Exploiting Two-user Superimposed Signals for Wireless Communication Systems

by

Wen Cui

B. Eng., Xi'an University of Science and Technology, 2013

M. Eng., Northwest University, 2016

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Exploiting Two-user Superimposed Signals for Wireless Communication Systems

by

Wen Cui

B. Eng., Xi'an University of Science and Technology, 2013

M. Eng., Northwest University, 2016

Supervisory Committee

Dr. Lin Cai, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Riham Altawy, Departmental Member

(Department of Electrical and Computer Engineering)

Dr. Daniela Constantinescu, Outside Member

(Department of Mechanical Engineering)

Supervisory Committee

Dr. Lin Cai, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Riham Altawy, Departmental Member

(Department of Electrical and Computer Engineering)

Dr. Daniela Constantinescu, Outside Member

(Department of Mechanical Engineering)

ABSTRACT

Wireless communication systems are growing at an unprecedented pace, making the wireless spectrum at a premium, especially as billions of new Internet-of-Things (IoT) devices worldwide are demanding wireless connections. To accommodate the ever-growing spectrum demand, a promising solution is Non-Orthogonal Multiple Access (NOMA) that enables two users to communicate with the same spectrum resource at the same time, while decoding the two-user superimposed signal at the receiver. By doing this, the previously detrimental wireless interference caused by two concurrent transmitters becomes decodable at the receiver, potential for higher utilization of the wireless spectrum. Existing NOMA technologies, however, rely on strict power control to sequentially decode the two-user superimposed signal, which

is infeasible for many IoT devices that are heterogeneous and often low-cost. In contrast, in this dissertation, we propose new NOMA schemes that are designed for wireless communication systems and can decode the two-user superimposed signals without power control.

This dissertation makes four major contributions. First, it presents the first design to implement dynamic signal offsets tracking and reacting schemes to detect and decode two-user superimposed signals, robust against hardware imperfections and feasible for heterogeneous IoT devices. Second, by investigating the relationship between the channel condition and the bit-error-rate (BER) in decoding superimposed signals, we design a reliable NOMA scheme to combat dynamic channel conditions that are inevitable in many practical scenarios and may cause severe decoding errors. Third, considering the wireless communication systems in mobile scenarios, mobility is a vital feature of many applications but can cause severe signal variations and make the hardware offsets harder to predict, resulting in an unreliable decoding performance. To address this, we develop a diversity transmission and smart combining scheme to achieve high reliable decoding performance. Finally, we combine rotation coding to transmit and decode the superimposed signal to achieve both high spectrum efficiency and high reliability performance.

To demonstrate our contributions, we derive the theoretical relationship of the BER under different practical settings, validate the performance with simulations, and conduct experiments using software-defined radio based platforms with static indoor, outdoor scenarios and mobile scenarios. The experimental results demonstrate that, compared with the state-of-the-art methods, our schemes can achieve higher reliability and spectrum efficiency in decoding the superimposed signal for wireless communication systems without power control.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank **my supervisor Dr. Lin Cai** with my deepest gratitude for her offering the great opportunity to me to explore the beautify of science, and guiding me with her valuable suggestions. It is a great privilege for me to work with her who has led an amazing research group with her knowledge and enthusiasm. I already forget how many times I knocked on her office door for advice, but I do remember that every time when I got back to my seat, there would be a more clear picture in my head to lead me to go further and deeper. I sincerely thank her for these memorable four years, for the knowledge I gained, and for the courage I obtained to open the next chapter of my career.

Thanks to **Dr. Riham AlTawy, Dr. Daniela Constantinescu, and Dr. Hai Jiang** for serving as my committee members. They have spent their valuable time for my candidacy exam and my thesis with their constructive suggestions. I am extremely grateful for these suggestions which have improved my thesis substantially. Thanks to **Dr. Chen Liu** for her earnest advice in every research work that I have done in pursuing my Ph.D. I am thankful for her companion in catching every deadline of the paper submission with the best support from her side. We have gone through minor implementation details and critical questions from reviewers, and all of these are the real foundations of this thesis.

Thanks to **Dr. Jianping Pan and all the group members** for raising kind suggestions and comments in every group meeting and discussion. Their critical thinking has helped me to overcome the difficulties at each step. The time we have spent together is the most unforgettable part of my study. Thanks to **Dr. Hamed Mosavat**. It is an honor for me to have a friend like him.

Thanks to **My family** for their endless love.

Wen Cui, Victoria, BC, Canada

DEDICATION

To my family for their unconditional love, care and belief in me.

Chapter 1

Introduction

1.1 Background

Wireless systems are experiencing explosive growth in these years, especially with the demand from the Internet-of-Things (IoT). It is estimated that around 15 billions of IoT devices will be deployed [1] and the global market value will reach $1.2 trillion in 2022 [2]. The massive number of wireless connections, however, makes the wireless spectrum at a premium, and thus improving the spectrum efficiency becomes inevitable [3, 4]. Traditionally, dedicated wireless resources, in the time/frequency/space/code domains, are allocated orthogonally to each wireless user by using Carrier Sense Multiple Access (CSMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), etc. However, this orthogonality makes the spectrum not fully utilized; moreover, it can lead to severe spectrum competitions [5, 6].

Past theoretical work showed that the superposition coding reaches the capacity limit of two-transmitter Gaussian broadcast channel [7], and [8] proved that superposition coding can achieve a higher-rate region than orthogonal schemes. When

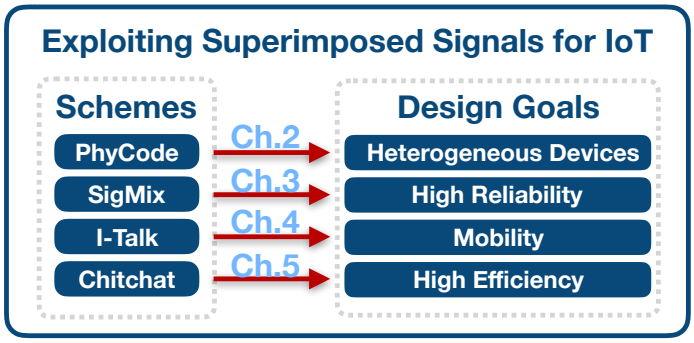


Figure 1.1: An overview of this dissertation.

we investigate the broadcast channel in a reciprocity manner, for two concurrent users and if designed appropriately, can also achieve higher-rate region than orthogonal schemes, leading to the increasing interests in Non-Orthogonal Multiple Access (**NOMA**). With NOMA, two users transmit signals through the same spectrum resource at the same time, and then the receiver decodes the two-user superimposed signals [9–12]. By doing so, the spectrum can be shared by two concurrent users and the efficiency can be improved. Existing NOMA schemes use a technique known as Successive Interference Cancellation (SIC) to decode the superimposed signal; however, SIC relies on strict power control on the transmitted signal [13–16], making it infeasible for heterogeneous and low-cost IoT devices. A question is raised naturally: can we enable decoding the two-user superimposed signals for NOMA without power control, so that it can be applicable to a wide range of applications such as IoT? Driven by this question, the main goal of this dissertation is to design NOMA schemes without power control. More than that, we design NOMA scheme to address four essential and practical issues for IoT—feasible to heterogeneous devices [17], high reliability [18], robust to mobility [19], and high efficiency [20], making the proposed NOMA solutions feasible and desirable to many IoT scenarios. We draw the overview of this dissertation in Fig. 1.1 and detail the design in the following sections.

1.2 Research Objectives and Contributions

1.2.1 Decoding Two-user Superimposed Signals for Heterogeneous Devices

In traditional NOMA schemes, two-user superimposed signals are decoded in signal-level which requires a tight power control. In this dissertation, instead of decoding the two-user superimposed signal in the signal-level, we were inspired by recent works that decode the two-user superimposed signal in the modulation-level, i.e., Physical-layer Network Coding (PNC) [21, 22]. In particular, PNC decodes the two-user superimposed signal in the constellation map of the receiver, so it requires no separation of the superimposed signal in the power domain, making it desirable for IoT applications. However, to implement PNC in practical scenarios, tight synchronizations are needed to avoid signal offsets, including the symbol-level time synchronization, the carrier-frequency synchronization, and the carrier-phase synchronization. The symbol-level time synchronization can be readily obtained by using a shared central clock, e.g., GPS clocks. For the carrier-frequency synchronization and the carrier-phase synchronization, the previous PNC schemes assumed that the devices are homogeneous and operating in static environments, so they can use an average signal offset to compensate the offsets from different transmitters [22, 23]. However, since many IoT devices are heterogeneous and operating in dynamic environments, the signal offsets may not be compensated well by directly using an average signal offset and hence result in serious decoding errors.

In Chapter 2, we present PhyCode, a new approach to enable the two-user superimposed signal detection and decoding in practical systems for IoT. We introduce the first design to implement dynamic signal offsets tracking and reacting schemes to detect and decode superimposed signals, making our system feasible for heterogeneous

IoT devices. To the best of our knowledge, the work reported in Chapter 2 is the first design to implement dynamic offset tracking and reacting schemes to detect and decode two-user superimposed signals, a critical step to achieve a lower bit-error-rate (BER) in practical systems.

1.2.2 Reliability Analysis and Diversity Transmissions for Decoding Two-user Superimposed Signals

While decoding the two-user superimposed signal in the modulation-level is desirable for IoT as it requires no power control, the intrinsic characteristics of superimposed signals in the constellation map—indistinguishable constellation points—can cause high decoding errors. For IoT systems under dynamic channel conditions and with hardware imperfections, the situation becomes worse and eventually leads to a significant decrease in reliability.

To deal with this challenge, in Chapter 3, we develop SigMix which takes the following steps. First, we derive an exact BER expression to model the relationship between the channel conditions and the decoding error probability. This BER expression provides us an important guideline in manipulating the channel. Next, we propose a shifting code that enables SigMix to truly avoid the indistinguishable constellation points by transmitting two copies of the signal to achieve a substantial diversity gain. Based on that, we propose an adaptive decoding scheme by considering both the dynamic channel conditions and the hardware imperfections. Finally, we build a software-defined radio based platform to evaluate the performance of SigMix across various scenarios. The extensive experimental results illustrate that SigMix obtains a one-order lower median BER than the state-of-the-art.

1.2.3 Decoding Two-user Superimposed Signal in Mobile Environments

Considering the IoT systems in mobile environments, mobility is an inseparable component of many IoT applications [24–27]. Along with the ever-growing opportunity brought by the mobility, the dynamic channel conditions from mobile devices are inevitable and it can cause severe signal variations upon the received two-user superimposed signals. Moreover, the hardware imperfection becomes more unpredictable under mobile scenarios. These challenges by mobility all contribute to an unreliable decoding performance. Therefore, a reliable superimposed signal decoding scheme for mobile IoT is highly desirable.

In Chapter 4, we design I-Talk, a new NOMA scheme to achieve high reliability in the presence of hardware imperfections and mobile channel conditions. We first study the signal offsets caused by the mobility and the hardware imperfection, and design a synthesis channel coefficient to represent all these offsets. By doing so, we can trace all the offsets and then eliminate the side effects of these offsets, providing a stable synchronization performance. Second, by exploiting the complementary property of the constellation points and the subcarriers' differences in mobile channel environments, as well as the substantial diversity gain of transmitting two copies of the signal, we propose a diversity transmission and smart combining scheme to achieve high reliable decoding performance. We implement I-Talk on a software-defined radio platform and evaluate its performance across various scenarios. Our extensive experimental results demonstrate that I-Talk achieves a one-order lower bit-error-rate and a 1.47× higher throughput gain in the mobile scenario compared to PhyCode.

1.2.4 Efficient and Reliable Two-user Superimposed Signal Decoding

The modulation-level approaches, such as SigMix [18], I-Talk [19] in Chapter 3 and Chapter 4, and NCMA [28–30], can decode the two-user superimposed signals from the modulation domain without power control, which is desirable for IoT scenarios. But to maintain a high reliability, they introduce repetitive symbols or signal copies, leading to a trade-off between higher efficiency and reliability.

In Chapter 5, we design ChitChat, an effective superimposed-encoding/decoding system to substantially improve the spectrum efficiency and achieve reliable performance. In ChitChat, rather than requiring adding repetitive symbols or signal copies, we enable decoding the two-user superimposed signals from the original symbols directly. In particular, we leverage a rotation code in encoding the transmitted signal, and then analyze the relationship between the decoding performance and the channel conditions. Based on that, we propose a weighted rotation code (WRC) for further manipulating the encoding scheme so that ChitChat can avoid indistinguishable constellation points at the receiver, which offers a reliable performance. We implement ChitChat on a software-defined radio platform and the extensive experimental results demonstrate that ChitChat outperforms previous works.

1.3 Dissertation Organization

The remaining parts of this dissertation are organized as follows.

In Chapter 2, we first explore the feasibility of two-user decoding superimposed signals without power control, and then we propose PhyCode that includes dynamic signal offsets tracking and reacting schemes to detect and decode two-user superimposed signals for heterogeneous IoT devices.

In Chapter 3, we study the relationship between the channel conditions and the decoding error probability. Based on that, we introduce SigMix which includes a diversity encoding and adaptive decoding scheme by considering both the dynamic channel conditions and the hardware imperfections, and therefore, we can decode the two-user superimposed signal with high reliability.

In Chapter 4, we investigate the challenges of decoding two-user superimposed signals in mobile scenarios. According to our new findings, we propose I-Talk, which includes a new diversity transmission scheme and a smart combining scheme to address the mobility challenges.

In Chapter 5, to decode two-user superimposed signals while preserving both spectrum efficiency and reliability, we propose ChitChat, a reliable NOMA scheme that includes a weighted rotation code in decoding the superimposed signal without introducing any repetitive signal symbols.

Chapter 6 concludes this dissertation and lays out the promising research directions as future works.

1.4 Bibliographic Notes

The work in Chapter 2 has been published in [17], and the work in Chapter 3 has been published in [18]. The work in Chapter 4 has been submitted as [19] and the work in Chapter 5 has been submitted as [20].

1.5 Abbreviation

The abbreviations in the thesis have been summarized in the table below:

Table 1.1: Abbreviations

Abbreviation	**Full Name**
IoT	Internet-of-Things
NOMA	Non-Orthogonal Multiple Access
SIC	Successive Interference Cancellation
PNC	Physical-layer Network Coding
CSMA	Carrier Sense Multiple Access
TDMA	Time Division Multiple Access
FDMA	Frequency Division Multiple Access
BER	Bit-Error-Rate
GPS	Global Positioning System
WRC	Weighted Rotation Code
CFO	Carrier Frequency Offset
SFO	Sampling Frequency Offset
STO	Symbol Timing Offset
NLOS	Non-Line-Of-Sight
LOS	Line-Of-Sight
OFDM	Orthogonal Frequency-Division Multiplexing
RFID	Radio-Frequency IDentification
LTE	Long-Term Evolution
CP	Cyclic Prefix
FFT	Fast Fourier Transform
ISI	Inter-Symbol Interference
STS	Short Training Sequence
LTS	Long Training Sequence
SNR	Signal-to-Noise Ratio
USRP	Universal Software Radio Peripheral
EVM	Error Vector Magnitude
ANC	Analog Network Coding
CSI	Channel State Information
MRC	Maximum Ratio Combining
CRC	Cyclic Redundancy Check
3GPP	3rd Generation Partnership Project
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
QAM	Quadrature Amplitude Modulation
GPSDO	GPS-Disciplined Oscillator

Chapter 2

PhyCode: Decoding Two-user Superimposed Signals for Heterogeneous Devices

2.1 Introduction

Wireless spectrum shortage is escalating with the growth of wireless applications. To solve this problem, a promising enabling technology is to allow wireless transmissions to overlap in the time/frequency/spatial domains, and decode the two-user superimposed signals[1] to significantly increase the throughput of the wireless networks [21, 31–34]. Although the theory underlying the superimposed signal has been around for several decades [35], until recently, we have seen compelling advances in moving the superimposed signal from theory to practice [22, 31]. Multiple systems have been implemented aiming to approach the theoretical throughput upper bound of wireless networks [32, 36].

[1] In this dissertation, the superimposed signal refers to a signal that is generated from two signal sources and mixed in the same channel purposely.

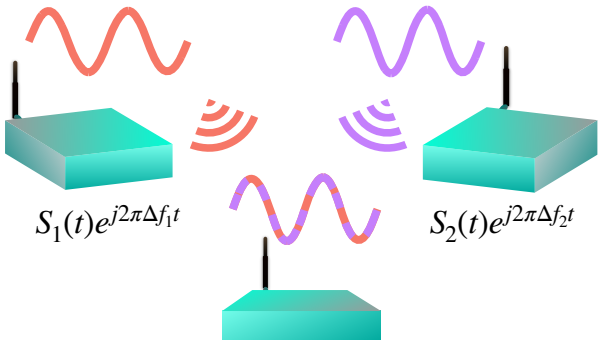


Figure 2.1: A superimposed signal contains multiple offsets. No single Δf can be applied to $S(t)$ to compensate Δf_1 and Δf_2 simultaneously.

There are two common approaches for decoding the two-user superimposed signal. One is Successive Interference Cancellation (SIC) [9], which typically requires power control to guarantee that one signal has much higher power than the other. Another approach is Physical-layer Network Coding (PNC) [22] which is more promising for IoT devices since it does not require power control. However, PNC requires the symbol-level time synchronization, the carrier-frequency synchronization, and the carrier-phase synchronization. Specifically, as for the symbol-level time synchronization, existing systems required transmitters sharing a central clock (e.g., GPS clocks) to align their signals at the receiver [22, 23], which is feasible for many IoT devices.

But for the carrier-frequency synchronization and the carrier-phase synchronization, they assumed homogeneous devices and static environments, where an average synchronization offset can be compensated to different signal sources. However, such an assumption may not hold in IoT scenarios, where heterogeneous devices operating in highly dynamic environments. In particular, first, the heterogeneous devices may vary in terms of vendors and types in a practical system. For each device, when it converts signals from the carrier to the baseband, the oscillator incorporated in that device inherently brings in a Carrier Frequency Offset (CFO) [37]. Because this offset can cause decoding errors, traditionally, compensation is needed to eliminate

this offset. However, since oscillators are different from each other in terms of the crystal vibration characteristic[2], when it comes to a superimposed signal in which signals with different CFOs are intertwined with each other, applying one compensation cannot eliminate different CFOs simultaneously (as shown in Fig. 2.1), which eventually cause severer decoding errors. Additionally, different oscillators will also cause a Sampling Frequency Offset (SFO) when the receiver samples a signal, resulting in a remarkable phase shift to the signal. Therefore, the existing solutions are vulnerable in real applications with heterogeneous devices. Second, IoT systems are typically deployed in dynamic environments with moving objects around, e.g., moving humans, animals, or vehicles, and will naturally span large geographic areas full of Non-Line-Of-Sight (NLOS) scenarios. Under this condition, the theoretical detection accuracy of the superimposed signal cannot be maintained well, causing a Symbol Timing Offset (STO). Therefore, the existing solutions are lack of robustness in dynamic environments.

In this chapter, we present PhyCode, a new approach to address the above limitations, aiming to enable the two-user superimposed signal detection and decoding in practical systems. First, PhyCode employs a two-step CFO correction scheme, where the CFO is calibrated in a coarse-grained manner at the transmitters and then the residual offset can be further corrected through a dynamic decoding scheme. By doing so, PhyCode can react to the exact offsets from different signal sources simultaneously. Second, unlike the existing works which ignored the difference between oscillators in the sampling process, PhyCode compensates SFO caused by different oscillators, which makes the design ubiquitous to the device heterogeneity. Third, PhyCode employs a two-step correlation method for the superimposed signal detection, by which the computation cost can be reduced significantly and the corre-

[2]The different transmitters may have up to hundreds of kHz offset in the frequency domain caused by the oscillators. Even oscillators from the same producer can be different [38].

sponding detection accuracy is improved under a practical NLOS scenario. Fourth, PhyCode corrects STO that is caused by time synchronization error in a dynamic environment. Finally, PhyCode exploits the nature of varying offsets, and designs a dynamic decoding scheme. Thus, PhyCode is robust against dynamic environments.

To the best of our knowledge, the work reported in this chapter is the first design to implement dynamic offset tracking and reacting schemes to detect and decode two-user superimposed signals, and thus to achieve a lower BER in practical systems. PhyCode focuses on OFDM, implementable for IEEE 802.11 a/g/n/p systems, LTE, etc. PhyCode can be a key enabling technology for many promising wireless technologies requiring the decoding of two-user superimposed signals, such as Non-Orthogonal Multiple Access (NOMA).

2.2 Related Work

Prior work falls into the following three categories.

(a) Successive Interference Cancellation in NOMA: Interference cancellation schemes typically require power control to guarantee that one signal has much higher power than the others. In this case, they can decode one first, and then cancel it out and decode the others. Such a design has been proposed for cellular systems [9]. Unlike interference cancellation schemes, PhyCode is a new type of NOMA which does not need power control on interfered signals, as controlling power tightly is hard for lightweight and ubiquitous IoT devices in most of the cases [39].

(b) Physical-layer Network Coding (PNC): PNC [22] can operate on two-user superimposed signals without power control. PhyCode can also be used for the PNC implementation in wireless communication systems. Theoretically, PNC requires the symbol-level time synchronization, the carrier-frequency synchronization,

and the carrier-phase synchronization. The existing PNC implementation [29] used a GPS clock to align the transmitted signals at the receiver to achieve the symbol-level synchronization in the time domain (more details are discussed in Section 2.3). Therefore, the main challenges are to develop solutions to address the carrier-frequency synchronization and the carrier-phase synchronization issues. Although several approaches can achieve a tight synchronization in the frequency domain, the phase domain [40] and the time domain [41, 42], they require either sophisticated hardware or a strict scheduler, which will introduce onerous burden for implementing PNC, especially in IoT systems. To solve this problem, the prior work [22,23,29] compensated an average carrier-frequency offset to different signal sources under the assumption of homogeneous devices and static environments. However, such an assumption may not hold, which limits the two-user superimposed signal detection and decoding in ubiquitous and robust IoT systems. Different from the existing solutions, PhyCode takes the heterogeneous devices and dynamic environments into consideration; therefore, PhyCode can dynamically react to the exact offsets from different signal sources simultaneously.

Some other related work falls in the area of network coding in the physical layer. One is analog network coding [31]. Although this solution does not require time synchronization, it needs a relay node to amplifies the two-user superimposed signal before decoding. Moreover, it amplifies not only the wanted signal but also environmental noises, which causes error propagation. Another is BiPass [32], which has also been investigated to improve the throughput by decoding the superimposed signal; however, it requires dedicated full-duplex devices, and more importantly, it still suffers from the noise propagation issue.

(c) Superimposed Signal in Other Techniques: Decoding superimposed signal has also been widely adopted in other popular wireless techniques, such as

RFID [43–45], Zigbee [36] and LoRa [46,47]. However, these techniques were designed to transmit at a low data rate, which limits wireless spectrum efficiency. In contrast to these works, PhyCode focuses on OFDM, implementable for IEEE 802.11 a/g/n/p, LTE, etc., which enables higher spectrum efficiency.

2.3 Preliminary

In this section, we present some background information and discuss the time synchronization requirement for decoding the two-user superimposed signal. For simplicity, we consider two sources transmitting signals simultaneously. Let Y be the received signal, X_1 and X_2 the transmitted signals, and H_1 and H_2 the corresponding channels between the two transmitters and receiver, respectively. For notation simplicity, we represent the received two-user superimposed signal as

$$Y = H_1X_1 + H_2X_2, \tag{2.1}$$

in the frequency domain. Note that the above representation is only valid in an ideal scenario. When it comes to practice, the time synchronization, oscillator offsets, and environmental noises should be taken into account. We will address these issues in this section and Section 2.4.

Two questions should be concerned before we design a practical system for decoding superimposed signals: (i) what is the required time synchronization level? (ii) can we achieve that level of synchronization accuracy on off-the-shelf devices?

We can address the first question by considering the usable number of Cyclic Prefix (CP) samples that are designed for a single source signal to address the synchronization problem. In detail, a single source signal would be reflected by the environmental objects, i.e., a multi-path fading channel, before arriving at the receiver, and the sig-

nals from different reflection paths would arrive at the receiver at the different time regarding the path length. These arriving time differences would cause a synchronization problem. To address this, typically, a receiver can use a sliding Fast Fourier Transform (FFT) window to slide within CP samples that are prefixed in each signal, which enables the receiver to include the signals with the highest power so that a high signal-to-noise ratio can be obtained. Here, the superimposed signal experiences the same multi-path fading channel as a single source signal does, since each signal has its own multipath components. However, not all the samples in CP can be used to adjust the FFT window for superimposed signals due to the inter-symbol interference (ISI) problem.

To see it clearly, we conducted a benchmark experiment in an office with rich multipath. As shown in Fig. 2.2, most of the multipath signals arrive at the receiver within the first 2 sample intervals, which is consistent with the previous observations [5]. In this case, the rest 14 sample intervals can be utilized to adjust the FFT window. Specifically, for IEEE 802.11 a and 802.11 p, the time duration of CP is 0.8 μs and 1.6 μs, respectively. Accordingly, the time duration of 14 samples is 700 ns and 1400 ns, respectively. Hence, the required time synchronization for the implementation of the superimposed signal is about hundreds of ns.

For the second question, many synchronization techniques have been developed recently, making this level of synchronization achievable on off-the-shelf devices. For instance, devices can maintain around 300 ns accuracy by only using a GPS clock [48]. Thus, current time synchronization techniques provide a solid foundation for implementing the superimposed signal on off-the-shelf devices. In this chapter, we focus on the unsolved problems that affect the practical implementation of the superimposed signal decoding, i.e., heterogeneous devices and dynamic environments with Non-Line-Of-Sight (NLOS) scenarios.

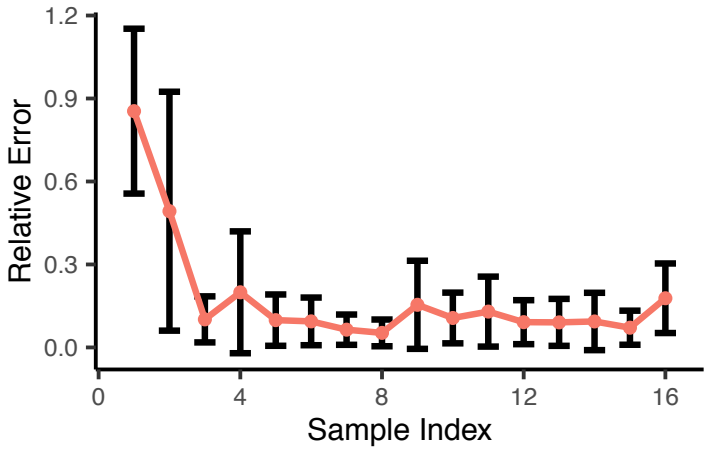


Figure 2.2: The multipath effect on CP. Delayed multipath signals appear only in the first two samples of the CP part.

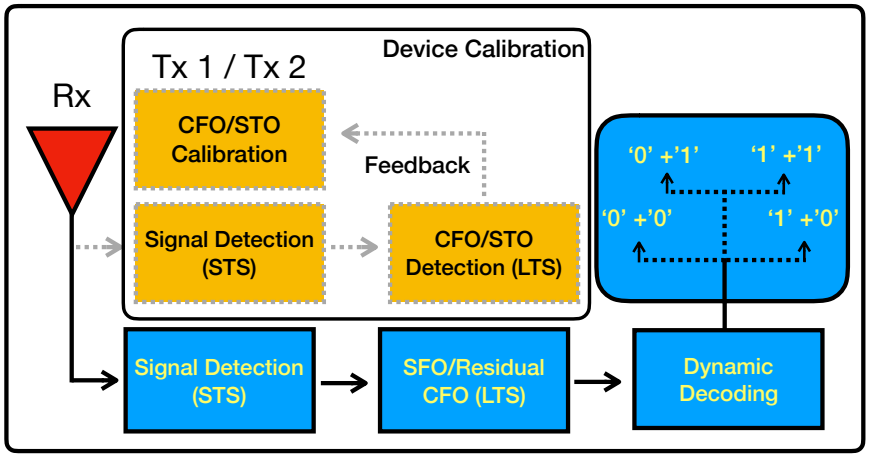


Figure 2.3: PhyCode overview.

2.4 The Design of PhyCode

In this section, we introduce the design of PhyCode. Four main modules of PhyCode have been shown in Fig. 2.3.

2.4.1 Preamble Design

We start the description of our design from the preamble pat of the transmitted signal. In a Wi-Fi system, every packet starts with a preamble including a Short Training Sequence (STS) and a Long Training Sequence (LTS). STS is used for signal

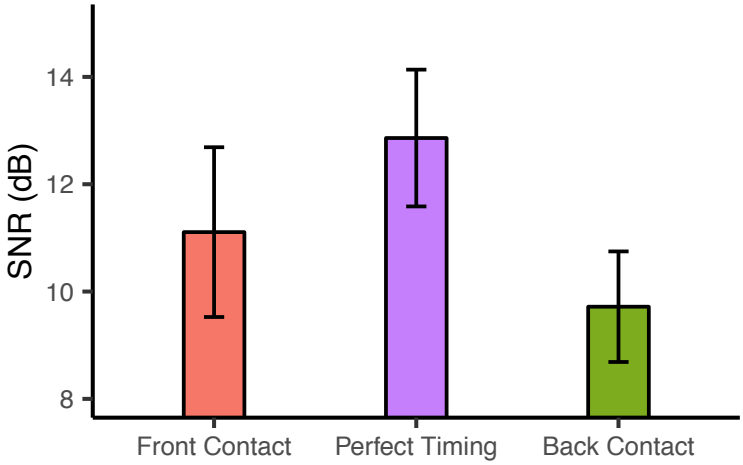


Figure 2.4: SNR under different positions of the FFT window.

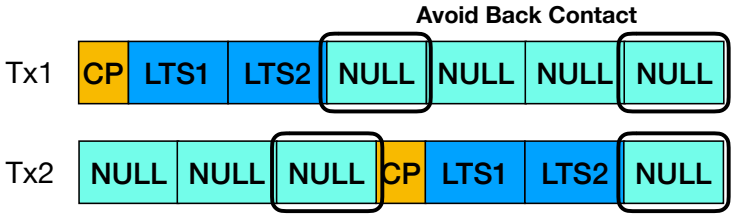


Figure 2.5: LTS design for the superimposed signal.

detection, while LTS is designed for measuring the difference between the received and transmitted pilot. For a superimposed signal, STS can still be used for signal detection (see Section 2.4.2). However, collision makes the existing solutions infeasible to measure those differences based on LTS. Naively, we can design an orthogonal LTS on signals from each source. In this case, when two signals arrive at the receiver, the differences can be measured separately. Unfortunately, it is infeasible to guarantee perfect orthogonality due to the device and channel variations in practice.

To analyze the effect of this non-perfect orthogonality on decoding performance, such as the Signal-to-Noise Ratio (SNR), we further conduct an experiment where we artificially introduce latency for different sources. For simplicity, we define LTS collision as "contact". In particular, *back contact* refers to the case where samples from other sources are included in the FFT process, while *front contact* means that a sliding FFT window only contains its own CP and data samples. The experiment

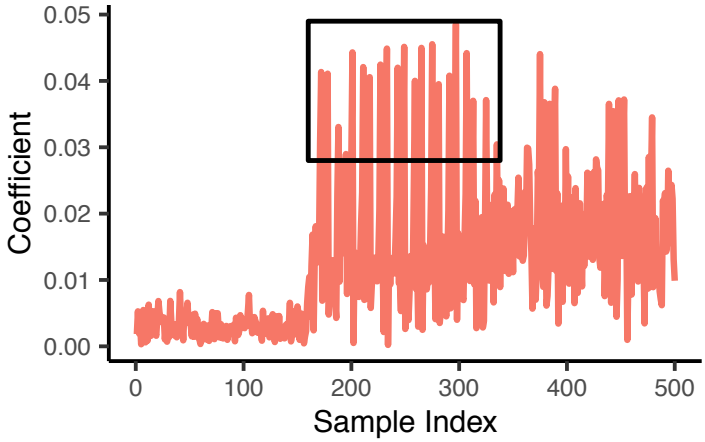


Figure 2.6: Cross-Correlation.

result is shown in Fig. 2.4. It is observed that the perfect FFT position gives us the best SNR, while the *front contact* induces a phase shift to the signal. In contrast, the *back contact* gives us the worst SNR. Note that the phase shift caused by the *front contact* can be corrected later through STO calibration (see Section. 2.4.4). However, the SNR degradation induced by the *back contact* cannot be corrected in further steps. This observation implies that it is needed to avoid the *back contact* cases. To do this, we design our own LTS as described in Fig. 2.5. Different from directly applying orthogonal LTS, we insert two extra NULL symbols to enlarge the distance of two sources in the time domain, which can successfully avoid the *back contact* in most of the practical cases. Note that the overhead of these newly added NULL symbols and the orthogonal LTS is negligible compared to the length of a packet.

2.4.2 Superimposed Signal Detection

Up to here, we build a clear view of the transmitted preamble. Next, we need to detect the superimposed signal properly at the receiver. In dealing with the superimposed signal detection, cross-correlation has been widely adopted to obtain a high detection accuracy [49]. The cross-correlation involves every coming sample to execute complex multiplication with a significant computation cost (e.g., 64 times of complex multipli-

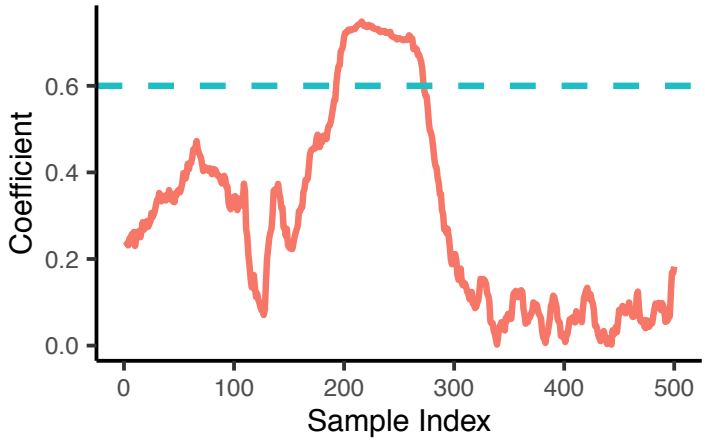


Figure 2.7: Auto-Correlation.

cation for each sample in 802.11 a/p). *However, when it comes to practice, there are two challenges remaining.* First, this theoretical detection accuracy cannot be maintained anymore in practical dynamic environments where channels are complicated. For example, Fig. 2.6 illustrates the correlation pattern of a superimposed signal in a common indoor environment with rich multipath. As we can see, the correlation pattern is not so clear to easily distinguish the two signals, leading to a low detection accuracy. Second, a high computation cost would lead to a low performance of the whole system [50], which will eventually hurt the decoding accuracy (i.e., incur many more error bits) at the receiver.

In order to solve these two challenges, we propose a two-step correlation method with a low computation cost and a comparable detection accuracy. In the first step, we apply auto-correlation [51] with STS to detect the signals in a coarse-grained manner (e.g., only one complex multiplication for each sample). By doing so, the computation cost can be reduced significantly. In the following step, we design a cross-correlation only for LTS to further improve the detection accuracy under the complicated channel. We plot the result of a benchmark experiment in Fig. 2.7. Clearly, this two-step correlation can achieve a comparable detection accuracy as the cross-correlation in a real-world scenario.

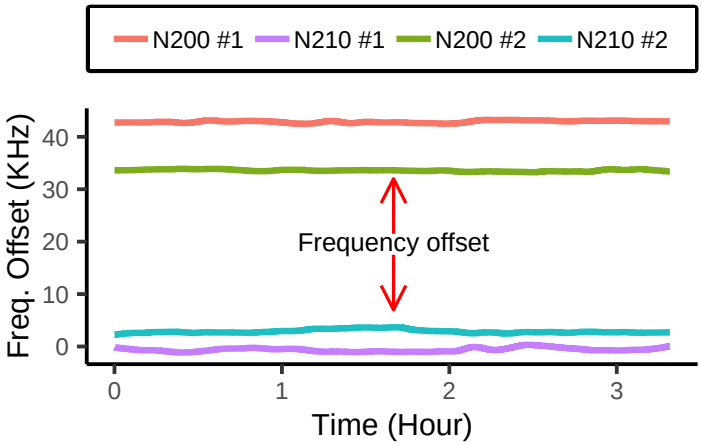


Figure 2.8: Carrier frequency offsets.

2.4.3 Carrier Frequency Offsets Calibration

After a proper detection of the superimposed signal, we can address the harmful synchronization errors. To understand the side effect of synchronization errors, we start from a single source signal. Typically, losing synchronization will distort the signal from three main respects: the Symbol Timing Offset (STO) denoted by n_ϵ, the Carrier Frequency Offset (CFO) denoted by Δf, and the Sampling Frequency Offset (SFO) denoted by ΔT. Specifically, the distorted signal $r(t_n)$ can be represented as below,

$$r(t_n) = e^{j2\pi\Delta f n T} \sum_i h_i(nT')(s(n - n_\epsilon)T' - \tau_i) + n_0, \tag{2.2}$$

where T' and T are the sampling time at the receiver and the transmitter, respectively [37] (the difference between T' and T will cause SFO). We denote h_i as the channel impulse response, τ_i as the delay, and n_0 refers to the white Gaussian noise. In the superimposed signal, however, the synchronization error becomes even worse as multiple offsets from different sources are involved. Here, we focus on addressing CFO first, and then STO and SFO will be analyzed in the next subsection.

As mentioned early (see Fig. 2.1), suppose that there are two transmitted signals with CFO Δf_1 and Δf_2, respectively. Therefore, applying one compensation to two

different CFOs simultaneously will lead to a high error rate. To design a practical implementable system, we must deal with the device heterogeneity and compensate Δf_1 and Δf_2, respectively. To do this, we design a two-step CFO correction scheme. In the first step, we calibrate CFO in a coarse-grained manner at the transmitters, while during the second step, the residual offset can be further corrected by applying a dynamic decoding scheme. In this case, CFO can be divided into two parts: $\Delta f_1 = \Delta f_{1_{\text{step1}}} + \Delta f_{1_{\text{step2}}}$ (Δf_2 can also be written like this). We first focus on the first step and the details of the second step are discussed in Section 2.4.5.

To design the first step of CFO calibration, we conduct a three-hour experiment in an ordinary office with different devices, including one USRP N210 as the transmitter and four USRPs as the receivers (two N210s and two N200s). As we can notice from Fig. 2.8, a large amount of CFO exists in different devices, especially when their types are different. However, it is very interesting to observe that the CFO differences are considerably stable even after a few hours. Hence, we leverage this observation, and compensate this CFO (i.e., $\Delta f_{1_{\text{step1}}}$) beforehand in a coarse-grained manner. After this calibration, only a small CFO still left, and based on that, we further develop the second step for fine-grained correction.

2.4.4 Timing Offsets Calibration

Besides CFO, the remaining offsets are STO and SFO. Intuitively, STO and SFO are both timing problems. The difference is that STO comes from the receiving process, such as limited computation power, noisy circuits, etc., which could introduce a few samples latency. On the other hand, SFO comes from the oscillator, sharing the same reason with CFO. Although STO and SFO are caused by different reasons, they both induce a phase shift to the signal. Here, we define θ^{STO} and θ^{SFO} for the phase shifts caused by STO and SFO, respectively. The phase shifts in one symbol can be

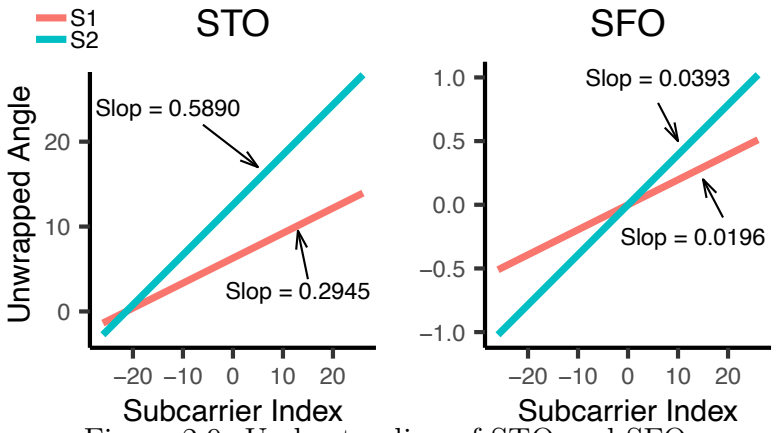


Figure 2.9: Understanding of STO and SFO.

described as

$$\theta_k^{STO} = 2\pi k n_\epsilon / N_c, \tag{2.3}$$

$$\theta_k^{SFO} = 2\pi k \gamma (N_c + L)/N_c, \tag{2.4}$$

where $\gamma = (T - T')/T$ is defined as the sampling time error ratio, k the subcarrier index, L the length of CP, and N_c the length of the data part in every symbol.

To understand STO and SFO clearly, we emulate signals to show how the time latency affects the phase shift. Specifically, we artificially add an integer multiple of sample interval as latency to emulate STO. For SFO, we add a fractional multiple of sample interval as latency. As shown in Fig. 2.9, there is a linear relationship between the subcarrier index and the phase shift. More importantly, the slope of STO is much bigger than SFO, which indicates that we can first calibrate STO and then based on this calibration result, we can zoom in to detect SFO.

STO Calibration: Since STO is relatively stable, we can measure and compensate it beforehand at the transmitter, just like the calibration of CFO.

SFO Measurement: Recall that the oscillator differences are the main cause of CFO and SFO. Hence, we can use the CFO error ratio $\epsilon = \Delta f / f$ to infer the SFO error ratio γ. Particularly, we have $\gamma \approx \epsilon$. According to Eq. (2.4), we can obtain θ^{SFO}

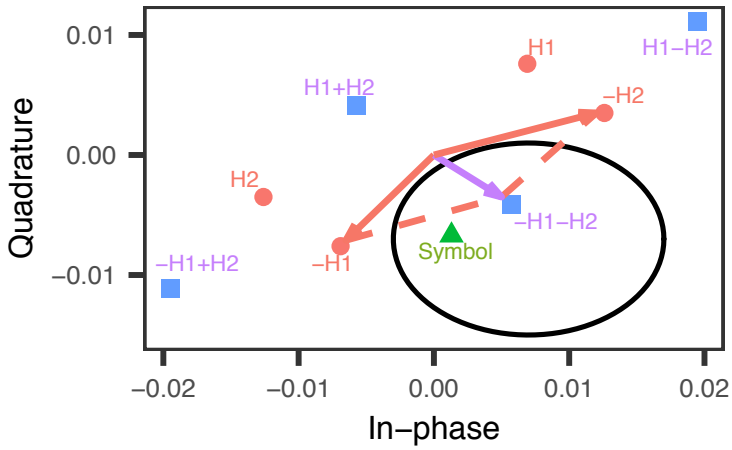


Figure 2.10: The constellation map.

with considerable high accuracy due to the stability of CFO. However, the residual CFO remains a problem for both CFO and SFO correction. To solve this problem, we assign two pilot samples for each source to keep tracking the residual CFO (Δf_{step2}) in every symbol.

2.4.5 A Dynamic Decoding Scheme

So far we have corrected CFO, calibrated STO and measured SFO. The remaining part is to decode the superimposed signal. However, as we emphasized before, it is extremely challenging to correct all offsets from multiple sources simultaneously. To solve this problem, we propose a dynamic decoding scheme that changes its decoding criterion for every sample according to the measured SFO and the residual CFO. Specifically, PhyCode combines channel conditions, SFO and residual CFO together to define this dynamic decoding criterion. By using the Binary Phase Shift Keying (BPSK) modulation scheme as an example (Fig. 2.10), there are four decoding possibilities based on different channel conditions H. In detail, $H_1 + H_2$ and $-H_1 - H_2$ represent "11" or "00". In contrast, $H_1 - H_2$ and $-H_1 + H_2$ represent "10" or "01". We can therefore decode a symbol by calculating the shortest Euclidean distance to

these constellation points.

After combining SFO and residual CFO, we denote C as the decoding criterion which is a combination of the above channel condition H, SFO and residual CFO. Next, by solving an optimization problem, we can decode the superimposed signal at the receiver. For simplicity, we take two transmitters as an example. Accordingly, the optimization problem can be written as follows,

$$\min_{m,n} \|(p_m C_1 + p_n C_2) - \tilde{x}\|_2, \tag{2.5}$$

where $\tilde{x}$ is the superimposed signal symbol to be decoded, and C_1 and C_2 are the decoding criteria of the two transmitters, respectively. We denote the constellation points set for the K-QAM modulation as $P^K = \{p_1, p_2, ..., p_{2^K}\}$. We have $p_m, p_n \in P^K$ and $m, n = 1, 2, ..., 2^K$. For example, in the BPSK modulation, $P^2 = \{-1, 1\}$. More generally, suppose that there are N transmitters using the K-QAM modulation, hence, the above optimization problem can be extended as

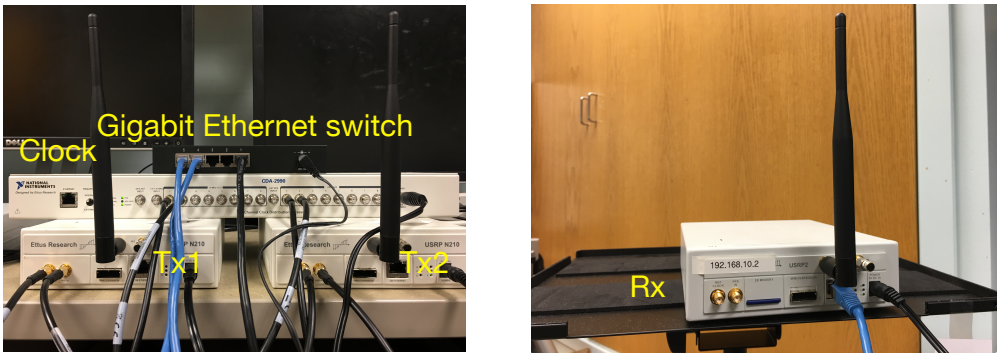
$$\min_{g_1, g_2, ..., g_N} \|(p_{g_1} C_1 + p_{g_2} C_2 +, ..., + p_{g_N} C_N) - \tilde{x}\|_2, \tag{2.6}$$

where $p_{g_1}, p_{g_2}, ..., p_{g_N} \in P^K$ and $g_1, g_2, ..., g_N = 1, 2, ..., 2^K$.

2.5 Experimental Evaluation

2.5.1 Implementation

Hardware-wise: We implement PhyCode on a software-defined radio platform. The hardware setup of PhyCode is shown in Fig. 2.11(a) and Fig. 2.11(b). Specifically, we use 7 Universal Software Radio Peripheral (USRP) embedded with XCVR2450 daughterboards, including three N210s and four N200s, and two USRPs connect



(a) Transmitters

(b) Receiver

Figure 2.11: The testbed.

to a PC through a Gigabit Ethernet switch. Without loss of generality, PhyCode follows the IEEE 802.11 p standard, i.e. the 5.8 GHz carrier frequency and 10 MHz bandwidth, which can be easily applied to other OFDM related protocols. For the time synchronization, we use NI CDA-2990 as the central clock and each USRP that acts as a transmitter is connected to this clock via SMA cables.

Software-wise: The software of PhyCode is based on a recent Wi-Fi project programmed in GNURadio [50]. In detail, we develop PhyCode transmitters by modifying the preamble and pilot samples as described in Section 2.4.1 and Section 2.4.5, respectively. For PhyCode receivers, we implement the two-user superimposed signal detection and offset compensations, such as CFO, STO and SFO, as presented in Section 2.4.2 to Section 2.4.5.

2.5.2 Methodology

Our goal is to evaluate the performance of PhyCode for dealing with heterogeneous devices in dynamic environments. First, we focus on the influence of heterogeneous devices. To do this, we use power combiners and 30 dB attenuators to connect two transmitters to the receiver, which can emulate a stable wireless channel in order to avoid the impact of the dynamic environment. We randomly pick up 3 out of 7 USRPs

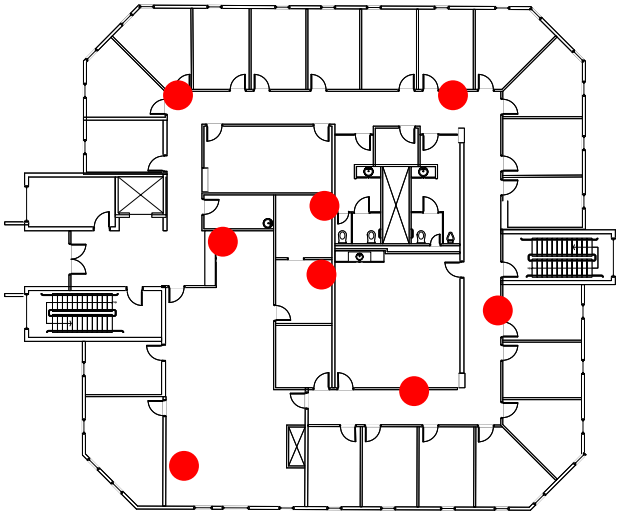
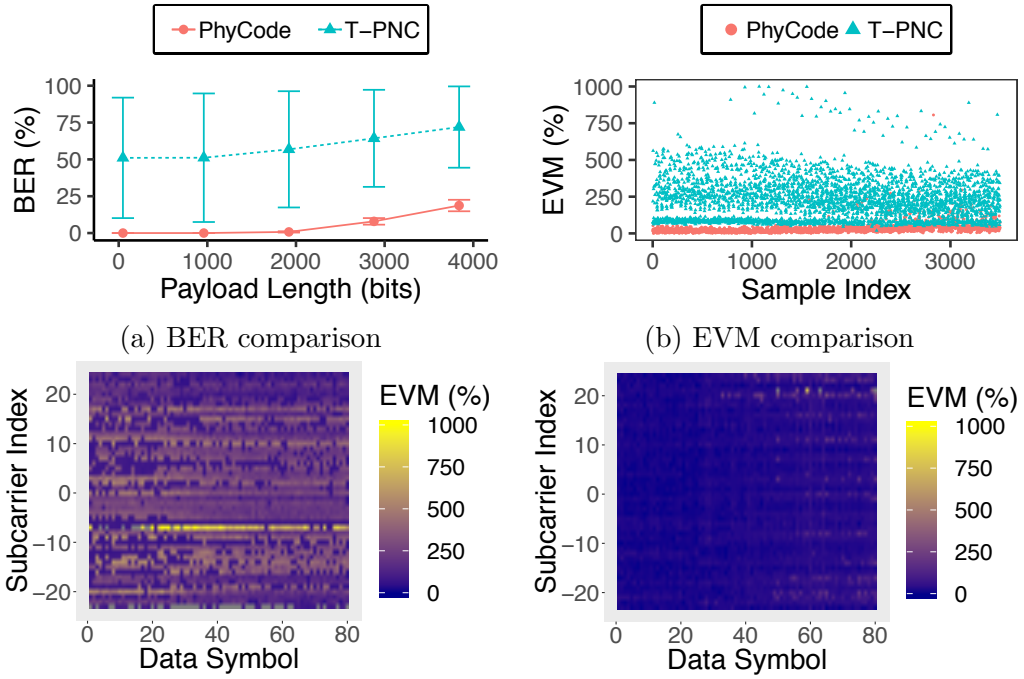


Figure 2.12: The deployment layout.

to be the transmitters and the receiver at each time. This experiment was repeated 3 times in an ordinary office. Specifically, during each time, we vary the payload length from 48 bits to 4000 bits. For a fixed payload length, PhyCode transmits a packet every 5 seconds and it lasts for one hour.

Second, we evaluate our design in dynamic environments with NLOS scenarios. To this end, we deploy our devices at 8 different locations (Fig. 2.12) in our building, including both LOS and NLOS scenarios. Specifically, each time we randomly choose 2 out of 8 locations—one location for the two transmitters and the other location for the receiver, to deploy PhyCode, and the minimum distance between the two transmitters are at least 50 cm in order to form independent channels. Both transmitters send 2000 bits payload every 5 seconds. This experiment was repeated 10 times and the total experiment lasts for 5 hours. During the experiments, people in the building work as usual, i.e., they can either sit in their offices or walk around the corridors, which contributed to a dynamic environment.



(a) BER comparison

(b) EVM comparison

(c) EVM of T-PNC (subcarriers)

(d) EVM of PhyCode (subcarriers)

Figure 2.13: Performance comparison.

2.5.3 Metrics

We use the following two metrics for the evaluation: (a) Bit Error Rate (BER): the percentage of bits in error in a PhyCode packet; (b) Error Vector Magnitude (EVM): a measure of how far the received symbols are from the ideal locations—the constellation points—in the constellation map, which is a fine-grained error analysis of each sample. Both metrics are related to the decoding rate. We compare PhyCode with the state-of-the-art PNC implementation [22, 23], where only an average CFO was compensated to the superimposed signal. For simplicity, we denote these kinds of PNC implementation as T-PNC in the following comparison.

2.5.4 Impact on Heterogeneous Devices

We plot the comparison result of BER in Fig. 2.13(a). Obviously, PhyCode outperforms T-PNC substantially. The underlying reason is that T-PNC suffers deeply from

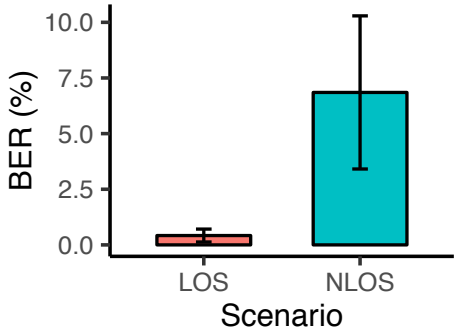


Figure 2.14: BER in dynamic environments.

the offsets, especially for the diverse behaviors of different oscillators. In contrast, PhyCode reacts to multiple offsets effectively. To see it more clearly, we randomly pick up one payload with 4000 bits to evaluate the result from signal constellations in a fine-grained manner. As shown in Fig. 2.13(b), the EVM result of T-PNC is scattered around and is larger than that of PhyCode, which indicates the presence of a large amount of CFO. Although they compensated the signal with an average offset, the superimposed signal is still severely affected by the offsets. Furthermore, from the view of each subcarrier, we investigate the effects of STO and SFO. As shown in Fig. 2.13(d), PhyCode can mitigate STO and SFO effectively. However, T-PNC gets hurt from the offsets in every subcarrier as revealed in Fig. 2.13(c).

We note that PhyCode compensates the signal well in most of the cases. But with the symbol index increasing, the damage of residual offsets becomes more obvious. In dealing with this situation, we can insert channel estimation pilot symbols periodically to ensure an accurate estimation, so we can always keep our decoding success rate at an acceptable level.

2.5.5 Impact on Dynamic Environment with NLOS

Fig. 2.14 shows the BER results of PhyCode in both the LOS and NLOS scenarios, respectively. All results indicate that our design is feasible to be implemented in a

practical dynamic environment with a considerably low raw BER.

2.6 Conclusion and Discussion

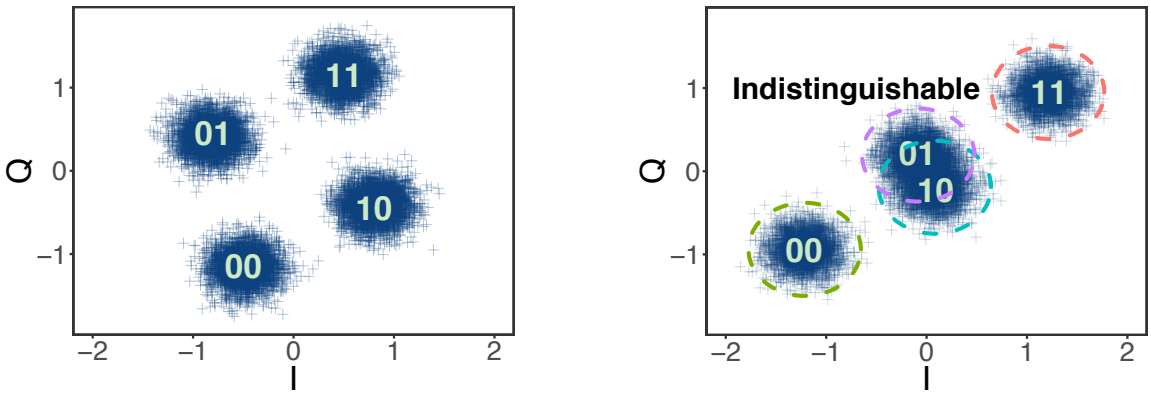
This chapter introduces PhyCode, a practical wireless prototype of superimposed signals in the presence of heterogeneous devices and dynamic environments, such as time synchronization errors and oscillator offsets. We demonstrate the feasibility of our design through implementation on a software-defined radio based platform.

Chapter 3

SigMix: Decoding Two-user Superimposed Signals with High Reliability

3.1 Introduction

In chapter 2, we understand that to boost the spectrum efficiency under severe competition, a promising solution is to allow concurrent wireless transmissions, and decode the two-user superimposed signals via careful signal processing. PhyCode paves a way for decoding the two-user superimposed signal for wireless communications, however, the underlying idea of it still comes from the modulation-level decoding for a single source signal, which does not reveal the rich features of two-user superimposed signals, and therefore, cannot fully utilize the superimposed signal. As a result, we notice that in certain scenarios the decoding reliability of PhyCode can vary noticeably, making an unstable decoding performance and wastage of wireless spectrum. Using a toy example shown in Fig. 3.1(a), when two Binary Phase-Shift Keying (BPSK)



(a) Example A (b) Example B

Figure 3.1: Two-user superimposed signals in the IQ domain.

modulated signals arrive at a receiver concurrently, four possible combinations (i.e., four constellation points) exist in the IQ domain, representing "11", "10", "01" and "00". Specifically, with noises, the received signals are four clusters centered at the four constellation points. Next, following a maximum likelihood decoding scheme, the superimposed signal can be decoded. One thing worth noting is that this decoding process is based on an assumption—the constellation points are genuinely distinguishable. However, in practice, as shown in Fig. 3.1(b), the constellation points may be so close to each other that their corresponding clusters are indistinguishable, resulting in a high probability of decoding error. This situation occurs due to the signal variation caused by dynamic channel conditions and hardware imperfections, and it will be exacerbated in IoT systems. Consequently, directly using the modulation-level decoding scheme—PhyCode—for two-user superimposed signals may lead to a decoding performance with low reliability.

In this chapter, we present SigMix, the first practical system that can reliably decode the two-user superimposed signals under dynamic channel conditions and hardware imperfections for IoT systems. SigMix is based on the understanding of the new features specifically appearing in superimposed signals. By doing this, SigMix provides a reliable performance across a wide range of practical scenarios. Beyond

that, given the popularity of Orthogonal Frequency-Division Multiplexing (OFDM), SigMix is applied to OFDM systems with multiple subcarriers.

In a nutshell, SigMix is based on the observation that, for superimposed signals, the distance between constellation points are largely determined by the phase shift between transmitted signals. Therefore, we can manipulate the phase shift to achieve desirable decoding performance. It is non-trivial to realize SigMix due to the following challenges:

- First, a guideline is missing for manipulating the phase shift. To be specific, the optimal phase shift for decoding the superimposed signal is unknown yet.
- Second, due to dynamic channel conditions and hardware imperfections in IoT systems, the signal variations are unpredictable, making it difficult to maintain the optimal phase shift at the receiver.
- Third, the superimposed signal decoder may encounter serious signal variations. Especially, in OFDM systems, the channel conditions of different subcarriers are different [52, 53]. Furthermore, as discussed in Chapter 2, the hardware imperfections cause the Carrier Frequency Offset (CFO), the Sampling Frequency Offset (SFO) and the Sample Timing Offset (STO) to the signal. These offsets cannot be easily compensated for superimposed signals, and also exacerbate the differences among subcarriers, resulting in a poor decoding performance.

To deal with the first challenge, we derive an exact Bit-Error-Rate (BER) expression to model the relationship between the phase shift and the decoding error probability. This BER expression provides us an important guideline to manipulate the phase shift to achieve a high decoding performance, in terms of a low decoding error rate.

To deal with the second challenge, we propose a shifting code that enables SigMix to transmit two copies of the signal to achieve a substantial diversity gain. Specifically,

we only shift the phase of one copy of the signal and keep the other copy as the original one. By doing so, we can eliminate the signal variation caused by dynamic channel conditions and hardware imperfections. Thus, the receiver can eventually decode the superimposed signal with the best phase shift. To further obtain the optimal shifting angle, we propose a searching scheme based on our theoretical analysis.

To deal with the last challenge, we propose an adaptive decoding scheme considering both the subcarriers' differences and the three offsets. With this design, SigMix decodes the superimposed signal based on the behaviors of each subcarrier, which can handle the signal variations well and achieve a high decoding performance.

To the best of our knowledge, this paper is the first to present a practical decoding approach for the two-user superimposed signals in the presence of dynamic channel conditions and hardware imperfections. Note that SigMix can be an enabler for many promising wireless technologies requiring the decoding of superimposed signals, such as NOMA and PNC. SigMix is presented in the context of OFDM, hence, the basic idea can be extended to many application scenarios, e.g., LTE, IEEE 802.11 a/g/n/p, etc.

Contributions: This paper makes the following contributions:

- It is the first to reveal the relationship between the phase shift of the concurrent signals and the decoding BER by leveraging Craig's analytical model. As a result, we use this relationship as a guideline in manipulating the phase shift so that a lower BER can be obtained.
- It presents the first practical approach for decoding superimposed signals through a shifting-code based diversity transmission and an adaptive decoding scheme. Our approach can achieve a high decoding performance regardless of the practical challenges in IoT systems, i.e., dynamic channel conditions and hardware imperfections, substantially enhancing the decoding ability in practice.

Properties	BASIC [54]	CoReCast [16]	PNC [21, 22]	ANC [31]
Data Rate	∼ Mbps	∼ Mbps	∼ Mbps	∼ Kbps
Synchronization Level	Packet	Symbol	Symbol	Packet
Power Control	Yes	Yes	No	No
Interference Decoding	Yes	Yes	No	No
Modulation Scheme	OFDM	OFDM	OFDM	MSK
Device Cost	Low	High	Low	Low
Error Control	Retran.	Retran.	Coding	Retran.

a

Properties	BiPass [32]	NetScatter [47]	Hubble [45]	SigMix
Data Rate	∼ Mbps	∼ Kbps	∼ Kbps	∼ Mbps
Synchronization Level	Packet	Symbol	Symbol	Symbol
Power Control	No	No	No	No
Interference Decoding	No	Yes	Yes	Yes
Modulation Scheme	OFDM	CSS	On-Off Key	OFDM
Device Cost	High	Low	Low	Low
Error Control	Retran.	Retran.	Coding	Coding

b

Table 3.1: **Comparison of related work in handling superimposed signals.**

- It demonstrates a practical system on a software-defined radio based platform and evaluates its performance across various scenarios. The extensive experimental results illustrate that SigMix obtains a one-order lower median BER than the state-of-the-art system.

The rest of this chapter is organized as follows. Section 3.2 discusses the related work. Section 3.3 presents the background knowledge. Section 3.4 introduces the design of SigMix. Section 3.5 discusses several important practical issues. Section 3.6 presents the evaluation results and further discussions. We conclude our work in Section 3.7.

3.2 Related Work

Prior work falls into the following four categories.

(a) Successive Interference Cancellation (SIC): To decode the superimposed signal, SIC requires strict power control to guarantee that one signal has much higher power than the others. With the power differences, they can decode one of the signals first while treating others as noise, and then cancel it out to decode the rest. Eventually, all signals can be decoded separately by repeating this procedure, e.g., BASIC [54] and CoReCast [16]. As a major NOMA technology, SIC is promising for future cellular networks [9, 10, 55]. However, it relies on infrastructure and channel feedback for the strict power control, which may not be desirable for low-cost IoT devices and may cause extra delay [56, 57]. In contrast, SigMix does not need power control and therefore can support a wider range of IoT applications.

(b) Physical-layer Network Coding (PNC): The superimposed signals can be also decoded with the help of a relay node [1] [35, 58]. This idea has been moved from theory to practice by the implementation of PNC [21, 22] and analog network coding (ANC) [31]. Further works extended these systems to be more robust [22, 23], scalable [33, 34], and achieving higher throughput (e.g., BiPass [32] employed the high-cost full-duplex devices). However, these works were designed for relay networks, making them incompatible to decode the superimposed signals without the help of other nodes. Therefore, these approaches cannot be used in a general scenario such as multiple access.

Accordingly, more recent works proposed to decode the superimposed signals directly at the receiver without any helper. Strong assumptions such as a perfect channel measurement and a stable environment are required [59]. The perfect channel measurement can only be obtained by dedicated hardware [5, 59], such as cellular base stations, which can hardly be achieved in the low-cost IoT devices [60]. Instead

[1] In relay networks, two end nodes transmit signals concurrently so signals are superimposed at the relay, and then the relay node broadcasts this superimposed signal back the the end nodes. Each end node can decode the signal transmitted from the other end node by canceling out its own signal.

of a perfect channel measurement, researchers proposed to decode the superimposed signals with retransmissions to reduce the decoding error [16, 28, 61, 62]. But the retransmission may introduce undesirable long delay and higher energy cost. Here, we include the design in chapter 2, PhyCode, into this category as it can decode the superimposed signal without any power control. Although PhyCode requires no helper, it does not capture the rich features of the superimposed signal, and hence rely on a strong assumption that constellation points are naturally distinguishable, providing a unreliable decoding performance. To address this problem, similar to PNC, PhyCode requires retransmissions to improve the reliability.

Different from the above work, SigMix can decode the superimposed signal directly at the receiver without any helper. Furthermore, with an adaptive decoding scheme, SigMix can achieve high decoding performance (i.e., a lower BER) without relying on the perfect channel measurement nor retransmissions.

(c) Superimposed Signals in Low Data Rate Systems: Decoding superimposed signal has also been widely adopted in many IoT techniques with a low data rate, such as LoRa [46] (e.g., NetScatter [47]), RFID [43,44,56] (e.g., Hubble [45]) and ZigBee [36]. These approaches are promising for low data rate applications, such as weather reports that only cost a few packets per minute [63]. However, applications with general data rate requirements, for example, farm monitoring [64] are beyond the capability of these approaches. Indeed, decoding the superimposed signals in general data rate technology supported by OFDM is much challenging than that in the low data rate technology, since the signals are more vulnerable to dynamic channel conditions. Different from these works, SigMix can be applied to general data rate systems, such as OFDM, which is more efficient in using the spectrum.

(d) Multiple-Input and Multiple-Output (MIMO) Systems and Superimposed Signals: Decoding the superimposed signal using MIMO technologies has

attracted many research interests [5, 15], where signals are transmitted concurrently via multiple antennas co-located within one transmitter instead of multiple transmitters, but the MIMO solution may be too expensive for low-cost IoT devices. On the other hand, SigMix is not only applicable for low-cost IoT devices, but also compatible with MIMO systems, especially for distributed MIMO applications [40, 65].

As a summary, a detailed comparison between SigMix and the existing works is shown in Table 3.1.

3.3 Preliminary

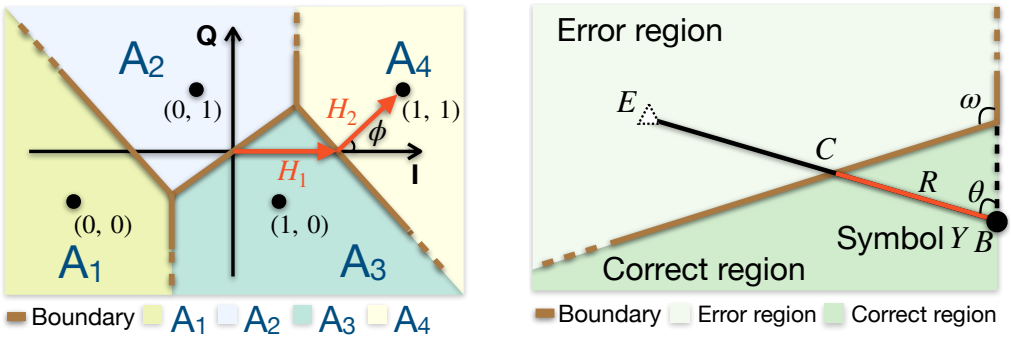
In this section, we present the background information related to SigMix.

3.3.1 Two-user Superimposed Signal's Representation

We provide the background of two-user superimposed signals in chapter 2, and here we present the knowledge again with more details. When different transmitters transmit to the same receiver concurrently, the receiver will receive a two-user superimposed signal, i.e., a combination of two transmitted signals over the air. In the frequency domain, without loss of generality, we define the received superimposed signal as Y, the channel matrix as $\mathbf{H}$, the transmitted signals as $\mathbf{X}$, and the additive white Gaussian noise as N. Then, we have the expression of Y as

$$Y = \boldsymbol{H}\boldsymbol{X} + N \tag{3.1}$$

Here, m elements in $\mathbf{X}$ and $\mathbf{H}$ respectively represent m transmitters. $\mathbf{H}$ represents the signal variation caused by both the channel and hardware. It is worth noting that the concurrent transmissions fall into the category of symbol-level synchronization that can be achieved by existing implementations [17, 41]. More details will be discussed



(a) Craig's polar coordinate (b) The unit shape

Figure 3.2: Craig's analytical model.

in Section 3.5.

For decoding the superimposed signal, we convert the above frequency-domain representation to the IQ domain. We start from considering the mix of two BPSK modulated signals for simplicity. When a receiver receives two signals simultaneously, if without noises, there are four possible combinations (i.e., four constellation points) exist in the IQ domain, representing "11", "10", "01", and "00". Given random noises, the received signals will be distributed in clusters centred at these constellation points (see Fig. 3.1).

3.3.2 Craig's Analytical Model

Craig's Polar Coordinate

For superimposed signals, the locations of the constellation points are irregular, making the decoding error probability difficult to calculate. The Craig's polar coordinate [66] can largely simplify the decoding error probability calculation. As shown in Fig. 3.2(a), in the constellation map, one of the channel coefficients (e.g., H_1) is co-located with the I axis when using Craig's polar coordinate. The phase shift between transmitted signals is defined as ϕ (i.e., the angle between H_1 and H_2). Four black dots that represent the four constellation points. The whole area is partitioned into

four decision regions, i.e., region A_1, A_2, A_3 and A_4, namely, a Voronio diagram. The boundaries of the decision regions are shown as the brown lines. Here, region A_1 and region A_4 are symmetrical with the origin. Similarly, region A_2 and region A_3 are also symmetrical. Note that both H_1 and H_2 determine the locations of the constellation points and the corresponding decision region. Following a maximum likelihood decoding, any received symbol[2] in each region will be decoded to the symbol represented by the constellation point in that region.

Decoding Error Probability

When a received symbol is incorrectly located to other decision regions, an error happens. We next introduce how to use Craig's analytical model to calculate the probability of making that error.

Considering a general case shown in Fig. 3.2(b), Y is a received symbol and it can be correctly decoded only if it is located in the correct decision region, say location B. Due to the effect of Gaussian noise, Y may be drifted away to a location, say E, in an error region. Here, the error region is a wedge area bounded by two rays (i.e., two boundaries). The correct location B is outside the error region and it is on the extension line of one boundary ray. For simplicity, we define the above structure as a *unit shape*. The Craig angle ω is defined as the angle between two boundaries. The segment EB cross the decision boundary at point C, and the distance from B to C is notated as R. Then, according to the theory in [67], the possibility when a received symbol is incorrectly located to the error region can be defined as the decoding error probability P, given by

$$P = \int_0^{\omega} d\theta \int_L^{\infty} p(r, \theta) dr, \tag{3.2}$$

[2]A two-user superimposed signal consists of many symbols. When two BPSK modulation signal are superimposed, each symbol carriers two bits from two transmitters.

where $p(r, \theta)$ is the polar form of the bivariate Gaussian distribution function, and r is the length of segment EB. After a series of transformations [67], Eq. (3.2) can be written as

$$P = \frac{1}{2\pi} \int_0^{\omega} \exp\left(-\frac{R^2(\theta)}{2\sigma^2}\right) d\theta, \tag{3.3}$$

where $2\sigma^2$ refers to the noise density and $R(\theta)$ means the distance R is a function of θ. Fortunately, Eq. (3.3) only contains elementary functions, which can largely simplify the calculation process. We refer the readers who are interested in Eq. (3.3) to [67] for more details. Note that Eq. (3.3) can only calculate the decoding error probability under the unit shape.

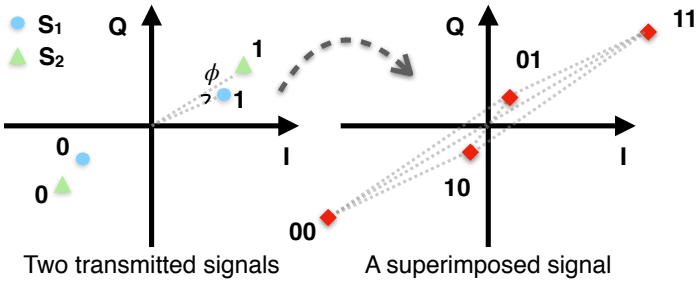
3.4 The Design of SigMix

In this section, we first formulate a closed-form BER expression to quantify the decoding performance and derive the factors affecting the decoding performance. With the guideline from the detailed quantitative analysis, we propose how to manipulate the phase shift between the transmitted signals, and then adaptively decode the two-user superimposed signal.

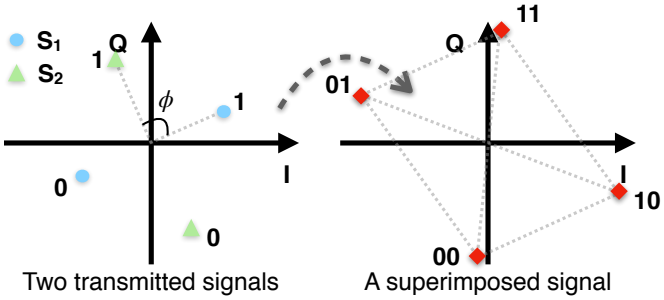
3.4.1 A Theoretical Guideline

Intuitive Observation

As shown in Fig. 3.3(a), two transmitters, S_1 and S_2, transmit signals concurrently. We first consider perfect channels where no signal variation exists. When the phase shift ϕ between two transmitted signals is 10°, two constellation points ("10" and "01") are very close to each other, making it difficult to decode successfully. However,



(a) $\phi = 10°$



(b) $\phi = 90°$

Figure 3.3: Phase shift vs. Constellation points.

when the phase shift is changed[3] to 90°, as shown in Fig. 3.3(b), the four constellation points are far away from each other, making it easy to decode.

From the above analysis, we observed that the distances between constellation points in the IQ domain are largely determined by the phase shift. Apparently, enlarging the distances can provide a high chance to decode the superimposed signal correctly. Therefore, a theoretical guideline is needed to model the relationship between the phase shift and the decoding error rate. It is worth noting that what we desire is an exact BER expression including both amplitude and phase related information, rather than coarse bounds of error probability obtained in [28, 62, 68] that relate to the Signal-to-Noise Ratio (SNR) only.

[3]Theoretically, we can also change the signal amplitude to change the distances between the constellation points. However, manipulating the signal amplitude may be impractical due to the low-cost IoT devices, and it requires feedback channel information which may cause extra delay.

A Closed-form BER Expression

We use Craig's polar coordinate to derive the exact BER expression. The decoding error probability, P, can be written as

$$P(e) = \sum_{i=1}^{4} P(e|A_i)P(A_i), \tag{3.4}$$

where $P(A_i)$ is the probability of transmitted symbols corresponding to the constellation point in decision region A_i, and $P(e|A_i)$ is the probability when an error happens given A_i, i.e., the received symbol is wrongly located in the error region outside A_i. Obviously, the received symbol shares equal probabilities to be in any decision region, so $P(A_i) = \frac{1}{4}$. For $P(e|A_i)$, we have

$$P(e|A_i) = \sum_{j \neq i} P(A_j|A_i), \tag{3.5}$$

where $P(A_i|A_j)$ refers to the probability that the received symbol should be in region A_j but be wrongly located in region A_i ($i, j = 1, 2, 3, 4$ and $i \neq j$). Recall that in Craig's polar coordinate, region A_1 and region A_4 are symmetrical, while region A_2 and region A_3 are symmetrical. For simplicity, we notate the symmetrical region for region A_i as region $A_{\bar{i}}$. Thanks to this symmetry, we have $P(A_i|A_j) = P(A_{\bar{i}}|A_{\bar{j}})$, where $i \neq j$ (e.g., $P(A_1|A_3) = P(A_4|A_2)$). Placing Eq. (3.5) into Eq. (3.4), and applying the above symmetrical property, the decoding error probability can be simplified as

$$P = \frac{1}{2}(P(e|A_3) + P(e|A_4)). \tag{3.6}$$

Here, $P(e|A_3)$ is the summation of $P(A_1|A_3)$, $P(A_2|A_3)$, and $P(A_4|A_3)$. In Eq. (3.6), $P(A_1|A_3)$, $P(A_4|A_3)$ and $P(e|A_4)$ can be directly calculated by past work [66, 69]. So, we only need to calculate $P(A_2|A_3)$, i.e., the probability when the received

symbol corresponding to the constellation point in A_3 ("10") is located in A_2 ("01") by mistake. Indeed, "10" and "01" are the most challenging cases to be distinguished for decoding.

To calculate $P(A_2|A_3)$, recall that the decoding error probability can be calculated by Craig's analytical model under the unit shape (Section 3.3). However, the boundaries of decision regions in Craig's polar coordinate vary with the channel coefficients. Furthermore, the wedge shape of all decision regions (see Fig. 3.2(a)) is more complicated than a unit shape, making it difficult to calculate the decoding error probability.

To solve this problem, we start from the condition when $|H_1| \geq |H_2|$ and $0 \leq \phi \leq \frac{\pi}{2}$ (ϕ is the phase shift between H_1 and H_2)[4]. Then, we extend our expression to other conditions, such as $|H_1| < |H_2|$ or $\frac{\pi}{2} < \phi \leq \pi$. Under each condition, we divide this condition into three cases for further simplification. For each case, we carefully partition the wedge decision regions into several unit shapes, and then we can apply Craig's analytical model to each divided unit shape, respectively. We detail the derivation for all conditions and cases in the Appendix, and we encourage the readers who are interested in the derivation to go through the calculations.

The derivation reveals that $P(A_2|A_3)$ is a function of ϕ, $|H_1|$ and $|H_2|$. Also, from [66], we know that $P(A_1|A_3)$, $P(A_4|A_3)$ and $P(e|A_3)$ are all functions of ϕ, $|H_1|$ and $|H_2|$. Therefore, the decoding error probability P is a function of ϕ, $|H_1|$ and $|H_2|$ base on Eq. (3.6). For simplicity, we do not show the detailed expression in the chapter. We notate the expression of decoding error probability P as $P_f(\phi, |H_1|, |H_2|)$.

[4]For $\phi = 0$, all the possible constellation points are located on one line in the constellation map. So, we can directly employ the BER result from the existing solution [69].

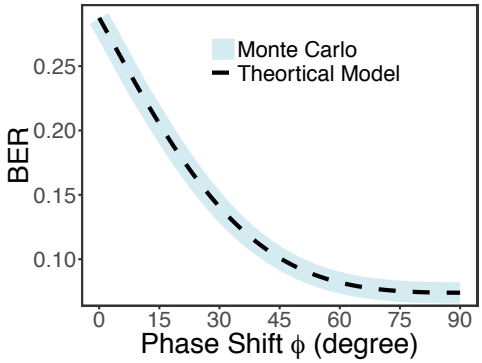


Figure 3.4: A comparison result of the BER expression.

Finally, the expression of P under all conditions can be calculated as

$$P = \begin{cases} P_f(\phi, |H_1|, |H_2|), & \text{if } |H_1| \geq |H_2|, 0 \leq |\phi| \leq \pi/2, \\ P_f(\pi - \phi, |H_1|, |H_2|), & \text{if } |H_1| \geq |H_2|, \pi/2 < |\phi| \leq \pi, \\ P_f(\phi, |H_2|, |H_1|), & \text{if } |H_1| < |H_2|, 0 \leq |\phi| \leq \pi/2, \\ P_f(\pi - \phi, |H_2|, |H_1|), & \text{if } |H_1| < |H_2|, \pi/2 < |\phi| \leq \pi. \end{cases} \tag{3.7}$$

With the above expression, we can finally obtain the relationship between the phase shift ϕ and the decoding error probability when given the channel coefficients. This closed-form BER expression is scalable to higher modulation and multiple concurrent transmitters.

Validation of the BER Expression

To validate the proposed closed-form BER expression, we conduct a Monte Carlo simulation and compare the simulation results with the BER expression. For simplicity, two transmitters have the same power and transmit signals under equal-gain channels (i.e., $|H_1| = |H_2|$) but varying phase shift, ϕ. We set the SNR of transmitted signals as 5 dB. The comparison results are shown in Fig. 3.4. The results from our theoretical model and the Monte Carlo simulation are very close to each other, which validates the correctness of the proposed closed-form BER expression. Note that in

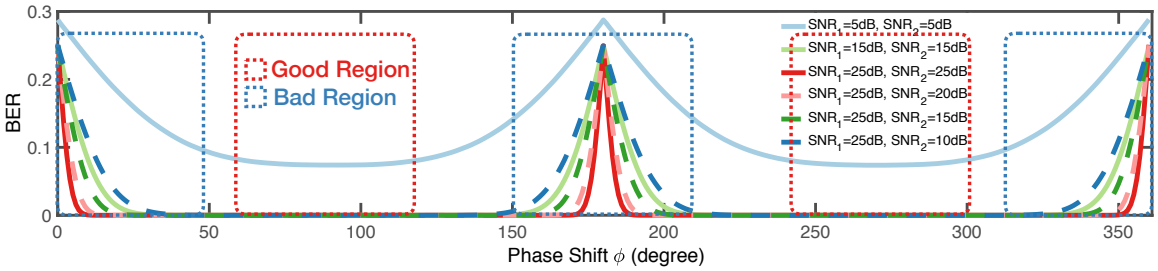


Figure 3.5: Phase shift vs. BER.

the Monte Carlo simulation, we also validate other settings, such as different SNRs and power ratios of two transmitters, and all the simulation results are consistent with our BER expression.

Numerical Analysis of the BER Expression

Eq. (3.7) reveals that the decoding error probability is determined by three factors: $|H_1|$, $|H_2|$ and ϕ. As mentioned earlier, controlling $|H_1|$ and $|H_2|$ requires strict power control which may be impractical for the IoT devices. Therefore, we first study the effect of phase shift ϕ on the decoding performance, given $|H_1|$ and $|H_2|$.

As shown in Fig. 3.5, we plot the BER curves under different SNRs according to Eq. (3.7). Here, $\text{SNR}_i = 10\log_{10}|H_i|^2/(2\sigma^2), i = 1, 2$. In the following, we use SNR_i which is widely used in communication systems instead of $|H_i|$ for the BER expression. Our goal is to analyze the relationship between phase shift ϕ and the BER given SNR. Specifically, the SNR value varies from 5 dB to 25 dB which is a common range for communication systems [31]. For the two transmitters, we randomly choose the SNR values within this range. Given each setting of SNR_1 and SNR_2, we vary the phase shift ϕ from 0° to 360° and plot the BER curve. We reveal several unique *properties*:

(i) The BER during the process of increasing the phase shift is non-monotonous. Two valleys appear when the phase shift $\phi = 90°$ and $\phi = 270°$. We call the range of ϕ corresponding to a lower BER as 'good regions'. Thus, the best choice of ϕ for the 'good regions' is centered around 90° or 270°. This motivates us to manipulate the

phase shift to be in 'good regions' to achieve high decoding performance.

(ii) In contrast to 'good regions', the peaks occur at certain phase shift values, i.e., $\phi = 0°, 180°$. We call these phase shift ranges as 'bad regions'. Interestingly, the 'good regions' and 'bad regions' appear alternatively. Besides, they are symmetrical with $\phi = 180°$. This property reveals that there is a high chance that we can manipulate the phase shift into one of the 'good regions'.

(iii) For different SNR settings, the BER curves and the positions of 'good regions' and 'bad regions' follow the same trend, which indicates that the positions of 'good regions' and 'bad regions' are not affected by SNR values. This presents us with a hint that we can manipulate the phase shift to achieve high decoding performance even without knowing the channel conditions (i.e., SNRs).

From the above properties, to obtain a lower BER, the phase shift ϕ should lie in 'good regions', i.e., around 90° or 270°, for any given channel conditions. This is consistent with our intuitive observation in Fig. 3.3. In fact, when $\phi = 90°$ or $\phi = 270°$, the distances between the constellation points are the largest, which is desirable for decoding.

3.4.2 A Shifting Code

So far we have known the exact best values of the phase shift ϕ for a lower BER. However, in practice, we cannot precisely manipulate the phase shift for maintaining the theoretically best values at the receiver. In fact, the phase shift is a random variable for each transmission and it follows a uniform distribution $[60, 70]$.

Increasing the Diversity Gain

Recall that the 'good regions' and 'bad regions' are alternative and symmetrical. By utilizing this property, we let SigMix transmit two copies of the signal to increase the

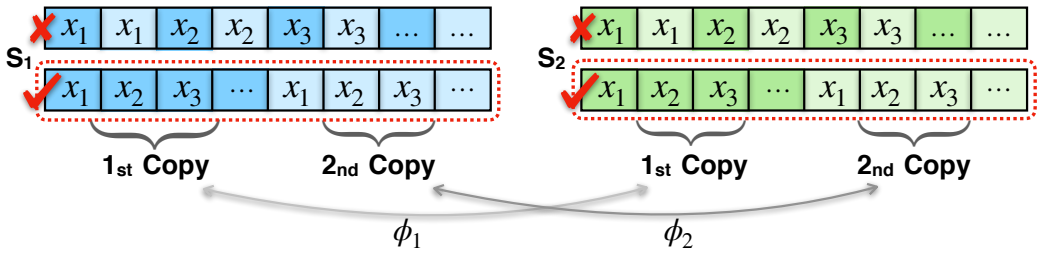


Figure 3.6: The format of the two copies. S_1 and S_2 can transmit either identical or different signals.

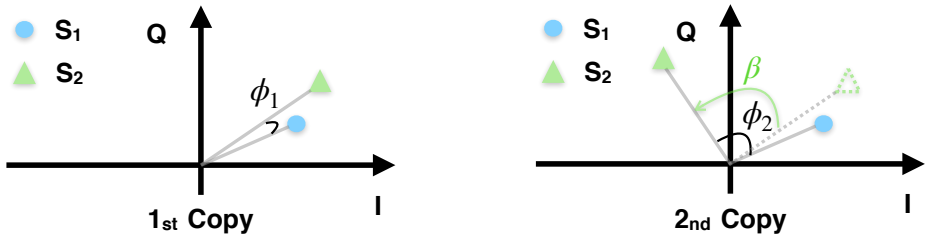


Figure 3.7: An illustration of the shifting code.

diversity gain. For simplicity, we notate the phase shift between the first copies of the transmitted signals as ϕ_1, and the phase shift between the second copies as ϕ_2. By doing so, we have a higher chance to guarantee that at least one phase shift (i.e., ϕ_1 or ϕ_2) is within the 'good regions'.

Specifically, as shown in Fig. 3.6, to avoid the deep fading in the time domain, we directly insert the copied signal at the end of the original one instead of interleaving the original samples and the copied samples. Here, the original signal refers to the first copy. For a given transmitter, the two consecutive copies are from the same hardware and experience almost identical channel condition, so two consecutive superimposed signals will have almost identical phase shift. To increase the diversity gain, we need to guarantee that $\phi_1 \neq \phi_2$. To do so, we rotate the second copy of one transmitted signal with a certain angle and keep the other copies unchanged. Fig. 3.7 illustrates the idea of the shifting code. For the first copies from S_1 and S_2, the phase shift is ϕ_1. When we rotate the second copy from S_2 with angle β, the phase shift between the second copies from S_1 and S_2 is $\phi_1 + \beta$, and we denote $\phi_2 = \phi_1 + \beta$. As long as

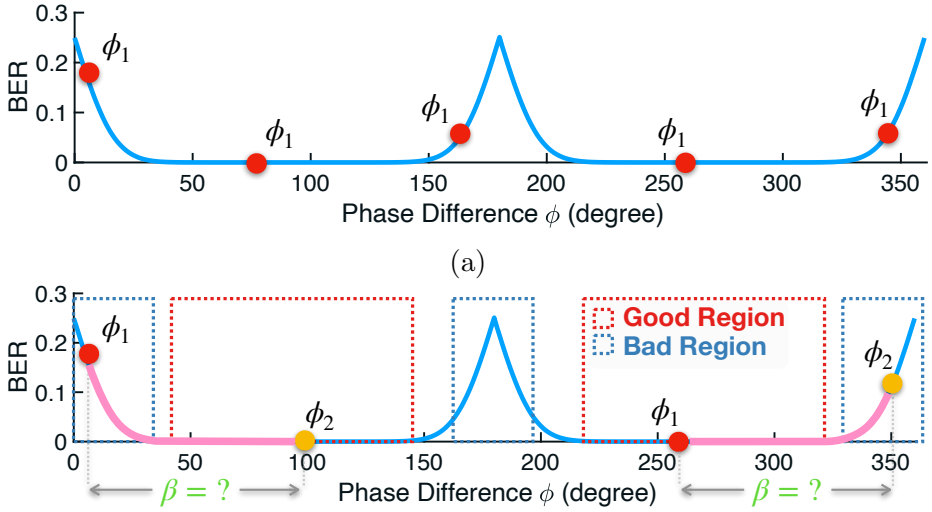


Figure 3.8: A searching scheme for the shifting angle β.

$\beta \neq 0°$, we have $\phi_1 \neq \phi_2$.

A Searching Scheme

Up to now, we have two phase shifts ϕ_1 and ϕ_2. As shown in Fig. 3.8(b), the shifting code changes the phase shift between the second copies by β. However, we have no idea of the phase shift between the first copies ϕ_1 since it follows the uniform distribution. That is to say, ϕ_1 can be located at any position of the BER curve (see Fig. 3.8(a)). Therefore, without known ϕ_1, we propose a searching scheme to find the optimal shifting angle β, so that we can guarantee that at least ϕ_1 or ϕ_2 is within the 'good regions'. Specifically, we formulate this problem as an optimization problem,

$$\begin{aligned} \min_{\beta} \quad & \{P(\phi_1), P(\phi_1 + \beta)\} \\ \text{s.t.} \quad & 0° \leq \phi_1 < 360° \\ & 0° < \beta < 360° \end{aligned} \tag{3.8}$$

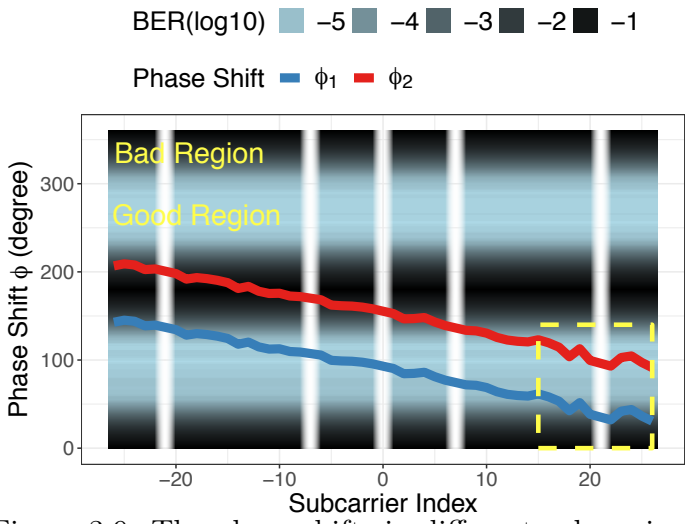


Figure 3.9: The phase shifts in different subcarriers.

where $P(.)$ is the BER given certain value of ϕ_1. As the size of solution set is very small, we search all possible shifting angles. Given ϕ_1 is within $[0°, 360°)$, let shifting angle β vary from 0° to 360° with a fixed step size, e.g., 1°. With a brute-force trial of all possible shifting angles, we can obtain the optimal values that minimize the BER, which are 90° and 270°. Without loss of generality, we set $\beta = 90°$ in our implementation.

3.4.3 An Adaptive Decoding Scheme

When the receiver receives the superimposed signal, there are two copies for each symbol. Which one should be used to decode the superimposed signal? Intuitively, the receiver can first calculate the phase shifts ϕ_1 and ϕ_2 based on the preambles, since the preambles of the transmitted signals can be made orthogonal to each other, making it easy to obtain the phase shifts separately. Then, the receiver decides which phase shift is in the 'good regions', and selects the corresponding signal copy for decoding. However, for OFDM systems, the channel behaviors of different subcarriers are different due to dynamic channel conditions and the frequency selective fading [52, 53]. Furthermore, the hardware imperfection exacerbate the differences

among subcarriers. To see this clearly, we conduct a benchmark experiment and the detailed setup is introduced in Section 3.6.1. Fig. 3.9 plots the phase shifts (ϕ_1 and ϕ_2) in different subcarriers. For some subcarriers, ϕ_1 is in 'good regions', while for other subcarriers, ϕ_2 is in 'good regions'. Furthermore, we notice that when randomly subcarriers are selected, or the same subcarrier but with different time, both ϕ_1 and ϕ_2 are not consistent. These benchmark experimental results reveal that the phase shifts not only vary with subcarriers but also with time. Therefore, using a fixed metric for all subcarriers at any given time, like prior work [34, 59], will result in poor decoding performance. In contrast, in our design, the receiver decides which phase shift is in the 'good region' for every subcarrier at any given time, and eventually selects the corresponding signal copies for decoding signals across different subcarriers. Hence, SigMix handles the signal variations very well and can achieve high decoding performance.

3.5 Practical Issues

Decoding the two-user superimposed signals demands a high synchronization accuracy, i.e., the time synchronization, the frequency synchronization and the phase synchronization [21]. However, in practice, the hardware imperfection of IoT devices will cause more challenges to achieve the required synchronization level.

Time synchronization. Recent studies on the time synchronization provide us a chance to achieve symbol-level synchronization in the time domain. For example, a 300 ns accuracy for outdoor can be achieved by GPS clocks [48] and below 200 ns accuracy for indoor can be achieved by WiFi devices [42, 71], which are much lower than the symbol duration in current OFDM systems. Importantly, these solutions are feasible for IoT devices. Therefore, in this chapter, we focus on the challenges of

frequency synchronization and phase synchronization.

Frequency synchronization and phase synchronization. Typically, losing synchronization mainly means that the signal has offsets [72]. In particular, using a single source signal as an example, three offsets exist when the oscillators of the transmitter and receiver keep drifting away from their clock at different speeds. Specifically, the carrier frequency offset (CFO) appears when downconverting signal from the carrier to the baseband, and the sampling frequency offset (SFO) happens when convert the analog signal into the digital one. Also, due to the limited processing ability of the hardware, the sampling time offset (STO) occurs when the receiver wrongly identifies the starting point of the signal. Note that CFO means losing the frequency synchronization, while SFO and STO mean losing the phase synchronization. In fact, compensating these offsets has been well studied for the single source signal [37]. However, when it comes to the superimposed signal, the traditional solutions cannot be applied any more. This is because applying a single compensation to different signal sources simultaneously will lead to a high decoding error rate.

To solve the above problem, we use the method proposed in chapter 2 that compensated the offsets by using a composite channel coefficient [17]. Specifically, the composite channel coefficient comes from not only the varying channel condition, but also the offsets. By using orthogonal preambles for the signal sources, the receiver can obtain the composite channel coefficient for each signal source. This method can be directly applied to SigMix as the compensation of these offsets is not affected by the rotation code or the adaptive decoding scheme.

Figure 3.10: **Hardware in SigMix.**

3.6 Evaluation

3.6.1 Implementation

Hardware-wise. We implement SigMix on a software-defined radio platform which consists of three Universal Software Radio Peripheral (USRPs) embedded with XCVR2450 daughter boards. Specifically, two USRP N210s connect to a laptop through a Gigabit Ethernet router as two transmitters, while one USRP N200 connects to another laptop through Ethernet cable as the receiver. For the time synchronization, each USRP that acts as a transmitter is connected to a central clock (i.e., NI CDA-2990) via SMA cables. We employ HG2458RD-SM omni-directional antenna with 3 dBi gain for each USRP. To be easily extended to other standard protocols, SigMix follows IEEE 802.11p standard, i.e. the 5.8 GHz carrier frequency and 10 MHz bandwidth. We control the SNR of the received signal around 10 dB to 20 dB, which is a common range in practice [73]. The hardware setup of SigMix is shown in Fig. 3.10.

Software-wise. SigMix is built upon a recent Wi-Fi project programmed in GNU-radio [50]. Specifically, for SigMix transmitters, we implement the proposed shifting code as presented in Section 3.4.2. Each transmitter transmits two copies of

the signal with a total length of 1500 bytes payload and a preamble, which follows the standard length of Wi-Fi packets. For SigMix receiver, we implement the adaptive decoding scheme as described in Section 3.4.3 and compensate CFO, STO and SFO as introduced in Section 3.5. To focus on our design, we remove other schemes along with the Wi-Fi protocol for simplicity, such as scrambling, interleaving, channel coding, etc. The program is running on laptops operating on Ubuntu 16.04, and each laptop processes the data streams coming from the USRP.

Metrics. We use the following four metrics for the evaluation: (i) *Bit Error Rate (BER):* the percentage of bits in error; here, the BER refers to the raw BER without considering channel coding. (ii) *BER Decline Ratio (BDR):* defined as the ratio of BER in the existing scheme to BER in SigMix; (iii) *Throughput gain:* the ratio of throughput in SigMix to throughput in the existing scheme, while keeping the traffic pattern constant; (iv) *Delay distribution:* the number of retransmissions until the packet is successfully received. Note that both BER and BDR are related to the reliability, while throughput gain and delay distribution refer to the efficiency and delay of the system performance, respectively.

We evaluate SigMix in comparison with the state-of-the-art PNC and NOMA schemes.

- **Traditional PNC (T-PNC).** We denote the traditional physical-layer network coding scheme as T-PNC [22, 23]. To be fair, we enable T-PNC to decode the two-user superimposed signal at the receiver and leave the offset compensation unchanged. T-PNC only compensates an average CFO, while it does not consider other offsets or the shifting code. Consequently, it may not perform well in practical scenarios.

- **PhyCode.** PhyCode [17] is a new type of NOMA without power control, where the superimposed signal can be decoded directly at the receiver. PhyCode

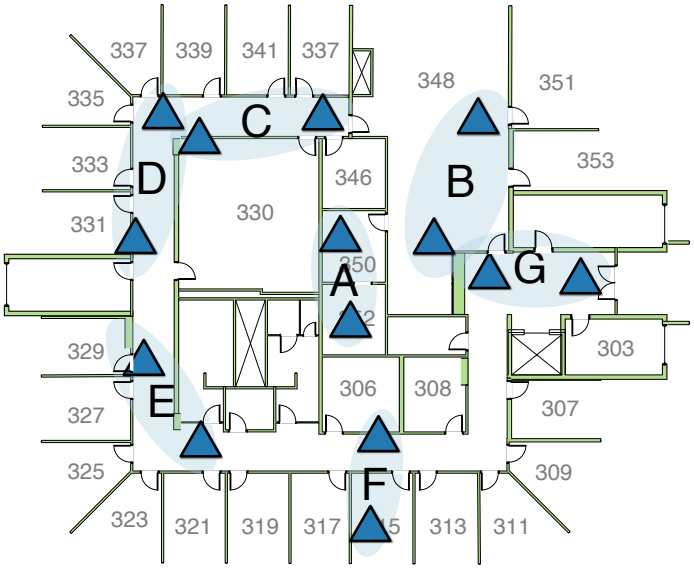


Figure 3.11: The deployment layout.

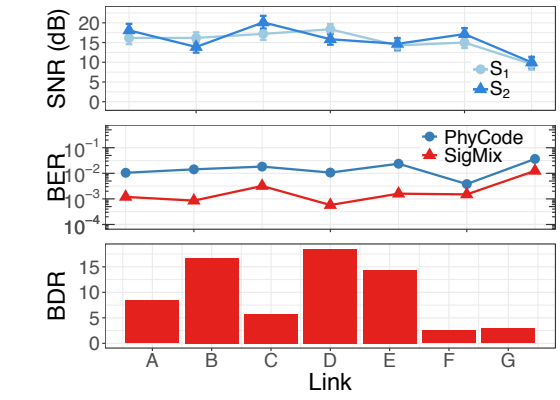
compensates CFO, SFO and STO, but it assumes the constellation points are distinguishable. So, it decodes the superimposed signal without considering the shifting code to increase the diversity gain.

Methodology: Our goal is to evaluate the performance of SigMix for decoding the two-user superimposed signal across different scenarios. First, we evaluate SigMix in indoor dynamic environments with rich multipath effects. Specifically, we conduct the experiments in an office building and we randomly select 7 regions including both line-of-sight and non-line-of-sight scenarios as shown in Fig. 3.11. For each region, we deploy two transmitters and one receiver. The distance between transmitters and receiver varies from 1 m to 10 m. To guarantee independent channels, we keep the minimum distance between two transmitters as 20 cm which is much larger than the signal wavelength (i.e., 5.2 cm for the 5.8 GHz wireless signal). To mimic IoT traffic with periodic data, the transmitter transmits one packet per second. For each region, the experiment lasts for one hour (i.e., 3600 rounds of concurrent transmissions) and the total experiment lasts for 7 hours. During the experiments, people in the building either sit in their desk space or walk around the corridors as usual, which

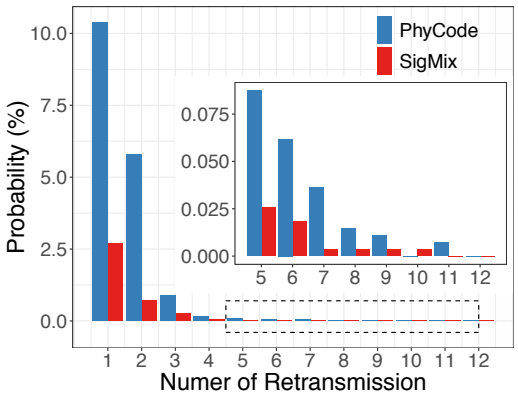
Table 3.2: Comparison of BER.

	Scheme		
Metric	T-PNC	PhyCode	**SigMix**
BER	3.82×10^{-1}	1.43×10^{-2}	1.05×10^{-3}

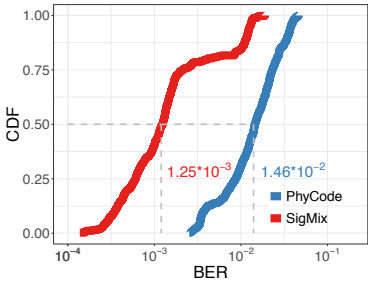
contributes to a dynamic indoor environment. Second, to focus on the influence of SNR and avoid the environment noise, we deploy SigMix in an anechoic chamber. The SNR is adjusted by changing the transmission power. For each SNR setting, the experiment lasts for one hour. Lastly, we evaluate SigMix under outdoor mobile scenarios. Specifically, we deploy the transmitters and the receiver in two vehicles. The two vehicles are driven along a ring road in a city.



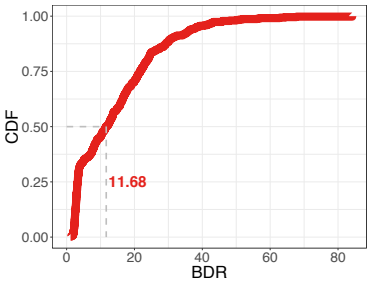
(a) BER analysis for different links



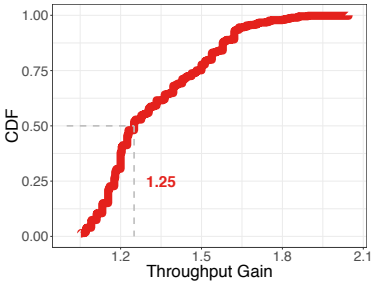
(b) Number of retransmissions



(c) BER comparison



(d) BDR



(e) Throughput gain

Figure 3.12: Overall performance.

3.6.2 Overall Performance

We compare the decoding performance of T-PNC, PhyCode and the proposed SigMix under the indoor deployment.

BER and BDR. As shown in Table 3.2, we summarize the average BER from all 7 regions for each scheme. Obviously, the BER of SigMix is significantly lower compared with T-PNC and PhyCode. Note that the BER of T-PNC is much higher since it does not compensate CFO, SFO and STO very well. So, T-PNC cannot perform well in IoT systems with low-cost devices that have high and heterogeneous offsets. In the following comparisons, we only show the results of PhyCode and SigMix.

Fig. 3.12(c) plots the CDF of BER from all 7 regions. As we can see, SigMix has a median BER of 1.25×10^{-3}, outperforming PhyCode by 11.68×. In other words, SigMix achieves a one-order lower BER. This is because SigMix utilizes the shifting code and adaptive decoding scheme, making it more robust against dynamic channel conditions and hardware imperfection. To see the improvement clearly, Fig. 3.12(d) shows the CDF of the BDR. Furthermore, Fig. 3.12(a) illustrates the BER analysis of each deployed region. As we can see, the SNRs of the two signals, S_1 and S_2, vary in different deployed regions due to the different environmental noises and layouts. In all 7 deployed regions, the BER of SigMix outperforms PhyCode and the corresponding BDR is 9.91 on average. It is worth noting that in region G, the BER of SigMix slightly increases and close to that of PhyCode. This is because there are more environmental noises in region G, making the cluster of a constellation point more sparse.

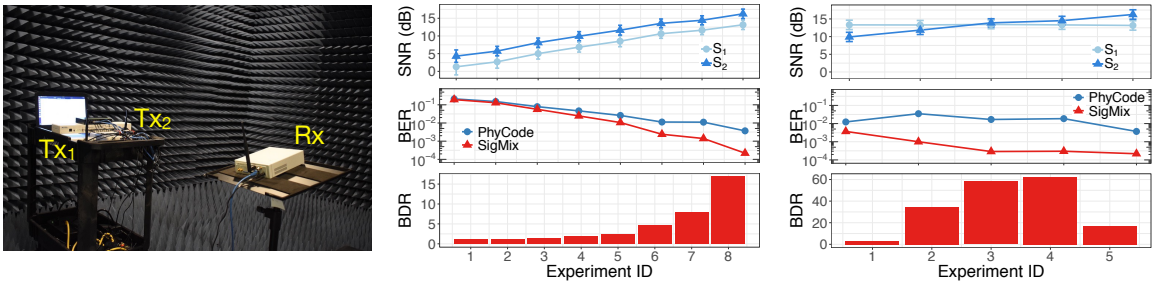
Throughput gain. We perform the trace-driven simulations to investigate the throughput gain of SigMix. Specifically, we record the bit error information of 25000 packets from USRP testbed for each scheme. For each transmitter, we emulate 25000

packets applying the CRC-32 algorithm and convolutional codes with 1/2 code rates (common settings in Wi-Fi) in Matlab. Then we use the recorded bit error information as the physical-layer decoding results. By using the Viterbi decoding and checking the CRC, we can obtain the packet loss results. The throughput gain is calculated according to the packet loss results of SigMix and PhyCode. As shown in Fig. 3.12(e), the median throughput gain of SigMix is 1.25, which indicates a high efficiency of the system.

Number of retransmissions. Similar to the throughput gain, we perform the trace-driven simulations to compare the delay distribution of SigMix and PhyCode. We calculate the number of retransmissions of both schemes based on their packet loss results. Fig. 3.12(b) shows the number of retransmissions for each lost packet till a success. As we can see, for all numbers of retransmissions, the probability of SigMix is lower than that of PhyCode, implying a lower delay of the transmission. Specifically, the probability of retransmitting once and twice of SigMix is 3.84× and 7.98× lower than that of PhyCode, respectively. Note that the BER determines both the throughput gain and the number of retransmissions, and thus, we only show the BER analysis in the following comparisons.

3.6.3 Impact of the Signal SNR

We evaluate the influence of SNR on the decoding performance. To avoid environmental interference, we conduct the experiment in the anechoic chamber shown in Fig. 3.13(a). We manually tune the SNR of the two transmitters from 5 dB to 15 dB. As shown in Fig. 3.13(b), SigMix outperforms PhyCode under different SNRs. The BER of both SigMix and PhyCode is reduced as the SNR increases. One thing worth noting is that when the SNR is lower, the BDR becomes smaller as well. The reason is that when the SNR is under a lower level, the environmental noise plays



(a) Setup in the chamber (b) Impact of SNR (c) Impact of SNR ratio

Figure 3.13: Performance under different SNRs and SNR ratios.

a vital role in distorting the signal, making the cluster of a constellation point more sparse. Overall, with the typical SNR range [31] (e.g., 10 dB to 20 dB), the BDR of our scheme is high.

3.6.4 Impact of the Signal SNR Ratio

To further evaluate the impact of the SNR ratio of two transmitted signals on the decoding performance, we keep the SNR of one transmitter stable and change the other one gradually. We again conduct the experiment in the anechoic chamber. We show the comparison results in Fig. 3.13(c). SigMix outperforms PhyCode in any given SNR ratio. More importantly, the BER is not changed obviously by the SNR ratio, which reveals that SigMix is robust to the SNR ratio. These results further demonstrate that unlike traditional NOMA techniques (i.e., SIC), SigMix can perform well without power control of the transmitters.

3.6.5 Impact of the Packet Length

To evaluate the effect of packet length on the decoding performance, we keep the SNR of two transmitters stable and vary the packet length from 10 bytes to 1500 bytes. This experiment was conducted in an ordinary office. From Fig. 3.14, we observe that SigMix always outperforms PhyCode for any given packet length. The

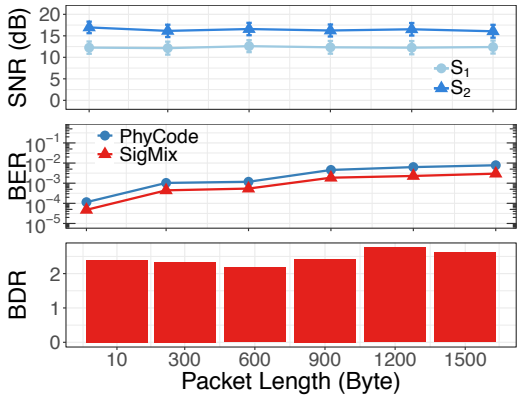


Figure 3.14: Performance of different packet lengths.

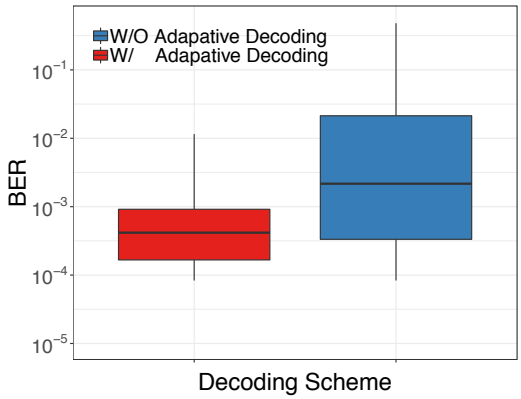


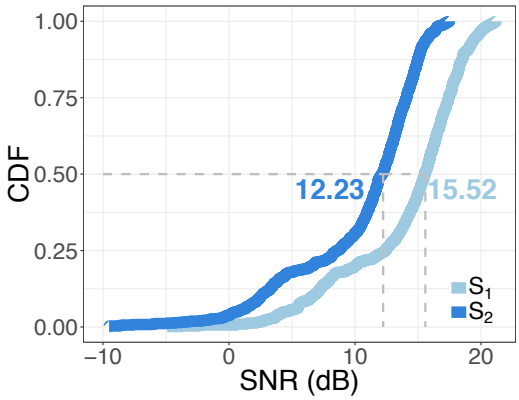
Figure 3.15: W/ and W/O adaptive decoding scheme.

BER of SigMix becomes much smaller when we reduce the packet length (e.g., the BER is 10^{-5} with a packet length of 10 bytes). This is because, with the growth of the packet length, the damage of residual offsets becomes more obvious. But even with the longest packet length in our experiment (i.e., following the Wi-Fi standard), SigMix can still achieve a better BDR, and a considerable low BER result, which verifies that SigMix compensates CFO, SFO and STO very well.

3.6.6 Impact of the Subcarriers' Difference

To validate the effectiveness of the proposed adaptive decoding scheme, we conducted the experiment in an ordinary office with rich multipath. As we can see from Fig. 3.15, the average BER is higher and its variation is larger when without applying the

(a) Setup for vehicles (b) Road experiments



(c) SNR

Figure 3.16: Outdoor Setups.

adaptive decoding scheme, even if we compensate the offsets well and apply the shifting code. In contrast, after applying the adaptive decoding scheme, the BER of SigMix becomes lower, which demonstrates that SigMix can precisely respond to the subcarriers' difference to achieve high decoding performance.

3.6.7 The Outdoor Mobile Scenario

To evaluate the mobility on the decoding performance, we deploy the transmitters and receiver in two vehicles and the setup is shown in Fig. 3.16(a). During the road experiment, the two vehicles are continuously driven along a ring road for 10 rounds (see Fig. 3.16(b)) with a maximum speed at 40 km/h. As shown in Fig. 3.16(c),

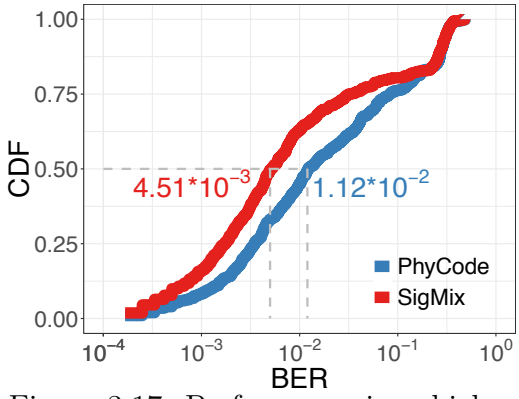


Figure 3.17: Performance in vehicles.

the SNRs of the two transmitters vary from -10 dB to 20 dB due to the mobility effect. The median SNRs of the two transmitters are 15.52 dB and 13.23 dB, respectively. Fig. 3.17 illustrates the comparison of the BER. SigMix has a median BER of 4.51×10^{-3} and a median BDR of 2.48, outperforming PhyCode. Overall, SigMix can perform well in the presence of mobility.

3.7 Conclusion

This chapter presents SigMix, a practical solution to decode the two-user superimposed signal in the presence of dynamic channel conditions and hardware imperfection in wireless communication systems. We demonstrate the effectiveness of our design through the implementation on a software-defined radio platform. SigMix is promising towards addressing the grand challenge of wireless applications, such as the spectrum efficiency, delay and reliability, achieving a 11.68× improvement in the median BER compared to the state-of-the-art.

Chapter 4

I-Talk: Decoding Two-user Superimposed Signal in Mobile Environments

4.1 Introduction

4.1.1 Background

The proposed schemes in both chapter 2 and chapter 3 can decode the superimposed signal in modulation-level and then without any power control, however, they all base on static environments, making them less compatible to general IoT scenarios that include mobility. Although the implementation in chapter 3 reveal the performance of SigMix in mobile scenarios, both of the transmitters are located in the same vehicle during the experiment, which does not cover the full features of mobility and may limits the application scenarios.

After a thoroughly analysis, we notice that along with the ever-growing opportunity brought by the mobility, the dynamic channel conditions from mobile devices are

inevitable and it can cause severe signal variations upon the received superimposed signals. Moreover, the hardware imperfection becomes more unpredictable under mobile scenarios. These newly introduced challenges by mobility all contribute to an unreliable decoding performance. Therefore, a reliable superimposed signal decoding scheme that covers mobile IoT is highly desirable. In this chapter, we propose I-Talk, a new NOMA technology that can decode the superimposed signals for mobile environments. I-Talk aims to achieve high reliability in the presence of mobile channel conditions and hardware imperfections. It is non-trivial to realize I-Talk due to the following practical challenges.

4.1.2 Challenges and Proposed Solutions

In this chapter, we propose I-Talk, a new NOMA technology that can decode the superimposed signals from two concurrent transmitters in the modulation level without power control. I-Talk aims to achieve high reliability in the presence of hardware imperfections and mobile channel conditions. However, it is non-trivial to realize I-Talk due to the following practical challenges.

- First, to decode the superimposed signal in the modulation level, symbol-level synchronization is required in both the time and frequency domains, i.e., signals from the two users should arrive at the receiver simultaneously and on the same frequency. However, synchronizing IoT devices in the presence of hardware imperfections and mobile channel conditions is non-trivial. This is mainly because the hardware imperfections can cause unpredictable signal offsets in the time, frequency, and phase domains. Moreover, the mobility together with the Doppler effect from the mobile IoT devices can cause extra signal offsets compared to static scenarios, which deteriorates the synchronization accuracy.
- Second, decoding the superimposed signal from the modulation level is intrinsically

difficult in matching the location of the received signal symbol to its representative code precisely. In particular, both the hardware imperfections and mobility can cause severe signal variations, so that some codes in the constellation map are too close to each other and cannot be distinguished clearly, resulting in an unreliable decoding performance.

To deal with the above challenges, we first study the signal offsets caused by the hardware imperfection and mobility, i.e., the Carrier Frequency Offset (CFO), the Sampling Frequency Offset (SFO) and the Sampling Time Offset (STO), and design a synthesis channel coefficient to represent all these offsets. By doing so, we can trace all the offsets and then eliminate the side effects of these offsets, providing a stable synchronization performance. Second, by exploiting the complementary property of the representative codes and the subcarriers' differences in complicated channel environments as well as the substantial diversity gain of transmitting two copies of the signal, we propose a diversity transmission and smart combining scheme to achieve high reliable decoding performance.

The main contributions of this chapter are as follows:

- We derive a synthesis channel and based on this design, I-Talk can achieve near-perfect synchronization for IoT devices in the presence of the hardware imperfections and mobile channel conditions.
- We propose a reliable superimposed signal decoding approach for NOMA without any power control. Moreover, with the proposed diversity transmission and smart combining scheme, I-Talk can achieve high reliability regardless of the hardware imperfections and mobility in both static and mobile systems.
- We implement I-Talk on a software-defined radio platform and evaluate its performance across various scenarios. Our extensive experimental results demonstrate that I-Talk achieves a one-order lower bit-error-rate and a 1.47× higher throughput

gain in the mobile scenario, compared with PhyCode.

4.1.3 Related Work

Non-Orthogonal Multiple Access (NOMA). NOMA technology aims to better utilize the spectrum by enabling multiple users to transmit their signals using the same wireless spectrum, i.e., at the same time and on the same frequency [9, 10, 17]. To implement NOMA in practice, it requires a considerable large signal power gap between different users, so that each signal can be separated in the power domain [16,54]. Although promising, when it comes to heterogeneous and often low-cost IoT devices, the signal power gap cannot be maintained due to the complexity and hardware imperfection, making the existing NOMA technologies infeasible to many IoT systems, which motivated this work.

Physical-layer Network Coding (PNC) and Analog Network Coding (ANC). PNC [21–23, 33, 34, 74] and its counterpart solution ANC [31] have drawn many attentions recently in the superimposed signal decoding for relay networks. These solutions can be adopted for relay system where the receiver knows one of the two signals superimposed, and then the unknown signal can be decoded by removing the known signal from the superimposed signal. PNC decodes the superimposed signal in modulation-level from the constellation map, while ANC decodes in analog-level by canceling out the known signal. Both cannot be extended to multiple access scenarios. Furthermore, existing PNC and ANC are dedicated to static scenarios, and may encounter a performance degradation in the presence of mobility.

Decoding superimposed signals for low data rate technologies. Many recent work focus on decoding superimposed signals in low data-rate technologies, such as RFID [45], LoRa [75], ZigBee [36] by leveraging their spread-spectrum feature or over-sampling long symbols. But these solutions cannot be utilized by other general

data-rate systems, such as Wi-Fi. I-Talk focuses on OFDM, which is feasible for many technologies with a general date rate, e.g., IEEE 802.11.

Space-time codes. The space-time code is applied in the literature to obtain a diversity gain, such as the Alamouti Code [76–78]. But it requires multi-antenna to transmit the coded symbol, which may not always be feasible or desirable for IoT devices. We notice that some work studied the use of space-time code for concurrent users in theory [79]. However, the work requires perfect knowledge of the Channel State Information (CSI) before each transmission and neglects the influence of the hardware noises. As a result, a varying channel in practice may lead to its performance degradation. In I-Talk, we aim to design a practical NOMA scheme for IoT, and therefore we make no assumption on the CSI and work well for a single-antenna system. On the other hand, when the IoT devices do have multiple antennas, it is possible to combine both space-time codes and I-Talk technologies for a better performance, as they are orthogonal.

4.2 I-Talk Overview

I-Talk is designed to decode the superimposed signal for IoT systems, aiming to achieve high reliability in the presence of mobility and hardware imperfections. I-Talk first addresses several critical issues in practice, including the time, frequency and phase synchronizations in the mobile scenario with hardware imperfections. Next, I-Talk investigates the underlying reasons of the decoding error for superimposed signals, and reveals a complementary property of the decoding error in the constellation map. Based on that, I-Talk designs a diversity transmission and smart combining decoding scheme for superimposed signals to guarantee high reliability. We elaborate on the above components and provide the technical details in the next few sections.

4.3 Practical Challenges in Decoding Two-user Superimposed Signals

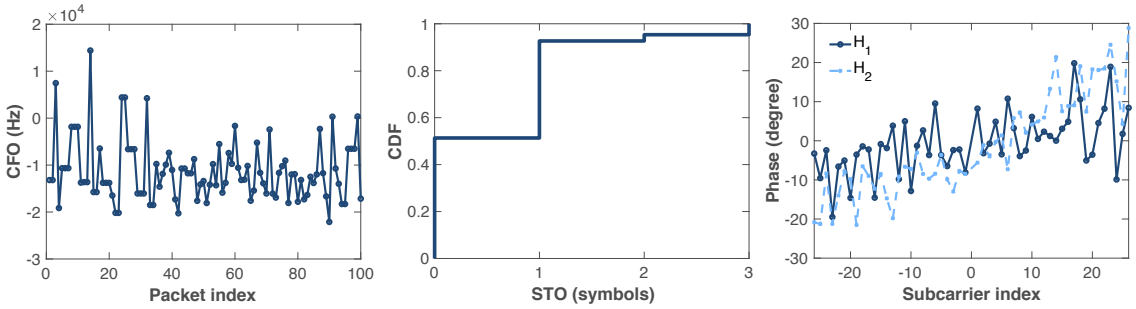
I-Talk is proposed to decode the two-user superimposed signal in modulation-level. To achieve that, the system requires an accurate estimation of the signal variation before decoding. However, in practice, the hardware imperfection and mobility will decrease the estimation accuracy significantly. In particular for a superimposed signals, the decrease of the estimation accuracy would result in a much higher decoding error. To address that, we first illustrate the preliminary of the superimposed signal. Second, we study the key factors in affecting the estimation accuracy. Last, we propose a solution to improve the the estimation accuracy in the presence of the mentioned practical challenges.

4.3.1 Preliminary

Considering the basic case of two concurrent transmitters, the received superimposed signal Y can be represented as

$$Y = H_1X_1 + H_2X_2 + N, \tag{4.1}$$

in the frequency domain. Here, X_1 and X_2 are the transmitted signals, and H_1 and H_2 are the corresponding channels between the two transmitters and receiver, respectively. N refers to the white Gaussian noise. Eq. 4.1 is valid in an ideal scenario where the channel is the only factor to cause signal variations, without considering the hardware imperfection. In practice, precisely measuring the channel is the key to successful decoding. Indeed, decoding the superimposed signal demands a high synchronization accuracy in the time, frequency and phase domains [21].



(a) CFO. (b) STO. (c) SFO.

Figure 4.1: Signal offsets in the mobile scenario with Doppler effect.

First, symbol-level time synchronization is needed [21–23] for decoding the superimposed signal. Specifically, by using IEEE 802.11p as an example, the time synchronization error should be no larger than 2 μs. Recent work has shown that this level of synchronization is achievable on off-the-shelf devices, e.g., a 300 ns accuracy for outdoor scenarios can be achieved by GPS clocks [48]. For simplicity, we use a commercial GPS clock to achieve the required time synchronization. More details are presented in Sec. 4.6. Second, besides the time synchronization, due to the highly mobile channel conditions and hardware imperfections, various signal offsets deteriorate the frequency and phase synchronizations, making decoding superimposed signals non-trivial. In the following, we focus on these unsolved synchronization problems, and assume the time synchronization have been well achieved.

4.3.2 Vital Factors of the Signal Variation Estimation

For a single pair of transmitter and receiver, decoding a signal requests to deal with three offsets at the receiver [72, 80]: the Carrier Frequency Offset (CFO), the Sampling Frequency Offset (SFO) and the Sampling Time Offset (STO). Generally, CFO and SFO exist because of the differences between the internal oscillators, and STO is caused by signal detection failures [37]. To correct these offsets, a precise offsets measurement is needed at the receiver. But the mobile channel condition and hardware

imperfections make the precise offsets measurement hardly achievable, even using the GPS clock [48, 50]. Moreover, concurrent transmitters cause more offsets and make channel conditions more complicated.

Signal offsets. To see these offsets clearly, we conduct a benchmark experiment with two USRP devices as a transmitter and a receiver deployed in two vehicles, respectively. Each device connected to a GPS clock for synchronization. The signal is operating at 5.9 GHz carrier frequency with 10 MHz bandwidth, and the synchronization error under ideal condition is around ± 50 ns. The more detailed setup is introduced in Sec. 3.6.1. Specifically, Fig. 4.1(a) shows that CFO has a large fluctuation up to 3.5 KHz in the presence of the Doppler effect and the hardware imperfection. Fig. 4.1(b) plots the mismatch in terms of symbols for STO. This mismatch would not directly cause a synchronization failure, but it would lead to more phase offsets, making the signal hard to be decoded successfully. Note that the symbol and the sample are used interchangeably as no oversampling is assumed in our system. The probability of a successful match is only 50% due to mobility. Fig. 4.1(c) shows that there is a large phase variation for SFO caused by mobility and hardware imperfections. Note that CFO affects the frequency synchronization, while SFO and STO damage the phase synchronization. Therefore, to decode the superimposed signal, we need to track and react to these offsets using precise channel measurements.

Preamble design. To ensure precise packet detection and channel measurement, each packet should start with a preamble, which is also critical for handling the three offsets. In IEEE 802.11p, a preamble containing a Short Training Sequence (STS) for the packet detection and a Long Training Sequence (LTS) for measuring the channel condition and the three offsets. However, LTS can be easily be collided under concurrent transmissions. To understand this more clearly, we start from introducing the procedure of the use of LTS. Specifically, once a signal is detected by

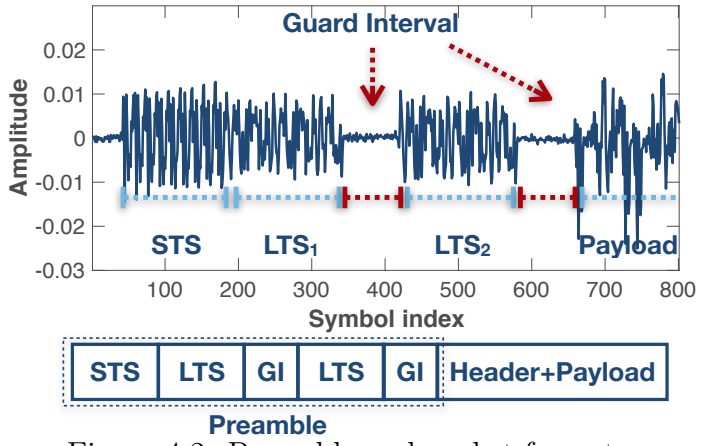


Figure 4.2: Preamble and packet formats.

a receiver via STS, the receiver will apply an FFT operation with an FFT window to capture LTS. Because LTS is predefined in advance, the comparison between the received LTS and the transmitted LTS would reveal the signal variations that caused by the channel condition and the offset. In a single transmitter system, it is straightforward to capture LTS with the FFT window. But for a system with two concurrent transmitters, especially with mobility, different LTS may collide with each other due to the propagation delay and hardware processing delay, and then the FFT window would later capture a mixed LTS from two transmitters. As a result, the receiver cannot distinguish the signal variations caused by the interruption of other LTS from the variations caused by the channel condition and the offsets, making the following procedure unable to react accurately. To address this problem, we insert two extra NULL symbols as the guard intervals shown in Fig. 4.2 to enlarge the distance of two transmitted signals in the time domain. By doing so, although LTS from different transmitters may have time-domain shifts caused by the mentioned delays, the guard intervals provide a robust range to avoid the collision of LTS in practice. The newly added NULL symbols are sufficient for addressing the LTS collision problem and the additional overhead is negligible as detailed in Sec. 4.5. Note that during a packet transmission time, e.g., 1 ms, the channel distance and propagation delay changes of

due to user mobility (in nano-second level) are negligible, which will not affect I-Talk performance. In the following design of I-Talk, we will use the proposed preamble as our default settings.

In our design, the performing of the decoding relies on an aligned preamble between two users and the allocated pilot subcarriers should be known in advance. Note that given the existing feedback used in infrastructure-based IoT systems, such as 3GPP, the additional bits to notify the settings of I-Talk in the feedback message are negligible.

4.3.3 Synthesis Channel Estimation

Next, we describe the mathematical representation of the superimposed signal at a receiver. For a received sample $r(t_n)$, it contains a summation of two signals, s_1 and s_2, and the white Gaussian noise n_0. Due to the offsets, more changes will be reflected on the received signal beyond the channel response. Specifically, Δf_i causes CFO, and $n_{i\epsilon}$ causes STO, and then the received superimposed signal can be represented as

$$\begin{aligned} r(t_n) = e^{j2\pi\Delta f_1 n T_1} \sum_i h_{1i}(nT_1')(s_1(n-n_{1\epsilon})T_1' - \tau_{1i}) \\ + e^{j2\pi\Delta f_2 n T_2} \sum_i h_{2i}(nT_2')(s_2(n-n_{2\epsilon})T_2' - \tau_{2i}) + n_0, \end{aligned} \tag{4.2}$$

where T_i' and T_i are the sampling time at the receiver and the transmitter (i.e., SFO), respectively [37], and h_i and τ_i are the channel impulse response and the delay, respectively.

Although CFO, STO and SFO are caused by different reasons, all of them induce a phase shift to the signal. These phase shifts can be described as

$$\theta_n^{k,CFO} = 2\pi\Delta f n T, \tag{4.3}$$

$$\theta_n^{k,STO} = 2\pi k n_\epsilon / N_d, \tag{4.4}$$

$$\theta_n^{k,SFO} = 2\pi k \gamma (N_d + L)/N_d, \tag{4.5}$$

where n refers to the n-th sample of the signal sequence and k is the k-th subcarrier index of one OFDM symbol. $\gamma = (T - T')/T$ is defined as the sampling time error ratio, and L the length of the Cyclic Prefix (CP), and N_d the length of the data part in every OFDM symbol. Here, we can use CFO error ratio $\epsilon = \Delta f / f$ to infer SFO error ratio γ. We assign two pilot samples for each transmitter to keep tracking the varying offsets in every symbol. Overall, the total phase shift can be written as

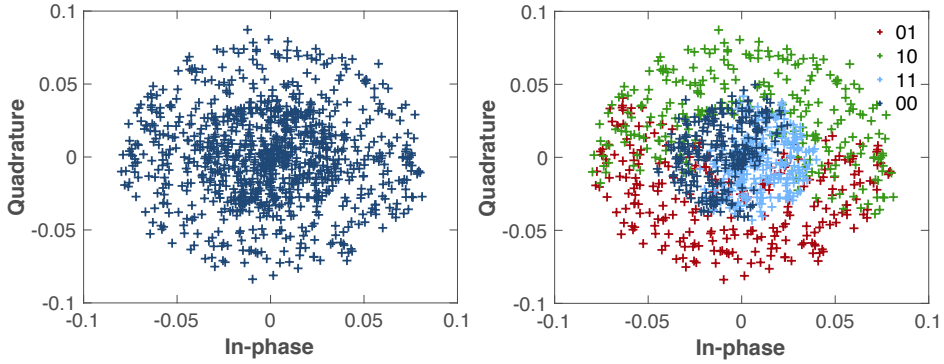
$$\theta_n^k = \theta_n^{k,CFO} + \theta_n^{k,STO} + \theta_n^{k,SFO}. \tag{4.6}$$

One thing worth noting is that for a single transmitter, the offsets can be measured first and compensated later. However, for a superimposed signal, compensating the offset for one transmitter would hurt the other, since the offsets from different transmitters have different characters. To solve this problem, instead of compensating the signal, I-Talk combines the channel conditions and the offsets to define a synthesis channel $\mathcal{H}$,

$$\mathcal{H} = H e^{j\theta_n^k}, \tag{4.7}$$

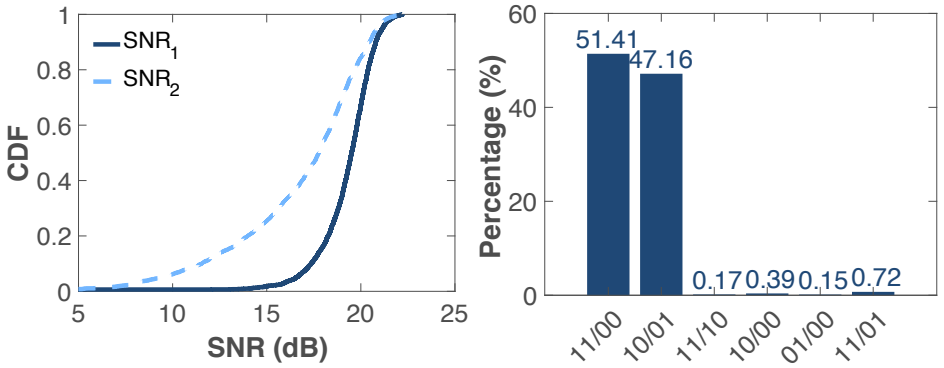
which can enable the receiver to decode the signal dynamically according to the channel condition and the offsets.

By using the BPSK modulation as an example, we have four combinations representing “11”, “00”, “10” and “01”. As shown in Fig. 4.3(a), in the presence of offsets and varying channel conditions, the signal from the concurrent transmitters do not have a constant phase relationship with each other, resulting to a varying constellation map over time. In contrast, with the proposed synthesis channel, we can precisely



(a) Without synthesis channel. (b) With synthesis channel.

Figure 4.3: The constellation map of the received signal.



(a) SNR. (b) Error distributions.

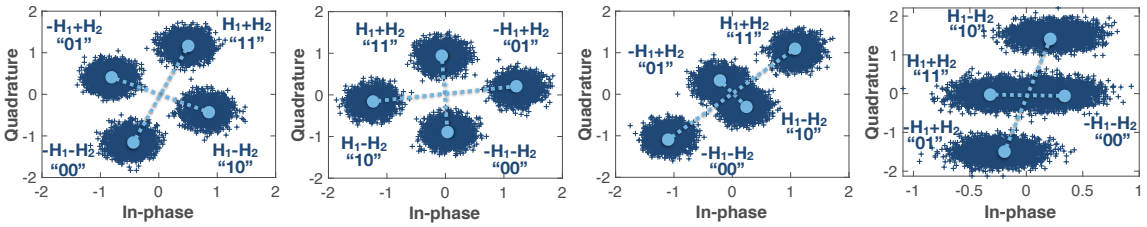
Figure 4.4: Benchmark experiment results. "11/00" represents that "11" is mistakenly decoded as "00" and vice versa.

track the varying pattern of the constellation map to identify the combinations of the superimposed signal which is shown in Fig. 4.3(b) labeled by different colors.

4.4 Understanding the Decoding Error

4.4.1 Decoding Error Analysis

To deal with the decoding error, existing solutions employ packet retransmissions [16, 17, 28, 61, 62]. However, the retransmission may introduce extra communication overhead and long delay, which is not desirable for many IoT applications. To design a reliable decoding scheme and reduce the decoding error, we start by understanding



(a) Distinguishable case 1. (b) Distinguishable case 2. (c) Indistinguishable case 1. (d) Indistinguishable case 2.

Figure 4.5: Position variation of received symbols in the constellation map.

what are the specific decoding errors and where are these errors from.

Signal variations in the IQ domain. We conducted a benchmark experiment with the same setting in Sec. 4.3.2. OFDM with the BPSK modulation is used for communications. The varying channel during the experiment is represented by the SNRs of the two transmitters shown in Fig. 4.4(a). When two BPSK modulated signals arrive at a receiver concurrently, four possible combinations exist in the IQ domain, representing "11", "10", "01" and "00". A decoding error happens when the received signal is misidentified as wrong combinations by the receiver. Fig. 4.4(b) shows the decoding error distribution, which is calculated from more than 2500 concurrently transmitted packets. Obviously, in most cases, it cannot distinguish "11" from '00" and "10" from "01" with the percentage of 51.47% and 47.16%, respectively. Furthermore, we observe that the positions of the codes may change and become too close to be distinguished (Fig. 4.5(c) and Fig. 4.5(d)) due to the signal variation caused by dynamic channel conditions and hardware imperfections, which will be exacerbated in mobile scenarios. *This observation implies that controlling the positions of the codes is the key to reduce the decoding error.* Specifically, we focus on reducing the decoding errors of "11/00" and "10/01" since these two error cases account for a dominant portion (i.e, 98.63%) of all error cases.

Impact factors of the codes position. The positions of the codes in the constellation map and the decoding error probability are mainly determined by two

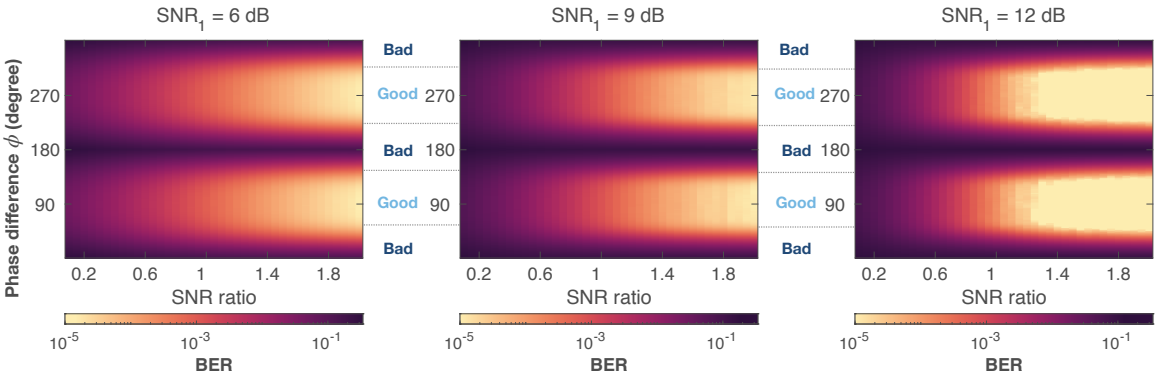
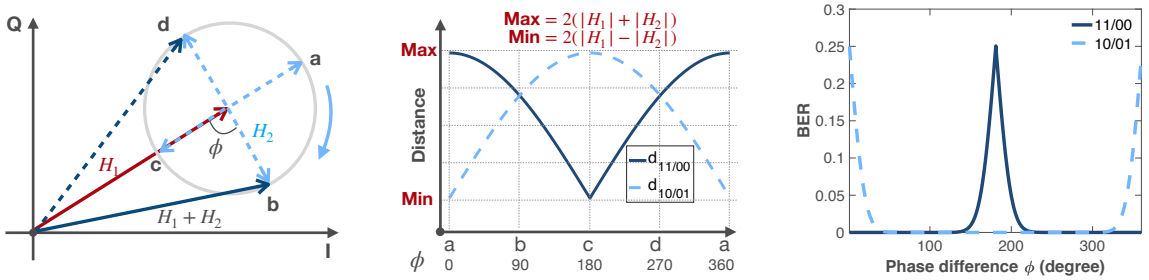


Figure 4.6: Decoding performance under different SNRs and phase differences. SNR ratio is defined as $\mathrm{SNR}_2/\mathrm{SNR}_1$.

factors: the amplitude of both signals ($|H_1|$ and $|H_2|$) and the phase difference ϕ of the two signals. More detailed quantitative analysis and BER expression can be found in [59, 66]. To further understand the effect of each factor on the decoding performance, we plot the BER heatmaps under different SNRs and phase differences in Fig. 4.6 via Monte Carlo simulations. Here, we use SNR instead of $|H|$ since SNR is a common metric for signal amplitude. We observe the following property:

Given SNR_1 and the SNR ratio, the lower BER and higher BER appear alternatively as the phase difference ϕ varies from 0° to 360°. The best choice of ϕ for the lower BER is centered around 90° or 270° (defined as 'good regions'). In contrast, the higher BER occurs when ϕ is centered around 0° or 180° (defined as 'bad regions'). Furthermore, the three BER heatmaps have the same trend as SNR_1 varies from 6 dB to 12 dB, which indicates that the positions of 'good regions' and 'bad regions' are not affected by SNR values. Note that by using this knowledge of 'good regions' and 'bad regions', a receiver can directly anticipate the possible BER range through measuring the phase difference ϕ with the preamble.



(a) (b) (c)

Figure 4.7: Conceptual illustration of the complementarity between "11/00" and "10/01": (a) H_2 rotates from point a to b,c,d, and distances $d_{11/00}$ and $d_{10/01}$ vary accordingly. (b) $d_{11/00}$ and $d_{10/01}$ fragments with respect to different positions (i.e., different phase differences ϕ). (c) Theoretical results from the BER expression.

4.4.2 Manipulating the Phase Difference

The above property motives us to manipulate the phase difference ϕ for a good decoding performance [1].

Complementarity between "11/00" and "10/01". Recall that our goal is to reduce the decoding errors of "11/00" and "10/01" cases. So, we zoom in on the two 'bad regions' (i.e., $\phi = 0°, 180°$) for "11/00" and "10/01" cases, respectively.

Apparently, as shown in Fig. 4.5(d), when "11" and "00" cannot be distinguished easily, the distance between code "11" and code "00" (denoted as $d_{11/00}$) is small. The similar situation for "10/01" case can be observed in Fig. 4.5(c). Thus, we need to manipulate the phase difference of the transmitted signals to adjust $d_{11/00}$ and $d_{10/01}$ in the constellation map.

To change the phase difference, for simplicity, we assume H_1 is fixed and then rotate H_2 from point a to b, c, d as shown in Fig. 4.7(a). Apparently, $d_{11/00}$ gets its maximum value $2(|H_1| + |H_2|)$ and minimum value $2(|H_1| - |H_2|)$ when H_1 and H_2 are in the same direction (point a) and the opposite direction (point c), respectively. In contrast, $d_{10/01}$ gets its maximum value and minimum value at point c and a,

[1] In fact, we can either control the SNR or the phase difference to achieve a lower BER. However, controlling signal amplitude requires strict power control which may be impractical for most IoT devices that are heterogeneous and often low-cost.

respectively. Furthermore, Fig. 4.7(b) shows the variation trend of $d_{11/00}$ and $d_{10/01}$ with respect to different phase differences. We obtain the following complementary property:

The two 'bad regions' for "11/00" and "10/01" are complementary to each other. Specifically, when it is difficult to distinguish "11" from "00", "10" and "01" can be distinguished easily, and vice versa. To further validate the complementary property, we conduct simulations based on the BER expression [59] and plot the BER curves in Fig. 4.7(c). We observe that the 'bad region' with $\phi = 0^\circ$ (denoted as $\text{bad}_{10/01}$) is only bad for the "10/01" case, while it is actually good for the "11/00" case. Similarly, the 'bad region' with $\phi = 180^\circ$ (denoted as $\text{bad}_{11/00}$) is bad for "11/00", but good for "10/01".

4.5 A Reliable Decoding Scheme

In this section, we propose a reliable approach for decoding the superimposed signal in the presence of the hardware imperfection and mobility.

4.5.1 Rotation Code Based Diversity Transmission

The observation mentioned above motivates us to leverage the complementary property to reduce the decoding error. Specifically, our idea is to explore the diversity gain to let each node transmit two copies of the signal within one transmission. If we can control the phase difference of the first copy pair (ϕ_A) to fall into $\text{bad}_{11/00}$ and the phase difference of the second copy pair (ϕ_B) in $\text{bad}_{10/01}$, we can perfectly eliminate these two error cases when combining the two copies for decoding. Here, we assume the two concurrent transmitters transmit two copies of their signals sequentially. At the receiver, it would receive the two superimposed signal copies from the two trans-

mitters. Hence, we denote the phase difference of the first copy as ϕ_A, and the second as ϕ_B.

However, in practice, it is challenging to precisely manipulate the phase difference into the specific 'bad regions' due to the large initial phase noise in many devices [60, 70]. Indeed, we cannot precisely control the values of ϕ_A and ϕ_B. Our solution is that, first, we need to guarantee $\phi_A \neq \phi_B$. To do so, we rotate the second copy of one transmitted signal with a certain angle α, while keeping the other copies unchanged, i.e., $\phi_B = \phi_A + \alpha$. Here, we can safely assume that the angle relationship is consistent within one packet [5]. Second, if one phase difference (ϕ_A or ϕ_B) is in one 'bad region', we need to let the other phase difference in the complementary region. To this end, we observe that the distance between two 'bad regions' is 180° and the BER curve is monotonous from one 'bad region' to its adjacent 'good region' (Fig. 4.7(c)). Therefore, we find the optimal rotating angle α to be 180°.

4.5.2 Smart Combining

At the receiver side, for each copy, the received symbols can be decoded following the minimum Euclidean distance scheme in the IQ domain. So, we have two decoding results for each received symbol, say $\hat{X}_A$ and $\hat{X}_B$. *Which one should be selected as the final decoding result or how to combine these two results?*

Decoding selection. Intuitively, we can employ Maximum Ratio Combining (MRC) to combine these two results. However, MRC performs well when the constellation map is regular and the same for all copies [6, 81]. Due to the mobile channel conditions and hardware imperfections, the constellation maps become irregular and different. So, we cannot directly use the MRC approach. Another possible solution is to assign weights to each result based on the Error Vector Magnitude (EVM). However, this solution normally requires a stable channel condition and a

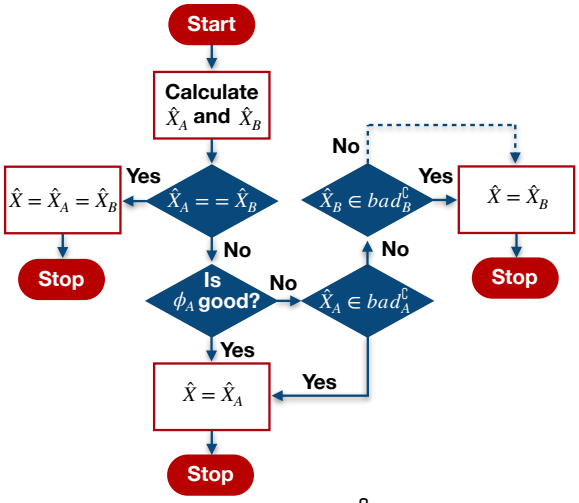


Figure 4.8: Process of decoding selection. $\text{bad}_A^{\complement}$ refers to the complementary set of the codes that this 'bad region' specified. For example, if ϕ_A is in $\text{bad}_{11/00}$, $\text{bad}_A^{\complement}$ contains "10" and "01".

high SNR [82, 83], which may not be available in mobile scenarios.

Our idea is to leverage the complementarity between "11/00" and "10/01". Specifically, if $\hat{X}_A$ equals $\hat{X}_B$, we can directly obtain the final result $\hat{X} = \hat{X}_A = \hat{X}_B$. Otherwise, if ϕ_A is in a 'good region', we can trust the first copy and use $\hat{X}_A$ as the final decoding result. But, if ϕ_A is in a 'bad region', we need to check this 'bad region' specifying $\text{bad}_{11/00}$ or $\text{bad}_{10/01}$. If $\hat{X}_A$ is not in the code set that this 'bad region' specified, we can still trust $\hat{X}_A$. Otherwise, we will trust $\hat{X}_B$. The procedure is illustrated in Fig. 4.8.

Subcarriers' diversity. For 802.11p, using OFDM, the channel behaviors of different subcarriers are different due to the hardware imperfections and the frequency selective fading [52, 53]. Furthermore, the mobile channel conditions exacerbate the differences among subcarriers. To see this clearly, we conduct a benchmark experiment. Fig. 4.9 plots ϕ of one signal copy pair within one packet for different subcarriers. Clearly, the values of ϕ vary among subcarriers. In some subcarriers, ϕ is around 90° or 270°, which is in a 'good region'; in other subcarriers, ϕ is around 0° or 180°, which is in the 'bad region' for "10/01" or "11/00", respectively. Fig. 4.10 plots ϕ

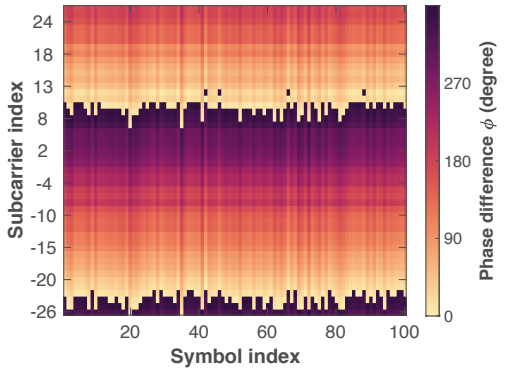


Figure 4.9: Subcarriers' diversity within a packet.

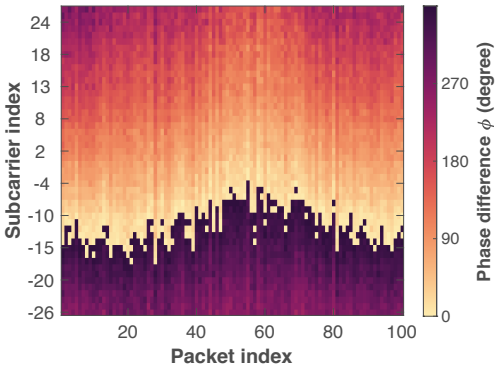


Figure 4.10: Subcarriers' diversity among packets.

Algorithm 1: Smart Combining.

Result: The set of final decoding results, $\hat{X} = \{\hat{X}^k_{[i]}\}$.
for $i = 1$ **to** number_of_symbols **do**
  for $k = 1$ **to** number_of_subcarriers **do**
    if $(\hat{X}^k_{A[i]} == \hat{X}^k_{B[i]})$ *or* $(\phi^k_{A[i]} \in R_{good})$ *or* $(\hat{X}^k_{A[i]} \in bad^{\mathfrak{C}}_{A[i]^k})$ **then**
      $\hat{X}^k_{[i]} = \hat{X}^k_{A[i]}$;
    else
      $\hat{X}^k_{[i]} = \hat{X}^k_{B[i]}$;
    end
  end
end

for different subcarriers under different time index (represented by the packet index). Apparently, even for the same subcarrier, ϕ is not stable and varies with time.

The above experimental results reveal that the phase difference not only varies with subcarriers but also with time. Therefore, we need to apply the above decoding selection approach for every subcarrier at any given time. For simplicity, we summarize the smart combining in Algorithm 1. I-Talk system can keep the packet-level structure intact so our design can be readily adopted together with the existing error control mechanisms, such as using ACK and retransmission. After I-Talk decodes the superimposed signal from two users, the receiver will check the CRC of both the received packets. Next, at the ACK time slot, the receiver will broadcast the

packet sequences that it has successfully received, and then the transmitter will decide whether to transmit new packets or retransmit the previous one. This decision is independent of the other transmitter, which makes I-Talk easy to implement.

4.5.3 Analysis

The orthogonal preamble. In the preamble design, the two LTS from each concurrent transmitter are orthogonal to each other in the time domain, which only introduces a small extra overhead. For example, for an 802.11p packet with 1460 bytes, and 6 Mbps transmission rate, the extra overhead is only 2.1% for two concurrent transmitters.

The overhead of the smart combining. I-Talk requires a traversal procedure of the packets to complete the smart combining. As the procedure can be piggybacked to the signal decoding component, the overhead is linearly increased and limited by the packet length. For the transmission part, at first glance, our scheme seems to introduce an extra overhead as we transit two repetitive symbols within in one packet, and therefore although we enable two users to transmit their packet at the same time, the overall throughput may equal to the back-to-back transmission of two users. In practice, our scheme can outperform back-to-back transmission, and we explain this in three-folds.

First, current IoT wireless communications often experience complicated fading channels, so to conquer this challenge, many IoT standards and protocols [84–87] require users to blindly transmit their packets multiple times so that at least one of the packets can be correctly received. For example, the 3GPP standard requires the blind retransmission of multiple times for IoT/machine-to-machine transmissions given different SNR [85]. Under this design principle, we actually need to compare our scheme with the back-to-back transmission that includes a blind retransmission,

and I-Talk can achieve a much higher throughput.

Second, our solution zooms in to the particular characters of the superimposed signal, and we develop encoding and coding schemes accordingly which offers a superior communication performance. Therefore, when compared with other concurrent transmission schemes that rely on retransmission [16,23], our scheme aims to successfully decode the superimposed signal in the first place, which saves the transmission energy and spectrum.

Third, our current implementation uses repeating symbols to obtain a diversity gain. By doing this, the decoding accuracy can be enhanced. To further improve the performance of our design, more sophisticated coding strategies can be combined with our encoding part. Here we use the rotational coding [5] as an example. In a nutshell, the rotational code would encode two symbols from two consecutive packets with a rotation matrix, and by doing this, the receiver can decode both of the two transmitted symbols as long as one of the symbols is correctly decoded, which offers a diversity gain plus a coding gain. Therefore, we can use this newly obtained coding gain to compensate our cost of the diversity gain, and eventually offers even higher throughput gain when competing with back-to-back transmission without blind transmission. We see this as a promising direction, and we will investigate it in our future work.

Capacity regions. In chapter, we introduce I-Talk a NOMA technology, for enabling two users to transmit their packets at the same spectrum resources simultaneously. Our current implementation of I-Talk is compared with IoT communication that enables blind transmission, and the results in Fig. 4.11 show that I-Talk, by using half of the channel resources, the BER is below 10^{-4} when SNR is higher than 10 dB. In particular, in Fig. 4.11, we compare the average BER of two concurrent users between the NOMA scheme with SIC and with I-Talk, and we also plot the

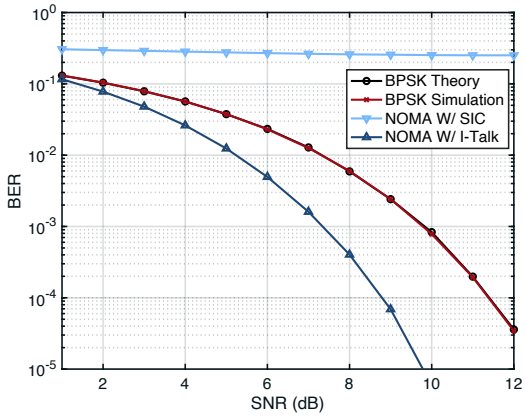


Figure 4.11: Decoding performance comparisons.

theoretical and simulation results of the BPSK modulation for a singular transmitter as a reference. Note that the results are based on a case that both of the received signal SNR are identical, and under such a case, the SIC would perform much worse compared with the performance of I-Talk, and we will detail the reason as follows.

The capacity analysis of OMA and NOMA with SIC have been well-studied in the literature [5]. Here, we start from some of the concepts of these knowledge and then explain our scheme. For traditional two-user OMA schemes, such as FDMA, TDMA, and CDMA, the capacity is limited by the user's capacity region. For example, under AWGN channels, the capacity regions for user 1 at a communication rate R_1, user 2 at a communication rate R_2 can be found as

$$\begin{cases} R_1 \leq W \log_2(1 + \frac{P_1}{N_0}), \\ R_2 \leq W \log_2(1 + \frac{P_2}{N_0}), \end{cases} \tag{4.8}$$

where W is channel bandwidth in Hz, P_1 and P_2 are the received power for user 1 and user 2, respectively, and N_0 denotes the power spectral density of the white Gaussian noise. Since OMA schemes need to divide the channel resource, an increased share and rate for user 1 will lead to a decreased for user 2, and the overall performance can be seen in Fig. 4.12.

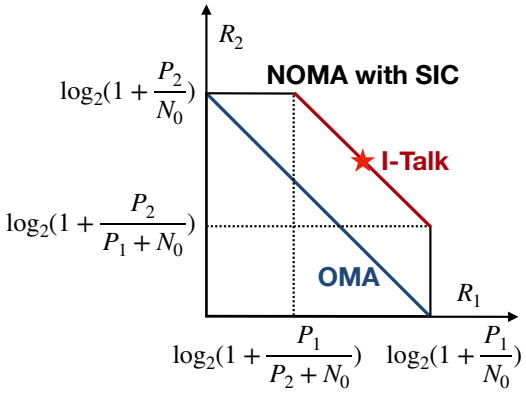


Figure 4.12: Capacity analysis.

Instead of competing for the spectrum resources, NOMA aims to let the two users access the channel simultaneously with some mutual interference. As a result, a new capacity region for the sum of two users can be represented as

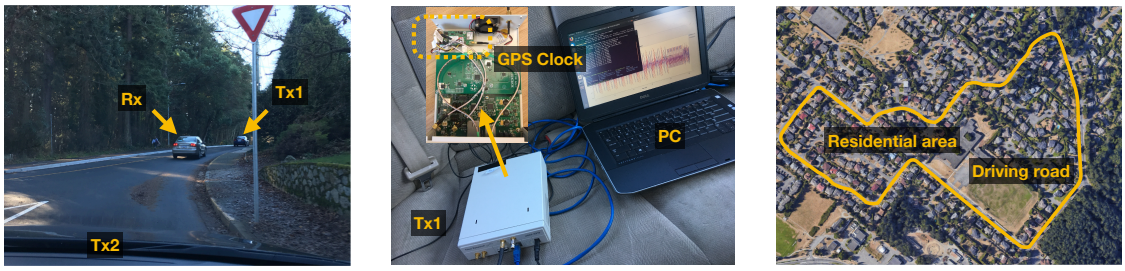
$$R_1 + R_2 \leq \log_2(1 + \frac{P_1 + P_2}{N_0}). \tag{4.9}$$

To achieve this goal, current NOMA leverages SIC to decode the signals from two concurrent users. Without loss of generality, we assume that $P_1 \leq P_2$, the receiver decodes the signal from user 2 first, and then cancels it out from the received signal. After that, the signal from user 1 can be decoded as interference from other user is cancelled. From the capacity perspective, the user 1 can use its full rate as $\log_2(1+\frac{P_1}{N_0})$, and user 2 can have a rate as

$$R_2 = \log_2(1 + \frac{P_2}{P_1 + N_0}). \tag{4.10}$$

By doing this, the overall capacity of the two users can be improved compared with OMA schemes.

However, to implement SIC in practice, a strong assumption is needed. In particular, an obvious power gap should exist between the transmitted signals from the



(a) The driving road (b) Hardware setups (c) The driving map

Figure 4.13: The road test.

two users; otherwise, the SIC cannot separate the signals effectively and obtain the desired capacity gain. Let us use numerical analysis to study this situation. For example, we have two signals at SNR of 5 dB, which means

$$\frac{P_1}{N_0} = \frac{P_2}{N_0} = 5\text{dB}. \tag{4.11}$$

When the receiver uses SIC to decode the superimposed signal, the SNR for user 2 would be decreased to 0.76, or -1.1 dB, which is undesirable for practical communication systems. Note that this low SNR will appear as long as the received signal powers are similar. To address this problem, I-Talk releases the constraint of the power gap. We decode the signal in the modulation level so that the signal would be separated in the constellation map instead of in the power domain, and by doing this, the capacity bound of I-Talk approaches $\log(1 + \frac{P_1+P_2}{N_0})$, as shown in Fig. 4.12.

4.6 Evaluation

4.6.1 Experiment Settings

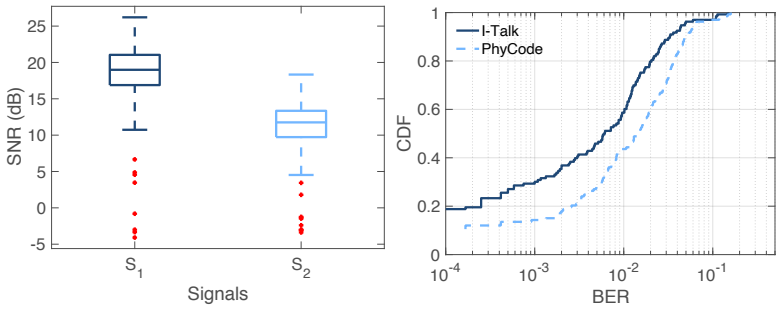
We build a prototype of I-Talk using USRP N210 embedded with an XCVR2450 daughter board. Each USRP connects to a GPS-Disciplined Oscillator (GPSDO) for the time synchronization. I-Talk is based on the PHY configurations of the IEEE

802.11p standard (i.e. the 5.9 GHz carrier frequency and 10 MHz bandwidth), and is built upon a recent project programmed in GNU-radio [50]. Prototyping in a software-defined radio platform, USRP (one of the most widely used software-defined radio system for research), is to demonstrate the feasibility of the proposed I-Talk solution. With the successful implementation in USRP, we can conclude that the proposed I-Talk is practical and promising.

To evaluate I-Talk in the mobile scenario, experiments with three vehicles are performed on city roads as shown in Fig. 4.13(c). During the experiment, the first and the third vehicles act as the source nodes transmitting signals simultaneously, while the second vehicle is the receiver node (Fig. 4.13(a)). The vehicles drove at a maximum speed of 40 km/h and following the normal traffic rules. The setup in one vehicle is shown in Fig. 4.13(b) where the communication node is placed on the back seat. We deploy an ECOM9-5500 mag-mount antenna with 9 dBi gain on top of each vehicle, near to the GPS antenna.

4.6.2 Overall Performance

Among all two-user superimposed signal decoding approaches, PNC can work well without power control [21–23] which is desirable for IoT systems. Since PhyCode [17] is the state-of-the-art approach based on PNC and can decode the superimposed signals for NOMA, we evaluate I-Talk in comparison with PhyCode. In particular, PhyCode decodes the superimposed signal in the modulation-level so it requires no power control. However, it assumes that all the constellation points are naturally distinguishable. This assumption can be held only for ideal conditions where no hardware imperfection and complicated channel conditions are taken into consideration. To be fair, we enable two-copy transmissions in PhyCode but without the rotation code, and select the signal copy that can pass the CRC check.



(a) SNR (b) BER

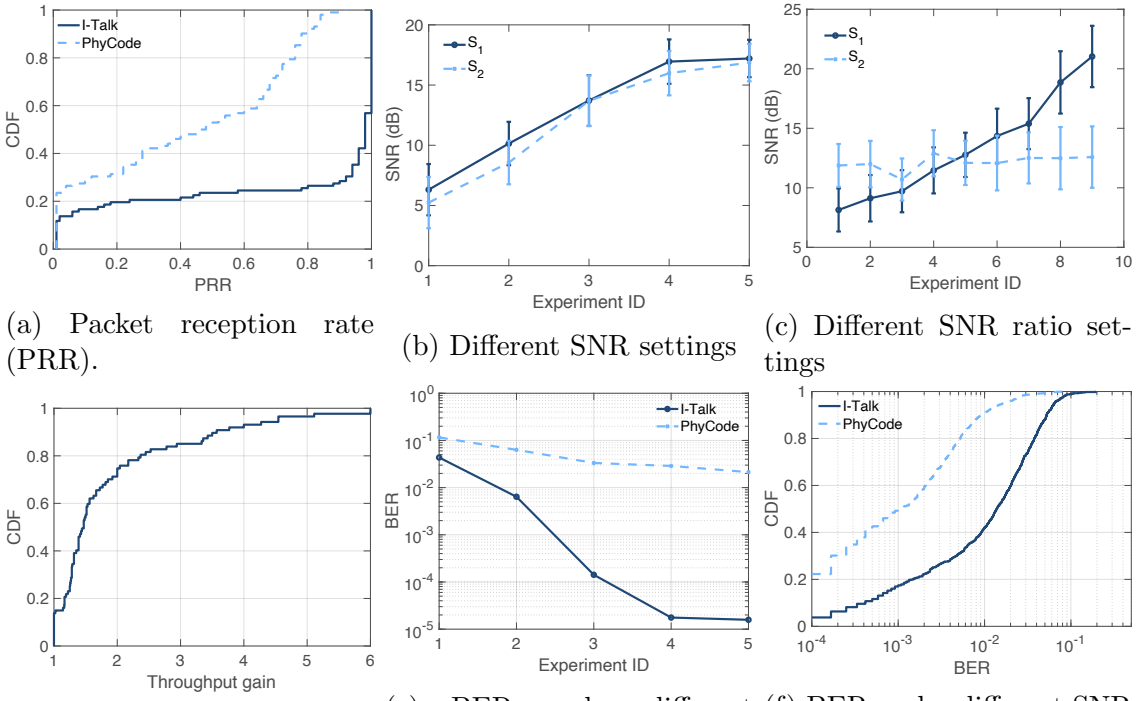
Figure 4.14: Road test results.

Road Test Performance

For the road test in the mobile scenario, we plot the SNR of two concurrent transmitted signals in Fig. 4.14(a), which shows that although the median SNR can be achieved as 18 dB and 11 dB, respectively, the variance of the SNR is very high as some outliers are below 0 dB, resulting in a highly dynamic channel condition. Fig. 4.14(b) shows the bit-error-rate (BER) comparison of I-Talk against PhyCode. The BER is calculated in each received packet. Obviously, the performance of I-Talk is 2.73× higher than PhyCode with the median BER of 5.75×10^{-3} over 1.56×10^{-2}. Overall, I-Talk can perform well under mobile scenarios.

Packet Level Performance

We evaluate I-Talk from the packet level in terms of the packet reception rate (PRR) and the throughput gain of I-Talk over PhyCode. We use the bit error information from 5000 packets received by the USRP device under different SNR settings. A common channel coding scheme in 802.11 p is used, i.e., the CRC-32 algorithm along with convolutional codes at 1/2 code rates. As plotted in Fig. 4.15(a), I-Talk can achieve a higher PRR with a median value of 0.98 than PhyCode with a median value of 0.52, which represents a significant improvement of reliability. Fig. 4.15(d) shows that I-Talk achieves a median 1.47× higher throughput gain than PhyCode.



(a) Packet reception rate (PRR).

(b) Different SNR settings

(c) Different SNR ratio settings

(d) Throughput gain

(e) BER under different SNRs

(f) BER under different SNR ratios

Figure 4.15: Performance evaluations.

4.6.3 Impacts of Practical Factors

To demonstrate that I-Talk can perform well without power control, we evaluate the impact of both the SNR and the SNR ratio of two transmitted signals. To focus on the influence of each parameter, we conduct this experiment in an open space where the GPS signals can be stably received. We collect 500 superimposed packets for each setting. These experiments evaluated our performance in static scenarios.

SNR. As shown in Fig. 4.15(b), we manually tune the SNR of the two transmitters from 5 dB to 15 dB. The BER results of I-Talk and PhyCode are illustrated in Fig. 4.15(e). Clearly, I-Talk outperforms PhyCode under different SNRs. Moreover, as the SNR increases, the performance of I-Talk is much higher than PhyCode. For example, when the SNR is around 15 dB, the BER results of I-Talk and PhyCode are 1.57×10^{-5} and 2.13×10^{-2}, respectively. These results reveal that I-Talk performs well within the typical SNR range (>5 dB [31]).

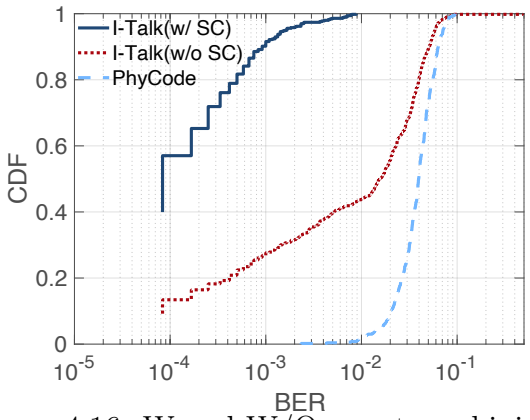


Figure 4.16: W and W/O smart combining.

SNR ratio. To evaluate the SNR ratio influence, we keep the SNR_2 around 12 dB and change the SNR_1 gradually. SNR_1 can be either larger or smaller than SNR_2. In Fig. 4.15(f), we see that I-Talk still outperforms PhyCode with the varying SNR ratio. The results reveal that I-Talk can perform well over a large range of the SNR ratio without any power control.

4.6.4 Evaluation of Smart Combining

We then focus on the other core component of I-Talk, i.e., smart combining. For simplicity, in this experiment, we set SNR_1 as 15 dB and SNR_2 as 12 dB. As shown in Fig. 4.16, the median BER is 1.6×10^{-2} without applying the smart combining. In contrast, the median BER of I-Talk significantly reduces to 8.33×10^{-5} after applying the smart combining scheme. This result demonstrates that I-Talk can precisely respond to the subcarriers' diversity. Besides, I-Talk outperforms PhyCode even without smart combining. Because I-Talk can still conduct a simple selection scheme with the knowledge of the phase differences compared to PhyCode.

4.7 Conclusion

In this chapter, we chapter I-Talk, a new NOMA approach designed for IoT systems that can decode the superimposed signals without power control. I-Talk can achieve high reliability and high throughput in the presence of mobility and hardware imperfections. We design a synthesis channel coefficient together with a diversity transmission and smart combining scheme to track the signal offsets caused by the hardware imperfection and reduce the decoding error under the mobile channel conditions. We demonstrate the performance of I-Talk with a software-defined radio platform in the mobile scenario. The results reveal that I-Talk outperforms the state-of-the-art scheme in terms of a higher reliability and a higher throughput gain.

Chapter 5

ChitChat: Efficient and Reliable Two-user Superimposed Signal Decoding

5.1 Introduction

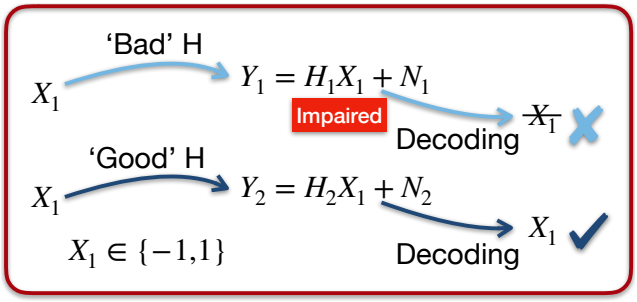
Since the wireless spectrum is at a premium, how to improve the spectrum efficiency becomes a major challenge for today's wireless communication systems. To solve this problem, a promising solution is to employ Non-Orthogonal Multiple Access (**NOMA**) by allowing more users to transmit signals in the same frequency at the same time, and then decode the two-user superimposed signals at the receiver. By doing so, the spectrum efficiency can be improved significantly.

Decoding the two-user superimposed signal for NOMA can be mainly divided into two categories: signal-level approaches and modulation-level approaches. For the signal-level approaches, Successive Interference Cancellation (SIC) [14–16] is the major technology. However, it relies on dedicated infrastructure to guarantee strict

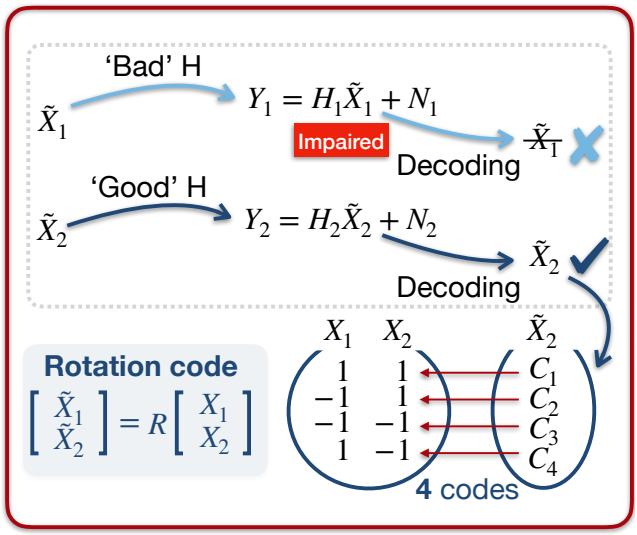
power control, which may not be feasible for low-cost and heterogeneous IoT devices. The modulation-level approaches, such as SigMix [18], I-Talk [19] and NCMA [28–30], can decode the two-user superimposed signals from the modulation domain without power control, which is desirable for IoT scenarios. Especially, SigMix [18] and I-Talk [19] provide a stable decoding performance for static and mobile applications, separately, which is essential for many IoT applications and paves the way for the further development of the two-user superimposed signal decoding schemes.

In this chapter, we introduce ChitChat, an effective encoding/decoding system for concurrent transmissions from two transmitters. In ChitChat, rather than requiring adding repetitive symbols or signal copies, we enable decoding the superimposed signals from the original symbols directly. We are inspired by the *rotation code* [5] which has been used for the single transmitter communication. The intuition behind is that the rotation code can exploit the degrees of freedom available in the wireless channel more effectively, i.e., a coding gain can be obtained beyond the diversity gain. Hence, both the reliability and spectrum efficiency can be achieved.

To see the basic idea of the rotation code clearly, consider a single transmitter transmitting signals to a receiver through two consecutive transmissions. Typically, if the first transmission fails, the same symbol will be retransmitted as shown in Fig. 5.1(a). In contrast, with the rotation code, the transmitter will transmit two different encoded symbols (i.e., $\tilde{X}_1$ and $\tilde{X}_2$) over two transmissions as shown in Fig. 5.1(b). So, the average information rate can be maintained to two over two symbol duration, which can improve the spectrum efficiency compared to blindly retransmissions. More importantly, its decoding procedure jointly considers the two received symbols from two transmissions (Y_1 and Y_2) and employs a combining decoding approach. This combining decoding approach actually provides a more reliable performance, since each encoded symbol contains two wanted symbols, X_1 and X_2.



(a) Average information rate is 1/2



(b) Average information rate is 2/2

Figure 5.1: Illustration of the rotation code.

In this case, even if one of the received symbols, say Y_2, is impaired, we can still successfully decode both wanted symbols X_1 and X_2. There is thus an opportunity to adopt the rotation code in concurrent transmissions (NOMA) to achieve a reliable decoding performance and preserve the spectrum efficiency.

However, the rotation code was invented for the single transmitter communication and cannot be adopted directly into the concurrent transmissions due to the following challenges.

- First, although the rotation code helps to improve the spectrum efficiency and reliability, **the code size and the number of constellation points are increased exponentially, making it difficult to decode superimposed signals**.

Considering the previous example again, there are 4 codes and 4 corresponding constellation points for BPSK modulation after using the rotation code. However, when two transmitters transmitting signals concurrently, the code size and the number of constellation points will be increased to $4^2 = \mathbf{16}$. When the number of constellation points increases, the system will face fundamental difficulties in obtaining clearly distinguishable constellation points at the receiver side which are essential for demodulation [18].

- Second, for a concurrent transmission, we observe that the distinguishable constellation points are mainly depend on the phase difference γ between transmitters. Furthermore, when using the rotation code, there are γ_1 and γ_2 from the two consecutive concurrent transmissions. Then, the difference between γ_1 and γ_2 (denoted as $\Delta\gamma$) is the key factor to obtain distinguishable constellation points. However, due to the dynamic environment and hardware imperfection in practice, γ varies and is uncontrollable at each concurrent transmission. Therefore, **it is non-trivial to guarantee a $\Delta\gamma$ for a reliable decoding performance by directly manipulating γ_1 and γ_2 at each concurrent transmission, respectively**.

To deal with all the above challenges, ChitChat makes the following contributions:

- For the first time, we understand the potential of using the rotation code in the concurrent transmissions to substantially improve the spectrum efficiency and achieve a reliable performance without requiring any repetitive transmissions.
- To truly implement the rotation code for concurrent transmissions in practice, we analyze the relationship between the decoding performance and the phase differences γ_1 and γ_2. Based on that, we derive the best value of the difference between γ_1 and γ_2 (i.e., $\Delta\gamma$) to achieve a lower decoding error rate. Hence, although the number of constellation points becomes larger (e.g., 16) when using the rotation code, ChitChat can still avoid indistinguishable constellation points through the

$\Delta\gamma$ for a reliable performance.

- According to the analysis result, we propose a weighted rotation code (WRC) for concurrent transmissions by creating a 'virtual' $\Delta\gamma$ that can be easily manipulated in practice. Specifically, instead of using two consecutive packets, we leverage the stability of γ within one packet and move γ_1 and γ_2 to one superimposed packet. So we can guarantee a suitable $\Delta\gamma$ by adding a specific weight to the transmitted symbol from one transmitter.

We implement ChitChat on a software-defined radio platform and evaluate its performance across various scenarios. Our extensive experimental results demonstrate that ChitChat outperforms the state-of-the-art concurrent transmission scheme with lower BERs and a higher throughput gain in static and dynamic channel environments.

5.2 Related Work

(a) NOMA: Successive Interference Cancellation (SIC) is a major NOMA technology, which requires strict power control to guarantee the power differences among transmitted signals. Thus SIC can decode the strongest signal first while treating others as noise, and then cancel the decoded signal out. By repeating this procedure, other signals can be decoded separately [16, 54]. Although SIC is promising, it relies on dedicated infrastructure and channel feedback for the strict power control [5, 59], which may not be desirable for low-cost and heterogeneous IoT devices [60] and may cause extra delay [56,57]. Instead of strict power control, other works were proposed to either increase the diversity gain or coding gain to decode the two-user superimposed signals. For the diversity gain, SigMix [18] transmits each signal twice back-to-back to guarantee a low error rate. Although SigMix provides high reliability, its through-

put upper bound is equal to that of sequential transmissions, which limits its gain in the network throughput. NCMA [28–30] employs error correction code by adding repetitive transmissions to recover the decoding error in the MAC layer. However, the spectrum efficiency cannot be fully utilized due to the repetitive transmissions. More importantly, their error recovering performance is severely impaired when the raw error rate is high, resulting in a poor reliability. In contrast to the previous works, ChitChat does not need power control and can obtain both diversity gain and coding gain without any repetitive transmissions. The theoretical throughput of ChitChat is doubled compared to sequential transmissions without sacrificing reliability can be improved significantly.

(b) Two-way relay networks: In two-way relay networks, two end nodes are scheduled to transmit signals concurrently, so signals are superimposed at the relay, and then the relay node broadcasts this superimposed signal back to the end nodes. Each end node can decode the signal transmitted from the other end node by canceling out its own signal [35, 58]. Several systems have been proposed to implement this idea including Physical-layer Network Coding (PNC) [21–23], Analog Network Coding (ANC) [31] and full duplex [88–91]. More recent works extended these systems to be more robust [23] and scalable to a long-hop scenario [32, 33]. However, these works were designed for relay networks and required knowing one of the two concurrent transmitted packets in decoding, making them incompatible to decode the superimposed signals for multiple access. In contrast, ChitChat can be applied for multiple access scenarios where the superimposed signals can be decoded without knowing one of the concurrently transmitted packets.

(c) General collisions: Due to the hidden terminals problem, multiple users may transmit packets to the same receiver simultaneously, which creates interference at the receiver. Unlike NOMA and bidirectional relay networks, this concurrent transmission

collides naturally without any synchronization scheme. Past work leveraged time offsets among collisions to separate multiple packets which contain two copies of each signal (e.g., ZigZag [49]). Given the redundancy of two copies per signal, ZigZag's throughput upper bound is limited to that of sequential transmissions. Decoding collisions in other IoT techniques, such as LoRa [46, 47], RFID [43, 45, 56, 92, 93] and ZigBee [36], has attracted many research interests. These approaches leverage some notable features, often associated with a lower rate, are not spectrum efficient. Motivated to achieve high efficiency and reliability for two-transmitter NOMA, we design ChitChat, which is applicable for high-rate wireless systems such as Wi-Fi with OFDM.

5.3 Background

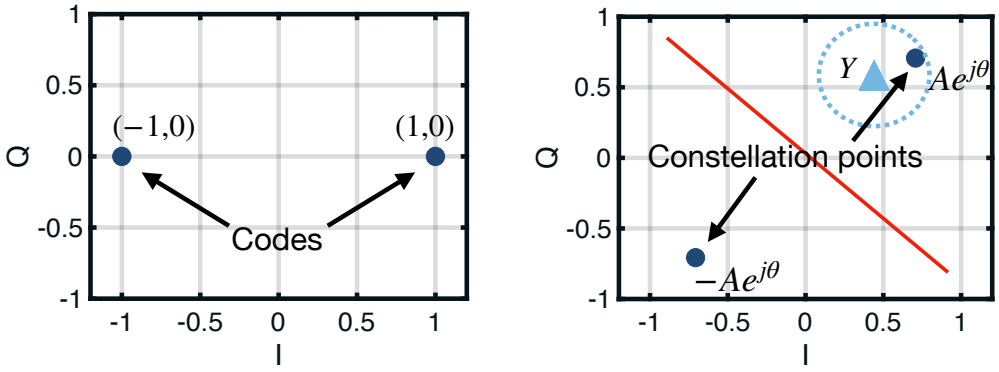
5.3.1 A Single Transmitter Communication

The wireless signal is a complex and discrete function of time. For a single transmitter, at each discrete time, the received symbol Y can be represented as

$$Y = HX + N, \tag{5.1}$$

where H is the wireless channel coefficient, X the transmitted symbol, and N the Gaussian noise with a zero-mean.

Specifically, if we have a binary digit S that needs to be transmitted, where $\forall S \in \{1, 0\}$, we first need to *modulate* it to a specific complex number X from the *code set* $\mathbf{C}$, say $\mathbf{C} = \{(1, 0), (-1, 0)\}$ for BPSK modulation. So, we will have $X = (1, 0)$ for "1" and $X = (-1, 0)$ for "0" as shown in Fig. 5.2(a). Then, this symbol X will transverse the wireless channel and this channel can cause signal variations on X. The signal variations can be represented by the channel coefficient H which is a



(a) Modulation (b) Demodulation

Figure 5.2: The modulation and demodulation of a single transmitter communication with BPSK as an example.

complex number $Ae^{j\alpha}$. Here, A is the signal strength attenuation and α is the signal shifting in the phase part. The received symbol Y is the varied X, which will be either $Ae^{j\alpha}+N$ or $-Ae^{j\alpha}+N$ based on Eq. 5.1. Here, we define $H\mathbf{C}$ as *constellation points*. For instance, $Ae^{j\alpha}$ and $-Ae^{j\alpha}$ are constellation points for codes $(1,0)$ and $(-1,0)$, respectively. Note that the constellation points in the constellation map are actually the codes shifted by the channel coefficient (see as shown in Fig. 5.2(b). *Demodulating* Y is to identify Y belonging to which code that the constellation point refers to. This can be achieved by calculating the shortest Euclidean distance to these constellation points as shown in as shown in Fig. 5.2(b).

5.3.2 The Concurrent Communication

Similar to the single transmitter case, the received two-user superimposed signal from a concurrent communication is still a complex function at each discrete time. The major difference is that the symbols from different transmitters will transverse different wireless channels. For simplicity, we use two concurrent transmitters Alice

($\mathcal{A}$) and Bob ($\mathcal{B}$) as an example, hence, we have

$$\begin{aligned} Y &= H_{\mathcal{A}} X_{\mathcal{A}} + H_{\mathcal{B}} X_{\mathcal{B}} + N \\ &= A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} X_{\mathcal{A}} + A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}} X_{\mathcal{B}} + N, \end{aligned} \tag{5.2}$$

where $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$ are the channel coefficients for $X_{\mathcal{A}}$ from Alice and $X_{\mathcal{B}}$ from Bob, respectively. For the transmitter side, the modulation requires no change at this moment, but the demodulation part at the receiver has two notable changes as shown below.

- The size of the constellation points have grown from **2** to **4**, i.e., $A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}}$, $-A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}}$, $-A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}}$, and $A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}}$. As a result, to demodulate the received symbol Y, the receiver needs to calculate the Euclidean distances to **4** constellation points instead of **2**. **Generally, if we have M codes and n concurrent transmitters, we will have M^n constellation points.**
- Some of the constellation points may be *indistinguishable* to each other. For example, if $H_{\mathcal{A}} = H_{\mathcal{B}}$, then two of the constellation points are identical to each other, i.e., $-A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}} = A_{\mathcal{A}} e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}} e^{j\alpha_{\mathcal{B}}}$. Hence, the receiver cannot demodulate the symbol Y correctly as there is no difference between the two constellation points. **This implies that the distinguishable constellation points are essential for demodulation.**

Note that **the constellation points mainly depends on the difference between the channel coefficients (denoted as channel difference for short)**. This provides us an opportunity to obtain distinguishable constellation points by manipulating channel difference.

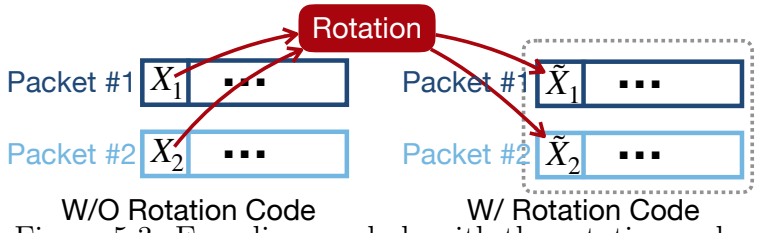


Figure 5.3: Encoding symbols with the rotation code.

5.4 The Design of ChitChat

The design of ChitChat starts with a question: *how can we ensure a 'good' channel difference, so that the constellation points are distinguishable?* Intuitively, each transmitter can transmit the same packet multiple times, so there is a high chance to have at least one 'good' channel difference among all transmitted packets. Then, the packets with the 'good' channel difference can be used for decoding, while other packets are discarded. Apparently, this solution offers a diversity gain to achieve reliable decoding performance, but at the cost of a lower spectrum efficiency.

ChitChat aims to obtain 'good' channel difference without sacrificing the spectrum efficiency. We are inspired by the *rotation code* [5] used for the single transmitter communication. The purpose of the rotation code is to exploit the degrees of freedom available in the wireless channel effectively. Specifically, the rotation code can not only offer a diversity gain, but also a coding gain to ensure both the reliability and spectrum efficiency.

5.4.1 The Rotation Code in a Single Transmitter Communication

The rotation code has been used in the single transmitter communication as follows.

Encoding. Suppose X_1 and X_2 are two symbols from two consecutive packets, respectively. Without the rotation code, they will directly transverse different channels

represented by H_1 and H_2. When applying the rotation code, as shown in Fig. 5.3, these two symbols will be encoded to two new symbols $\tilde{X}_1$, $\tilde{X}_2$ with a rotation matrix $\mathbf{R}$,

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{5.3}$$

where θ is the rotation angle that belongs to $[0, 360^\circ)$. Mathematically, this encoding procedure is given by

$$\begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \cos\theta X_1 - \sin\theta X_2 \\ \sin\theta X_1 + \cos\theta X_2 \end{bmatrix}. \tag{5.4}$$

Then, the transmitter transmits the new symbols $\tilde{X}_1$ and $\tilde{X}_2$ through different channels H_1 and H_2. The received symbols Y_1 and Y_2 can be represented as

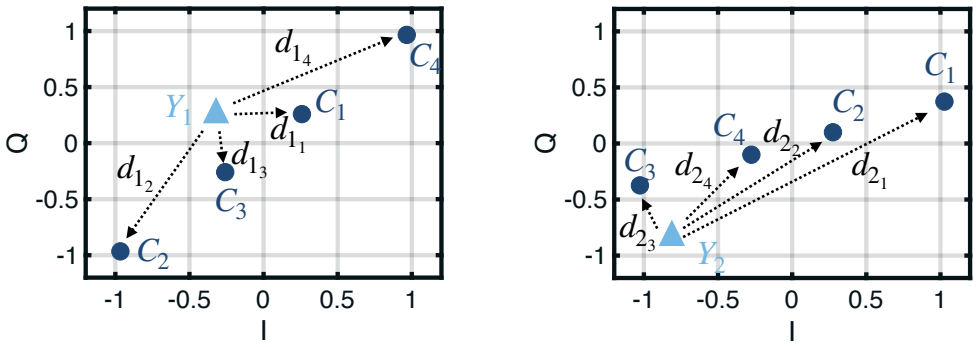
$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix}. \tag{5.5}$$

For simplicity, we omit the Gaussian noise in the above equation.

Note that given X_1, $X_2 \in \{(1,0), (-1,0)\}$ for BPSK modulation, if we substitute X_1 and X_2 into Eq. 5.4, then there are **4** unique codes for each encoded symbol, $\tilde{X}_1$ or $\tilde{X}_2$, i.e.,

$$\begin{aligned} \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \in \Bigg\{ & \underbrace{\begin{bmatrix} (\cos\theta - \sin\theta, 0) \\ (\sin\theta + \cos\theta, 0) \end{bmatrix}}_{\mathbf{C}_1}, \underbrace{\begin{bmatrix} (-\cos\theta - \sin\theta, 0) \\ (-\sin\theta + \cos\theta, 0) \end{bmatrix}}_{\mathbf{C}_2}, \\ & \underbrace{\begin{bmatrix} (-\cos\theta + \sin\theta, 0) \\ (-\sin\theta - \cos\theta, 0) \end{bmatrix}}_{\mathbf{C}_3}, \underbrace{\begin{bmatrix} (\cos\theta + \sin\theta, 0) \\ (\sin\theta - \cos\theta, 0) \end{bmatrix}}_{\mathbf{C}_4} \Bigg\} \end{aligned} \tag{5.6}$$

Decoding. From the receiver side, these **4** codes refer to **4** constellation points in the constellation map. As shown in Fig. 5.4, the two received symbols Y_1 and Y_2 have different constellation maps. According to Section 5.3, $\tilde{X}_1$ can be demodulated by



(a) Y_1 (b) Y_2

Figure 5.4: Decoding symbols with the rotation code. In the above example, the demodulation result is C_3, as the summation $d_{1_3} + d_{2_3}$ is the smallest.

calculating the shortest Euclidean distance from the received symbol Y_1 to these **4** constellation points. $\tilde{X_2}$ can be also demodulated in the same way. However, due to the dynamic environment in practice, either Y_1 or Y_2 may suffer from the 'bad' channel difference, resulting in an unreliable decoding performance. To solve this problem, in the design of the rotation code, the decoding procedure is to jointly consider both Y_1 and Y_2 and employ the combining approach. This combining approach is to minimize the summation of d_{1_i} and d_{2_i} $(i = 1, 2, 3, 4)$ as shown in Fig. 5.4, which can be calculated by the maximum likelihood (ML) as

$$\hat{\tilde{\mathbf{X}}} = \arg \min_{\tilde{\mathbf{X}}=[\tilde{X_1},\tilde{X_2}]^T} \|\mathbf{Y} - \mathbf{H}\tilde{\mathbf{X}}\|^2. \tag{5.7}$$

Apparently, once $\tilde{\mathbf{X}}$ can be demodulated correctly, both wanted symbols X_1 and X_2 can be decoded. Similarly, other symbols in packet #1 and packet #2 can be pairwisely encoded and decoded.

Benefits. With rotation code, the two transmissions contain two different symbols ($\tilde{X_1}$ and $\tilde{X_2}$). So the average information rate is 2/2, preserving the spectrum efficiency. More importantly, the combining decoding approach provides a higher reliability. For example, even if one of the received symbols, say Y_2, is impaired, we can still successfully decode both wanted symbols X_1 and X_2. We just need to **avoid the situation when both Y_1 and Y_2 are impaired**.

The above analysis reveals that for a single transmitter, **by using the rotation code, we can achieve the reliable performance and preserve the spectrum efficiency simultaneously** without transmitting the same packets multiple times.

5.4.2 The Rotation Code in the Concurrent Communication

In ChitChat, we adopt the rotation code in the concurrent communication to ensure a high reliability without sacrificing efficiency.

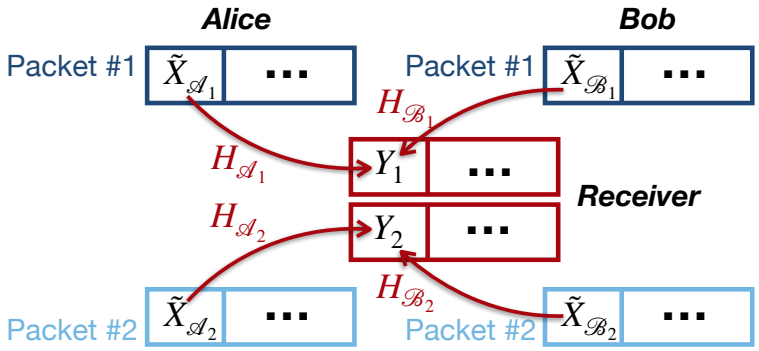


Figure 5.5: The rotation code in concurrent transmissions.

Consider two transmitters, Alice and Bob, transmitting signals simultaneously to one receiver. Each transmitter encodes its symbols with the rotation code described in Section 5.4.1. Hence, as shown in Fig. 5.5, the received symbol Y_1 and Y_2 are superimposed symbols which can be represented as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_{\mathcal{A}_1} & H_{\mathcal{B}_1} & 0 & 0 \\ 0 & 0 & H_{\mathcal{A}_2} & H_{\mathcal{B}_2} \end{bmatrix} \begin{bmatrix} \tilde{X}_{\mathcal{A}_1} \\ \tilde{X}_{\mathcal{B}_1} \\ \tilde{X}_{\mathcal{A}_2} \\ \tilde{X}_{\mathcal{B}_2} \end{bmatrix}. \tag{5.8}$$

Similar to a single transmitter, when adopting the rotation code for concurrent transmissions, it can offer a diversity gain for a reliable performance and also a coding gain to preserve the spectrum efficiency. However, recall that in the design of the rotation code, we need to avoid the situation when both Y_1 and Y_2 are failed to be demodulated. Hence, a question is raised naturally: *how can we avoid this situation happening in concurrent transmissions?*

Recall that for the concurrent transmissions, obtaining the distinguishable constellation points depends on the 'good' channel difference, which is essential for the demodulation. Therefore, to answer the above question, we need to understand the relationship between the decoding performance and the channel difference. Specifically, which values of the channel difference are 'good' for decoding and which are 'bad'.

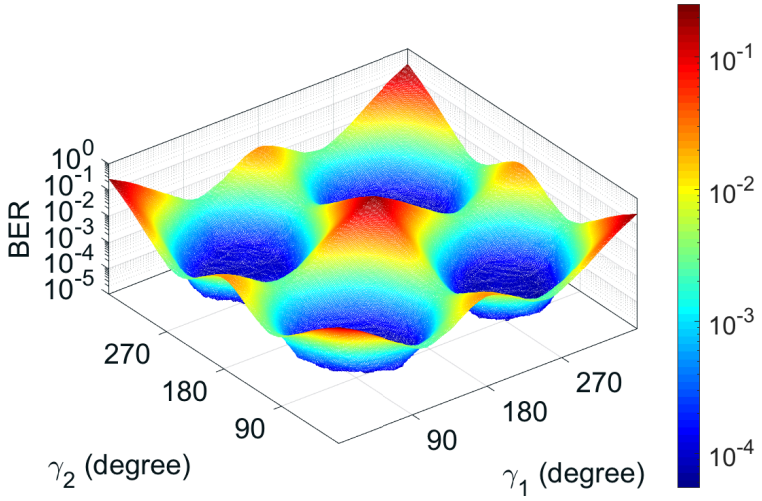
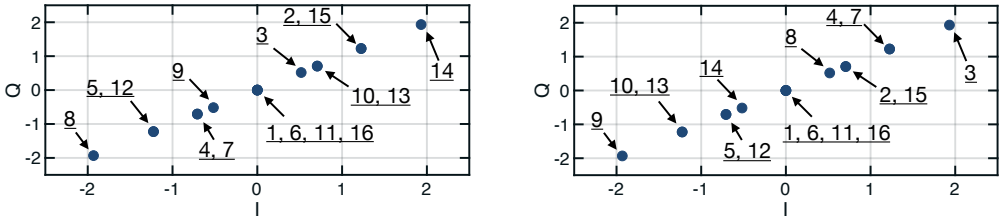


Figure 5.6: The BER under different γ_1 and γ_2.

Understanding the errors. The channel difference contains two parts: the attenuation ratio ($\eta = \frac{|H_{\mathcal{A}}|}{|H_{\mathcal{B}}|} = \frac{A_{\mathcal{A}}}{A_{\mathcal{B}}}$) and the phase difference ($\gamma = |\angle H_{\mathcal{A}} - \angle H_{\mathcal{B}}| = |\alpha_{\mathcal{A}} - \alpha_{\mathcal{B}}|$). Manipulating the attenuation ratio requires dedicated devices for a strict power control is infeasible for low-cost IoT devices. Therefore, in ChitChat, we focus on manipulating the phase difference γ to control the channel difference. We use the bit-error-rate (BER) as the metric for decoding performance. For each received symbol, Y_1 or Y_2, there is a corresponding phase difference, i.e., γ_1 or γ_2. Thus, we start by understanding the relationship between BER and the phase difference (γ_1 and γ_2). Specifically, we conducted Monte-Carlo simulation to emulate all the possible values of γ_1 and γ_2 within [0°, 360°]. The rotation angle θ is fixed as 30° which has been proved optimal [94]. Here, we use one of the simulation results as an example shown in Fig. 5.6 to explain our observations, where the SNR for Alice and Bob are 10 dB and 15 dB, respectively. Clearly, the error peaks (i.e., the highest BER) appear at certain values of γ_1 and γ_2, such as $\gamma_1 = 180°, \gamma_2 = 180°$, and $\gamma_1 = 0°(360°), \gamma_2 = 360°(0°)$. This is the 'bad' situation that we need to avoid in our design. One thing worth noting here is that we also tested different values of SNR for Alice and Bob, and we notice that the varying pattern of the BER is the same with arbitrary SNRs. One



(a) For received symbol Y_1. (b) For received symbol Y_2.

Figure 5.7: The distribution of constellation points when $\gamma_1 = 180°, \gamma_2 = 180°$.

key feature is revealed from this observation: the BER varying pattern is dominated by the γ_1 and γ_2, and it is independent to the SNRs. Therefore, we can ignore the influence of the SNR which cannot easily be controlled, and focus on manipulating the phase difference γ.

To further understand why these errors happen, we zoom into the constellation points of the received symbols. Note that since each transmitted symbol (either from Alice or Bob) has **4** possible codes after using the rotation code, the received superimposed symbol will have **16** possible codes corresponding to **16** constellation points. Taking $\gamma_1 = 180°, \gamma_2 = 180°$ as an example, we plot the **16** constellation points for Y_1 and Y_2 as shown in Fig. 5.7(a) and Fig. 5.7(b), respectively. Our observations are as below. The similar observation can be obtained in other cases, such as $\gamma_1 = 0°(360°), \gamma_2 = 360°(0°)$.

- **Distinguishable constellation points:** some constellation points are clearly distinguishable, such as points 8 and 14 for Y_1, and 3 and 9 for Y_2.
- **Close constellation points:** some constellation points are too close to the nearby constellation points, such as points 3 and 9 for Y_1, and 8 and 14 for Y_2.
- **Indistinguishable constellation points:** some constellation points are overlapped and merge to one point, resulting in indistinguishable cases, such as points 1/6/11/16, 2/15, 4/7, 5/12 and 10/13 for both Y_1 and Y_2.

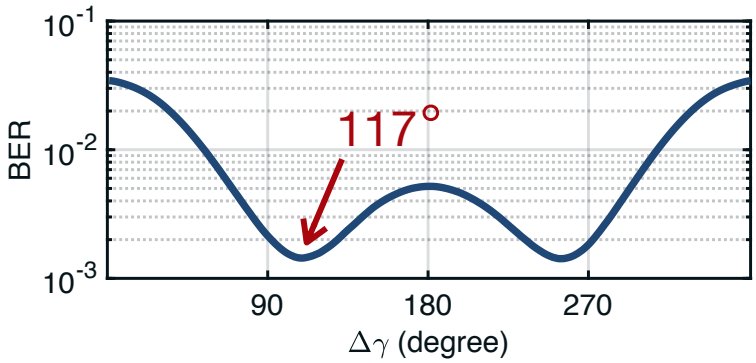


Figure 5.8: The BER under different $\Delta\gamma$.

The above observations reveal two properties: (1) *complementary property:* the close constellation points for Y_1 are actually distinguishable for Y_2, and vice versa; (2) *consistent property:* the overlapped constellation points are the same for both Y_1 and Y_2. The complementary property can help the combining decoding approach, so that the close constellation points problem can be solved. But the consistent property makes the overlapped constellation points unsolvable and causes decoding errors, such as the BER peak shown in Fig. 5.6.

We conclude that *the main reason for decoding errors is the overlapped or close constellation points, and they are caused by a 'bad' difference between* γ_1 *and* γ_2, i.e., a 'bad' $\Delta\gamma$, where $\Delta\gamma = |\gamma_1 - \gamma_2|$. Thus, we need to obtain a 'good' $\Delta\gamma$ and avoid the 'bad' $\Delta\gamma$.

The value of 'good' $\Delta\gamma$. According to the relationship between the BER and γ_1, γ_2 shown in Fig. 5.6, the BER can be considered as a function of γ_1, γ_2, notated as $P(\gamma_1, \gamma_2)$. For simplicity, we rewrite it as $P(\Delta\gamma)$. Thus, the 'good' $\Delta\gamma$ that refers to a lower BER can be calculated by

$$\Delta\gamma_{good} = \arg \min_{\Delta\gamma \in [0°, 360°)} P(\Delta\gamma). \tag{5.9}$$

To obtain the 'good' $\Delta\gamma$, we further plot the BER curve under different $\Delta\gamma$ in Fig. 5.8, which is the statistical result from the above Monte-Carlo simulations (Fig. 5.6). Note that different values of γ_1 and γ_2 may result in the same $\Delta\gamma$. Hence,

the BER under a particular $\Delta\gamma$ shown in Fig. 5.8 is an average result. Apparently, there are two valleys when $\Delta\gamma = 117°$ and $\Delta\gamma = 243°$, and they are symmetrical with $\Delta\gamma = 180°$. Thus, the best choice of a 'good' $\Delta\gamma$ is centered around 117° or 243°. We choose $\Delta\gamma = 117°$ in our following experiments and this value performs well. Also note that the 'good' value of $\Delta\gamma$ may be slightly different given different hardware and channel environments.

5.4.3 Weighted Rotation Code in the Concurrent Communication

Next, given the 'good' value of $\Delta\gamma$, *how can we guarantee this value in the practical system*?

In practice, the phase difference γ varies at each concurrent transmission due to the dynamic environments caused by the moving objects [95–97] and the system level noises caused by the hardware imperfections [37, 60, 70]. Recall that γ_1 and γ_2 are originally from two consecutive concurrent transmissions, so it is very challenging to guarantee a 'good' $\Delta\gamma$ by directly manipulating γ_1 and γ_2, respectively.

Weighted encoding. To solve this problem, we observe that although γ is different among different superimposed packets, it is stable and measurable a packet transmission time. Therefore, when γ_1 and γ_2 are from the same superimposed packet, we can add a weight (i.e, a phase shift) to the transmitted symbol to create a 'virtual' $\Delta\gamma$. Apparently, this virtual $\Delta\gamma$ can be easily manipulated to the 'good' value. To this end, we propose the Weighted Rotation Code (WRC) for concurrent transmissions as follows. For each transmitter, instead of encoding two symbols from two consecutive packets, we encode two consecutive symbols within one packet. For example, with the typical rotation code, Alice encodes two symbols $\tilde{X}_{\mathcal{A}_1}$ and $\tilde{X}_{\mathcal{A}_2}$ from packet #1 and packet #2 (see Fig. 5.5). But when employing our proposed weighted rotation

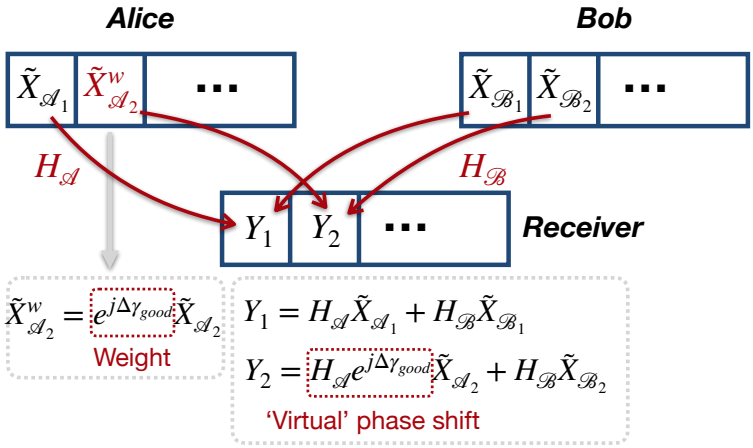


Figure 5.9: Illustration of the weighted rotation code.

code, Alice will encode two consecutive symbols $\tilde{X}_{\mathcal{A}_1}$ and $\tilde{X}_{\mathcal{A}_2}$ from the same packet as shown in Fig. 5.9. Bob will encode the symbols in the same way. Then, Alice adds a weight $e^{j\Delta\gamma_{good}}$ to its second symbol, i.e., $\tilde{X}^{\omega}_{\mathcal{A}_2} = e^{j\Delta\gamma_{good}}\tilde{X}_{\mathcal{A}_2}$.

For the receiver side, Y_1 and Y_2 are two consecutive symbols in the same superimposed packet. Thus, we have $\gamma_1 = \gamma_2 = |\angle H_{\mathcal{A}} - \angle H_{\mathcal{B}}|$. However, since the second encoded symbol from Alice is a weighted symbol, this weight can be considered as a 'virtual' phase shift to the channel coefficient $H_{\mathcal{A}}$, so we will have a new channel coefficient $e^{j\Delta\gamma_{good}}H_{\mathcal{A}}$. Therefore, γ_1 and γ_2 can be rewritten as

$$\begin{aligned} \gamma_1 &= |\angle H_{\mathcal{A}} - \angle H_{\mathcal{B}}| \\ \gamma_2 &= |(\angle H_{\mathcal{A}} + \Delta\gamma_{good}) - \angle H_{\mathcal{B}}|. \end{aligned} \tag{5.10}$$

Apparently, by doing so, we will have a 'virtual' $\Delta\gamma = |\gamma_1 - \gamma_2| = \Delta\gamma_{good}$. That is to say, we can now guarantee a 'good' $\Delta\gamma$ by adding a specific weight ($\Delta\gamma_{good}$) to the second symbol from one transmitter (Alice). Note that the remaining symbols within the same packet from Alice and Bob can be pairwisely encoded in the same way, such as the third and fourth symbols, and the fifth and sixth symbols, etc.

Smart decoding scheme. The decoding scheme in ChitChat is similar to the decoding procedure in the rotation code introduced in Section 5.4.1. Suppose there are $2n$ symbols in each packet, any two consecutive symbols from Alice and Bob,

i.e., $\tilde{X}_{\mathcal{A}_i}$, $\tilde{X}_{\mathcal{A}_{i+1}}$, $\tilde{X}_{\mathcal{B}_i}$, $\tilde{X}_{\mathcal{B}_{i+1}}$, where $i \in [1, 2n]$ can be demodulated by the combining approach as below. Note that if there are odd number symbols, we will pad a 0 at the end of the packet.

$$\begin{bmatrix} \hat{\tilde{X}}_{\mathcal{A}_i} \\ \hat{\tilde{X}}_{\mathcal{B}_i} \\ \hat{\tilde{X}}_{\mathcal{A}_{i+1}} \\ \hat{\tilde{X}}_{\mathcal{B}_{i+1}} \end{bmatrix} = \arg \min_{\tilde{X}_{\mathcal{A}_i}, \tilde{X}_{\mathcal{A}_{i+1}}, \tilde{X}_{\mathcal{B}_i}, \tilde{X}_{\mathcal{B}_{i+1}}} [Y_i - (H_{\mathcal{A}} \tilde{X}_{\mathcal{A}_i} + H_{\mathcal{B}} \tilde{X}_{\mathcal{B}_i})] \\ + [Y_{i+1} - (e^{j\Delta\gamma_{good}} H_{\mathcal{A}} \tilde{X}_{\mathcal{A}_{i+1}} + H_{\mathcal{B}} \tilde{X}_{\mathcal{B}_{i+1}})]. \tag{5.11}$$

Note that the channel coefficient $H_{\mathcal{A}}$ ($H_{\mathcal{B}}$) slightly varies within each packet, but the two consecutive symbols have the similar channel coefficients. Thus, it is safe to use $H_{\mathcal{A}}$ ($H_{\mathcal{B}}$) for both $\tilde{X}_{\mathcal{A}_i}$ ($\tilde{X}_{\mathcal{B}_i}$) and $\tilde{X}_{\mathcal{A}_{i+1}}$ ($\tilde{X}_{\mathcal{B}_{i+1}}$) in Eq. 5.11. Now, we have the demodulation results $\tilde{X}_{\mathcal{A}_i}$, $\tilde{X}_{\mathcal{A}_{i+1}}$, $\tilde{X}_{\mathcal{B}_i}$, $\tilde{X}_{\mathcal{B}_{i+1}}$, and then the original wanted symbols $X_{\mathcal{A}_i}$, $X_{\mathcal{A}_{i+1}}$, $X_{\mathcal{B}_i}$ and $X_{\mathcal{B}_{i+1}}$ can be decoded according to the possible **16** codes.

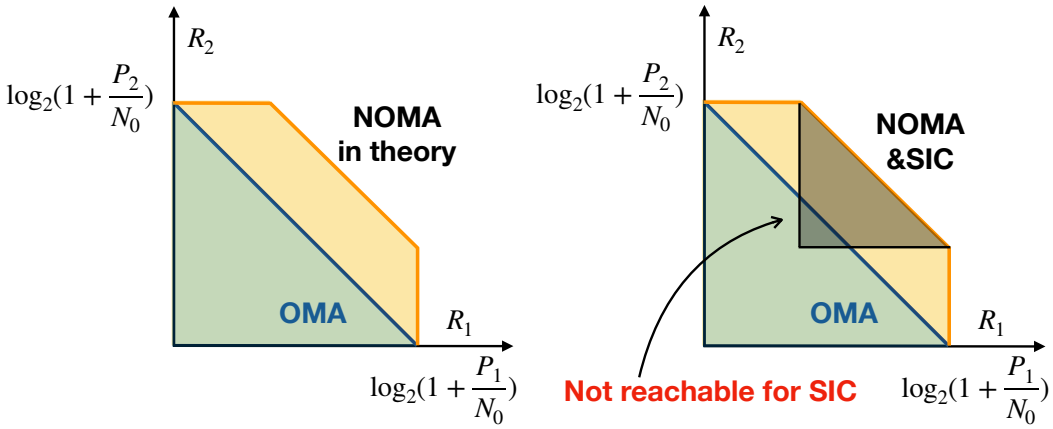
5.4.4 Analysis

Capacity regions for NOMA with ChitChat and NOMA with SIC.

The capacity region of OMA and NOMA has been studied extensively in existing work [5]. In particular, for AWGN channels and using two users as an example, the communication rate (bit per second per Hz) constrains R_1 and R_2 for user 1 and user 2 are shown below.

$$\begin{cases} R_1 \leq \log_2(1 + \frac{P_1}{N_0}), \\ R_2 \leq \log_2(1 + \frac{P_2}{N_0}), \\ R_1 + R_2 \leq \log_2(1 + \frac{P_1 + P_2}{N_0}), \end{cases} \tag{5.12}$$

where N_0 refers to the power spectral density of the white Gaussian noise, and P_1 and P_2 are the received power for user 1 and user 2, separately. Considering existing OMA solutions are sharing the spectrum with different users, we can see the capacity



(a) NOMA with ChitChat (b) NOMA with SIC

Figure 5.10: Capacity region for NOMA.

region marked as the green color for OMA as shown in Fig. 5.10.

When it comes to NOMA, it aims at enabling concurrent transmissions and then decode the superimposed signal, which would offer a higher capacity gain than OMA as shown in the orange region of Fig. 5.10. Here, we can use an numerical example to describe it more clearly. Given $\frac{P_1}{N_0} = 10\text{dB}$ and $\frac{P_2}{N_0} = 5\text{dB}$, and then we would have

$$\text{OMA} \begin{cases} R_1 = a\log_2(1+\frac{P_1}{N_0}) = 3.46a, \\ R_2 = (1-a)\log_2(1+\frac{P_2}{N_0}) = 2.06(1-a), \end{cases} \tag{5.13}$$

$$\text{NOMA with SIC} \begin{cases} R_1 = \log_2(1+\frac{P_1}{N_0+P_2}) = 1.77, \\ R_2 = \log_2(1+\frac{P_2}{N_0}) = 2.06, \end{cases} \tag{5.14}$$

where $0 \leq a \leq 1$, and the sum rate of NOMA with SIC is greater than the OMA schemes.

However, the implementation of SIC in practice has a strong assumption that the concurrent signals should be separable in the power domain, and when it is not, the SNR of one user would be decreased significantly, leading to an undeniable communication performance. For example, given $\frac{P_1}{N_0} = \frac{P_2}{N_0} = 5\text{dB}$, and then we would

have

$$\text{OMA} \begin{cases} R_1 = a\log_2(1+\frac{P_1}{N_0}) = 2.06a, \\ R_2 = (1-a)\log_2(1+\frac{P_2}{N_0}) = 2.06(1-a), \end{cases} \tag{5.15}$$

$$\text{NOMA with SIC} \begin{cases} R_1 = \log_2(1+\frac{P_1}{N_0+P_2}) = 0.815, \\ R_2 = \log_2(1+\frac{P_2}{N_0}) = 2.06. \end{cases} \tag{5.16}$$

Note that although NOMA with SIC still has a better capacity gain in theory, but the SNR for user 1, $\frac{P_1}{N_0+P_2}$, would be decreased to -1.1 dB. Most of communication systems cannot maintain reliable communications in such a low SNR. Furthermore, given P_1 and P_2, the dark region marked in the Fig. 5.10(b) is not reachable for SIC. Note that for a modulation-level superimposed signal decoding scheme, ChitChat, it requires no power control, and it has the potential to approach to the theoretical capacity region of NOMA, resulting in a better performance for low-cost devices when compared with SIC.

Spectrum efficiency/throughput. Current NOMA approaches in modulation-level (e.g., SigMix [18] and NCMA [28–30]) either require to add one repetitive signal copy or several repetitive symbols to achieve a reliable decoding performance. Hence, their throughput upper bound is equal or similar to that of sequential transmissions. Unlike the state-of-the-art modulation-level NOMA approaches, ChitChat can decode the two-user superimposed signal without requiring any repetitive transmissions. The theoretical throughput of ChitChat is actually doubled compared to sequential transmissions. Hence, ChitChat can truly enable concurrent transmissions in the same frequency at the same time and substantially boost the spectrum efficiency for NOMA.

Complexity. At the transmitter side, ChitChat introduces a weighted encoding scheme. This scheme can be seen as a patch of the existing modulation block, and its overhead is growing linearly and limited by the number of transmitted symbols.

That is to say the use of ChitChat will cause no notable extra energy consumption on the transmitter, making it friendly to IoT devices. Similarly, the receiver decodes the superimposed signal by using a maximum-likelihood solution as the existing WiFi decoder does, which causes no notable extra complexity energy consumption.

5.5 Implementation

Note that the modulation-level NOMA implementation is not trivial, given the frequency and phase offsets in practical systems. Next, We implement ChitChat by using three USRPs embedded with XCVR2450 daughter boards. For each USRP, we employ an ECOM9-5500 mag-mount antenna with 9 dBi gain or an HG2458RD-SM omni-directional antenna with 3 dBi gain, and the selection of the antenna depends on the deployment requirement as the mag-mount antenna is preferred to be placed on the ground or on metal surfaces. ChitChat is built upon a recent project programmed in GNU-radio [50], which follows IEEE 802.11p standard with 5.9 GHz carrier frequency and 10 MHz bandwidth. The program is running on PCs operating on Ubuntu 16.04, and each PC processes the data streams coming from the USRP. To focus on our design, we remove other schemes along with the Wi-Fi protocol for simplicity, such as scrambling, channel coding, etc.

Practical issues. Decoding the two-user superimposed signals for NOMA demands symbol-level time synchronization, frequency synchronization and phase synchronization [21], which can guarantee different transmitters transmit signals in the same frequency at the same time. For the time synchronization, fortunately, there have been extensive studies recently to provide the symbol-level time synchronization with low-cost solutions, such as by using GPS clocks in outdoor scenarios [48], and by using WiFi devices in indoor scenarios [41, 42]. For simplicity, we use a central clock as

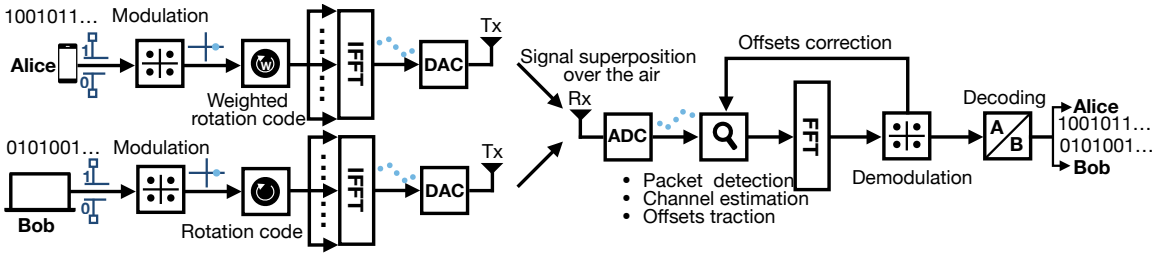


Figure 5.11: The block diagram of ChitChat.

a solution in this chapter for time synchronization, and focus on the other unsolved synchronization problems in the frequency and phase domains. In practice, due to the differences between oscillators and signal detection failures [37], three types of signal offsets exist, including the Carrier Frequency Offset (CFO), the Sampling Frequency Offset (SFO) and the Sampling Time Offset (STO). These offsets will influence the required synchronization accuracy negatively. In ChitChat, we employ and implement a precise offsets tracking at the receiver to correct these offsets. Specifically, this mechanism introduces orthogonal preambles in the packet header to track the offsets of each transmitted signal. After that, the detected offsets can be added up to the channel coefficient for further demodulation. The block diagram of ChitChatis shown in Fig.5.11.

The transmitters. Two USRP N210s connect to a PC through a Gigabit Ethernet router as two transmitters, Alice and Bob. For the time synchronization, each USRP is connected to a central clock (i.e., NI CDA-2990) via SMA cables. Each transmitted packet has a total length of 1500 bytes payload and a preamble, which follows the Wi-Fi standard. Specifically, after the BPSK modulation, every two consecutive symbols within the packet from Alice are encoded by the proposed weighted rotation code, and Bob follows the rotation code without weighting. We then transform the frequency domain BPSK symbols into the time domain samples via the IFFT. The time domain samples are ordered by the parallel-to-serial converter and passed through a DAC converter, resulting in the baseband OFDM signal which is then upconverted to the

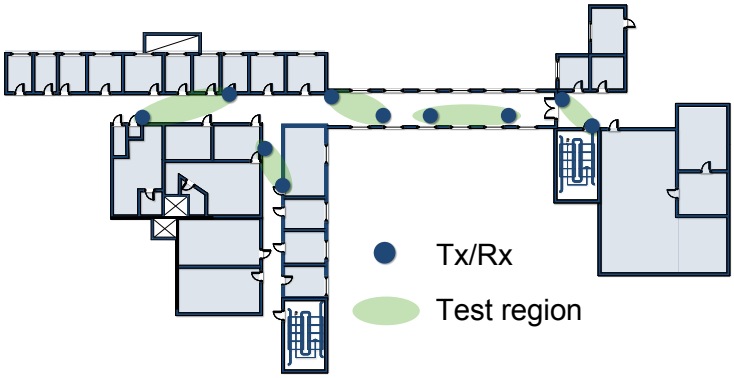


Figure 5.12: The deployment layout.

carrier frequency.

The receiver. One USRP N210 connects to another PC through Ethernet cable as the receiver. The received two-user superimposed signal is downconverted to baseband and filtered to remove the high-frequency components. Then, the ADC converter samples the signal. The receiver detects the beginning of a superimposed packet by extracting the long training sequence, and tracks CFO, STO and SFO. These time samples are serial-to-parallel converted and passed through the FFT. The FFT output is passed through the demodulator. Then, the proposed smart decoding scheme is implemented to decode the original data. Note that we omit some detailed descriptions, e.g., the cyclic prefix, due to space limit.

5.6 Evaluation

We conduct experiments in an office building including labs, offices and corridors. We randomly select 5 regions as the test regions shown in Fig. 5.12. For each region, we deploy two transmitters and one receiver. The distances among the three nodes vary from 1 m to 8 m according to the size of each region. Fig. 5.13 shows the experimental setup in one of the corridors. The sending rate is 1 pkt/s and we collect 500 concurrent transmissions in each region. **Compared schemes.** We compare the

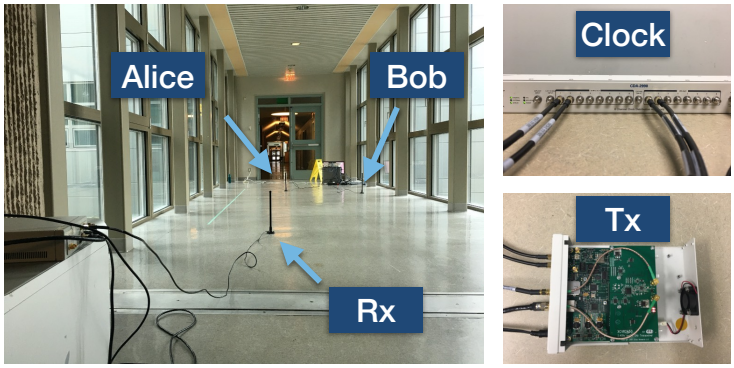


Figure 5.13: Experimental setups.

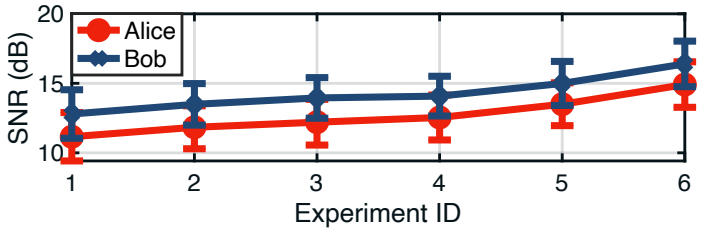
performance of ChitChat with two state-of-the-art schemes.

- **SigMix.** SigMix [18] requires to transmit two same signal copies within each packet and adds a phase shift between the two copies. It then decodes the two-user superimposed signal by selecting a good copy and discards the other for a reliable decoding performance. The design philosophy of I-Talk [19] is similar to SigMix but it more focus on the mobile network, so here we compare to SigMix, and leave the comparison in mobile network as a promising extension.
- **PhyCode.** PhyCode [17] requires no change at the transmitter side, but it assumes the constellation points are distinguishable when decoding the two-user superimposed signals.

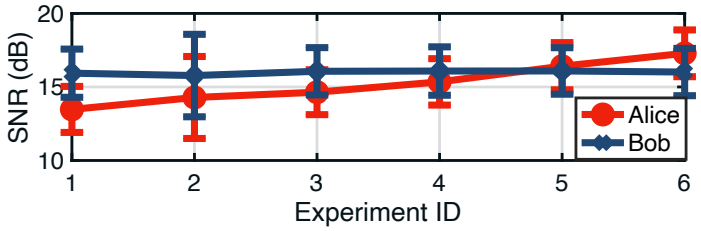
Note that both SigMix and PhyCode use the same offsets tracking mechanism as ChitChat for the implementation.

Metrics. We employ the following metrics:

- Bit Error Rate (BER): The ratio of averaged incorrect bits within one packet; here, the BER refers to the raw BER without considering channel error coding.
- Packet Reception Rate (PRR): The ratio of successfully received packets averaged over every 50 packets. We use the bit error information from all the packets received



(a) Different SNRs settings



(b) Different SNR ratios settings

Figure 5.14: The setting of the signal strength.

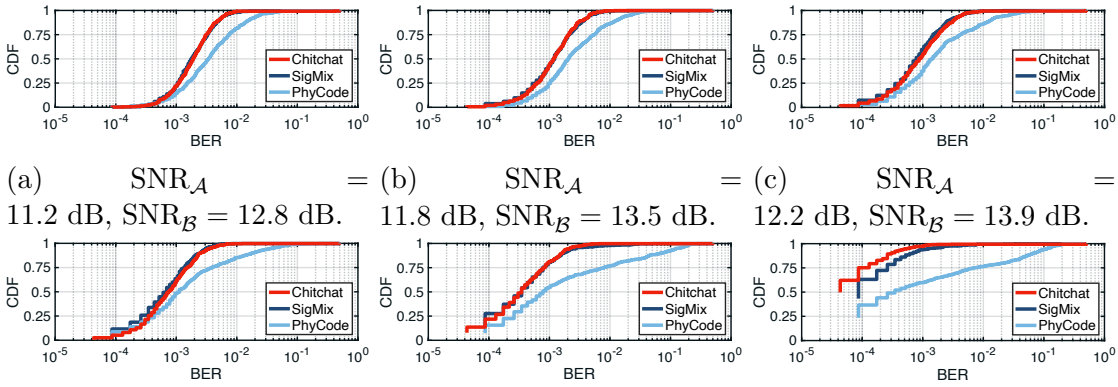
by the USRP and then apply the CRC-32 algorithm along with convolutional codes at 1/2 code rates to calculate the PRR.

- Throughput gain: the ratio of throughput in ChitChat to the throughput in the existing scheme (i.e., SigMix or Phycode).

5.6.1 The Impact of the Signal Strength

To demonstrate that ChitChat can perform well without power control, we evaluate the impact of both the SNR and the SNR gap of two transmitted signals. To focus on the influence of SNR and SNR gap, we use 30 dB attenuators to connect two transmitters to the receiver, which can emulate a stable wireless channel and avoid the impact of the dynamic environment. This experiment was conducted in an ordinary office. We collect 500 packets for each setting.

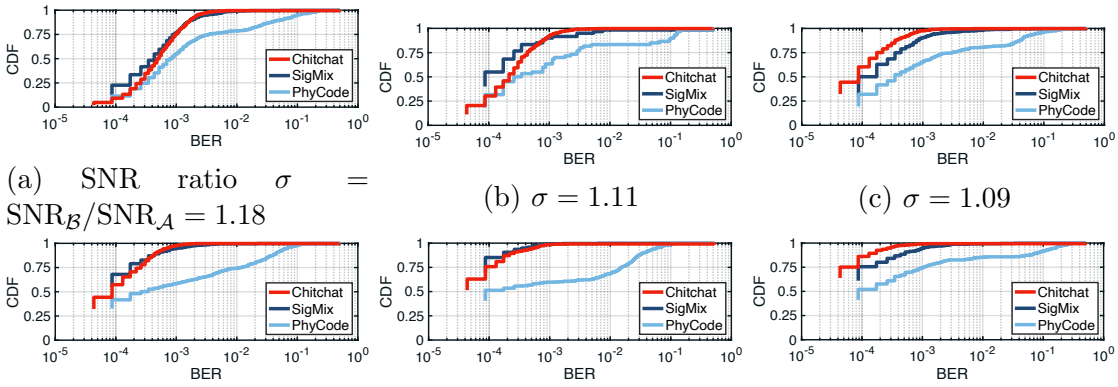
The Impact of the SNR. As shown in Fig. 5.14(a), we manually tune the SNR of Alice and Bob from 9.5 dB to 18 dB. The BER results are illustrated in Fig. 5.15. Clearly, for each SNRs setting, ChitChat outperforms PhyCode significantly. ChitChat achieves comparable performance with SigMix in most cases. As the SNR increases,



(a) $\text{SNR}_{\mathcal{A}} = 11.2$ dB, $\text{SNR}_{\mathcal{B}} = 12.8$ dB. (b) $\text{SNR}_{\mathcal{A}} = 11.8$ dB, $\text{SNR}_{\mathcal{B}} = 13.5$ dB. (c) $\text{SNR}_{\mathcal{A}} = 12.2$ dB, $\text{SNR}_{\mathcal{B}} = 13.9$ dB.

(d) $\text{SNR}_{\mathcal{A}} = 12.5$ dB, $\text{SNR}_{\mathcal{B}} = 14.1$ dB. (e) $\text{SNR}_{\mathcal{A}} = 13.5$ dB, $\text{SNR}_{\mathcal{B}} = 15$ dB. (f) $\text{SNR}_{\mathcal{A}} = 14.9$ dB, $\text{SNR}_{\mathcal{B}} = 16.4$ dB.

Figure 5.15: BER under different SNR settings.



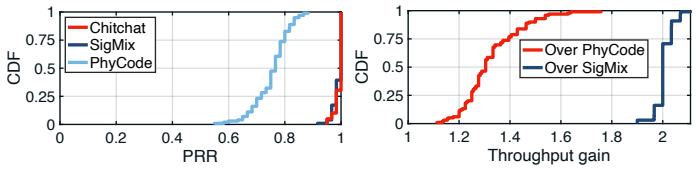
(a) SNR ratio $\sigma = \text{SNR}_{\mathcal{B}}/\text{SNR}_{\mathcal{A}} = 1.18$ (b) $\sigma = 1.11$ (c) $\sigma = 1.09$

(d) $\sigma = 1.05$ (e) $\sigma = 0.98$ (f) $\sigma = 0.93$

Figure 5.16: BER under under different SNR gaps.

the performance of ChitChat is much higher than PhyCode and even better than SigMix. For example, as shown in Fig. 5.15(f), when the SNR is around 15 dB, the BER results of ChitChat, SigMix and PhyCode are 4.34×10^{-5}, 8.68×10^{-5} and 2.6×10^{-4}, respectively. This experiment result reveals that ChitChat performs well within the typical SNR range.

The Impact of the SNR gap. We use SNR ratio as a metric to represent the SNR gap between transmitters. To evaluate the influence of a small SNR gap, we keep the ratio close to 1. Specifically, $\text{SNR}_{\mathcal{B}}$ is around 16 dB and $\text{SNR}_{\mathcal{A}}$ is gradually changed as shown in Fig. 5.14(b). We plot the BER in Fig. 5.16. Apparently, ChitChat still outperforms PhyCode and has the comparable performance of SigMix with the



(a) PRR (b) Throughput gain

Figure 5.17: The performance comparisons in static environments.

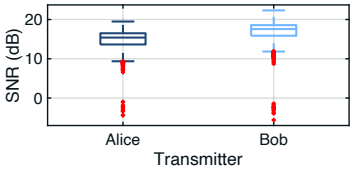


Figure 5.18: The SNR distributions in dynamic environments.

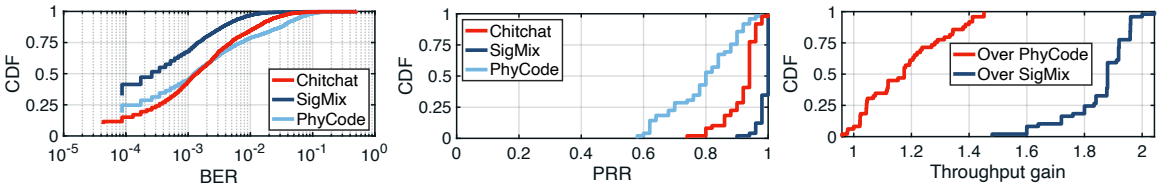
varying SNR ratio. The results reveal that ChitChat can perform well under a small SNR gap between transmitters, so ChitChat does not need the power control.

PRR. We then evaluate the impact of the signal strength from the packet level. We calculate the PRR according to the bit error information from the above two experiments. As plotted in Fig. 5.17(a), ChitChat can achieve a comparable PRR of SigMix with a median value of 1, while PhyCode only has a median value of 0.77. Hence, ChitChat can achieve a reliable performance under the typical signal strength.

Throughput gain. Fig. 5.17(b) shows that ChitChat achieves a median 1.3× higher throughput gain than PhyCode. Importantly, although ChitChat and SigMix have the comparable reliability performance, ChitChat achieves a median 2× higher throughput gain than SigMix. This is because SigMix requires to transmit two signal copies within one packet and only use one copy for decoding. The amount of information is therefore reduced to half. In contrast, ChitChat can achieve reliable performance without redundant symbols, which represents a significant improvement of the spectrum efficiency.

5.6.2 The Performance in Dynamic Environment

We next evaluate ChitChat in dynamic environments with 5 test regions as shown in Fig. 5.12. During the experiments, people in the environment worked as usual (e.g., walking around), which contributes to a dynamic channel condition. We plot the SNR of two concurrent transmitted signals in Fig. 5.18. As we can see, although the average



(a) BER (b) PRR (c) Throughput gain

Figure 5.19: The performance comparisons in dynamic environments.

SNR can be achieved to 14.97 dB and 17.12 dB for Alice and Bob, respectively, the variance of the SNR is very high as some outliers are below 0 dB, resulting in a highly dynamic channel condition. The overall performance is as follows.

BER and PRR. From Fig. 5.19(a) and Fig. 5.19(b), we see that although ChitChat has a similar median BER (1.38×10^{-3}) with PhyCode, 90 percent of BER for ChitChat is below 1.52×10^{-2} compared with 3.87×10^{-2} for PhyCode, which gives a higher median PRR, 0.94, for ChitChat compared with 0.8 for PhyCode in the dynamic environment. Note that both the BER and PRR of ChitChat is not as good as that of SigMix. This is reasonable since the SNRs vary below 0 dB in the dynamic environment SigMix actually transmitted the same symbol twice within each packet for a reliable performance, but at a cost of much lower throughput. In contrast, ChitChat can decode the two-user superimposed signal without redundant symbols.

Throughput gain. Fig. 5.19(c) shows that ChitChat achieves a median 1.88× and 1.18× higher throughput gain than SigMix and PhyCode, respectively. This is expected as ChitChat can achieve a reliable performance without introducing redundant information. Note that these results are slightly lower than the above experiments in the static environment. The reason is that the dynamic environment will cause more signal variations and SNR fluctuation. Overall, ChitChat can still outperform the state-of-the-art schemes even in the dynamic environment.

5.7 Conclusion

Guided by the similar work of Cover [7], both the academia and industry are inspired to apply superimposed signals to achieve a higher spectrum efficiency, and SIC-type of NOMA has been proposed as a candidate radio access technology for 5G cellular systems. As SIC is limited to deal with the situations that two concurrent transmitted signals have a significant SNR difference, in this chapter, we propose and implement Chitchat, a rotation based encoding and decoding system for two-transmitter NOMA, without the SNR difference constraint. It is inherently challenging to decode superimposed signals given the uncontrollable and time-varying channels which may make the constellation maps of the superimposed signals difficult and sometimes even infeasible to decode. By addressing the design and implementation challenges, this work nevertheless demonstrates the feasibility in this promising direction. It is anticipated that the modulation-level NOMA will lead to a wide range of applications, from radio access for infrastructure-based wireless systems to two-way relay for multi-hop wireless mesh networks, and even for collision resolution in random access networks, which can be seen as further research issues.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The growth of the wireless communication systems is at an unprecedented pace, especially for Internet-of-things (IoT), however, in the meantime, this growth is challenged by the limited spectrum that not originated to serve for the billions of ubiquitous IoT devices. In this dissertation, we start from a question: can we use the spectrum better so that we can push the limit of current spectrum efficiency, and eventually, make the spectrum ready for the prosper of the IoT? After a thoroughly study, we notice that a promising answer to this question is to enable Non-Orthogonal Multiple Access (NOMA) in which superimposed signals are exploited and multiple users share the spectrum instead of competing with each other. But NOMA was proposed for cellular networks, a direct use of current schemes for IoT devices may cause severer errors and the design goal may be infeasible for the heterogeneous and often low-cost IoT devices. To address this, in this dissertation, we propose NOMA schemes that are implementable and desirable to current IoT devices. Specifically, we present our study from four practical and essential dimensions——feasible to heterogeneous de-

vices, high reliability, robust to mobility, and high efficiency. Finally, we conclude the main contributions of each chapter as follows.

In Chapter 2, in dealing with heterogeneous devices, we present PhyCode, a new NOMA scheme for the first time to decode the two-user superimposed signal in modulation-level. Compared with current NOMA schemes, PhyCode requires no power control so it is more desirable for IoT devices. To make PhyCode practical, we introduce the first design to implement dynamic signal offsets tracking and reacting schemes to detect and decode two-user superimposed signals, making PhyCode practically feasible for heterogeneous IoT devices. We build a real system to validate the idea of PhyCode. The experimental results demonstrate that PhyCode can obtain a comparable performance in an ordinary office building with heterogeneous devices.

In Chapter 3, in dealing with the reliability challenge and based on the knowledge in Chapter 2, we propose SigMix, the first practical system that can reliably decode the two-user superimposed signals under dynamic channel conditions and hardware imperfections for IoT systems. We first derive an exact bit-error-rate (BER) expression to model the relationship between the channel conditions and the decoding error probability. Most importantly, for the first time, we understand the main decoding error factor of two-user superimposed signals—the indistinguishable constellation anchor points. With this detailed BER expression, we can have a clear view on how to avoid the indistinguishable constellation anchor points by manipulating the channel condition, and hence to obtain a reliable decoding performance. To this end, we propose a shifting code to achieve a substantial diversity gain and then propose an adaptive decoding scheme by considering both the dynamic channel conditions and the hardware imperfections. Last, we build a software-defined radio based platform to evaluate the performance of SigMix across various scenarios. The extensive experimental results illustrate that SigMix obtains a one-order lower median BER than the

state-of-the-art system.

In Chapter 4, in deal with the mobility challenge, we propose I-Talk, a new NOMA scheme that aims to implement modulation-level two-user superimposed signal decoding in the presence of both hardware imperfections and mobile channel conditions. First, we conduct experimental analysis to study the signal offsets caused by the mobility and the hardware imperfection. According to the study results, we propose to use a synthesis channel coefficient to represent all these offsets, and by doing so, we can trace all the offsets and then eliminate the side effects of these offsets, providing a stable synchronization performance. Second, for the first time, we reveal the complementary property of the constellation anchor points and then leverage it in decoding two-user superimposed signals in mobile channel environments. We implement I-Talk on a software-defined radio platform and evaluate its performance in mobile scenarios. Our extensive experimental results demonstrate that I-Talk achieves a one-order lower bit-error-rate and a 1.47× higher throughput gain in the mobile scenario, outperforming the state-of-the-art scheme.

In Chapter 5, in dealing with efficiency challenge, we introduce ChitChat, an effective encoding/decoding system for NOMA that is able to substantially improve the spectrum efficiency and achieve reliable performance. For the first time, we leverage a rotation code in encoding the transmitted signal, so that we can obtain the efficiency and the reliability simultaneously. Based on that, we then propose a weighted rotation code for further manipulating the encode scheme to avoid indistinguishable constellation anchor points at the receiver. We implement ChitChat on a software-defined radio platform and the extensive experimental results demonstrate that ChitChat outperforms the state-of-the-art scheme with a BER of 4.34×10^{-5} and 1.38×10^{-3} and a throughput gain of 2 × higher and 1.38× higher in static channel environments and dynamic channel environments, respectively.

6.2 Future Work

We believe the findings in this dissertation would lead a novel and promising direction in designing NOMA schemes, and pave the way for the coming prosperity of the wireless communication systems. We list some of the future works as follows.

Chapter 2: To further improve our work, PhyCode can be extended to support a large number of sources concurrently transmitting signals, as a sign of the scalability, where a scheduling algorithm for the pilot sample assignment would be considered to reduce the offset tracking overhead. We can further apply the channel and error coding to improve the BER. Also, we can design a distributed time management scheme to fully support multi-source and multi-hop scenarios.

Chapter 3: **Soft-value decoding.** In SigMix, we select one copy whose phase shift is in the 'good regions' to decode the two-user superimposed signal. To further utilize the diversity gain of our scheme, we can apply many soft-value combining schemes to further reduce the BER [98, 99] under strong environmental noise. Specifically, two copies can be utilized together based on their soft values to decode the superimpose signal. We leave this part of work as one of our future work.

MAC-layer design and other coding schemes. So far, the implementation of SigMix is focused on the physical layer. To further boost the performance and expand the applications of our system, more research on how to apply sophisticated channel coding and MAC-layer protocols to our system is needed. With the newly proposed testbed [100], we can further implement MAC-layer protocols and we leave this for further work.

Scale to a large network. We implement SigMix in a two-transmitter system. By leveraging a fine-designed orthogonal preamble and distributed time clock [16], e.g., GPS clock, we can involve more transmitters to transmit signal concurrently and decode the two-user superimposed signal. Moreover, involving more antennas with

more RF chains can also enable us to utilize the spatial spectrum resource. This will be an important future research issue.

Chapter 4: **Various modulation schemes.** The IoT related communication techniques such as RFID, LoRa, ZigBee and WiFi are attracting the most attention. In particular, RFID, LoRa and ZigBee are operating on narrow-band carriers, designed for applications with low-date-rate, in a range of tens of Kbps to hundreds of Kbps. WiFi signals operates on wide-band carriers with the OFDM technology, achieving a data rate of a few Mbps even with the lowest modulation scheme—BPSK modulation. In this chapter, we devote our efforts to enable concurrent transmission of WiFi (IEEE 802.11) signals modulated by BPSK. We believe the data rate provided by the BPSK WiFi signal is sufficient for the majority of IoT applications. Moreover, BPSK modulated signals are more robust to complicated channel conditions, making our scheme feasible for a wider range of applications. On the other hand, a higher order of modulation can provide further enhancement to our scheme in terms of the data rate at the cost of higher complexity. If a future application prefers a higher-order modulation, the idea proposed in this chapter can be further extended to support it. This is because the symmetric property among the constellation points still exist, which can ensure a stable decoding performance for I-Talk.

Decoding superimposed for two-user scenarios. Although NOMA was proposed to address a massive number of users, the existing SIC-based NOMA implementations are also focusing on two-user concurrent transmission cases [9, 101–104]. The underlying reason for this is that maintaining a power gap between users is challenging in mobile environments. For more users, the popular solution is to apply grouping or other scheduling solutions to arrange the two-user NOMA communication only [101].

A large-scale testbed with many communication nodes. The implementation of our work includes few communication nodes due to the limited number of the

experimental devices, i.e., testing vehicles and USRPs. We believe the study of this work can be seen as a solid foundation for further researches, where more devices can be involved.

A promising application to combat collisions. Collisions in wireless communication can cause severe performance degradation. To address this problem, like existing NOMA technologies [9], I-Talk can group a large number of users into subgroups with a smaller size, say two users within each subgroup, so that the overall collisions would decrease and the throughput can be increased substantially. By exploiting multipath, the natural mode of operation for RFIDs, PinIt presents a solution based on dynamic time warping that achieves 16 cm 90th percentile accuracy and 25 cm 99th per- centile accuracy, enabling reliable centimeter-scale positioning in densely packed environments.

Chapter 5: **Under a lower SNR.** The decoding part in ChitChat is genuinely more difficult than the schemes without the rotation code, as more constellation anchor points exist in the constellation map. When the SNR is low, say below 10 dB [105], the indistinguishable constellation anchor points will not be the major reason for the decoding errors. Instead, it is transferred to environmental noises, which may weaken the performance of ChitChat. In such a case, ChitChat can apply higher layer techniques, i.e., error coding, or use more antennas for a higher diversity gain, to work complementarily in solving the problem.

High-order modulations. The implementation of ChitChat focused on wireless transmissions with the BPSK modulation, and we believe that this data rate is sufficient for the majority of IoT applications. Although a higher order of modulation, say 16-QAM, can widen the application scenarios, it also at a cost of introducing practice problems, e.g., a tight signal strength restriction which may lead to heavy burden to IoT devices. We leave the trade-off between the modulation order and the

practice challenges as future work.

Appendix A

The Detailed Derivation of the BER Expression

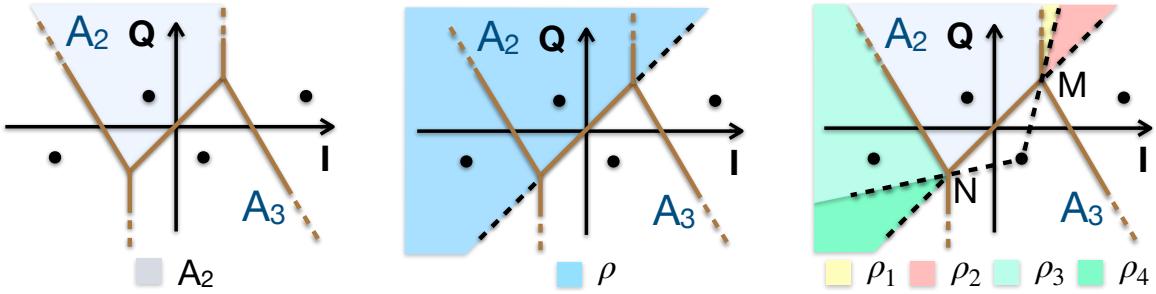
A.1 BER Expression in Case 1

Under the condition that $|H_1| \geq |H_2|$ and $0 \leq \phi \leq \frac{\pi}{2}$, we can further divide this condition into three cases based on the phase shift ϕ as follows [66]

$$\begin{cases} \text{Case 1: } \cos\phi > \frac{|H_1|}{2|H_2|}, \\ \text{Case 2: } \frac{|H_2|}{2|H_1|} \leq \cos\phi \leq \frac{|H_1|}{2|H_2|}, \\ \text{Case 3: } \cos\phi < \frac{|H_2|}{2|H_1|}. \end{cases} \tag{A.1}$$

Given each case, the boundaries of decision regions are now fixed in Craig's polar coordinate. So, we can calculate $P(A_2|A_3)$ for the three cases, separately.

To divide the decision regions properly, let us start from case 1. As shown in Fig. A.1, to calculate $P(A_2|A_3)$, we design a two-step partition method. In the first step, as shown in Fig. A.1(b), we extend the boundary between region A_2 and region A_3. By doing so, we can obtain a new region in the constellation map, i.e., region



(a) Origin (b) Step 1 (c) Step 2

Figure A.1: The two-step partition method for case 1.

ρ. In the second step, as shown in Fig. A.1(c), we connect the location of "10" to the two intersections (i.e., point M and point N) and extend these two lines. Then, we obtain region ρ_1, ρ_2, ρ_3, and ρ_4 respectively. Obviously, each region ρ_i is a unit shape. The region ρ contains five regions including region A_2, region ρ_1, ρ_2, ρ_3, and ρ_4. Therefore, we have

$$P(A_2|A_3) = P(\rho|A_3) - \sum_{i=1}^{4} P(\rho_i|A_3), \tag{A.2}$$

where, $P(\rho|A_3)$ can be calculated by [69]. Next, we can directly apply Craig's analytical model to calculate $P(\rho_i|A_3)$ since each region ρ_i is a unit shape. Here, taking $P(\rho_1|A_3)$ as an example (see Fig. A.2), we have

$$P(\rho_1|A_3) = \frac{1}{2\pi} \int_0^{\omega_1} \exp\left(-\frac{{R_1}^2(\theta)}{2\sigma^2}\right) d\theta, \tag{A.3}$$

where ${R_1}^2(\theta)$ and ω_1 can be calculated based on analytic geometry [66], i.e., ${R_1}^2(\theta) = \frac{(|H_1|-2|H_2|\cos\phi)^2}{\sin^2\theta}$ and $\omega_1 = \pi/2 - \alpha - \phi$. Here, $\alpha = \arctan\frac{|H_1|-|H_2|\cos\phi}{|H_2|\sin\phi}$. Similar to the calculation of $P(\rho_1|A_3)$, we can calculate $P(\rho_i|A_3), i = 2, 3, 4$ in the same way. Specifically, we first derive the expressions of ${R_i}^2(\theta), i = 2, 3, 4$ based on simple analytic geometry, and then we can obtain the expression of $P(\rho_i|A_3)$ based on Eq. 3.3

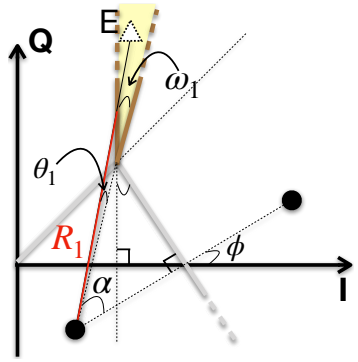


Figure A.2: **Zoom in to region ρ_1.**

as follows

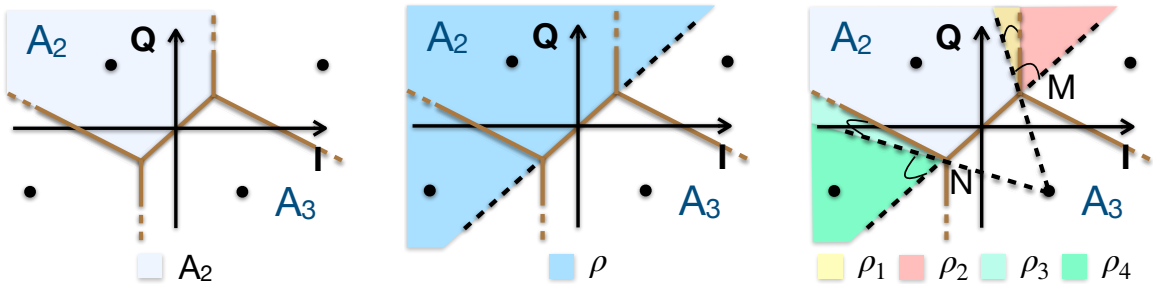
$$\begin{cases} P(\rho_2|A_3) = \frac{1}{2\pi}\int_0^{\omega_2} \exp[-\frac{|H_1|^2-2|H_1||H_2|\cos\phi+|H_2|^2}{2\sigma^2\sin^2\theta}]d\theta \\ P(\rho_3|A_3) = \frac{1}{2\pi}\int_0^{\omega_3} \exp[-\frac{(2|H_1|\cos\phi-|H_2|)^2}{2\sigma^2\sin^2\theta}]d\theta \\ P(\rho_4|A_3) = \frac{1}{2\pi}\int_0^{\omega_4} \exp[-\frac{|H_1|^2-2|H_1||H_2|\cos\phi+|H_2|^2}{2\sigma^2\sin^2\theta}]d\theta \end{cases}, \tag{A.4}$$

where ω_i can be also calculated using analytic geometry as follows

$$\begin{cases} \omega_2 = \arcsin(\frac{\cos\alpha\sqrt{|H_1|^2-2|H_1||H_2|\cos\phi+|H_2|^2}}{|H_2|}) \\ \omega_3 = \alpha + 2\arccos(\frac{\cos\alpha\sqrt{|H_1|^2-2|H_1||H_2|\cos\phi+|H_2|^2}}{|H_2|}) - \frac{\pi}{2} \\ \omega_4 = \arcsin(\frac{\cos\alpha\sqrt{|H_1|^2-2|H_1||H_2|\cos\phi+|H_2|^2}}{|H_2|}) \end{cases}. \tag{A.5}$$

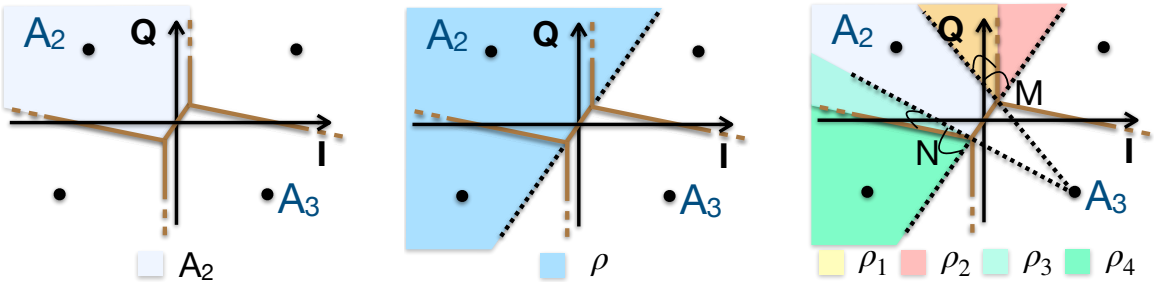
So far we have calculated $P(A_2|A_3)$ for case 1. By placing Eq. A.2, Eq. A.3 and combing the calculation results in [66, 69], we can derive the detailed expression of P for case 1.

Similarly, for case 2 and case 3, we can also divide all the decision regions into several unit shapes and repeat the above derivation process to calculate $P(A_2|A_3)$. We show the partition process and expressions in the following subsection.



(a) Origin (b) Step 1 (c) Step 2

Figure A.3: Region partition for case 2.



(a) Origin (b) Step 1 (c) Step 2

Figure A.4: Region partition for case 3.

A.2 Region Partition for Case 2 and Case 3

For case 2, we also use the proposed two-step partition method to divide the decision regions into several unit shapes. As shown in Fig. A.3(b) and Fig. A.3(c), region ρ contains four regions including region A_2, ρ_2, ρ_3 and ρ_4. We denote the overlapped region between A_2 and ρ_2 as ρ_1. So, we have

$$P(A_2|A_3) = P(\rho|A_3) - P(\rho_2|A_3) - P(\rho_3|A_3) - P(\rho_4|A_3) + P(\rho_1|A_3). \tag{A.6}$$

From Fig. A.3(c), we note that only the shape of region ρ_1 has been changed compared with case 1 (see Fig. A.1(c)). Therefore, for case 2, we only need to update $P(\rho_1|A_3)$ as follows, while other items are the same with that of case 1.

$$P(\rho_1|A_3) = \frac{1}{2\pi}\int_0^{\omega_1} \exp[-\frac{(|H_1| - 2|H_2|\cos\phi)^2}{2\sigma^2\sin^2\theta}]d\theta, \tag{A.7}$$

where ω_1 is updated as $\omega_1 = \alpha + \phi - \frac{\pi}{2}$.

Similarly, for case 3, the region partition is shown in Fig. A.4. Obviously, region ρ contains three regions including region A_2, ρ_2 and ρ_4. We denote the overlapped region between ρ_2 and A_2 as ρ_1, while the overlapped region between ρ_4 and A_2 as ρ_3. So, we have

$$P(A_2|A_3) = P(\rho|A_3) - P(\rho_2|A_3) - P(\rho_4|A_3) + P(\rho_1|A_3) + P(\rho_3|A_3). \tag{A.8}$$

From Fig. A.4(c), we observe that only the shape of region ρ_3 has been changed compared with case 2 in Fig. A.3(c). Therefore, in Eq. A.8, $P(\rho|A_3)$, $P(\rho_2|A_3)$ and $P(\rho_4|A_3)$ are the same as that of both case 1 and case 2, while $P(\rho_1|A_3)$ is the same as that of case 2. So, for case 3, we only need to update $P(\rho_3|A_3)$ as

$$P(\rho_3|A_3) = \frac{1}{2\pi}\int_0^{\omega_3} \exp[-\frac{(|H_2| - 2|H_1|\cos\phi)^2}{2\sigma^2 \sin^2\theta}]d\theta, \tag{A.9}$$

where ω_3 can be updated as

$$\omega_3 = \frac{\pi}{2} - \alpha - 2\arccos(\frac{\cos\alpha\sqrt{|H_1|^2 - 2|H_1||H_2|\cos\phi + |H_2|^2}}{|H_2|}). \tag{A.10}$$

Bibliography

[1] Cisco VNI Forecast et al. Cisco visual networking index: Global mobile data traffic forecast update 2016-2021. *Cisco Public Information*, 2017.

[2] Marcus Torchia and Michael Shirer. IDC forecasts worldwide technology spending on the Internet of Things to reach $1.2 trillion in 2022. *IDC Media Center*, 2018.

[3] Athar Ali Khan, Mubashir Husain Rehmani, and Abderrezak Rachedi. Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions. *IEEE Wireless Communications*, 24(3):17–25, 2017.

[4] Xin Liu, Min Jia, Xueyan Zhang, and Weidang Lu. A novel multichannel internet of things based on dynamic spectrum sharing in 5G communication. *IEEE Internet of Things Journal*, 6(4):5962–5970, 2018.

[5] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[6] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[7] Thomas Cover. Broadcast channels. *IEEE Transactions on Information Theory*, 18(1):2–14, 1972.

[8] P Bergmans and Thomas Cover. Cooperative broadcasting. *IEEE Transactions on Information Theory*, 20(3):317–324, 1974.

[9] Yuya Saito, Yoshihisa Kishiyama, Anass Benjebbour, Takehiro Nakamura, Anxin Li, and Kenichi Higuchi. Non-orthogonal multiple access (NOMA) for cellular future radio access. In *2013 IEEE 77th vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2013.

[10] Linglong Dai, Bichai Wang, Yifei Yuan, Shuangfeng Han, I Chih-Lin, and Zhaocheng Wang. Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends. *IEEE Communications Magazine*, 53(9):74–81, 2015.

[11] Zhiguo Ding, Mugen Peng, and H Vincent Poor. Cooperative non-orthogonal multiple access in 5G systems. *IEEE Communications Letters*, 19(8):1462–1465, 2015.

[12] Zhiguo Ding, Zheng Yang, Pingzhi Fan, and H Vincent Poor. On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users. *IEEE signal processing letters*, 21(12):1501–1505, 2014.

[13] Kenichi Higuchi and Anass Benjebbour. Non-orthogonal multiple access (NOMA) with successive interference cancellation for future radio access. *IEICE Transactions on Communications*, 98(3):403–414, 2015.

[14] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Successive interference cancellation: A back-of-the-envelope perspective. In

Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, page 17. ACM, 2010.

[15] Shyamnath Gollakota, Samuel David Perli, and Dina Katabi. Interference alignment and cancellation. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 159–170, 2009.

[16] Tanmoy Das, Lu Chen, Rupam Kundu, Arjun Bakshi, Prasun Sinha, Kannan Srinivasan, Gaurav Bansal, and Takayuki Shimizu. Corecast: Collision resilient broadcasting in vehicular networks. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 217–229. ACM, 2018.

[17] Wen Cui, Chen Liu, Lin Cai, and Jianping Pan. PhyCode: A practical wireless communication system exploiting superimposed signals. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[18] Wen Cui, Chen Liu, Hamed Mosavat-Jahromi, and Lin Cai. SigMix: Decoding superimposed signals for IoT. *IEEE Internet of Things Journal*, 7(4):3026–3040, 2020.

[19] Wen Cui, Chen Liu, Wenjun Yang, and Lin Cai. I-Talk: Reliable and practical superimposed signal decoding without power control. *Under revision of IEEE Transactions on Wireless Communications*, 2020.

[20] Wen Cui, Chen Liu, and Lin Cai. Chitchat: Efficient and reliable decoding of two-transmitter superimposed signals for IoT. *Submitted to IEEE Internet of Things Journal*, 2020.

[21] Shengli Zhang, Soung Chang Liew, and Patrick P Lam. Hot topic: Physical-layer network coding. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 358–365. ACM, 2006.

[22] Lu Lu, Taotao Wang, Soung Chang Liew, and Shengli Zhang. Implementation of physical-layer network coding. *Physical Communication*, 6:74–87, 2013.

[23] Lizhao You, Soung Chang Liew, and Lu Lu. Reliable physical-layer network coding supporting real applications. *IEEE Transactions on Mobile Computing*, 16(8):2334–2350, 2016.

[24] IEEE Standards Association et al. 802.11 p-2010-IEEE standard for information technology-local and metropolitan area networks-specific requirements-part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications amendment 6: Wireless access in vehicular environments. *URL: http://standards. ieee. org/findstds/standard/802.11 p-2010. html*, 2010.

[25] Daniel Jiang and Luca Delgrossi. IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008-IEEE Vehicular Technology Conference*, pages 2036–2040. IEEE, 2008.

[26] Rob Alderfer, Dirk Grunwald, and Kenneth Baker. Optimizing DSRC safety efficacy and spectrum utility in the 5.9 Ghz band. *Available at SSRN 2841346*, 2014.

[27] Jhihoon Joo, Myung Chul Park, Dong Seog Han, and Veljko Pejovic. Deep learning-based channel prediction in realistic vehicular communications. *IEEE Access*, 7:27846–27858, 2019.

[28] Haoyuan Pan, Lu Lu, and Soung Chang Liew. Practical power-balanced non-orthogonal multiple access. *IEEE Journal on Selected Areas in Communications*, 35(10):2312–2327, 2017.

[29] Lizhao You, Soung Chang Liew, and Lu Lu. Network-coded multiple access II: Toward real-time operation with improved performance. *IEEE Journal on Selected Areas in Communications*, 33(2):264–280, 2014.

[30] Lu Lu, Lizhao You, and Soung Chang Liew. Network-coded multiple access. *IEEE Transactions on Mobile Computing*, 13(12):2853–2869, 2014.

[31] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. Embracing wireless interference: Analog network coding. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 397–408, 2007.

[32] Lu Chen, Fei Wu, Jiaqi Xu, Kannan Srinivasan, and Ness Shroff. BiPass: Enabling end-to-end full duplex. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 114–126. ACM, 2017.

[33] Haoyuan Zhang and Lin Cai. Bi-directional multi-hop wireless pipeline using physical-layer network coding. *IEEE Transactions on Wireless Communications*, 16(12):7950–7965, 2017.

[34] Haoyuan Zhang and Lin Cai. Design of channel coded heterogeneous modulation physical layer network coding. *IEEE Transactions on Vehicular Technology*, 67(3):2219–2230, 2017.

[35] Claude E Shannon. Two-way communication channels. In *4th Berkeley Symposium on Math. Stat. and Prob.*, 1961.

[36] Linghe Kong and Xue Liu. mZig: Enabling multi-packet reception in ZigBee. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 552–565. ACM, 2015.

[37] Heinrich Meyr, Marc Moeneclaey, and Stefan Fechtel. *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. John Wiley & Sons, Inc., 1997.

[38] Michael A Lombardi. Fundamentals of Time and Frequency. *The Mechatronics Handbook*, 2002.

[39] Veluppillai Mahinthan, Lin Cai, Jon W Mark, and Xuemin Shen. Partner selection based on optimal power allocation in cooperative-diversity systems. *IEEE Transactions on Vehicular Technology*, 57(1):511–520, 2008.

[40] Ezzeldin Hamed, Hariharan Rahul, and Bahar Partov. Chorus: Truly distributed distributed-MIMO. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 461–475, 2018.

[41] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. SourceSync: A distributed wireless architecture for exploiting sender diversity. *ACM SIGCOMM Computer Communication Review*, 41(4):171–182, 2011.

[42] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single WiFi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, 2016.

[43] Jue Wang, Haitham Hassanieh, Dina Katabi, and Piotr Indyk. Efficient and reliable low-power backscatter networks. *ACM SIGCOMM Computer Communication Review*, 42(4):61–72, 2012.

[44] Meng Jin, Yuan He, Xin Meng, Yilun Zheng, Dingyi Fang, and Xiaojiang Chen. Fliptracer: Practical parallel decoding for backscatter communication. *IEEE/ACM Transactions on Networking (TON)*, 27(1):330–343, 2019.

[45] Meng Jin, Yuan He, Xin Meng, Dingyi Fang, and Xiaojiang Chen. Parallel backscatter in the wild: When burstiness and randomness play with you. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 471–485, 2018.

[46] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321. ACM, 2017.

[47] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. Netscatter: Enabling large-scale backscatter networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 271–284, 2019.

[48] Johannes Schmitz, Felix Bartsch, Manuel Hernández, and Rudolf Mathar. Distributed software defined radio testbed for real-time emitter localization and tracking. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1246–1252. IEEE, 2017.

[49] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 159–170, 2008.

[50] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. Performance assessment of IEEE 802.11 p with an open source SDR-based prototype. *IEEE Transactions on Mobile Computing*, 17(5):1162–1175, 2017.

[51] Timothy M Schmidl and Donald C Cox. Robust frequency and timing synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, 1997.

[52] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *ACM SIGCOMM Computer Communication Review*, 41(4):159–170, 2011.

[53] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. LiFS: Low human-effort, device-free localization with fine-grained subcarrier information. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 243–256. ACM, 2016.

[54] Wenjie Zhou, Tanmoy Das, Lu Chen, Kannan Srinivasan, and Prasun Sinha. Basic: Backbone-assisted successive interference cancellation. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 149–161. ACM, 2016.

[55] Mansoor Shafi, Andreas F Molisch, Peter J Smith, Thomas Haustein, Peiying Zhu, PD Silva, Fredrik Tufvesson, Anass Benjebbour, and Gerhard Wunder. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE Journal on Selected Areas in Communications*, 2017.

[56] Pan Hu, Pengyu Zhang, and Deepak Ganesan. Laissez-faire: Fully asymmetric backscatter communication. *ACM SIGCOMM Computer Communication Review*, 45(4):255–267, 2015.

[57] Zhenyu Song, Longfei Shangguan, and Kyle Jamieson. Wi-Fi goes to town: Rapid picocell switching for wireless transit networks. In *Proceedings of the*

Conference of the ACM Special Interest Group on Data Communication, pages 322–334. ACM, 2017.

[58] Edward C Van Der Meulen. Three-terminal communication channels. *Advances in applied Probability*, 3(1):120–154, 1971.

[59] Alice Yen-Chi Peng, Shahram Yousefi, and Il-Min Kim. On error analysis and distributed phase steering for wireless network coding over fading channels. *IEEE Transactions on Wireless Communications*, 8(11):5639–5649, 2009.

[60] Thomas H Lee and Ali Hajimiri. Oscillator phase noise: A tutorial. *IEEE journal of solid-state circuits*, 35(3):326–336, 2000.

[61] Haoyuan Pan, Lu Lu, and Soung Chang Liew. Network-coded multiple access with high-order modulations. *IEEE Transactions on Vehicular Technology*, 66(11):9776–9792, 2017.

[62] Haoyuan Pan, Soung Chang Liew, Jiaxin Liang, Yulin Shao, and Lu Lu. Network-coded multiple access on unmanned aerial vehicle. *IEEE Journal on Selected Areas in Communications*, 36(9):2071–2086, 2018.

[63] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the limits of LoRaWAN. *IEEE Communications magazine*, 55(9):34–40, 2017.

[64] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. Farmbeats: An IoT platform for data-driven agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 515–529, 2017.

[65] Haitham Hassanieh, Omid Abari, Michael Rodriguez, Mohammed Abdelghany, Dina Katabi, and Piotr Indyk. Fast millimeter wave beam alignment. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 432–445, 2018.

[66] Moonseo Park, Ilhwan Choi, and Inkyu Lee. Exact BER analysis of physical layer network coding for two-way relay channels. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2011.

[67] John W Craig. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. In *MILCOM 91-Conference record*, pages 571–575. IEEE, 1991.

[68] John G Proakis and Masoud Salehi. *Digital Communications*. McGraw-hill New York, 2001.

[69] Kejie Lu, Shengli Fu, Yi Qian, and Hsiao-Hwa Chen. SER performance analysis for physical layer network coding over AWGN channels. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–6. IEEE, 2009.

[70] Ville Syrjala, Mikko Valkama, Nikolay N Tchamov, and Jukka Rinne. Phase noise modelling and mitigation techniques in OFDM communications systems. In *2009 Wireless Telecommunications Symposium*, pages 1–7. IEEE, 2009.

[71] Ezzeldin Hamed, Hariharan Rahul, Mohammed A Abdelghany, and Dina Katabi. Real-time distributed MIMO systems. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 412–425, 2016.

[72] Renjie Zhao, Fengyuan Zhu, Yuda Feng, Siyuan Peng, Xiaohua Tian, Hui Yu, and Xinbing Wang. OFDMA-Enabled Wi-Fi backscatter. In *The 25th Annual*

International Conference on Mobile Computing and Networking, pages 1–15. ACM, 2019.

[73] Michael R Souryal, Luke Klein-Berndt, Leonard E Miller, and Nader Moayeri. Link assessment in an indoor 802.11 network. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, volume 3, pages 1402–1407. IEEE, 2006.

[74] Haoyuan Zhang, Lei Zheng, and Lin Cai. Design and analysis of hierarchical physical layer network coding. *IEEE Transactions on Wireless Communications*, 16(12):7966–7981, 2017.

[75] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. Challenge: Unlicensed LPWANs are not yet the path to ubiquitous connectivity. In *The 25th Annual International Conference on Mobile Computing and Networking*, page 43. ACM, 2019.

[76] Siavash M Alamouti. A simple transmit diversity technique for wireless communications. *IEEE Journal on selected areas in communications*, 16(8):1451–1458, 1998.

[77] Vahid Tarokh, Hamid Jafarkhani, and A Robert Calderbank. Space-time block codes from orthogonal designs. *IEEE Transactions on Information theory*, 45(5):1456–1467, 1999.

[78] Hamid Jafarkhani. A quasi-orthogonal space-time block code. *IEEE Transactions on Communications*, 49(1):1–4, 2001.

[79] Jaewon Bae, Ki-Hun Lee, Jong Min Kim, Bang Chul Jung, and Jingon Joung. Performance analysis of uplink NOMA-IoT networks with space-time line code.

In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5. IEEE, 2019.

[80] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282, 2015.

[81] Titus KY Lo. Maximum ratio transmission. In *1999 IEEE International Conference on Communications (Cat. No. 99CH36311)*, volume 2, pages 1310–1314. IEEE, 1999.

[82] Hisham A Mahmoud and Huseyin Arslan. Error vector magnitude to snr conversion for nondata-aided receivers. *IEEE Transactions on Wireless Communications*, 8(5):2694–2704, 2009.

[83] Apostolos Georgiadis. Gain, phase imbalance, and phase noise effects on error vector magnitude. *IEEE Transactions on Vehicular Technology*, 53(2):443–449, 2004.

[84] Yue Li and Lin Cai. Cooperative device-to-device communication for uplink transmission in cellular system. *IEEE Transactions on Wireless Communications*, 17(6):3903–3917, 2018.

[85] 3GPP TR 36.888. Study on provision of low-cost Machine-Type Communications (MTC) user equipments (UEs) based on LTE. 2013.

[86] Sung-Min Oh and JaeSheung Shin. An efficient small data transmission scheme in the 3gpp nb-iot system. *IEEE Communications Letters*, 21(3):660–663, 2016.

[87] Mahyar Shirvanimoghaddam, Mischa Dohler, and Sarah J Johnson. Massive non-orthogonal multiple access for cellular IoT: Potentials and limitations. *IEEE Communications Magazine*, 55(9):55–61, 2017.

[88] Dinesh Bharadia and Sachin Katti. FastForward: Fast and constructive full duplex relays. *ACM SIGCOMM Computer Communication Review*, 44(4):199–210, 2015.

[89] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. AirExpress: Enabling seamless in-band wireless multi-hop transmission. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 566–577. ACM, 2015.

[90] Lu Chen, Fang Liu, and Kannan Srinivasan. Verification: Constructive and destructive full duplex relays. In *The 25th Annual International Conference on Mobile Computing and Networking*, page 44. ACM, 2019.

[91] Yunfei Ma, Nicholas Selby, and Fadel Adib. Drone relays for battery-free networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 335–347. ACM, 2017.

[92] Meng Jin, Yuan He, Xin Meng, Yilun Zheng, Dingyi Fang, and Xiaojiang Chen. Fliptracer: Practical parallel decoding for backscatter communication. *IEEE/ACM Transactions on Networking*, 27(1):330–343, 2019.

[93] Tanmoy Das and Prasun Sinha. ADS: Accurate decoding of RFID tags at scale. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 207–219, 2019.

[94] Ali Zarei Ghanavati and Daniel C Lee. Tight bound on the error probability of rotation code in rayleigh fading channels. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5. IEEE, 2014.

[95] Chen-Yu Hsu, Rumen Hristov, Guang-He Lee, Mingmin Zhao, and Dina Katabi. Enabling identification and behavioral sensing in homes using radio reflections.

In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[96] Chen Liu, Jie Xiong, Lin Cai, Lin Feng, Xiaojiang Chen, and Dingyi Fang. Beyond respiration: Contactless sleep sound-activity recognition using rf signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–22, 2019.

[97] Chen Liu, Dingyi Fang, Zhe Yang, Hongbo Jiang, Xiaojiang Chen, Wei Wang, Tianzhang Xing, and Lin Cai. Rss distribution-based passive localization and its application in sensor networks. *IEEE Transactions on Wireless Communications*, 15(4):2883–2895, 2015.

[98] Bo Han, Aaron Schulman, Francesco Gringoli, Neil Spring, Bobby Bhattacharjee, Lorenzo Nava, Lusheng Ji, Seungjoon Lee, and Robert Miller. Maranello: Practical partial packet recovery for 802.11. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, pages 14–14, 2010.

[99] Kate Ching-Ju Lin, Nate Kushman, and Dina Katabi. ZipTx: Harnessing partial packets in 802.11 networks. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 351–362, 2008.

[100] Haoyang Wu, Tao Wang, Zengwen Yuan, Chunyi Peng, Zhiwei Li, Zhaowei Tan, Boyan Ding, Xiaoguang Li, Yuanjie Li, Jun Liu, et al. The tick programmable low-latency SDR system. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 101–113, 2017.

[101] SM Riazul Islam, Ming Zeng, Octavia A Dobre, and Kyung-Sup Kwak. Nonorthogonal Multiple Access (NOMA): How it meets 5G and beyond. *Wiley 5G Ref: The Essential 5G Reference Online*, pages 1–28, 2019.

[102] Yuanwei Liu, Zhijin Qin, Maged Elkashlan, Zhiguo Ding, Arumugam Nallanathan, and Lajos Hanzo. Non-orthogonal multiple access for 5G and beyond. *Proceedings of the IEEE*, 105(12):2347–2381, 2017.

[103] Xiong Xiong, Wei Xiang, Kan Zheng, Hengyang Shen, and Xingguang Wei. An open source sdr-based noma system for 5G networks. *IEEE Wireless Communications*, 22(6):24–32, 2015.

[104] Anass Benjebbour, Keisuke Saito, Anxin Li, Yoshihisa Kishiyama, and Takehiro Nakamura. Non-orthogonal multiple access (NOMA): Concept, performance evaluation and experimental trials. In *2015 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6. IEEE, 2015.

[105] Jim Geier. Wi-Fi: Define minimum SNR values for signal coverage. *Enterprise Networking Planet: Standards & Protocols*, 2008.