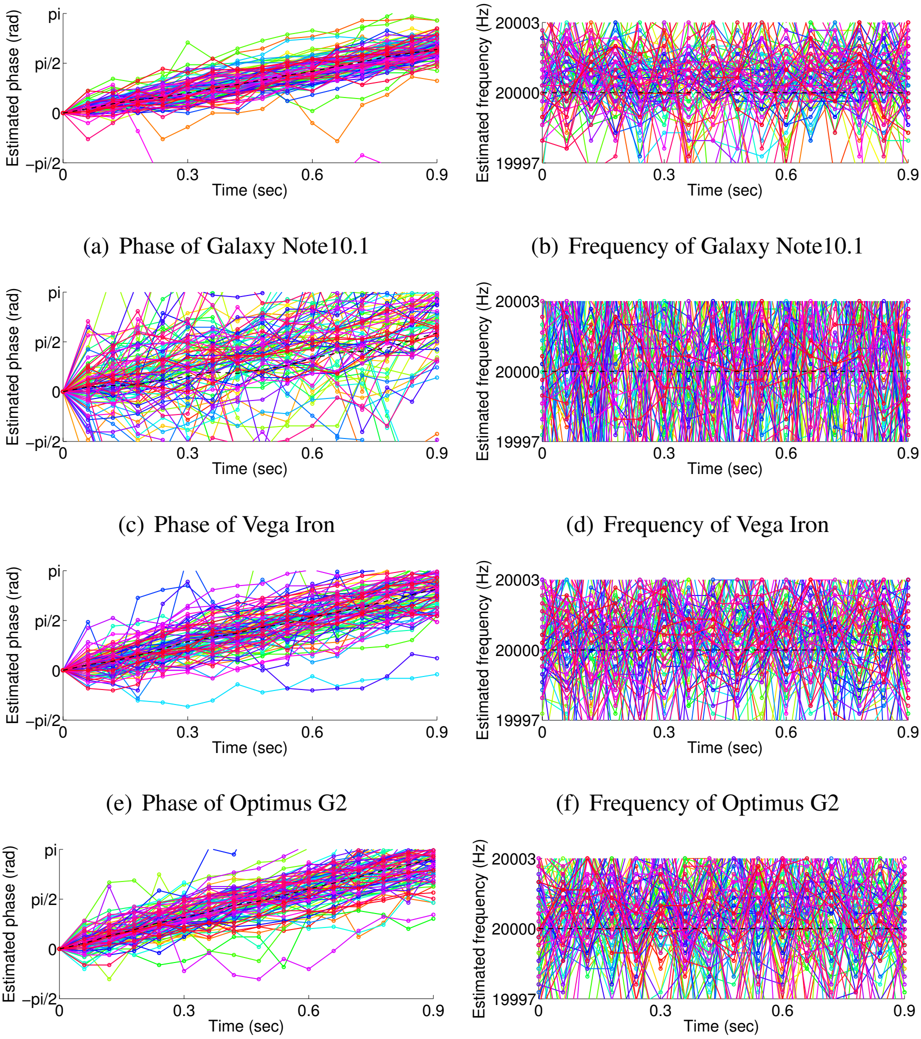
(h) Frequency of Galaxy Note3

Figure 2.6: Estimated phase of recorded audio using various smart devices in 5 m LOS channel.

It should be noted that the drifting of estimated phase is not static in Fig. 2.6. That is, each model has different level (slopes in the figures) of drifting. Each experiment also has different slopes for the same device. We claim that this is due to the instability of local oscillators the devices. Note that the audio sampling frequency of 44.1 kHz is assembled by downsampling the clock of local oscillators. Depending on the downsampling process of each device, the resulted audio sampling frequency can fluctuates. We cannot measure the exact level of instability as the internal downsampling process is not open in public, but we qualitatively observe the instability in Fig. 2.6. For instance, the audio sampling frequency of Vega Iron is unstable, so the variance over experiments is greater than other devices.

The phase drift in LOS channel observed in Fig. 2.6 has a specific pattern. Therefore the receiver can calculate the phase drift over time by using training signals. For example, one can add a tone signal as a preamble at the beginning of each PSK signal transmission. The receiver can calculate the phase shift in a function of time using the received preamble, and then compensate for the changes in the following data transmissions.

In NLOS channel, however, it is very hard to track the phase drift due to the channel fluctuation. Figure 2.7 presents the phase and frequency estimation results with the same experiment except the channel status; peoples are moving around the speaker and smart device so that the acoustic channel varies over time. Note that the phase of the received tone randomly varies over time. The receiver cannot track the phase changes in this case, so that the compensation is likely to fail. In the following section, we evaluate the performance of binary PSK (BPSK) modulation in order to validate this observation.



(a) Phase of Galaxy Note10.1

(b) Frequency of Galaxy Note10.1

(c) Phase of Vega Iron

(d) Frequency of Vega Iron

(e) Phase of Optimus G2

(f) Frequency of Optimus G2

(g) Phase of Galaxy Note3

(h) Frequency of Galaxy Note3

Figure 2.7: Estimated phase of recorded audio using various smart devices in 5 m NLOS channel.

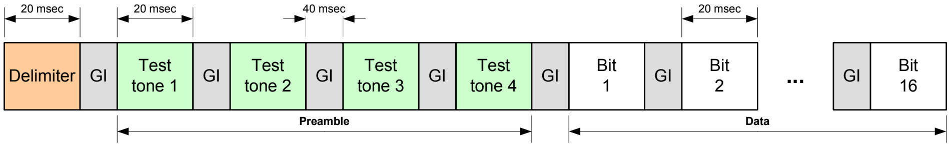


Figure 2.8: Frame structure of BPSK signal.

BPSK Demodulation using Preamble

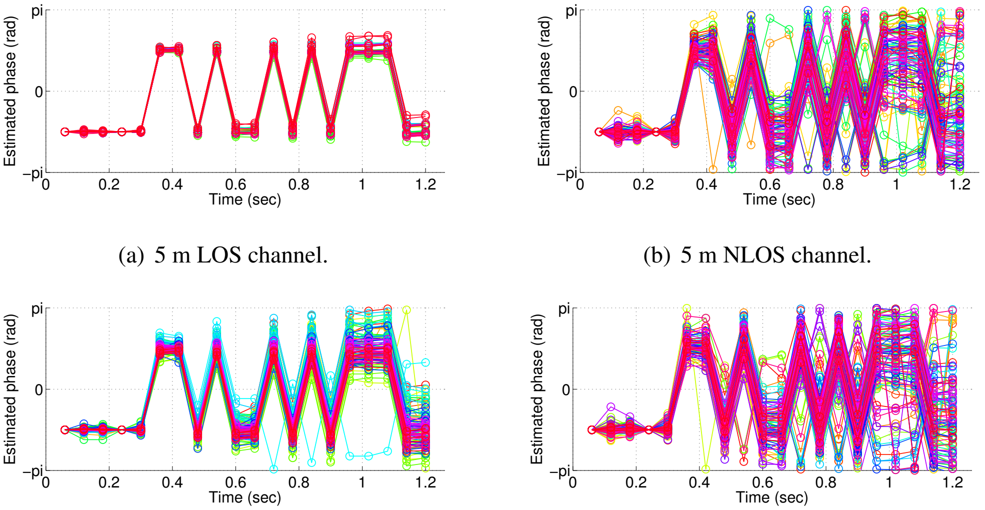
We emulate the BPSK modulation in aerial acoustic communication. We deploy Genelec 6010A reference speaker and smart devices in 25 m corridor with 2.5 m width and 3 m height. The distance between the speaker and smart device is set to 5 m and 25 m to test short and long range communication, respectively. The reference speaker plays a BPSK modulated signal in Fig. 2.8. Then the smart device record the BPSK frame in wave file format, and MATLAB analyzes the recorded audio to demodulate the BPSK signal. The demodulation process works as follows.

1. **Find delimiter.** We use an LFS signal to find the beginning of the frame. Finding the peak of auto-correlation plot yields the ending point of the delimiter.

2. **Track the phase drift.** The delimiter is followed by a preamble, composed of four test tones with the same phase. By estimating the phases of the four test tones, the receiver can calculate the phase drift over time.

3. **Demodulate data bits.** 16 bit data is modulated into BPSK signal. Specifically, bit 0 is mapped to phase $-\pi/2$, and bit 1 is to phase $\pi/2$. The receiver first estimate the received phase of each symbol, and then compensate the phase drift over time based on the result from the preamble.

4. **Guard interval (GI).** We add 40 msec guard intervals in between each symbol. This is for removing ISI observed in Section 2.2.3.



(a) 5 m LOS channel.

(b) 5 m NLOS channel.

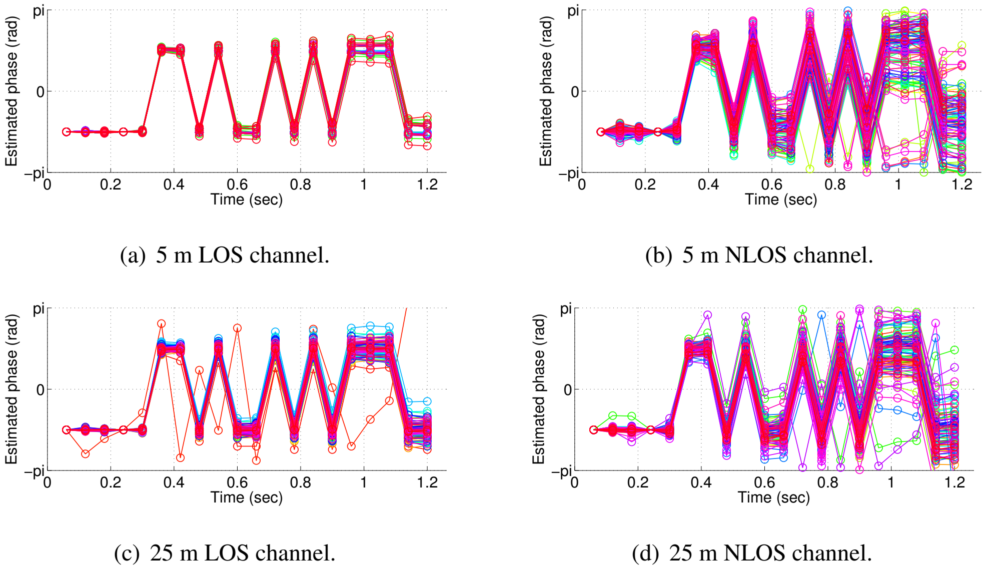
(c) 25 m LOS channel.

(d) 25 m NLOS channel.

Figure 2.9: Demodulation results of BPSK signals in Galaxy Note10.1.

Figures 2.9, 2.10, and 2.11 present the demodulation results of Galaxy Note10.1, Galaxy Note3, and Vega Iron devices, respectively, repeated by 100 runs. These plots can be considered as an eye-opening diagram [50]; the clearer the graphs are, the better its BER is. The first four symbols correspond to the preamble with phase $-\pi/2$. The revised phases of the preamble is expected to be constant in an ideal case (e.g., Fig. 2.9(a)). The last 16 symbols denote the estimated phases of data symbols, revised by the phase drift tracked from the preamble. Note that this phase compensation works in LOS channels (e.g., Figs. 2.9(a) and 2.9(c)). The effect of compensation for the latter symbols degrades so that the estimated phases parries in each experiment, but it is in an acceptable margin of error. Therefore the BER (or equivalently, symbol error rate (SER)) of 5 m LOS channels in Galaxy Note3 and Galaxy Note10.1 is zero.

In case of NLOS channels, however, the SER increase as the channel fluctuation difficulties the phase tracking at the receiver. As shown in Figs. 2.9(b), 2.9(d), 2.10(b),



(a) 5 m LOS channel.

(b) 5 m NLOS channel.

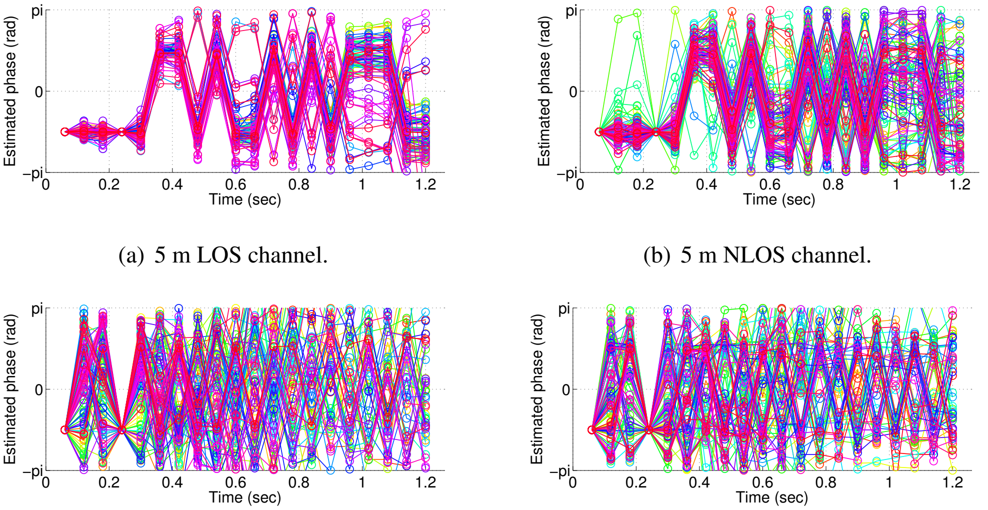
(c) 25 m LOS channel.

(d) 25 m NLOS channel.

Figure 2.10: Demodulation results of BPSK signals in Galaxy Note3.

and 2.10(d), the phase estimation error increases especially for the latter symbols. In the worst case, Vega Iron in Fig. 2.11, even the SER in 5 m LOS channel is too bad due to the instability of its local oscillator. Vega Iron cannot work in 25 m distance, regardless of the channel condition of LOS and NLOS.

The coherent PSK receiver requires much more complex modules such as PLL at the beginning of the demodulation process in order to enhance its performance. This increases the complexity of the modem, which is not appropriate for the software modems to be implemented in smart devices. We here conclude that the coherent PSK demodulation is hard to be applied for the aerial acoustic communication for smart devices due to the channel fluctuation, instability of the local oscillator, and complexity issue of the receiver. In the following section, we study the performance of non-coherent FSK modulation.



(a) 5 m LOS channel.

(b) 5 m NLOS channel.

(c) 25 m LOS channel.

(d) 25 m NLOS channel.

Figure 2.11: Demodulation results of BPSK signals in Vega Iron.

2.3.2 Case Study: Frequency Shift Keying

FSK is known to suffer from the frequency selectivity, as its constellation points of received tones gets asymmetric [51]. Suppose a binary FSK (BFSK) case using two frequency tones at 19 and 21 kHz. For the devices that cannot record over 20 kHz band due to its frequency response, the tones at 21 kHz is rarely received. Therefore the receiver can only receive the symbol with low frequency, and this deteriorates the SER. Remind that this is very likely for the smart devices as we observed in Section 2.2.2.

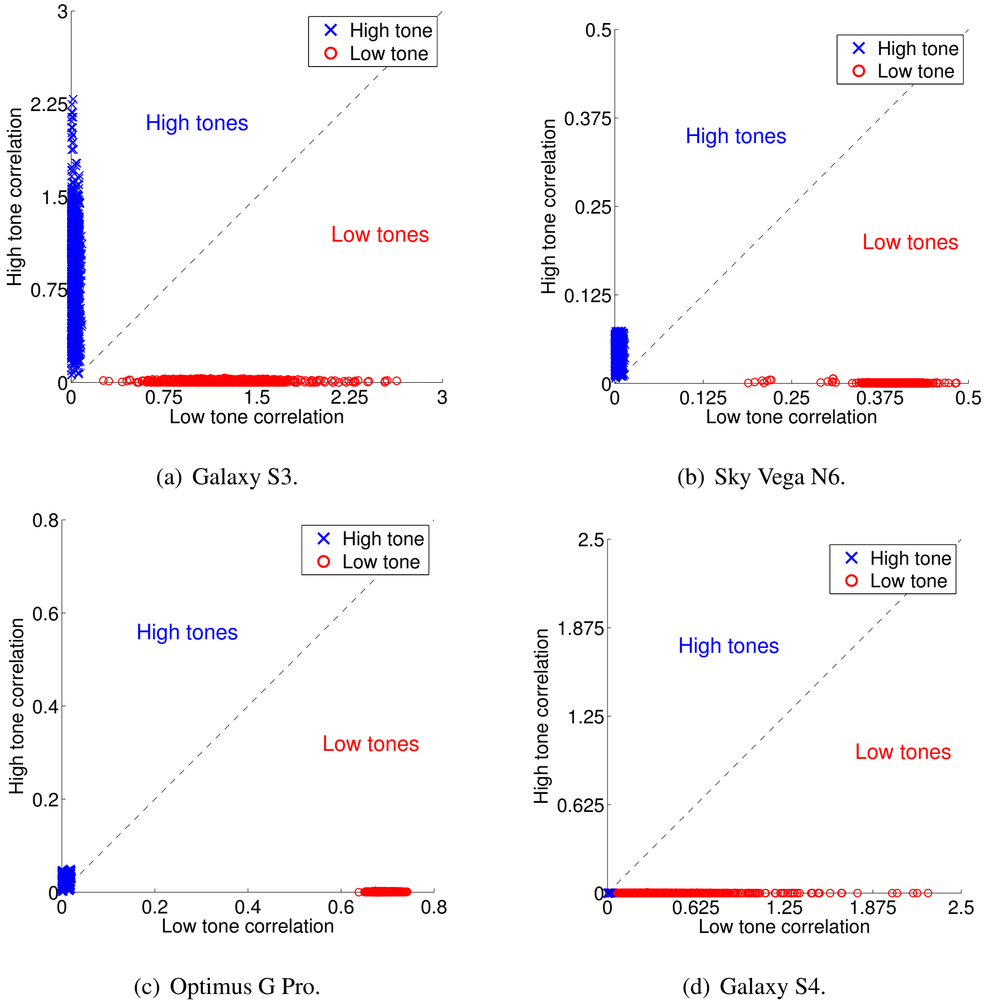
We set an experiment to validate the feasibility of FSK on aerial acoustic communication for smart devices. We set a reference speaker and smartphones in a long corridor, as in Section 2.3.1. Assuming BFSK, we sequentially play two types of cosine tones in inaudible band using the reference speaker; low frequency (19.5 kHz for bit 0) and high frequency (21 kHz for bit 1) tones. The order of transmitted symbols is the same as in Fig. 2.8, except the preamble tones — this training is not required in non-coherent FSK reception. Smart devices record each frame for 100 times at 5 m and 25 m distance, and we emulate non-coherent FSK receiver using MATLAB to present the received constellation point in Fig. 2.12.

Constellation Point of FSK

We observe in Fig. 2.12 that the capability to catch high frequency tones differs by the devices. More into detail, Galaxy S3 has relatively flat frequency response up to 21 kHz. (Recall Fig. 2.4(a).) Therefore Galaxy S3 can receive both low and high frequency tones as in Fig. 2.12(a). In case of Sky Vega N6 and Optimus G Pro, high frequency tones are partly missed due to the frequency response, so that the average strength (or correlation outputs) are asymmetric. Galaxy S4 is the worst case; most of the high frequency tones are lost, and hence, only low frequency tones are received.

The simplest form for symbol decision of non-coherent FSK receiver is directly comparing the correlation values of low and high frequency tones. The receiver decides that the frequency tone with greater correlation value is transmitted. This can be graphically presented as a decision bound of $y = x$ (dashed lines) in Fig. 2.12. Note that this decision bound likely to fail in the smart devices equipped with a microphone that has severely selective frequency response due to the asymmetric constellation points discussed above.

One can adaptively change the decision bound for the non-coherent FSK demodulation in frequency selective channel. The receiver is trained to adapt the decision bound based on the received training tones as explained in detail in the followings.



(a) Galaxy S3.

(b) Sky Vega N6.

(c) Optimus G Pro.

(d) Galaxy S4.

Figure 2.12: Empirical constellation point of non-coherent BFSK receivers for four smart devices in 5 m LOS channel.

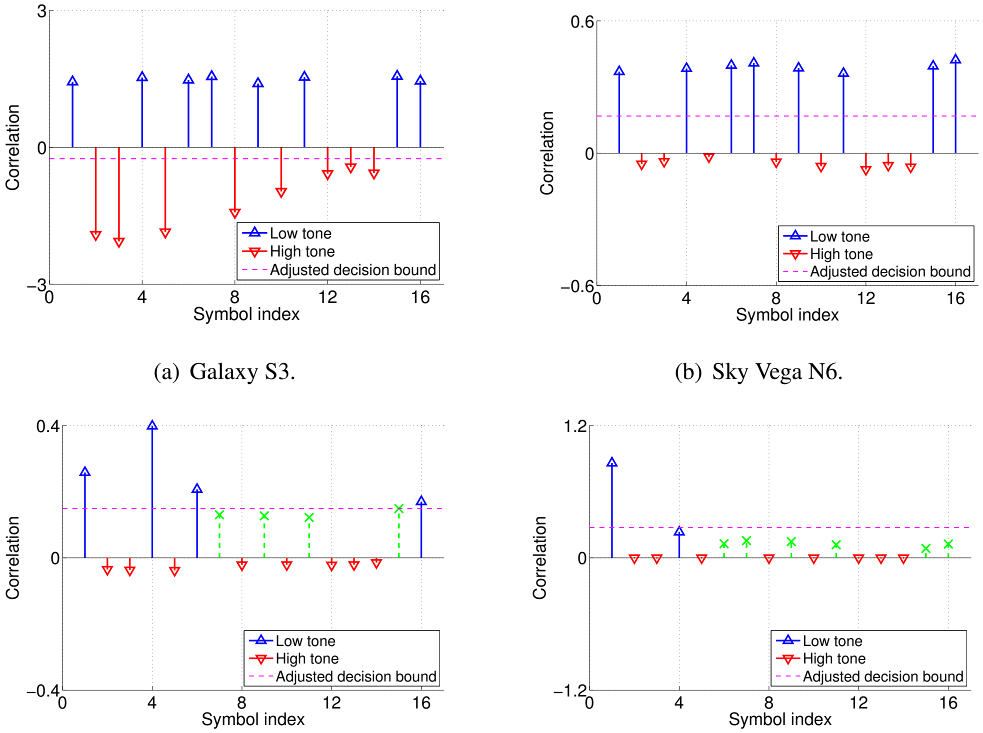
FSK Demodulation with Adaptive Decision Bound

The adaptation of BFSK decision bound is enabled by receiving a number of training tones that evenly alternates high and low frequency components. On the reception of the training bits whose transmission order is known at the receiver in off-line, the receiver can estimate the relative frequency gains of two frequency tones. Then the receiver uses revised decision bound for the following data symbol demodulations.

We emulate this decision bound adaptation of non-coherent FSK receivers and evaluate its effectiveness. The transmitter plays BFSK modulated acoustic signal. The frequencies of the first four symbols are 19, 21, 21, 19 kHz, and these symbols are used as training symbols at the receiver. The receiver adjusts the decision bound to the average difference of the training symbols. Then the following symbols are decided based on the revised decision bound.

Figure 2.13 present the experimental results on the decision bound adjustment in four smart devices. Blue and red stems denote the correlation output of low tone and high tone, respectively. Dashed horizontal line present the adjusted decision bound calculated from the first four symbols. As Galaxy S3 has almost symmetric constellation point as shown in Fig. 2.12(a), the adjusted decision bound in Fig. 2.13(a) is not very different from the original decision bound ($y = 0$ in this figure). In case of Sky Vega N6 in Fig. 2.13(b), the adjusted decision bound effectively differentiates the received tones even though the correlation values of high frequency tones are very small.

However, the adaptive decision bound does not always work as shown in Fig. 2.13(c). As the strength of the received high frequency tone is too small, the adjusted decision bound is close to half of the average correlation values of low frequency tones. It should be noted that the reasons of frequency selectivity of aerial acoustic channel is not limited to the frequency response of the microphone, but air medium itself



(a) Galaxy S3.

(b) Sky Vega N6.

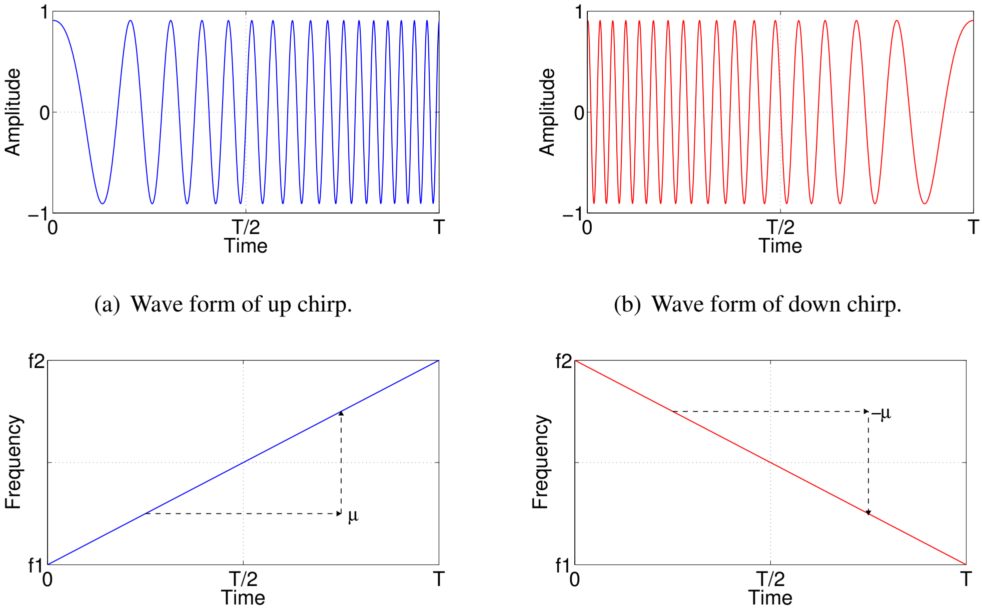
(c) Optimus G Pro.

(d) Galaxy S4.

Figure 2.13: Adaptive ecision bound of non-coherent BFSK receivers for four smart devices in 5 m LOS channel.

causes the channel frequency selectivity, too. Hence the received low frequency tones are missed (green x marks) under channel gain fluctuations as shown in Figs. 2.13(c) and 2.13(d). The signal strength of data symbols after the training tones are too small to be detected over the adjusted decision bound.

Therefore we claim that the non-coherent FSK is not applicable for indoor aerial acoustic channel due mainly to the frequency selectivity of audio interfaces. Adjusting the decision bound to overcome the frequency selectivity even fails when the frequency selectivity is too severe and acoustic channel fluctuates over time.



(a) Wave form of up chirp.

(b) Wave form of down chirp.

(c) Frequency of up chirp.

(d) Frequency of down chirp.

Figure 2.14: Wave form and frequency changes of up chirp and down chirp.

2.3.3 Chirp Binary Orthogonal Keying (BOK)

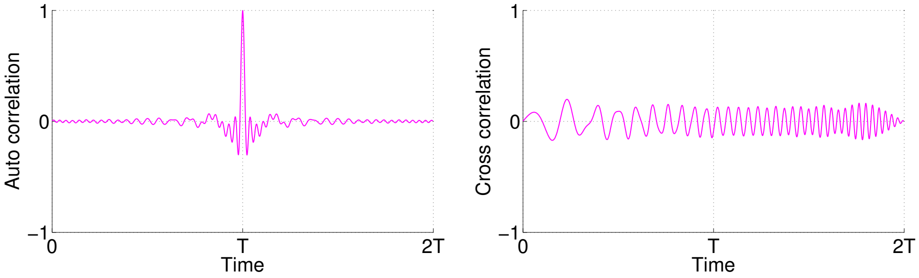
Throughout the case studies in Sections 2.3.1 and 2.3.2, we fount that the aerial acoustic channel suffers from sampling frequency offset and frequency selectivity. The signal modulation scheme for aerial acoustic communication hence should be robust to these impairments.

As a solution, we adopt *chirp signal* [34]. Different from PSK and FSK signal which have time invariant frequency for a specific symbol, the frequency of chirp signal varies over time. Up chirp and down chirp have increasing and decreasing frequencies, respectively, as shown in Figs. 2.14(a) and 2.14(b), where T denotes symbol duration. Up and down chirps use the same frequency band (Figs. 2.14(c) and 2.14(d)), and hence, the frequency selectivity of acoustic channel distorts both up and down chirp symbols in a symmetric manner. The slope of Fig. 2.14(c) is referred to as *chirp sweep rate*, and it will be explained in detail in Chapter 3.

Chirp binary orthogonal keying (BOK) modulation maps the digital bit 1 and 0 to up and down chirps, respectively. Chirp BOK can also map the symbols and bits in the reverse order, i.e., bit 1 for down chirp and bit 0 for up chirp. In this dissertation, without loss of generality, we assign bit 1 and bit 0 to up and down chirps, respectively. Figure 2.16(a) depicts the mapping of bit sequence [1, 0, 1, 0, 1] into corresponding chirp symbols.

Demodulation of chirp BOK signal can easily be implemented via matched filtering. One of the advantages of chirp signal is its auto correlation and cross correlation characteristics. Figure 2.15(a) and Figure 2.15(b) depicts the auto correlation and cross correlation plots, respectively, where T denotes symbol duration. Note that the auto correlation has a distinct peak at T, which can be differentiated from cross correlation at T. Therefore the demodulation of the received chirp signal is realized by (1) correlation with up chirp and down chirp, (2) sampling at mT time epochs where $m = 1, 2, \cdots$, and (3) comparing the sampled correlation at mT. The detailed receiver architecture will be described in Chapter 3.

Figure 2.16(b) and 2.16(c) shows an example of chirp BOK demodulation using correlation with up chirp and down chirp symbols. Comparing the correlation values of up chirp and down chirp at $T \sim 4T$ yields the transmuted bit sequence [1, 0, 1, 0]. Note that the demodulation does not need precise synchronization of transmitter and receiver; the receiver can demodulate the received signal once it finds the beginning of a frame and symbol period. Therefore the deceiver is non-coherent compared with PSK. Moreover, up and down chirp symbols are sweeping wide same frequency band (19.5∼22 kHz band in our implementation), and hence, the frequency selectivity does



(a) Auto correlation.

(b) Cross correlation.

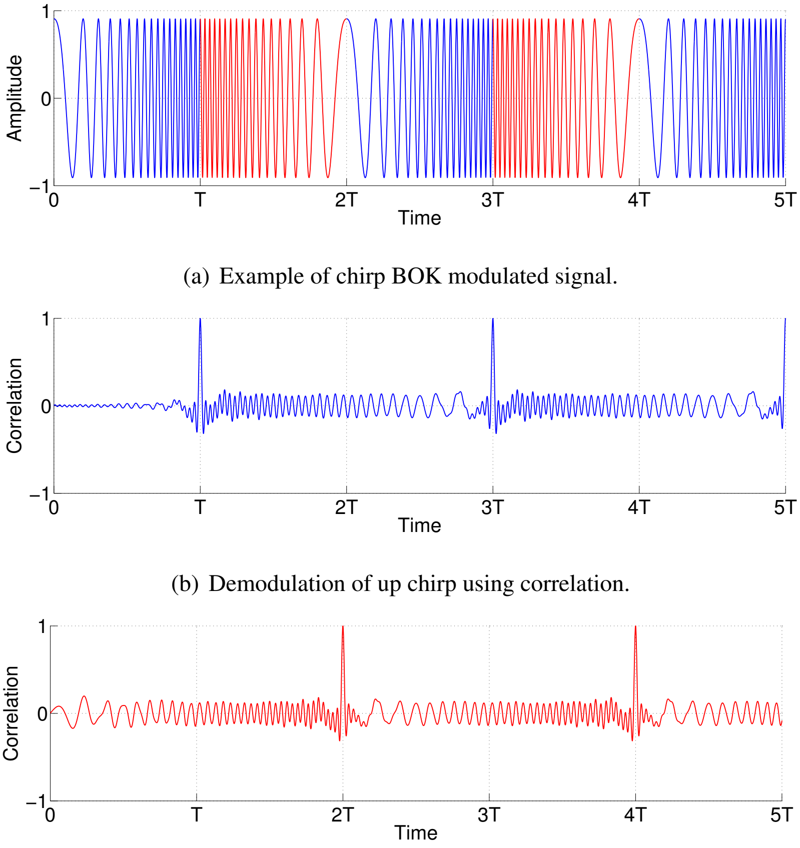
Figure 2.15: Auto correlation and cross correlation of chirp symbols.

not severely affects the distortion of symbols. If a device cannot play and/or record some part of the signal bandwidth, then up and down chirp symbols are partly lost no the entire symbols. Moreover this distortion applies to both up and down chirps in a symmetric manner, separate from the case in FSK.

As a chirp signal has good auto correlation characteristic, it has been widely used for radar applications as well as localizations [52, 53]. Based on the fact that up and down chirps are nearly orthogonal, the authors in [49, 54, 55] also propose chirp BOK for wireless communications. To our best knowledge, however, this is the first literature that applies chirp BOK to aerial acoustic communication in order to cope with the frequency selective and time-varying acoustic channel.

2.4 Performance Evaluation of PHY Modulation Schemes

In this section, we evaluate the bit error rate (BER) of PSK, FSK, and chirp BOK. Based on the experimental results in long-range indoor environment, we verify that chirp BOK modem outperforms PSK and FSK modulations.



(a) Example of chirp BOK modulated signal.

(b) Demodulation of up chirp using correlation.

(c) Demodulation of of down chirp using correlation.

Figure 2.16: An example of chirp BOK: modulation and demodulation.

2.4.1 Experimental Environment

We place Genelec 6010A reference speaker and smart devices at the end side of 25 m corridor with 2.5 m width and 3 m height. MackBook Air laptop plays PSK, FSK, and chirp BOK modulated acoustic signals through the reference speaker, where the volume of the laptop and speaker is set to 50%.

We use 12 different smart devices to record the acoustic signal in 5, 10, and 25 m distance. Basically the speaker and microphones of smart devices are facing with each other. In order to differentiate LOS and NLOS channel in this scenario, we conduct the experiment when the corridor is empty (for LOS) and people are moving around (for NLOS), so there are 6 different experimental environment. Each smart device record 1,600 symbols of PSK, FSK, and chirp BOK symbols in 16 bit PCM format with 44,100 Hz sampling rate in each experimental environment. The demodulators for PSK, FSK, and chirp BOK are implemented in MATLAB. We move the recorded wave audio files to MATLAB after the recording ends, and demodulate 9,600 symbols in total as follows.

2.4.2 PSK Demodulator

As discussed in Section 2.3.1, the receiver first tracks the phase drift over time from the training tones of preamble. For the phase estimation of each training tone and data symbols, we adopt exhaustive searching over $[0 \sim 2\pi]$ range. That is, the phase of k-th symbol, θ_k is estimated by

$$\hat{\theta}_k = \arg\max_{\theta} \int_T r_k(t) \cos(2\pi f_c t + \theta)\,\mathrm{d}t, \tag{2.1}$$

where f_c = 20 kHz and $r_k(t)$ denotes k-th received symbol.[4]

[4] $k = 1, 2, 3, 4$ corresponds to the training tones, and $k = 5, 6, \cdots$, 20 is for data symbols.

The average phase drift of PSK symbols, θ_d is calculated as the average phase difference of four training symbols, i.e.,

$$\theta_d = \frac{\hat{\theta}_4 - \hat{\theta}_1}{4}. \tag{2.2}$$

Then the phase of k-th data symbol is compensated considering the time difference between k-th and the first symbol for $k > 5$, i.e.,

$$\acute{\theta}_k = \left(\hat{\theta}_k - \hat{\theta}_1\right) - (k-1)\,\theta_d,\ k = 5, 6, \cdots, 20. \tag{2.3}$$

Finally the receiver decides whether i-th transmitted bit b_i was 1 or 0 by comparing $\acute{\theta}_k$ and 0.

$$b_i = \begin{cases} 0, & \text{if } \acute{\theta}_{i+4} \leq 0, \\ 1, & \text{if } \acute{\theta}_{i+4} > 0. \end{cases} \tag{2.4}$$

2.4.3 FSK Demodulator

We implement a non-coherent FSK receiver by using quadrature receiver [50]. Figure 2.17 represents the block diagram of non-coherent FSK receiver. As discussed in Secton 2.3.2, we assume that $f_1 = 19$ kHz and $f_2 = 21$ kHz in this experiment. For the k-th received symbol $r_k\,(t)$, the receiver first determines the decision bound ξ_d for the first four symbols as follows.

$$\xi_d = \frac{(\xi_{1,1} + \xi_{4,1}) - (\xi_{2,2} + \xi_{3,2})}{2}. \tag{2.5}$$

For the rest of symbols, $r_k\,(t)$ for $k = 5, 6, \cdots, 16$, the transmitted bit is estimated by comparing $\xi_{k,1} - \xi k, 2$ and ξ_d,

$$b_i = \begin{cases} 0, & \text{if } \xi_{i,1} - \xi i, 2 \geq \xi_d, \\ 1, & \text{if } \xi_{i,1} - \xi i, 2 < \xi_d, \end{cases} \tag{2.6}$$

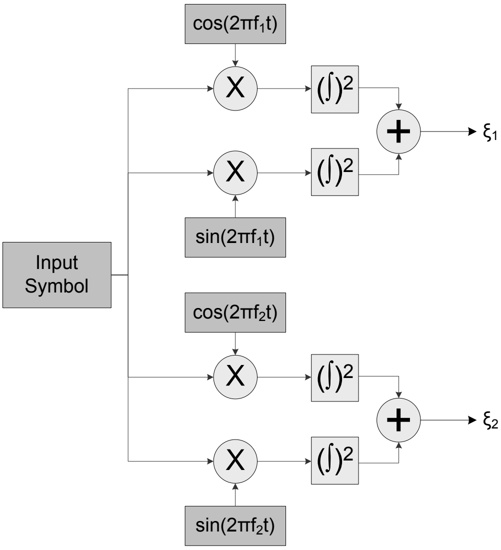


Figure 2.17: Block diagram of non-coherent FSK receiver.

2.4.4 BER of PHY Modulation Schemes

Figure 2.18 depicts the empirical BER of PSK, FSK, and chirp BOK modulations, obtained from the experiment of Section 2.4.1. The demodulation of PSK and FSK follows the processes in Sections 2.4.2 and 2.4.3, respectively. The demodulation of chirp BOK is based on the correlation, where the details will be discussed in Chapter 3. The number of total symbols processed in each modulation schemes is 9,600 symbols.

We first observe that PSK modulation has BER of 10^{-1} order in average, as the devices suffer from the frequency offset. Note that the BER of Vega Iron and Galaxy Tab is close to 0.5, which is equal to random guessing. Hence PSK cannot be used for these devices.

Non-coherent FSK performs better than PSK, and its BER of 10^{-2} order. Galaxy Note, Galaxy S3, and Optimus G Pro works better than other devices thanks to its flat fre-

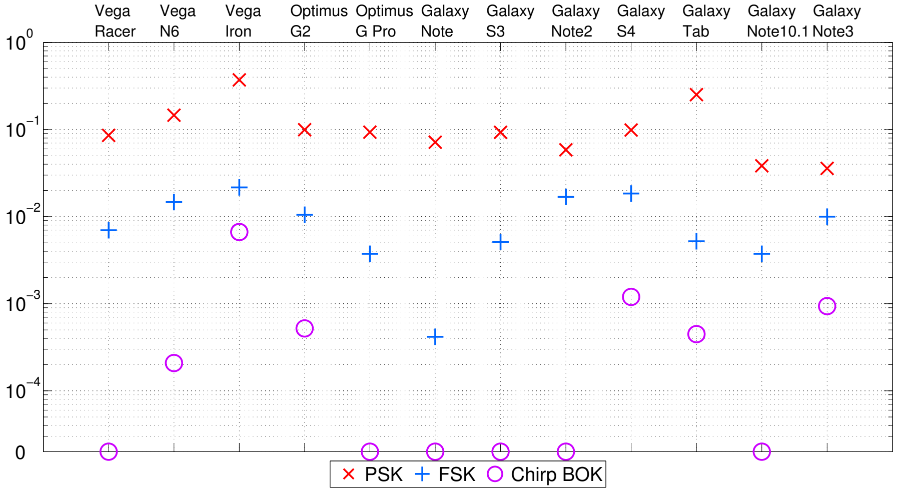


Figure 2.18: Empirical BER of PSK, FSK, and chirp BOK modulations in indoor acoustic channel.

quency response in high frequency band.

We can observe that chirp BOK outperforms PSK and FSK for all types of devices. The empirical BER of Vega Racer, Optimus G Pro, Galaxy Note, Galaxy S3, Galaxy Note2, and Galaxy Note10.1 is shown to be zero, i.e., none of 9,600 symbols were missed in this experiment. Hence we conclude that chirp BOK modulation can be an appropriate solution for indoor aerial acoustic communication.

2.5 Summary

In this chapter, we have studied the characteristics of indoor aerial acoustic channel. We define inaudible sound frequency band as $>$ 19 kHz through theoretic and experimental study. Measurement of the indoor acoustic impulse response reveals that the delay spread of NLOS indoor acoustic channel spans up to 40 msec for -6 dB threshold. As the aerial acoustic communication utilizes speakers of microphones as transmitter and receiver, we measure the frequency response of audio interfaces of various smart devices in an anechoic chamber. The measurement results show that the aerial acoustic channel is severely frequency selective due to the characteristics of the audio interfaces.

Based on the observations of these results, we empirically evaluated PSK and FSK modulation schemes that are widely adopted in vast of previous study. We present that the communication range of PSK and FSK in inaudible band is limited to few meters due to (1) instability of local oscillator and channel fluctuation, and (2) frequency selectivity of speakers and microphones.

We adopt chirp signal for the indoor aerial acoustic communication. Thanks to the auto correlation and cross correlation characteristics of chirp signals, we can design a simple non-coherent receiver. The chirp BOK modulation is proven to less suffers from the frequency offset as well as selectivity problem. We evaluate the BER of chirp BOK modulation in indoor environment, and show that chirp BOK outperforms PSK and FSK.

Note that chirp BOK is a type of spread spectrum modulation, as a symbol covers wide bandwidth. We plan to compare the performance of chirp BOK against other spread spectrum modulations, e.g., code devisor multiple access (CDMA) schemes. The experiments in this chapter is adopting 44.1 kHz sampling rate at the receiver side. Studying the effect of increased audio sampling rate (e.g., 48 kHz) on the performance of chirp BOK can also be a future work. Increasing the audio sampling rate can extend the available bandwidth over 22 kHz, and hence, the bit rate of chirp BOK is expected to be enhanced.

Chapter 3

Receiver Design for Acoustic Chirp BOK

3.1 Introduction

We have measured the indoor acoustic channel in the previous chapter. The indoor acoustic channel is shown to have large delay spread due to the multi-path propagation caused by walls, ceilings, and floors. The audio interfaces' frequency selectivity also affects the performance of aerial acoustic communication. Hence we adopt chirp BOK in inaudible band (19.5$\sim$22 kHz band) for PHY modulation scheme.

The communication scenario of this dissertation focuses on one-way transmission from loud speakers to smart devices. Therefore we aim to implement the receiver on the smart devices as a software modem that runs on their application layer. If the receiver architecture is too complex, then the smart devices consume lots of computing power for the signal reception. This is not appropriate for mobile devices, considering the limited computing power and battery lifetime. We hence need to keep the receiver architecture as simple as possible, while maintaining the signal reception capability.

Demodulation of the chirp signal can be realized via correlation and sampling (or equivalently, matched filter), as discussed in Section 2.3.3. This is based on the shape of auto correlation function of chirp signals — a distinct main lobe of auto correlation function can tell whether up chirp or down chirp is received. In practice, however, the auto correlation plot is not as clear as in Fig. 2.15(a) mainly for two reasons; (1) high frequency component of chirp signals, and (2) Doppler shift of channel environment. The receiver of chirp BOK modem has to effectively find the peak of auto correlation function in spite of the fluctuation and Doppler shifts.

In this chapter, we design a chirp acoustic signal in inaudible band based on the findings in Chapter 2. Then we describe corresponding receiver architecture for smart devices. Due to the limitation of the computing power of smart devices, we design the receiver as simple and efficient as possible. Mathematical study on the auto correlation of chirp signals reveals that the main lobe can better be found from its envelope, rather than the primitive auto correlation function. We also describe that Doppler shift causes time-shift of envelope peaks, and propose a method to adjust the sampling timing.

The main contribution of this chapter is two folds. First, we design the signal format and frame structure of inaudible chirp BOK modulation in indoor acoustic communication. Each data symbol is separated by 40 msec guard interval, which is the worst-case delay spread in indoor environment. The frame structure, composed of 16 data symbols, includes a preamble at the beginning, which is used as a delimiter at the receiver. Second, we propose a receiver architecture specifically designed for the reception of chirp BOK signal in smart devices. We reduce the complexity of matched filter by applying FFT instead of convolution. Furthermore, we combine the FFT with Hilbert transform so that the receiver derives envelope from the correlation function. By adjusting the symbol sampling timing of the envelope, the designed chirp BOK modem can easily find the envelope peaks.

The remainder of this chapter is organized as follows. Section 3.2 mathematically analyze the chirp signal and its auto correlation function. We propose an envelope detector for chirp BOK receiver that combines and Hilbert transform for efficient detection of correlate peaks. In Section 3.3, system parameters and frame structure of the proposed chirp BOK signal is described. We also design a receiver architecture for chirp signal reception. Section 3.4 presents performance evaluation results in indoor experiments. The proposed chirp signal can be detected in 25 m distance, which is far more extended compared to few meter range of the previous work. Finally Section 3.5 concludes the chapter with discussion on future work to further enhance the modem performance.

3.2 Chirp Signals and Matched Filter Receiver

3.2.1 Notation of Chirp Signals

A binary modulation scheme basically uses a pair of signal set. The pair of (linear) chirp signals, $s_1(t)$ and $s_2(t)$ for chirp BOK in frequency band from f_1 Hz to f_2 Hz ($f_2 > f_1$) is defined as

$$s_1(t) = \cos\left(2\pi f_1 t + \mu t^2/2 + \phi_1\right),\ 0 \leq t \leq T, \tag{3.1}$$

$$s_2(t) = \cos\left(2\pi f_2 t - \mu t^2/2 + \phi_2\right),\ 0 \leq t \leq T, \tag{3.2}$$

for Symbol 1 and Symbol 2, respectively. Here T is symbol duration, and ϕ_1 and ϕ_2 are arbitrary initial phase which is assumed to be zero without loss of generality. μ is referred to as *chirp sweep rate* in bandwidth $B = f_2 - f_1$, which is represented as

$$\mu = \frac{2\pi\left(f_2 - f_1\right)}{T} = \frac{2\pi B}{T}. \tag{3.3}$$

in radian. Note that the instantaneous frequency of $s_1(t)$ and $s_2(t)$ is $f_1 + (f_2 - f_1)\,t/T$ and $f_2 - (f_2 - f_1)\,t/T$, respectively. These are increasing and decreasing functions in time. We refer to $s_1(t)$ and $s_2(t)$ as *up chirp* and *down chirp*, respectively.

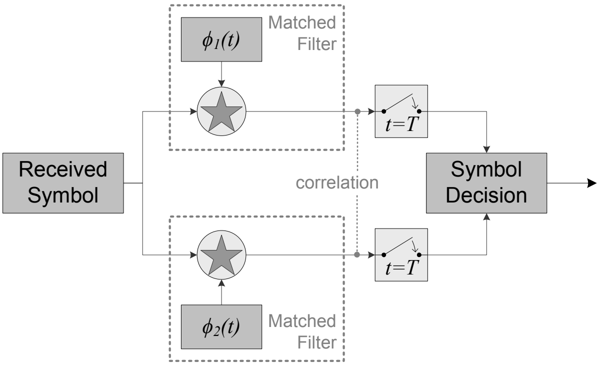


Figure 3.1: Binary receiver for signal set $\phi_1(t)$ and $\phi_2(t)$ using matched filter.

As discussed in Section 2.3.3, up chirp and down chirp are almost orthogonal. (Remind Fig. 2.15.) It is known that the cross correlation of up and down chirps is hard to be expressed in a close form [34]. Their auto correlation, however, is expressed as in Eq. (3.8). We discuss the auto correlation function in detail in Section 3.2.3.

A common method to implement the *matched filter* (or equivalently, *correlator*) receiver is convolution. Received signal $r(t)$ is convolved with time-reversed versions of $s_1(t)$ and $s_2(t)$ to generate estimator c_1 and c_2, respectively, that is,

$$c_1 = \int r(\tau)\, s_1(T-\tau)\, d\tau,$$

and

$$c_2 = \int r(\tau)\, s_2(T-\tau)\, d\tau. \tag{3.4}$$

If $c_1 > c_2$, the receiver estimates that Symbol 1 is transmitted, and Symbol 2 otherwise. Figure 3.1 depicts a typical receiver architecture for binary modulated signal set using matched filter. Operation marked by stars in two matched filters denotes convolution.

3.2.2 Matched Filter and FFT

The aerial acoustic communication is based on sound signals. The received signal $r\left(t\right)$ is obtained from recording process in OS kernel of smart devices. Note that the internal delay that takes to fetch recorded data from audio buffer cannot be tightly controlled. Therefore if a receiver tries symbol-by-symbol recording, then the acoustic signal might partly be lost in between recordings. Instead, we design the modem to process multiple symbols from a single recording with long duration. The receiver records acoustic signals long enough to cover a whole frame, and then demodulate the embedded information.

One problem of this bulk processing for the matched filter implemented by using convolution is its complexity. For instance, 1.1 second[1] recording with 44.1 kHz audio sampling rate returns approximately 4.8 MB data. It is well known that the complexity of convolution is $O\left(N^2\right)$. A rough benchmark using Galaxy S3 device shows that the convolution of 4.8 MB data takes approximately 5 sec. This is too long delay to run the receiver in real-time services.

We can simplify the matched filter using FFT[2] with complexity $O\left(N\log N\right)$. Note that the convolution in time domain is equivalent to the product in frequency domain. By substituting convolution to FFT, we can drastically reduce the complexity, especially for large N. More into detail, we can rewrite Eqs. (3.4) and (3.4) as follows.

$$c_1 = \mathcal{F}^{-1}\Big\{\mathcal{F}\{r\left(t\right)\}\mathcal{F}\{s_1\left(T-t\right)\}\Big\}, \tag{3.5}$$

and

$$c_2 = \mathcal{F}^{-1}\Big\{\mathcal{F}\{r\left(t\right)\}\mathcal{F}\{s_2\left(T-t\right)\}\Big\}, \tag{3.6}$$

[1] This corresponds to the frame duration of the proposed chirp BOK signal, as to be explained in Section 3.3.

[2] We use Kiss FFT open-source library [56] to implement the proposed receiver on Android devices.

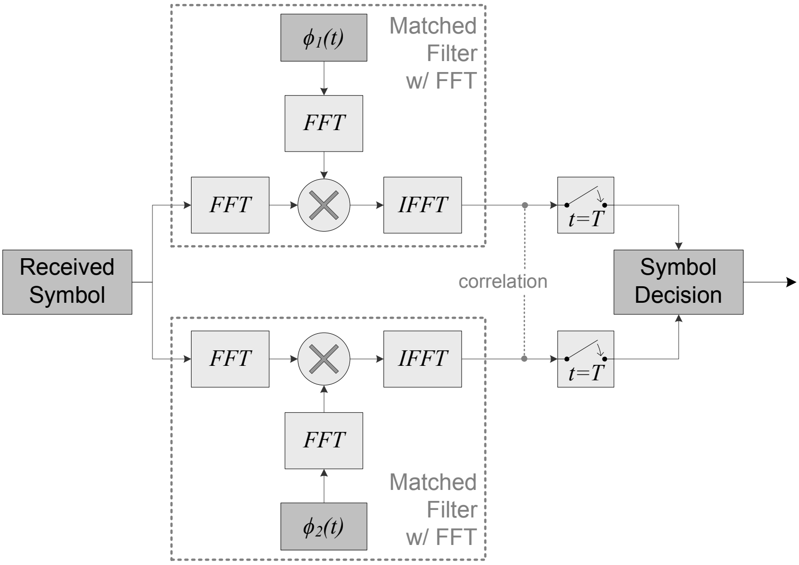


Figure 3.2: Binary receiver for signal set $\phi_1(t)$ and $\phi_2(t)$ using matched filter, implemented by FFT and IFFT.

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote FFT and inverse FFT (IFFT), respectively. Figure 3.2 presents the block diagram of binary receiver, where the matched filter is substituted by FFT instead of convolution.

One might argue that the matched filter with FFT (Fig. 3.2) has multiple FFT blocks, compared with two convolutions in Fig. 3.1. However, the FFT of signal set $\phi_1(t)$ and $\phi_2(t)$ does not need to be calculated every symbol reception. The FFT of those two can pre-calculate in off-line, e.g., just once at initialization. The input of two matched filters, FFT of the received symbol, is the same data so that the first two FFT blocks can be unified into a single block. Hence the actual FFT operations in real time is three blocks; FFT of received symbol, and two IFFTs for $\phi_1(t)$ and $\phi_2(t)$. Therefore the computing overhead is much more reduced compared with the matched filter with convolution.

3.2.3 Envelope Detection of Chirp Auto Correlation

Once the receiver calculates correlation using FFT and IFFT as in Fig. 3.2, the correlation output is sampled every symbol interval. If the symbol sampling timing is not exactly matched to the symbol interval, the sampling output is not an optimal value for the symbol decision. In this section, we investigate the correlation function of chirp signals and propose an efficient method to enhance the sampling operation.

Without loss of generality, we begin with up chirp in Eq. (3.1). The auto correlation function of an up chirp is calculated as

$$\psi\left(t\right)=\int_0^T \cos\left(2\pi f_1\tau+\frac{\pi B\tau^2}{T}\right)\cos\left(2\pi f_1\left(t-\tau\right)+\frac{\pi B\left(t-\tau\right)^2}{T}\right)d\tau. \quad (3.7)$$

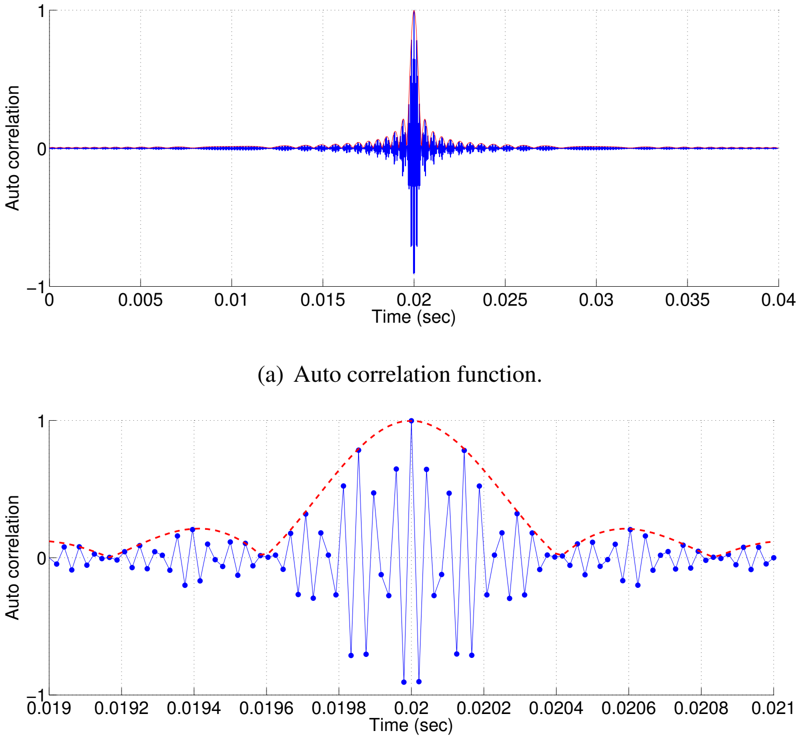
Trigonometric expansion of Eq. (3.7) yields two frequency components; original frequency term f_1 and high frequency term $2f_1$. We ignore the high frequency term $2f_1$ for most cases of practical interest. We also assume that $f_1 \gg B$.[3] Then we can write the close-form auto correlation as

$$\begin{aligned}\psi\left(t\right)&=T\Big(1-\frac{|t|}{T}\Big)\mathrm{sinc}\left[\pi B\Big(1-\frac{|t|}{T}\Big)\right]\cos\left(2\pi f_1 t\right),\\ &\triangleq e\left(t\right)\cos\left(2\pi f_1 t\right), \qquad (3.8)\end{aligned}$$

which is a form of cosine-modulated time-decaying sinc function [34].

Figure 3.3 presents an example of auto correlation of chirp signal. The sweeping frequency and symbol duration is set to 19.5 $\sim$ 20 kHz and 20 msec, respectively, with 44.1 kHz audio sampling rate in this example. Separate from the example in Section 2.3.3, we observe the fluctuation of auto correlation function.Therefore the receiver has to pick the sample at exact symbol duration to maximize the receiver performance. For example, in Fig. 3.3(b), suppose that a receiver misses the symbol

[3] In our chirp BOK signal design, $f_1 = 19.5$ kHz and $B = 2.5$ kHz. This satisfies the assumption.



(a) Auto correlation function.

(b) Auto correlation function magnified around the symbol time (20 msec).

Figure 3.3: An example of auto correlation for chirp signals sweeping from 19.5 to 22 kHz with 20 msec symbol duration.

sampling timing by 1 audio sample difference ($1/44,100 = 22.7\ \mu$sec). This can frequently happen in smart devices with unstable local oscillators as we discussed in Chapter 2. Then the sampled value of auto correlation function severely decrease.

We can resolve this problem by finding the envelope of auto correlation function, depicted in red dashed line in Fig. 3.3(b). In order to model the envelope, we here investigate the auto correlation function in detail. Equation (3.8) can be divided into two terms as follows.

Envelope term $T\Big(1-\frac{|t|}{T}\Big)\mathrm{sinc}\Big[\pi B\Big(1-\frac{|t|}{T}\Big)\Big]$ is a form of sinc function where the decaying rate is scaled by a triangular function. We denote the envelope function as $e\left(t\right)$.

Oscillating term $\cos\left(2\pi f_1 t\right)$ causes the oscillation of auto correlation function shown in Fig. 3.3(b).

We aim to remove oscillating term and find the envelope in order to easily read clear peak values. We notice that Eq. (3.8) is very similar to an amplitude modulated (AM) signal with carrier frequency of f_1. Therefore, the envelope $e\left(t\right)$ can be obtained in a similar method that is used to demodulate AM signals. Here we begin from obtaining the absolute value of the envelope. Note that the main lobe of $e\left(t\right)$ has positive values, so that the absolute operation does not make any difference. We can represent $\left|e\left(t\right)\right|$ using Euler's formulae as

$$\begin{aligned}\left|e\left(t\right)\right| &= \Big|e\left(t\right)\Big(\cos\left(2\pi f_1 t\right)+\jmath\sin\left(2\pi f_1 t\right)\Big)\Big|,\\ &= \left|e\left(t\right)\cos\left(2\pi f_1 t\right)+\jmath e\left(t\right)\sin\left(2\pi f_1 t\right)\right|. \end{aligned} \tag{3.9}$$

Note that $e\left(t\right)\cos\left(2\pi f_1 t\right)$ is equal to $\psi\left(t\right)$, and $e\left(t\right)\sin\left(2\pi f_1 t\right)$ is $\pi/2$ phase-lagging formula of $\psi\left(t\right)$. This can be represented using *Hilbert transform* [57]. That is,

$$\begin{aligned}\left|e\left(t\right)\right| &= \left|e\left(t\right)\cos\left(2\pi f_1 t\right)+\jmath e\left(t\right)\sin\left(2\pi f_1 t\right)\right|,\\ &= \left|\psi\left(t\right)+\jmath\hat{\psi}\left(t\right)\right|, \end{aligned} \tag{3.10}$$

where $\hat{\psi}\left(t\right)$ denotes *Hilbert transform* of $\psi\left(t\right)$.

As mentioned in Section 3.2.2, we design the matched filter using FFT. Therefore we consider frequency domain representation of $e\left(t\right)$. More into detail, let $\Psi\left[w\right]$ and $\hat{\Psi}\left[w\right]$ be FFT representations of $\psi\left(t\right)$ and $\hat{\psi}\left(t\right)$, respectively. Note that $\psi\left(t\right)$ and $\hat{\psi}\left(t\right)$ are related by

$$\hat{\Psi}\left[w\right] = -\jmath\,\mathrm{sgn}\left[w\right]\Psi\left[w\right], \tag{3.11}$$

where $\mathrm{sgn}\left[\omega\right]$ is a *signum function*, i.e.,

$$\mathrm{sgn}\left[w\right] = \begin{cases} 1, & w > 0, \\ 0, & w = 0, \\ -1, & w < 0. \end{cases} \tag{3.12}$$

This is due to the fact that Hilbert transform has $\pi/2$ phase lagging compared with the original function.

Then the frequency domain representation of the right-hand-side of Eq. (3.10) is

$$\begin{aligned} \mathcal{F}\{\psi\left(t\right) + \jmath\hat{\psi}\left(t\right)\} &= \mathcal{F}\{\psi\left(t\right)\} + \mathcal{F}\{\jmath\hat{\psi}\left(t\right)\}, \\ &= \hat{\Psi}\left[w\right] - \jmath\,\mathrm{sgn}\left[w\right]\Psi\left[w\right], \qquad (3.13) \\ &= \begin{cases} 2\Psi\left[w\right], & w > 0, \\ \Psi\left[w\right], & w = 0, \\ 0, & w < 0. \end{cases} \qquad (3.14) \end{aligned}$$

IFFT and absolute of Eq. (3.13) yields

$$\begin{aligned} \left|e\left(t\right)\right| &= \left|\psi\left(t\right) + \jmath\hat{\psi}\left(t\right)\right|, \\ &= \left|\mathcal{F}^{-1}\{\Psi\left[w\right] + \jmath\hat{\Psi}\left[w\right]\}\right|, \\ &= \left|\mathcal{F}^{-1}\{2\Psi\left[w\right]u\left[w\right] - \Psi[0]\}\right|, \qquad (3.15) \end{aligned}$$

where $u\left[w\right]$ is unit step function such that

$$u\left[w\right] = \begin{cases} 1, & w > 0, \\ 0, & w = 0. \end{cases} \tag{3.16}$$

Therefore, finding envelopes in Eq. (3.15) can be easily processed after the convolution. Equation 3.15 is referred to as *envelope detector*.

Let us assume that a receiver records $r\left(t\right)$, and calculates the envelope of correlation with an up chirp signal. The computing process of the envelope detector works as follows.

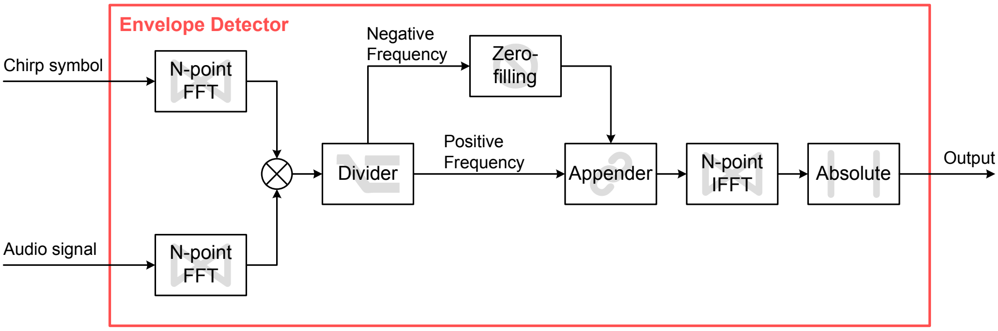


Figure 3.4: A block diagram of envelope detector.

(i) Calculate FFT of received audio ($\mathcal{R}$) and up chirp ($\mathcal{U}$).

(ii) Multiply $\mathcal{R}$ and $\mathcal{U}$, and substitute negative frequency terms to zero. Here the scalar factor 2 and $\Psi[0]$ in Eq. (3.15) are omitted here for simplicity.

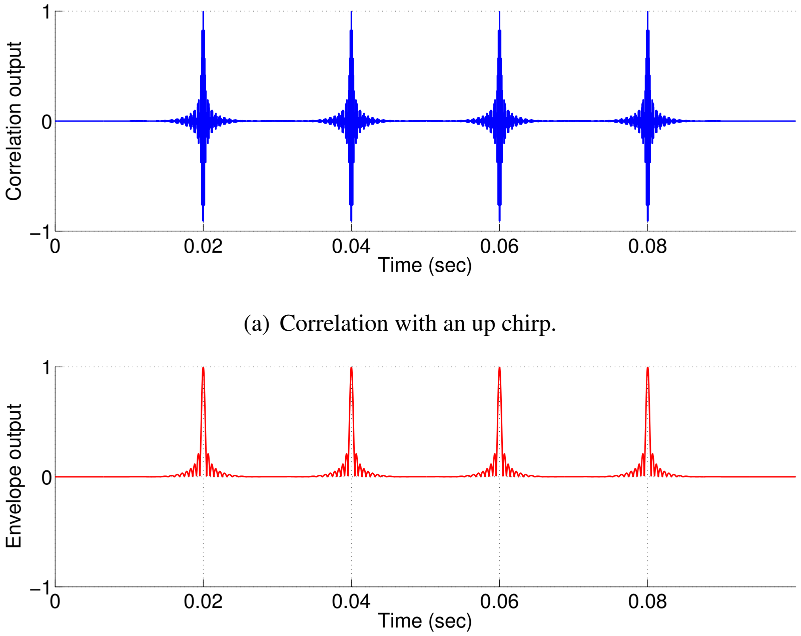
(iii) IFFT and take absolute values, and this yields envelope.

Figure 3.4 graphically presents the processing flow of envelope detector.

Figures 3.5(a) and 3.5(b) plots the correlation and envelope output obtained by convolution and envelope detector, respectively, assuming that four up chirps are serially received. For the rest of the dissertation, we denote *envelope* of a chirp signal as the envelope of the correlation with the chirp signal for the ease of notation.

3.3 System Design and Receiver Architecture

Now we design the frame structure and receiver architecture. The system parameters are chosen based on the measurement results as well as the receiver design concept of Section 3.2, and they are summarized in Table 3.1.



(a) Correlation with an up chirp.

(b) Envelope of the correlation with an up chirp.

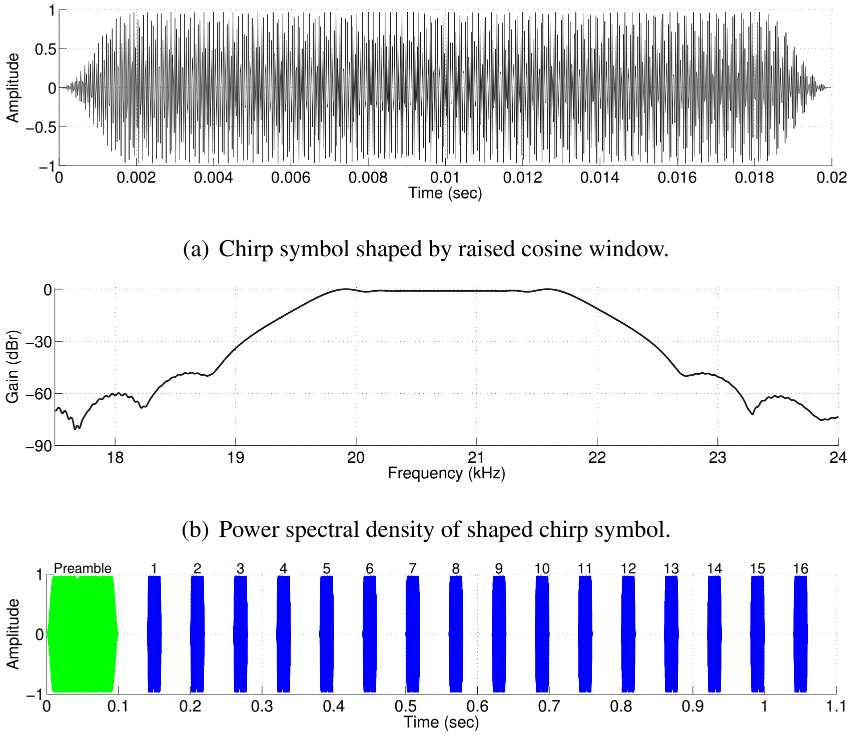
Figure 3.5: Comparison of correlation output and its envelope.

3.3.1 Frame and Symbol Design

In this dissertation, we consider the communication scenario that chirp signals are used to broadcast a short ID broadcast. We assume that 16 bit ID is repeatedly played by loud speakers.[4] Smart devices then records the chirp BOK signal, and demodulate the embedded 16 bit ID by using the proposed envelope detector.

Note that the speaker and the smart device (and its microphone, eventually) are not synchronized with each other. Therefore the begging of the recorded audio is very likely to have an offset from the actual beginning of the 16 bit ID. In order for the

[4] One might argue that this 16 bit ID is too short. In Chapter 4, we propose backend server architecture in order to compensate for this low bit rate.



(a) Chirp symbol shaped by raised cosine window.

(b) Power spectral density of shaped chirp symbol.

(c) Frame structure including preamble and guard intervals.

Figure 3.6: Design of chirp signals and corresponding frame structure with preamble and guard interval.

receiver to find the beginning of a 16 bit ID, we deploy 0.1 sec up chirp signal as a *preamble* that can be used as a frame delimiter. Then the first thing that the receiver has to do is finding this preamble.

In addition, indoor acoustic channel typically suffers from large delay spread caused by multi-paths. As discussed in Section 2.2, we found that the typical delay spread in indoor aerial communication is 40 msec. Implementing an equalizer that removes ISI from the multi-path propagation can be a solution for this delay spread problem. However, remind that our goal of the receiver design is its simplicity. The equalizers such as zero forcing (ZF) or minimum mean square error (MMSE) are generally implemented via digital filters, which is realized by convolution. In this dissertation, we do not implement the channel equalizer to keep the receiver simple. Instead we adopt a simple solution to avoid the delay spread; we insert 40 msec guard interval after preamble as well as in between symbols, as shown in Fig. 3.6(c).

Next we design the data symbols that follow the preamble. We set the duration of data symbol to 20 msec. Note that we implement the chirp BOK receiver using FFT. Considering the computing power of general smart devices, we choose 65,536 point FFT size. As the most common sampling rates in audio interfaces are 44.1 kHz or 48 kHz, 20 msec symbol duration with 16 data symbols and corresponding guard intervals builds 48,510 samples and 52,800 samples frame size for 44.1 kHz and 48 kHz sampling rates, respectively. In order for FFT to be equivalent to convolution, the FFT size should be grater than or equal to the sum of received audio buffer size and chirp signal size [58]. 20 msec is the longest symbol duration satisfies this condition for both 44.1 kHz and 48 kHz sampling rates, and results in continuous-phase chirp signals. Note that this phase continuity is important to reduce the frequency leakage out of the signal bandwidth [59].

Each data symbol is shaped using raised cosine window with roll-off factor 0.1 as shown in Fig. 3.6(a). Note that the receiver convolves the received audio with up and down chirp symbols. Therefore the raised cosine window also works as an ingress bandpass filter at the receiver. We can regard the power spectral density (PSD) of the shaped chirp signal in Fig. 3.6(b) as the spectral mask at both the transmitter (speaker) and receiver (microphone) accordingly. The roll-off factor of 0.1 is designed to reduce (1) out-of-band leakage to -10 dB at frequency boundaries of 19.5 kHz and 22 kHz, and (2) ringing and rising effects at speakers and microphones [26]. Table 3.1 summarizes system parameters used in our design.

Table 3.1: System Parameters of Chirp BOK Receiver

Parameter	Value	Number of samples	
		@44.1 kHz	@48 kHz
Frequency band	19.5$\sim$22 kHz	-	-
Symbol duration	20 msec	882	960
Guard interval	40 msec	1764	1,920
Preamble	100 msec	4,410	4,800
Number of bits per frame	16 bits	-	-
Frame size	1.1 sec	48,510	52,800
FFT size	65,536 point	-	-
Raised cosine filter	roll-off 0.1	-	-

3.3.2 Signal Reception Process

Now we explain the signal reception process specifically designed for the reception of chirp BOK signal. The brief reception process is as follows. Basically a smart device begins the signal reception from recording, which can be triggered on-demand and/or periodic scheduling. This is based on the assumption that a loud speaker repeatedly plays chirp BOK signal — the smart device asynchronously tries to received the acoustic signal in an opportunistic manner. The received audio data is then delivered to receiver module at the application layer. The receiver first verifies whether there is chirp BOK signs or not in the recorded audio.[5] If there exists a chirp BOK signal, then the receiver try to decode the embedded information. If there is no signal, then the receiver flushes the received audio data and waits until the next recording.

[5] The triggering of recording process and signal detection algorithm will be discussed in Chapter 4.

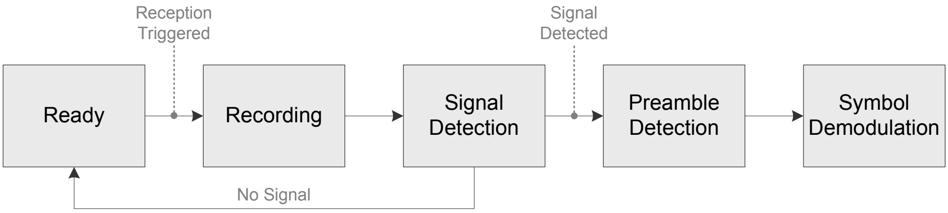


Figure 3.7: Reception process of the chirp BOK receiver.

Figure 3.7 depicts the reception process. More into detail, each reception process works in the following order.

Recording

A receiver begins signal reception by recording. The receiver at the application layer first initializes the microphone of the smart device, and calls recording function. In case of Android OS, the OS calls back the receiver processor when the microphone's buffer is ready to be fetched. We set the recording size as 1.2 sec, considering 1.1 sec frame duration and 0.1 sec preamble duration. This recording size guarantees that the audio data always includes at least one fully-recorded preamble in it.

Signal Detection

Once the receiver fetches 1.2 sec audio data from microphone buffer, it first tries to check the existence of high frequency signal. We calculate power spectral density (PSD) from FFT of the recorded audio. If the *peak PSD ratio* of in-band (19.5$\sim$22 kHz) to out-of-band (16$\sim$18 kHz) is grater than or equal to 4.5 dB,[6] then the receiver judges that there exists acoustic signal and proceeds to the next step; otherwise prepares the next recording in stand-by mode.

[6] We discuss the hard numbers and details in Chapter 4

Preamble Detection

The receiver then searches the preamble in the recorded audio via envelope detection based on Eq. (3.15). We use 65,536 point FFT in the envelope detector of Fig. 3.4, so that the FFT-IFFT operation exactly works the same as convolution. By finding the location of the maximum value of preamble envelope, the receiver can find the beginning of the frame, referred to as *preamble offset*.

Symbol Demodulation

Now the receiver calculates up chirp envelope and down chirp envelope from the recorded audio. The envelope detection in preamble detection is repeated, where preamble is substituted by up and down chirps. Beginning from the *preamble offset*, the receiver compares envelopes of up and down chirps every symbol interval, and recovers the received bit by comparing them. If the receiver encounters end-of-buffer, then it traces back for one frame duration and continues until all16 bits are decoded. This wrap-around peak sampling is feasible since the transmitter repeatedly broadcasts the same ID over time.

3.3.3 Receiver Architecture

Figure 3.8 presents the block diagram of the receiver. The demodulation process works in the order of (i) recording, (ii) signal detection, (iii) preamble detection, and (iv) symbol demodulation. In Fig. 3.8, preamble detection and symbol demodulation procedures are illustrated in detail, where the envelope detector is described in Fig. 3.4. Here we discuss the technical issues on the implementation of receiver, and describe an example of reception process.

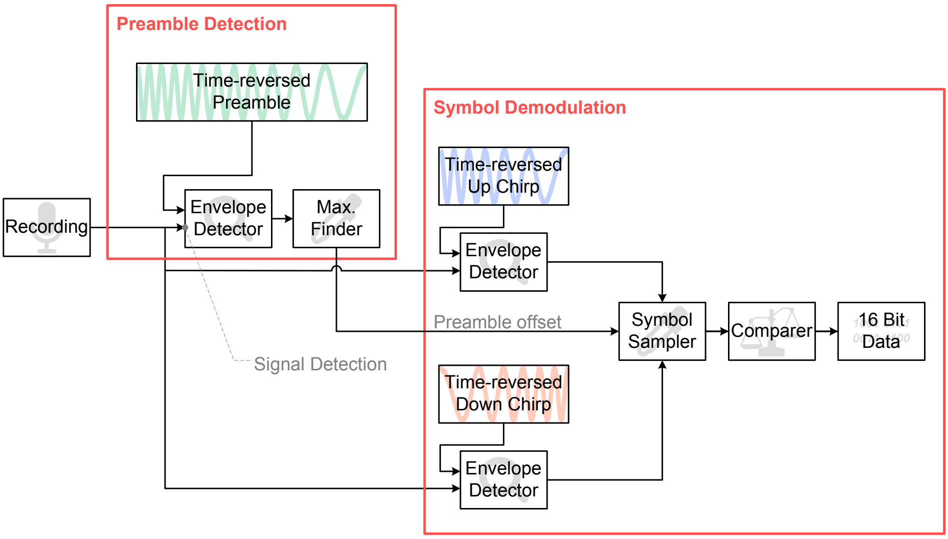
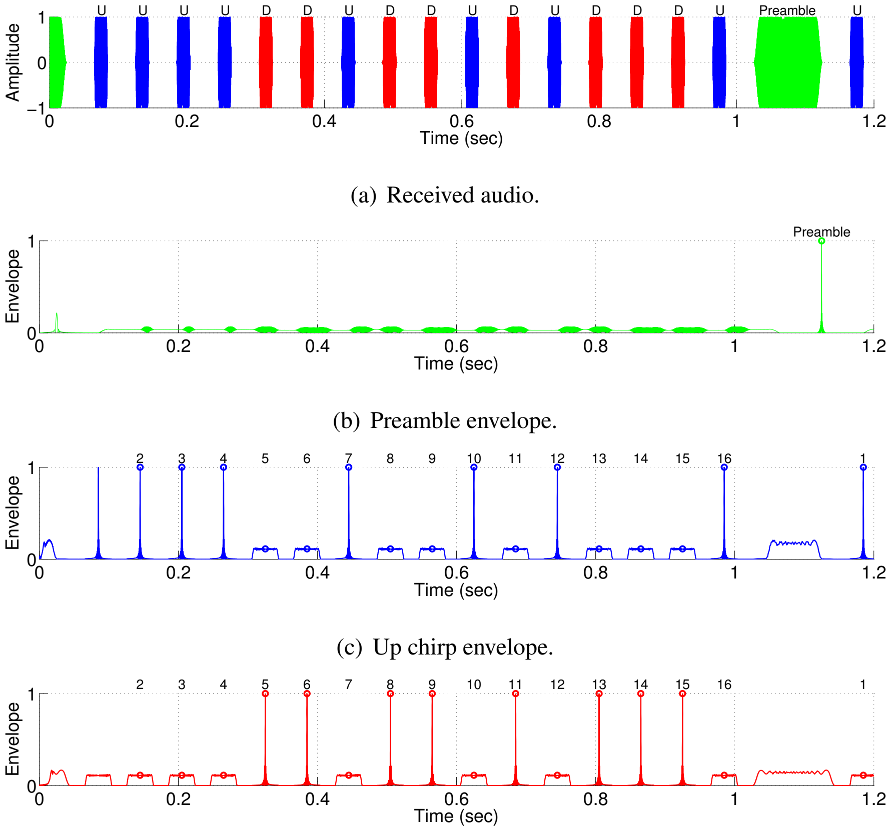


Figure 3.8: Block diagram of the chirp BOK receiver.

Discussions on the Receiver Implementation

Three envelope detectors in Fig. 3.8 takes one of their input from the recorded audio data. Remind 3.4 that the input of the envelope detector first goes through FFT module inside the envelope detector — as the inputs of three envelope detectors are the same, they can share a common memory space in the reception processor in order to save the computing memory.

The other inputs of three envelope detectors are preamble, up chirp, and down chirp signals. These are fed in a time-reversed order so that the envelope detection via FFT works exactly the same as matched filter with convolution. The FFT of these three chirp signals are pre-calculated in off-line, e.g., at the initialization of the receiver processor. It should be noted that up chirp and down chirp signals are symmetric in time domain. Hence their FFTs are complex conjugate with each other; this can further reduce the overhead of FFT calculation as well as the memory space.



(a) Received audio.

(b) Preamble envelope.

(c) Up chirp envelope.

(d) Down chirp envelope.

Figure 3.9: Decoding example of the proposed modem.

Reception Example

Figure 3.9 depicts an example of signal demodulation. Suppose that the receiver records the audio in Fig. 3.9(a). It is composed of a preamble (denoted as *Preamble* above the long green symbol) and 16 chirp symbols, where bit 1 is modulated as an up chirp (denoted as *U* above the blue symbols) and bit 0 is modulated as a down chirp (denoted as *D* above the red symbols). As the microphone is not synchronized with the speaker, the beginning of the recorded audio is not the beginning of the preamble; only some part of the preamble is recorded in this example. As we set the recording duration to 1.2 sec, however, the receiver can find the whole preamble that is recorded from 1.025 sec to 1.125 sec. This can be found from the preamble envelope of Fig. 3.9(b). The small peak at the beginning of preamble envelope comes from the partly recorded preamble. The full preamble at 1.025 sec produces a high peak at 1.125 sec when the preamble ends. This is detected by searching for the maximum value of the preamble envelope, and its location 1.125 sec is saved as the preamble offset.

Now the receiver calculates up chirp envelope and down chirp envelope in Figs. 3.9(c) and 3.9(d), respectively. After finding the end of the preamble at 1.125 sec, receiver samples up chirp envelope and down chirp envelope every 60 msec, which is equivalent to the sum of symbol duration 20 msec and guard interval 40 msec. The first symbol, which is an up chirp, is obtained by comparing the up chirp envelope and down chirp envelope at 1.185 sec. The next symbol is supposed to appear after 60 msec, i.e., at 1.245 sec. However, this is beyond the recorded audio duration 1.2 sec. As the receiver encounters end-of-buffer, it traces one frame duration (1.1 sec) back to sample the second symbol. The receiver then demodulates the second symbol at 0.145 sec, and proceeds the demodulation by comparing up chirp envelope and down chirp envelope every 60 msec, until the receiver decodes 16-th symbol at 0.985 sec. The demodulation order of 16 data symbols are denoted above the up chirp envelope and down chirp envelope.

3.3.4 Symbol combining for BER enhancement

In the communication scenario of the dissertation, we assume that a transmitter (i.e., a speaker) repeatedly plays the same chirp signal over time. Therefore a receiver (i.e., a microphone) can receive multiple copies of the transmitted chirp signal by prolonging the recording duration. Combining the multiple copies of chirp frames (and eventually symbols) improves the BER thanks to the *time diversity*. We present our receiver design with *selective combining* for the decoding of chirp symbols.

Revisit of maximum likelihood symbol combining

The design of the combined symbol decoder is based on *maximum likelihood* (ML) decoder. Without loss of generality, we focus on a specific symbol in a 16 bit frame. Let $\mathbf{R}$ be the vector of N copies of the symbol recorded over time, i.e.,

$$\mathbf{R} = \left[r_1, \; r_2, \; \cdots, \; r_N \right]. \tag{3.17}$$

As we demodulate a recorded chirp BOK symbol by using up and down chirps, the i-th symbol copy r_i generates two matched filter output (and equivalently *envelope*) values.

$$r_i = \left(\bar{u}_i, \; \bar{d}_i \right), \tag{3.18}$$

where $\bar{u}_i$ and $\bar{d}_i$ denotes the up chirp envelope and down chirp envelope, respectively, normalized to the preamble envelope. According to Eq. (3.7), the height of envelope peak is proportional to the duration of a chirp signal. Preamble symbol in our signal design is five times longer than data symbols, and hence,

$$\bar{u}_i = \frac{5u_i}{p_i}, \tag{3.19}$$

$$\bar{d}_i = \frac{5d_i}{p_i}, \tag{3.20}$$

where p_i, u_i, and d_i denotes the envelope outputs of preamble, up chirp, and down chirp, respectively.

Note that r_i can be depicted on a two-dimensional constellation plot. Assuming additive white Gaussian noise (AWGN) on the constellation points, *post priori* probabilities that the received chirp symbol was an up chirp (H_0) and down chirp (H_1) are given by

$$p\left(\mathbf{R}|H_0\right)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(r_i-\mathrm{u})^2}{2\sigma_i^2}}, \tag{3.21}$$

and

$$p\left(\mathbf{R}|H_1\right)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(r_i-\mathrm{d})^2}{2\sigma_i^2}}, \tag{3.22}$$

respectively, where σ_i is the noise power of the i-th symbol copy, which is assumed to be proportional to $5/p_i$. $\mathrm{u}=(1,\epsilon)$ and $\mathrm{d}=(\epsilon,1)$ are the signal constellation points of up and down chirps in a cartesian coordinate, assuming the cross correlation of up and down chirps is ϵ. Now the ML decision is based on Eqs. (3.21) and (3.22), i.e.,

$$\underset{H\in\{H_0,\,H_1\}}{\arg\max}\;p\left(\mathbf{R}|H\right). \tag{3.23}$$

Taking natural logs on Eqs. (3.21) and (3.22) and omitting negative sign by taking minimums, Eq. (3.23) deduces to

$$\min\left\{\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\left\|r_i-\mathrm{u}\right\|^2,\;\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\left\|r_i-\mathrm{d}\right\|^2\right\}, \tag{3.24}$$

where $\|\cdot\|^2$ denotes Euclidean distance between two constellation points.

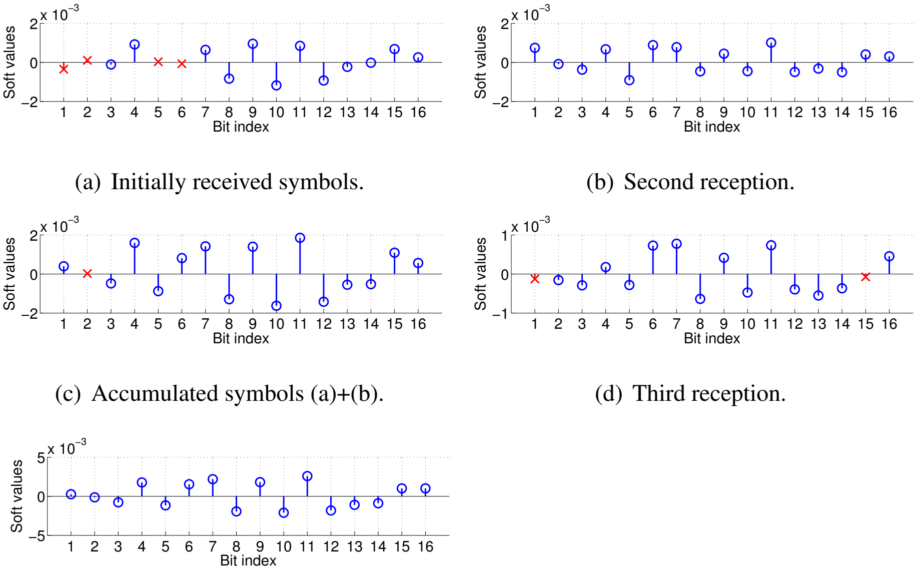
Soft symbol metric

It should be noted that the binary symbol decision is made by comparing the difference of the arguments in Eq. (3.24) and zero, i.e.,

$$\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\Big(\left\|r_i-\mathrm{u}\right\|^2-\left\|r_i-\mathrm{d}\right\|^2\Big)\overset{?}{\gtrless}0. \tag{3.25}$$

From Eq. (3.18) and substituting $(1,\epsilon)$ and $(\epsilon,1)$ to u and d, respectively, the left hand side of Eq. (3.25) deduces to

$$\sum_{i=1}^{N}\left(u_i-d_i\right)\triangleq\sum_{i=1}^{N}\eta_i. \tag{3.26}$$



(a) Initially received symbols.

(b) Second reception.

(c) Accumulated symbols (a)+(b).

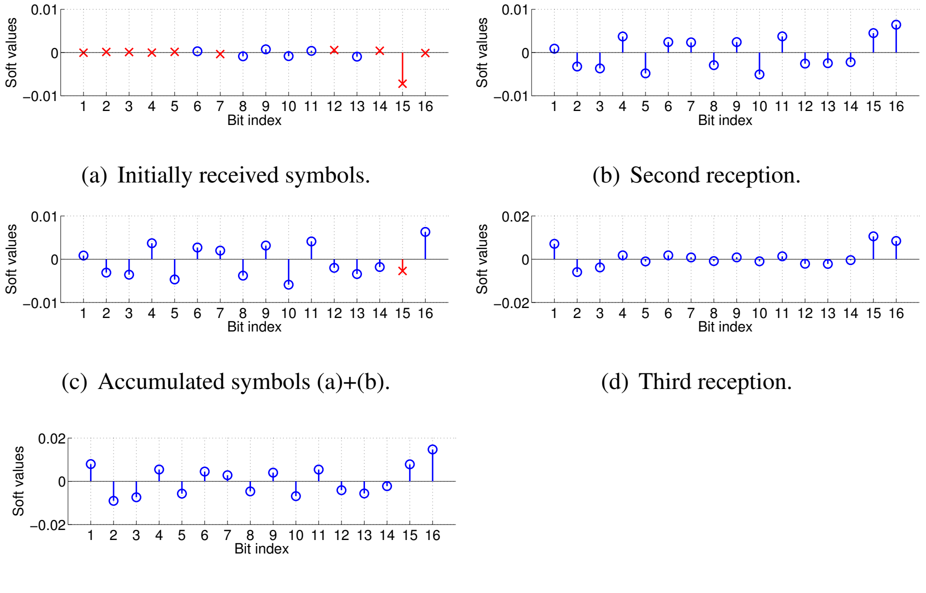
(d) Third reception.

(e) Accumulated symbols (a)+(b)+(d).

Figure 3.10: An snapshot of symbol combining.

Here we define soft symbol $\eta_i \triangleq (u_i - d_i)$ for i-th symbol copy. Note that Eq. (3.26) for $N = 1$ is the case of single reception case, i.e., without combining, and this is equivalent to the receiver design in Section 3.3.2. Hence the implementation of combined receiver does not require additional signal processing blocks, but accumulating the soft symbols up to the combining limit is enough.

Figure 3.10 depicts an example of combined decoding of Vega Iron device when the combining limit is 3 copies. Correctly and erroneously received symbols are denoted as o and x marks, respectively, out of 16 symbols. Initially received symbols (Fig. 3.10(a)) have bit error on 1, 2, 5, and 6-th bits. By combining the second symbol copies in Fig. 3.10(b), the receiver can recover the bit errors (Fig. 3.10(c)) bit still has an error on the second bit. Combining the third copy in Fig. 3.10(d), the receiver finally can recover all bits as shown in Fig. 3.10(e).



(a) Initially received symbols.

(b) Second reception.

(c) Accumulated symbols (a)+(b).

(d) Third reception.

(e) Accumulated symbols (a)+(b)+(d).

Figure 3.11: An snapshot of symbol combining with a severely corrupted symbol.

Selective combining

The limitation of the repeated combining arises when a symbol is severely corrupted so that the combining is not helpful. Such an example is shown in Fig. 3.11. Bit 15 in Fig. 3.11(a) is erroneously received, where its soft value is far away from a correct decision. Therefore the symbol recovery of Bit 15 requires two additional copies in Figs. 3.11(b) and 3.11(d). Note that, however, the second reception (Fig. 3.11(b) already includes perfectly recovered bits, the receiver would better chose the second copy rather than combining it with the erroneous copy.

Hence we design a *selective combining* of received copies. Specifically, when the receiver triggers combined decoding due to the bit error of the first copy, the receiver checks the bit error of the second copy. If the second copy does not have a bit error,

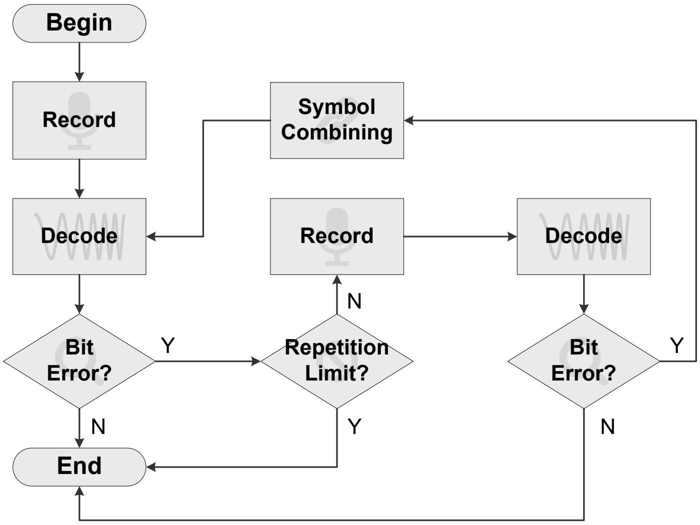


Figure 3.12: Algorithm flow chart of selective combining.

then the receiver adopts the second copy without combining. The receiver combines the soft values only when none of the copies can solely recover the received bits. The algorithm flow chart of the selective combining is presented in Fig. 3.12.

3.4 Performance Evaluation of Chirp BOK Receiver

In this section, we evaluate the performance of the proposed chirp BOK receiver through extensive experimental results. The proposed modem is proven to work in long distance, up to 25 m distance, in indoor environment. We also conduct an experiment using a set of stereo speaker to validate the multi-path resolution of the designed chirp signals. Experiment results show that the proposed chirp signals can differentiate the path difference greater than or equal to 0.4 m. Finally we design an algorithm for adjusting the symbol sampling timing of up and down chirp envelopes. This adjustment algorithm can cope with the envelope peak shifts caused by Doppler shift.

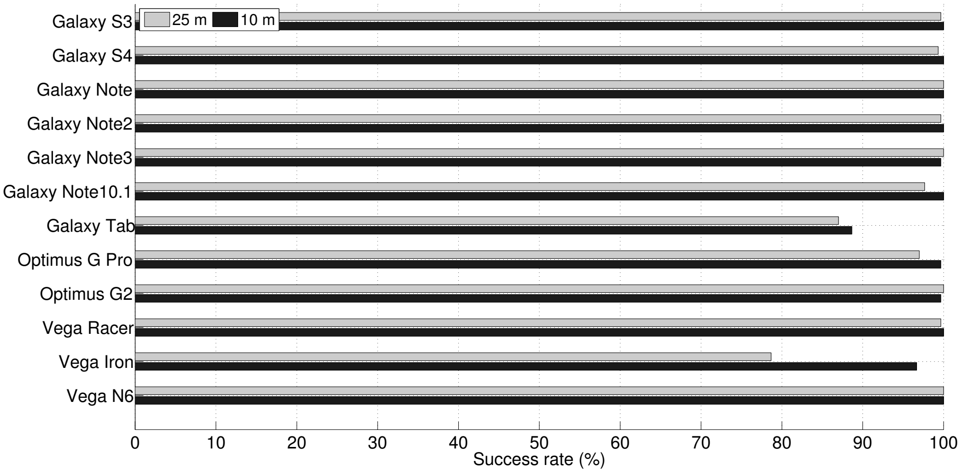


Figure 3.13: Success probability of signal reception at 10 m and 25 m distances for various smart devices.

3.4.1 Experimental Environment

We implement our chirp BOK modem on 12 Android devices with OS versions from Froyo (Android 2.2) to Kitkat (Android 4.4). The modem core is implemented in C language, and it is wrapped by JAVA NDK wrapper. Then an test application that is implemented in JAVA language loads the modem core as a library. Most experiments are conducted in a static indoor office environment unless mentioned otherwise.

3.4.2 Transmission Range in Indoor Environment

We first verify the transmission range of the proposed modem. We deploy Genelec 6010A speaker and 12 different smartphones at the end side of 25 m corridor with 2.5 m width and 3 m height. MacBook Air laptop plays chirp BOK modulated signal, encoded by 16 bit PCM wave file format with 44.1 kHz sampling rate. The volumes of laptop and speaker are set to 50%.

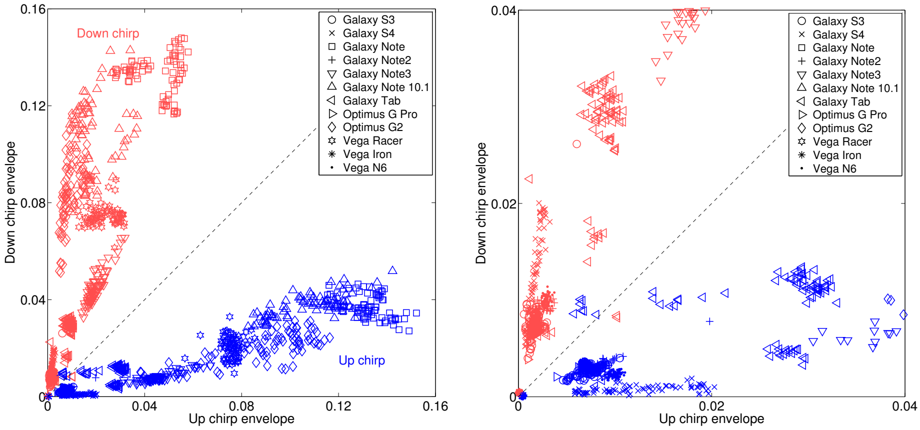
Figure 3.13 presents the success probability of signal reception, averaged over 300 repetitions of 16 bit ID reception. It can be observed that most devices can successfully receive the chirp signal with 97% probability, which verifies that the proposed modem works in long transmission range. The low success rates of Galaxy Tab and Vega Iron at 25 m distance are due to their low volume of recording, as shown in Fig. 2.4. These devices rarely cannot the recorded audio, so their decoding rate degrades.

It should be noted that the sound pressure level of the received audio at 25 m distance was approximately 10 dBSPL for 20 kHz frequency band. This is almost the same level as the background noise when there is not high frequency signal. We hence expect that increasing the audio output level could further increase the transmission range.

Figure 3.14 presents the signal constellation points of chirp BOK signals obtained in the same experiment. x and y axis denotes sampled values of up chirp envelope and down chirp envelope, respectively. The constellation points distribute in the first quarter as the envelope is obtained from absolute values. Note that the constellation points of Vega Iron and Galaxy Tab are close to the origin (Fig. 3.14(b)), which means that their recorded volume is smaller than the other devices. This leads to the low frame reception rate of these smart devices. The black dashed line presents the decision bound $y = x$. As up and down chirp signals use the same frequency band, the constellation points of chirp BOK are symmetric in spite of the frequency selectivity of smart devices. Accordingly, the decision bound does not need to be adjusted asymmetrically.

3.4.3 Multi-path Resolution Capability of Chirp Signal

In Section 3.3, we present the chirp envelope obtained from MATLAB simulations. Note that the chirp envelope is not clearly derived in practical recordings due to the



(a) Constellation points (whole picture).

(b) Constellation points (around the origin).

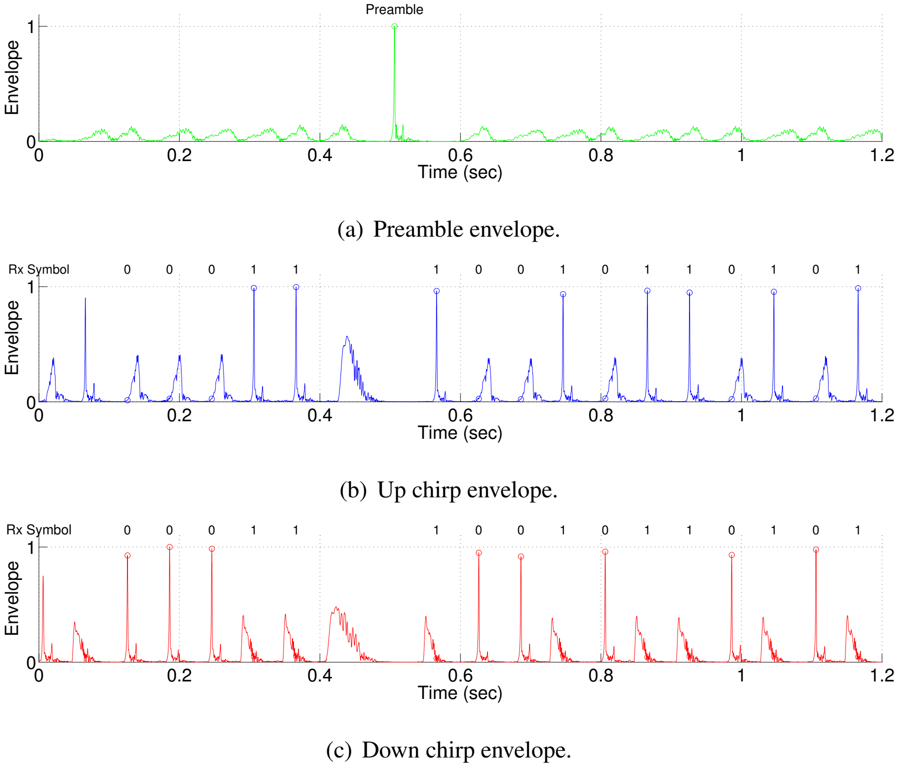
Figure 3.14: Signal constellation points of chirp BOK modem.

channel environment as well as the frequency response of speaker and microphone. In this section, We study the pattern of chirp envelopes obtained from experiment in single-math and multi-path environments.

Single-path Environment

Figure 3.15 presents the permeable, up chirp, and down chirp envelopes in single-path propagation environment. We play a chirp BOK signal using Genelec 6010A reference speaker, and Galaxy Note2 records the acoustic signal in 1 m distance. The experiment is conducted in an open space of a grass field, so we remove the multi-path components as much as possible.

In this specific snapshot, the preamble peak appears at 0.5 sec as seen in Fig. 3.15(a). Then the receiver demodulates the received signal using up chirp envelope and down chirp envelope of Figs. 3.15(b) and 3.15(c), respectively, where the transmitted (and



(a) Preamble envelope.

(b) Up chirp envelope.

(c) Down chirp envelope.

Figure 3.15: Chirp envelopes of Galaxy Note2 in single-path environment.

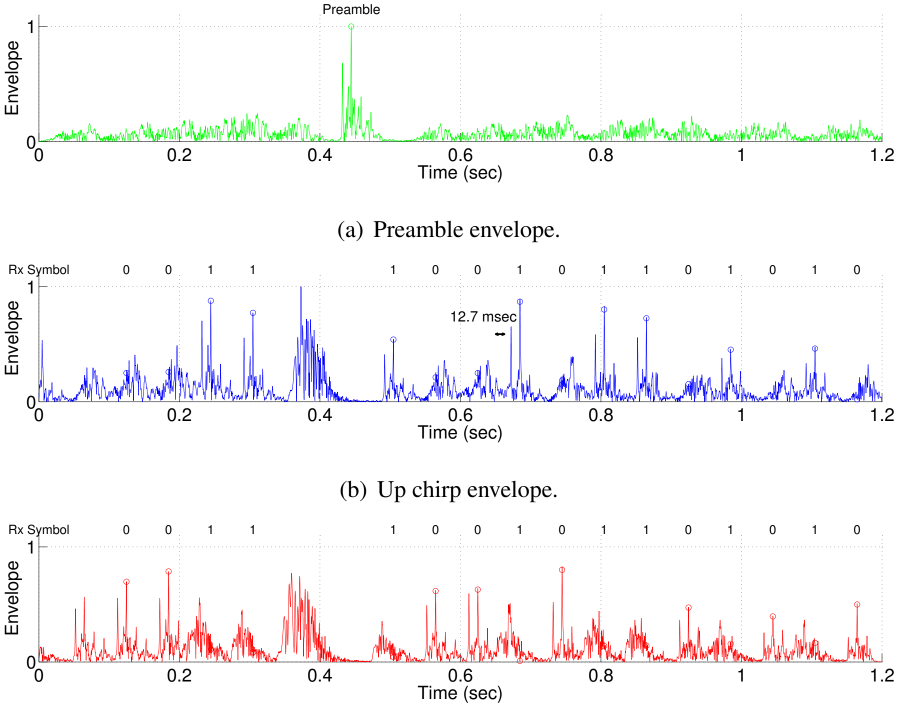
eventually the received) bits are denoted above the envelope curves. Note that the envelope peaks are clearly derived from the envelope detector. The receiver can sample the envelope peaks as seen in circle marks in the figure, and successfully demodulate the transmitted bits.

Interestingly, the up chirp envelope depicts saw shaped peaks when down chirp is received, and this is the same for the down chirp envelope. This comes from the non-ideal orthogonality of chirp BOK signals, i.e., the cross correlation of up and down chirps is not zero. In an ideal case (e.g., Fig. 3.9), the cross correlation should be shaped in a square form. However, the loss of high frequency signals at the microphone distorts this into saw pulse.

Multi-path Environment

One of the reasons why chirp signal is robust in indoor environments is its capability of resolving multi-paths. We first study the envelope curves of chirp signals in typical indoor communication environment. Figure 3.16 depicts the chirp envelopes in multi-path environment. This experiment runs in a large office room. We again use Genelec 6010A reference speaker and Galaxy Note2 smartphone, separated by 3 m. The multi-path is artificially generated by placing a partition screen, parallel to the LOS link between the speaker and smartphone.

We observe clustered peaks of preamble envelope in Fig. 3.16(a). This multiple peaks appear due to the multi-path propagation. Each distinct peaks of Fig. 3.16(a) denotes different multi-paths, where two of them are distinct multi-path components, separated by 12.7 msec (equivalently, 4.3 m path difference). This two distinct multi-path components affects the up chirp envelope and down chirp envelope, too, as shown in Figs. 3.16(b) and 3.16(c). Even though the multiple peaks from the multi-path components are mingled in the envelope curves, the receiver can differentiate the mingled



(a) Preamble envelope.

(b) Up chirp envelope.

(c) Down chirp envelope.

Figure 3.16: Chirp envelopes of Galaxy Note2 in multi-path environment.

signal from the envelope. Note that our receiver tracks the highest peaks in the envelope curves. In this specific snapshot, the receiver picks the latter dominant multi-path component as its received strength appears to be the maximum. By tracking the envelope samples, the receiver can successfully recover the received symbols in multi-path environment.

More into detail, the multi-path resolution of the chirp signal depends on the width of the main lobe of its auto correlation envelope [34]. From Eq. (3.10), the width of the main lobe of the auto correlation envelope is $1/B$ sec where B denotes the bandwidth of the chirp signal. Suppose a two-ray propagation model of chirp signals. We assume

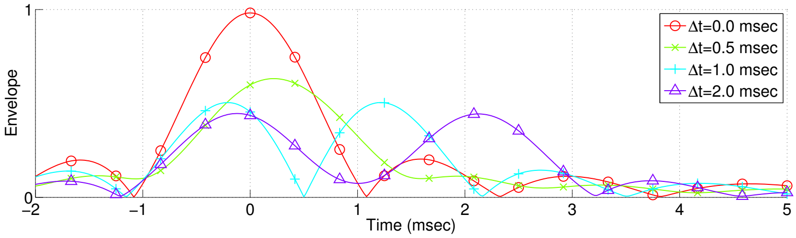
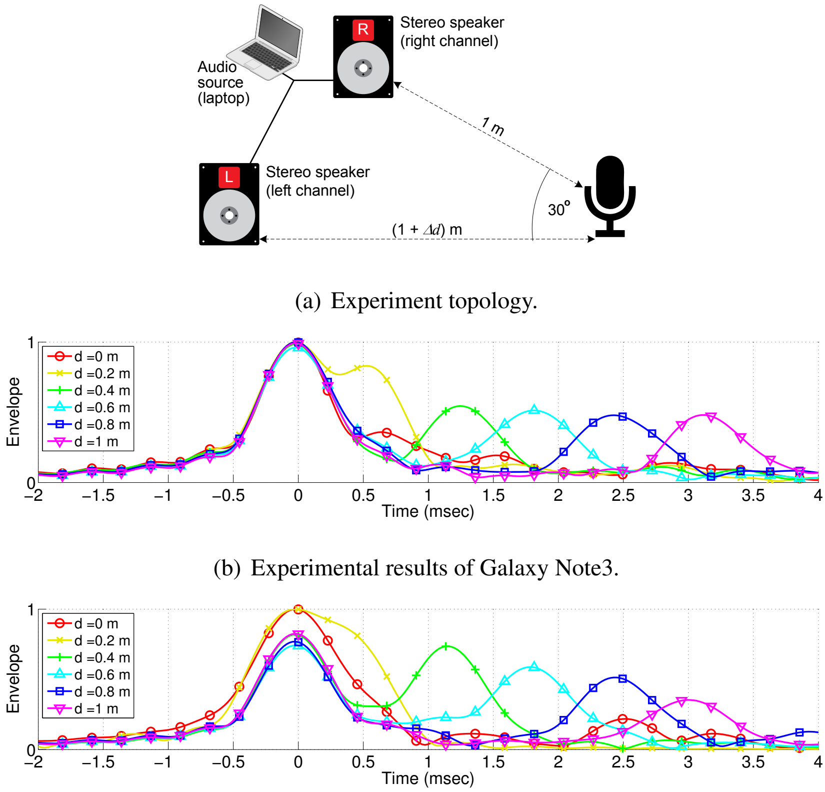


Figure 3.17: Chirp envelopes of two-ray propagation model obtained from MATLAB.

that the LOS component arrives at the receiver at time $t = 0$, and second signal arrives at $t = \Delta t$. Note that the delayed arrival of the chirp signal produces delayed peak of envelope, where its location shifts to $t = \Delta t$. Figure 3.17 depicts an simulation results of this two-ray propagation model, for the chirp signal sweeping from 1 kHz to 2 kHz. We observe that two multi-path components produce distinct envelope peaks when the path difference is greater than or equal to 1 msec, which is equivalent to the inverse of chirp bandwidth 1 kHz. Therefore the chirp signal sweeping B Hz can resolve two different chirp signals traversing with $1/B$ sec path difference.

Considering 2.5 kHz bandwidth of our signal design, the proposed chirp BOK modem can differentiate 0.4 msec difference of multi-path delays in an ideal case, which is equivalent to 0.14 m path difference for 340 m/sec propagation speed of sound. In practice, however, this could be worse due to the decreased chirp sweeping band caused by frequency selectivity, as shown in the following experiment.

In order to assess the capability of multi-path resolution, we emulate the multi-path propagation of sound waves by using stereo speakers. We play chirp preamble through stereo speakers connected to a common audio source in an open space, as shown in Fig. 3.18(a). We make the path difference of two signals by controlling the distance between microphone and left channel speaker. Figure 3.18 presents the envelopes of Galaxy Note3 and Optimus G2, while those of the other devices are omitted as they



(a) Experiment topology.

(b) Experimental results of Galaxy Note3.

(c) Experimental results of Optimus G2.

Figure 3.18: Experiment topology and results in stereo chirp transmission.

have similar patterns. The minimum distance that we can observe distinct envelope peaks of stereo signals differs by model. In general, most devices are found to be able to differentiate the multi-path signals separated by 1.18 msec (equivalently, 0.4 m path difference). This is slightly worse than the theoretical capability 0.4 msec (equivalently, 0.14 m path difference) as chirp signals are partly lost due to the frequency response of speakers and microphones.

3.4.4 Symbol Sampling and Doppler Shift

Wireless communication usually suffers from Doppler shift due to mobility, and this is the same for acoustic communications. We study the effect of Doppler shift on chirp BOK modem in this section.

Doppler shift causes frequency shift to chirp signals, which results in shift of peaks in envelope output [34]. Suppose that the frequency of recorded chirp BOK signal is changed by f_D due to the Doppler shift. Then the auto-correlation of the recorded audio is given by

$$\psi_D\left(t\right) = T\Big(1 - \frac{|t - t_D|}{T}\Big)\text{sinc}\left[\pi B\Big(1 - \frac{|t - t_D|}{T}\Big)\right]\cos\left(2\pi f t\right), \tag{3.27}$$

which is a t_D time-shifted version of Eq. (3.7). t_D can be numerically obtained form Doppler shift f_D caused by Doppler shift v_D as

$$t_D = \frac{f_D}{B} = \frac{f_c T}{B}\frac{v_D}{v_s}, \tag{3.28}$$

for the speed of sound v_s, where f_c, B, and T are center frequency of chirp signal, bandwidth of chirp signal, and symbol duration.

Figure 3.19 depicts an example of envelope peaks shifted by Doppler shift in MATLAB simulation. In a static environment ($v = 0$ m/sec), the envelope peak is supposed to appear at 20 msec, which is equal to the symbol sampling timing. In mobile environment, however, Doppler shift makes the peaks move forward. For example, Eq. (3.28) yields 0.489 msec shift of the main lobe for $v = 1$ m/sec (square marks in Fig. 3.19). Therefore, the symbol sampling timing needs to be adjusted in mobile environment in order to enhance the signal reception rate.

We adjust the symbol sampling timing by finding *local maxima* around the original sampling points. Considering that 1 m/sec walking speed causes approximately up to 0.5 msec peak shift depending on the moving direction [34], we choose 1 msec

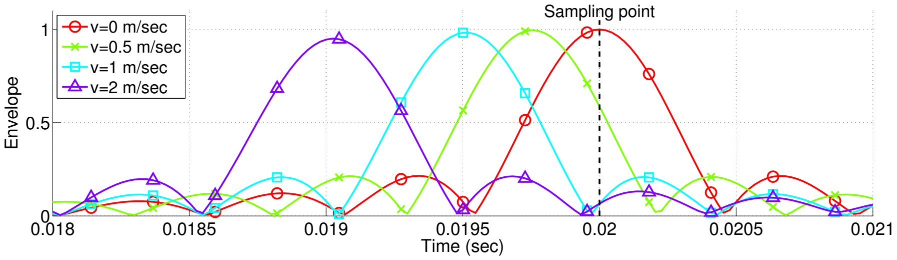


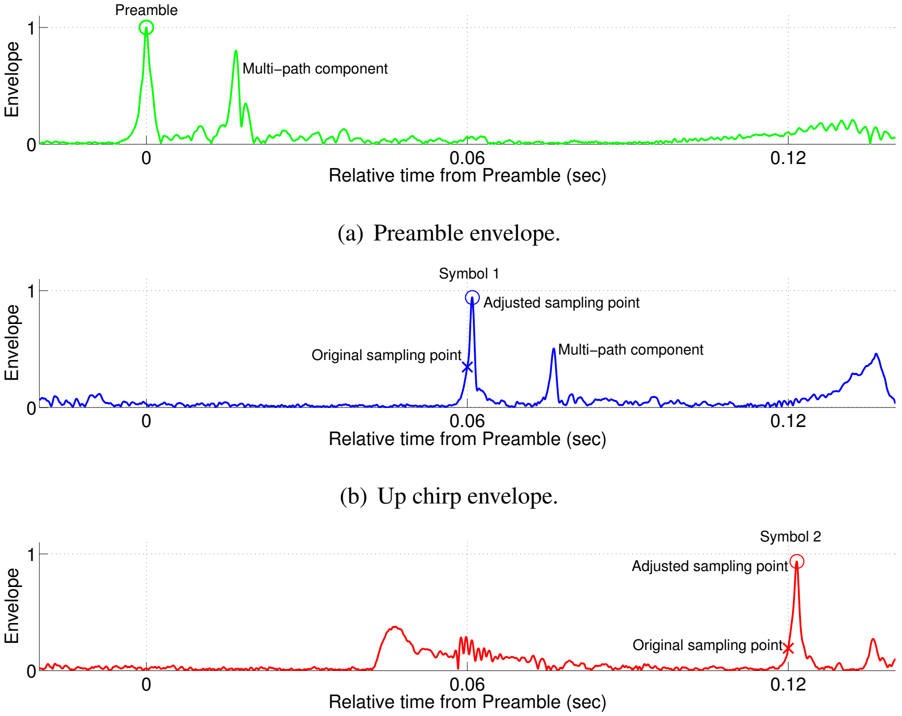
Figure 3.19: Shift of peaks due to Doppler shift, assuming 20 msec symbol is transmitted.

searching range to cover both directions moving forward and away. Remind that the receiver judges the received symbol by comparing the sampled values of up chirp and down chirp envelopes. Therefore the difference of up chirp and down chirp envelopes can be considered as relative signal-to-noise ratio (SNR) of the received signal, just as the Euclidean distance of constellation points in PSK demodulation is regarded as the SNR of the received signal. The proposed sampling timing t_k^* of k-th symbol is represented as follows.

$$t_k^* = \underset{\hat{t}_k - \tau \le t < \hat{t}_k + \tau}{\arg\max} \left\{ \left| e_{\mathrm{U}}(t) - e_{\mathrm{D}}(t) \right| \right\}, \tag{3.29}$$

where $e_{\mathrm{U}}(t)$ and $e_{\mathrm{D}}(t)$ represent up chirp envelope and down chirp envelope, respectively, and $\hat{t}_k$ denotes the expected symbol sampling timing of symbol k from the location of preamble t_{P}, i.e., $\hat{t}_k = t_{\mathrm{P}} + k(T + G)$ for symbol duration T (20 msec) and guard interval G (40 msec). The searching window τ is set to 0.5 msec as discussed above.

Figure 3.20 presents the experimental results of adjusted sampling timing. We slowly move a smartphone (Galaxy Note2) back and forth in front of a speaker, and then obtain the corresponding envelope output. According to the frame structure, the first symbol is supposed to appear at 60 msec after the preamble (x mark at 60 msec



(a) Preamble envelope.

(b) Up chirp envelope.

(c) Down chirp envelope.

Figure 3.20: Experimental results of Doppler shift and shifts of envelope peaks.

in Fig. 3.20. Due to the Doppler shift, however, the first peak appears approximately 1 msec later (o mark at 61 msec in Fig. 3.20). By searching for the local maxima, we can correctly sample the optimal peak.

It should be noted that extending the searching range can enhance the capability to overcome Doppler shift, while too wide range can increase the receiver complexity. Especially in a severe multi-path environment, indistinct multiple peaks make the symbol timing decision even harder. As our future work, we plan to design a *Rake receiver* along with the proposed symbol timing adjustment algorithm in order to appropriately combine multiple peaks of multi-path propagation.

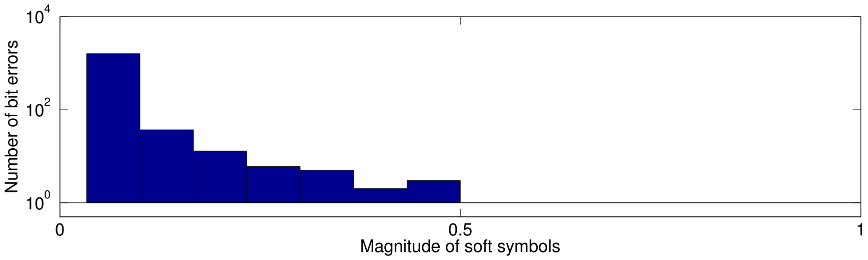


Figure 3.21: Histogram of BER in function of soft value.

3.4.5 Selective combining

We evaluate the BER gain of selective combining of Section 3.3.4 and discuss its overhead in implementations. We play the chirp signal using a TV in a typical indoor environment, and record the chirp signal using various smart devices in 10 different locations. MATLAB is used to analyze the trace of received signal in order for the fair evaluation of the proposed selective combining against a single reception.

Figure 3.21 depicts the histogram of the number of bit errors in function of the magnitude of soft symbols, i.e., . Note that the soft symbols with small magnitude are error-prone, and hence, it is reasonable to accumulate the difference of up and down chirp envelope outputs to exclude the symbols that are expected to be erroneous.

By applying the selective combining with decision variable η_i, we evaluate its BER of various smart devices in Fig. 3.22. We compare two different soft combining algorithms, (1) repetition combining that combines the soft symbols regardless of its bit error events, and (2) selective combining that stops combining when an error-free frame is received. We first verify that the selective combining improves BER, where each iteration halves the BER in general. It is also seen that selective combining slightly outperforms repetition combining by excluding severely corrupted symbols.

Note that the repetition decoding requires additional recording to collect copies

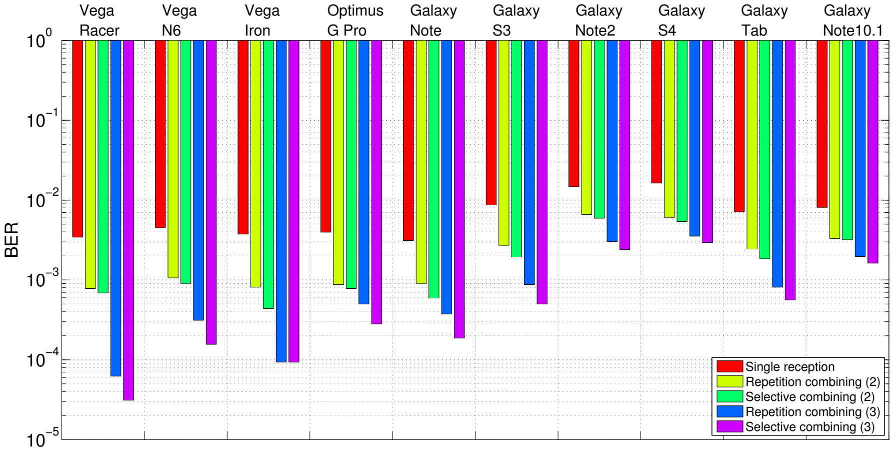


Figure 3.22: BER of selective combining emulated by the acoustic signal trace of various smart devices.

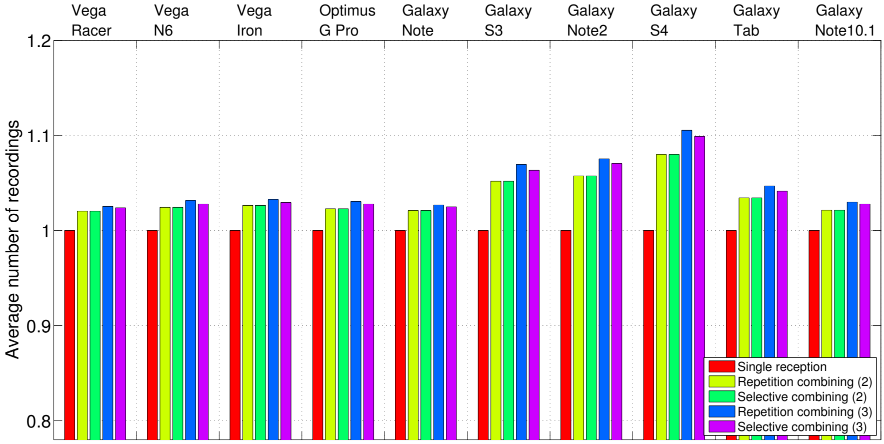


Figure 3.23: Average recording overhead of selective combining emulated by the acoustic signal trace of various smart devices.

of acoustic signal. Figure 3.23 presents the average overhead of repetition combining schemes, compared with a single reception case. The detailed increase of the overhead varies over device and/or channel state, i.e., the device in more error-prone situation requires more repeated decodings. However, we observe that the average overhead for the repetition decoding is not very significant as they are less than 10% in general. We also see that the selective combining with three repetitions requires less additional recordings compared with repetition decoding, as the receiver might stop the repeated decoding if the second copy can fully recover the bit errors.

3.5 Summary

In this chapter, we have designed the inaudible chirp BOK symbol and corresponding frame structure. Raised cosine window shapes each data symbol in order to reduce the out-of-band leakage. This guarantees the inaudibility of the chirp signal as well as provides bandpass ingress filter at the receiver. The frame is designed to include 16 data symbols and one preamble. The preamble is used as a delimiter of a frame, assuming that the transmitter repeatedly plays the same chirp signal frame. The proposed chirp signal frame achieves approximately 16 bps data rate accordingly.

We also have proposed an efficient receiver architecture that is specifically designed for the chirp signal reception of smart devices. The convolution of the matched filter is substituted by FFT and IFFT in order to drastically reduce the computing complexity. The proposed frequency-domain matched filter is combined with Hilbert transform to realize the envelope detector. The envelope detector derives the envelope of auto correlation function, so that the receiver can effectively sample the peak values. This enhances the detectability of preamble as well as SNR of data symbols.

The reception process of chirp BOK signal is presented and discussed in the view point of practical implementation on smart devices. The proposed process flow of the receiver is (i) recording, (ii) signal detection, (iii) preamble detection, and (iv) symbol demodulation. The recording duration is defined to include at least one full-preamble in each recording. Hence the receiver can efficiently detect the preamble. The implementation of the receiver architecture can save its computing memory by sharing the FFT results that are commonly used by three envelope detectors.

The performance of the proposed receiver is evaluated in indoor environment. Our experimental results show that the proposed signal can be delivered over long-range indoor acoustic channel for most smart devices, in spite of their various and selective frequency response. The constellation point of chirp BOK signal is investigated, and it is turned out that the chirp BOK receiver can be kept simple as the designed chirp signal set is symmetric in frequency domain.

We study the multi-path resolution capability of the chirp signal in practical communication scenarios. In an ideal case, the designed chirp signal can differentiate multi-path components that arrives with path difference greater than or equal to 0.14 m (or 0.4 msec, equivalently). Due to the imperfect frequency response of smart devices, however, the multi-path resolution is shown to be 0.4 m (or 1.18 msec, equivalently) in practice. We also investigate the effect of Doppler shift on the reception of the chirp signal. Mathematical derivation testifies that the Doppler shift results in time-shift of envelope peaks, where 1 m/sec mobility makes 0.5 msec peak shifts. Based on this observation, we propose sampling timing adjustment algorithm that searches for local maxima around sampling timing. The sampling criterion is given by the difference of up and down chirp envelopes, and it is equivalent to the sampling points that maximizes SNR.

As a future work, we plan to further simplify the receiver by optimizing its calculations. Currently we are using 65,536 point FFT and IFFT at the envelope detector.

The complexity of FFT is known to be $O(N \log N)$, hence reducing the FFT size can relax the computing overhead of the receiver. Note that we can reduce this FFT size by applying the methods such as*overlap-and-add*, where the optimal FFT size and the number of division blocks can be found through mathematical derivation.

As discussed in Section 3.4.3, the chirp signal can differentiate the multi-path propagations thanks to its good auto correlation characteristic. We expect that we can further enhance the signal reception rate of the proposed receiver by combining the multiple peaks that are collected from envelope curves. Note that this works similar to *Rake receiver* of CDMA — another spread spectrum modulation. For multi-path components that span over a symbol duration, implementing *equalizer* can cope with the ISI. Moreover, equalizer provides a chance to remove the guard interval of the current design, which is one of the main reason of its low bit rate. We are currently studying the Rake receiver and equalizer implementation in order to improve the data rate and enhance the signal reception probability in long-distance communications.

Chapter 4

Applications of Chirp BOK Receiver

4.1 Introduction

The use of smart devices have been a routine activity nowadays. Not only smartphones, but also wearable smart devices such as Galaxy Gear or Google Glass are widely being used. These smart devices are equipped with voice UI. Accordingly, there have been a number of tech start-ups that introduce new services based on the aerial acoustic communication that can utilize the voice UI as an communication interface [31–33]. As discussed in Chapter 1, however, most of these services are limited in their applications due to the short transmission range and/or demands on special hardwares for ultrasound transmission and reception.

We have designed chirp BOK signal and corresponding receiver in Chapter 3 in order to overcome the limitations of the previous work. The chirp signal and receiver design in Chapter 3 supports approximately 16 bps data rate, which is sufficient for short ID transmission. The available services based on the proposed receiver cover (but not limited to) (1) multimedia content recognition, (2) indoor location tracking, (3) social discovery, and (4) proximity network.

Nonetheless, the low data rate is not enough to deliver detailed information to smart devices. In order to compensate the low bit rate of chirp BOK signal, we implement a backend server to which a smart device can send a query. Note that most smart devices have a connection to the Internet through their communication interfaces such as WiFi and LTE. Thus the smart device can fetch more information corresponding to the received 16 bit ID from the Internet with help of our backend server.

Energy consumption issue is one of the major problems of today's mobile smart devices. The smart devices rely on the battery power, which is very limited especially due to their large liquid crystal displays (LCDs). Hence the proposed chirp signal receiver cannot run for 24 hours; if the smart devices contiguously use the microphone for recording, the battery consumption would be too much. In this chapter, we design a low-power operation of chirp BOK receivers based on the signal detection algorithm based on short-length recording. The receiver can stay in an idle state if there is no signal, and this can save the operation power.

One of the application for the proposed chirp receiver is TV content recognition. We assume that a user watches TV while putting his/her smart device around the TV, e.g., on a sofa or a table. TV content is assumed to embed chirp signal that notifies its content ID in a form of inaudible watermark. The chirp signal receiver basically runs as a background service of the smart device, and periodically checks whether there exists the chirp signal or not. On the detection of chirp signal, the receiver checks the TV program that the user is currently watching, and provides additional information, e.g., the actor/actress on the program, product placement (PPL) online markets, etc. A technical challenge of this application is the loss of chirp signal through TV network — note that TV contents typically are encoded in Dolby Digital format, which is an lossy encoding. We evaluate the performance degradation due to this lossy audio codecs in this chapter.

Another application described in this chapter is indoor location tracking for public transportation system and buildings. The motivation is that vast of buses, subways, and buildings are already installed with public announcement (PA) systems for announcement as well as emergency alarms using sound and/or human voice. We utilize the speakers of public transportations and buildings as beacons that broadcast the location information using 16 bit chirp signal. The smart devices decodes the embedded location information, and displays to users. This service is especially useful for those who are hard to hear due to their disabilities — our chirp signal receiver works as a translator that converts acoustic information visible. In this chapter, we tackle the frequency response characteristics for the realization of this service. The PA systems in public transportations and old buildings are likely to be obsolete, so that we need field test results to test the feasibility of such application.

In this chapter, we present our backend server implementation as well as signal detection method. The proposed auxiliary features compensate for the low bit rate of chirp BOK system, and supports low-power operation in smart devices. We also propose two practical applications of the chirp BOK receiver, TV content recognition and indoor location tracking. Throughout realistic experiments and field test results, we verify the feasibility of the applications.

This chapter is composed as follows. Section 4.2 and Section 4.3 describes our backend server implementation and signal detection method, respectively, and discuss technical issues and evaluation results. In Section 4.4, we propose two applications of the chirp BOK receiver – TV content recognition and indoor location tracking, and examines their feasibility based on experiments and field tests. Finally Section 4.5 summarizes the chapter.

4.2 Backend Server Architecture

In this section, we aim to design a backend server in order to overcome the low bit rate of chirp BOK receiver. The backend server basically manages a database that contains uniform resource locators (URLs). On the reception of query with an ID from a smart device, the backend server responds with corresponding URL that the smart device needs to contact. The smart device can fetch virtually infinite amount of information from the URL, and hence, the limited data rate of the chirp signal receiver can be resolved.

We assume that the smart devices are connected to the Internet through WiFi and/or 4G interface. Note that this assumption is reasonable as most smart devices are equipped with multiple communication interfaces, and at least one of the interfaces is on in order to provide connectivity to the users. We also assume that the URL of the backend server is known to the smart device in off-line. One can hard-code the URL of the backend server, as our chirp signal receiver runs as a software modem in the application layer. In the followings, we first describe our implementation of the backend server and communication protocol between a smart device and server. Then technical challenges are discussed with preliminary evaluation results for further developments.

4.2.1 Implementation of Backend Server

Traditionally the development of server system has been using server computers that provides powerful computing power and memory space compared with personal computers. This is due to the roll of the server computer that deals with simultaneous connections from numerous clients. The server computers generally are equipped with multiple core processors (CPUs) and huge repositories (HDDs). Thus, the volume of the server computer is too large to be merged into a small package. A number of server racks are installed in server rooms instead.

This *physical servers* are hard to be managed not only due to the care of computer machines itself, but also due to its scalability. The specification of server performance is based on the expectation of connections at the beginning of the installation. If the target specification is over estimated, it would be too costly to provide that level of service. On the other hand, it costs extra capita to upgrades the servers in case of underestimation.

As a solution of the problems above, *virtual servers* are being hosted recently. The host prepares a high-performance server system, where its computing resources can be distributed based on the cloud computing. Service users then requests for a part of the resource of the host server on-demand, so it is very cost-effective in terms of both capital expenditure (CAPEX) and operating expenses (OPEX).

Our backend server is implemented using Amazon web service [60]. We run Apache 2.2.22 HTTP server with PHP 5.3.10 on Ubuntu 12.04.4 OS. The backend server basically manages a MySQL database that contains (i) API Key which is a unique ID of a specific application that is combined with our chirp signal receiver, (ii) 16 bit ID received and demodulated by the receiver, and (iii) URL to be contacted by the smart device, and auxiliary comment to be displayed on the response. This enables the smart device to fetch virtually infinite amount of data from the URL with help of our backend server.

It should be noted that our chirp signal receiver is implemented as a software modem, and distributed as a library for JAVA developers of Android OS. As we discussed in Chapter 3, the receiver library can be combined with an arbitrary Android application to support signal reception functionality. This means that there could be multiple applications that have received the same chirp signal. Our backend server needs to differentiate those in the following situations.

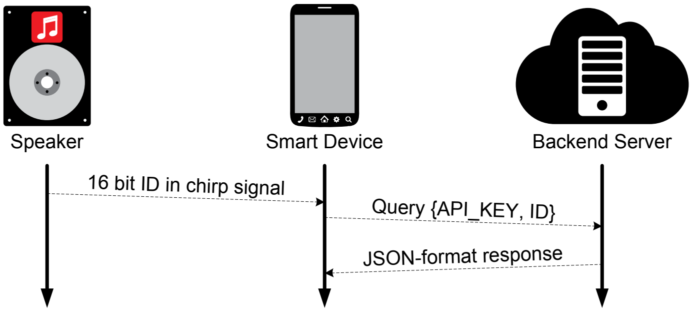
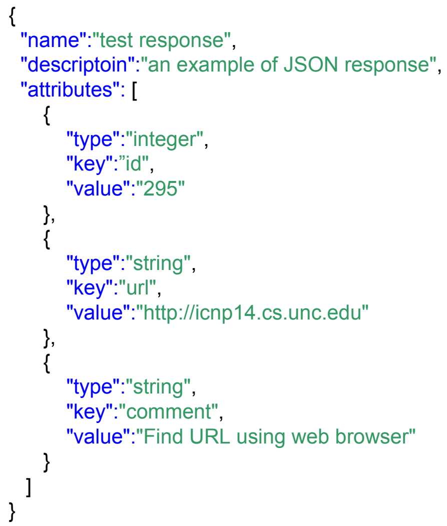


Figure 4.1: Message flow of smart device and backend server.

Suppose that two different second screen applications for TV content recognition are combined with our chirp signal receiver. On the reception of the chirp signal embedded in a music program of TV, one application tries to display the information of the artist, while the other does to guide the user to an online album market. In this scenario, our backend server cannot differentiate two queries from the applications if the queries contain only the 16 bit ID in the chirp signal. We assign API keys to identify the applications that are combined with our chirp signal receiver. In the example above, our backend server responds with corresponding URLs to two different applications by using the API keys.

4.2.2 Operation of Backend Server

Figure. 4.1 presents the message flow of smart device and backend server. On the reception of chirp signal, the receiver first tries to demodulate the received audio. The receiver then sends a query to the backend server with API Key and 16 bit ID that is successfully decoded from the audio. The backend server then responds corresponding URL and comment. The application developer can freely design the processing of the received URL and comment.



```
{
  "name":"test response",
  "descriptoin":"an example of JSON response",
  "attributes": [
     {
        "type":"integer",
        "key":”id",
        "value":"295"
     },
     {
        "type":"string",
        "key":"url",
        "value":"http://icnp14.cs.unc.edu"
     },
     {
        "type":"string",
        "key":"comment",
        "value":"Find URL using web browser"
     }
  ]
}
```

Figure 4.2: An example of JSON response format.

The response from our backend server is based on JavaScript object notation (JSON) [61]. JSON is a data interchange format that is very simple and easy to read and write for human, and standardized by ECMA 404. Through the name of JSON includes JavaScript, the application of JSON is not limited to JAVA but spans for C, C++, JAVA, Perl, Python, and many others, thanks to its simple structure. Basically a JSON is a collection of name/value pairs as shown in Fig. 4.2. So the receiver can easily parse a JSON message.

Management of ID Information

The database of the backend server is meaningful only when the URL information is combined with the chirp signal ID. An application developer first register an application API key to our backend server. This creates a repository of the database whose entry is a tuple of 16 bit ID, URL, and comment, and the repository is unique to each API key. The application developer can add, change, and remove the entry of the database on his/her own demand.

The policy of using database to deliver the related information reduces the effort to modify the acoustic signal in case of context change. Suppose that the developer of the second screen application in the above example now wants to invite the users to a concert. If the URL is hard-coded in the acoustic signal, then the acoustic signal needs to be modified so that different information can be delivered to the users. In our system, however, the application developer just needs to change the URL entered in our backend server's database. The backend server then responds with a new JSON message that contains modified URL. This can be much more cost-effective, considering that the audio mixing in broadcast companies is much more exhaustive process.

Scalability of Backend Server

Note that our backend server is implemented using Amazon web services, which automatically controls its computing power to cover large volume of traffic. The capacity of manageable traffic volume actually depends on the service program that we pay for. Our preliminary test on the large volume of traffic shows that the current implementation can handle up to 7,000 simultaneous clients in 3.78 sec round trip time (RTT) delay. We are still working on the improvement of the backend server architecture for efficient traffic and database handling. For example, we can divide the backend server into two different functionalities; one is for traffic control and the other is for database management, which is our ongoing work. In this dissertation, we continue to resolve technical issues on the chirp signal receiver from the next section.

4.3 Low Power Operation for Smart Devices

The acoustic signal reception of a smart device is based on the recording and signal processing. The communication interfaces such as WiFi, Bluetooth, and LTE has dedicated signal reception and processing chipset in the smart device, and they are well-optimized in general for low power operation. However, it is not the case for acoustic communication. Recording process itself consumes more battery power than wireless communications, since the operation of microphone is a mechanical process — the vibration of membrane, and conversion of mechanical energy to electric energy is power consuming. The signal processing of the receiver keeps the application processor (AP) in an active state so that battery consumption further increases. Therefore, if a smart device keeps recording and processing contiguously, then the battery would run out in a few hour.

In this section, we revisit the reception process in Chapter 3 and investigate the chance for saving its power consumption from adaptive control of reception timing. We point out that shortening the recording duration of microphone and eliminating unnecessary signal processing in false alarms can reduce the power consumption of the chirp signal receiver. To this end, efficient *signal detection* method is designed, and its technical issues on implementation using FFT is discussed in detail.

4.3.1 Reception Process of Chirp BOK receiver

Our assessment begins by reminding Fig. 3.7 in Chapter 4. The receiver is designed assuming that the transmitter (i.e., loud speaker) repeatedly plays chirp signal. That is, the receiver basically expects the existence of the acoustic signal when it triggers the recording in the reception process. If the receiver proceeds signal processing after the recording regardless of the actual existence of the chirp signal, then the comput-

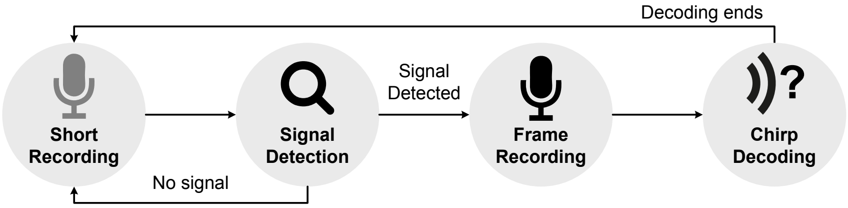


Figure 4.3: Two-step recording for signal detection and demodulation.

ing power is wasted in vain. This is very likely as the acoustic signal is not always being played near a smart device. Therefore we aim to design an intermediate *signal detection* module just before the preamble detection.

We can further reduce the operating power of the chirp receiver by controlling the recording duration for signal detection. It is obvious that the longer the microphone records, the more battery power is consumed. Hence the receiver would better to shorten the recording time for the signal detection. The receiver can re-record 1.2 sec full frame only when the receiver judges that there is a signal, as shown in Fig. 4.3. It should be noted that, however, the shortest recording duration is limited by the OS of smart devices, where its detailed value varies over OS version and smart device.[1] The too short recording also might deteriorate the signal detection probability, which results in miss detection.

Hence we aim to design a signal detection algorithm for the proposed chirp signal that balances power consumption and miss detection/false alarm probability. Specifically, we try to balance the trade-off by controlling the recording duration for signal detection. We also take into account the receiver architecture with FFT in the design of signal detection algorithm.

[1] We have investigated a number of smart devices and Android OS versions, and we claim that the minimum recording size is approximately 4,000 bytes in general, which is close to 0.1 sec duration for 44,100 Hz audio sampling rate.

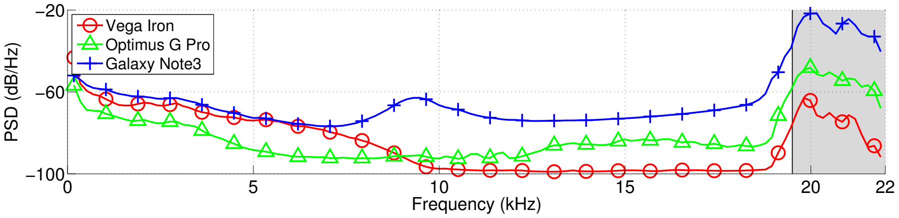


Figure 4.4: PSD of recorded audio at three smart devices.

4.3.2 Revisit of Signal Detection in Wireless Communications

We first revisit the typical signal detection algorithms in wireless communications, namely, energy detection and SNR detection. Discussion on pros and cons of those algorithms leads to a new metric that is designed for the chirp signal detection in acoustic communications.

Energy Detection

A typical method to find electromagnetic signal is *energy detection*. A receiver claims the existence of a valid signal if the received signal energy is greater than or equal to a certain threshold. Energy detection is widely adopted especially for the wireless communication technologies that operates in unlicensed spectrum such as industrial, scientific, and medical (ISM) band, e.g., WiFi and Bluetooth, as it is simple and easy to support coexistence among different wireless signals.

In case of acoustic signal in 19.5∼22 kHz band, however, it is difficult to determine the detection threshold because the frequency responses of different audio interfaces are significantly different. Figure 4.4 plots the PSD of recorded chirp signal at Vega Iron, Optimus G Pro, and Galaxy Note3 devices in 1 m distance. The difference of recording volume, frequency response, and hardware specification makes received

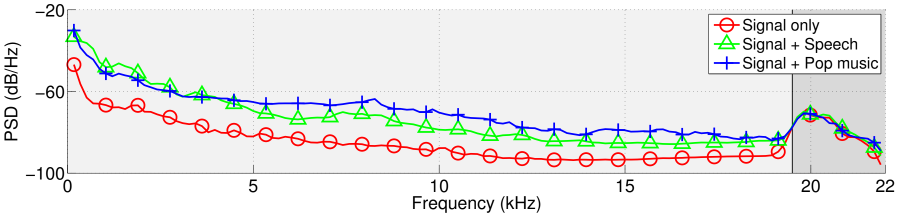


Figure 4.5: PSD of different types of audio at Vega N6.

energy in the signal band 19.5∼22 kHz (shaded region in Fig. 4.4). Therefore the energy detection cannot be applied to acoustic signal detection in high frequency band.

SNR Detection

Alternatively, one can detect a signal based on SNR of the signal. Valid signal is assessed by continuously measuring the SNR of the received signal. In this case, the receiver first needs to define the noise band of SNR calculation. Remind that we aim to reduce the operational power of the receiver by shortening the recording time. We assume that the noise band is different from the signal band as the receiver cannot continuously monitor the in-band signal. The receiver calculates the SNR from the PSD of the recorded audio.

Let us assume that the receiver regards the noise band as the frequencies under 19.5 kHz. Then the noise variance is significant as the noise bandwidth is too wide. The noise level dynamically varies especially for the acoustic communication as audible-band background environmental noise changes over time. For example, human voice mingled with the chirp signal can decrease the SNR, and the signal is missed eventually. Figure 4.5 illustrates an experimental results on this phenomena. We record the inaudible chirp signal using Vega N6 device, while the signal is mixed with TV sounds (pop music and news speech). The PSD patterns under 19.5 kHz (lightly shade region

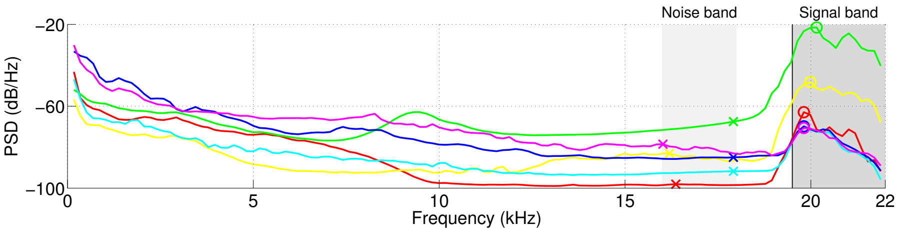


Figure 4.6: An example of calculating the peak PSD ratio.

of Fig. 4.5) of three scenarios are different. Hence SNR detection is also hard to be used for chirp signal detection.

4.3.3 Chirp Signal Detection using PSD

Instead, we calculate the noise power in 16∼18 kHz band, in which environmental noise rarely exists as we observed in Fig 2.1. We define *peak PSD ratio* as a metric to detect the chirp signal in this dissertation. Basically both signal and noise power can be calculated by accumulating PSD of the corresponding frequency band. We further simplify the signal detection by substituting the summation into maximum searching in order to reduce the computing complexity of summation. That is, we simply find the maximum values of PSD in both in-band and out-of-band; and then the peak PSD ratio is defined as the ratio of two maximum values in dB scale.

Figure 4.6 presents the graphical illustration of calculating peak PSD ratio. Darkly shaded and lightly shaded regions present signal band and noise band, respectively. The receiver calculates the peak PSD ratio by finding the maximum values of signal band (o marks) and noise band (x marks).

Table 4.1: False alarm and miss detection of chirp signals

		Signal state	
		On	Off
Decision	On	True positive	**False alarm**
	Off	**Miss detection**	True negative

Peak PSD Ratio and FFT Size

Note that the PSD can easily be obtained from FFT of the recorded audio. It is natural to include the calculation of peak PSD ratio in our chirp receiver architecture, as seen in Fig. 3.8. Now we find the optimal FFT size for the signal detection. As discussed above, short recording (and eventually short FFT size) can save the computing power, while degrades signal detection capability due to insufficient samples for decision.

We adopt *receiver operating characteristic* curve for the criterion of selecting optimal FFT size [62, 63]. In signal detection theory, receiver operating characteristic curve illustrates the performance of binary classifier system in function of the decision threshold. Table 4.1 summarizes false alarm and miss detection events.

We collect 100,000 recorded samples each when the chirp signal is on and off, using 10 different smart devices. The recorded sample has 1.2 sec duration with 44.1 kHz audio sampling rate, where its duration is equivalent to the recording size of the proposed chirp receiver. We first check the effect of FFT size on the signal detection. The FFT size varies from 4,096 point to 65,536 point, where 4,096 point comes from the minimum recordable duration of Android devices, and 65,536 point is the FFT size of the proposed chirp receiver. For the calculation of PSD from the FFT, we trim the first N samples of the recorded audio, calculate its N-point FFT, and then obtain its PSD

$P_N\left[w\right]$ in discrete frequency domain by

$$P_N\left[w\right] = \frac{2}{N \cdot F_s} \left|\mathcal{R}_N\left[w\right]\right|^2, \; w = 1, \cdots, N/2, \tag{4.1}$$

where F_s denotes the audio sampling frequency. $\mathcal{R}_N\left[w\right]$ is N-point FFT of recorded (discrete) audio $r\left[n\right]$ for $n = 1, 2, \cdots, N$, i.e.,

$$\mathcal{R}_N\left[w\right] = \mathcal{F}\left\{r\left[n\right]\right\}, \; n = 1, \cdots, N. \tag{4.2}$$

Equation (4.1) calculates the PSD of positive frequency terms as it is seen in its frequency range $w = 1, \cdots, N/2$, and the scalar factor 2 in Eq. (4.1) takes into account the loss of negative frequency terms. Finally the peak PSD ratio Λ_N is calculated as

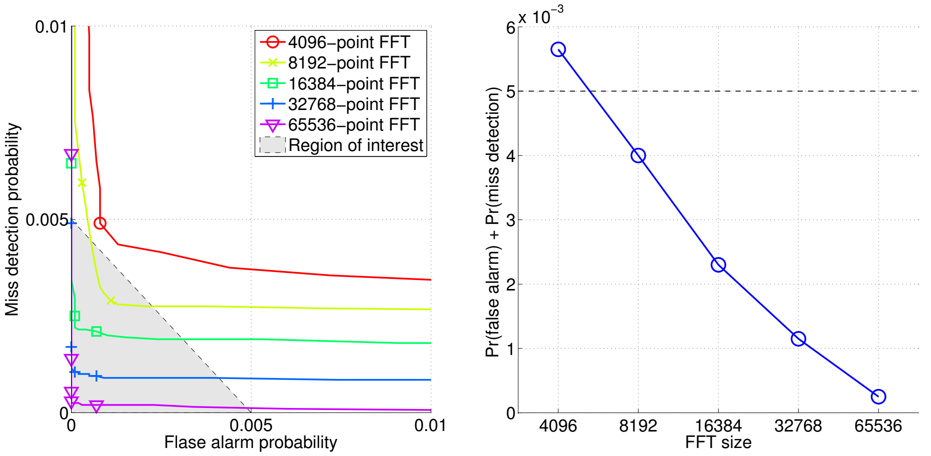
$$\Lambda_N = 10 \log_{10} \left(\frac{\max\limits_{w \in W_{N,\mathrm{s}}} \left\{\mathcal{R}_N\left[w\right]\right\}}{\max\limits_{w \in W_{N,\mathrm{n}}} \left\{\mathcal{R}_N\left[w\right]\right\}} \right), \tag{4.3}$$

where $W_{N,\mathrm{s}}$ and $W_{N,\mathrm{n}}$ denotes signal band and noise band, which are functions of FFT size N and audio sampling frequency F_s in kHz, i.e.,

$$W_{N,\mathrm{s}} = \lfloor 19.5N/F_s \rfloor \sim \lfloor 22N/F_s \rfloor, \tag{4.4}$$

$$W_{N,\mathrm{n}} = \lfloor 16N/F_s \rfloor \sim \lfloor 18N/F_s \rfloor. \tag{4.5}$$

Now we calculate the false alarm and miss detection probabilities for detection threshold in a range of $-20 \sim 60$ dB based on the peak PSD ratio given in Eq. (4.1). Figure 4.7(a) illustrates the receiver operating characteristic curve for FFT sizes from 4,096 to 65,536. It is well known that the false alarm and miss detection probabilities in a trade-off relationship. The receiver misses all signal but never warned by false alarms if the threshold is too high. On the other hand, the receiver suffers from frequent false alarms but never misses a signal as it reacts to all stimuli if the threshold is too low. In Fig. 4.7(a), we also observe that the minimum achievable miss detection and false



(a) Receiver operating characteristic curve.

(b) False alarm and miss detection probabilities.

Figure 4.7: Receiver operating characteristic curve and the effect of FFT size.

alarm probability decreases as the FFT size increases. This is obvious as the receiver can make correct decision from sufficient audio samples.

In this dissertation, we set our design object for limiting the sum of miss detection and false alarm probabilities under 0.5%. This is denoted as shaded region in Fig. 4.7(a) (region of interest). We hence choose 8,192 point FFT size for the signal detection. Figure 4.7(b) plots the minimum of sum of achievable miss detection and false alarm probabilities, which corresponds to the corner points in Fig. 4.7(a). We find the optimal threshold that yields the minimum of the objective function in Section 4.3.4.

4.3.4 Performance Evaluation of Signal Detection Algorithm

We now evaluate the proposed signal detection algorithm in practical communication environment. We first test the feasibility of the proposed metric, peak PSD ratio. We record the chirp signal in various acoustic channel, calculate the peak PSD ratio, and check whether the proposed metric can effectively differentiate the signal existence. Then we find the optimal detection threshold that yields our design objective – minimizing the sum of false alarm and miss detection probability.

Experimental Environment

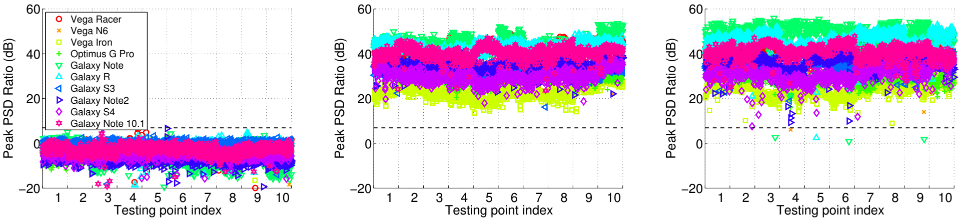
Our experiment is conducted in a typical office environment. As a signal player, we use a smart TV that can receive audio sources from USB memory. Two different types of audio sources are tested in this experiment; one is lossless WAV format, and the other is lossy AC3 format. This aims to validate the feasibility of the proposed metric in practical applications.[2] We choose 10 different locations in the office as testing points. 10 smart devices collect recording data and calculates peak PSD ratio (with various FFT sizes) for 100 times at each testing points in three experimental scenarios; (1) TV is off, (2) WAV signal is being played, and (3) AC3 signal is being played.

FFT Size and Differentiability of Signal Existence

Figure 4.8 presents the history of 100,000 peak PSD ratios collected from 10 smart devices using 65,536 point FFT. Figures 4.8(a), 4.8(b), and 4.8(c) present when TV is off, playing WAV signal, and playing AC3 signal, respectively. One can clearly differentiate signal on and off cases from the figure, where its threshold (7 dB) is plotted with a dashed horizontal line. We also observe from Fig. 4.8(b) and 4.8(c) the signal detection probability of AC3 signal is slightly worse than WAV signal due to its lossy coding. The degradation, however, is not significant as only five recordings out of 100,000 samples are lost in this experiment.

Now we reduce the FFT size to 8,192 points in Fig. 4.9, where the same recording

[2] We discuss the details on AC3 format and signal losses in Section 4.4

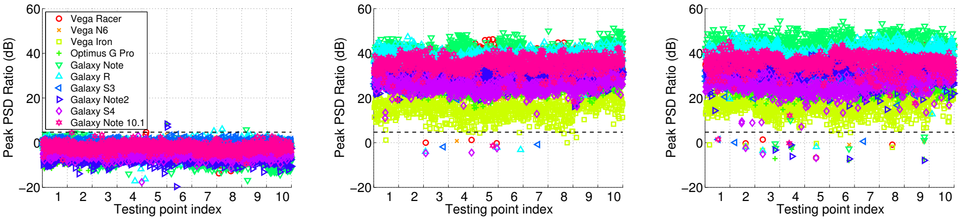


(a) No signal.

(b) WAV signal.

(c) AC3 signal.

Figure 4.8: Peak PSD ratio values in indoor experiment for 65,536- point FFT.



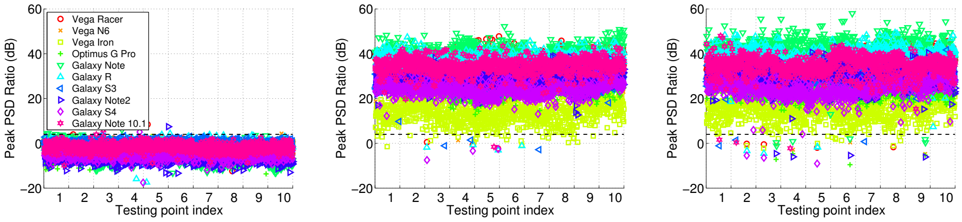
(a) No signal.

(b) WAV signal.

(c) AC3 signal.

Figure 4.9: Peak PSD ratio values in indoor experiment for 8,192-point FFT.

data is used with trimming in the metric calculation. Still we can differentiate the signal existence from the figures, where the optimal threshold is given by 4.5 dB. The degradation of signal detection in AC3 encoded signal is the same as in the previous case. If we further reduce the FFT size to 4,096 point as in Fig. 4.10, however, the number of miss detection and false alarm events increases. This cannot satisfy our design objective of 0.5% miss detection d false alarm probabilities. Hence we claim that 8,192 point FFT can achieve both our design criterion and power power signal detection.

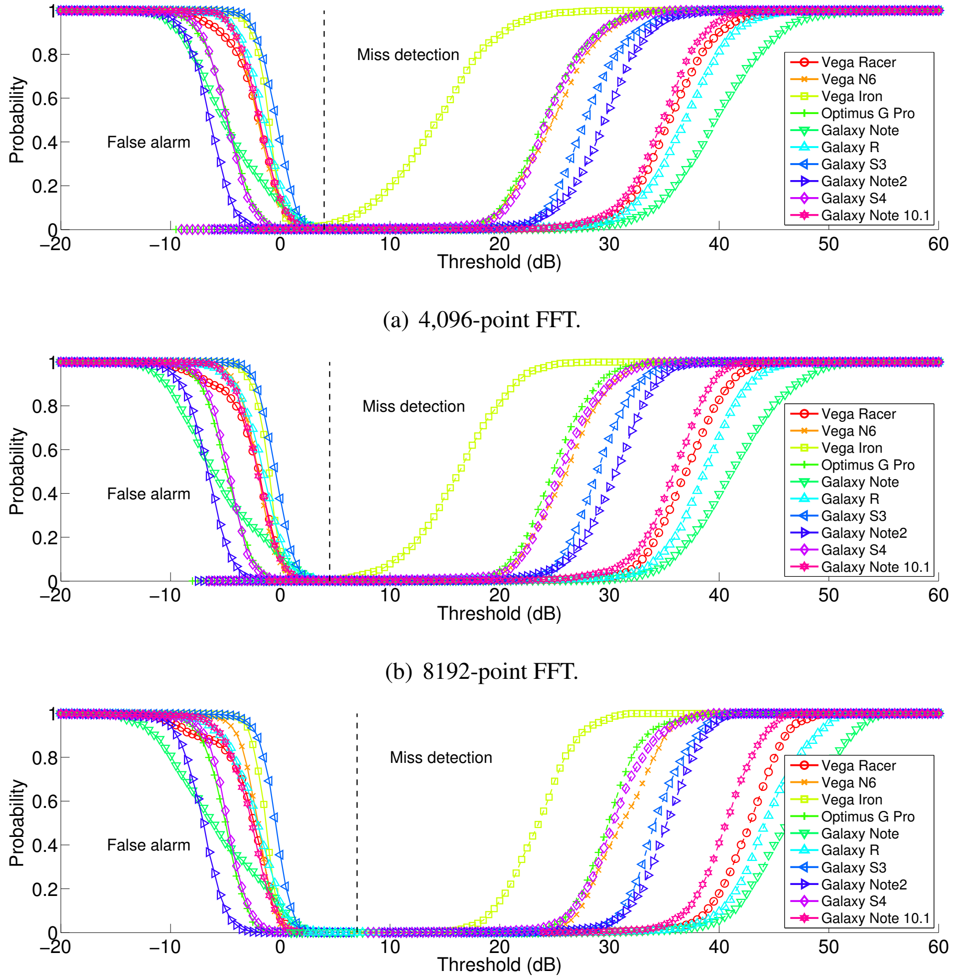


(a) No signal. (b) WAV signal. (c) AC3 signal.

Figure 4.10: Peak PSD ratio values in indoor experiment for 4,096-point FFT.

False Alarm and Miss Detection Probability

Figure 4.7 shows the false alarm and miss detection probabilities of 10 smart devices for various FFT sizes. We numerically find the optimal threshold by sweeping the threshold from -20 to 60 dB and calculating the corresponding false alarm and miss detection probabilities. From Figure 4.7, one can observe that the main reason of miss detection is Vega Iron, which has low recording volume due to its hardware limitation. This disability can also be observed in Figs. 4.8, 4.9, and 4.10. Note that, however, the proposed signal detection metric can effectively differentiate the existence of signal with very small false alarm and miss detection probability. The proposed 8,192 point FFT yields 0.1% and 0.3% probability of false alarm and miss detection, respectively.



(a) 4,096-point FFT.

(b) 8192-point FFT.

(c) 65,536-point FFT.

Figure 4.11: False alarm and miss detection probability of the propose signal detection algorithm.

4.4 Applications of Chirp BOK Receiver and Feasibility Test

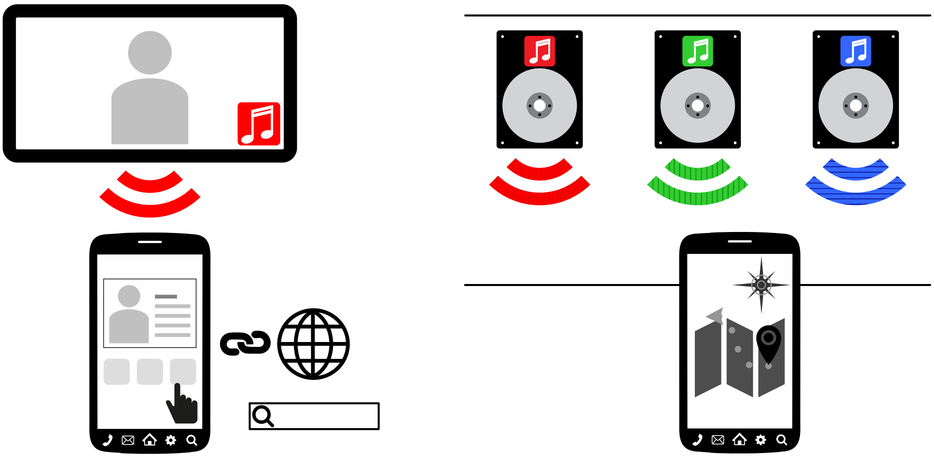
We have presented chirp signal design, receiver architecture, and auxiliary features for supporting the chirp signal receiver. As discussed in Chapter 1, the indoor aerial acoustic communication has numerous application examples, thanks to its versatility. Any types of audio interface that can play an audio file is utilized as signal beacons, and all smart devices equipped with an microphone and computing processor can work as a receiver. In this section, we present two presentative applications as follows.

TV content recognition. We assume that the audio of TV program contains chirp signal. On the reception of the chirp signal, the users watching TV with their smart device can be provided with additional information, e.g., the profile of an actor in the TV program. This can be realized through a second screen application empowered by our chirp signal receiver. Figure 4.12(a) illustrates such an example.

Indoor location tracking. In this example, we assume that the loud speakers installed in buildings and/or subways can play different source of audio, as shown in different patterns in Fig. 4.12(b).[3] Then each loud speaker works as a beacon that broadcasts location ID. An indoor navigation application combined with our chirp signal receiver can guide the users based on the signal reception.

We also discuss the technical issues that can arise during the actual implementation of the applications in practice. For the TV content recognition, the biggest challenge for the chirp signal is audio encoding of TV signals. Note that Dolby Digital (or AC3) and advanced audio coding (AAC), which are one of the most popular audio encoders

[3] This is reasonable as most PA systems support localized broadcasting. Remind that we can listen different announcements, for instance, in different floors of a building.



(a) TV content recognition.

(b) Indoor location tracking.

Figure 4.12: Application examples of chirp signal receiver.

for TV networks, are lossy audio coding algorithms [64, 65]. This can cause signal losses through the TV network. In case of indoor location tracking service, different signals from multiple sound sources might interfere with each other. Hence the chirp signal receiver needs to be able to differentiate the different signals in order to correctly trace the location. We discuss the these challenges in the followings, and describe possible solutions.

4.4.1 TV Content Recognition

The application requires co-operation with TV broadcast companies. That is, TV broadcast company mixes their TV contents and our chirp signal. The proposed chirp signal receiver is then combined with social commerce apps so that they can decode the acoustic signals embedded in TV contents. Social commerce app users who enabled the opt-in chirp signal receiver can pervasively receive the information of product placement (PPL) during and/or after watching TV. The click-through rate of this targeted advertisement is expected to be higher than randomly distributed one.

The advantage of this service for TV broadcast company is having an extra channel of advertisement market. Note that mixing our inaudible signal in TV contents require additional effort for the engineers. However, the newly obtained advertisement market can profit the TV broadcast companies by selling more advertisements, and this derives their participations. They can also precisely calculate the view rate of each TV program, assuming that the number of users using the social commerce application is large enough.

Power Saving Operation

One of the challenging issues on the realization of the service is battery consumption of smart devices. Continuously recording and processing audio signal for 24 hours is not feasible since it would consume large amount of energy. Instead, we implement scheduling functionality for chirp signal receiver so that apps can be scheduled to activate the receiver process based on the time table of TV programs. For instance, to check if a user actually watched an one-hour TV program, it would be enough to record and process audio signal every 10 minutes. Our experiment has shown that the receiver process consumes only 1% more battery when the processing interval is 10 minutes. This can drastically reduce the power consumption of the smart device, combined with the signal detection algorithm in Section 4.3

Effect of Lossy Encoding in TV Networks

Another technical problem of the chirp signal is encoding loss through TV networks. Going through the TV encoder, the acoustic signal embedded in audio content could be partially lost due to the lossy encoding. Typical TV audio encoders such as advanced audio coding (AAC) and Dolby digital (AC3) have different frequency responses depending on the bit-rate settings.

Figure 4.13 depicts the frequency response of 20∼20,000 Hz LFS signal with 48 kHz audio sampling rate encoded by AAC and AC3 encoders with different bit rate settings using FFmpeg [66]. We address that the bit rate of AC3 encoder rarely changes the frequency response of LFS signal, but the maximum playable frequency is limited to approximately 21 kHz. In case of AAC, the encoding bit rate significantly changes the frequency response. It is shown that the encoding bit rate greater than or equal to 68 kbps passes the high frequency signal over 22 kHz. Considering that the popular encoding rate of AAC in YouTube is 96 kbps, we assess that the chirp signal can pass through the encoder in most AAC encoders, and partially lost through AC3 encoders especially for high frequency components.

It is worth to state that the high frequency signal loss during the encoding process is similar to the loss of high frequency signal due to the speaker and microphone's frequency response. Note that our chirp signal is designed to overcome this frequency selectivity by linearly sweeping the signal band. The devices with frequency selective frequency response can play and record up to its capability; the signal can be demodulated in spite of the severe selectivity as we extensively evaluated the reception rate of various smart devices in Chapter 2. Hence, we expect that the chirp signal can be delivered through the TV network. Leaving the field test through the actual TV network as our future work, we present in-lab test results below.

In order to study the effect of lossy encodings on the signal reception probability, we play chirp signals using a smart TV. Three different types of audio files are used for the experiment; loss-less wave format, and AC3 (640 kbps) and AAC (180 kbps) for lossy codec. 12 smart devices are randomly placed in 5 m distance from the TV, where the direct path between TV and smart devices are partially blocked by partition screens.

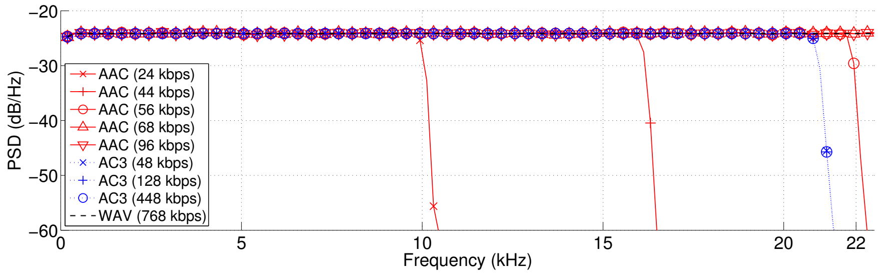


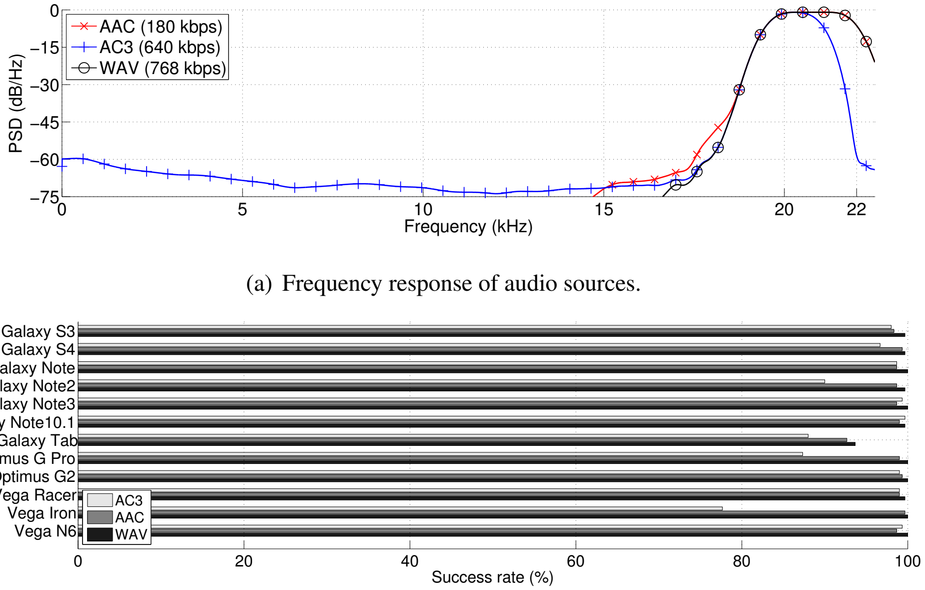
Figure 4.13: Frequency response of AAC and AC3 encoders.

Figure 4.14(a) presents the frequency response of AAC and AC3 encoded audio files. Note that AAC format with180 kbps encoding rate guarantees almost the same PSD with lossless WAV format. AC3 encoder, however, losses the high frequency components. Accordingly, the success rate degrades in Vega Iron, Optimus G Pro, and Galaxy Note2 models degrades for AC3 encodings due to the signal loss. As those models have small frequency gain in high frequency band, they sensitively reacts to the loss of high frequency signal compared to the others. Nonetheless, we claim that the encoded chirp signals can reach the smart devices through TV audio format with mostly acceptable probability of success.

4.4.2 Indoor Location Tracking in Seoul Subway

We have tested a prototype of the location alarm service for commuters in Seoul Metro. We develop an navigation application of Seoul Metro that is equipped with our chirp signal receiver. We assume that the audio source for the announcement of a specific subway station is mixed with our chirp signal that carries the station's ID information. The navigation application has a database of ID and subway station so that the application can acquire the current location by recording the announcement.

This application is specifically planned for those who have difficulty with hear



(a) Frequency response of audio sources.

(b) Success rate of signal reception.

Figure 4.14: Experiments on audio sources in AAC, AC3, and WAV audio formats.

the announcement due to their disabilities. Our navigation application transforms the acoustic announcement into visual message on the smart device. Note that this is very different from speech recognition. The speech recognition actually cannot work in subways due to the noisy environment – conversations from nearby customers severely interfere with the announcement. Our application is also helpful for ordinary people. We have witnessed that many peoples are too much absorbed in their smartphones in subway. They often miss their destination while texting message, playing games, or watching videos. This can be solved by using the location alarm functionality of the proposed application.

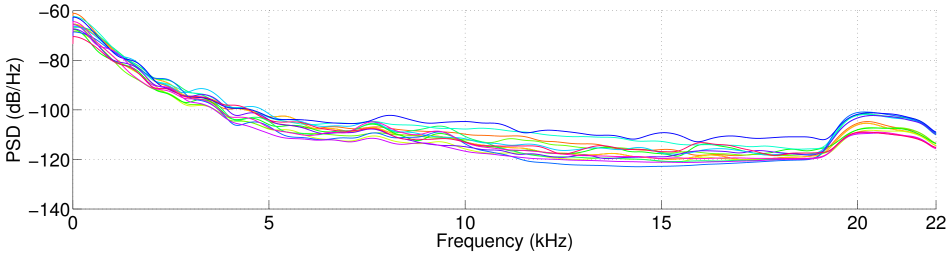


Figure 4.15: PSD of Chirp signal passed through the PA system of Seoul Metro.

Frequency Response of Subway PA Systems

As in lossy audio encoders in TV networks, PA systems in Seoul Metro has a problem in high-frequency gain. Many audio devices including audio players, amplifiers, and loud speakers in Seoul Metro are older than 15 years. These outdated audio interfaces might not be appropriate to play the chirp signal in high frequency band.

We have tested the PA system of Seoul Metro whether it can playback our chirp signal in 19.5∼22 kHz band. We have visited the repairing workshop of Seoul Metro and conducted our experiment in actual subway cars that are currently running. The audio source of the announcement of Seoul National University station is stereo-mixed with our chirp signal. Then it is played though the PA system, and we record the sound using Audix TM1 reference microphone in different location inside the subway car.

Figure 4.15 illustrates the PSD of 15 recorded sounds in the subway car. We observe that our chirp signal is delivered through the PA system of Seoul Metro. Hence we insist that the proposed application is feasible in practice, in spite of the old PA systems are designed for voice announcements.

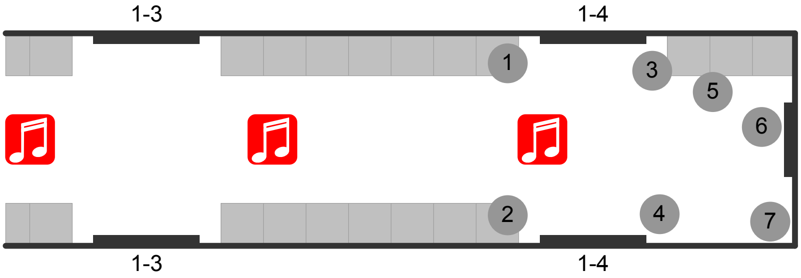


Figure 4.16: BER testing points in a subway car.

Table 4.2: Measured BER of Testing Points

Testing point	1	2	3	4	5	6	7
BER	0.0	0.0	0.0	0.0	3.1e-2	1.9e-2	1e-1

BER Performance in a Subway

The previous results are obtained using the reference microphone that has flat frequency response. The actual performance of smart devices might be different from the previous results due to their frequency selectivity. We conduct another experiment in the same environment, i.e., in the subway car of Seoul Metro. During the play of chirp signal, Galaxy Note3 device tries to demodulate the chirp signal in the testing points 1∼7 in Fig. 4.16. We hold the smartphone on hand in order to emulate realistic usage scenario.

Figure 4.16 illustrates a part of subway car of Seoul Metro, compartments 1-3 to 1-4. Each car has 6 loud speakers installed at the ceiling, as depicted by red squares in the figure. We measure the BER at the very end of the subway car, where is expected to have the worst BER due to the distance from loud speakers. Table 4.2 summarizes the measured BER out of 9,600 symbol receptions. The BER is worse than the experiments in Section 2.4.1, as there is no LOS between the speaker and microphone. As the microphone of smartphone is facing the floor, the smartphone received the signal reflected on the floor instead of direct path. Note that, however, most testing points achieve BER in the order of 10^{-2}, except the very corner case. Moreover, the receiver can overcome this high BER from repetition coding that was discussed in Chapter 3.

4.4.3 Device to Device Communication

We have presented the communication scenario that smart devices are used as receivers. That is, loud speakers and/or TVs play chirp signal, and smart devices catch it. It should be noted that, however, the smart device are also equipped with speakers. Therefore the designed chirp signal can link two smart devices using their speaker and microphones, i.e., enable device-to-device (D2D) acoustic communication. We here briefly present the experimental results of the proposed modem for device to device communication and corresponding applications.

Performance Evaluation

We select a pair of smart devices from Galaxy Note 10.1, Galaxy Tab, Optimus G Pro, and Galaxy Note 2, and set them in 5 m distance with LOS link. The volume of transmitter is set to 80% of its maximum. The success probabilities of signal reception out of 300 trials for all combinations of transmitter-receiver pair are shown in Table 4.3. We observe that the reception probabilities of all pairs are over 95% in spite of the severe frequency selectivity of their speakers as observed in Fig. 2.3. The achievable link distance might decrease in NLOS channels, but we observe the feasibility of chirp acoustic signals for D2D communication among smart devices.

Table 4.3: Signal Reception Probabilities of Various Transmitter-Receiver Pairs

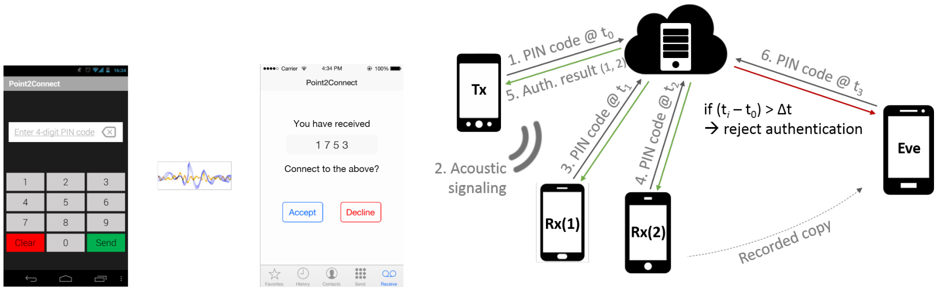
		Transmitter			
		Galaxy Note 10.1	Galaxy Tab	Optimus G Pro	Galaxy Note 2
Receiver	Galaxy Note 10.1	N/A	95%	99%	97%
	Galaxy Tab	99.7%	N/A	99.7%	98.7%
	Optimus G Pro	99.7%	97.3%	N/A	98%
	Galaxy Note 2	100%	96.7%	98.3%	N/A

D2D Acoustic Communication Applications

Accordingly, we claim that the proposed modem can be also used to implement proximity networking-based applications. Figure 4.17(a) presents an example of setting up such a D2D application for Android and iOS devices. A transmitter sends a short PIN number through chirp signal, and the receiver demodulates the chirp signal in order to receive the PIN number. A user then accepts or declines the D2D connection by verifying the PIN number. This method can be used to simultaneously set up multiple D2D connections among smart devices due to the broadcast nature of sound waves.

Note that there could be an eavesdropper that records the chirp signal and then reuses the recorded audio to mimic the D2D link setup. In order to prevent the unauthorized use of chirp signals, we can utilize our backend server for the authentication of D2D connections as shown in Fig. 4.17(b). At the beginning of a link setup (t_0), the transmitter first registers the PIN code to our backend server. The recipients then reports the reception of chirp signals to the backend server. If the time difference of the i-th report t_i and t_0 is longer than a certain threshold, the backend server rejects the authentication request (Eve in Fig. 4.17(b)) so that the chirp signal cannot be reused after wise.

The above mentioned D2D connection can enable the services in the following.



(a) Setting up a D2D connection.

(b) Authentication of D2D connections.

Figure 4.17: An exemplary application of D2D connection and backend server framework to authenticate the chirp signal and D2D connection.

WiFi connection assistance The most time-consuming process of connecting WiFi and/or WiFi Direct is searching the other devices in WiFi channels. In order to reduce the time for WiFi scanning, a transmitter can send the WiFi channel number and the last digit of its MAC address via chirp signal, which takes approximately 2.5 sec. Then a receiver quickly searches the specified channel and connects to the transmitter by receiving that information through a chirp signal.

Proximity network D2D link setup can also be used to authenticate a physical contacts of two (and more) smart devices. For instance, a user can authenticate that he or she has met a number of persons in exhibitions, conferences, and seminars. The authentication can also be used to set up a screen sharing among multiple smart devices to enlarge the screen.

4.5 Summary

In this chapter, we have designed auxiliary functionalities for the chirp receiver in order to compensate its limitations. Our backend server implemented in Amazon web service manages a database that maps the 16 bit data in a chirp signal and URL to be contacted for more informations. The backend server architecture can overcome the limited data rate of the proposed chirp receiver, as well as eases the maintenance of information in case of service and/or application updates.

Motivated by the battery consumption problem of recording process, we also proposed a framework for the low power operation of the chirp receiver. The chirp receiver basically waits in a stand-by state, and periodically activates the recording process on demand. The period of recording activation can be designed considering on the trade-off between power saving and reaction time limit, where the details depend on the objective of the application.

The key procedure for the low power operation framework above is its signal detection algorithm. The receiver can save the computing energy for signal processing when there is no received signal in the audio data. Considering the difference of acoustic communication compared with wireless communication, we propose a novel signal detection metric, peak PSD ratio. The signal detection algorithm checks the existence from a short record data by calculating pseudo SNR from its PSD. Our experimental results present that the proposed algorithm can achieve 0.3% miss detection and 0.1% false alarm probability in practice.

We have presented two applications of the chirp receiver, namely, TV content recognition and indoor location tracking. The service scenario and system implementation of the two applications are discussed in a technical point of view. The chirp signal mainly needs to cope with the frequency response of the application system, i.e., lossy audio codecs of TV network and legacy PA systems with out-dated device. In-lab experiments and field test results in realistic environments have shown the feasibility of the proposed applications, where we still need more research to successfully provide the application services as follows.

We are currently planning a number of field tests using actual TV network of broadcast companies. It should be noted that the TV cable network is composed of multiple tiers that repeats decodings and encodings carrying over the local service distributors. Throughout in-depth analysis on the encoding/decoding algorithms of TV audio codes as well as transcoding devices of TV network, we expect to optimize our signal design for better delivery. For example, we can pre-shape our chirp signal in reverse form of the TV transcoders' frequency response, so that the re-distributed TV signal can carry a frequency-flat chirp signal.

We are also investigating the method to mix the chirp signal and audio sources, by cooperating with a number of manufacturers of PA system equipments. Note that current PA systems in subways are automized to play the announcements when the subway car approaches each subway station. Our chirp signal hence needs to be mixed with the announcement audio files in off-line, and saved in the PA systems. During the stereo mixings of chirp signal and announcement audio, we have found that clippings over the representable and/or linear operating range of digital audio amplifiers generates audible noise due the distortion of chirp waveform. Hence we need careful control of mixing levels to make it inaudible.

The superposed chirp signals with different information from different audio sources also need to be resolved to further expand its applications. This is different from resolving multi-path propagation in Chapter 3 where all multi-path components carry the same information. Basically the demodulation of chirp signals will have the same procedure in superposed signal receptions. The main problem is separating the multipath components and superposed components. We are studying the differentiation of the preamble design so that the receiver can easily detach a target signal from mingled audio.

Chapter 5

Conclusion and Future Work

5.1 Research Contributions

In this dissertation, we propose chirp signal-based aerial acoustic communication technique for smart devices. Measurement results of indoor acoustic channel reveals severe multi-path propagation and device-dependent frequency selectivity. We adopt chirp signal for indoor acoustic communication to cope with the impairments of the indoor acoustic channel. The proposed chirp BOK signal is shown to achieve approximately 16 bps data rate up to 25 m distance with 97% success rate for most smart devices. We also present a receiver architecture that is designed for the efficient chirp signal reception of smart devices. We combine FFT and Hilbert transform in order to build an envelope detector so that the peaks of chirp auto correlation can be easily found. The chirp signal receiver is augmented by our backend server and signal detection algorithm in order to compensate for the low data rate and save unnecessary power consumptions. Based on the receiver and system design using chirp signal, we introduce two application examples, namely, TV content recognition and indoor location tracking, and discuss technical issues in their implementations. Accompanying field test and experiment results evaluate the feasibility of the applications in realistic communication environments.

More into detail, the research contributions of each chapter in the dissertation are summarized as follows.

In Chapter 2, we study the characteristics of indoor aerial acoustic channel. We define inaudible frequency band from 19.5 to 20 kHz through theoretic and experimental analysis. Measuring the impulse response of indoor acoustic channel shows 40 msec delay spread for -6 dB threshold in NLOS channel. The frequency response of speaker and microphones in various smart devices are turned to be severely selective especially in the inaudible band. We evaluate the performance of two representative PHY modulation schemes, PSK and FSK, in indoor aerial acoustic communication. Experimental results show that the instability of local oscillator and device-dependent frequency selectivity are the main reasons for the performance degradation of PSK and FSK, respectively. We adopt chirp signal that originally used for radar applications thanks to its good auto correlation characteristic. The chirp BOK modulation scheme is turned out to outperform PSK and FSK modulations in long-range indoor acoustic communication.

In Chapter 3, we design a receiver architecture for the demodulation of chirp BOK signal. The main goal of receiver design is its simplicity for the implementation in smart devices with limited computing power and battery. We substitute the convolution in matched filter to FFT-IFFT pairs so that the complexity can be reduce from $O\left(n^2\right)$to $O\left(n \log n\right)$. The FFT-based demodulator is also combined with Hilbert transform to realize an envelope detector, and the location of envelope peaks are adaptively detected based on SNR maximizing criterion. The proposed envelope detector effectively finds the peaks of auto correlation by removing the high-frequency component and searching the shifted peaks due to the Doppler effect. Additionally, we evaluate the capability of the chirp signal for differentiating multi-path components in indoor aerial acoustic channel. In practice, it is shown that the chirp signal can distinguish two multi-path propagations if they are separated more than 1.18 msec, which is equivalent to 0.14 m path difference.

In Chapter 4, we propose auxiliary features for the chirp signal receiver. The receiver refers to our backend server in order to map the the received signal into an URL that will be contacted for additional information. This backend server architecture can overcome the limited bit rate of the chirp receiver. A signal detection algorithm is essential for the receiver to save its operating power by removing unnecessary signal processing when the recorded audio does not contain any chirp signal. We introduce a novel detection metric, peak PSD ratio, that can be easily obtained from FFT of the proposed receiver architecture. Peak PSD ratio, which is a pseudo SNR of the high frequency signal, can classify the signal existence, where its false alarm and miss detection probabilities are shown to be less than 0.3%. We also present two application examples of the chirp signal, TV content recognition and indoor location tracking, and discuss the technical issues in their implementations. Our experiments testify that the chirp signal can be delivered over the TV network in spite of the signal losses due to the lossy audio codecs. Field test results in Seoul Metro also verifies that the chirp receiver works in the subway cars with old PA systems.

5.2 Future Work and Concluding Remark

The backend server in Chapter 4 compensates for the low bit rate of the chirp signal receiver, but it cannot work if the smart devices does not have an Internet connection. Hence improving the bit rate of chirp signals is an important future work for supporting various service scenarios. We plan to design an equalizer and Rake receiver that can cope with the multi-path propagation. The basic concept for the equalization and Rake receiver is similar for the chirp signal receiver; the multiple envelope peaks from multi-path components are gathered and co-processed. Their objectives, however, are different as the equalizer aims to remove the ISI among consecutive symbols while Rake receiver tries to combine the duplicate peaks within a symbol duration. The design of the equalizer and Rake receiver is promising as the chirp signal can differentiate multi-path components as seen in Chapter 3. It is expected to shorten and/or remove the guard intervals in the current frame structure and improve the data rate.

Reducing the battery consumption of the propose chirp signal receiver is also a challenging research, considering the limited power of smart devices. The reception process discussed in Chapter 3 is triggered by a local timer, and supported by the signal detection algorithm in Chapter 4. The triggering period of the local timer can be optimized based on the objective of a service. Note that there is a trade-off between the power saving and agile signal detection, depending on the triggering period. We are currently developing an adaptive wake-up algorithm for activating the chirp signal receiver, so that it automatically controls its triggering period in a context-aware manner. For example, the receiver might activate the microphone at the very beginning of a phone call and/or message display, as the basic power consumed by the smart device overwhelms that of the microphone. The adaptation algorithm is supposed to balance the power consumption and detection time simultaneously.

Finally we address the medium access control issue of the chirp signal. Chapter 4 propose an indoor location tracking application of the chirp signal. There should be multiple beacons with different chirp signals in a dense area in order to increase the resolution of location tracking. In such a case, a smart device is suppose to receive multiple chirp signals from different beacons simultaneously. Therefore the receiver needs to be able to distinguish different signals in such environments, and pick up the right signal that comes from the closest beacon. The major challenge in designing such a receiver is classifying multi-path components and different signal sources. We are investigating the usage of preamble for provisioning the signal resolution capability. For example, different beacons can have different types of preamble, so that the receiver can find distinct preambles in the recorded audio.

The dissertation has presented an aerial acoustic communication technique for smart devices in indoor environment. The research contribution covers from the signal and receiver design to its practical applications. We have demonstrated its performance and feasibility via theoretical analysis, extensive experiments, and field test in realistic environments. We expect that the findings and discussions in the dissertation can be a cornerstone for further research, including our future work, on the aerial acoustic communication for smart devices.

Bibliography

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: research challenges,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[2] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: propagation models and statistical characterization,” *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, 2009.

[3] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (Revision of IEEE Std. 802.11-2007)*, IEEE Std., 2012.

[4] IEEE 802.15.1, *Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*, IEEE Std., 2005.

[5] D. B. Kilfoyle and A. B. Baggeroer, “The state of the art in underwater acoustic telemetry,” *IEEE J. Oceanic Eng.*, vol. 25, no. 1, pp. 4–27, 2000.

[6] M. Stojanovic, *Acoustic (underwater) communications, Encyclopedia of Telecommunications*. Wiley Online Library, 2003.

[7] J. Catipovic, A. B. Baggeroer, K. von der Heydt, and D. Koelsch, “Design and performance analysis of a digital acoustic telemetry system for the short range underwater channel,” *IEEE J. Oceanic Eng.*, vol. 9, no. 4, pp. 242–252, 1984.

[8] A. Kaya and S. Yauchi, “An acoustic communication system for subsea robot,” in *Proc. IEEE OCEANS*, vol. 3, 1989, pp. 765–770.

[9] G. R. Mackelburg, “Acoustic data links for UUVs,” in *Proc. IEEE OCEANS*, 1991, pp. 1400–1406.

[10] G. Howe, O. Hinton, A. Adams, and A. Holt, “Acoustic burst transmission of high rate data through shallow underwater channels,” *IET Electron. Lett.*, vol. 28, no. 5, pp. 449–451, 1992.

[11] M. Suzuki, T. Sasaki, and T. Tsuchiya, “Digital acoustic image transmission system for deep-sea research submersible,” in *Proc. IEEE OCEANS*, vol. 2, 1992, pp. 567–570.

[12] M. Stojanovic, J. Catipovic, and J. G. Proakis, “Adaptive multichannel combining and equalization for underwater acoustic communications,” *The Journal of the Acoustical Society of America*, vol. 94, no. 3, pp. 1621–1631, 1993.

[13] K. F. Scussel, J. A. Rice, and S. Merriam, “A new MFSK acoustic modem for operation in adverse underwater channels,” in *Proc. IEEE OCEANS*, vol. 1, 1997, pp. 247–254.

[14] J. Jones, A. Di Meglio, L. Wang, R. Coates, A. Tedeschi, and R. Stoner, “The design and testing of a DSP, half-duplex, vertical, DPSK communication link,” in *Proc. IEEE OCEANS*, vol. 1, 1997, pp. 259–266.

[15] L. Freitag, M. Grund, S. Singh, S. Smith, R. Christenson, L. Marquis, and J. Catipovic, "A bidirectional coherent acoustic communication system for underwater vehicles," in *Proc. IEEE OCEANS*, vol. 1, 1998, pp. 482–486.

[16] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, "Shallow water acoustic networks," *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 114–119, 2001.

[17] C. V. Lopes and P. M. Aguiar, "Acoustic modems for ubiquitous computing," *IEEE Pervasive Comput.*, vol. 2, no. 3, pp. 62–71, 2003.

[18] A. Madhavapeddy, R. Sharp, D. Scott, and A. Tse, "Audio networking: the forgotten wireless technology," *IEEE Pervasive Comput.*, vol. 4, no. 3, pp. 55–60, 2005.

[19] A. Mandal, C. V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, and P. Baldi, "Beep: 3D indoor positioning using audible sound," in *Proc. IEEE CCNC*, 2005, pp. 348–353.

[20] C. V. Lopes, A. Haghighat, A. Mandal, T. Givargis, and P. Baldi, "Localization of off-the-shelf mobile devices using audible sound: architectures, protocols and performance assessment," *ACM SIGMOBILE MC2R*, vol. 10, no. 2, pp. 38–50, 2006.

[21] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of WiFi based localization for smartphones," in *Proc. ACM MobiCom*, 2012, pp. 305–316.

[22] R. Nandakumar, K. K. Chintalapudi, and V. Padmanabhan, "Centaur: locating devices in an office environment," in *Proc. ACM MobiCom*, 2012, pp. 281–292.

[23] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “BeepBeep: a high accuracy acoustic ranging system using cots mobile devices,” in *Proc. ACM SenSys*, 2007, pp. 1–14.

[24] V. Gerasimov and W. Bender, “Things that talk: using sound for device-to-device and device-to-human communication,” *IBM Systems Journal*, vol. 39, no. 3.4, pp. 530–546, 2000.

[25] C. V. Lopes and P. M. Aguiar, “Aerial acoustic communications,” in *Proc. IEEE WASPAA*, 2001, pp. 219–222.

[26] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, “Dhwani: secure peer-to-peer acoustic NFC,” in *Proc. ACM SIGCOMM*, 2013, pp. 63–74.

[27] H. Matsuoka, Y. Nakashima, and T. Yoshimura, “Acoustic communication system using mobile terminal microphones,” *NTT DoCoMo Tech. J*, vol. 8, no. 2, pp. 2–12, 2006.

[28] H. S. Yun, K. Cho, and N. S. Kim, “Acoustic data transmission based on modulated complex lapped transform,” *IEEE Signal Processing Lett.*, vol. 17, no. 1, pp. 67–70, 2010.

[29] S. Holm, O. B. Hovind, and S. Rostad, “Indoors data communications using airborne ultrasound,” in *Proc. IEEE ICASSP*, vol. 3, 2005, pp. iii–957.

[30] M. Hanspach and M. Goetz, “On covert acoustical mesh networks in air,” *J. of Commun.*, vol. 8, no. 11, 2013.

[31] SonicNotify. [Online]. Available: http://sonicnotify.com/

[32] ShopKick. [Online]. Available: https://www.shopkick.com

[33] Zoosh. [Online]. Available: http://www.verifone.com/industries/taxi/way2ride/

[34] C. E. Cook, “Linear FM signal formats for beacon and communication systems,” *IEEE Trans. Aerosp. and Electron. Syst. Mag.*, no. 4, pp. 471–478, 1974.

[35] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[36] S. T. Neely and J. B. Allen, “Invertibility of a room impulse response,” *The Journal of the Acoustical Society of America*, vol. 66, no. 1, pp. 165–169, 1979.

[37] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 2, pp. 145–152, 1988.

[38] B. C. J. Moore, *An introduction to the psychology of hearing*, 5th ed. Academic Press San Diego, 2003.

[39] B. C. Moore and B. R. Glasberg, “A revision of Zwicker’s loudness model,” *Acta Acustica united with Acustica*, vol. 82, no. 2, pp. 335–345, 1996.

[40] B. C. J. Moore, B. R. Glasberg, and T. Baer, “A model for the prediction of thresholds, loudness, and partial loudness,” *Journal of the Audio Engineering Society*, vol. 45, no. 4, pp. 224–240, 1997.

[41] ISO 226:2003 *Acoustics – normal equal-loudness-leevl contours*, ISO, 2003.

[42] ISO 226:1997 *Acoustics – normal equal-loudness-leevl contours*, ISO, 1997.

[43] E. Terhardt, “Calculating virtual pitch,” *Hearing research*, vol. 1, no. 2, pp. 155–182, 1979.

[44] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proc. Audio Engineering Society Convention*, 2000.

[45] ——, "Advancements in impulse response measurements by sine sweeps," in *Proc. Audio Engineering Society Convention*, 2007.

[46] "Genelec 6010A – Bi-amplified loudspeaker system." [Online]. Available: http://www.genelec.com/products/previous-models/6010a/

[47] "Audix TM1 – Test and measurement microphone." [Online]. Available: http://www.audixusa.com/docs_12/units/TM1.shtml

[48] G. Turin, "Error probabilities for binary symmetric ideal reception through non-selective slow fading and noise," *Proc. of the IRE*, vol. 46, no. 9, pp. 1603–1619, 1958.

[49] A. J. Berni and W. Gregg, "On the utility of chirp modulation for digital signaling," *IEEE Trans. Commun.*, vol. 21, no. 6, pp. 748–751, 1973.

[50] J. G. Proakis, M. Salehi, N. Zhou, and X. Li, *Communication systems engineering*. Prentice-hall Englewood Cliffs, 1994, vol. 2.

[51] A. Viterbi, "Optimum detection and signal selection for partially coherent binary communication," *IEEE Trans. Inform. Theory*, vol. 11, no. 2, pp. 239–246, 1965.

[52] C. E. Cook and M. Bernfeld, *Radar signals: an introduction to theory and application*. Academic House, 1967.

[53] IEEE 802.15.4a, *Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Chirp Spread Spectrum (CSS) PHY Layer*, IEEE Std., 2007.

[54] M. R. Winkley, "Chirp signals for communications," in *WESCON Convention Record*, vol. 14, no. 2, 1962.

[55] S. E. El-Khamy, S. E. Shaaban, and E. Tabet, “Efficient multiple-access communications using multi-user chirp modulation signals,” in *Proc. IEEE ISSSTA*, vol. 3, 1996, pp. 1209–1213.

[56] “Kiss FFT — a fast fourier transform library.” [Online]. Available: http://sourceforge.net/projects/kissfft/

[57] R. N. Bracewell and R. Bracewell, *The Fourier transform and its applications*. McGraw-Hill New York, 1986.

[58] E. Bedrosian, “A product theorem for Hilbert transforms,” *Proc. IEEE*, vol. 51, no. 5, pp. 868–869, 1963.

[59] W. Hirt and S. Pasupathy, “Continuous phase chirp (CPC) signals for binary data communication-Part I: Coherent detection,” *IEEE Trans. Commun.*, vol. 29, no. 6, pp. 836–847, 1981.

[60] “Amazon web services.” [Online]. Available: http://aws.amazon.com/

[61] “JSON — JavaScript Object Notation.” [Online]. Available: http://json.org/

[62] J. A. Hanley, “Characteristic (ROC) Curvel,” *Radiology*, vol. 743, pp. 29–36, 1982.

[63] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, “Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach,” *JSTOR Biometrics*, pp. 837–845, 1988.

[64] “Dolby Digital.” [Online]. Available: http://www.dolby.com/

[65] ISO/IEC 13818-7 *MPEG-2 Part 7: Advanced audio coding (AAC)*, ISO/IEC, 2007.

[66] “FFmpeg.” [Online]. Available: http://www.ffmpeg.org

초록

오늘날 스마트폰 및 웨어러플 컴퓨터와 같은 스마트 기기(smart device)들은 일반적으로 사용자의 직관적인 기기 이용을 위한 음성 사용자 인터페이스(voice user interface)를 지원한다. 본래 음성 사용자 인터페이스는 사용자의 목소리, 주변의 음악 및 배경잡음과 같은 소리를 녹음하거나, 스마트 기기의 상태 혹은 데이터를 사용자에게 도시하기 위한 용도로 사용되어 왔다. 최근 새롭게 각광받고 있는 *음파 통신 기술*은 이러한 음성 사용자 인터페이스를 통신용 인터페이스로 활용하여, 무선랜(WiFi)이나 블루투스(Bluetooth)를 이용하지 않고도 데이터를 전달할 수 있다는 점에서 다양하게 연구되고 있다. 특히 스마트 기기들은 음향 데이터를 재생/녹음하고, 중앙처리장치(application processor, AP)를 통해 신호처리를 자유롭게 수행할 수 있기 때문에, 음파 통신을 위한 소트프웨어 모뎀(software modem, software defined radio) 개발 플랫폼으로 유용하다. 또한 음파 통신 기술은 스피커와 마이크가 장착되어 있는 비통신 장비(예: TV, 오디오 장비 등)에 통신 기능을 부여할 수 있으므로 그 활용도가 매우 광범위하다. 본 학위논문은 실내 환경에서 스마트 기기들을 위한 장거리(long-range) 음파 통신 기법에 대한 연구 결과를 다루고, 그 응용 서비스들의 구현 및 분석 결과를 논의한다.

먼저 실내 환경에서 적합한 음파 통신의 물리계층(physical layer) 신호 설계를 위해, 제 2장에서 스피커, 공기, 마이크로 구성된 음향 채널에 대한 측정을 수행하고 그 결과를 검토한다. 측정 결과의 분석을 통해 실내 음향 채널은 약 40 msec에 달하는 긴 지연확산(delay spread)을 가지며, 스피커와 마이크의 주파수 응답

(frequency response)특성으로 인해 매우 주파수 선별적(frequency selective)이라는 사실을 알 수 있다. 이러한 실내 음파 통신 환경에서의 실험결과를 바탕으로 위상 변조(phase shift keying)이나 주파수 변조(frequency shift keying)과 같은 기존의 물리계층 신호 변조(modulation) 기법들은 스마트 기기의 내부 진동자(local oscillator)의 불안정성 및 음향 채널의 주파수 선별성으로 인해 크게 성능 열화를 보임을 확인한다.

이와 같은 실내 음향 채널의 문제점을 극복하기 위한 수단으로 제 3장에서는 처프 신호(chirp signal)을 음파 통신에 도입한다. 처프 신호를 이용해 설계된 음파 신호와 그 수신기(receiver)가 주파수 선별적이고 긴 지연확산을 가지는 장거리 실내 음향 채널에서 효율적으로 동작할 수 있음을 다양한 실험 결과와 함께 증명한다. 또한 스마트 기기에 처프 신호 수신기를 구현하기 위해, 본 학위논문은 매우 간단하면서도 효율적인 처프 신호 수신기 구조(receiver architecture)를 선보인다. 고속 푸리에 변환(fast Fourier transform) 및 힐버트 변환(Hilbert transform)을 응용하여 설계된 처프 신호 수신기는, 도플러 효과(Doppler effect)에 의하여 천이(transit)하는 처프 신호 자기상관(auto correlation) 함수의 포락선(envelope)을 효과적으로 탐지할 수 있다. 이와 같이 구현된 처프 신호 및 그 수신기는 실내 25 m 거리의 음파 통신 환경에서 약 16 bps의 전송률을 달성한다. 본래 처프 신호는 그 자기상관 특성이 우수하여 레이더(radar)에 주로 사용되어 왔다. 그러나 스마트 기기를 위한 공기중 음파 통신(aerial acoustic communication)에 처프 신호를 도입하고, 이에 최적화된 수신기 설계를 소개한다는 점에서 본 학위논문의 의의를 찾을 수 있다.

마지막으로 제 4장은 처프 신호와 수신기를 적용하는 응용 서비스의 예를 소개하고, 실제 서비스의 구현에 수반되는 기술적인 논의들을 다룬다. 처프 신호 및 수신기의 초당 16 비트 전송률은 십수미터 내외의 근거리에서 짧은 아이디(ID)를 전송하는 용도로 충분할 수 있다. 그러나 그보다 높은 전송률을 요구하는 응용 서비스에는 적합하지 않기 때문에, 이를 해결하기 위해 백엔드(backend)

서버를 설계한다. 16 비트 아이디를 통해 정보를 직접 전달하는 대신, 스마트 기기가 아이디를 백엔드 서버에 질의(query)하고, 그에 상응하는 정보를 확인할 수 있는 인터넷 주소(URL)를 회신(response)받도록 함으로써, 처프 신호 수신기의 부족한 전송률을 보완할 수 있다. 또한 처프 신호가 없는 상황에서 스마트 기기가 무의미한 신호처리에 낭비하는 전력소모를 최소화 하기 위한 신호 감지(signal detection) 방안을 설계하고, 다양한 환경에서의 실험을 통해 허위 경보(false alarm) 및 검출 실패(miss detection) 확률이 0.3% 이내로 제한됨을 보인다. 마지막으로 처프 신호와 수신기를 이용한 응용 서비스로 (1) TV 컨텐츠 인식(content recognition)과 (2) 실내 위치 파악(indoor location tracking)을 소개하고, 실환경(realistic environment)에서의 실험 및 현장검증(field test)을 통해 해당 서비스들의 구현 가능성을 검증한다.