Joint Multiple Parameter Estimation and Channel Decoding for Physical-layer Network Coding and Multiuser Detection

WANG, Taotao

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
April 2015

Abstract of thesis entitled:

Joint Multiple Parameter Estimation and Channel Decoding for Physical-layer Network Coding and Multiuser Detection

Submitted by WANG, Taotao

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in April 2015

This thesis investigates the joint multiple parameter estimation and channel decoding problem for physical-layer network coding (PNC) and multiuser detection (MUD) systems. Both of PNC and MUD can take advantages from the simultaneous transmissions by multiple users. However, the superimposition of multiple transmissions brings with it new challenges for signal processing. The first major challenge is the estimation of the multiple parameters at the receiver. The second major challenge is how to compensate for system impairments caused by these parameters. This thesis consists of two parts that tackle these challenges: The first part is related to PNC systems and the second part is related to MUD systems.

Part I:

The first part of this thesis addresses the problem of joint channel estimation and channel decoding in PNC systems. In PNC, multiple

users transmit to a relay simultaneously. PNC channel decoding is different from conventional multiuser channel decoding: Specifically, the PNC relay aims to decode a network-coded message rather than the individual messages of the users. Although prior work has shown that PNC can significantly improve the throughput of a relay network, the improvement is predicated on the availability of accurate channel estimates. Channel estimation in PNC, however, can be particularly challenging because of 1) the overlapped signals of multiple users; 2) the correlations among data symbols induced by channel coding; and 3) time-varying channels. We combine the expectation-maximization (EM) algorithm and belief propagation (BP) algorithm on a unified factor-graph framework. In this framework, channel estimation is performed by an EM subgraph, and channel decoding is performed by a BP subgraph that models a virtual encoder matched to the target of PNC channel decoding. Iterative message passing between these two subgraphs allows the optimal solutions for both to be approached progressively. We present extensive simulation results demonstrating the superiority of our PNC receivers over other PNC receivers.

Part II:

The second part of this thesis investigates a channel-coded MUD system operated with orthogonal frequency division multiplexing (OFDM) and interleaved division multiple-access (IDMA). In general, there are many variations to MUD systems. Our choice of the combination of OFDM and IDMA is motivated by its ability to achieve multiuser diversity gain in frequency-selective multiple-

access channels. However, to realize this potential advantage of OFDM-IDMA, we must first solve the *frequency asynchrony* problem induced by the multiple carrier frequency offsets (CFOs) of the signals of multiple users. This part of the thesis tackles the following two major challenges. The first, as in PNC systems, is the estimation of multiple channel parameters (e.g., CFOs and channel gains). A particular challenge is how to contain the estimation errors of the channel parameters of the multiple users, considering that the overall estimation errors may increase with the number of users because the estimations of their channel parameters are intertwined with each other. The second is how to compensate for the multiple CFOs. A particular difficulty is that, different from a single-user receiver for which there is only one CFO, it is not possible for our multiuser receiver to compensate for all the multiple CFOs simultaneously. To tackle the two challenges, we put forth a framework that solves the joint problem of multiuser channel-parameter estimation, CFO compensation, and channel decoding iteratively by employing the space alternating generalized expectation-maximization (SAGE) and expectation-conditional maximization (ECM) algorithms. Our study reveals that treating the data rather than the channel parameters as the hidden data in ECM will lead to better performance. We further show that Gaussian message passing is an effective complexity reducing technique. Simulations and real experiments based on software-defined radio (SDR) indicate that, compared with other approaches, our approach can achieve significant performance gains.

Overall, this thesis puts forth two frameworks (EM-BP for PNC, SAGE-ECM for MUD) to address the problem of multiple parameter estimation and channel decoding. We believe our frameworks are promising solutions for the signal processing challenges arising from the superimposition of multiple transmissions in multiuser systems.

摘要

本文研究在物理層網絡編碼（PNC）系統和多用戶檢測（MUD）系統中的聯合多參數估計與信道譯碼問題。PNC 與 MUD 都是從多個用戶的同時信號傳輸中獲利的技術。然而，多個同時傳輸信號的迭加也對信號處理帶來了若干挑戰。首先一個挑戰是在接收機處的多參數估計問題。另外一個挑戰是，如何同時補償多個參數。本文包括兩部分，每一部分的貢獻分別是在 PNC 或 MUD 系統中，針對上述問題的解決方案。

第一部分:

在本文的第一部分中，我們解決在 PNC 系統中的聯合信道估計與信道譯碼問題。在 PNC 系統中，多個用戶同時給中繼傳輸信號。PNC 系統的信道譯碼不同於傳統的多用戶系統的信道譯碼。具體地，中繼的目標是譯碼出網絡編碼後的信息而非單獨的每個源信息。雖然之前的研究工作顯示 PNC 可以很大程度上提高中繼網絡的吞吐量，但是這個提高的前提假設是能夠獲得精確的信道估計。然而，因為以下原因，PNC系統中的信道估計尤其具有挑戰性：1）多個用戶的信號迭加在一起；2）信道編碼使得數據符號之間非獨立；3）信道是時變的。為解決這些難題，我們將 expectation-maximization（EM）算法和 belief

propagation（BP）算法結合在一個統一的 factor graph 框架之下。在這個 factor graph 框架下，信道估計由 EM subgraph 完成，信道譯碼由建模了和 PNC 信道譯碼目標相匹配的虛擬編碼器的 BP subgraph 完成。在兩個 subgraph 的迭代消息傳輸使得我們可以逐漸逼近信道估計和信道譯碼的最優解。我們提供了大量的模擬結果來說明我們所提出方案的優越性。

第二部分

在本文的第二部分中，我們研究了一個信道編碼的多用戶檢測（MUD）系統。該系統是基於正交頻分複用（OFDM）調製和交織分多址接入（IDMA）技術的。將OFDM與IDMA結合的動機是其可以在頻率選擇多址接入信道環境下獲得多用戶分集增益的能力。然而，為了實現這個能力，我們必須首先解決由多個載波頻率偏移（CFO）所引起的頻率異步問題。論文本部分解決如下挑戰。首先的挑戰是多信道參數（CFO，信道增益等）的估計。考慮到各個用戶的參數估計問題互相影響而導致總的參數估計誤差會隨用戶數目而增長，一個具體地難題是如何克制多個用戶多個參數的估計誤差。第二個挑戰是如何補償多個 CFO。一個具體的難題是，不同於只存在一個 CFO 的單用戶接收機，我們的多用戶接收機不可能同時補償多個不同的 CFO。為解決以上兩個挑戰，我們提出了在一個多用戶系統中聯合、迭代解決多信道參數估計、CFO 補償和信道譯碼的框架。該框架利用了 space alternating generalized expectation-maximization（SAGE）算法和 expectation-conditional maximization （ECM）算法。我們的研究揭示，在 ECM 算法中，將數據符號而非信道參數設置為 hidden data 將導致更好的系統性能。進一步地，我們

用 Gaussian message passing 技術將算法複雜度有效降低。計算機仿真和軟件無線電平臺上的真實實驗表明，和傳統多用戶方法相比，我們方法能獲得非常高的性能增益。

總體來說，本文提出了兩個算法框架（EM-BP，SAGE-ECM）來解決聯合多參數估計和信道解碼問題。我們相信，針對多用戶系統中多個信號疊加而帶來的信號處理挑戰，我們所提算法框架是非常具有前景的解決方案。

Publications

- **T. Wang**, S. C. Liew, "Joint Channel Estimation and Channel Decoding in Physical-Layer Network Coding Systems: An EM-BP Factor Graph Framework", *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2229–2245, Apr. 2014.

- **T. Wang**, S. C. Liew, "Frequency-Asynchronous Multiuser Joint Channel-Parameter Estimation, CFO compensation and Channel Decoding", submitted to *IEEE Trans. Wireless Commun.*, Apr. 2015.

- **T. Wang**, T. Lv, H. Gao and Y. Lu, "BER Analysis of Decision-Feedback Multiple-Symbol Detection in Noncoherent MIMO Ultrawideband Systems", *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4684–4690, Nov. 2013.

- L. Lu, **T. Wang**, S. C. Liew and S. Zhang, "Implementation of Physical-Layer Network Coding", *Elsevier Physical Communication*, vol. 6, no. 1, pp. 74-87, Mar. 2013.

- **T. Wang**, S. C. Liew, "An EM approach for joint channel estimation and channel decoding in systems employing Physical-

Layer Network Coding", in *Proc. IEEE ICASSP'13*, pp. 5104–5108, Vancouver, BC, May 2013.

- **T. Wang**, S. C. Liew, "Joint phase tracking and channel decoding for OFDM PNC: algorithm and experimental evaluation", in *Proc. ACM SRIF '14*, pp. 69–76, Chicago, USA, Aug. 2014.

- L. Lu, **T. Wang**, S. C. Liew and S. Zhang, "Implementation of Physical-Layer Network Coding", in *Proc. IEEE ICC'12*, pp. 4734–4740, Ottawa, ON, Jun. 2012.

Acknowledgement

Now is the time I have finished my Ph.D study and thesis. When I recall the past four-year life at the Chinese University of Hong Kong (CUHK), I would have many feelings in my heart. I feel this is a journey full of various experiences. These different experiences redefine me and provide me with different possibilities in the further. However, I have to say that the pursuit of a Ph.D degree is not so easy. You need to undertake the pressures from the Chinese traditional society; you need to accept the loss of some possible opportunities. These are the costs you must pay when you want to gain something. In general, I felt blissful during these years, since I love to research. Therefore, the first one I want to thank is my Ph.D life itself.

Then, I would like to express my most sincere appreciation to my supervisor, Professor Soung-Chang Liew. It is my fortunate that Professor Liew supervised my Ph.D thesis. Without his guidance, I cannot finish my Ph.D thesis at this stage. I have learnt so many valuable things from him. His insights into theoretical problems, his research philosophy and his writing skill benefited my entire Ph.D study and, without doubt, will benefit my future research career.

There are many colleagues at CUHK I want to thank. All are my

good friends and good teammates. We did excellent collaborations in researching and playing football. I will not forget these working hours and happy hours we spent together. Thanks for the help from them during these years.

At last, I am grateful to my family, including my parents, my wife and her parents, for their loving considerations and great confidence in me all through these years. They are supporting me without any reservation. They are the most powerful motivations on my way forward. Since I own them, I just know what is the real meaning of the life.

To my lovely mier, who makes me laugh and weep.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

SIGNALS from multiple users in a wireless network are not bounded within a guided medium and isolated from each other. They permeate the wireless medium and they may cause mutual interference to each other. The traditional wireless technology paradigm treats such mutual interference as a destructive phenomenon [1]. Typically, different time or frequency channel resources are allocated to different users in an exclusive manner; so that one user transmits a radio signal on its allocated time or frequency channel only. This allows a receiver to isolate the signal from a particular user and avoids mutual interference among the signals of multiple users. If, however, multiple users transmit radio waves to their receivers on the same time or frequency channel, due to the broadcast nature of the wireless medium, a receiver receives signals from its transmitter as well as from other transmitters.

The radio waves from the other transmitters are often treated as

interferences that corrupt the intended signal. For example, in 802.11 wireless local area networks (WLAN), when multiple nodes transmit together, packet collisions occur and the receiver simply discards the signals and does not attempt to decode the overlapping packets [2,3]. This conservative design principle tries to prevent interference as far as possible. As a consequence, as long as interfering transmissions occur in wireless networks, the spectral efficiency is wasted. Unfortunately, interfering transmissions occur quite frequently today due to the densely deployed wireless terminals.

To increase spectral efficiency (thus also the data throughput), recent emerging wireless technologies advocate that multiple transmitters can transmit signals together on the same time-frequency channel. From the overall viewpoint of the system, overlapping signals are not considered as interfering signals; and all transmitted signals are useful. The issue is how to exploit advanced signal processing and communication techniques to harness the useful information contained in the overlapping signals. This thesis focuses on two such technologies: physical-layer network coding (PNC) and multiuser detection (MUD).

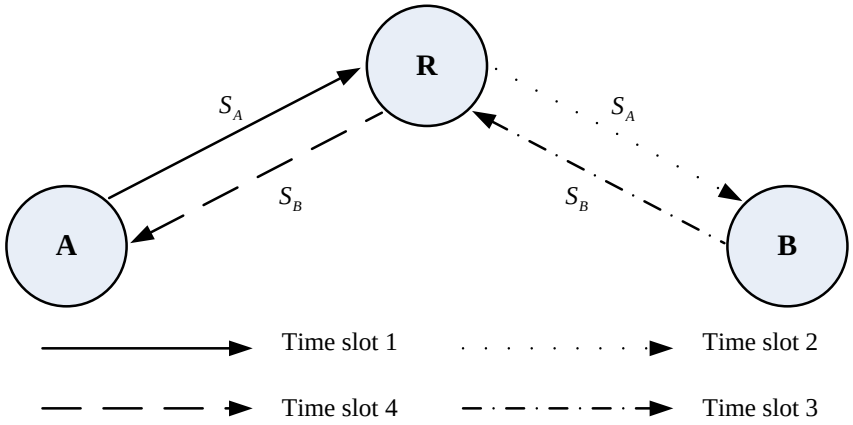
1.1.1 Physical-layer Network Coding (PNC)

The concept of PNC was first proposed at *Mobicom* 2006 [4].[1] As a subfiled of network coding [6, 7], PNC means performing network coding at the physical-layer of wireless networks. This is achieved based on a simple fact in physics that when multiple radio waves (transmitted by multiple transmitters) come together within the same physical space, they add. The mixing of radio waves is a form of network coding. Therefore, by allowing multiple transmitters transmit together and exploiting the network coding operation performed by the wireless nature, PNC puts the "interference" into good use.

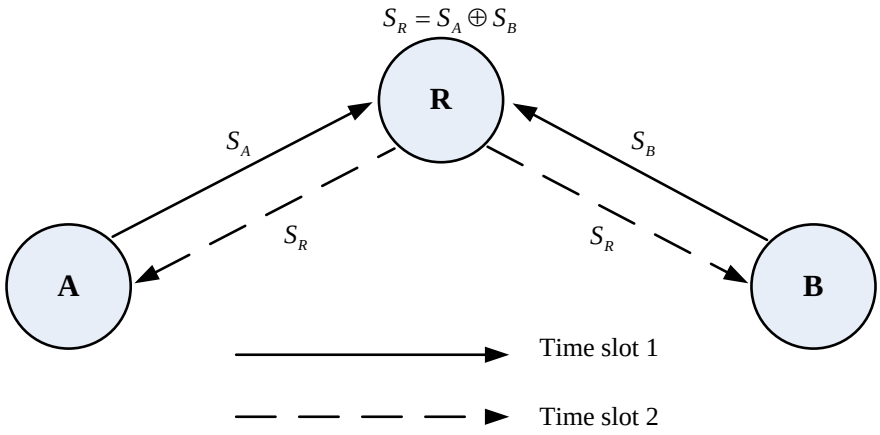
The core idea of PNC could be most easily demonstrated using a simple example network called the two way relay channel (TWRC). We illustrate the TWRC in Fig. 1.1. TWRC is a three-node linear network where two end nodes, nodes A and B, want to communicate via a relay node R. There is no direct signal path between nodes A and B. An example is a satellite network in which nodes A and B are the ground stations, and the relay R is the satellite. We ask the question how many time slots are required for nodes A and B to exchange one packet with each other.

The most straightforward solution for the question above can be

[1] The same idea was also independently proposed in [5] almost at the same time.



(a) Traditional scheduling scheme



(b) Physical-layer network coding scheme

Figure 1.1: Two way relay channel operated with (a) traditional scheduling scheme, and (b) physical-layer network coding scheme.

derived using the traditional design principle that tries to prevent interference in the network. With the traditional design principle, there is no concurrent transmission in TWRC; in each time slot, there is one node that is valid for transmitting. Then, a total of four time slots are required to exchange two packets, one in each direction. In time slot 1, node A transmits a packet S_A to relay R; in time slot 2, relay R forwards S_A to node B; in time slot 3, node B transmits a

packet S_B to relay R; and in time slot 4, relay R forwards S_B to node A. This is also illustrated in Fig. 1.1a, and we refer to this scheme as the traditional scheduling scheme.

Compared with TS scheme, PNC can improve the throughput of TWRC by 100%, i.e., the required time slots for exchanging two packets are reduced from four to two. With PNC, in time slot 1, the two end nodes simultaneously transmit signals to the relay; the relay maps the superimposed signals of the simultaneous packets to a network-coded packet $S_R = S_A \oplus S_B$, where $\oplus$ denotes the XOR operation, a form of network coding.[2] In time slot 2, the relay broadcasts the network-coded packet S_R back to the end nodes; one end node can extract the packet of the other end node from the received S_R by using itself information. For example, node A can obtain S_B by doing $S_B = S_R \oplus S_A$. Fig. 1.1b illustrates the idea of PNC. By making use of concurrent transmissions, PNC "embraces" interference, thus increases the network throughput.

1.1.2 Multiuser Detection (MUD)

The other scheme that benefits from multiple concurrent transmissions is MUD [9, 10]. For MUD, we usually consider a multiple-access

[2] A key issue in PNC is how relay R deduces $S_R = S_A \oplus S_B$ from the superimposed signals. We refer to this process as "PNC mapping". The details about PNC mapping can be found in [8]. PNC mapping is not the main subject investigated of this thesis.

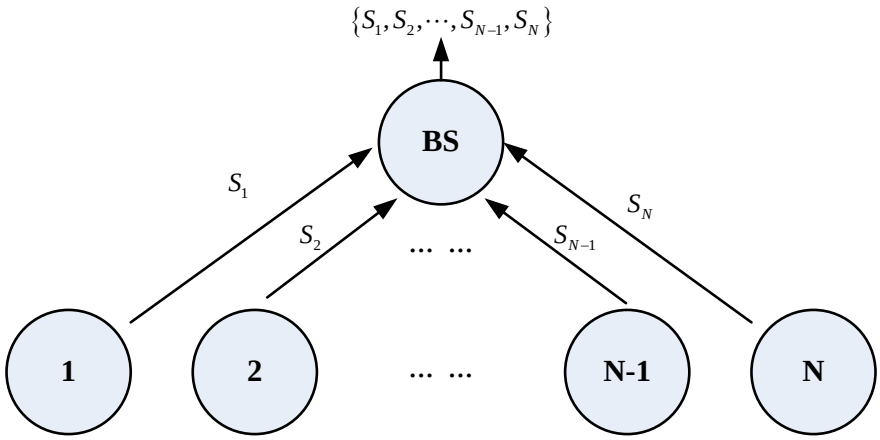


Figure 1.2: Multiple Access Channel with multiuser detection (MUD).

channel, where N terminal users want to communicate with a common base station (BS).

To achieve the goal of multiple-access communications, the traditional design principle divides the wireless channel into N orthogonal subchannels, along with the time or frequency axis. One time or frequency subchannel is allocated to one user for communicating with BS. The corresponding channel allocation scheme is time division multiple-access (TDMA) or frequency division multiple-access (FDMA) [11]. Since the subchannels are orthogonal to each other, the receiver of BS can extract the message of one particular user employing a single-user decoding scheme on its corresponding subchannel.

Different from the traditional scheme above, we now allow the N users transmit simultaneously on the whole wireless channel. Then, the receiver of BS employs MUD to extract the messages from the N user jointly. The concept of MUD is illustrated in Fig. 1.2,

where the packet contained the message of user n is denoted by S_n, $n = 1, 2, \cdots, N$. We remark that although the signal transmission process of MUD systems is similar to that of PNC systems, the decoding objectives of PNC and MUD are different. The decoding objective of PNC is to obtain the network-coded message at the relay; that of MUD is to recover the N source messages at BS.

To assist MUD, some user separation schemes are often adopted at the transmitter or receiver side. For example, the receiver of BS and/or the transmitters of users can deploy multiple antennas to separate the N users by creating parallel spatial pipes, and this separation scheme is called spatial division multiple-access (SDMA) [12]; the transmitters of users can exploit different spreading codes or interleaving patterns to separate themselves, the separation scheme is called code division multiple-access (CDMA) [13] or interleaving division multiple-access (IDMA) [14]. Since the transmitted signals of SDMA, CDMA and IDMA are overlapping on the same time-frequency channel and usually non-orthogonal to each other, these new user separation schemes together with MUD can improve the spectral efficiency of multiple-access channels.

1.2 Motivation and Problem

While benefiting PNC and MUD, the superimposition of multiple signals also brings out many challenges for signal processing. One challenge is the estimation of the multiple parameters at the receiver. Moreover, how to compensate for these multiple parameters is also

challenging.

Many prior studies on PNC and MUD simplify assumptions that were made to ease analysis. For example, it was assumed that channel gains are known perfectly and that carrier-frequency offsets (CFO) are negligible. Our experimental prototyping efforts of PNC systems [15] indicate that these parameters can have significant effects on system performance, and that they are not negligible and are often difficult to estimate. Furthermore, the negative effects of some parameters (e.g., CFO) cannot be perfectly compensated in PNC and MUD systems using the traditional approach in which parameter estimation, compensation, and channel decoding are performed successively one after another. To fill this gap, this thesis proposes to investigate "iterative algorithms" that perform parameter estimation and channel decoding in an integrated and iterative manner.

To illustrate our point, in the following we focus on the challenge of CFO estimation and compensation for superimposed signals. Modern multicarrier system such as orthogonal frequency-division multiplexing (OFDM) is very sensitive to CFO: it causes inter-carrier interference that destroys the orthogonality among subcarriers.

Consider a network in which two transmitters simultaneously transmit signals to a common receiver. This transmission process occurs in both PNC and MUD systems. The receiver receives the following baseband signal (in time domain)

$$R(i) = S_1(i)\, e^{j(\varepsilon_1 i + \phi_1)} + S_2(i)\, e^{j(\varepsilon_2 i + \phi_2)} + Z(i) \tag{1.1}$$

where $S_n(i)$ is the i^{th} symbol from transmitter $n \in \{1, 2\}$; ε_n is the

CFO of user n; ϕ_n is the initial phase offset of user n; and $Z(i)$ is the noise.

In a single-user point-to-point system, there is only one transmitter and one receiver. Thus, in (1.1) suppose that transmitter 1 is the only transmitter. Then the expression becomes

$$R_1(i) = S_1(i)\, e^{j(\varepsilon_1 i + \phi_1)} + Z(i) \tag{1.2}$$

If the receiver could estimate ε_1 and ϕ_1 perfectly, then it could perform compensation by multiplying the above equation by $\exp[-j(\varepsilon_1 i + \phi_1)]$, yielding

$$\widetilde{R}_1(i) = R_1(i)\, e^{-j(\varepsilon_1 i + \phi_1)} = S_1(i) + \widetilde{Z}(i) \tag{1.3}$$

where $\widetilde{Z}(i) = Z(i) \exp[-j(\varepsilon_1 i + \phi_1)]$. If $Z(i)$ is circularly-symmetric Gaussian, so is $\widetilde{Z}(i)$. Thus, we see that the crux of the problem in the point-to-point case is a good estimation of ε_1 and ϕ_1, so that they can be completely removed by compensation.

Unfortunately, such is not the case with PNC and MUD. Returning to (1.1), even if we could estimate ε_1, ϕ_1, ε_2 and ϕ_2 perfectly, we could not remove all of them simultaneously. If we removed the CFO and phase offset of transmitter 1, those of transmitter 2 would remain, and vice versa.

Instead of performing CFO estimation and compensation before channel decoding, we could use iterative algorithms in which CFO estimation and channel decoding are performed jointly and iteratively to take into account the imperfect compensation optimally.

In general, besides CFO, other system parameters (e.g. channel gains) also need to be estimated and dealt with. The design of

such iterative algorithms for PNC, as far as we know, has not been investigated by prior research, and that they offer a promising path to solve many practical problems that arise in PNC.

For MUD, although there are some research works that investigated the problem of joint channel gain estimation and channel decoding, the research that considers the problem of joint parameter (channel gains, CFOs) estimation, CFO compensation and channel decoding is still lacking. Moreover, the decoding objectives of PNC and MUD are different. Thus, the techniques used for PNC and MUD channel decoding are also different. Dfferent channel decoding objectives may have different impacts on the overall framework of the joint problem. Therefore, we need to consider the problem for PNC and MUD, respectively and carefully.

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows.

Chapter 2 propose a combined use of the expectation-maximization (EM) algorithm and belief propagation (BP) algorithms for tackling the problem of joint channel estimation and channel decoding in PNC systems.

Chapter 3 investigates a channel-coded MUD system operated with OFDM and IDMA, where we focus on the problems induced by multiple CFOs. We propose to solve the multiuser joint problem of parameter estimation, CFO compensation and channel decoding using an iterative algorithm that employs space alternating

generalized expectation-maximization (SAGE) and expectation-conditional maximization (ECM) algorithms.

Chapter 4 concludes the thesis with discussions on future topics.

□ **End of chapter.**

Chapter 2

Joint Channel Estimation and Channel Decoding in Physical-Layer Network Coding Systems: An EM-BP Factor Graph Framework

2.1 Introduction

RECENTLY, the research community has shown growing interest in a simple relay network in which two terminal nodes communicate via a relay. This network is referred to as the two-way relay channel (TWRC). Much of the interest in TWRC is on the exploitation of physical-layer network coding (PNC) [4,8] to boost its throughput.

Ref. [4] showed that PNC could increase TWRC throughput by 100% compared with traditional relaying [8]. In TWRC operated with PNC, the two terminal nodes first transmit their messages simultaneously to the relay. The relay then maps the overlapped

signals to a network-coded message (e.g., bit-wise XOR of the messages of the terminal nodes) and broadcasts the network-coded message to the two terminal nodes. Each terminal node then extracts the message of the other terminal node by subtracting its own message from the network-coded message. Thus, the two terminal nodes exchange one message with each other in two time slots. With traditional relaying, four time slots are needed [4].

This throughput advantage of PNC, however, is predicated on the accurate estimate of the channels between the terminal nodes and the relay. For optimality, it is desirable to obtain the maximum *a posteriori* probability (MAP) channel estimates. This is, however, a particularly challenging task for PNC. The involved issues addressed by this chapter are as follows:

- For reliable communication, we consider channel-coded PNC [8, 16]. Specifically, the source messages of the two terminal nodes are channel-coded into channel-coded messages before transmission. The signals received by the relay contain the overlapped channel-coded messages as well as overlapped preambles (training symbols) and pilots.

- PNC channel decoding is different from the multiuser joint channel decoding. The goal of the PNC channel decoding at the relay is to obtain a network-coded message rather than the two individual source messages [8, 16]. In other words the relay aims not to decode the two source messages, but to decode a network-coding function of the two source messages (in this chapter, we

assume the network-coding function is the bit-wise XOR of the two source messages).

- For successful decoding, accurate channel estimates are needed. For optimality, it is desirable to estimate the channels using not just the preambles and pilots, but also the data in the signals. This is because the data portion also contains useful information related to the channels.

- We are interested in time-varying channels in which the channel gains vary from symbol to symbol within a packet.

Overall, performing channel estimation and PNC channel decoding when (i) the signals are overlapping; (ii) the data symbols are correlated due to channel coding; and (iii) the channels are time-varying, is a particularly challenging task.

To tackle this challenge, this chapter proposes and investigates a joint channel estimation and channel decoding framework. We argue that directly trying to solve the MAP channel estimation problem and the channel decoding problem in a separate manner is not viable; a solution is found in a combined use of expectation-maximization (EM) algorithm and belief propagation (BP) algorithm that solves the two problems jointly in an iterative manner.

We implement the EM-BP computation as a message passing algorithm on a factor graph [17, 18], in which the component for channel estimation (implemented by EM) and the component for PNC channel decoding (implemented by BP) are interconnected. Through

iterative message passing between the EM and BP components and iterative message passing between elements within the BP channel decoding component, the results of EM channel estimation and BP channel decoding improve progressively toward optimality.

Overall, there are three major contributions to this chapter:

1. This is the first work that applies the EM algorithm for joint channel estimation and channel decoding in PNC systems. Notably, our algorithmic framework includes a schema to deal with time-varying channels. To reduce computation complexity, we further extend our framework by replacing EM with its variant named space alternating generalized expectation-maximization (SAGE).

2. We outline a factor graph framework for iterative message passing algorithm based on the foundations of EM/SAGE and BP. This is the first time EM/SAGE-BP computation for a channel-coded communication system is fully implemented as a message passing algorithm on a factor graph (prior work either did not consider channel coding or did not use the factor graph schema). In particular, we explicitly establish this EM/SAGE-BP factor graph framework from a rigorous theoretical foundation. We remark that although our focus here is on PNC systems, this framework is also applicable to the conventional single-user system and multi-user system.

3. Through extensive computer simulations, we investigate the performance of our EM/SAGE-BP PNC receivers and compare

them with other existing receivers in the literatures. The simulation results demonstrate the superiority of our receivers over other receivers, confirming the theoretical optimality of EM/SAGE-BP PNC.

2.1.1 Related Works

Theory of BP and Its Application in PNC:

BP, also known as the sum-product message passing algorithm, is a general algorithmic inference method for graph models [17, 19]. It has found great success in the decoding of powerful channel codes (e.g., Turbo codes and LDPC codes [20]) in point-to-point communications. BP was first applied to PNC channel decoding in [16], which puts forth the concept of "virtual encoding". The concept was further generalized in [21, 22]. Refs. [16, 21, 22] were followed by many other papers on channel-coded PNC [8]. The prior studies of PNC channel decoding mostly assume that the channels are perfectly known. In practice, these channels have to be estimated. PNC systems are multi-user systems in which there are multiple channel parameters, whose estimation is particularly challenging.

It is desired that we could jointly solve channel estimation and channel decoding by BP. Recently, [23] proposed a BP method for joint channel estimation and channel decoding in PNC systems. The direct application of BP to channel estimation, however, requires summations (integrations) over continuous channel variables, which are computationally intensive. To reduce computation complexity,

a moment matching (MM) technique is used in [23]. Specifically, the original Gaussian-mixture messages in the BP algorithm are approximated by Gaussian messages that have the same first and second moments. However, as a consequence of the approximation by MM, the optimality cannot be guaranteed. By contrast, in this chapter, instead of using a pure BP method for both channel estimation and channel decoding, we use an EM-BP method that obviates the need for computation-intensive integration without sacrificing optimality.

Theory of EM and Its Application in Single-user Systems:

Ref. [24] first proposed EM as a general iterative algorithm for finding the maximum likelihood (ML) estimates of parameters in statistical model with hidden variables that cannot be observed directly. A small extension allows the finding of the MAP estimates also [25]. A variant of EM named SAGE was proposed in [26] to increase the convergence rate. Recently, [27,28] proposed a framework to implement EM computation as a message passing algorithm on a factor graph. Application for communications systems (and in particular, channel-coded communications systems) was not its main target. It is not clear that the factor graph representation in [27, 28] is applicable to the communication problem of interest to us here. With respect to our contribution 2) listed above, in this chapter we fill in the missing details and show specifically how a factor graph representation can be used in our problem. In addition, we extend the framework of EM message passing over factor graphs to SAGE message passing.

Papers [24–28] concern EM/SAGE in general. There have also been many investigations on the application of EM/SAGE in communication systems specifically. Refs. [29–35] for example, applied EM to the problem of joint channel estimation and detection/decoding in single-user systems. Refs. [36, 37] applied SAGE to the same problem. The work [35] directly applied the results in [27, 28] to implement EM message passing for joint channel estimation and detection in a single-user system. However, the incorporation of channel codes by [35] is heuristic and is not in accordance with the rigorous theoretical derivation in this chapter.

EM Application in Multi-user Systems:

Refs. [38–41] applied EM to joint channel estimation and multi-user detection in CDMA systems. Channel coding was not considered. Ref. [42] incorporated channel coding. However, the proposed scheme performs serial interference cancelation (SIC) and tries to decode the individual messages of different users using separate channel decoders. Decoding individual messages is an overkill for PNC systems and may lead to suboptimal performance (we will provide results showing this in Section 2.5). The application of SAGE in multi-user systems for joint channel estimation and detection can be found in [39, 43, 44], where [39, 43] assume CDMA systems and [44] assumes OFDMA systems.

Overall, there has been little multi-user EM work that incorporates channel coding. Furthermore, with respect to our contribution 1) listed above, it is worth re-emphasizing that PNC channel decoding [16, 21, 22] is different from conventional multiuser channel decoding

[42], because PNC aims to channel-decode the overlapped received signals into a network-coded message [16, 21, 22] rather than the individual messages of different users.

The rest of this chapter is organized as follows. Section 2.2 describes our system model. In Section 2.3, we apply EM-BP to PNC and derive the update equations for EM-BP message passing. Section 2.4 applies SAGE-BP to PNC. Section 2.5 presents the simulation results.

Notations: We denote matrices by bold capital letters, vectors by bold small letters, and scalars by regular letters throughout this chapter. All vectors are column vectors. $\mathbf{I}_r$ denotes the $r \times r$ identity matrix and $\mathbf{0}_r$ denotes the $r \times r$ matrix with all zero elements. $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}^{\mathrm{H}}$, $\mathbf{A}^{-1}$ and $\det(\mathbf{A})$ denote the transpose, the conjugate transpose, the inverse and the determinant of $\mathbf{A}$, respectively. $\mathcal{CN}(\mathbf{x} : \mathbf{m}, \mathbf{K}) \triangleq \frac{1}{\pi^r \det(\mathbf{K})} \exp\left[-(\mathbf{x}-\mathbf{m})^{\mathrm{H}} \mathbf{K}^{-1}(\mathbf{x}-\mathbf{m})\right]$ denotes the probability density function (PDF) of an r-dimension complex Gaussian random variable $\mathbf{x}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{K}$. $\|\mathbf{x}\|$ is the Euclidean norm of a vector $\mathbf{x}$. $(\cdot)^*$ denotes the conjugate of a complex number. $|C|$ is the cardinality of a set C. $x_{i:j}$ is a set containing the elements in a sequence x indexed from i to j, inclusively. Finally, $\otimes$ denotes the Kronecker product, and $\oplus$ denotes the exclusive-or operation.

2.2 System Model

We consider a PNC transmission scheme for TWRC consisting of two time slots. In time slot 1, two terminal nodes A and B transmit packets to a relay node R simultaneously. From the overlapped signals received from A and B, R constructs a network-coded packet and broadcasts it to A and B in time slot 2. From the network-coded packet, A(B) then recovers the packet of B(A) using its self information [8].

This chapter focuses on time slot 1 of PNC; the problem of reliably transmitting the network-coded packet in time slot 2 is similar to that in a conventional point-to-point link. We assume A and B have one transmit antenna, and R has one receive antenna. In time slot 1, the received signal at R in the i^{th} symbol duration can be expressed as

$$y_i = h_{A,i}x_{A,i} + h_{B,i}x_{B,i} + n_{R,i} = \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i + n_{R,i} \tag{2.1}$$

where $x_{A,i}$ ($x_{B,i}$) is the i^{th} transmitted symbol of node A(B); h_i^{A} $\left(h_i^{\mathrm{B}}\right)$ is the i^{th} fading coefficient of the channel between A(B) and R; $n_{R,i}$ is the complex white Gaussian noise sample with zero mean and variance N_0; $\mathbf{h}_i \triangleq [h_i^{\mathrm{A}}, h_i^{\mathrm{B}}]^{\mathrm{T}}$; and $\mathbf{x}_i \triangleq [x_{A,i}, x_{B,i}]^{\mathrm{T}}$. A block diagram of the system model is shown in Fig. 2.1, where $\{s_{A,j}\}$ and $\{s_{B,j}\}$ are the source information bits from nodes A and B. The transmitted symbols $\{x_{A,i}\}$ and $\{x_{B,i}\}$ are generated after channel encoding, interleaving, constellation mapping and pilot insertions at the transmitters. In this

chapter, we assume that A and B use the same channel encoder[1] (the valid set of codewords is denoted by C) and the same interleaver when mapping their source bits $\{s_{A,j}\}$ and $\{s_{B,j}\}$ to transmitted symbols. Pilot symbols are inserted periodically among coded data symbols. The assumed frame structure is shown in Fig. 2.2 where P and D represent the pilot symbols and coded data symbols, respectively. Each frame consists of l data symbols, divided into l/Δ blocks. Each block has Δ data symbols and two pilot symbols. We refer to Δ as the pilot interval.[2] The total frame length is $L = l + 2\,(l/\Delta)$ symbols.

We assume time-varying Rayleigh fading channels. The time-varying channel gains $\{h_{A,i}\}$ and $\{h_{B,i}\}$ are modeled as two independent first-order Gauss-Markov processes [45]:

$$\begin{aligned} h_{A,i} &= \alpha h_{A,i-1} + \sqrt{1-\alpha^2} z_{A,i} \\ h_{B,i} &= \alpha h_{B,i-1} + \sqrt{1-\alpha^2} z_{B,i} \end{aligned} \tag{2.2}$$

where $\{z_{A,i}\}$ and $\{z_{B,i}\}$ are independent white complex Gaussian processes with zero mean and variances σ_{A}^2 and σ_{B}^2 for all i. The parameter α is a correlation coefficient used for modeling how fast the channel varies with time (i.e., it is related to the channel coherence time) [45]. [3] The distributions of $h_{A,i}$ and $h_{B,i}$ are zero mean complex

[1] Discussion on how to deal with two nodes with different channel encoders can be found in Section 2.4 of this chapter.

[2] Simulation results on the impact of the pilot interval on system performance can be found in Section 2.5.

[3] Different α for the channels of the two nodes will not change the effectiveness of our

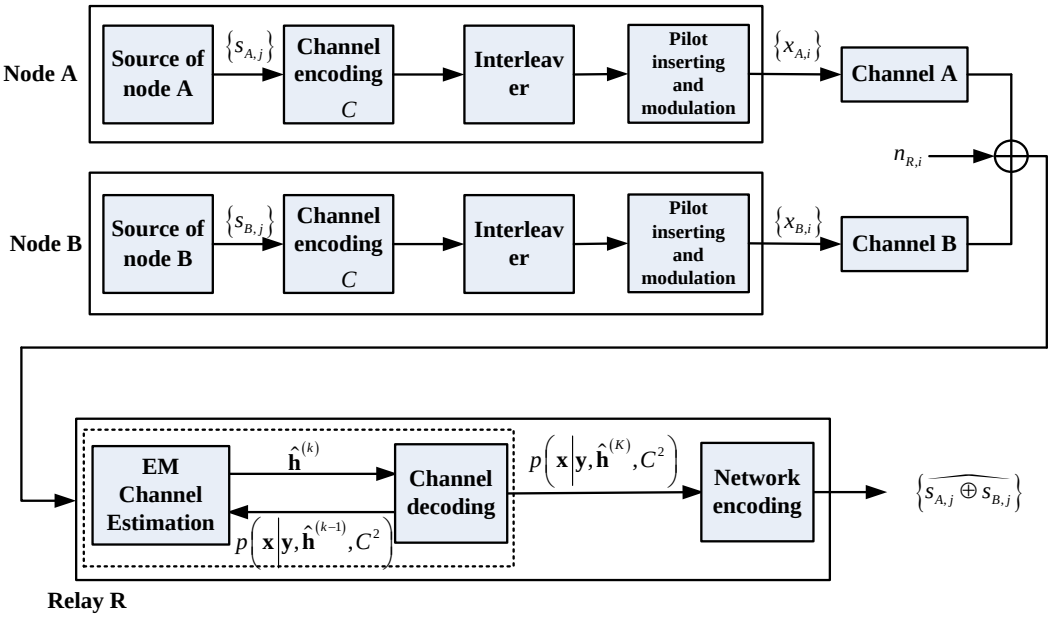


Figure 2.1: The system model of time slot 1 of PNC in TWRC.

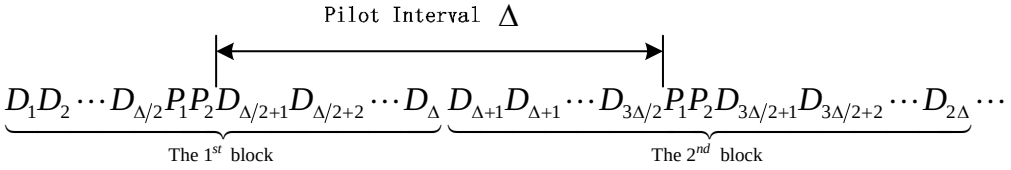


Figure 2.2: The frame structure employed by the two user nodes.

Gaussian with variances σ_A^2 and σ_B^2, respectively. Therefore, the amplitude of every element of them is Rayleigh distributed, and the phase is uniformly distributed.

algorithm. Here, we assume the same α of the two nodes for simplifying the descriptions.

2.3 Application of EM-BP to PNC

2.3.1 Objectives of EM PNC Receiver

Let $\mathbf{h} \triangleq \{\mathbf{h}_i\}$ be the set containing channel gains of all time. Similarly, let $\mathbf{x}$ be the set of all transmitted symbols $\{\mathbf{x}_i\}$, and $\mathbf{y}$ is the set of all received signals $\{y_i\}$. To the relay, both $\mathbf{h}$ and $\mathbf{x}$ are unknowns to be estimated and decoded.

In a conventional receiver, $\mathbf{h}$ is first estimated, followed by the decoding of codewords $\mathbf{x}$. Pilots, corresponding to known $\mathbf{x}_i$ at specific positions i, are used for the estimation of $\mathbf{h}$. After that, $\mathbf{h}_i$ for the positions occupied by data are estimated by interpolation. This estimate of $\mathbf{h}$ is then substituted into (2.1) for the decoding of the data $\mathbf{x}_i$. This estimate of $\mathbf{h}$ makes use of only the pilot parts, and does not exploit useful information contained in the data part of $\mathbf{x}$.

In our PNC receiver design, we wish to make use of the pilots as well as the data parts of $\mathbf{x}$ in the channel estimation process. In particular, observations $\mathbf{y}$ at all positions i and the knowledge of the used channel code will be employed to estimate $\mathbf{h}$. A possible strategy for our channel estimation and channel decoding problem in PNC receiver is as follows:

Step 1 (channel estimation): Find MAP estimate

$$\widehat{\mathbf{h}}_{\text{MAP}} = \arg\max_{\mathbf{h}} \left\{ \log p\left(\mathbf{h} \left| \mathbf{y}, C^2 \right.\right) \right\} = \arg\max_{\mathbf{h}} \left\{ \log \sum_{\mathbf{x}} p\left(\mathbf{x}, \mathbf{h} \left| \mathbf{y}, C^2 \right.\right) \right\}$$

Step 2 (channel decoding): Given $\widehat{\mathbf{h}}_{\text{MAP}}$, find $p\left(\mathbf{x} \left| \widehat{\mathbf{h}}_{\text{MAP}}, \mathbf{y}, C^2 \right.\right)$

Step 3 (network coding): Compute the network-coded source

message $\left\{\widehat{s_{A,j} \oplus s_{B,j}}\right\}$ based on the channel decoding output from Step 2 [8].

This is the PNC receiver with optimal channel estimation. Some remarks on the receiver are as follows:

1. A subtlety here in Step 2 for the PNC system is that C^2 is the code constraint by the "virtual channel encoder" which takes the original information source symbols from nodes A and B $\{s_{A,j}, s_{B,j}\}$ as inputs, and outputs $\{\mathbf{x}_i\}$ as coded symbols (see [8] for details).

2. If the channel coefficients were perfectly known (as assumed in previous works [16, 21, 22]), then Step 1 would not be needed. Step 2 and Step 3 then form the so-called Channel-decoding-Network-Coding (CNC) process, an essence of channel-coded PNC systems [8, 16]. Compared to conventional channel decoding, the goal of CNC is not to decode the individual source messages of A and B, but to decode a network-coded message that mixes the two source messages (we refer the interested readers to [8] and references therein for details on CNC). If the MAP estimation in Step 1 could be achieved, then Step 2 and Step 3 could be implemented using the conventional CNC methods, substituting $\widehat{\mathbf{h}}_{\text{MAP}}$ as the channel coefficients [8, 16]. Unfortunately, this is not viable because the exact MAP estimate of $\mathbf{h}$ is difficult to obtain due to the complexity of the computation of $\sum_{\mathbf{x}} p(\mathbf{h}, \mathbf{x} \,|\mathbf{y})$. A difficulty, for example, is that the symbols in $\mathbf{x}$ are correlated due to channel coding; in

addition, signals of the two terminal nodes are overlapped in $\mathbf{y}$.

To facilitate the design of PNC receiver, this chapter makes use of the EM algorithm for channel estimation. Specifically, EM tries to find $\widehat{\mathbf{h}}_{\text{MAP}}$ iteratively rather than attacking the problem directly. The objective of EM is still to obtain the MAP estimate of $\mathbf{h}$ as in Step 1. However, EM combines Step 1 and Step 2 in an iterative manner to refine the estimate of $\mathbf{h}$ and the decoding of the network-coded message. In the following, we first describe the procedure of the EM computation and then present its implementation as a message passing algorithm on a factor graph.

In the terminology of EM, $\mathbf{y}$ is the observed data, $\mathbf{x}$ is the hidden data, the pair $(\mathbf{x}, \mathbf{y})$ is the complete data, and $\mathbf{h}$ is the unknown parameter. The k^{th} iteration of EM consists of an E-step (expectation) and an M-step (maximization) as follows [25]:

E-step: Given the previous estimate $\widehat{\mathbf{h}}^{(k-1)}$, compute the conditional expectation

$$Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) = \sum_{\mathbf{x}} p\left(\mathbf{x}\left|\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right) \log p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right) \quad (2.3)$$

M-step: Then, compute $\widehat{\mathbf{h}}^{(k)}$ by

$$\widehat{\mathbf{h}}^{(k)} = \arg\max_{\mathbf{h}} \left[Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) + \log p(\mathbf{h})\right] \quad (2.4)$$

The E-step in (2.3) can be broken down as follows. First, compute $p\left(\mathbf{x}\left|\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)$ from $\mathbf{y}$ and $\widehat{\mathbf{h}}^{(k-1)}$. This computation of $p\left(\mathbf{x}\left|\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)$ is similar to Step 2 above, with $\widehat{\mathbf{h}}^{(k-1)}$ replacing $\widehat{\mathbf{h}}_{\text{MAP}}$. If the algorithm were to stop at iteration $k-1$, we could

simply go to Step 3 to obtain the network-coded message based on $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$. Otherwise, the E-step continues and uses $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ to compute the $Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)$ in (2.3). After that, the M-step finds a new estimate of $\mathbf{h}$ as in (2.4). In this way, the computation procedure jointly estimates the channels $\mathbf{h}$ and decodes the data $\mathbf{x}$, refining the solutions from one iteration to the next. This process is shown in the lower half of Fig. 2.1, which depicts our PNC receiver.

Let us now re-examine (2.3) and (2.4) to explain why they work. Combining the E-step and M-step by substituting (2.3) into (2.4) and after some manipulations, we find that the EM algorithm is actually a fixed-point iteration algorithm as follows:

$$\begin{aligned}\widehat{\mathbf{h}}^{(k)} &= \arg\max_{\mathbf{h}}\left[\sum_{\mathbf{x}} p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\log p\left(\mathbf{y},\mathbf{x}|\mathbf{h},C^2\right)\right. \\ &\qquad\qquad \left.+\log p(\mathbf{h})\right] \\ &= \arg\max_{\mathbf{h}}\left[-D_{KL}\left(p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)||p\left(\mathbf{x}\left|\mathbf{y},\mathbf{h},C^2\right.\right)\right)\right. \\ &\qquad\qquad \left.+\log p\left(\mathbf{h}\left|\mathbf{y},C^2\right.\right)\right]\end{aligned} \tag{2.5}$$

where D_{KL} is the Kullback-Leibler (K-L) divergence. It is known that miminizing K-L divergence D_{KL} corresponds to minimizing the difference between two distributions [46]. The last line of (2.5), however, is not exactly minimizing D_{KL}; besides D_{KL} , there is an additional term $\log p\left(\mathbf{h}\left|\mathbf{y},C^2\right.\right)$ in the function be optimized. Another way to look at (2.5) is as follows. Recall that finding $\arg\max_{\mathbf{h}}\log p\left(\mathbf{h}\left|\mathbf{y}\right.,C^2\right)$ is the original objective of Step 1. In (2.5), this objective is modified by the additional term D_{KL}. Appendix A

shows that the fixed-point EM iteration in the above equation is still an attempt to find $\arg\max_{\mathbf{h}} \log p\left(\mathbf{h}\,|\mathbf{y}\,, C^2\right)$ despite the additional D_{KL} term. Indeed, finding $\arg\max_{\mathbf{h}} \log p\left(\mathbf{h}\,|\mathbf{y}\,, C^2\right)$ is facilitated by adding D_{KL}, which goes to zero when the algorithm converges. The solution of EM is at least a local optimal $\log p(\widehat{\mathbf{h}}_{\mathrm{MAP}}\,|\mathbf{y}, C^2)$ although it may not be the global optimum. A good initial point will often lead to a global optimum [47].

2.3.2 Implementation of EM-BP PNC on Factor Graph

Although EM can iteratively find the MAP estimate, the computations of the E-step (2.3) and M-step (2.4) are still non-trivial. We next establish a framework that implement the EM computation as a message passing algorithm on a factor graph, where we can rigorously combine EM message passing for channel estimation with BP message passing for channel decoding. This framework allows us to more clearly see the intricacies of the EM computation, pointing to a systematic and practical way to implement it.

Refs [27, 28] showed how to transform a generic EM computation to a factor graph implementation. It is not clear, however, that the assumptions in [27, 28] on the functional forms of the parameter and variable probabilities are valid for our specific problem. Here, we give a theoretical derivation tailored to channel-coded communication systems.

A key to factor graph implementation is to factorize $p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right)$

in (2.3) and $p(\mathbf{h})$ in (2.4). For $p\left(\mathbf{y},\mathbf{x}|\mathbf{h},C^2\right)$, we write

$$p\left(\mathbf{y},\mathbf{x}|\mathbf{h},C^2\right)=p\left(\mathbf{y}\left|\mathbf{x},\mathbf{h}\right.\right)p\left(\mathbf{x}\left|C^2\right.\right)=\frac{I_{C^2}(\mathbf{x})\prod_i p(y_i|\mathbf{x}_i,\mathbf{h}_i)}{|C^2|} \tag{2.6}$$

where $I_{C^2}(\mathbf{x})$ is a indicator function defined as: $I_{C^2}(\mathbf{x})=1$ if $\mathbf{x}\in C^2$; $I_{C^2}(\mathbf{x})=0$ if $\mathbf{x}\notin C^2$, where we have assumed all codewords are equally likely. Note that we have used (2.1) to justify the factorization of $p\left(\mathbf{y}\left|\mathbf{x},\mathbf{h}\right.\right)$ into $\prod_i p\left(y_i\left|\mathbf{x}_i,\mathbf{h}_i\right.\right)$ in (2.6), where y_i is independent of y_j for all $i\neq j$ given $\mathbf{x}$ and $\mathbf{h}$ thanks to their independent noise components. Substituting (2.6) into the Q function in (2.3) and dropping the term $-\log\left|C^2\right|$, which is independent of $\mathbf{h}$ and therefore does not matter as far as the M-step is concerned, we have

$$Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)=\sum_i\sum_{\mathbf{x}_i}\log p\left(y_i\left|\mathbf{x}_i,\mathbf{h}_i\right.\right)\underbrace{\sum_{\mathbf{x}_{1:i-1},\mathbf{x}_{i+1:L}}p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)}_{\triangleq p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)} \tag{2.7}$$

where $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ is the *a posteriori probability* (APP)[4] that can be computed using BP (sum-product) message passing algorithm for channel decoding on the factor graph [17] of the given channel code C^2, with a fixed channel $\widehat{\mathbf{h}}^{(k-1)}$. We define the symbol-wise Q

[4]If $\mathbf{x}_i$ is a vector of coded symbols, its APP is given by the soft channel decoder. If it is a vector of pilot symbols, this probability is either 0 or 1, given by the *a priori* information available at the receiver about the pilots.

function as

$$Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) \triangleq \sum_{\mathbf{x}_i} \log p\left(y_i\left|\mathbf{x}_i,\mathbf{h}_i\right.\right) p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right) \quad (2.8)$$

With complex Gaussian white noise, the above $\log p\left(y_i\left|\mathbf{x}_i,\mathbf{h}_i\right.\right)$ as a function of the variables $\mathbf{x}_i$ and $\mathbf{h}_i$ can be obtained in closed form as $-\left\|y_i-\mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\Big/N_0$. We see that once $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ is computed by BP channel decoding, $Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)$ as a function of $\mathbf{h}_i$ can be obtained by the weighted sum of $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ with weight $-\left\|y_i-\mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\Big/N_0$ over different possible values of $\mathbf{x}_i$. The overall Q function is the sum of symbol-wise Q functions:

$$Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) = \sum\nolimits_{i=1}^{L} Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) \quad (2.9)$$

Using (2.9), the M-step in (2.4) is equivalent to

$$\widehat{\mathbf{h}}^{(k)} = \arg\max_{\mathbf{h}}\left(p\left(\mathbf{h}\right)\cdot\prod\nolimits_{i=1}^{L} e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}\right) \quad (2.10)$$

To see what will happen in the M-step, let us employ the Gauss-Markov channel model (2.2) and factorize $p\left(\mathbf{h}\right)$ as

$$p\left(\mathbf{h}\right) = \prod\nolimits_{i=1}^{L} p\left(\mathbf{h}_i\left|\mathbf{h}_{i-1}\right.\right) \quad (2.11)$$

where $p\left(\mathbf{h}_1\left|\mathbf{h}_0\right.\right) = p\left(\mathbf{h}_1\right) = \mathcal{CN}\left(\mathbf{h}_1:\mathbf{0},\mathbf{Q}\right)$ with $\mathbf{Q} \triangleq diag\left(\left[\sigma_A^2,\sigma_B^2\right]\right)$ is the priori information of $\mathbf{h}_1$, and

$$p\left(\mathbf{h}_i\left|\mathbf{h}_{i-1}\right.\right) = \mathcal{CN}\left(\mathbf{h}_i:\mathbf{m}_{\mathbf{h}_i|\mathbf{h}_{i-1}},\mathbf{K}_{\mathbf{h}_i|\mathbf{h}_{i-1}}\right)$$

for $i \geq 2$ is the Markovian transition probability defined by (2.2). Specifically, $\mathbf{m}_{\mathbf{h}_i|\mathbf{h}_{i-1}} = \alpha\mathbf{h}_{i-1}$, $\mathbf{K}_{\mathbf{h}_i|\mathbf{h}_{i-1}} = \left(1-\alpha^2\right)\mathbf{Q}$ for $i \geq 2$.

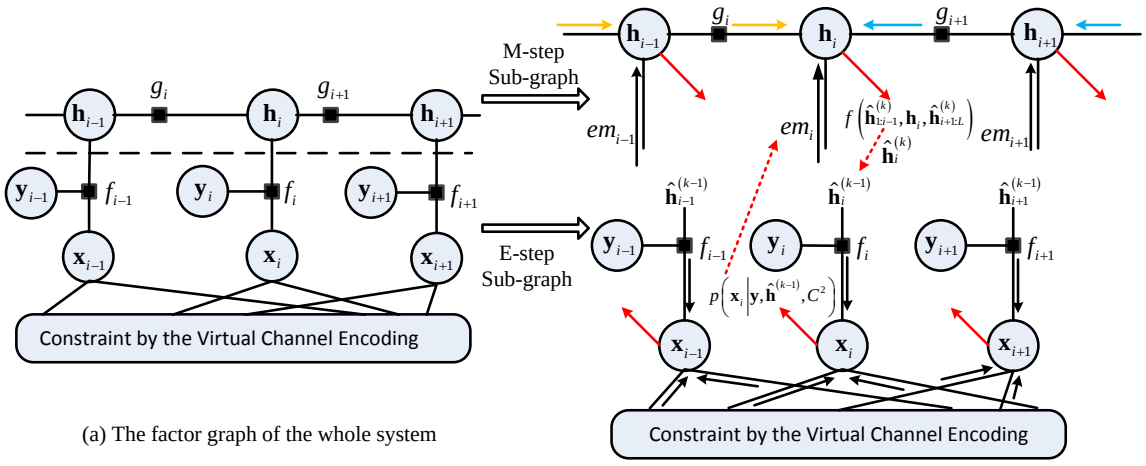
Substituting (2.11) into (2.10) and after some manipulations, we observe that finding $\widehat{\mathbf{h}}^{(k)}$ amounts to finding

$$\widehat{\mathbf{h}}_i^{(k)} = \arg\max_{\mathbf{h}_i} f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)}, \mathbf{h}_i, \widehat{\mathbf{h}}_{i+1:L}^{(k)}\right), \forall i \tag{2.12}$$

where

$$\begin{aligned}
& f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)}, \mathbf{h}_i, \widehat{\mathbf{h}}_{i+1:L}^{(k)}\right) \\
&= \max_{\mathbf{h}_{1:i-1}, \mathbf{h}_{i+1:L}} \left\{ \prod_{j=1}^{L} p\left(\mathbf{h}_j \left| \mathbf{h}_{j-1}\right.\right) \prod_{j=1}^{L} e^{Q_j\left(\mathbf{h}_j \left| \widehat{\mathbf{h}}^{(k-1)}\right.\right)} \right\} \\
&= \underbrace{\max_{\mathbf{h}_{1:i-1}} \left\{ p\left(\mathbf{h}_1 \left| \mathbf{h}_0\right.\right) \prod_{j=1}^{i-1} \left(e^{Q_j\left(\mathbf{h}_j \left| \widehat{\mathbf{h}}^{(k-1)}\right.\right)} p\left(\mathbf{h}_j \left| \mathbf{h}_{j-1}\right.\right) \right) \right\}}_{(1)} \times \\
& \underbrace{e^{Q_i\left(\mathbf{h}_i \left| \widehat{\mathbf{h}}^{(k-1)}\right.\right)}}_{(2)} \times \underbrace{\max_{\mathbf{h}_{i+1:L}} \left\{ \prod_{j=i+1}^{L} \left(e^{Q_j\left(\mathbf{h}_j \left| \widehat{\mathbf{h}}^{(k-1)}\right.\right)} p\left(\mathbf{h}_j \left| \mathbf{h}_{j-1}\right.\right) \right) \right\}}_{(3)}
\end{aligned} \tag{2.13}$$

We can now solve the M-step by a message passing algorithm that implements the max-product rule [17, 18] on a factor graph. We construct the factor graph corresponding to (2.11), and treat $e^{Q_i\left(\mathbf{h}_i \left| \widehat{\mathbf{h}}^{(k-1)}\right.\right)}$ (used in (2.13)) as the input message to the variable node of $\mathbf{h}_i$. The message passing algorithm is a bidirectional algorithm consisting of forward and backward message passing. For each direction, the message passing mechanism is very similar to Viterbi algorithm (VA) for convolutional codes except that the variables $\{\mathbf{h}_i\}$ are continuous. As indicated in (2.13), there are three messages needed for the computation of $\widehat{\mathbf{h}}_i^{(k)}$: message (1) is the result of message passing in the forward direction; message (2) is the input message to $\mathbf{h}_i$; and message (3) is the result of message passing in



(a) The factor graph of the whole system

(b) The two sub-graphs

Figure 2.3: The illustration for the implementation of EM-BP PNC on factor graph: (a) The factor graph for the whole system; (b) The two sub-graphs.

the backward direction. At first sight, it may seem that to obtain the messages in (2.13) (in particular, in the computation associated with the max operation), we need to search over the continuous space of the variables. This would be highly complex. It turns out that that is not the case, as explained below.

Borrowing the jargon from [28], the term $e^{Q_i\left(\mathbf{h}_i|\widehat{\mathbf{h}}^{(k-1)}\right)}$ is the i^{th} EM message and we abbreviate it as

$$em_i = e^{Q_i\left(\mathbf{h}_i|\widehat{\mathbf{h}}^{(k-1)}\right)}.$$

For our application, this EM message has a Gaussian functional form, greatly facilitating message passing on the factor graph. To see the

Gaussian form, we write

$$\begin{aligned}
&\exp\left(Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)\right)\\
&=\exp\left(-\sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\left\|y_i-\mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\Big/N_0\right)\\
&\propto\mathcal{CN}\left(\mathbf{h}_i:\mathbf{m}_{em_i},\mathbf{K}_{em_i}\right)
\end{aligned}$$

with mean vector and covariance matrix

$$\mathbf{m}_{em_i}=\left(\mathbf{K}_{\mathbf{x}_i}+\mathbf{m}_{\mathbf{x}_i}\mathbf{m}_{\mathbf{x}_i}^{\mathrm{H}}\right)^{-1}\mathbf{m}_{\mathbf{x}_i}y_i \tag{2.14a}$$

$$\mathbf{K}_{em_i}=\left(\mathbf{K}_{\mathbf{x}_i}+\mathbf{m}_{\mathbf{x}_i}\mathbf{m}_{\mathbf{x}_i}^{\mathrm{H}}\right)^{-1}N_0 \tag{2.14b}$$

where

$$\begin{aligned}
\mathbf{m}_{\mathbf{x}_i}&=\sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\mathbf{x}_i\\
\mathbf{K}_{\mathbf{x}_i}&=\sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\left(\mathbf{x}_i-\mathbf{m}_{\mathbf{x}_i}\right)\left(\mathbf{x}_i-\mathbf{m}_{\mathbf{x}_i}\right)^{\mathrm{H}}
\end{aligned}$$

are the mean and variance of $\mathbf{x}_i$ computed based on its APP. Therefore, the model underlying the maximization problem (2.10) is a pure linear Gaussian model: all the messages on the factor graph turn out to be Gaussian functions of variables of interest. The maximization problem associated with a linear Gaussian model such as (2.10) can be solved by Gaussian message passing [18] that implements the max-product rule. For Gaussian message passing, we only need to track the mean vectors and covariance matrices of messages, since together they fully characterize the Gaussian distributions. This avoids searching over the whole continuous feasible region, allowing practical implementation of the message passing algorithm for the M-step.

We have now presented the big picture on how to implement EM channel estimation as a message passing algorithm on a factor graph and to combine it with BP message passing for channel decoding. Before going into the detailed derivation of the update equations for the EM-BP message passing, we make two important remarks:

1. We have shown above that, in channel-coded communication systems, the EM messages of the channels are Gaussian. As specified by (2.14), the mean and the variance of the i^{th} EM message are linked to the APP of the i^{th} transmitted symbol. This link connects the BP message passing for channel decoding with EM iterations.

2. The APP of $\mathbf{x}_i$ is a product of three parts:

$$\begin{aligned} &p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right) \\ &= Ap\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right) p\left(y_i \left| \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)} \right.\right), \end{aligned} \tag{2.15}$$

   where A is a constant independent of $\mathbf{x}_i$, $p\left(y_i \left| \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)} \right.\right)$ is the information provided by the observation y_i, and

$$p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right)$$

   is the extrinsic information sent from the BP message passing for channel decoding. When the channel code used has cycles in its factor graph (e.g., LDPC, Turbo codes), the BP message passing algorithm for channel decoding requires multiple iterations to compute the extrinsic information $p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right)$ for all i [17]. Then, the APPs

are updated according to (2.14). Meanwhile, each EM iteration requires one set of such newly computed APPs for the transmitted symbols. Therefore, the theoretical construct of EM requires us to carry out the multiple iterations of BP channel decoding to update the extrinsic information as soon as the channel has been updated to a new estimate.

Fig. 2.3a presents the factor graph of the joint probability of our system, i.e., $p\left(\mathbf{y}, \mathbf{x}, \mathbf{h} \middle| C^2\right)$. On the factor graph, variable nodes $\{y_i, \mathbf{x}_i, \mathbf{h}_i\}$ are denoted by circles; and check nodes $\left\{ f_i \triangleq p\left(y_i \middle| \mathbf{x}_i, \mathbf{h}_i\right), g_i \triangleq p\left(\mathbf{h}_i \middle| \mathbf{h}_{i-1}\right) \right\}$ are denoted by solid squares. Further, according to the factorization $p\left(\mathbf{y}, \mathbf{x}, \mathbf{h} \middle| C^2\right) = p\left(\mathbf{y}, \mathbf{x} | \mathbf{h}, C^2\right) p\left(\mathbf{h}\right)$, we split this overall factor graph into two parts:

E-step subgraph: The subgraph on the lower part of Fig. 2.3b corresponds to the representation of $p\left(\mathbf{y}, \mathbf{x} | \mathbf{h}, C^2\right)$ through factorization (2.6). The E-step of EM is implemented by message passing on this subgraph, thus it is labeled as the E-step subgraph. Note that the subgraph that represents the constraint imposed by virtual channel encoding $I_{C^2}\left(\mathbf{x}\right)$ is embedded inside the E-step subgraph. BP message passing for channel decoding is operating on the subgraph of $I_{C^2}\left(\mathbf{x}\right)$. Thus, we can regard BP as being a sub-step within the E-step also.

M-step sub-graph: The subgraph on the upper part of Fig. 2.3b corresponds to the representation of $p\left(\mathbf{h}\right)$ through factorization (2.11). The M-step of EM is implemented by message passing on this subgraph, thus it is labeled as the M-step subgraph.

In the following, we derive the message update equations for EM-

BP PNC. We denote the message sent by a node x to a node y by $\mu_{x\to y}(\cdot)$. We perform the BP message passing algorithm for *virtual channel decoding* on the E-step subgraph to obtain the soft APP outputs $\left\{p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)\right\}$. The BP message passing algorithm for virtual channel decoding can be derived directly by applying the sum-product rule on the factor graph that incorporates the constraints imposed by *virtual channel encoding* [8, 22] that models the simultaneous transmissions by terminal nodes. For virtual channel decoding within the k^{th} EM iteration, the channel $\mathbf{h}$ from the M-step subgraph is fixed to the previous estimate $\widehat{\mathbf{h}}^{(k-1)}$. Thus, the message $\mu_{f_i\to\mathbf{x}_i}$, which initializes the virtual decoding, is given by

$$\mu_{f_i\to\mathbf{x}_i}(\mathbf{x}_i) = p\left(y_i \left|\mathbf{x}_i, \widehat{\mathbf{h}}_i^{(k-1)}\right.\right) = \mathcal{CN}\left(y_i : \left(\widehat{\mathbf{h}}_i^{(k-1)}\right)^{\mathrm{T}}\mathbf{x}_i, N_0\right) \tag{2.16}$$

for all i. The E-step of EM is completed by sending the EM messages em_i to the M-step subgraph after BP channel decoding, as indicated by the red dotted arrows in Fig. 2.3b. This is fulfilled by computing the mean vectors and covariance matrices of Gaussian EM messages as in (2.14) based on the channel decoding outputs $\left\{p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)\right\}$.

Since the structure of the M-step subgraph is a tree, the M-step (2.10) is exactly implemented by a forward message passing (yellow arrows in upper sub-graph in Fig. 2.3b) and a backward message passing (blue arrows in upper sub-graph in Fig. 2.3b). This observation also coincides with the mathematical expression for message passing by equation (2.13). Due to the assumed Gauss-

Markov channel model and the Gaussian feature of the incoming EM messages, all the messages on the M-step subgraph preserve Gaussianity. For forward message passing, it is required to compute the message

$$\mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) = \max_{\mathbf{h}_{1:i-1}} \left\{ p(\mathbf{h}_1) \prod_{j=2}^{i} em_{j-1} p(\mathbf{h}_j | \mathbf{h}_{j-1}) \right\}$$

from $\mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1})$ recursively. Given that

$$\mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1}) \propto \mathcal{CN}\left(\mathbf{h}_{i-1} : \mathbf{m}^f_{\mathbf{h}_{i-1}}, \mathbf{K}^f_{\mathbf{h}_i - 1}\right)$$

and the Markovian property, we have the following update equations for forward message passing:

$$\begin{aligned} \mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) &= \max_{\mathbf{h}_{i-1}} \left\{ \mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1}) \, em_{i-1} p(\mathbf{h}_i | \mathbf{h}_{i-1}) \right\} \\ &\propto CN\left(\mathbf{h}_i : \mathbf{m}^f_{\mathbf{h}_i}, \mathbf{K}^f_{\mathbf{h}_i}\right) \end{aligned}$$

with

$$\mathbf{m}^f_{\mathbf{h}_i} = \alpha \mathbf{m}^f_{\mathbf{h}_{i-1}} + \alpha \mathbf{K}^f_{\mathbf{h}_{i-1}} \left(\mathbf{K}_{em_{i-1}} + \mathbf{K}^f_{\mathbf{h}_{i-1}}\right)^{-1} \left(\mathbf{m}_{em_{i-1}} - \mathbf{m}^f_{\mathbf{h}_{i-1}}\right) \tag{2.17a}$$

$$\begin{aligned} \mathbf{K}^f_{\mathbf{h}_i} = \alpha^2 \mathbf{K}^f_{\mathbf{h}_{i-1}} + (1 - \alpha^2)\mathbf{Q} - \left(\alpha \mathbf{K}^f_{\mathbf{h}_{i-1}}\right) \left(\mathbf{K}_{em_{i-1}} + \mathbf{K}^f_{\mathbf{h}_{i-1}}\right)^{-1} \\ \times \left(\alpha \mathbf{K}^f_{\mathbf{h}_{i-1}}\right) \end{aligned} \tag{2.17b}$$

The update equations (2.17) are essentially the famous Kalman one-step prediction equations. The backward update equations that compute message $\mu_{g_{i+1} \to \mathbf{h}_i}(\mathbf{h}_i)$ from

$$\mu_{g_{i+2} \to \mathbf{h}_{i+1}}(\mathbf{h}_{i+1}) \propto \mathcal{CN}\left(\mathbf{h}_{i+1} : \mathbf{m}^b_{\mathbf{h}_{i+1}}, \mathbf{K}^b_{\mathbf{h}_{i+1}}\right)$$

are easily obtained similarly by symmetry; we therefore omit them here. Finally, the message flowing out of variable node $\mathbf{h}_i$ (the red solid arrow in the upper sub-graph in Fig. 2.3b) is the product of all the incoming messages

$$\begin{aligned}\mu_{\mathbf{h}_i \to out}(\mathbf{h}_i) &= f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)}, \mathbf{h}_i, \widehat{\mathbf{h}}_{i+1:L}^{(k)}\right) \\ &= \mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) \times em_i \times \mu_{g_{i+1} \to \mathbf{h}_i}(\mathbf{h}_i) \\ &\propto \mathcal{CN}(\mathbf{h}_i : \mathbf{m}_{\mathbf{h}_i}, \mathbf{K}_{\mathbf{h}_i})\end{aligned}$$

where the mean vector $\mathbf{m}_{\mathbf{h}_i}$ is used to update the new estimate of $\mathbf{h}_i$:

$$\begin{aligned}\widehat{\mathbf{h}}_i^{(k)} &= \arg\max_{\mathbf{h}_i} \mu_{\mathbf{h}_i \to out}(\mathbf{h}_i) \\ &= \mathbf{m}_{\mathbf{h}_i} = \mathbf{m}_{\mathbf{h}_i^-} + \mathbf{G}_{\mathbf{h}_i}\left(\mathbf{m}_{em_i} - \mathbf{m}_{\mathbf{h}_i^-}\right)\end{aligned} \tag{2.18}$$

with

$$\begin{aligned}\mathbf{K}_{\mathbf{h}_i^-} &= \left(\left(\mathbf{K}_{\mathbf{h}_i}^f\right)^{-1} + \left(\mathbf{K}_{\mathbf{h}_i}^b\right)^{-1}\right)^{-1} \\ \mathbf{m}_{\mathbf{h}_i^-} &= \mathbf{K}_{\mathbf{h}_i^-}\left(\left(\mathbf{K}_{\mathbf{h}_i}^f\right)^{-1}\mathbf{m}_{\mathbf{h}_i}^f + \left(\mathbf{K}_{\mathbf{h}_i}^b\right)^{-1}\mathbf{m}_{\mathbf{h}_i}^b\right) \\ \mathbf{G}_{\mathbf{h}_i} &= \mathbf{K}_{\mathbf{h}_i^-}\left(\mathbf{K}_{\mathbf{h}_i^-} + \mathbf{K}_{em_i}\right)^{-1}\end{aligned}$$

This completes the M-step of EM. We then iterate back to the E-step with the new estimate.

2.3.3 Initialization and Termination of EM Iteration

EM iteration needs to be bootstrapped with a good initial point; otherwise there is no guarantee that the algorithm will converge to the global maximum [47]. For each block, we obtain the initial $\widehat{\mathbf{h}}^{(0)}$ by minimum mean square error (MMSE) estimation [48] using only the

pilot symbols. Then, the channel coefficients within each data block are simply set to the estimated channel coefficient of the closest pilot.

We repeat the E-step and M-step iteratively. When the number of iterations k reaches a preset maximum limit K, we terminate the EM algorithm after obtaining the final channel estimate $\widehat{\mathbf{h}}^{(K)}$. Substituting $\widehat{\mathbf{h}}^{(K)}$ into the signal model (2.1) as the real channel $\mathbf{h}$, we could carry out a final round of channel decoding to obtain $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$, where $\mathbf{x}$ is the overall codeword pair. The objective of this final channel decoding is consistent with Step 2 in Section 2.3.1. However, instead of finding $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$ for channel decoding, we choose to modify Step 2 by still employing BP for virtual channel decoding to find $p(\mathbf{x}_i|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2)$ for each and every symbol pair $\mathbf{x}_i$. There are three reasons for this. First, we already have the virtual channel decoder at hand in the factor graph implementation of EM-BP PNC receiver. Second, for many advanced channel codes (e.g., LDPC, Turbo code), the decoding process finds $p(\mathbf{x}_i|\mathbf{y},\widehat{\mathbf{h}}^{(K)}C^2)$ rather than $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$, because finding $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$ for all possible codewords is generally a difficult computation-intensive problem. Finding $p(\mathbf{x}_i|\mathbf{y},\widehat{\mathbf{h}}^{(K)}C^2)$ for all i also may be treated as a sort of approximation of the original objective 2. Third, even if we could find $p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$, it would not be easy to obtain the ML network-coded message because the ML network-coded codeword is given by

$$\widehat{\mathbf{x}}_R = \widehat{\mathbf{x}_A \oplus \mathbf{x}_B} = \arg\max_{\mathbf{x}_R} \sum_{\substack{\mathbf{x}: \\ \mathbf{x}_A \oplus \mathbf{x}_B = \mathbf{x}_R}} p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(K)},C^2\right.\right)$$

where $\mathbf{x}_A \triangleq \{x_{A,i}\}$ and $\mathbf{x}_B \triangleq \{x_{B,i}\}$ are the codeword transmitted by nodes A and B. The complexity for the decoding of $\widehat{\mathbf{x}}_R$ is exponential in the length of codewords. By contrast, the ML network-coded symbol is much easier to find from $p(\mathbf{x}_i|\mathbf{y}, \widehat{\mathbf{h}}^{(K)}C^2)$. It is given by

$$\widehat{x_{A,i} \oplus x_{B,i}} = \arg\max_{x} \sum_{\substack{\mathbf{x}_i: \\ x_{A,i} \oplus x_{B,i} = x}} p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2 \right.\right).$$

To summarize, with the final channel estimate $\widehat{\mathbf{h}}^{(K)}$, we can use the BP message passing for virtual channel decoding in the factor graph to compute the final decoding results $p(\mathbf{x}_i|\mathbf{y}, \widehat{\mathbf{h}}^{(K)}C^2)$ and $p\left(s_{A,j}, s_{B,j} \left| \mathbf{y} \right.\right.$ $\left.\left. \widehat{\mathbf{h}}^{(K)}, C^2 \right.\right)$ at the same time. Then, the network-coded source message is obtained by

$$\widehat{s_{A,j} \oplus s_{B,j}} = \arg\max_{s} \sum_{\substack{s_{A,j}, s_{B,j}: \\ s_{A,j} \oplus s_{B,j} = s}} p\left(s_{A,j}, s_{B,j} \left| \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2 \right.\right) \quad (2.19)$$

for all j. After that, the relay channel-encodes the network-coded source message and broadcasts the channel-coded message to nodes A and B in time slot 2. Obtaining the estimate of the network-coded message from $p\left(s_{A,j}, s_{B,j} \left| \mathbf{y}, C^2 \right.\right)$, which is in turn decoded by imagining a virtual encoder as the source of the information, is referred to as the joint CNC process in [8, 16]. Whereas [8, 16] assumes the channel coefficients are perfectly known, here we assume the channel coefficients are unknown and need to be estimated as part of the joint CNC process. Our EM-BP factor graph framework seamlessly bridges the channel estimation process and the joint CNC process. We summarize the EM-BP message passing implementation

Algorithm 1 The EM-BP Message Passing Implementation of Joint Channel Estimation and Channel Decoding in PNC Systems

Initialization:

compute $\widehat{\mathbf{h}}^{(0)}$ by MMSE estimation from pilot symbols;

for $i = 1$ to K **do**

The E-step of EM

compute the inputs message to virtual decoding as (2.16), with the tentative channel estimate $\widehat{\mathbf{h}}^{(k-1)}$;

compute the soft output APPs, $p\left(\mathbf{x}_i|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$, from BP message passing for virtual channel decoding; the number of iterations in the virtual channel decoding is denoted by N_{cd1};

compute the mean vectors and covariance matrices of the EM messages as (14);

The M-step of EM

perform the forward message passing for EM channel estimation as (17);

perform the backward message passing for EM channel estimation similarly to (17);

compute the new channel estimate $\widehat{\mathbf{h}}^{(k)}$ as (18);

end for

Termination:

perform BP message passing for the final virtual channel decoding, with the final channel estimate $\widehat{\mathbf{h}}^{(K)}$; the number of iterations in the virtual channel decoding is denoted by N_{cd2};

compute the network-coded information symbols as (19).

of joint channel estimation and channel decoding in PNC systems in Algorithm 1, where N_{cd1} denotes the number of iterations for the BP channel decoding within the EM iteration; and N_{cd2} denotes the number of iterations for the final BP channel decoding after the termination of EM.

2.4 The Application of SAGE-BP to PNC

SAGE sequentially updates a subset of parameters. Doing so essentially decomposes a higher dimension problem into several lower dimension sub-problems, since only a subset of the parameters are estimated and updated according to the EM mechanism each time. It can be shown that the sequential updates of subsets of parameters always still guarantee convergence [26]. In this section, we apply the theory of SAGE to the problem of joint channel estimation and channel decoding in PNC systems. We also extend the framework of EM-BP message passing over factor graph to SAGE-BP message passing. The motivation for introducing SAGE is to reduce the complexity of the channel estimation part of our framework. To our best knowledge, this is the first attempt to apply SAGE for channel estimation in PNC systems. In fact, in the context of general estimation problems, this is probably the first attempt that integrates the use of SAGE and BP on a unified factor graph.

In our setting, it is natural to break up the problem of estimating $\mathbf{h}$ into two sub-problems, estimation of $\mathbf{h}_A$ and estimation of $\mathbf{h}_B$. That is, we update the channel of node A, $\mathbf{h}_A \stackrel{\Delta}{=} \{h_{A,i}\}$ and the channel

of node B, $\mathbf{h}_B \triangleq \{h_{B,i}\}$ separately and alternatively. When updating one channel, we keep the other channel fixed. We formulate the k^{th} SAGE iteration as a two-stage process.

The 1^{st} stage of the k^{th} iteration: $\mathbf{h}_B$ is fixed to $\widehat{\mathbf{h}}_B^{(k-1)}$; and $\mathbf{h}_A$ is updated by

$$\widehat{\mathbf{h}}_A^{(k)} = \arg\max_{\mathbf{h}_A} \left(p\left(\mathbf{h}_A\right) \cdot \prod\nolimits_{i=1}^{L} e^{Q_{A,i}\left(\mathbf{h}_{A,i} \middle| \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right)} \right) \tag{2.20}$$

where

$$\begin{aligned} &Q_{A,i}\left(\mathbf{h}_{A,i} \middle| \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right) \\ &= \sum_{\mathbf{x}_i} \log p\left(y_i \middle| \mathbf{x}_i, \mathbf{h}_{A,i}, \widehat{\mathbf{h}}_{B,i}^{(k-1)}\right) p\left(\mathbf{x}_i \middle| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right) \end{aligned} \tag{2.21}$$

is the Q function of $\mathbf{h}_{A,i}$.

The 2^{nd} stage of the k^{th} iteration: $\mathbf{h}_A$ is fixed to $\widehat{\mathbf{h}}_A^{(k)}$; and $\mathbf{h}_B$ is updated by

$$\widehat{\mathbf{h}}_B^{(k)} = \arg\max_{\mathbf{h}_B} \left(p\left(\mathbf{h}_B\right) \cdot \prod\nolimits_{i=1}^{L} e^{Q_{B,i}\left(\mathbf{h}_{B,i} \middle| \widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}\right)} \right) \tag{2.22}$$

where

$$\begin{aligned} &Q_{B,i}\left(\mathbf{h}_{B,i} \middle| \widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}\right) \\ &= \sum_{\mathbf{x}_i} \log p\left(y_i \middle| \mathbf{x}_i, \widehat{\mathbf{h}}_{A,i}^{(k)}, \mathbf{h}_{B,i}\right) p\left(\mathbf{x}_i \middle| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right) \end{aligned} \tag{2.23}$$

is the Q function of $\mathbf{h}_{B,i}$.

The computations of Q functions (2.21) and (2.23) for all i correspond to the E-step; the maximization problems (2.20) and (2.22) correspond to the M-step. We still employ BP channel decoding to compute the APPs used in the Q functions of SAGE. The framework of message passing on a factor graph is still applicable to

SAGE-BP. We describe the key steps on how to transform the SAGE-BP computation onto a factor graph in the following.

SAGE breaks up the EM channel estimation into two sub-problems. Each sub-problem is solved by message passing on a factor subgraph. For the M-step of SAGE, the factor subgraph that solves (2.20) ((2.22)) only consists of the channel variables of terminal node A (B): $\{h_{A,i}\}$ ($\{h_{B,i}\}$). The message passing algorithms for computing $\widehat{\mathbf{h}}_A^{(k)}$ and $\widehat{\mathbf{h}}_B^{(k)}$ are initialized with the incoming EM messages $\left\{em_{A,i} \triangleq e^{Q_{A,i}(\cdot)}\right\}$ and $\left\{em_{B,i} \triangleq e^{Q_{B,i}(\cdot)}\right\}$, respectively. It can be proven that Gaussianity is preserved. Namely, substituting (2.21) into $em_{A,i} = e^{Q_{A,i}(\cdot)}$ leads to the following Gaussian expression:

$$\begin{aligned} em_{A,i} &= \exp\left(Q_{A,i}\left(h_{A,i}\left|\widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right.\right)\right) \\ &\propto \mathcal{CN}\left(h_{A,i} : m_{x_{A,i}^*} y_i - m_{x_{A,i}^* \cdot x_{B,i}} \widehat{h}_{B,i}^{(k-1)}, N_0\right) \end{aligned} \tag{2.24}$$

where $x_{A,i}^*$ is the conjugate of $x_{A,i}$, and $m_{x_{A,i}^*}$, $m_{x_{A,i}^* \cdot x_{B,i}}$ are given by

$$\begin{aligned} m_{x_{A,i}^*} &= \sum_{x_{A,i}} \left(\sum_{x_{B,i}} p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right)\right) x_{A,i}^* \\ &= \sum_{x_{A,i}} p\left(x_{A,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right) x_{A,i}^* \\ m_{x_{A,i}^* \cdot x_{B,i}} &= \sum_{x_{A,i}^* \cdot x_{B,i}} \left(\sum_{\mathbf{x}_i : x_{A,i}^* \cdot x_{B,i} = x} p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right)\right) \\ &\qquad \times x_{A,i}^* \cdot x_{B,i} \\ &= \sum_{x_{A,i}^* \cdot x_{B,i}} p\left(x_{A,i}^* \cdot x_{B,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right) x_{A,i}^* \cdot x_{B,i} \end{aligned} \tag{2.25}$$

Based on this Gaussian expression for $\{em_{A,i}\}$ and the Gauss-Markov channel model of $\{h_{A,i}\}$ in (2.2), we can solve the M-step of the first stage (2.20) by a Gaussian message passing algorithm that is almost similar to the one derived in Section 2.3 for EM, with the difference that the state space here only includes the channel of one terminal

node. Thus, all the length-2 channel vectors, in the algorithm, are reduced to scalars. Similarly, we can derive the Gaussian expression for $\{em_{B,i}\}$ and solve the M-step of the second stage (2.22) by the same Gaussian message passing algorithm. We round off the discussion of SAGE-BP PNC with the following remarks:

1. In (2.25), $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)$ is the decoding result for the transmitted symbol of node A, $x_{A,i}$;

$$p\left(x_{A,i}^*\cdot x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)$$

   can be regarded as the decoding result for the variable $x_{A,i}^*\cdot x_{B,i}$. Based on these decoding results, the posterior means of $x_{A,i}^*$ and $x_{A,i}^*\cdot x_{B,i}$ are computed according to (2.25) and used in (2.24). The intuitive interpretation of the expression for the mean of the Gaussian expression in (2.24) is that it corresponds to an interference cancelation process for the target channel.

2. Comparing the EM messages in (2.14) and (2.24), the message passing implementation of EM-BP needs to compute the mean vector and the covariance matrix of the transmitted symbol pair; that of SAGE-BP needs to compute the mean of the symbol transmitted from the target channel and the cross-correlation between the two symbols $x_{A,i}, x_{B,i}$.

3. We compare the complexities of PNC in the following. Let us just focus on the algorithms within the EM/SAGE-BP iteration loop, since the complexities of initialization and termination are the same for EM-BP and SAGE-BP. The results are summarized

Table 2.1: Complexity of EM-BP PNC per iteration

	EM-BP PNC
channel estimation	$O\left(2\left(5N_u^3+4N_u^2+3N_u\right)L\right)$
computing EM messages	$O\left(\left(N_u^3+\left(N_m^{N_u}+3\right)N_u^2+\left(2N_m^{N_u}+1\right)N_u+2N_m^{N_u}-2\right)L\right)$
channel decoding	$O\left(N_{cd1}N_m^{N_u}\left(2N_m^{N_u}-1\right)l\right)$

Table 2.2: Complexity of SAGE-BP PNC per iteration

	SAGE-BP PNC
channel estimation	$O\left(24N_uL\right)$
computing EM messages	$O\left(N_u\left(2N_m^{N_u}+2N_m+1\right)L\right)$
channel decoding	$O\left(N_uN_{cd1}N_m^{N_u}\left(2N_m^{N_u}-1\right)l\right)$

in Table 2.1 and Table 2.2. We denote the number of elements in the channel vector $\mathbf{h}_i$ by N_u. For our TWRC system, $N_u = 2$. We denote the size of the modulation by N_m (e.g., $N_m = 2$ for BPSK and $N_m = 4$ for QPSK). For EM-BP, the computation of (2.17) needs $O\left(5N_u^3+4N_u^2+3N_u\right)$ operations.[5] Therefore, the complexity of channel estimation in EM-BP is $O\left(2\left(5N_u^3+4N_u^2+3N_u\right)L\right)$, where the factor 2 is due to one forward and one backward message passing, L is the frame

[5]Strictly speaking, the complexities of matrix computations are different for different computing algorithm. Here, we assume that the inversion of an $n \times n$ matrix requires $O\left(n^3\right)$ computations. The product of an $n \times m$ matrix and an $m \times p$ matrix needs $O\left(nmp\right)$ computations for our analysis. Some fast algorithms of matrix computations can reduce these complexities to some extent.

length. By counting the operations involved in (2.14), we can figure out that the complexity of computing the EM messages for EM-BP is

$$O\left(\left(N_u^3+\left(N_m^{N_u}+3\right)N_u^2+\left(2N_m^{N_u}+1\right)N_u+2N_m^{N_u}\ -\ 2\right)L\right).$$

For SAGE-BP, there is no matrix operation, and the computation of (16) becomes $O(12)$. Since we perform once bidirectional message passing for each node, the complexity of channel estimation in SAGE-BP is $O(24N_uL)$. The complexity of computing EM messages for SAGE-BP as in (2.24) is $O\left(N_u\left(2N_m^{N_u}+2N_m+1\right)L\right)$. From the above results, we can see that SAGE-BP simplifies the complexity of channel estimation by removing the need for matrix inversions and multiplications. Now, let us look at the channel decoding part. EM-BP updates $\widehat{\mathbf{h}}_A^{(k)}$ and $\widehat{\mathbf{h}}_B^{(k)}$ simultaneously; and requires one virtual channel decoding in each iteration. By contrast, SAGE-BP requires two virtual channel decodings, one before the update of $\widehat{\mathbf{h}}_A^{(k)}$ and one before the update of $\widehat{\mathbf{h}}_B^{(k)}$. That is, each time one of the channels is updated, the APPs of $\mathbf{x}_i$ will need to be re-computed before the other channel is updated. To analyze the complexity of virtual channel decoding, let us consider the regular repeat accumulate (RA) code [49] (used in our simulations). The codeword length is l, and there is no operation of channel decoding on the $L-l$ pilots symbols in each frame. The BP message passing requires $O\left(N_m^{N_u}\left(2N_m^{N_u}-1\right)\right)$ computations per check node. There are

l check nodes on the factor graph of the RA code. We perform N_{cd1} iterations for each round of channel decoding. It follows that the complexity of channel decoding in EM-BP is $O\left(N_{cd1}N_m^{N_u}\left(2N_m^{N_u}-1\right)l\right)$; the complexity of channel decoding in SAGE-BP is $O\left(N_uN_{cd1}N_m^{N_u}\left(2N_m^{N_u}-1\right)l\right)$, where the factor $N_u=2$ is due to the two virtual channel decodings in each SAGE iteration. Therefore, SAGE-BP simplifies channel estimation but adds complexity to channel decoding. One simple method to maintain the same complexity in the channel decodings of SAGE-BP and EM-BP is to make the N_{cd1} in SAGE-BP equal to half of the N_{cd1} in EM-BP. This will cause SAGE-BP to suffer some performance loss. We will study this by simulations in the next section.

4. The decoding results $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)$, $p\left(x_{A,i}^*\cdot x_{B,i}\right|\ \mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\Big)$ are both obtained based on the outputs of BP message passing for virtual channel decoding (see (2.25)). Another way to do this is to employ parallel interference cancelation (PIC) [50, 51] to compute $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)$ $\left(p\left(x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)\right)$ and then the joint probability $p(\ \mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\Big)$. Thus, we can still compute the decoding result for $x_{A,i}^*\cdot x_{B,i}$ from $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(k-1)},\widehat{\mathbf{h}}_B^{(k-1)},C^2\right.\right)$. To compute these decoding results, PIC employs two single-user channel decoders for the two terminal nodes. And it requires iterative message passing between the two single-user channel decoders, besides the

iterative message passing within each of the channel decoder. Since PIC is well implemented by BP message passing [51], we can also incorporate it into our factor graph framework for SAGE-BP. This method is referred to as SAGE-BP PIC. After the termination of SAGE iteration, SAGE-BP PIC obtains $\left\{p\left(s_{A,j}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(K)},\widehat{\mathbf{h}}_B^{(K)},C^2\right.\right)\right\}$, $\left\{p\left(s_{B,j}\left|\mathbf{y},\widehat{\mathbf{h}}_A^{(K)},\widehat{\mathbf{h}}_B^{(K)},C^2\right.\right)\right\}$ from the final channel decoding. Then, we have, from the rule of sum-product algorithms,

$$\begin{aligned} p(s_{A,j},s_{B,j}|\mathbf{y},\widehat{\mathbf{h}}_A^{(K)},\widehat{\mathbf{h}}_B^{(K)},C^2) \approx\ & p(s_{A,j}|\mathbf{y},\widehat{\mathbf{h}}_A^{(K)},\widehat{\mathbf{h}}_B^{(K)},C^2) \\ & \times p(s_{B,j}|\mathbf{y},\widehat{\mathbf{h}}_A^{(K)},\widehat{\mathbf{h}}_B^{(K)},C^2) \end{aligned}$$

for all j, from which we can perform network coding as in (18). Since PIC employs single-user decoding whose complexity does not increase exponentially with the number of nodes N_u (as virtual channel decoding does), SAGE-BP PIC can make our framework scalable with the number of nodes (if we want to extend our treatment to beyond TWRC). However, we will see in Section 2.5 that the performance of SAGE-BP PIC is not as good as SAGE-BP PNC. The reason is that there is a small cycle between node $x_{A,i}$ and node $x_{B,i}$ for each i in the factor graph of PIC, and these small cycles degrade the performance of the BP algorithm. On the factor graph of virtual channel decoding, we cluster $x_{A,i}$ and $x_{B,i}$ together as one node $\mathbf{x}_i$. Since $x_{A,i}$ and $x_{B,i}$ now become a single variable node, the edges connecting them disappear. By this clustering technique [17], the small cycles between $x_{A,i}$ and $x_{B,i}$ are removed from the factor graph

of virtual channel decoding.

5. We comment here that if nodes A and B employ different channel encoders, a corresponding virtual channel decoding does not exist. Then, we can only apply a channel decoding method for MUD systems (such as PIC) to compute the APP of $x_{A,i}$ and the APP of $x_{B,i}$, respectively. By replacing the virtual channel decoding with PIC channel decoding, we can deal with the set-up of different terminal nodes using different channel encoders under the EM/SAGE-BP framework.

6. Since the complexities of both SAGE for channel estimation and BP for PIC channel decoding are scalable with the number of nodes, we can apply SAGE-BP PIC to systems where collisions of more than two signals are possible [52, 53]. The study of this application awaits future work.

2.5 Simulation Results

In this section, we perform computer simulations to evaluate the performances of the proposed schemes. We assume the channels of both terminal nodes have the same average power $\sigma_A^2 = \sigma_B^2$. Unless stated otherwise, the channel correlation coefficient α is set to 0.99. The regular RA code with coding rate 1/3 is employed. In the case of BPSK modulation, each frame has 1024 information bits (thus, 3072 coded modulated data symbols); and in the case of QPSK modulation, each frame has 2048 bits (also 3072 coded modulated data symbols).

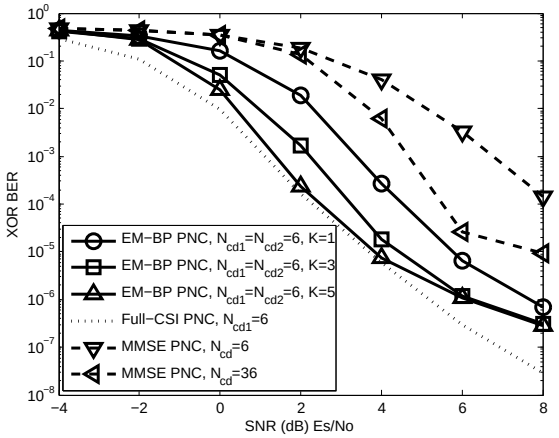


Figure 2.4: The BER results of the EM-BP PNC receiver with BPSK modulation.

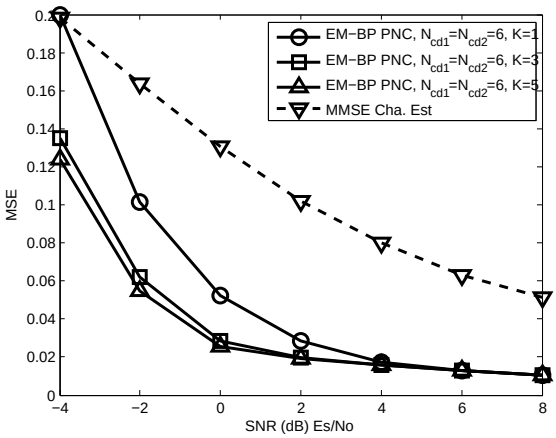


Figure 2.5: The MSE results for the estimated channels of the EM-BP PNC receiver with BPSK modulation.

We insert two pilots every Δ data symbols. Unless stated otherwise, the pilot interval Δ is set to 16 (this corresponds to a 11.1% pilot load). The two terminal nodes adopt orthogonal pilots, wherein $P_1 = 1, P_2 = 1$ for node A and $P_1 = 1, P_2 = -1$ for node B. All simulation results presented here are obtained by averaging over 10^5 pairs of frames. The signal to noise ratio (SNR) is defined as E_s/N_0 where E_s is the energy per coded bit. Specifically, for coding rate 1/3, $E_s = E_b/3$ where E_b is energy per source bit.

2.5.1 Performance of EM-BP PNC

First, we investigate the performance of the EM-BP PNC receiver. We evaluate the BER of the network-coded messages and the mean square error (MSE) of the estimated channels. The results of the PNC receiver using just a one-shot MMSE channel estimation (this is equivalent to our EM-BP PNC receiver with $K = 0$) and the ideal

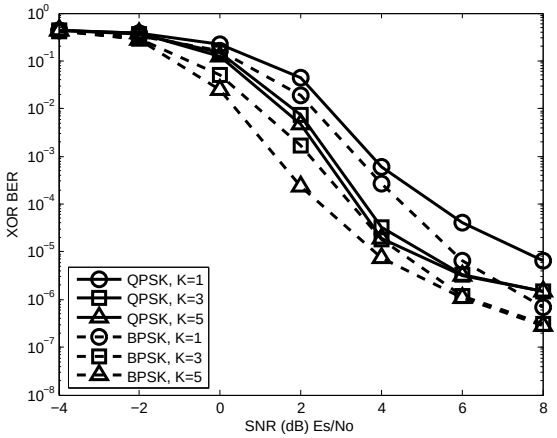


Figure 2.6: The BER results of the EM-BP PNC receiver with QPSK modulation.

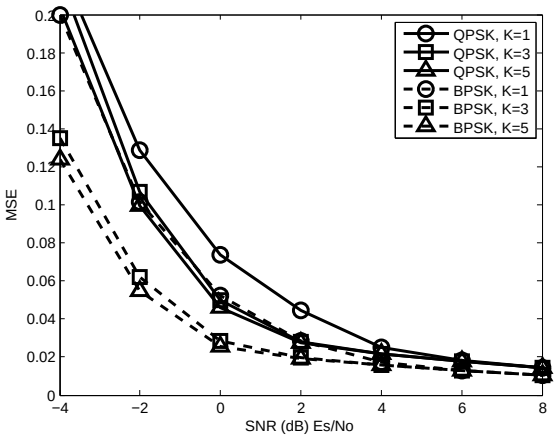


Figure 2.7: The MSE results for the estimated channels of the EM-BP PNC receiver with QPSK modulation.

PNC receiver with the full channel state information (CSI) will be given as benchmarks.

Fig. 2.4 presents the BER results of the three receivers: the EM-BP PNC, the MMSE PNC, and the Full-CSI PNC; and Fig. 2.5 presents their MSE results for the estimated channel. The BPSK modulation is used for all receivers. The number of channel decoding iterations in the MMSE PNC receiver and the Full-CSI PNC receiver is denoted by N_{cd}. Recall that for the EM-BP PNC, K is the number of EM iterations, N_{cd1} is the number of iterations for the virtual channel decoding in the E-step of each EM iteration; and N_{cd2} is the number of iterations in the final virtual channel decoding at the conclusion of all EM iterations. Since we use the MMSE channel estimation to initialize the EM-BP algorithm, we expect the EM-BP algorithm to give more accurate channel estimation than the one-shot MMSE channel estimation. This is confirmed by our simulation results in

Fig. 2.5. In particular, the channel estimation accuracy in EM-BP improves progressively with the number of iterations. We can also observe that the first EM iteration can already extract most of the gain in MSE. Our simulations also indicates that the EM-BP PNC algorithm has almost converged after $K = 5$ iterations. These MSE improvements by EM-BP PNC are reflected into BER results. Comparing the BER results of EM-BP PNC receiver and MMSE PNC with $N_{cd} = 6$ in Fig. 2.4, we can see that there is a 4 dB gain by EM-BP PNC just after the first EM iteration ($K = 1$). There is a 6 dB gain after EM has converged ($K = 5$). Furthermore, the BER result of EM-BP PNC at $K = 5$ can approach the BER of the Full-CSI PNC very well.

For a fairer comparison, let us examine the performance of EM-BP PNC with $N_{cd1} = N_{cd2} = 6$, $K = 5$, and the performance of MMSE PNC receiver with $N_{cd} = 36$: i.e., the total numbers of channel decoding iterations are the same in the two cases. We observe that for BER, EM-BP PNC receiver has around 4 dB gain over MMSE PNC. The observed error floor in both BER and MSE as SNR increases are due to the time-varying property of the channel, which is also analyzed and reported in [54]. Essentially, even if the receiver noise is zero, the channel randomness in between pilots induces uncertainty that cannot be removed entirely regardless of the SNR. Effectively, the channel randomness in between pilots is a source of noise besides the thermal circuit noise in the receiver.

Fig. 2.6 and Fig. 2.7 present the BER and MSE results of EM-BP PNC when QPSK modulation is used. The results of EM-

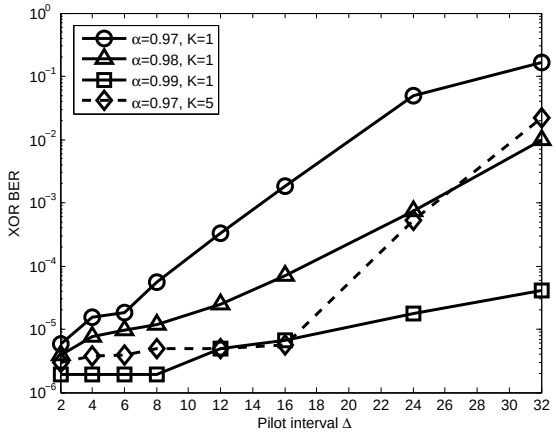


Figure 2.8: The impact of the channel correlation coefficient α on the BER performance of the EM-BP PNC receiver.

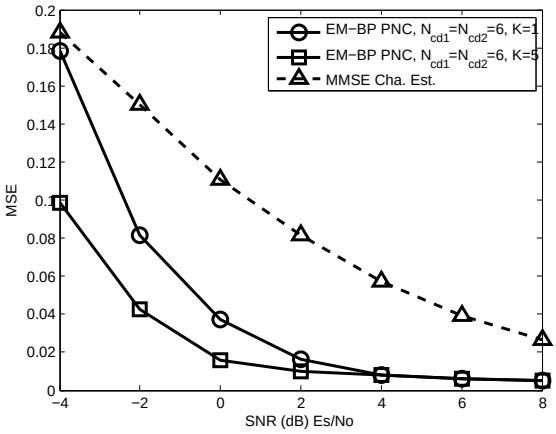


Figure 2.9: The MSE results for the estimated channels of the EM-BP PNC receiver under the Clarke's channel (with normalized maximum Doppler spread 0.005).

BP PNC with BPSK are also shown as benchmarks. For the data here, $N_{cd1} = N_{cd2} = 6$. We can observe that the BER and MSE results for QPSK are slightly worse than that for BPSK. The denser constellation map in QPSK makes the channel estimation tougher. Thus, the MSEs for QPSK are larger than for BPSK. Moreover, the denser constellation makes the channel decoding more sensitive to the channel estimation error. The performance gap in the BER results of the BPSK and QPSK is about 0.5-1 dB in the low SNR regime ($E_s/N_0 < 6$ bB), and about 2 dB in the high SNR regime ($E_s/N_0 > 6$ bB). Since the performance trends of BPSK and QPSK are the same, we just focus on the performance of BPSK hereinafter for simplicity.

We now investigate the impact of channel correlation coefficient α on the BER performance of EM-BP PNC. For virtual channel

decoding, we also set $N_{cd1} = N_{cd2} = 6$. The BER results of EM-BP PNC versus pilot interval Δ for various α are shown in Fig. 2.8. The operating SNR E_s/N_0 is 6 dB. As α decreases from 0.99 to 0.97, the channel varies faster and the BER gets worse. The BER is more sensitive to the pilot load as α gets smaller. When the pilot interval Δ goes from 2 to 32, the BER in a channel with $\alpha = 0.99$ ranges from 10^{-6} to 10^{-5}; however, the BER in a channel with $\alpha = 0.97$ ranges from 10^{-6} to 10^{-1}. To maintain the performance in an environment of rapidly varying channel, we can insert more pilot symbols or increase the number of EM iterations (K). For example, the BER of $\alpha = 0.97$ and $K = 5$ can approach the BER of $\alpha = 0.99$ and $K = 1$ when $\Delta \leq 16$.

In Section 2.3, we derived the proposed message passing algorithm for channel estimation using the first-order Gauss-Markov channel model. Indeed, the Gauss-Markov channel model is an approximation of the real physical channel. Here, we conduct a simulation using a more realistic mobile channel. Our aim is to demonstrate the robustness of the channel-estimation message passing when there is a mismatch between the channel model and the actual realized physical channel. All the set-ups of the simulation are the same as the previous cases except that the actual channel gains are generated according to the Clarke's channel model [11]. The normalized maximum Doppler spread in the Clarke's channel model is set to 0.005. Following the relation between the Doppler spread and the correlation coefficient α established in [55], we set the correlation coefficient to $\alpha = 0.989$ and use it for channel-estimation message passing. We evaluate the

MSEs of the channel estimation outputs, and present the results in Fig. 2.9. We can see that despite the channel mismatch, our channel estimation message passing still works well. We can still observe its MSE gains over the one-shot MMSE channel estimation. Comparing the MSE results of the Charke's channel in Fig. 2.9 with the MSE results of the Gauss-Markov channel in Fig. 2.5, we can see that there is no obvious performance difference between them.

2.5.2 Performance of SAGE-BP PNC

We now investigate the performance of the SAGE-BP PNC receiver. We first focus on SAGE-BP PNC with $N_{cd1} = N_{cd2} = 6$. The BER results are shown in Fig. 2.10, where the BER results of EM-BP PNC with $N_{cd1} = N_{cd2} = 6$ are also presented as benchmarks. It can be observed that SAGE-BP PNC has the same performance as EM-BP PNC. Both SAGE-BP PNC and EM-BP PNC converge to the same performance after $K = 5$ iterations. This is consistent with the fundamental theory of SAGE [26]. Substituting the simulation parameters ($N_{cd1} = 6$, $N_u = 2$, $N_m = 2$, $l = 8L/9$) into Table I, we can find that per EM-BP iteration needs $O(333L)$ computations; per SAGE-BP iteration needs $O(369L)$ computations. The complexity of SAGE-BP now is slightly higher than that of EM-BP due to the more complex channel decoding in SAGE-BP.

For SAGE-BP PNC, since the two-dimension estimation problem of $\mathbf{h} = \{\mathbf{h}_A, \mathbf{h}_B\}$ in the M-step of EM has been decomposed into two one-dimension estimation sub-problems of $\mathbf{h}_A$ and $\mathbf{h}_B$, the complexity

of its channel estimation is smaller than that of EM-BP PNC. To 'equalize' the complexities of the channel decoding of SAGE-BP PNC and EM-BP PNC, we set $N_{cd1} = 3$, $N_{cd2} = 6$ for SAGE-BP PNC, i.e., the total number of iterations for BP channel decoding in the E-step after both $\mathbf{h}_A$ and $\mathbf{h}_B$ are updated is $2 \times N_{cd1} = 6$; and $N_{cd1} = 6$, $N_{cd2} = 6$ for EM-BP PNC. Recall that the complexity of channel estimation in SAGE-BP PNC is smaller than that in EM-BP PNC (see the complexities in Table I). Since we equalize the complexities of the channel decoding processes of EM-BP and SAGE-BP (by halving the number of channel decoding iterations in each SAGE-BP iteration), the overall complexity of SAGE-BP PNC (per SAGE-BP iteration needs $O(219L)$ computations) is now smaller than that of EM-BP PNC because of its less complex channel estimation. From the results in Fig. 2.10, we can see that the performance of SAGE-BP PNC with $N_{cd1} = 3$, $N_{cd2} = 6$ is not as good as that of EM-BP PNC with $N_{cd1} = 6$, $N_{cd2} = 6$.

We next compare the performances of SAGE-BP PIC and SAGE-BP PNC. We report the BER and MSE results in Fig. 2.11 and Fig. 2.12, respectively. We denote the number of iterations for the message passing between the two single-user channel decoders in PIC by P. For SAGE-BP PIC, we can increase P and N_{cd1} (N_{cd2}) to allow convergence of the PIC channel decoding. From our simulation results, PIC converges with $N_{cd1} = N_{cd2} = 18$, $P = 2$. When we continue to increase these numbers of iterations, no observable improvement on performance can be obtained. For SAGE-BP PNC, we can increase N_{cd1} (N_{cd2}) to make the virtual channel decoding

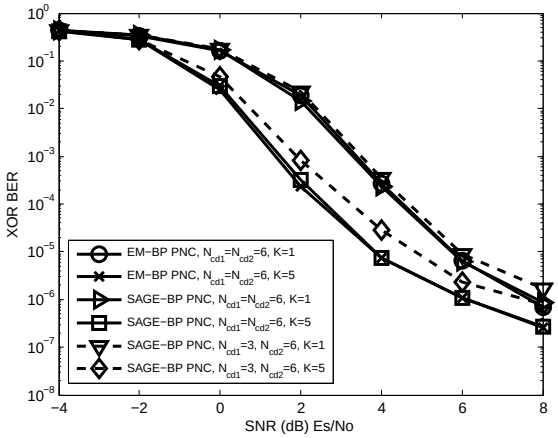


Figure 2.10: The BER results of the SAGE-BP PNC receiver.

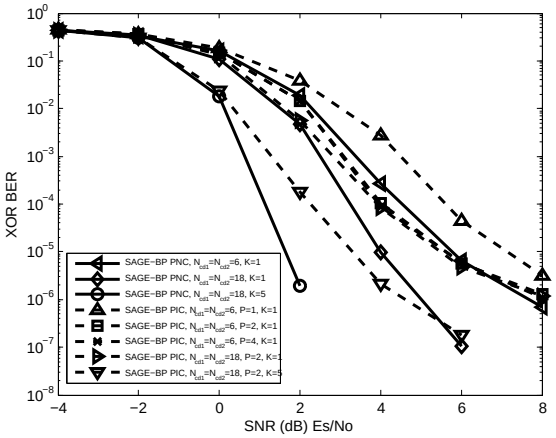


Figure 2.11: The BER results of the SAGE-BP PIC/PNC receiver.

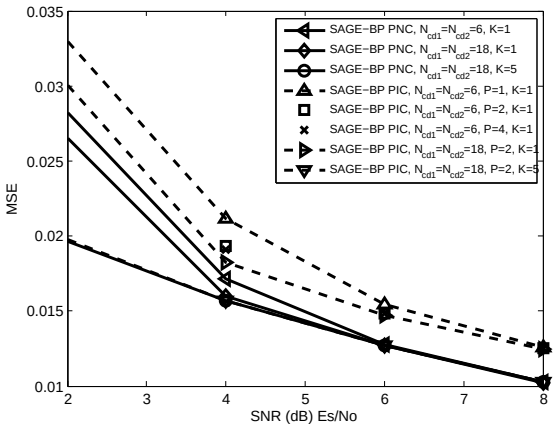


Figure 2.12: The MSE results of the SAGE-BP PIC/PNC receiver.

converge. Virtual channel decoding converges with $N_{cd1} = N_{cd2} = 18$. From the BER results in Fig. 2.11, we can see that the BER of SAGE-BP PIC is not as good as that of SAGE-BP PNC. Specifically, the error floor of SAGE-BP PIC in the high SNR regime is higher than that of SAGE-BP PNC. As explained earlier, this is because there are small cycles in the factor graph of PIC. The APPs of transmitted symbols computed by PIC is not as accurate as the ones computed

by virtual channel decoding for PNC. Therefore, we expect the SAGE channel estimation based on the worse APPs will result in worse estimate results. From the MSE results in Fig. 2.12, we observe that with one SAGE iteration ($K = 1$), SAGE-BP PIC indeed has worse MSE than SAGE-BP PNC. When we increase the number of iterations to allow SAGE-BP PNC and SAGE-BP PIC to converge ($N_{cd1} = N_{cd2} = 18$, $P = 2$, $K = 5$), SAGE-BP PIC can obtain the same MSE as SAGE-BP PNC. However, even with the same MSE, the final channel decoding results of SAGE-BP PIC now is still worse than that of SAGE-BP PNC (see the BERs in Fig. 2.11). This is because the APPs computed by the PIC process are still not as accurate as those computed by the virtual channel decoding process. In other words, as far as the channel estimation is concerned, the performances of both schemes are comparable, but the virtual channel decoding in SAGE-BP PNC gives better estimates of the network-coded symbols.

2.5.3 Comparison with Other Receiver Architectures

In this subsection, we compare our EM-BP PNC receiver with other receiver architectures [23, 34, 35, 42].

We first investigate the performances of two different strategies on how to combine EM with BP channel decoding: our EM-BP strategy and the strategy suggested in [34, 35]. Although [34, 35] do not investigate PNC systems, we can extrapolate their strategy for PNC application. Before we compare the performance results of the two strategies, let us explain a subtle point in the rigorous EM-

BP framework because this is where the strategy in [34, 35] deviates from this framework. We note that the extrinsic information used in (14) is a function of the channel estimate $\widehat{\mathbf{h}}^{(k)}$. According to the theoretical framework of EM, upon a new channel estimate, we should immediately run BP channel decoding to compute new extrinsic information to update the APPs. The new APPs are then used to compute the next channel estimate. Our EM-BP strategy conforms to the above operational sequence. In the following discussion, when we refer to the strategy in [34, 35], we mean the strategy as applied to PNC. In the strategy of [34, 35], each iteration of BP channel decoding is followed by several EM iterations for channel estimation using the same extrinsic information from that single iteration of BP channel decoding. For example, at EM iteration $k = k_1$, $\widehat{\mathbf{h}}^{(k_1-1)}$ was obtained. After one iteration of BP channel decoding, we obtain the new extrinsic information $p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2 \right.\right)$ for all i. For several subsequent EM iterations for channel estimation, indexed by $k = k_1+1, \cdots, k_2-1$, BP channel decoding will not be performed at all. EM updates a new channel estimate $\widehat{\mathbf{h}}^{(k)}$ in each iteration using an approximate APP of $\mathbf{x}_i$ rather than (14). Specifically,

$$\begin{aligned} &p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right) \approx \\ &A \cdot p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2 \right.\right) p\left(y_i \left| \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)} \right.\right). \end{aligned}$$

In particular, an approximation is made on the extrinsic information:

$$\begin{aligned} &p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2 \right.\right) \\ &\approx p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2 \right.\right). \end{aligned}$$

Obviously, this operation does not correctly compute the APP of $\mathbf{x}_i$

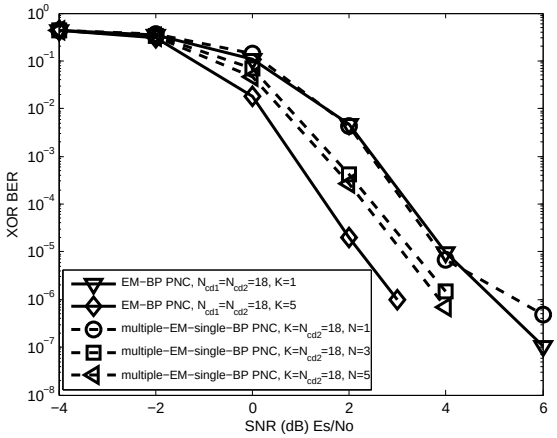


Figure 2.13: Comparing the different strategies on how to combine EM with BP channel decoding: the BER results.

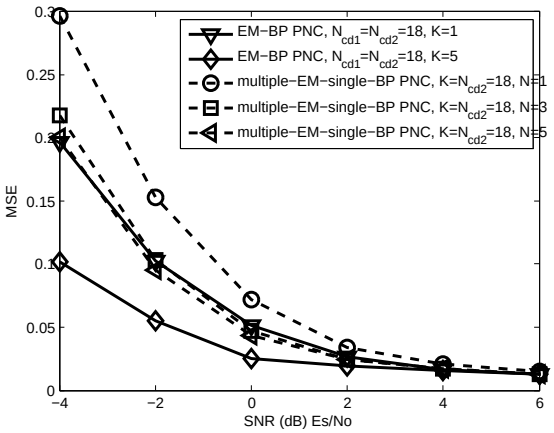


Figure 2.14: Comparing the different strategies on how to combine EM with BP channel decoding: the MSE results

needed by EM for iterations $k = k_1 + 1, \cdots, k_2 - 1$. Moreover, if there are cycles in the factor graph of the channel code (e.g., the RA code used in our simulation experiments), the computation of $p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2 \right.\right)$ with just one BP channel decoding iteration will not accurate either. This strategy is referred to as SP-EM in [34], where SP stands for sum-product. To avoid confusion, we rename it as multiple-EM-single-BP scheme here to represent the fact that there are multiple iterations of EM for each iteration of BP. Our EM-BP is single-EM-multiple-BP in that sense.

The BER results of EM-BP PNC and multiple-EM-single-BP PNC are shown in Fig. 2.13; their MSE results are shown in Fig. 2.14. For EM-BP PNC, there are N_{cd1} iterations for BP channel decoding after each EM iteration; and totally K EM iterations are performed (therefore, altogether there are K EM iterations and $N_{cd1}K$ BP channel decoding iterations). For multiple-EM-single-BP PNC, there

are K EM iterations after each for BP channel decoding iteration; and totally N iterations for BP channel decoding are performed prior to termination (therefore, altogether there are NK EM iterations and N BP channel decoding iterations). After termination, we perform additional $N_{cd2} = 18$ iterations for BP channel decoding in both strategies. The approximate APPs in multiple-EM-single-BP as explained above causes it to deviate from the principle of EM algorithm. The results in Fig. 2.13 and Fig. 2.14 show that compared with EM-BP PNC, multiple-EM-single-BP PNC exhibits worse performances and converges to an inferior operating point. From our simulation results in Fig. 2.13 and Fig. 2.14, we see that no further improvement on the performance of multiple-EM-single-BP PNC can be obtained when N is increased from 5 to 7. In other words, even if we make N very large to equalize the channel decoding complexity of multiple-EM-single-BP PNC with that of EM-BP PNC, the performance of multiple-EM-single-BP PNC will still be worse than that of EM-BP PNC.

Next, we compare our EM-BP PNC receiver and the BP-MM receiver proposed in [23] (see Section 2.1 for the overview of [23]). In EM-BP PNC, EM is employed to accomplish the task of channel estimation; and BP is employed to accomplish the task of channel decoding. EM can find $\widehat{\mathbf{h}}_{MAP}$ when it converges to the global optimal, and in this case the result of the final channel decoding in EM-BP PNC is $p\left(\mathbf{x}_i \left| \widehat{\mathbf{h}}_{MAP}, \mathbf{y}, C^2 \right.\right)$ for all i. This is the target of EM-BP. For EM, the convergence to the global optimal can always be guaranteed by a good initial point [47]. We can also finish the

tasks of both channel estimation and channel decoding using BP alone, as in [23]. In this case, the final target of channel decoding is $p\left(\mathbf{x}_i \left| \mathbf{y}, C^2\right.\right) = \int p\left(\mathbf{x}_i, \mathbf{h} \left| \mathbf{y}, C^2\right.\right) d\mathbf{h}$, which is different from the target of EM-BP. However, the integration over continuous variables required by BP channel estimation is computationally infeasible. BP-MM uses MM as an approximation to circumvent the need for explicit integration (i.e., integration of the approximate Gaussian distribution can be obtained in closed form) [23]. However, as a consequence of the approximation by MM, the optimality of BP cannot be guaranteed (i.e., even if the algorithm converges to the global optimal, it will not be the global optimal associated with the original non-Gaussian distribution). The BER results of BP-MM are shown in Fig. 2.15, where M denotes the number of iterations between channel estimation and channel decoding in BP-MM; and N_{cd} denotes the number of channel decoding iterations in BP-MM. Comparing the BERs of EM-BP PNC and BP-MM, EM-BP PNC is worse in the low SNR regime and it is better in the high SNR regime. Specifically, when both receivers have converged ($M = K = 5$, $N_{cd} = N_{cd1} = N_{cd2} = 18$), EM-BP PNC outperforms BP-MM by 1dB at the BER of 10^{-6}. We believe this gap could be due to the approximation by MM.

As summarized in Section 2.1, there is an EM approach for joint channel estimation and channel decoding in multi-user CDMA systems, proposed in [42]. In [42], channel decoding is implemented by MMSE SIC with separate single-user channel decoders for data from different users; and channel estimation is implemented by EM using APPs obtained from the single-user channel decoders. We compare

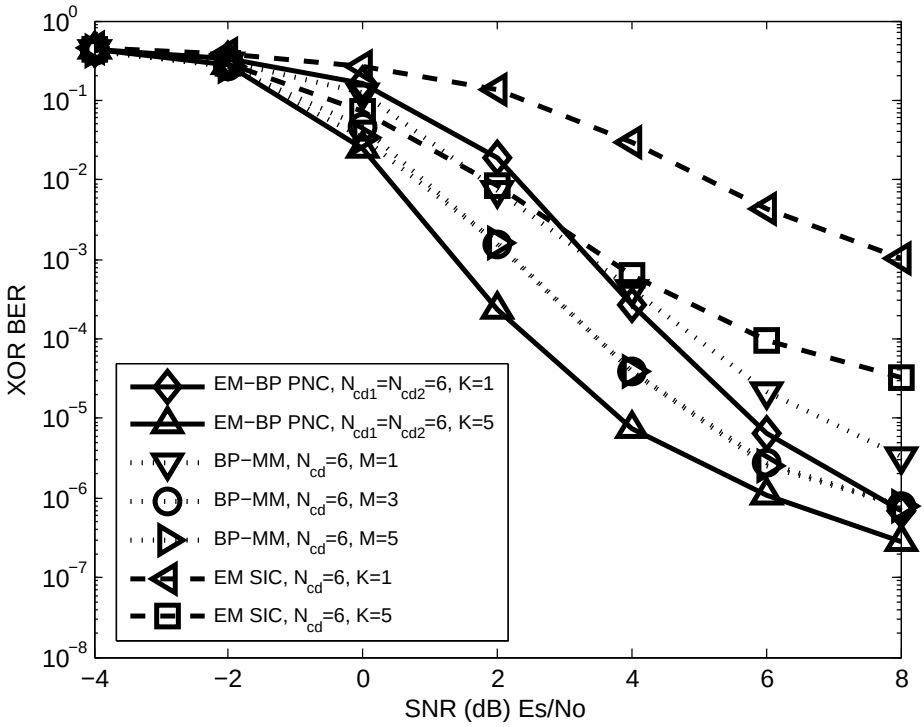


Figure 2.15: Comparing the proposed EM-BP PNC receiver with other receiver architectures.

EM-BP PNC with an EM SIC receiver modified from that in [42]. In the modified EM SIC receiver, there is no despreading operation, since we focus on PNC systems rather than CDMA systems. Specifically, given the channel estimate $\widehat{\mathbf{h}}^{(k-1)}$ from the last EM iteration, we directly employ MMSE SIC to compute the APP of the i^{th} symbol transmitted by a individual node: $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, $p\left(x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, for all i. MMSE SIC [51] is a technique in which the MMSE estimation on the signal of one node is subtracted from the received signal; then channel decoding is performed on the remaining signal using the standard sum-product algorithm to obtain the APP of the other node. Based on these APPs, the

channel estimation is implemented by the EM message passing as proposed in our work here. After the final round of channel decoding, we obtain $\{\widehat{s}_{A,j}\}$, $\{\widehat{s}_{B,j}\}$ from the two single-user channel decoders. We then perform network coding as $\{\widehat{s}_{A,j} \oplus \widehat{s}_{B,j}\}$. Compared with EM-BP PNC, there are two drawbacks to this EM SIC receiver: 1) the APPs $p\left(x_{A,i} \middle| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C\right)$, $p\left(x_{B,i} \middle| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C\right)$, are only approximately computed by MMSE SIC (even without considering the cycles in the factor graph of channel coding) because of the use of the aforementioned MMSE signal cancelation rather than the use of the strict sum-product formalism to link the computations of $p\left(x_{A,i} \middle| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C\right)$ and $p\left(x_{B,i} \middle| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C\right)$ together; and 2) its single-user channel decoding is not optimal for PNC. The BER results of EM SIC are shown in Fig. 2.15. When both receivers have converged ($K = 5$, $N_{cd} = N_{cd1} = N_{cd2} = 18$), we see that EM-BP PNC outperforms EM SIC by around 2dB at BER 10^{-5}.

□ **End of chapter.**

Chapter 3

Frequency-Asynchronous Joint Channel-Parameter Estimation, CFO compensation and Channel Decoding in MUD Systems

3.1 Introduction

THIS chapter investigates a channel-coded MUD system operated with orthogonal frequency-division multiplexing and interleaved division multiple access. While having many advantages, a major problem of OFDM-IDMA is the *frequency asynchrony* caused by the multiple carrier frequency offsets of the signals simultaneously transmitted by the multiple users. We put forth a framework that jointly performs multiuser channel-parameter estimation, CFO compensation and channel decoding that addresses the multiple-CFO problem in a comprehensive and systematic manner.

Why IDMA?

For a multiuser system in which multiple users transmit simultaneously to a common receiver, IDMA is a technique for facilitating the separation of user signals at the receiver [14]. In IDMA, different transmitters interleave their channel-coded symbols in different ways before transmission to create orthogonality among user signals. In that light, it is similar to code-division multiple access (CDMA) except that CDMA creates semi-orthogonality with different spreading codes rather than different interleavers. Thus, it is not surprising that, as with CDMA, IDMA can provide multiuser diversity gains and mitigate inter-cell interference [14, 56]. It has been shown that, all things being equal, IDMA outperforms CDMA in terms of error rate and receiver complexity [57].

Why IDMA with OFDM?

This chapter focuses on wideband communications. The channels in wideband communications are often frequency-selective because of multiple channel paths. The direct application of IDMA to wideband communications using time-domain signals leads to highly complex multiuser detectors [14]. This is because the multiuser detectors must deal with multiple-access interference (induced by multiple user signals) and inter-symbol interference (induced by multiple channel paths) at the same time.

OFDM is a multicarrier modulation technique that can combat frequency selectivity in wideband channels. Specifically, OFDM divides a user data stream into many parallel sub-streams and

transmits them over compactly spaced subcarriers, thereby converting a frequency-selective channel into a group of frequency-flat subchannels [58]. It is desirable to combine IDMA with OFDM, so that the multiuser detectors operating in the frequency domain only have to attend to multiple-access interference rather than both multiple-access interference and inter-symbol interference [59].

A no less important advantage of OFDM for multiuser systems is that it can tolerate unaligned symbol arrival times among the multiple user signals at the receiver. Specifically, as long as the symbol arrival times of different user signals are within the cyclic prefix (CP) of each other, the signal samples in the frequency domain are automatically aligned after the DFT [15].

Major Challenges in OFDM-IDMA

Despite its advantages, the OFDM-IDMA system is susceptible to multiple CFOs among the signals of multiple users. The multiple CFOs are caused by the different RF oscillators used at the different transmitters. CFO causes inter-carrier interference (ICI) among different subcarriers and induces cumulative phase drifts over the data frame [60].

Before the negative effects of CFOs can be alleviated, an issue is the estimation of channel parameters, which include the CFOs as well as the channel gains of different users. For multiuser OFDM-IDMA, the overall estimation errors may increase with the number of users. How to contain the estimation errors in OFDM-IDMA is a major challenge.

With accurate estimates of CFOs, the next issue is to alleviate the negative effects of CFOs. A possibility is to attempt to compensate for them at the receiver. For single-user OFDM systems, the receiver can compensate for the single CFO by multiplying the time-domain signal (before DFT) with the complex exponent of the inverse CFO. This inverse operation cancels out the CFO. After that, the receiver performs standard channel decoding in the frequency domain to extract the source message.

This method of separating CFO compensation and frequency-domain channel decoding, however, does not work for multiuser OFDM systems because of the multiple CFOs. Fundamentally, compensating for these CFOs simultaneously is impossible even if the CFOs were perfectly estimated without errors. This is because removing one of the CFOs in the received signal will necessarily leave behind some residual CFOs for the other CFOs, unless the CFOs of different users were exactly the same to begin with. As a consequence, the CFO-induced ICI inevitably remains in the frequency domain. Since the effects of all CFOs cannot be eliminated in one shot, we need an iterative method for multiuser joint CFO compensation and channel decoding for OFDM-IDMA systems.

Our Approaches and Solutions

We put forth a framework that jointly performs multiuser channel-parameter estimation, CFO compensation and channel decoding in an integrated manner. Our framework combines the space alternating generalized expectation-maximization (SAGE) [26] and expectation-

conditional maximization (ECM) [61] algorithms. The framework first employs the SAGE algorithm to decompose the multisuser problem into multiple single-user problems, and then employs the ECM algorithm to tackle each of the single-user subproblems. Iterative executions of SAGE and ECM in the framework allow the two aforementioned challenges to be tackled in an optimal manner. As far as we know, this is the first attempt to construct such a unified framework for OFDM-IDMA systems. Although SAGE and ECM algorithms are well known, many gaps, however, still need to be filled in order to build a complete and consistent framework with superior performance for multiuser joint channel-parameter estimation, CFO compensation and channel decoding. This chapter fills such gaps, as elaborated below.

After decomposing our multiuser problem into multiple single-user problems, we employ ECM to solve the joint channel-paramter estimation, CFO compensation and channel decoding problem for each user. Here, a key element of our framework is to exploit the sum-product message passing algorithm [17, 62] for channel decoding to refine the channel-parameter estimation as well as the CFO compensation in an iterative manner. The sum-product channel decoding corresponds to treating the coded symbols as the hidden data in the ECM algorithm. We can also adopt anthoer message passing algorithm — the min-sum message passing algorithm [17, 62] — to perform channel decoding. We will explain in Section 3.3.4 that the min-sum channel decoding corresponds to a pure SAGE framework (rather than the SAGE-ECM framework). We will further

show in Section 3.4 of this chapter that the proposed SAGE-ECM framework with the sum-product algorithm has better performance than the SAGE framework with the min-sum algorithm.

To apply the sum-product algorithm in our framework, a subtle issue must be addressed. Specifically, the sum-product algorithm, when applied for channel decoding, yields soft information on the data symbols in the frequency domain. In our iterative framework, this data soft information will in turn be used to refine channel-parameter estimation. But the CFO estimation is only feasible in the time domain (this will be elaborated in Section 3.3.3). Therefore, to use the data for CFO estimation, we need to first transform the data soft information from the frequency domain to the time domain. This problem can be treated as a soft IDFT problem: i.e., performing IDFT on probability functions. As will be elaborated, exact soft IDFT computation can be highly complex (of exponential order). To reduce complexity in soft IDFT, we adopt Gaussian message passing [18] to obtain approximate solutions. We show in Section 3.3.3 that Gaussian message passing reduces the complexity from exponential order to linear order.

In our simulations, we show that our joint framework has around 5—8 dB SNR gain over conventional multiuser approaches for systems with 2—3 users. We further performed real experiments using software-defined radio (SDR) to verify our approach. The experimental and simulated results are consistent with each other.

3.1.1 Related Works

Multiuser decoding in IDMA is a method that jointly performs multiple-access interference cancellation and channel decoding [14, 56]. Ref. [59] proposed the use of OFDM-IDMA for the multiuser communication. However, [59] assumed the absence of CFOs, and it directly applied the multiuser decoding technique originally developed for IDMA [14, 56] to OFDM-IDMA to deal with multiple-access interference in the frequency domain. Without CFOs, the application of IDMA in the frequency domain is essentially the same as that in the time domain without multipath. As mentioned earlier, unavoidable multiple CFOs can cause both inter-carrier interference among subcarriers of the same user and among subcarriers of different users. Our work here investigates this fundamental yet practical issue.

Subsequent to [59], [63] and [64] considered CFOs in OFDM-IDMA. The methods of [63, 64] cancel both inter-carrier interference (induced by CFOs) and multiple-access interference, and iterates between interference cancellation and channel decoding. However, perfect knowledge of CFOs and channel gains were assumed with no consideration given to their estimation. The methods of [63, 64] cancel both inter-carrier interference (induced by CFOs) and multiple-access interference, and iterates between interference cancellation and channel decoding. Our simulation results in Section 3.4 indicate that imperfect channel-parameter estimation using preambles and pilots may cause significant performance penalties (more than 10 dB) to the system. This motivates us to improve the accuracy of channel-

parameter estimation for OFDM-IDMA using not just the preambles and pilots, but also the data symbols in the signals. Doing so requires iterations between the channel-parameter estimation in the time domain and channel decoding in the frequency domain.

An alternative multiuser scheme to OFDM-IDMA is orthogonal frequency-division multiple access (OFDMA) uplink.[1] As in OFDM-IDMA, the multiple CFOs in the OFDMA uplink cannot be compensated for in one shot at the receiver. Thus, in the presence of multiple CFOs, the user signals of OFDMA uplink will overlap in the frequency domain (i.e., these subcarriers are not strictly orthogonal due to the CFOs). The authors of [65, 66] derived multiuser decoding methods for OFDMA to cancel the multiple-user interferences (induced by CFOs) and decode the data symbols in an iterative manner. As with the investigations of OFDM-IDMA in [63, 64], the investigations of OFDMA in [65, 66] did not consider the impact of imperfect CFO and channel estimations.

Recently, [44] proposed a method for multiuser joint channel-parameter estimation, CFO compensation and symbol detection for OFDMA. As with our current work, the method of [44] also aims to

[1] We remark that in the OFDM-IDMA system of interest to us, all the subcarriers are used by all users. This is different from OFDMA, where different users use non-overlapping subcarriers. Multiple-access in OFDM-IDMA is achieved by means of user-specific interleaving in IDMA. OFDM-IDMA has better spectrum efficiency than OFDMA.

combine SAGE and ECM for solving the problem. However, [44] did not consider the impact of channel coding, which necessitates a total recast of the algorithmic framework. In this chapter, we explore the proper way to combine SAGE and ECM for channel-coded OFDM-IDMA. The method of [44] can also be modified for application to channel-coded OFDM-IDMA in a straightforward manner. We will show that a simple extension of the method of [44] to the channel-coded OFDM-IDMA leads to a worse performance than our method.

The rest of this chapter is organized as follows. Section 3.2 describes our system model. Section 3.3 presents our framework and shows how to apply SAGE and ECM to OFDM-IDMA for multiuser joint channel-parameter estimation and channel decoding. Section 3.4 details our the simulation and experimental results.

Notations: We denote matrices by bold capital letters, vectors by bold small letters, and scalars by regular letters throughout this chapter. All vectors are column vectors. The $(i,j)^{th}$ entry of matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{i,j}$. In addition, $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}^{\mathrm{H}}$, $\mathbf{A}^{-1}$ and $\det(\mathbf{A})$ denote the transpose, the conjugate transpose, the inverse and the determinant of $\mathbf{A}$, respectively. $\mathrm{Re}(\cdot)$ means the real part of a complex number, and $\angle(\cdot)$ is the angle of a complex number. $\mathcal{CN}(\mathbf{x}:\mathbf{m},\mathbf{K}) \triangleq \frac{1}{\pi^r \det(\mathbf{K})} \exp\left[-(\mathbf{x}-\mathbf{m})^{\mathrm{H}} \mathbf{K}^{-1}(\mathbf{x}-\mathbf{m})\right]$ denotes the probability density function (PDF) of an r-dimension complex Gaussian random variable $\mathbf{x}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{K}$. The Euclidean norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|$. Finally, $\oplus$ denotes the exclusive-or operation.

3.2 System Model

3.2.1 Transmit Signal

Let us first look at the transmit side of our system model. In the uplink, U users transmit simultaneously to a base station. The transmitted messages employ OFDM signaling with IDMA.

Fig. 3.1 shows the block diagram of the system. User u, $u \in \{1, 2, \cdots, U\}$, generates a sequence of J information bits $\mathbf{b}_u = [b_{u,1} b_{u,2} \cdots b_{u,J}]^T$. The information bits are then channel-coded into J/R code bits $\mathbf{c}_u = \left[c_{u,1} c_{u,2} \cdots c_{u,J/R}\right]^T$, where R is the code rate. We assume all users adopt channel codes that is amenable to decoding by the message passing algorithm [17, 62]. After channel coding, a user-specific interleaver permutes the sequence of code bits $\mathbf{c}_u$ into an interleaved sequence of code bits $\widetilde{\mathbf{c}}_u = \left[\widetilde{c}_{u,1} \widetilde{c}_{u,2} \cdots \widetilde{c}_{u,J/R}\right]^T$. The user-specific interleavers together with the channel encoder serve as the *signatures* of the users. Then, $\widetilde{\mathbf{c}}_u$ is modulated to a sequence of complex data symbols $\mathbf{z}_u = \left[z_{u,1} z_{u,2} \cdots z_{u,J/RB}\right]^T$, where B is the number of code bits per complex data symbol. For simplicity, we focus on BPSK modulation in this chapter. The extension to higher order modulation under the framework of bit-interleaved coded modulation (BICM) [67] is straightforward.

The complex data symbols are transmitted by means of OFDM signaling. OFDM transmits signal on a block-by-block basis. Each OFDM block contains N subcarriers. User u maps each element of $\mathbf{z}_u$ to a subcarrier. We denote the m^{th} OFDM block of user u by a length-N vector $\mathbf{X}_{u,m} = [X_{u,m,1} X_{u,m,2} \cdots X_{u,m,N}]^T$, where the i^{th}

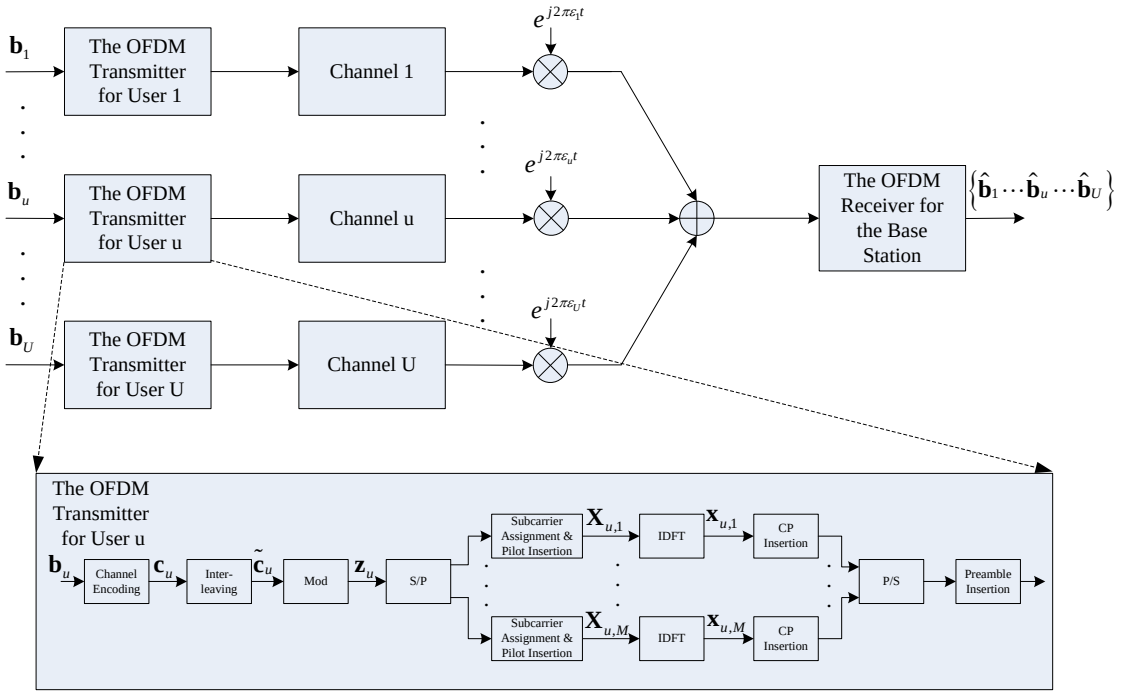


Figure 3.1: Block diagram of OFDM-IDMA.

element $X_{u,m,i}$, $i \in \{1, 2, \cdots, N\}$, is the symbol transmitted over the i^{th} subcarrier in the m^{th} OFDM block of user u.

For channel-parameter estimation, preambles are added to the beginning of the frame, and pilots are carried on selected subcarriers of every OFDM block. In our system, the first $2U$ OFDM blocks at the beginning of the frame are preamble blocks. User u transmits two identical training blocks over preamble block $2u-1$ and $2u$, and nulls its transmission over the other $2U-2$ preamble blocks. Thus, the preambles of different users are mutually orthogonal in the time domain. We stack the OFDM blocks of the overall frame of user u into a length-MN vector $\mathbf{X}_u = \left[\mathbf{X}_{u,1}^T \mathbf{X}_{u,2}^T \cdots \mathbf{X}_{u,M}^T\right]^T$, where M is the number of OFDM blocks contained in one frame — the first $2U$

blocks are preambles and the next $M-2U$ blocks are data payload.

We employ the following arrangement for the data and pilot subcarriers. The data subcarriers allocated to different users overlap completely. The index set of the subcarriers allocated to user data is denoted by I_D. The pilot subcarriers allocated to different users do not overlap. The index set of the subcarriers allocated to user u's pilots is denoted by $I_{P,u}$ with $I_D \cap I_{P,u} = \emptyset$ for $\forall u$ and $I_{P,u} \cap I_{P,v} = \emptyset$ for $u \neq v$. User u maps its complex data symbols $\mathbf{z}_u$ to data subcarriers according to: $X_{u,m,i} = z_{u,l(m,i)}$ for $i \in I_D$, $2U+1 \leq m \leq M$ and $2U+1 \leq m \leq M$, where $l(m,i)$ is the index of the complex data symbol assigned to the i^{th} subcarrier of the m^{th} OFDM block. User u fixes the "known" BPSK symbols on its pilot subcarriers to 1: $X_{u,m,i} = 1$ for $i \in I_{P,u}$, $2U+1 \leq m \leq M$ and $2U+1 \leq m \leq M$. On the other users' subcarriers and the guard-band subcarriers, user u transmits dummy null symbols:$X_{u,m,i} = 0$ for $i \notin I_D \cup I_{P,u}$, $2U+1 \leq m \leq M$ and $2U+1 \leq m \leq M$.

We denote the overall function that includes channel encoding, interleaving, modulation, subcarrier mapping, pilot insertion and preamble addition for user u by C_u, and we write $\mathbf{X}_u = C_u(\mathbf{b}_u)$ to express that the OFDM symbols in $\mathbf{X}_u$ are mapped from the information bits $\mathbf{b}_u$.

The OFDM modulation is implemented by an N point IDFT $\mathbf{x}_{u,m} = \mathbf{F}^H \mathbf{X}_{u,m}$, where $\mathbf{x}_{u,m} = [x_{u,m,1} x_{u,m,2} \cdots x_{u,m,N}]^T$ is the vector of time-domain samples, and $\mathbf{F}$ is the $N \times N$ DFT matrix whose $(p,q)^{th}$ entry is given by $\mathrm{e}^{-j2\pi(p-1)(q-(1+N/2))/N} \Big/ \sqrt{N}$, $1 \leq p,q \leq N$, $1 \leq p,q \leq N$. We stack all the time-domain sample vectors of the

whole frame into a length-MN vector $\mathbf{x}_u = \left[\mathbf{x}_{u,1}^T \mathbf{x}_{u,2}^T \cdots \mathbf{x}_{u,M}^T\right]^T$.

To overcome the delay spread of multipath channels, which causes inter-block interferences, the time-domain samples of each OFDM symbol is preceded by a cyclic prefix (CP). We denote the length of the CP by N_{cp}. Therefore, each OFDM block includes $N_s = N_{cp} + N$ time-domain samples $\{x_{u,m,i}\}_{i=-N_{cp}+1}^{N}$, where $x_{u,m,i} = x_{u,m,i+N}$ for $i \in \{-N_{cp}+1, \cdots, 0\}$. The time-domain samples are converted to signal waveform via a digital-to-analog converter (DAC). The U users simultaneously transmit their signal waveforms on their respective multipath channels. The base station receives their overlapped signal waveforms.

3.2.2 Channel Model

We denote the overall discrete time-domain channel impulse response of user u that captures the effects of both the physical channel and the transmit/receive filters by $\widetilde{\mathbf{h}}_u = [h_{u,1} h_{u,2} \cdots h_{u,L_u}]^T$, where $h_{u,l}$ is the l^{th} discrete tap of the multipath channel of user u, $l \in \{1, \cdots, L_u\}$, and L_u is the maximum channel delay spread. We assume that the channels remain static over the transmission time of one frame. This assumption is valid in scenarios where users exhibit low mobility [68]. We model the channel taps as mutually independent complex Gaussian variables with zero mean and an exponentially decaying power delay profile: $\mathrm{E}\left\{|h_{u,l}|^2\right\} = \beta \exp\left(-(l-1)/L_u\right)$, where β is a normalization factor to ensure that the average channel energy is one.

3.2.3 Receive Signal

The duration of a time-domain sample is denoted by T_s. The the timing mismatch between user u and the base station is denoted by τ_u. For convenience, we define the *zero reference time* as the first-channel-path arrival time of user 1 ($\tau_1 = 0$), and assume without loss of generality that the first-channel-path arrival times of all other users are later than that of user 1 ($\tau_u > 0$, $u \in \{2, \cdots, U\}$). Following [44, 69], we decompose τ_u into an integer part plus a fractional part with respect to the sample duration T_s: $\tau_u = (\mu_u + \delta_u) T_s$, where $\mu_u = \lfloor \tau_u / T_s \rfloor$ is the integer part and $\delta_u = \tau_u / T_s - \mu_u$ is the fractional part. The fractional part can be incorporated into the channel impulse response, thus we will not consider it going forward. We can avoid IBI in the system by assuming that the system satisfies a *loose time synchronization requirement*, specified as $\max_u \{\mu_u + L_u\} \leq N_{cp}$. As long as this requirement is satisfied, symbol misalignment in the time domain does not affect the subcarrier-by-subcarrier channel decoding in the frequency domain other than introducing relative phase offsets between the signals of the users — in particular, there is no symbol misalignment in the frequency domain and signals on different subcarriers are isolated from one another. Note that there is no need to explicitly estimate the timing parameters $\{\tau_u\}_{u=2}^{U}$. To meet the loose time synchronization requirement, the base station can broadcast a downlink beacon to prompt the users to transmit together [15, 44, 69, 70].

The presence of CFOs destroys the perfect orthogonalities among

subcarriers. A main focus of this chapter is the study of the effect of CFOs (i.e., the frequency asynchrony problem). After analog-to-digital conversion and removal of the first N_{cp} CP samples (counting from the first sample of the first channel path of user 1) of every OFDM block, the received discrete time-domain samples at the base station can be expressed as

$$\mathbf{r}_m = \sum_{u=1}^{U} e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{F}\mathbf{h}_u + \mathbf{n}_m \tag{3.1}$$

for $m = 1, 2, \cdots, M$, where

- $\mathbf{r}_m = \left[r_{m,1} r_{m,2} \cdots r_{m,N}\right]^T$ is the vector of the N discrete samples of the m^{th} received OFDM block; $\mathbf{n}_m = \left[n_{m,1} n_{m,2} \cdots n_{m,N}\right]^T$ is the vector of the complex white Gaussian noises with zero-mean and variance of σ_n^2;
- ε_u is the CFO (normalized to subcarrier space) between the base station and user u; $\mathbf{\Gamma}\left(\varepsilon_u\right) = diag\left\{\left[1, e^{j2\pi\varepsilon_u/N}, \cdots, e^{j2\pi\varepsilon_u(N-1)/N}\right]\right\}$ is the diagonal matrix that captures the time-domain effect of CFO on the m^{th} block of user u;
- $\theta_{u,m} = 2\pi\varepsilon_u(N_{cp} + mN_s)/N$ is the accumulated phase drift caused by CFO ε_u on the m^{th} block of user u;
- $\mathbf{D}\left(\mathbf{X}_{u,m}\right) = diag\left(\mathbf{X}_{u,m}\right)$ is the diagonal matrix with transmitted frequency-domain symbols $\mathbf{X}_{u,m}$ as its diagonal elements;
- $\mathbf{h}_u \triangleq \left[\mathbf{0}_{\mu_u}^T, \widetilde{\mathbf{h}}_u^T, \mathbf{0}_{N-L_u-\mu_u}^T\right]^T$ is the length-N vector that captures both the discrete CIR and the time asynchrony of node u.

For OFDM systems, channel decoding is performed in the frequency domain. The frequency-domain sample vector of the m^{th} OFDM symbol, $\mathbf{R}_m = [R_{m,1}R_{m,2}\cdots R_{m,N}]^T = \mathbf{F}\mathbf{r}_m$, is given by

$$\begin{aligned}\mathbf{R}_m &= \sum_{u=1}^{U} e^{j\theta_{u,m}} \underbrace{\mathbf{F}\boldsymbol{\Gamma}\left(\varepsilon_u\right)\mathbf{F}^H}_{\triangleq\boldsymbol{\Xi}(\varepsilon_u)} \mathbf{D}\left(\mathbf{X}_{u,m}\right) \underbrace{\mathbf{F}\mathbf{h}_u}_{\triangleq\mathbf{H}_u} + \underbrace{\mathbf{F}\mathbf{n}_m}_{\triangleq\mathbf{N}_m} \\ &= \sum_{u=1}^{U} e^{j\theta_{u,m}} \boldsymbol{\Xi}\left(\varepsilon_u\right)\mathbf{D}\left(\mathbf{X}_{u,m}\right)\mathbf{H}_u + \mathbf{N}_m\end{aligned} \tag{3.2}$$

where $\mathbf{H}_u \triangleq \mathbf{F}\mathbf{h}_u$ is the vector of the frequency channel responses for user u, $\mathbf{N}_m \triangleq \mathbf{F}\mathbf{n}_m$ is the vector of frequency-domain noises, $\boldsymbol{\Xi}\left(\varepsilon_u\right) \triangleq \mathbf{F}\boldsymbol{\Gamma}\left(\varepsilon_u\right)\mathbf{F}^H$ is the matrix that transforms the effect of CFO ε_u in the time domain to ICI in the frequency domain. It can be shown that $\boldsymbol{\Xi}\left(\varepsilon_u = 0\right) = \mathbf{I}$; $\boldsymbol{\Xi}\left(\varepsilon_u \neq 0\right) = \lambda\mathbf{I} + \boldsymbol{\Pi}\left(\varepsilon_u \neq 0\right)$, where λ is a common attenuation factor across all subcarriers, and $\boldsymbol{\Pi}\left(\varepsilon_u \neq 0\right)$ is the ICI component [71].

We stack the M length-N time-domain (frequency-domain) sample vectors into an overall length-MN sample vector $\mathbf{r} \triangleq \left[\mathbf{r}_1^T\mathbf{r}_2^T\cdots\mathbf{r}_M^T\right]^T$ ($\mathbf{R} \triangleq \left[\mathbf{R}_1^T\mathbf{R}_2^T\cdots\mathbf{R}_M^T\right]^T$). The receiver then undertakes two tasks: (i) estimation of channel parameters $\{\varepsilon_u, \mathbf{h}_u\}_{u=1}^U$; (ii) decoding of multiuser information bit sequences $\{\mathbf{b}_u\}_{u=1}^U$.

With reference to the signal model in (3.2), we can observe that CFO causes two negative effects on the frequency-domain signal: (i) the drifting of the signal phase over time; (ii) the inter-carrier interferences (ICI) among different subcarriers.

For conventional point-to-point OFDM systems, the receiver can first estimate the CFO and the channel gain from the preamble and

then compensate for the CFO for the whole frame in the time domain. That is, it attempts to remove the CFO in the signal using the estimated CFO. After that, it performs standard channel decoding in the frequency domain. Estimation error may leave behind an uncompensated residual CFO. If the estimation is accurate enough, the residual CFO will be small, and the remaining CFO-induced ICI will also be small. We can treat the residual ICI as additional noise that lowers the effective signal-to-noise ratio (SNR) slightly [60], but standard channel encoding/decoding can still be employed to ensure communication reliability [72]. This CFO compensation alone, however, cannot overcome the drifting phase. Even a small residual CFO can lead to large phase drifts accumulated over time. Standard channel decoding will fail if we ignore the phase drifts. Therefore, point-to-point OFDM systems typically employ pilot subcarriers in the OFDM block to track signal phase so that it can be corrected. [68].

The above design principle, however, does not work for the multiuser OFDM-IDMA system. In particular, there are multiple CFOs in OFDM-IDMA, one for the signal of each user. First of all, accurately estimating these CFOs and the channel gains is demanding. More fundamentally, compensating for these CFOs simultaneously in one shot is impossible even if the CFOs could be estimated without errors (by contrast, for conventional point-to-point systems, total CFO removal is possible given perfect CFO estimation). This is because removing one of the CFOs in the received signal will necessarily leave behind some remaining CFOs for the other CFOs, unless the CFOs of different users were exactly the same to begin

with. As a consequence, the CFO-induced ICI inevitably remains in the frequency domain. Instead of channel-parameter estimation, followed by CFO compensation, followed by channel decoding, that are typical in many single-user systems, an iterative method is called for in a multiuser system such as OFDM-IDMA. In what follows, we propose a SAGE-and-ECM framework to address the joint problem of multiuser channel-parameter estimation, CFO compensation and channel decoding.

3.3 Multiuser Joint Channel-Parameter Estimation, CFO Compensation and Channel Decoding

To improve the overall performance of the OFDM-IDMA system, we make use of the preambles, the pilots as well as the data payload of the received signal to jointly estimate channel parameters and decode information bits.

For OFDMA receivers, [44] developed an iterative solution that uses SAGE and ECM. However, [44] did not consider the impact of channel coding, the good performance of which necessitates a total recast of the algorithmic framework. In this chapter, we explore the proper algorithmic framework of SAGE and ECM for channel-coded OFDM-IDMA systems. We argue that incorporating channel decoding by simply extending the framework of [44] leads to subpar performance.

We formulate our target problem as a multidimensional estimation problem in Section 3.3.1. Section 3.3.2 gives an overview of a SAGE-based signal decomposition method. As in [44], we apply SAGE and break down the overall problem into U sub-problems, one for each user, by decomposing the received signal into U signal components. Readers who are familiar with SAGE decomposition can quickly go through Section 3.3.2 to familiarize themselves with our notations. Section 3.3.3 zooms in to the ECM-based joint channel-parameter estimation, CFO compensation and channel decoding problem within the U sub-problems. The main contribution of our work is contained in Section 3.3.3. In particular, a key issue is how to incorporate message passing algorithm for channel decoding into the joint framework. In that regard, two message passing strategies are possible for channel decoding: sum-product and min-sum message passing [17, 62]. In our framework, the ECM algorithm gives the sum-product channel decoding. We show that the integration of sum-product channel decoding with channel-parameter estimation leads to better system performance than min-sum channel decoding. However, for compatibility with the channel-parameter estimation part, the sum-product channel decoding cannot be applied directly: a transformation of the soft information from the time domain to the frequency domain is needed. We develop a new technique called "soft IDFT" to realize an overall compatible algorithmic framework. Exact computation of soft IDFT, however, is complex. Here, we obtain an effective approximate solution by Gaussian message passing.

3.3.1 Problem Statement

As expressed by the signal model in (3.1), the only unknown channel parameters are the CFOs $\{\varepsilon_u\}_{u=1}^{U}$ and the channel gains $\{\mathbf{h}_u\}_{u=1}^{U}$. In (3.1), the phase drift $\theta_{u,m} = 2\pi\varepsilon_u(N_{cp} + mN_s)/N$ is not independent and is a function of CFO ε_u. However, if we only estimate ε_u and derive the estimated phase drift $\theta_{u,m}$ based on the estimated ε_u, even a tiny estimation error in ε_u will accumulate to large estimation errors in $\theta_{u,m}$ for later blocks m. Furthermore, the expression $\theta_{u,m} = 2\pi\varepsilon_u(N_{cp} + mN_s)/N$ assumes that the signals of different users do not incur different phase noises and the phase $\theta_{u,m}$ is strictly due to that of the CFO. In practice, the phase drift is not due to CFO alone but also due to phase noise that behaves like a random walk [73]. Thus, in this chapter, for a robust system, we do not make use of the expression $\theta_{u,m} = 2\pi\varepsilon_u(N_{cp} + mN_s)/N$ when estimating $\theta_{u,m}$: we assume that $\{\theta_{u,m}\}$ are independent for different OFDM blocks and different users, and estimate $\theta_{u,m}$ of each block independently. The advantage of this scheme is that phase errors due to estimated CFO errors are not cumulative over blocks and random phase noise can be taken into account.

With the above, the overall unknown variables in the system are $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}_{u=1}^{U}$.

According to the maximum likelihood (ML) principle, we can express the objective of the joint channel-parameter estimation and

channel decoding problem as

$$\begin{aligned}
&\left(\left\{\widehat{\varepsilon}_u, \left\{\widehat{\theta}_{u,m}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u, \widehat{\mathbf{X}}_u\right\}_{u=1}^{U}\right) \\
&= \arg \max_{\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}\in C_u\right\}_{u=1}^{U}} \left\{ \right. \\
&\qquad \left. \log p\left(\mathbf{r} \left| \left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}_{u=1}^{U}\right.\right)\right\} \qquad (3.3) \\
&= \arg \min_{\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}\in C_u\right\}_{u=1}^{U}} \left\{ \right. \\
&\qquad \left. \sum_{m=1}^{M} \left\| \mathbf{r}_m - \sum_{u=1}^{U} e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{F}\mathbf{h}_u \right\|^2 \right\}
\end{aligned}$$

With regard to (3.3), we emphasize that since the sequence of transmit symbols is generated from the sequence of the original information bits via a one-to-one mapping, i.e., $\{\mathbf{X}_u = C_u(\mathbf{b}_u)\}_{u=1}^{U}$, decoding the transmit symbols $\{\mathbf{X}_u\}_{u=1}^{U}$ is equivalent to decoding the information bits $\{\mathbf{b}_u\}_{u=1}^{U}$. Directly solving the ML problem (3.3) is intractable because the exhaustive search over the multi-dimensional space of $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}_{u=1}^{U}$ is prohibitively complex.

3.3.2 Preliminary for the Signal Decomposition Using SAGE

In [74], the authors solved the problem of multiple parameter estimation (but not channel decoding) using a iterative method. Later on, the method of [74] evolved into the SAGE algorithm [26]. The key idea of [74] is to decompose the received overlapping signal into several signal components. Subsequently, [44] applied the method of [74] to

OFDMA systems. Since the signal decomposition is performed in the time domain, where the user signals overlap completely (for both OFDM-IDMA and OFDMA), we can directly apply this SAGE-based signal decomposition to our OFDM-IDMA system. This subsection extends the SAGE signal decomposition in [74] to incorporate channel decoding.

First, the overall estimation problem (3.3) is decomposed into U sub-problems [44,74]. For the m^{th} OFDM block, the signal component of user u is defined to be

$$\mathbf{r}_{u,m} \triangleq e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{F} \mathbf{h}_u + \mathbf{n}_{u,m} \tag{3.4}$$

where $\{\mathbf{n}_{u,m}\}_{u=1}^{U}$ are obtained by arbitrarily decomposing the total noise vector $\mathbf{n}_m$ into U circularly symmetric and statistically independent noise component vectors that satisfy $\mathbf{n}_m = \sum_{u=1}^{U} \mathbf{n}_{u,m}$ [74]. The received signal can then be written as

$$\mathbf{r}_m = \sum\nolimits_{u=1}^{U} \mathbf{r}_{u,m} \tag{3.5}$$

where $\mathbf{r}_m$ is decomposed into U components $\{\mathbf{r}_{u,m}\}_{u=1}^{U}$, with each being exclusively related to one user. As with the stacking of the overall received signals $\mathbf{r} = \left[\mathbf{r}_1^T \mathbf{r}_2^T \cdots \mathbf{r}_M^T\right]^T$ in the whole frame, we can also stack the M signal component vectors of user u into an overall vector $\mathbf{r}_u \triangleq \left[\mathbf{r}_{u,1}^T \mathbf{r}_{u,2}^T \cdots \mathbf{r}_{u,M}^T\right]^T$, $u \in \{1, 2, \cdots, U\}$. In the terminology of SAGE, $\mathbf{r}$ is the observed data and $\{\mathbf{r}_u\}_{u=1}^{U}$ is the complete data.

SAGE tries to find the ML estimates for $\{\varepsilon_u,$

$\{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\Big\}_{u=1}^{U}$ iteratively.[2] Let $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, \widehat{\mathbf{X}}_u^{(k)}\right\}_{u=1}^{U}$ be the updated estimates after the k^{th} SAGE iteration, where $k = 1, 2, \cdots K$ and $\left\{\widehat{\varepsilon}_u^{(0)}, \left\{\widehat{\theta}_{u,m}^{(0)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(0)}, \widehat{\mathbf{X}}_u^{(0)}\right\}_{u=1}^{U}$ are the initial estimates. The initial estimates for the CFOs and the channel gains are obtained from the orthogonal preambles; the initial estimates for the phase drifts are set to zeros; the initial estimates for the transmit symbols are set to ± 1 randomly. Each SAGE iteration consists of U cycles; the variables of user u are updated in the u^{th} cycle given that the variables of all other users are fixed to their last estimates.

With the initial estimates $\left\{\widehat{\varepsilon}_u^{(0)}, \left\{\widehat{\theta}_{u,m}^{(0)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(0)}, \widehat{\mathbf{X}}_u^{(0)}\right\}_{u=1}^{U}$, the SAGE algorithm first computes the initial estimates for the individual user signal components:

$$\widehat{\mathbf{r}}_{u,m}^{(0)} = e^{j\widehat{\theta}_{u,m}^{(0)}} \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(0)}\right) \mathbf{F}^H \mathbf{D}\left(\widehat{\mathbf{X}}_u^{(0)}\right) \mathbf{F}\widehat{\mathbf{h}}_u^{(0)} \tag{3.6}$$

where $m = 1, 2, \cdots, M$ and $u \in \{1, 2, \cdots, U\}$. The u^{th} cycle of the k^{th} SAGE iteration includes an E-step and an M-step [74] as follows:

E-step of SAGE:

[2]This means that the setup of SAGE here treats all variables $\{\varepsilon_u,\ \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\Big\}_{u=1}^{U}$, including $\{\mathbf{X}_u\}_{u=1}^{U}$ as the parameters; there is no hidden data in the setup [26, 74].

Compute the tentative estimate for the signal component of user u:

$$\widehat{\mathbf{r}}_{u,m}^{(k)} = \mathbf{r}_m - \sum_{v=1}^{u-1} \widehat{\mathbf{r}}_{v,m}^{(k)} - \sum_{v=u+1}^{U} \widehat{\mathbf{r}}_{v,m}^{(k-1)} \tag{3.7}$$

where $m = 1, 2, \cdots, M$, and $\sum_v^u = 0$ if $u < v$. Note that this computation is based on the signal decomposition in (3.5). We stack the estimates for all the OFDM blocks into an overall vector $\widehat{\mathbf{r}}_u^{(k)} \triangleq \left[\widehat{\mathbf{r}}_{u,1}^{(k)T} \widehat{\mathbf{r}}_{u,2}^{(k)T} \cdots \widehat{\mathbf{r}}_{u,M}^{(k)T}\right]^T$.

M-step of SAGE:

Update the estimates for the variables of user u:

$$\begin{aligned}
&\left(\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, \widehat{\mathbf{X}}_u^{(k)}\right) \\
&= \arg \max_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u \in C_u\right)} \Bigg\{ \\
&\qquad \log p\left(\widehat{\mathbf{r}}_u^{(k)} \,\middle|\, \varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right) \Bigg\} \qquad (3.8) \\
&= \arg \min_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u \in C_u\right)} \Bigg\{ \\
&\qquad \sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{m,u}^{(k)} - e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{F} \mathbf{h}_u \right\|^2 \Bigg\}
\end{aligned}$$

After the M-step, we then reconstruct the estimate for the signal component of user u using the updated variable estimates

$$\widehat{\mathbf{r}}_{u,m}^{(0)} = e^{j\widehat{\theta}_{u,m}^{(k)}} \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k)}\right) \mathbf{F}^H \mathbf{D}\left(\widehat{\mathbf{X}}_u^{(k)}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k)} \tag{3.9}$$

for $m = 1, 2, \cdots, M$. This completes the u^{th} cycle of the k^{th} SAGE iteration; we then proceed to the cycle of the next user.

After the variables of all users are updated to $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, \widehat{\mathbf{X}}_u^{(k)}\right\}_{u=1}^{U}$, we then proceed to the next SAGE iteration. When the number of iteration k reaches a preset maximum limit K, we terminate the SAGE algorithm after obtaining the final variable estimates $\left\{\widehat{\varepsilon}_u^{(K)}, \left\{\widehat{\theta}_{u,m}^{(K)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(K)}, \widehat{\mathbf{X}}_u^{(K)}\right\}_{u=1}^{U}$. According to the theory of SAGE [26], it is expected that the final variable estimates will converge to the global optimal as required by the ML estimation in (3.3).

As seen above, SAGE decomposes the multiuser problem of joint channel-parameter estimation and channel decoding in (3.3) into U single-user problems in (3.8). The complexity of the multiuser problem is reduced substantially. However, the computation involved in the single-user sub-problems as expressed in (3.8) is still non-trivial. For each user, we need to solve a multi-dimensional problem associated with simultaneously estimating $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}$. This is of high complexity, particularly if the data symbols $\{\mathbf{X}_u\}_{u=1}^{U}$ are channel-coded symbols and we want to exploit the correlations among the symbols induced by channel coding to optimize our estimation.

A simplified approach to solve (3.8) is to estimate the variables in $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}$ one at a time sequentially and iteratively. When one variable is under estimation, all other variables are fixed to their estimates from the last iteration. This approach is straightforward with an important caveat: it does not use the information obtained from channel decoding in an optimal way. We

will further elaborate on this simplified approach in Section 3.3.4 and will treat it as a benchmark for evaluating our approach to be presented in Section 3.3.3. To make better use of the information from channel decoding, Section 3.3.3 will construct a more comprehensive approach to our problem using the ECM algorithm.

3.3.3 Joint Channel-Parameter Estimation and Channel Decoding Using ECM

ECM is a variant of the expectation maximization (EM) algorithm, a general iterative algorithm for finding the ML estimates of parameters in a statistical model with hidden data [24]. EM updates all parameters in the model simultaneously at each iteration. This requires EM to operate in a multi-dimensional space. To reduce complexity, ECM updates the parameters sequentially. At each stage of the update sequence, ECM updates just one parameter, fixing the other parameters to their last estimates [61].

Within the framework of ECM, we could assign the role of hidden data and the role of parameters to the variables $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}$ in different ways, each yielding a different implementation. In that sense, it is more general than (3.8), in which all variables are treated as parameters.

In this chapter and the rest of this section, we focus on the assignment that treats $\mathbf{X}_u$ as the hidden data, and $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u\right\}$

as the parameters.[3] With this assignment, we can incorporate sum-product channel decoding into the joint framework. Now, ECM seeks to solve the following maximization problem:

$$\begin{aligned}
&\left(\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}\right) \\
&= \arg \max_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u\right)} \log p\left(\widehat{\mathbf{r}}_u^{(k)} \left| \varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u \right.\right) \\
&= \arg \max_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u\right)} \\
&\qquad \left\{ \log \sum_{\mathbf{X}_u} p\left(\widehat{\mathbf{r}}_u^{(k)}, \mathbf{X}_u \left| \varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, C_u \right.\right) \right\}
\end{aligned} \tag{3.10}$$

Note that problem (3.10) is different from problem (3.8) , where all variables $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}$ (including $\mathbf{X}_u$) are treated as parameters to be estimated.

After ECM finds $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}\right\}$ in (3.10), we then set $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u\right\} = \left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}\right\}$ in the signal model (3.2) and then perform sum-product channel decoding to find the a posteriori probabilities (APPs) of the coded symbol $p\left(X_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, C_u \right.\right)$ for all m and i. For example, if the convolution code is used, the corresponding channel decoding algorithm is the BCJR algorithm [75]. The estimates for the

[3]The assignment that treats $\mathbf{h}_u$ as the hidden data and $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u\right\}$ as the parameters is discussed in Section 3.3.5.

symbols in $\mathbf{X}_u$ are obtained by making hard decisions based on the symbol-wise APPs:

$$\begin{aligned}&\hat{X}_{u,m,i}^{(k)}\\&= \arg\max_{X_{u,m,i}} p\left(X_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \hat{\varepsilon}_u^{(k)}, \left\{\hat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, C_u\right.\right)\end{aligned}$$

We stack all the symbol estimates $\widehat{X}_{u,m,i}^{(k)}$ for all m and i into a vector $\widehat{\mathbf{X}}_u^{(k)}$, and $\widehat{\mathbf{X}}_u^{(k)}$ is treated as the estimate for $\mathbf{X}_u$.

Finally, the parameter estimates $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}\right\}$ by ECM together with the hidden data estimate $\widehat{\mathbf{X}}_u^{(k)}$ by channel decoding are treated as the overall solution for $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, \widehat{\mathbf{X}}_u^{(k)}\right\}$. We will see that the ECM and channel decoding can be integrated into one framework to assist each other for finding $\left\{\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}\right\}$ and $\widehat{\mathbf{X}}_u^{(k)}$. Here, we emphasize that $\widehat{\mathbf{X}}_u^{(k)}$ are the decoded soft information (APPs). We will show in the following how to use the APPs to refine the estimates of the parameters through the ECM iterations.

Within the k^{th} iteration of SAGE, the ECM algorithm for solving each sub-problem in (3.8) consists of Z iterations. We collect the parameters into a set $\Omega_u \triangleq \left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u\right\}$, and denote the updated parameter estimates after the z^{th} ECM iteration within the k^{th} SAGE iteration by $\widehat{\Omega}_u^{(k,z)} = \left\{\widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z)}\right\}$, $z = 1, 2, \cdots, Z$. The z^{th} iteration of ECM within the k^{th} SAGE iteration consists of an E-step and an M-step, as follows:

E-step of ECM:

Given the parameter estimates from the previous ECM iteration $\widehat{\Omega}_u^{(k,z-1)}$, the E-step of ECM aims to compute the conditional expectation defined by [61].

$$\begin{aligned} & Q\left(\Omega_u \left| \hat{\Omega}_u^{(k,z-1)} \right.\right) \\ & = \sum_{\mathbf{X}_u} p\left(\mathbf{X}_u \left| \widehat{\mathbf{r}}_u^{(k)}, \hat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \log p\left(\widehat{\mathbf{r}}_u^{(k)} \left| \Omega_u, \mathbf{X}_u \right.\right) \end{aligned} \tag{3.11}$$

This conditional expectation is called the Q function in the EM literature. We remark that the decoded soft information of the transmitted symbols, if obtainable, is used to assist the estimate of the parameters. With this setup, the data symbols serve as the hidden data in the ECM framework. Since the expectation (the summation) in (3.11) is taken over the data symbols in the frequency domain, we might be tempted compute the above Q function in the frequency domain. However, we will see that this does not work. With a new approach, we can compute the Q function in the time domain.

How to decompose Q function in time domain

A brute-force attack that attempts to compute the Q function in (3.11) directly will meet the following obstacles: (i) the ensemble of codewords $\mathbf{X}_u$ to be summed over in (3.11) is very large (in fact, exponentially large in terms of the number of symbols in $\mathbf{X}_u$); (ii) barring exhaustive enumeration, there is no known decoding algorithm that can give $p\left(\mathbf{X}_u \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right)$ for all these codewords; (iii) there is no concise expression for $p\left(\widehat{\mathbf{r}}_u^{(k)} \left| \Omega_u, \mathbf{X}_u \right.\right)$ due to the coupled terms brought about by the ICI effect. Ref. [44] did not take these obstacles into account, since it investigated uncoded

systems. In order to obtain a practical solution for (3.11), in this chapter, we transform the computation process to the time domain, where the Q function can be decomposed into many sample-wise factors.

We define the frequency-domain compound symbol of the subcarrier symbol and the channel gain as

$$Y_{u,m,i} \triangleq X_{u,m,i} H_{u,i}$$

Given the channel gain $H_{u,i}$, $Y_{u,m,i} \in \{\pm H_{u,i}\}$ when BPSK is adopted. The vector of frequency-domain compound symbols for the m^{th} OFDM block is

$$\mathbf{Y}_{u,m} \triangleq [Y_{u,m,1} Y_{u,m,2} \cdots Y_{u,m,N}]^T = diag\{\mathbf{X}_{u,m}\} \mathbf{H}_u;$$

the corresponding vector of time-domain compound samples is $\mathbf{y}_{u,m} \triangleq [y_{u,m,1} y_{u,m,2} \cdots y_{u,m,N}]^T = \mathbf{F}^H \mathbf{Y}_{u,m}$; and the vector of frequency-domain compound symbols (time-domain compound samples) for the whole frame is $\mathbf{Y}_u \triangleq \left[\mathbf{Y}_{u,1}^T \mathbf{Y}_{u,2}^T \cdots \mathbf{Y}_{u,M}^T\right]^T$ ($\mathbf{y}_u \triangleq \left[\mathbf{y}_{u,1}^T \mathbf{y}_{u,2}^T \cdots \mathbf{y}_{u,M}^T\right]^T$). Bear in mind that given the channel gains $\mathbf{H}_u$, the knowledge of the compound symbols $\mathbf{Y}_u$ is equivalent to the knowledge of the transmit symbols $\mathbf{X}_u$. We can therefore rewrite the Q function in (3.11) as

$$\begin{aligned} & Q\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)} \right.\right) \\ &= \sum_{\mathbf{Y}_u} \left\{ p\left(\mathbf{Y}_u \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \log p\left(\widehat{\mathbf{r}}_u^{(k)} | \Omega_u, \mathbf{Y}_u\right) \right\} \\ &= \sum_{\mathbf{y}_u} \left\{ p\left(\mathbf{y}_u \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \log p\left(\widehat{\mathbf{r}}_u^{(k)} | \Omega_u, \mathbf{y}_u\right) \right\} \end{aligned} \tag{3.12}$$

where the second equality is due to that the mapping between $\mathbf{Y}_u$ and $\mathbf{y}_u$ is a one-to-one correspondence.

Next, we decompose the probability functions in (3.12) to compute the time-domain Q function. We denote the i^{th} element of $\widehat{\mathbf{r}}_{u,m}^{(k)}$ (the estimate for the i^{th} sample of the m^{th} OFDM block of user u) by $\widehat{r}_{u,m,i}^{(k)}$. Since CFO only introduces a linear phase drift in the time-domain signals (and not ICI), we can immediately decouple the components $\widehat{r}_{u,m,i}^{(k)}$ in $\widehat{\mathbf{r}}_u^{(k)}$ and decompose $p\left(\widehat{\mathbf{r}}_u^{(k)}\,|\Omega_u, \mathbf{y}_u\right)$ as

$$p\left(\widehat{\mathbf{r}}_u^{(k)}\,|\Omega_u, \mathbf{y}_u\right) = \prod_{m=1}^{M}\prod_{i=1}^{N} p\left(\widehat{r}_{u,m,i}^{(k)}\,|\Omega_u, y_{u,m,i}\right) \tag{3.13}$$

where the sample-wise factor is given by

$$\begin{aligned} &p\left(\widehat{r}_{u,m,i}^{(k)}\,|\Omega_u, y_{u,m,i}\right) \\ &= \mathcal{CN}\left(\widehat{r}_{u,m,i}^{(k)} : e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} y_{u,m,i}, \sigma_n^2\right) \end{aligned} \tag{3.14}$$

for all m and i. Substituting (3.13) into (3.12), we obtain

$$\begin{aligned} Q\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)}\right.\right) = &\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{y_{u,m,i}} \log p\left(\widehat{r}_{u,m,i}^{(k)}\,|\Omega_u, y_{u,m,i}\right) \\ &\times \underbrace{\sum_{\{\mathbf{y}_u :\sim y_{u,m,i}\}} p\left(\mathbf{y}_u \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right)}_{\triangleq p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right)} \end{aligned} \tag{3.15}$$

where $\{\mathbf{y}_u :\sim y_{u,m,i}\}$ is the set that contains the all elements in vector $\mathbf{y}_u$ except $y_{u,m,i}$, and $p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right)$ is the sample-wise APP. We define the sample-wise Q function as

$$\begin{aligned} &Q_{m,i}\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)}\right.\right) \triangleq \\ &\sum_{y_{u,m,i}} \log p\left(\widehat{r}_{u,m,i}^{(k)}\,|\Omega_u, y_{u,m,i}\right) p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right) \end{aligned} \tag{3.16}$$

Finally, the overall Q function is the sum of sample-wise Q functions:

$$Q\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)}\right.\right) = \sum_{m=1}^{M}\sum_{i=1}^{N} Q_{m,i}\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)}\right.\right) \tag{3.17}$$

As can be seen from the above, unlike the frequency-domain Q function, the time-domain Q function can be computed on a sample-by-sample basis. This greatly reduces the computational complexity.

How to transform APPs from frequency domain to time domain

The next question is how to obtain the APPs of the time-domain samples $\left\{p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)\right\}$ in (3.16). Now, the symbols are channel-coded and transmitted in the frequency domain; after DFT the receiver performs channel decoding to find the APPs of the transmit symbols $\left\{p\left(X_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)\right\}$. Given the estimated channel gains $\widehat{\mathbf{H}}_u^{(k,z-1)} \triangleq \left[\widehat{H}_{u,1}^{(k,z-1)}\widehat{H}_{u,2}^{(k,z-1)}\cdots\widehat{H}_{u,N}^{(k,z-1)}\right]^T = \mathbf{F}\widehat{\mathbf{h}}_u^{(k,z-1)}$, this is equivalent to finding the APPs of the frequency-domain compound symbols, i.e., $p\left(X_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right) = p\left(Y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$ where $Y_{u,m,i} = X_{u,m,i}\widehat{H}_{u,i}^{(k,z-1)}$ for all m and i. The algorithm to achieve this decoding objective is the general sum-product algorithm [17, 62].

There is no known technique, however, for decoding the time-domain samples $\{y_{u,m,i}\}$. In this chapter, we introduce a new concept called "soft IDFT" to transform the APPs obtained by the sum-product decoding algorithm from the frequency domain to the time domain. At first sight, we might be tempted to obtain $p\left(\mathbf{Y}_{u,m}\left|\widehat{\mathbf{r}}_u^{(k)},\ \widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$ from $p\left(\mathbf{Y}_{u,m}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$ by considering the linear transformation between $\mathbf{y}_{u,m}$ and $\mathbf{Y}_{u,m}$, i.e., , $\mathbf{y}_{u,m} = \mathbf{F}^H\mathbf{Y}_{u,m}$ and then deriving the APP of sample $y_{u,m,i}$ by marginalizing out all other samples over $p\left(\mathbf{y}_{u,m}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$.

There are two obstacles to this approach. First, it is hard to obtain the APP of the vector of symbols $p\left(\mathbf{Y}_{u,m}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$ (the sum-product algorithm, for example, does not give this APP). Second, the aforementioned marginalization will introduce intractable complexity that is in the exponential order of the DFT size N. Here, we employ a Gaussian message passing [18] approach to solve the soft IDFT problem approximately.

We approximate the APPs of the frequency-domain symbols as independent Gaussian distributions. The independence can be achieved by the operation of interleaving in the transmitter. The Gaussianity is an assumption made to simplify computation.[4] The approximate APP of $Y_{u,m,i}$ is given by

$$\begin{aligned}\widetilde{p}\left(Y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\hat{\Omega}_u^{(k,z-1)},C_u\right.\right)\\ =\mathcal{CN}\left(Y_{u,m,i}:m_{Y_{u,m,i}},\sigma^2_{Y_{u,m,i}}\right)\end{aligned} \tag{3.18}$$

with mean and variance

$$m_{Y_{u,m,i}}=\sum\nolimits_{Y_{u,m,i}} Y_{u,m,i}p\left(Y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)$$

[4]Without the Gaussian assumption, we can solve the soft IDFT problem on a factor graph using the general message passing algorithm. We first construct the factor graph by considering the butterfly graph of FFT. With $\left\{p\left(Y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C_u\right.\right)\right\}_{i=1}^{N}$ as the input messages, we use sum-product rule to compute the output messages $\left\{p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)},\widehat{\Omega}_u^{(k,z-1)},C\right.\right)\right\}_{i=1}^{N}$ exactly. However, the number of messages on the graph increases in an exponential order as we progress from the input to the output of the IDFT factor graph (i.e., unlike the binary $Y_{u,m,i}$, $y_{u,m,i}$ has 2^N possible values.). We will treat this problem in our future study.

$$\sigma^2_{Y_{u,m,i}} = \sum_{Y_{u,m,i}} \left\| Y_{u,m,i} - m_{Y_{u,m,i}} \right\|^2 p\left(Y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \hat{\Omega}_u^{(k,z-1)}, C_u \right. \right)$$

which are computed using the APPs delivered from the sum-product channel decoding. Then, with the independence assumption, we can write the APP of $\mathbf{Y}_{u,m}$ as

$$\begin{aligned} p\left(\mathbf{Y}_{u,m} \left| \widehat{\mathbf{r}}_u^{(k)}, \hat{\Omega}_u^{(k,z-1)}, C_u \right. \right) & \\ &= \prod_{i=1}^{N} \widetilde{p}\left(Y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \hat{\Omega}_u^{(k,z-1)}, C_u \right. \right) \\ &= \mathcal{CN}\left(\mathbf{Y}_{u,m} : \mathbf{m}_{\mathbf{Y}_{u,m}}, \mathbf{C}_{\mathbf{Y}_{u,m}} \right) \end{aligned} \tag{3.19}$$

with mean vector and covariance matrix

$$\begin{aligned} \mathbf{m}_{\mathbf{Y}_{u,m}} &\stackrel{\Delta}{=} \left[m_{Y_{u,m,1}} m_{Y_{u,m,2}} \cdots m_{Y_{u,m,N}} \right]^T \\ \mathbf{C}_{\mathbf{Y}_{u,m}} &= diag\left(\left[\sigma^2_{Y_{u,m,1}} \sigma^2_{Y_{u,m,2}} \cdots \sigma^2_{Y_{u,m,N}} \right] \right) \end{aligned}$$

Finally, since $\mathbf{y}_{u,m} = \mathbf{F}^H \mathbf{Y}_{u,m}$ and $\mathbf{F}^H$ is unitary and $\mathbf{Y}_{u,m}$ is assumed to be Gaussian distributed, the APP of $\mathbf{y}_{u,m}$ is given by

$$p\left(\mathbf{y}_{u,m} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right. \right) = \mathcal{CN}\left(\mathbf{y}_{u,m} : \mathbf{m}_{\mathbf{y}_{u,m}}, \mathbf{C}_{\mathbf{y}_{u,m}} \right) \tag{3.20}$$

with mean vector and covariance matrix

$$\begin{aligned} \mathbf{m}_{\mathbf{y}_{u,m}} &= \mathbf{F}^H \mathbf{m}_{\mathbf{Y}_{u,m}} \\ \mathbf{C}_{\mathbf{y}_{u,m}} &= \mathbf{F}^H \mathbf{C}_{\mathbf{Y}_{u,m}} \mathbf{F}. \end{aligned}$$

A nice feature of Gaussian distributions is that every marginal distribution of a joint Gaussian distribution is itself a Gaussian distribution [76], and the APP of $y_{u,m,i}$ is immediately given by

$$p\left(y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C \right. \right) = \mathcal{CN}\left(y_{u,m,i} : m_{y_{u,m,i}}, \sigma^2_{y_{u,m,i}} \right) \tag{3.21}$$

with mean and variance

$$m_{y_{u,m,i}} = \left[\mathbf{m}_{\mathbf{y}_{u,m}}\right]_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \mathrm{e}^{j2\pi(i-1)(j-1)/N} m_{Y_{u,m,j}}$$

$$\sigma^2_{y_{u,m,i}} = \left[\mathbf{C}_{\mathbf{y}_{u,m}}\right]_{i,i}.$$

We will see later that when we use the Gaussian form of the APP shown in (3.21) for computing the Q function, only the mean $m_{y_{u,m,i}}$ has impact on the actual form of the Q function and the variance $\sigma^2_{y_{u,m,i}}$ can be dropped. Since the mean of $y_{u,m,i}$ is easy to compute because it is a linear combination of the means of $\{Y_{u,m,i}\}_{i=1}^{N}$, the complexity of soft IDFT is reduced to the linear order of N by Gaussian message passing.

How to obtain frequency-domain APPs

We have considered how to obtain the time-domain sample-wise APPs from the frequency-domain symbol-wise APPs, given that the symbol-wise APPs $\left\{p\left(X_{u,m,i}\left(Y_{u,m,i}\right)\middle|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right)\right\}$ are already computed by the sum-product channel decoding algorithm in the frequency domain.

To compute the symbol-wise APPs in the frequency domain, we need the estimates for the frequency-domain signals of user u. We have already computed the estimates for the time-domain signals $\left\{\widehat{\mathbf{r}}_{u,m}^{(k)}\right\}$ as in (3.7). Before transforming $\left\{\widehat{\mathbf{r}}_{u,m}^{(k)}\right\}$ into the frequency domain, we compensate for the CFO using the CFO estimate $\widehat{\varepsilon}_u^{(k,z-1)}$ from the last iteration. Then, we perform DFT on the compensated estimates for the time-domain signals

$$\widehat{\mathbf{R}}_{u,m}^{(k)} = \mathbf{F}\left(\mathbf{\Gamma}\left(-\widehat{\varepsilon}_u^{(k,z-1)}\right)\widehat{\mathbf{r}}_{u,m}^{(k)}\right)$$

for $m = 1, 2, \cdots, M$. After that, we can obtain the evidence information from $\left\{\widehat{\mathbf{R}}_{u,m}^{(k)}\right\}$

$$\begin{aligned} & p\left(\hat{R}_{u,m,i}^{(k)} \middle| X_{u,m,i}, \hat{\Omega}_u^{(k,z-1)}\right) \\ & = \mathcal{CN}\left(\hat{R}_{u,m,i}^{(k)} : e^{j\hat{\theta}_{u,m}^{(k,z-1)}} \hat{H}_{u,i}^{(k,z-1)} X_{u,m,i}, \sigma_{IN}^2\right) \end{aligned} \tag{3.22}$$

where $i = 1, 2, \cdots, N$, $m = 1, 2, \cdots, M$, $\widehat{R}_{u,m,i}^{(k)}$ is the i^{th} element of $\widehat{\mathbf{R}}_{u,m}^{(k)}$, and σ_{IN}^2 is the variance of the residual interference plus noise. The residual interference remaining in $\left\{\widehat{\mathbf{R}}_{u,m}^{(k)}\right\}$ includes the residual inter-carrier interference and multiple user interference. The variance σ_{IN}^2 is an unknown variable whose value is changing over the iterations. Before channel decoding, we employ a simple method for estimating σ_{IN}^2 in each iteration

$$\widehat{\sigma_{IN}^2} = \frac{1}{NM} \sum_{m=1}^{M} \sum_{i=1}^{N} \left(\left\| \widehat{R}_{u,m,i}^{(k)} \right\|^2 - \left\| \widehat{H}_{u,i}^{(k,z-1)} \right\|^2 \right) \tag{3.23}$$

The estimate $\widehat{\sigma_{IN}^2}$ is used to replace σ_{IN}^2 in (3.22) when we compute the evidence information.

The evidence information computed above is used to initialize the standard sum-product algorithm for channel decoding. We can derive the sum-product channel decoding algorithm as a message passing algorithm on the factor graph that models the encoding constraint C_u. Readers familiar with the sum-product algorithm and factor graphs [17] can readily complete this task; we omit the details here.

Ultimate Form for Q Function

Finally, using the Gaussian expression of the sample-wise APP

shown in (3.21), we can simplify the sample-wise Q function (3.16) into a compact form (the derivation is given in Appendix B):

$$\begin{aligned} & Q_{m,i}\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)} \right.\right) \\ & \propto -\left\| \widehat{r}_{u,m,i}^{(k)} - e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u (i-1)/N} m_{y_{u,m,i}} \right\|^2 \end{aligned} \tag{3.24}$$

From (3.24), we note that, due to the Gaussian expression for the APP of $y_{u,m,i}$ in (3.21), only the mean of the time-domain sample $y_{u,m,i}$ appears in the sample-wise Q function. This greatly reduces the complexities involved in the computation of the Q function. With (3.24), the overall Q function can now be written as

$$\begin{aligned} & Q\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)} \right.\right) \\ & \propto -\sum_{m=1}^{M} \sum_{i=1}^{N} \left\| \widehat{r}_{u,m,i}^{(k)} - e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u (i-1)/N} m_{y_{u,m,i}} \right\|^2 \\ & \propto -\sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{m}_{\mathbf{y}_{u,m}} \right\|^2. \end{aligned} \tag{3.25}$$

We denote the mean vector of the symbol vector $\mathbf{X}_{u,m}$ by $\mathbf{m}_{\mathbf{X}_{u,m}} = \left[m_{X_{u,m,1}} m_{X_{u,m,2}} \cdots m_{X_{u,m,N}} \right]^T$, whose i^{th} element is computed using the symbol-wise APP

$$m_{X_{u,m,i}} = \sum_{X_{u,m,i}} X_{u,m,i} p\left(X_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \tag{3.26}$$

With the above notations, we have the following relationship:

$$\mathbf{m}_{\mathbf{y}_{u,m}} = \mathbf{F}^H \mathbf{m}_{\mathbf{Y}_{u,m}} = \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \mathbf{h}_u \tag{3.27}$$

Substituting (3.27) into (3.25) gives the ultimate form of the Q function:

$$\begin{aligned} & Q\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)} \right.\right) \\ & \propto -\sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \mathbf{h}_u \right\|^2 \end{aligned} \tag{3.28}$$

So far, we have finished the E-step of ECM. We next turn to the M-step of ECM.

M-step of ECM:

The M-step of ECM updates the new parameter estimates $\widehat{\Omega}_u^{(k,z)} = \left\{\widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z)}\right\}$ by maximizing the Q function in (3.28). ECM breaks the maximization procedure of the z^{th} iteration into three stages, where the t^{th} stage updates the parameters from $\widehat{\Omega}_u^{\left(k,z-1+\frac{t-1}{3}\right)}$ to $\widehat{\Omega}_u^{\left(k,z-1+\frac{t}{3}\right)}$, $t = 1, 2, 3$, as follows:

The First Stage — CFO Estimation

The first stage updates the CFO with the phase drifts and the channel gains fixed to their last estimates. The new set of parameter estimates after the first stage is $\widehat{\Omega}_u^{\left(k,z-1+\frac{1}{3}\right)} = \left\{\widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z-1)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z-1)}\right\}$. The new CFO estimate $\widehat{\varepsilon}_u^{(k,z)}$ is obtained by

$$\widehat{\varepsilon}_u^{(k,z)} = \arg\max_{\varepsilon_u} \left\{ - \sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right\|^2 \right\} \tag{3.29}$$

where the objective function is the result of replacing the variables $\{\theta_{u,m}\}_{m=1}^{M}$ and $\mathbf{h}_u$ in the Q function (3.28) with the fixed values $\left\{\widehat{\theta}_{u,m}^{(k,z-1)}\right\}_{m=1}^{M}$ and $\widehat{\mathbf{h}}_u^{(k,z-1)}$.

The exhaustive search method for solving (3.29) is computationally complex, since ε_u is a continuous variable. To obtain a practical solution for (3.29), we could approximate $\mathbf{\Gamma}\left(\varepsilon_u\right)$ in (3.29) using its

Taylor expansion around $\widehat{\varepsilon}_u^{(k,z-1)}$ with terms above the second order truncated. We then differentiate the resulting objective function of (3.29) with respect to ε_u and set the derivative to zero. Solving the equation yields a closed-form solution for (3.29):

$$\begin{aligned}\widehat{\varepsilon}_u^{(k,z)} &= \widehat{\varepsilon}_u^{(k,z-1)} \\ &+ \frac{Re\left\{\sum_{m=1}^{M} e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \widehat{\mathbf{r}}_{u,m}^{(k)H} \mathbf{\Gamma}'\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)}\right\}}{Re\left\{-\sum_{m=1}^{M} e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \widehat{\mathbf{r}}_{u,m}^{(k)H} \mathbf{\Gamma}''\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)}\right\}}\end{aligned} \tag{3.30}$$

where

$$\begin{aligned}\mathbf{\Gamma}'\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) &= (j2\pi/N)\, \mathbf{\Psi}\mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) \\ \mathbf{\Gamma}''\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) &= -(2\pi/N)^2 \mathbf{\Psi}^2 \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z-1)}\right)\end{aligned}$$

with $\mathbf{\Psi} = diag\{[0, 1, 2, \cdots, N-1]\}$, are the first and second derivates of $\mathbf{\Gamma}(\varepsilon_u)$ at the point $\widehat{\varepsilon}_u^{(k,z-1)}$. The detailed derivation of (3.30) can be found in Appendix C. With the updated CFO estimate, we go to the second stage of the M-step.

The Second Stage — Phase Tracking

The second stage updates the phase drifts with the CFO and the channel gains fixed to their last estimates. The new set of parameter estimates after the second stage is $\widehat{\Omega}_u^{\left(k,z-1+\frac{2}{3}\right)} = \left\{\widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z-1)}\right\}$, where the new phase estimates are given by

$$\begin{aligned}\left\{\hat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M} = \arg \max_{\{\theta_{u,m}\}_{m=1}^{M}} -\sum_{m=1}^{M} \Bigg\{ \\ \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\theta_{u,m}} \Gamma\left(\hat{\varepsilon}_u^{(k,z)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right\|^2 \Bigg\}\end{aligned} \tag{3.31}$$

The objective function in (3.31) is obtained from the Q function in (3.28) $\varepsilon_u = \widehat{\varepsilon}_u^{(k,z)}$ with $\mathbf{h}_u = \widehat{\mathbf{h}}_u^{(k,z-1)}$.

Since we assume that the phase drifts $\{\theta_{u,m}\}_{m=1}^{M}$ are independent for different blocks, we can decouple the problem (3.31) into M sub-problems

$$\begin{aligned}\widehat{\theta}_{u,m}^{(k,z)} = \arg\max_{\theta_{u,m}} \Bigg\{ \\ -\left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right\|^2 \Bigg\}\end{aligned} \tag{3.32}$$

for $m = 1, 2, \cdots, M$, one for each block. Directly solving (3.32) gives the phase estimate for the m^{th} block:

$$\widehat{\theta}_{u,m}^{(k,z)} = \angle\left[\mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right]^H \widehat{\mathbf{r}}_{u,m}^{(k)} \tag{3.33}$$

With the new phase estimates $\left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}$, we go to the third stage.

The Third Stage — Channel Estimation

The third state updates the channel gains with the CFO and the phase drifts fixed to their last estimates. The new set of parameter estimates after the third stage is $\widehat{\Omega}_u^{(k,z)} = \left\{ \widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z)} \right\}$, where the new channel estimates are given by

$$\begin{aligned}\widehat{\mathbf{h}}_u^{(k,z)} = \arg\max_{\mathbf{h}_u} \Bigg\{ \\ -\sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\widehat{\theta}_{u,m}^{(k,z)}} \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z)}\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \mathbf{h}_u \right\|^2 \Bigg\}\end{aligned} \tag{3.34}$$

The objective function in (3.34) is obtained from the Q function in (3.28) with $\varepsilon_u = \widehat{\varepsilon}_u^{(k,z)}$, $\left\{\theta_{u,m} = \widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}$.

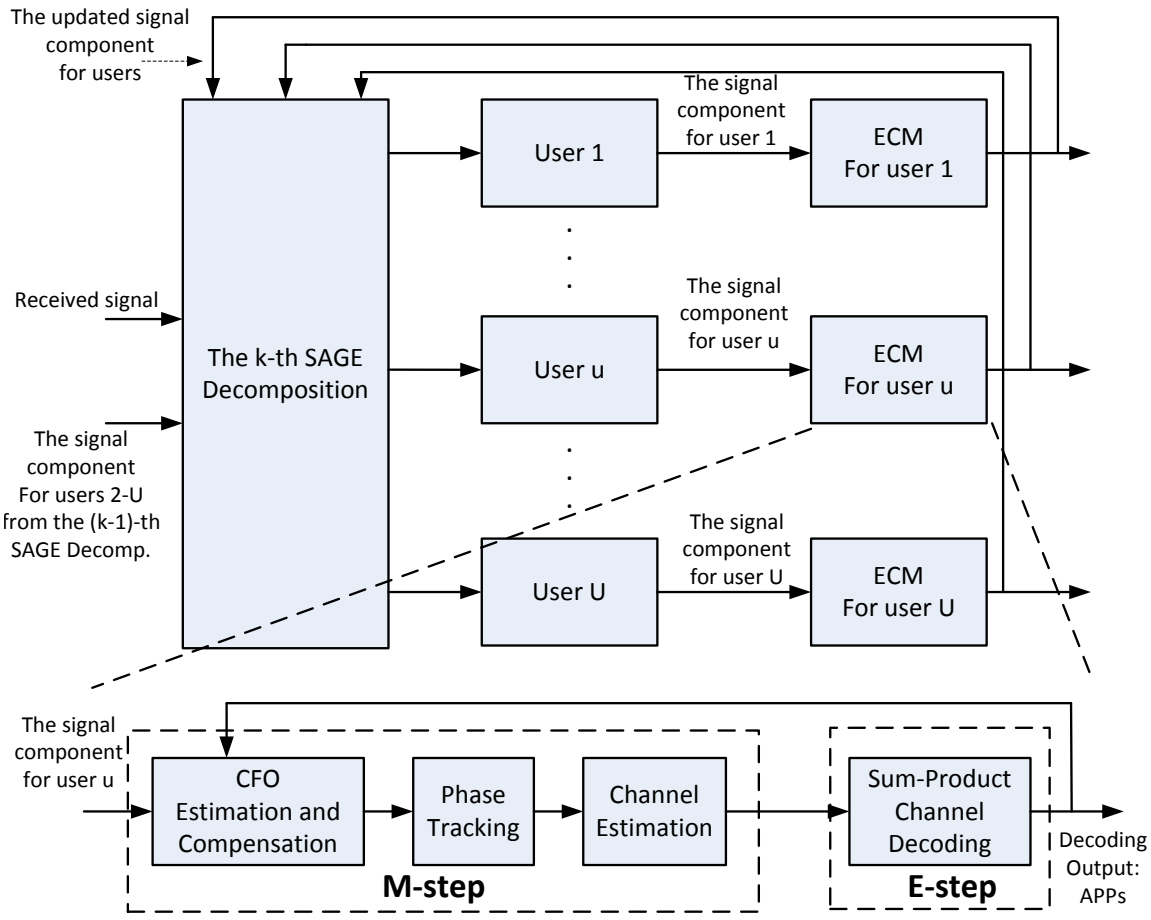


Figure 3.2: The operating flow chart for the proposed SAGE-ECM algorithm.

The solution for (3.34) is given by the least square (LS) estimate [77]:

$$\begin{aligned}&\widehat{\mathbf{h}}_u^{(k,z)}\\&= \tfrac{1}{M}\sum_{m=1}^{M} e^{-j\widehat{\theta}_{u,m}}\mathbf{F}^H\mathbf{D}^{-1}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right)\mathbf{F}\boldsymbol{\Gamma}_m\left(-\widehat{\varepsilon}_u^{(k,z)}\right)\widehat{\mathbf{r}}_{u,m}^{(k)}\end{aligned} \tag{3.35}$$

This finishes the final stage of the M-step. We then iterate back to the E-step with the set of new parameter estimates $\widehat{\Omega}_u^{(k,z)} = \left\{\widehat{\varepsilon}_u^{(k,z)}, \left\{\widehat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z)}\right\}$.

Initialization and Termination of ECM Iteration:

We bootstrap the ECM iteration with initial estimates $\widehat{\Omega}_u^{(k,0)} = \left\{\widehat{\varepsilon}_u^{(k,0)}, \left\{\widehat{\theta}_{u,m}^{(k,0)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,0)}\right\}$, where the initial CFO estimate and channel estimates $\widehat{\varepsilon}_u^{(k,0)}$, $\widehat{\mathbf{h}}_u^{(k,0)}$ are obtained from the preamble of user u; the initial phase estimate $\widehat{\theta}_{u,m}^{(k,0)}$ is obtained from the pilot subcarriers of the m^{th} block for every m.

We repeat the E-step and M-step of ECM iteratively. When the number of iterations z reaches the preset maximum limit Z, we terminate the ECM algorithm for user u and take the final estimates as the approximate solution for the joint estimation problem (8):

$$\begin{aligned}&\left\{\hat{\varepsilon}_u^{(k)}, \left\{\hat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k)}, \widehat{\mathbf{X}}_u^{(k)}\right\}\\ &= \left\{\hat{\varepsilon}_u^{(k,Z)}, \left\{\hat{\theta}_{u,m}^{(k,Z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,Z)}, \widehat{\mathbf{X}}_u^{(k,Z)}\right\}\end{aligned}$$

where $\left\{\widehat{\varepsilon}_u^{(k,Z)}, \left\{\widehat{\theta}_{u,m}^{(k,Z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,Z)}\right\}$ are the final parameter estimates obtained from the M-step of the Z^{th} ECM iteration, $\widehat{\mathbf{X}}_u^{(k,Z)}$ are obtained by making hard decisions based on the symbol-wise APPs $\left\{p\left(X_{u,m,i} \middle| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,Z)}, C_u\right)\right\}$. A operating flow chart for the proposed SAGE-ECM algorithm is shown in Fig. 3.2.

3.3.4 Alternative Iterative Receiver

We have described our proposed SAGE-ECM receiver in Section 3.3.3. We can obtain an alternative receiver by adopting another message passing algorithm — the min-sum algorithm — for channel decoding.

Let us revisit (3.8) and consider how to solve it iteratively. We can

apply SAGE again in this single-user subproblem. Specifically, we can break up one iteration into several stages where one stage updates just one parameter while fixing all other parameters to their last estimates. This iterative receiver corresponds to a pure SAGE framework, where all variables are treated as parameters and updated in a sequential manner.

Using the above idea, the z^{th} iteration for solving (3.8) consists of four stages. In the first, second and third stages, the CFOs, the phases and the channel gains are updated to $\widehat{\varepsilon}_u^{(k,z)}$, $\left\{\theta_{u,m}^{(k,z)}\right\}_{m=1}^{M}$ and $\widehat{\mathbf{h}}_u^{(k,z)}$ employing the methods of (3.30), (3.33) and (3.35), respectively. The fourth stage updates the transmit symbols by

$$\begin{aligned}\widehat{\mathbf{X}}_u^{(k,z)} = \arg\max_{\mathbf{X}_u} \Bigg\{ \\ -\sum_{m=1}^{M} \left\| \widehat{\mathbf{R}}_{u,m}^{(k)} - e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z)} \left(\widehat{\Omega}_u'^{(k,z-1)}\right) \right\|^2 \Bigg\}\end{aligned} \tag{3.36}$$

where $\widehat{\mathbf{R}}_{u,m}^{(k)} = \mathbf{F}\left(\mathbf{\Gamma}\left(-\widehat{\varepsilon}_u^{(k,z-1)}\right)\widehat{\mathbf{r}}_{u,m}^{(k)}\right)$ is the vector of the frequency-domain signals after CFO compensation. With $\left\{\hat{\varepsilon}_u^{(k,z)}, \left\{\hat{\theta}_{u,m}^{(k,z)}\right\}_{m=1}^{M}, \widehat{\mathbf{h}}_u^{(k,z)}, \widehat{\mathbf{X}}_u^{(k,z)},\right\}$, we complete the z^{th} iteration.

The fourth stage in (3.36) corresponds to the channel decoding operation. In contrast to the channel decoding in the SAGE-ECM framework in Section 3.3.3, where the sum-product algorithm was used to obtain the APPs of the transmit symbols, the channel decoding method corresponding to (3.36) is the min-sum algorithm [17, 62]. For example, if convolution codes are employed, the SAGE-ECM framework (in Section 3.3.3) would use the BCJR algorithm,

but the pure SAGE framework here would use the Viterbi algorithm (a special case of the min-sum algorithm [17, 62]).

Henceforth, we will refer to the iterative receiver in Section 3.3.3 as the SAGE-ECM Sum-Product Rx, and the iterative receiver here as the SAGE Min-Sum Rx. We compare their performances through simulation study in the next section.

The key difference between the SAGE Min-Sum Rx in this section and the SAGE-ECM Sum-Product Rx in Section 3.3.3 is how channel decoding assists the channel-parameter estimation. For the SAGE Min-Sum Rx, the symbol estimates $\widehat{\mathbf{X}}_u^{(k,z)}$ given by min-sum channel decoding are hard decisions — we lose soft information on $\mathbf{X}_u$ that specifies the levels of confidence on our estimates. As will be seen in the next section, these hard decisions cause significant performance degradation when accepted as parameter estimates. The SAGE-ECM Sum-Product Rx assists the channel-parameter estimation using the soft information on the transmit symbols in a more sophisticated manner. Finally, we remark that the SAGE Min-Sum Rx here is equivalent to the receiver proposed in [44] (see a detail discussion about [44] in the nest subsection).

3.3.5 Discussion of Receiver in [44]

Ref. [44] proposed an iterative receiver for solving the problem of joint channel-parameter estimation, CFO compensation and data detection in uncoded OFDMA systems. The authors of [44] attempted to construct the receiver based on a SAGE-ECM framework. As

explained later in this appendix, this attempt ended up with a pure SAGE framework due to an approximation.

A major difference between the framework in [44] and ours is the assignment of hidden data in the single-user subproblem within the SAGE-ECM framework. For us, the data symbols $\mathbf{X}_u$ are the hidden data and the channel gains $\mathbf{h}_u$ are one of the parameters. For in [44], the channel gains $\mathbf{h}_u$ are the hidden data, and the data symbols $\mathbf{X}_u$ are one of the parameters.

Another difference is that, unlike our work here, [44] did not incorporate channel coding/decoding — [44] only focused on detection of individual uncoded OFDM blocks $\mathbf{X}_{u,m}$, and the correlations among different data blocks of the overall frame $\mathbf{X}_u$ induced by channel coding are not exploited to further improve performance. In particular, each OFDM symbols are decoded independently. Here, for consistency with our problem, we extend the treatment of [44] to coded OFDM-IDMA systems (i.e., we incorporate channel coding into the framework of [44]). With this extension, we have $\Omega'_u = \left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u\right\}$ as the parameters and $\mathbf{h}_u$ as the hidden data.

The corresponding ECM aims to solve the following problem iteratively:

$$\begin{aligned}
&\left(\widehat{\varepsilon}_u^{(k)}, \left\{\widehat{\theta}_{u,m}^{(k)}\right\}_{m=1}^{M}, \widehat{\mathbf{X}}_u^{(k)}\right) \\
&= \arg \max_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u \in C_u\right)} \log p\left(\widehat{\mathbf{r}}_u^{(k)} \left| \varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u \right.\right) \\
&= \arg \max_{\left(\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u \in C_u\right)} \\
&\qquad \left\{\log \int p\left(\widehat{\mathbf{r}}_u^{(k)}, \mathbf{h}_u \left| \varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{X}_u \right.\right) d\mathbf{h}_u\right\}
\end{aligned} \tag{3.37}$$

Now, ECM solves (3.37) iteratively. In the z^{th} iteration of ECM, the E-step of ECM computes the Q function of

$$\begin{aligned} &Q\left(\Omega'_u \middle| \widehat{\Omega}'^{(k,z-1)}_u\right) \\ &= E_{\mathbf{h}_u}\left\{\ln p\left(\widehat{\mathbf{r}}^{(k)}_{u,m} | \mathbf{h}_u, \Omega'_u\right) p\left(\widehat{\mathbf{r}}^{(k)}_{u,m} \middle| \mathbf{h}_u, \widehat{\Omega}'^{(k,z-1)}_u\right)\right\} \end{aligned} \tag{3.38}$$

and the M-step of ECM updates the tentative estimates $\widehat{\Omega}'^{(k,z-1)}_u$ for Ω'_u by finding the Ω'_u that maximize (3.37).

The exact computation of the Q function in (3.38) is complex and not feasible from the implementation viewpoint (cf .eq (13) in [44] for the Q function of the uncoded case). To reduce complexity, [44] made an approximation on the computed Q function. Extending the result in [44] to channel-coded case (cf. eq (16) in [44]), the E-step of ECM approximates the Q function as

$$\begin{aligned} &Q\left(\Omega'_u \middle| \widehat{\Omega}'^{(k,z-1)}_u\right) \\ &= -\sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}^{(k)}_{u,m} - e^{j\theta_{u,m}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{X}_{u,m}\right) \right. \\ &\qquad \left. \times \mathbf{F} \widehat{\mathbf{h}}^{(LS)}_u \left(\widehat{\Omega}'^{(k,z-1)}_u\right) \right\|^2 \end{aligned} \tag{3.39}$$

where $\widehat{\mathbf{h}}^{(LS)}_u \left(\widehat{\Omega}'^{(k,z-1)}_u\right)$ is the LS estimate of $\mathbf{h}_u$ obtained with ${\Omega_u}' = \widehat{\Omega}'^{(k,z-1)}_u$ in the signal model (3.2). With this Q function, the CFO and the phases can be updated to $\widehat{\varepsilon}^{(k,z)}_u$ and $\left\{\theta^{(k,z)}_{u,m}\right\}_{m=1}^{M}$ using the same methods of (3.30) and (3.33) in Section 3.3.3, respectively. However, the transmit symbols $\mathbf{X}_u$ here is updated in a different way than in Section 3.3.3. Specifically, the stage in the M-step for updating the transmit symbols now becomes (3.36), which can be solved using the min-sum channel decoding, as explained in Section 3.3.4.

A subtle but important consequence of approximating the Q function as in (3.39) is that, although the development of the receiver in [44] begins with the ECM algorithm, it ends up with a pure SAGE solution to (3.8). In other words, it is not a SAGE-ECM algorithm anymore, but a pure SAGE algorithm. The algorithm essentially treats all the variables in $\left\{\varepsilon_u, \{\theta_{u,m}\}_{m=1}^{M}, \mathbf{h}_u, \mathbf{X}_u\right\}$ as parameters, and none of them as hidden data. As in the SAGE algorithm, the parameters are estimated sequentially one at a time; when one parameter is under estimation, all other parameters are fixed to their estimates from the last iteration. This pure SAGE algorithm is exactly the same as the SAGE Min-Sum Rx we discussed in Section 3.3.4 and adopted as one of the benchmarks in our simulation studies in Section 3.4.

3.4 Numerical Results

This section presents computer simulation results and experimental results on software defined radio.

In all simulations, we consider an OFDM-IDMA system with $U = 3$ users. The frame format is a slightly modified version of the 802.11a frame format [78]. The DFT size is $N = 64$. The CP is of length $N_{cp} = 16$. Among the $N = 64$ subcarriers, there are $N_d = 48$ data subcarriers, $N_p = 6$ pilot subcarriers and 10 unused guard band subcarriers. Each user transmits known symbols on 2 of the 6 pilot subcarriers, and nulls the signal on the other 4 pilot subcarriers. The modulation is BPSK. A way to realize a low-rate channel code scheme

for IDMA is to serially concatenate a forward error correction (FEC) code with a repetition code [56]. In our simulation, the FEC is a regular Repeat Accumulate (RA) code [49] with code rate $R_1 = 1/3$, and the code rate of the repetition code is $R_2 = 1/U = 1/3$. Therefore, the overall code rate is $R = R_1 R_2 = 1/9$. The interleavers of all three users are generated randomly. The payload of each frame has 450 data OFDM blocks. The two preamble blocks for all three users are two successive copies of the long training sequence (LTS) defined in the 802.11a standard. However, as described in Section 3.2.1, the two preambles of different users occupy different block times.

We set the length of the discrete channel vectors for all three users to $L_1 = L_2 = L_3 = 4$. The channel taps are generated according to the channel mode given in Section 3.2.2. We assume that the receiver of the base station can capture the first channel path of user 1, so that the timing mismatch between user 1 and the base station is set to $\mu_1 = 0$. Furthermore, μ_2 and μ_3 are randomly chosen from the interval $[0, 9]$. Thus, the loose time synchronization requirement is satisfied in simulations. The CFOs of the three users are set to $[\varepsilon_1, \varepsilon_2, \varepsilon_3]^T = \rho[1, -1, -1]^T$, where ρ is the so-called CFO attenuation factor [65], and it is a deterministic parameter ranging in $[0, 0.5]$. In our simulations, we vary the value of ρ to investigate its impact on system performance. SNR is defined as E_b/N_0, where E_b is the energy per information bit and N_0 is the noise variance. All simulation results presented here are obtained by averaging over 3×10^3 frames.

For performance comparison, we investigate the following four approaches for OFDM-IDMA systems: (i) the iterative interference

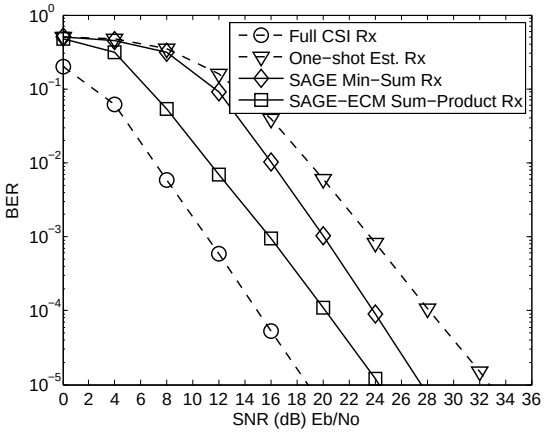


Figure 3.3: BER versus SNRs with $\rho = 2$, $U = 3$.

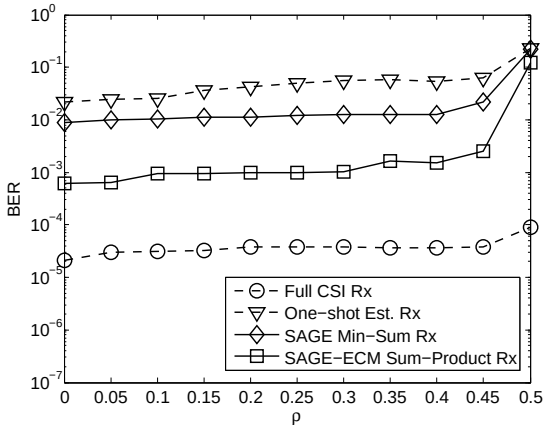


Figure 3.4: BER versus ρ with SNR=16 dB, $U = 3$.

cancellation approach in [64] with perfect knowledge of channel parameters (Full CSI Rx); (ii) the iterative interference cancellation approach in [64] with the one-shot channel parameter estimations obtained from the preambles and pilots (One-shot Est. Rx); (iii) the SAGE Min-Sum RX approach described in Section 3.3.4 as a benchmark; and (iv) the SAGE-ECM Sum-Product Rx approach proposed in Section 3.3.3. For all receivers, we performed iterations until the algorithm converged (with SAGE iteration number $K = 6$, and ECM iteration number $Z = 10$). We remark that the performance of One-shot Est. Rx is the initial point of SAGE-ECM Sum-Product Rx.

3.4.1 Simulation Results

Fig. 3.3 presents the results of bit error rate (BER) versus SNR. The CFO attenuation factor is fixed to $\rho = 0.2$. From the results, we

first note that the BER gap between Full CSI Rx and One-shot Est. Rx is large. Specifically, the estimation errors of one-shot estimation induce around 13 dB SNR loss at BER = 10^{-5}. Intuitively, since each imperfect parameter estimate induces an SNR penalty and we have many channel parameters in the system, the SNR penalties accumulate to an overall large penalty.

The SAGE-ECM approach reduces the performance loss, as shown in Fig. 3.3. SAGE-ECM Sum-Produc Rx yields 8dB SNR improvement and SAGE Min-Sum Rx yields 5dB SNR improvement at BER = 10^{-5}. The 3 dB performance gap between SAGE-ECM Sum-Product Rx and SAGE Min-Sum Rx justifies that assigning the data symbols (as opposed to channel parameters) as the hidden data in the ECM framework.

Fig. 3.4 shows the impact of CFO attenuation factor ρ on BER, with SNR fixed to 16 dB. The results again confirm the better performance of SAGE-ECM Rx over One-shot Est. Rx. It also shows that the BERs of all approaches are insensitive to ρ: note that the different approaches have different performances; just that the performance of each approach is not sensitive to ρ. This result implies that it is not the magnitude of that affects performance; it is the estimation errors of ρ, which vary among the different approaches.

We also evaluate the mean square error (MSE) of the estimated channel parameters. Fig. 3.5, Fig. 3.6 and Fig. 3.7 present the MSEs of the estimated CFOs, channels and phases versus SNR. The CFO attenuation factor is fixed to $\rho = 0.2$. The units of the CFOs and the phases are Hz and radian. From the MSE results Fig. 3.5-

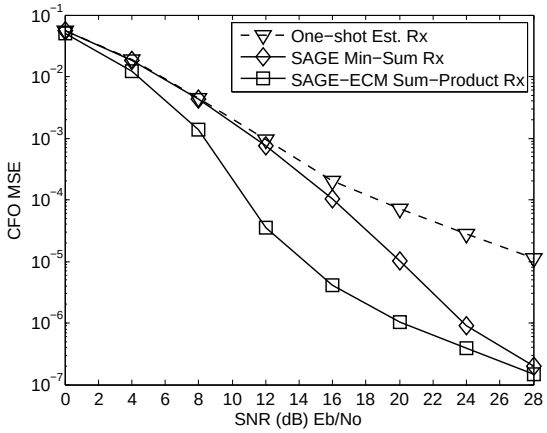


Figure 3.5: CFO MSEs versus SNRs with $\rho = 2$, $U = 3$.

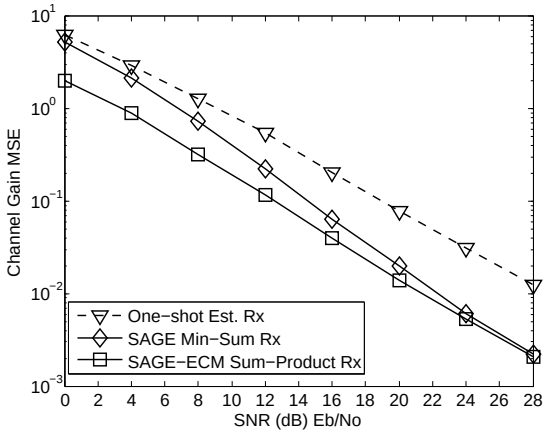


Figure 3.6: Channel gain MSEs versus SNRs with $\rho = 2$, $U = 3$.

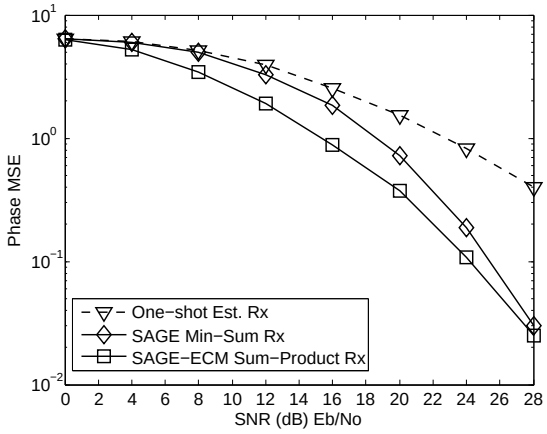


Figure 3.7: Phase MSEs versus SNRs with $\rho = 2$, $U = 3$.

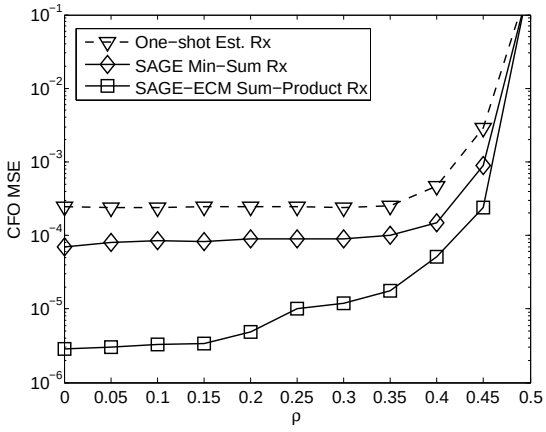


Figure 3.8: CFO MSEs versus ρ with SNR=16 dB, $U = 3$.

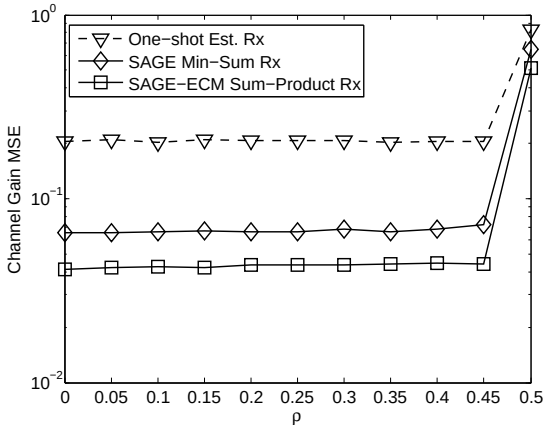


Figure 3.9: Channel gain MSEs versus ρ with SNR=16 dB, $U = 3$.

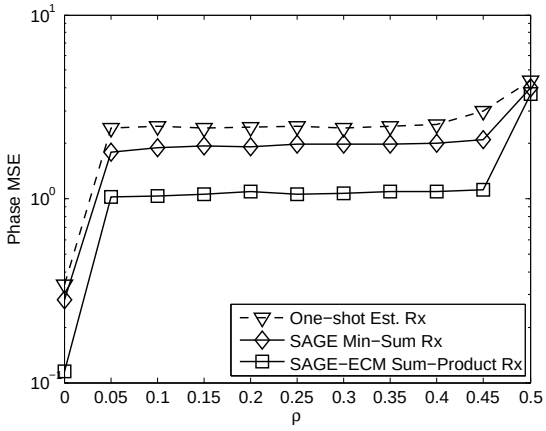


Figure 3.10: CFO MSEs versus ρ with SNR=16 dB, $U = 3$.

3.7, we clearly see that the SAGE-ECM approaches do have more accurate estimates than the traditional preamble/pilot-based one-shot estimations. Moreover, the estimations of SAGE-ECM Sum-Product RX are better than that of SAGE Min-Sum RX; the difference in MSEs between them decreases as the SNR increases. The reason is that the results of min-sum decoding are hard decisions. In the low SNR regime, the hard decisions are not reliable enough and will propagate the decoding errors to channel parameter estimations. In the high SNR regime, the decoding results of min-sum decoding approaches that of sum-product decoding, thus the error gaps in MSEs become narrow. We also note that the CFO MSE difference between SAGE-ECM Sum-Product Rx and SAGE Min-Sum Rx is the most obvious one. Therefore, the BER gap between SAGE-ECM Sum-Product Rx and SAGE Min-Sum Rx in Fig. 3.3 can be largely explained by the CFO MSE rather than the phase MSE and the channel gain MSE.

Fig. 3.8, Fig. 3.9 and Fig. 3.10 show the MSEs of the estimated CFOs, channels and phases versus ρ. The SNR is fixed to 16 dB. The performance trends are the same with the BER in terms of ρ, as shown in Fig. 3.4. SAGE-ECM Sum-Product Rx has better MSE performances than SAGE Min-Sum Rx and One-shot Est. Rx do. For all approaches, their estimation errors depends only weakly on ρ for a wide range of ρ; performances only degrade as ρ is near 0.5 (the worst case). We thus conclude that it is the error in the estimation of CFO that has more effects than the actual CFO value itself.

3.4.2 Experimental Results

Going beyond simulations, we also verify our proposed approach experimentally. We implemented an OFDM-IDMA system using a software defined radio (SDR) platform. We collected the data for the received signal from the experimental system and evaluate the performance of the proposed approach using the collected data.

The experimental system is built on the USRP N210 hardware [79] and the GNU Radio software with the UHD hardware driver [80]. We emulated an OFDM-IDMA system that includes one base station and two users ($U = 2$), by deploying three sets of USRP N210 with XCVR2450 boards [79] in our lab.[5] The base station

[5]We performed the experiment for OFDM-IDMA using the SDR prototype of the physical-

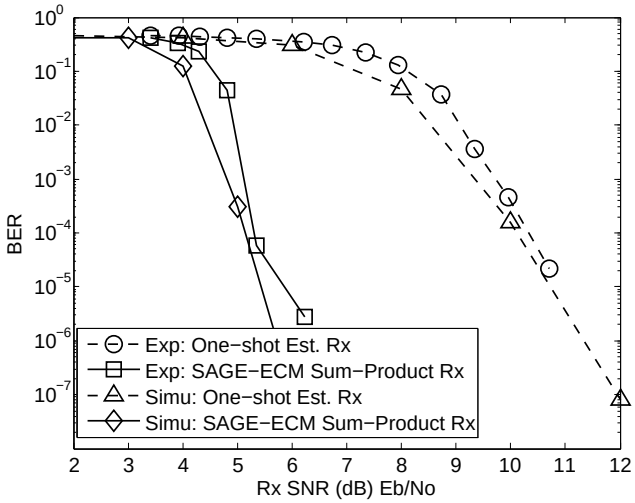


Figure 3.11: Experimental BER results with $U = 2$.

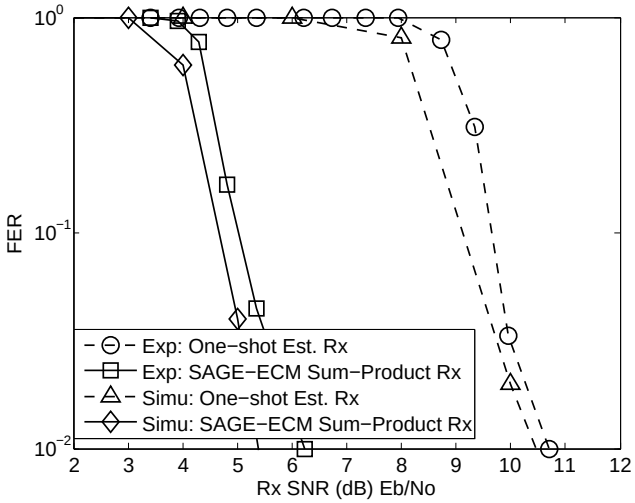


Figure 3.12: Experimental FER results with $U = 2$.

used 802.11 channel 1 (2.412GHz) to poll the two users to transmit together at channel 11 (2.462GHz), thereby achieving the loose-time synchronization mentioned in Section 3.2.3. The wireless channel bandwidth of our network is 4MHz. The use of 4 MHz bandwidth rather than a wider bandwidth is due to the limitation of the USRP hardware. After the terminal users received a beacon from the base station, they then transmit their signals simultaneously. The base station received the simultaneous transmissions and converted them to digital data for processing. We performed controlled experiments for different received SNRs. The receive powers of frames from two users at the base station were adjusted to be balanced (power imbalance within 1dB). The base station transmits 100 beacons to trigger 100 simultaneous uplink transmissions for each fixed SNR.

The frame format used is similar to the one used in simulations except for the following two differences. First, in the experimental setup, there are four pilot subcarriers within each OFDM block (as opposed to six in simulations) and each user transmits pilot symbols over two of them. Second, to save computation time in our experimental setup, the channel coding scheme used is only the 1/3 coding rate RA code without the repetition code. Each frame includes

layer network coding (PNC) systems reported in detail in [15, 70]. In the uplink of PNC, two users transmit signals to the relay simultaneously. This is similar to multiple-access systems. Therefore, we can borrow it for our use here.

256 OFDM blocks.

The experimental BER results are shown in Fig. 3.11. The frame error rate (FER) results are shown in Fig. 3.12. We compare the performances of SAGE-ECM Sum-Product Rx with One-shot Est. Rx Generally, the performance trends are similar to those in our simulation results. Specifically, we observe that SAGE-ECM Sum-Product Rx achieves around 5 dB SNR gain over One-shot Est. Rx at the BER of 10^{-5} in the experiment. We note that the shapes of the experimental BER curves are different from those of the simulations in Fig. 3.3. The reason is that the channel in our experimental environment is rather flat over the 4MHz bandwidth. To verify our experimental results, we perform an additional simulation where the system includes $U = 2$ users, the frame format is the same as that in our experiment, the channel has one Rayleigh path (thus it is flat), and the CFOs are set to $[\varepsilon_1, \varepsilon_2,]^T = [0.06, 0.11]^T$ that are the means of the measured CFO values in our experiment. The simulated BER and FER results under this setup are presented in Fig. 3.11 and Fig. 3.12, respectively. As can be seen, the simulated results are consistent with the experimental results in that there is also around 5 dB SNR gain by SAGE-ECM Sum-Product Rx over One-shot Est. Rx.

□ **End of chapter.**

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Two major signal processing challenges for multiuser systems are (i) how to estimate the multiple parameters in a satisfying manner and (ii) how to compensate for these parameters when they cause system degradations. In this thesis, we aim to solve these challenges for two types of multiuser systems: PNC and MUD systems. In particular, we put forth frameworks that enable joint multiple parameter estimation and channel decoding in PNC and MUD systems that tackle the two signal processing challenges in a comprehensive manner. The general contributions of this thesis are as follows:

- In Chapter 2, we proposed an EM-BP factor-graph framework for solving the problem of joint channel estimation and channel decoding in PNC systems. This framework consists of EM message passing for channel estimation and BP message passing for channel decoding. The output of one forms the input

of the other, and vice versa, so that the results of channel estimation and channel decoding can be refined in an iterative and progressive manner. A salient feature of our framework is the use of 'virtual channel decoding' to ensure optimal performance for PNC systems. Furthermore, we show that the EM messages in our factor-graph framework are Gaussian messages that can be characterized by their means and variances only, and this greatly reduces computation complexity. We refer to the receiver based on this framework as EM-BP PNC.

- Our simulation results in Chapter 2 indicate that the BER of EM-BP PNC can approach that of an ideal PNC receiver with perfect CSI. In addition, EM-BP PNC outperforms other receivers in terms of BER and MSE. Beyond PNC, we believe the EM-BP factor graph framework proposed in this work can also be used to construct receivers with superior performance in conventional single-user and multi-user systems.

- In Appendix D, We also applied the EM-BP algorithm to solve the problems of phase tracking and channel decoding in OFDM PNC systems jointly. Based on the framework of EM-BP proposed in Chapter 2, we designed the algorithm for OFDM PNC. To investigate the performance of the proposed method, we performed computer based simulations and SDR based real experiments. Our simulation and real experiment results show that, compared to a traditional method that performs phase tracking and channel decoding separately, the proposed method

can obtain 2 dB gain after the first iteration, and around 3 dB gain after convergence of the iterations.

- In Chapter 3, we exploited the SAGE and ECM algorithms to solve the problem of joint channel-parameter estimation, CFO compensation and channel decoding iteratively for multiuser OFDM-IDMA systems. Our framework is motivated by the fact that for multiuser OFDM systems, (i) one-shot non-iterative parameter estimation does not yield satisfactory accuracy; and (ii) one-shot non-iterative CFO compensation is impossible. For these reasons, we propose to solve the overall problem of channel-parameter estimation, CFO compensation, and channel decoding jointly and iteratively. Within our framework, we compared different schemes of assigning the role of hidden data in ECM, and concluded that treating the data symbol (as opposed to channel parameters) as the hidden data in ECM leads to better performance. A salient feature of our approach is that we bridged the time-domain channel-parameter estimation procedure and the frequency-domain channel decoding procedure using the technique of Gaussian message passing to approximate the computation of "soft IDFT".

- Our simulation results and real SDR experimental results in Chapter 3 indicate that compared with the traditional multiuser approach, the proposed SAGE-ECM approach can obtain 5 dB SNR gain for the two-user case, and 8 dB SNR gain for the three-user case.

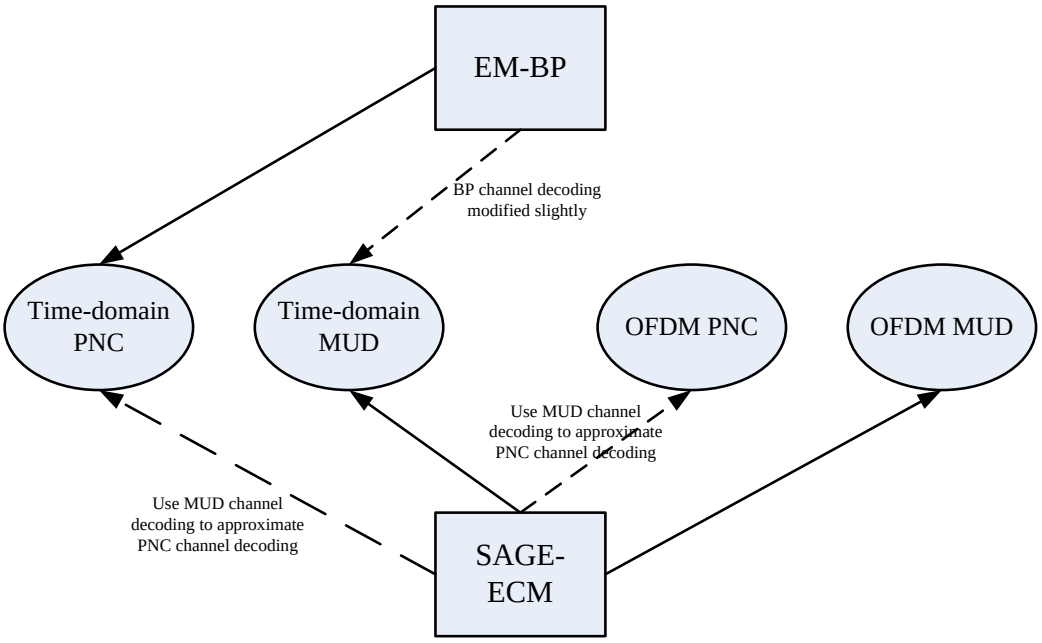


Figure 4.1: The summary for differences and the positioning of EM-BP and SAGE-ECM.

We summarize the differences and the positioning of EM-BP and SAGE-ECM in Fig. 4.1, where we relate them to four systems, time-domain PNC, time-domain MUD, OFDM PNC and OFDM MUD. A solid arrow points from a framework to a system if that framework can be applied to the system. In particular, there is no approximation on the channel decoding part in this case. On the other hand, a dashed arrow means approximations need to be made on the channel decoding part. No arrows, solid or dashed, between a framework and a system means the framework cannot be applied to the system. The next two paragraphs elaborates Fig. 4.1 further.

Since the EM-BP framework can incorporate virtual channel decoding, the channel decoding part of EM-BP for time-domain PNC

is optimal. A slight modification from virtual channel decoding to PIC channel decoding (as discussed in Chapter 2) allows EM-BP to be applied to time-domain MUD systems also. This modification introduces approximations on the decoded APPs used in EM-BP.

A shortcoming of the EM-BP framework is that it cannot be applied to OFDM PNC/MUD systems, because of the new problems caused by CFOs in the frequency-domain systems. The SAGE-ECM framework can deal with the CFOs in multiuser OFDM systems, thus it can be applied to OFDM MUD systems. Since the CFO estimation procedure in SAGE-ECM actually operates in the time-domain, it also can be directly applied to time-domain MUD systems for estimating CFOs. However, since SAGE-ECM is not compatible with virtual channel decoding, its application to time-domain/OFDM PNC systems will compromise the optimality of PNC channel decoding (as explained in the next section).

4.2 Future Work

4.2.1 Joint Channel-Parameter Estimation, CFO Compensation and Channel Decoding for OFDM PNC

One question for our future study is how to solve the problem of joint channel-parameter estimation, CFO compensation and channel decoding for OFDM PNC systems. A quick solution can be immediately obtained from the SAGE-ECM framework for OFDM-IDMA proposed in Chapter 3. We can directly apply the SAGE-

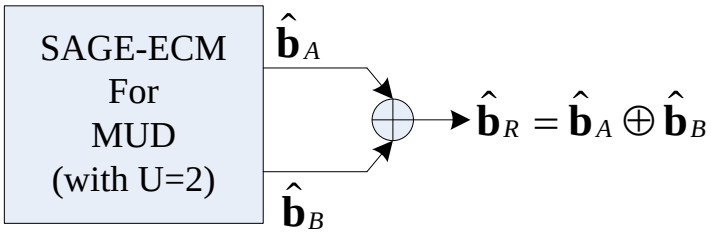


Figure 4.2: The illustration for MUD-XOR.

ECM framework of Chapter 3 to OFDM PNC; after SAGE-ECM has conveged, we can get the final decoding results for two source messages $\widehat{\mathbf{b}}_A$, $\widehat{\mathbf{b}}_B$; then, we perform bitwise XOR for network coding as $\widehat{\mathbf{b}}_R = \widehat{\mathbf{b}}_A \oplus \widehat{\mathbf{b}}_B$. The above scheme for PNC mapping is termed MUD-XOR [8] and illustrated in Fig. 4.2. Obviously, the application of SAGE-ECM to solve the problem of joint channel-parameter estimation, CFO compensation and channel decoding for OFDM PNC systems is feasible; however, the disadvantage of the application is that MUD-XOR exhibits subpar performance compared to virtual channel decoding [8].

For performance improvement, we are eager to replace the MUD-XOR with virtual channel decoding in the framework for OFDM PNC. However, there are two major obstacles for doing this: (i) In the E-step of ECM, it is hard to transform the APPs of symbol pairs (provided by virtual channel decoding) from the frequency domain to the time domain, even with Gaussian message passing; (ii) In the M-step of ECM, the closed-form solutions for the channel-parameter updates cannot be obtained (In Appendix D, we have to update the phase drifts for OFDM PNC by a exhaustive search method. The situations

will be the same for updating channel gians and CFOs). These two obstacles impose a high complexity on the framework operated with virtual channel decoding. How to enable virtual channel decoding in OFDM PNC systems for joint channel-parameter estimation, CFO compensation and channel decoding is an important future research topic.

4.2.2 Soft IDFT without Gaussian Approximation

In Chapter 3, we solved the "soft IDFT" problem using the Gaussian message passing. Chapter 3 showed that with Gaussian approximation, we just need to pass the means of the decoded symbols from the frequency domain to the time domain. This indeed greatly reduces the complexity involved in the problem. However, we do not know how large the penalty is by such an approximation. Can we do better than that? We believe that there are many applications that require the function of soft IDFT when we need to perform IDFT/DFT on a series of probability functions rather than deterministic complex numbers.

In general, without the Gaussian assumption, we can solve the soft IDFT problem on a factor graph using the sum-product message passing algorithm. We first construct the factor graph by considering the butterfly graph of IFFT. We show a butterfly graph with the DFT size $N = 4$ in Fig. 4.3 as an example. The IDFT is performed to transform the frequency domain samples $\{Y_i\}_{i=0}^{N-1}$ to the time-domain samples $\{y_i\}_{i=0}^{N-1}$. We do not know the exact values of $\{Y_i\}_{i=0}^{N-1}$;

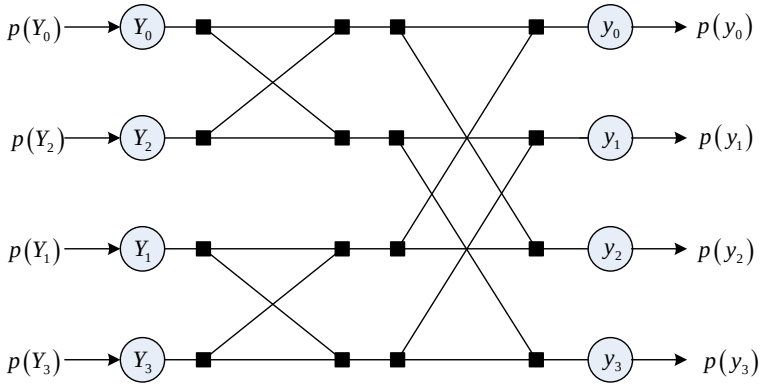


Figure 4.3: The factor graph for IDFT with $N = 4$.

we just can obtain the probability functions of $\{Y_i\}_{i=0}^{N-1}$. With the probability functions of the frequency-domain samples $\{p(Y_i)\}_{i=0}^{N-1}$ as the input messages, we then use sum-product rule on the factor graph to compute the output messages $\{p(y_i)\}_{i=0}^{N-1}$ that are the probability functions of the time domain samples. However, the alphabet size of messages on the graph increases in an exponential order as we progress from the input to the output of the IDFT factor graph. For example, even if $\{Y_i\}_{i=0}^{N-1}$ are binary numbers, each y_i will have 2^N values. We will treat this general soft IDFT problem in our future study.

□ End of chapter.

Appendix A

An Interpretation of EM Algorithm

In this appendix, we use the argument of Kullback-Leibler (KL) divergence to interpret the physical meaning of EM algorithm. Specifically, we shall see that the iteration expressed by (2.5) will eventually at least converge to a local optimum and possibly to a global optimum with respect to the target $\arg\max_{\mathbf{h}} \log p\left(\mathbf{h}\left|\mathbf{y}\right., C^2\right)$. The objective function being optimized in the EM algorithm can be interpreted as one in which an additional KL divergence term, $-D_{KL}$, has been added to the original objective $\log p\left(\mathbf{h}\left|\mathbf{y}\right., C^2\right)$. This additional $-D_{KL}$ term is guaranteed to converge to zero in the EM algorithm, hence the two objectives are consistent.

Proposition 1: Define a function of two variables $\mathbf{h}$ and $\mathbf{h}'$ as follows: $f\left(\mathbf{h}, \mathbf{h}'\right) \triangleq -D_{KL}\left(p\left(\mathbf{x}\left|\mathbf{y}, \mathbf{h}', C^2\right.\right)\middle\| p\left(\mathbf{x}\left|\mathbf{y}, \mathbf{h}, C^2\right.\right)\right) + \log p\left(\mathbf{h}\left|\mathbf{y}, C^2\right.\right)$. Furthermore, let

$$\mathbf{h}^* \triangleq \arg\max_{\mathbf{h}} \log p\left(\mathbf{h}\left|\mathbf{y}, C^2\right.\right)$$

and

$$\left(\mathbf{h}_f^*, \mathbf{h}_f'^*\right) \triangleq \arg\max_{\mathbf{h},\mathbf{h}'} f\left(\mathbf{h}, \mathbf{h}'\right).$$

Then,

$$\max_{\mathbf{h},\mathbf{h}'} f\left(\mathbf{h},\mathbf{h}'\right) = \max_{\mathbf{h}} \log p\left(\mathbf{h}\left|\mathbf{y},C^2\right.\right)$$

and

$$\mathbf{h}_f^* = \mathbf{h}_f'^* = \mathbf{h}^*.$$

Proof: It is known that $D_{KL}\left(p\left(\mathbf{x}\left|\mathbf{y},\mathbf{h}',C^2\right.\right)\middle\|\, p\left(\mathbf{x}\left|\mathbf{y},\mathbf{h},C^2\right.\right)\right) \geq 0$ for any tuple $(\mathbf{h},\mathbf{h}')$ [46]. Thus, we have

$$\begin{aligned} f\left(\mathbf{h},\mathbf{h}'\right) \leq \log p\left(\mathbf{h}\left|\mathbf{y},C^2\right.\right) &\leq \max_{\mathbf{h}} \log p\left(\mathbf{h}\left|\mathbf{y},C^2\right.\right) \\ &= \log p\left(\mathbf{h}^*\left|\mathbf{y},C^2\right.\right) \end{aligned}$$

for any duple $(\mathbf{h},\mathbf{h}')$. In particular,

$$f\left(\mathbf{h}_f^*,\mathbf{h}_f'^*\right) \leq \log p\left(\mathbf{h}^*\left|\mathbf{y},C^2\right.\right)$$

We also have that $D_{KL}\left(p\left(\mathbf{x}\left|\mathbf{y},\mathbf{h}^*,C^2\right.\right)\middle\|\, p\left(\mathbf{x}\left|\mathbf{y},\mathbf{h}^*,C^2\right.\right)\right) = 0$ [46]. Thus, setting $\mathbf{h}_f^* = \mathbf{h}_f'^* = \mathbf{h}^*$ gives us

$$f\left(\mathbf{h}_f^*,\mathbf{h}_f'^*\right) = \log p\left(\mathbf{h}^*\left|\mathbf{y},C^2\right.\right) \qquad \text{Q.E.D.}$$

With Proposition 1, algorithm (2.5) can be interpreted as trying to find $\mathbf{h}_f^* = \mathbf{h}_f'^* = \mathbf{h}^*$ that maximize $f\left(\mathbf{h},\mathbf{h}'\right)$. We can think about the algorithm in the following way. Since we know that the two arguments $\mathbf{h}$ and $\mathbf{h}'$ must be equal at the optimal, we could start out with a guess of $\mathbf{h} = \mathbf{h}' = \mathbf{h}^{(0)}$. This gives us an initial $f\left(\mathbf{h},\mathbf{h}'\right) = f\left(\mathbf{h}^{(0)},\mathbf{h}^{(0)}\right) = \log p\left(\mathbf{h}^{(0)}\left|\mathbf{y},C^2\right.\right)$. However, this may not be the optimal $\log p\left(\mathbf{h}^*\left|\mathbf{y},C^2\right.\right)$ even though the associated KL divergence is 0.

In the next iteration, we want to know whether we can change $\mathbf{h}$ to a different value, say $\mathbf{h} = \mathbf{h}^{(1)}$, and obtain a better $f\left(\mathbf{h}^{(1)},\mathbf{h}^{(0)}\right) >$

$f\left(\mathbf{h}^{(0)}, \mathbf{h}^{(0)}\right)$. This is exactly what (2.5) attempts to do. Notice that if such $f\left(\mathbf{h}^{(1)}, \mathbf{h}^{(0)}\right)$ can be found, then it is guaranteed that

$$\log p\left(\mathbf{h}^{(1)} \left| \mathbf{y}, C^2\right.\right) \geq f\left(\mathbf{h}^{(1)}, \mathbf{h}^{(0)}\right) > f\left(\mathbf{h}^{(0)}, \mathbf{h}^{(0)}\right) = \log p\left(\mathbf{h}^{(0)} \left| \mathbf{y}, C^2\right.\right).$$

Therefore,

$$f\left(\mathbf{h}^{(1)}, \mathbf{h}^{(1)}\right) = \log p\left(\mathbf{h}^{(1)} \left| \mathbf{y}, C^2\right.\right) \geq f\left(\mathbf{h}^{(1)}, \mathbf{h}^{(0)}\right) > f\left(\mathbf{h}^{(0)}, \mathbf{h}^{(0)}\right).$$

Thus, we see that (2.5) is an algorithm to successively find a better $\mathbf{h} = \mathbf{h}' = \mathbf{h}^{(k)}$ for substitution into $f\left(\mathbf{h}, \mathbf{h}'\right)$ until things converge. Note in particular that $f\left(\mathbf{h}^{(k+1)}, \mathbf{h}^{(k+1)}\right) \geq f\left(\mathbf{h}^{(k)}, \mathbf{h}^{(k)}\right)$ for all k by similar argument as above. Since $f\left(\mathbf{h}^{(k)}, \mathbf{h}^{(k)}\right)$ is upper bounded by $\log p\left(\mathbf{h}^{*} \left| \mathbf{y}, C^2\right.\right)$, it cannot increase indefinitely and convergence is guaranteed.

However, like all other 'peak seeking' algorithms, the ultimate point to which EM converges may or may not be the global peak $\log p\left(\mathbf{h}^{*} \left| \mathbf{y}, C^2\right.\right)$ if there are local optimal points. Therefore, for global optimum, EM usually requires a good initial point, which can be achieved using pilot symbols in our problem.

□ **End of chapter.**

Appendix B

Derivation of (3.24)

We first expand the sample-wise Q function in (3.16) as

$$
\begin{aligned}
& Q_{m,i}\left(\Omega_u \left| \widehat{\Omega}_u^{(k,z-1)} \right.\right) \\
&= \sum_{y_{u,m,i}} \log p\left(\widehat{r}_{u,m,i}^{(k)} \left| \Omega_u, y_{u,m,i} \right.\right) p\left(y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \\
&= -\sum_{y_{u,m,i}} \left\| \widehat{r}_{u,m,i}^{(k)} - e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} y_{u,m,i} \right\|^2 p\left(y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right) \\
&= -\sum_{y_{u,m,i}} \left(\left\| \widehat{r}_{u,m,i}^{(k)} \right\|^2 - 2R\left\{ \left(\widehat{r}_{u,m,i}^{(k)} \right)^H e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} y_{u,m,i} \right\} + \right. \\
&\quad \left. \|y_{u,m,i}\|^2 \right) p\left(y_{u,m,i} \left| \widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u \right.\right)
\end{aligned}
\tag{B.1}
$$

where we have already dropped the constant term. Then, we can easily compute the following expectations

$$\begin{aligned}
&\sum_{y_{u,m,i}} \left\|\widehat{r}_{u,m,i}^{(k)}\right\|^2 p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right) \\
&= \left\|\widehat{r}_{u,m,i}^{(k)}\right\|^2 Re\left\{\left(\widehat{r}_{u,m,i}^{(k)}\right)^H e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} m_{y_{u,m,i}}\right\} \\
&\sum_{y_{u,m,i}} Re\left\{\left(\widehat{r}_{u,m,i}^{(k)}\right)^H e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} y_{u,m,i}\right\} p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right) \\
&= Re\left\{\left(\widehat{r}_{u,m,i}^{(k)}\right)^H e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u(i-1)/N} m_{y_{u,m,i}}\right\} \\
&\sum_{y_{u,m,i}} \|y_{u,m,i}\|^2 p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right) = \left\|m_{y_{u,m,i}}\right\|^2 + \sigma^2_{y_{u,m,i}}
\end{aligned} \tag{B.2}$$

where the mean and variance of $y_{u,m,i}$ are given in the Gaussian expression for $p\left(y_{u,m,i}\left|\widehat{\mathbf{r}}_u^{(k)}, \widehat{\Omega}_u^{(k,z-1)}, C_u\right.\right)$ in (3.21). Substituting (B.2) in to (B.1), we have

$$\begin{aligned}
&Q_{m,i}\left(\Omega_u\left|\widehat{\Omega}_u^{(k,z-1)}\right.\right) = \\
&\underbrace{\left\|\widehat{r}_{u,m,i}^{(k)}\right\|^2 + 2R\left\{\left(\widehat{r}_{u,m,i}^{(k)}\right)^H e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u i/N} m_{y_{u,m,i}}\right\} - \left\|m_{y_{u,m,i}}\right\|^2}_{-\left\|\widehat{r}_{u,m,i}^{(k)} - e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u i/N} m_{y_{u,m,i}}\right\|^2} - \sigma^2_{y_{u,m,i}} \\
&= -\left\|\widehat{r}_{u,m,i}^{(k)} - e^{j\theta_{u,m}} e^{j2\pi\varepsilon_u i/N} m_{y_{u,m,i}}\right\|^2 - \sigma^2_{y_{u,m,i}}
\end{aligned} \tag{B.3}$$

Since $\sigma^2_{y_{u,m,i}}$ is not relevant to Ω_u. We just drop it. This gives the ultimate form of Q function in (3.24).

□ **End of chapter.**

Appendix C

Derivation of (3.30)

We denote the objective function in (3.29) by

$$f\left(\varepsilon_u\right) \triangleq -\sum_{m=1}^{M} \left\| \widehat{\mathbf{r}}_{u,m}^{(k)} - e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right\|^2 \tag{C.1}$$

Expanding and dropping the terms irrelevant to ε_u, we obtain

$$f\left(\varepsilon_u\right) = \sum_{m=1}^{M} Re\left\{ \left(\widehat{\mathbf{r}}_{u,m}^{(k)}\right)^H e^{j\widehat{\theta}_{u,m}^{(k,z-1)}} \mathbf{\Gamma}\left(\varepsilon_u\right) \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F} \widehat{\mathbf{h}}_u^{(k,z-1)} \right\} \tag{C.2}$$

We choose to approximate $\mathbf{\Gamma}\left(\varepsilon_u\right)$ in the objective function using its Taylor series expansion

$$\begin{aligned} &\mathbf{\Gamma}\left(\varepsilon_u\right) \approx \mathbf{\Gamma}\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) \\ &+ \left(\varepsilon_u - \widehat{\varepsilon}_u^{(k,z-1)}\right) \mathbf{\Gamma}'\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) + \tfrac{1}{2}\left(\varepsilon_u - \widehat{\varepsilon}_u^{(k,z-1)}\right)^2 \mathbf{\Gamma}''\left(\widehat{\varepsilon}_u^{(k,z-1)}\right) \end{aligned} \tag{C.3}$$

where we truncate the third-order terms and above and $\widehat{\varepsilon}_u^{(k,z-1)}$ is the starting point. Substituting (C.3) into (C.2) and differentiating the

resulting $\widehat{\varepsilon}_u^{(k,z-1)}$ with respect to ε_u yield

$$\frac{\partial f(\varepsilon_u)}{\partial \varepsilon_u} = \sum_{m=1}^{M} \mathrm{Re}\left\{ \left(\widehat{\mathbf{r}}_{u,m}^{(k)}\right)^H e^{j\hat{\theta}_{u,m}^{(k,z-1)}} \left[\mathbf{\Gamma}'\left(\hat{\varepsilon}_{\mathbf{u}}^{(\mathbf{k},\mathbf{z-1})}\right) + \left(\varepsilon_u - \hat{\varepsilon}_u^{(k,z-1)}\right)\mathbf{\Gamma}''\left(\hat{\varepsilon}_{\mathbf{u}}^{(\mathbf{k},\mathbf{z-1})}\right)\right] \mathbf{F}^H \mathbf{D}\left(\mathbf{m}_{\mathbf{X}_{u,m}}\right) \mathbf{F}\widehat{\mathbf{h}}_u^{(k,z-1)} \right\}$$

Finally, setting $\partial f\left(\varepsilon_u\right)/\partial \varepsilon_u = 0$ and solving the equation, we obtain the new CFO update shown in (3.30).

□ **End of chapter.**

Appendix D

Joint Phase Tracking and Channel Decoding for OFDM PNC: Algorithm and Experimental Evaluation

Applying the EM-BP algorithm proposed in Chapter 2, this appendix investigates the problem of joint phase tracking and channel decoding in OFDM based Physical-layer Network Coding (PNC) systems.

OFDM signaling can obviate the need for tight time synchronization among multiple simultaneous transmissions in the uplink of PNC systems. However, OFDM PNC systems are susceptible to phase drifts caused by residual carrier frequency offsets (CFOs). In the traditional OFDM system in which a receiver receives from only one transmitter, pilot tones are employed to aid phase tracking. In OFDM PNC systems, multiple transmitters transmit to a receiver, and these pilot tones are shared among the multiple transmitters. This reduces

the number of pilots that can be used by each transmitting node. Phase tracking in OFDM PNC is more challenging as a result. To overcome the degradation due to the reduced number of per-node pilots, this appendix supplements the pilots with the channel information contained in the data. In particular, we propose to solve the problems of phase tracking and channel decoding jointly. Our solution consists of the use of the expectation-maximization (EM) algorithm for phase tracking and the use of the belief propagation (BP) algorithm for channel decoding. The two problems are solved jointly through iterative processing between the EM and BP algorithms. Simulations and real experiments based on software-defined radio (SDR) show that the proposed method can improve phase tracking as well as channel decoding performance.

D.1 Introduction

Relays can be employed to extend coverage, enhance reliability and increase throughput in wireless networks. Recently, the research community has shown growing interest in a simple relay network in which two terminal nodes communicate through a relay. This network is referred to as the two-way relay channel (TWRC).

Physical-layer Network Coding (PNC), originally proposed in [4], can potentially boost the throughput in TWRC by 100% compared with the traditional multi-hop relaying method [8]. In TWRC operated with PNC, the two terminal nodes first transmit their messages simultaneously to the relay. The relay then maps the

overlapped signals into a network-coded message (e.g., bit-wise XOR of the messages of the terminal nodes) and broadcasts the network-coded message to the two terminal nodes. Each terminal then extracts the message of the other terminal by subtracting its own message from the network-coded message. Thus, the two terminal nodes exchange one message with each other in two time slots. With traditional relaying, four time slots are needed [8].

Although PNC has the potential to boost throughput in TWRC, there are new challenges for PNC. An important issue is how to perform PNC mapping at the relay when the two signals arrive with symbol and phase asynchronies. In [22], the authors devised a belief propagation (BP) [17, 19] method to decode network-coded messages in asynchronous PNC systems. Single-carrier time-domain signals are assumed. Another solution for asynchronous PNC is to use OFDM signals [81]. OFDM carries multiple data streams on multiple subcarriers in a parallel manner, and the symbol duration within each subcarrier is lengthened compared to that in the single-carrier system. To deal with the effect of multipath, cyclic-prefix (CP) is prepended at the beginning of each OFDM symbol. We designed and implemented an OFDM PNC system using software-defined radio (SDR) in [15,70]. If the "symbol delay spread" between the two terminals is within the CP, the time-domain misaligned samples will become aligned in the frequency-domain after DFT demodulation. This obviates the need for strict synchronization in PNC systems. Benefiting from the CP and the larger symbol duration, we can perform PNC mapping one-by-one on each subcarrier in a manner similar to that in synchronous

PNC.

One drawback of OFDM systems is their sensitivity to carrier frequency offset (CFO) between the crystal oscillators of the transmitter and receiver. CFO causes two negative effects: (i) drifting of the phase of the channel coefficient of each subcarrier; (ii) inter-carrier interferences (ICI) among different subcarriers. In systems such as 802.11 WLAN, we can estimate the CFO using preambles, and then compensate for the CFO for the whole packet. However, estimation error may leave behind an uncompensated residual CFO. Since the residual CFO in 802.11 systems are typically small, the CFO-induced ICI is usually negligible. We can treat the ICI as an additional noise which degrades the effective received signal-to-noise ratio (SNR) slightly [60]. However, even a small residual CFO can lead to large phase drifts which accumulate over time. Data decoding will fail if we ignore the phase drifts. It is therefore important to track phase drifts and compensate for them before data decoding.

For WLAN, it is common to track phase using pilot tones in the OFDM symbols. This appendix considers the more challenging phase tracking problem in OFDM PNC systems. The pilot tones are now shared by the two terminal nodes. This reduces the number of pilots that can be used by each node. Moreover, the superimposed constellation at the relay receiver is denser than that of a single user receiver.

In this appendix, we propose to tackle the problem of phase tracking jointly with channel coding in an iterative manner. In particular, we use the framework of expectation-maximization and

belief propagation (EM-BP) [82] to perform joint phase tracking and channel decoding. The BP algorithm performs channel decoding using the phase-adjusted CSI estimated by the EM algorithm, while the EM algorithm performs phase tracking using the pilot symbols as well as the soft information of data symbols provided by the BP algorithm.

Since the proposed EM-BP algorithm makes use of both the data symbols and pilot symbols for phase tracking, we can potentially improve the accuracy of phase tracking compared with the traditional pilot-based algorithm. We conducted simulations and experiments to evaluate the proposed EM-BP algorithm. Our experiments were based on the data collected from an OFDM PNC prototype [15, 70]. The experimental results show that the EM-BP algorithm can yield 2-3 dB gain in BER performance compared with the traditional method that performs pilot-based phase tracking and channel decoding in a disjoint manner.

The remainder of this appendix is organized as follows: Section D.2 describes the OFDM PNC system. Section D.3 puts forth our proposed EM-BP algorithm for OFDM PNC. Section D.4 presents our simulation and experimental results.

D.2 System Model

D.2.1 Transmit Signal

Fig. D.1 shows our OFDM PNC system model. In time slot 1, terminal nodes A and B simultaneously transmit signals to the relay

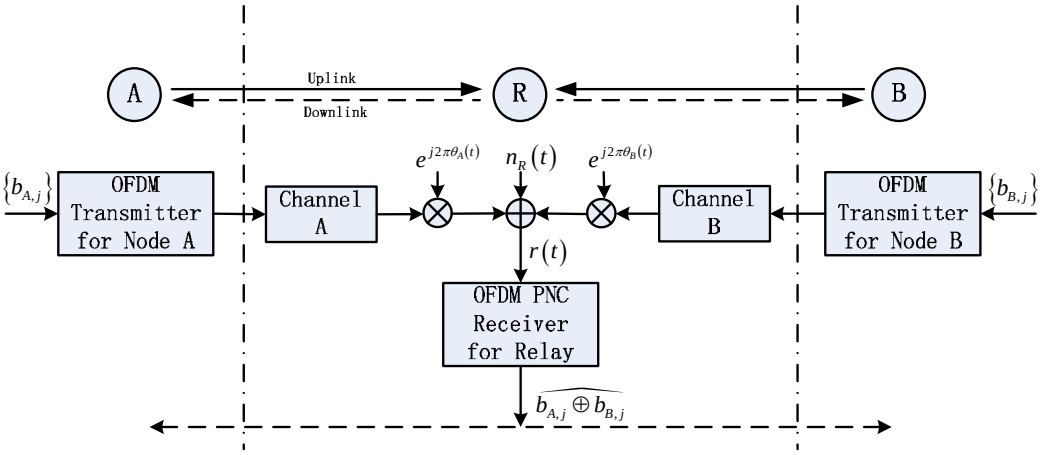


Figure D.1: The system model of OFDM PNC system.

R. We assume that the transmitters of nodes A and B employ the frame format proposed in [15], in which the preambles of A and B do not overlap, but the payloads overlap, in time. One frame contains M OFDM symbols for the payload. Let N be the number of subcarriers. The length-N vector $\mathbf{X}_{u,m} = [X_{u,m,0}X_{u,m,1} \cdots X_{u,m,N-1}]^T$ is the m^{th} frequency-domain symbol transmitted by node $u \in \{A, B\}$. Within the N subcarriers, each OFDM symbol has N_d data tones, N_p pilot tones and N_z zero tones ($N = N_d + N_p + N_z$). The symbols on data tones are obtained after channel encoding, interleaving and constellation mapping. In this appendix, we assume that nodes A and B use the same channel encoder (the valid set of codewords is denoted by C), interleaver and constellation when mapping their source information bits $\{b_{A,j}\}$, $\{b_{B,j}\}$ to the transmitted data symbols. The pilots assist the task of phase tracking. We adopt a pilot pattern similar to that employed in [15], where separate pilot tones in the frequency domain are used by the two end nodes. The zero tones

serve as the guard bands. The OFDM modulation is implemented with an N point IDFT: $\mathbf{x}_{u,m} = IDFT(\mathbf{X}_{u,m})$, where $\mathbf{x}_{u,m} = [x_{u,m,0}x_{u,m,1}\cdots x_{u,m,N-1}]^T$ is the vector of time-domain samples. The N time-domain samples of one OFDM symbol is preceded by a CP. The length of the CP is N_{cp}. Therefore, each OFDM symbol corresponds to $N_s = N_{cp} + N$ time-domain samples, where $x_{u,m,i} = x_{u,m,i+N}$ for $i \in [-N_{cp}, -1]$. After pulse shaping and digital-to-analogy conversion (DAC), the N_s time-domain samples of the m^{th} OFDM symbol are converted into analog waveforms:

$$s_{u,m}(t) = \sum_{i=-N_{cp}}^{N-1} x_{u,m,i}\varphi(t - iT_s) \tag{D.1}$$

where $\varphi(t)$ is the shaping pulse for ensuring the transmitted signal satisfy the spectrum requirement, and T_s is the sampling interval (The OFDM symbol duration is therefore$T = N_sT_s$).

The relay broadcasts a beacon to coordinate the uplink transmissions [70]. The beacon accomplishes two functions: (i) First, it triggers simultaneous transmissions by the two terminal nodes to meet a loose time synchronization requirement (we will elaborate on this later). (ii) Second, it allows the two terminal nodes to estimate CFO and compensate for it by precoding on the transmitter side in the uplink. Specifically, each node estimates the CFO between itself and the relay based on the received beacon signal; and then multiplies its signal with a compensating phase term before transmission [70]. In this way, we reduce the CFO to a small residual CFO.

D.2.2 Receive Signal

The received signal waveform at the relay can be expressed as

$$r(t) = \sum_{u\in\{A,B\}} \sum_{m=0}^{M-1} s_{u,m}(t - mT) \otimes h_u(t)\, e^{j\theta_u(t)} + n_R(t) \qquad \text{(D.2)}$$

where $\otimes$ is the convolution operator, $h_u(t)$ is the frequency-selective channel between node u and the relay, $\theta_u(t)$ is the phase drift between the relay and node u caused by the residual CFO, $n_R(t)$ is the thermal noise from the receiver frontend modeled as a zero-mean Gaussian random process with variance σ_n^2.

We define the delay spread of a multipath channel as the delay difference between its longest path and shortest path, and denote the delay spread of the channel of node u by τ_u. Without loss of generality, we assume the signal of the first path of node A arrives earlier than that of node B. Let $\tau > 0$ denote the delay difference between the first paths of the two nodes. It can be shown [15] that as long as the delay-spread-within-CP requirement, specified as $\max\{\tau_A, \tau + \tau_B\} \leq N_{cp}T_s$, is satisfied, the time-domain symbol misalignment does not cause any negative effect on the carrier-by-carrier PNC mapping in the frequency-domain. Note that while τ_A is the delay spread of channel A and τ_B is the delay spread of channel B, $\max\{\tau_A, \tau + \tau_B\} \leq N_{cp}T_s$ can be viewed as the aggregate delay spread of the two channels. This delay-spread-within-CP requirement can be regarded as a loose time synchronization requirement and can be achieved using beacon triggering [15, 70] of simultaneous transmissions by nodes A and B. Therefore, with OFDM signaling,

the time asynchrony in PNC systems is not a critical issue. However, the above conclusion is made under the assumption that there is no phase drift in the system. The presence of phase drift will degrade the system performance.

After ADC and removal of CP, the N discrete samples of the m^{th} received OFDM symbol are collected into a vector $\mathbf{r}_m = [r_{m,0} r_{m,1} \cdots r_{m,N-1}]^T$. The frequency-domain sample vector of the m^{th} OFDM symbol, $\mathbf{R}_m = [R_{m,0} R_{m,1} \cdots R_{m,N-1}]^T$, is obtained by performing DFT on $\mathbf{r}_m$. We can express $\mathbf{R}_m$ as

$$\begin{aligned} \mathbf{R}_m &= \sum_{u\in\{A,B\}} e^{j\Theta_{u,m}} \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{H}_u + \underbrace{\sum_{u\in\{A,B\}} \mathbf{\Lambda}_{u,m} + \mathbf{N}_{R,m}}_{\triangleq \mathbf{W}_{R,m}} \\ &= \sum_{u\in\{A,B\}} e^{j\Theta_{u,m}} \mathbf{D}\left(\mathbf{X}_{u,m}\right) \mathbf{H}_u + \mathbf{W}_{R,m} \end{aligned} \tag{D.3}$$

where $\Theta_{u,m}$ is the phase drift in the m^{th} frequency-domain OFDM symbol of node u; $\mathbf{D}\left(\mathbf{X}_{u,m}\right) = diag\left(\mathbf{X}_{u,m}\right)$ is the diagonal matrix with transmitted symbols $\mathbf{X}_{u,m}$ as its diagonal elements; $\mathbf{H}_u = [H_{u,0} H_{u,1} \cdots \ H_{u,N-1}]^T$ is the frequency response of the channel between node u and the relay, $\mathbf{\Lambda}_{u,m}$ is the ICI component from node u, and $\mathbf{N}_{R,m}$ is the frequency-domain noise. In (D.3), $\mathbf{W}_{R,m}$ is the ICI plus noise. In this appendix, we approximate $\mathbf{W}_{R,m}$ as a circularly symmetric white Gaussian noise with zero mean and covariance matrix $\sigma_W^2 \mathbf{I}_N$. Based on (D.3), we can apply the EM-BP algorithm to do joint phase tracking and channel decoding.

D.3 EM-BP for OFDM PNC

In this section, we derive the EM-BP algorithm for joint phase tracking and channel decoding in OFDM PNC systems. Define the pair of phase drifts on the m^{th} OFDM symbol as $\Theta_m \triangleq [\Theta_{A,m}\Theta_{B,m}]^T$; the vector containing the phase drifts over all OFDM symbols as $\mathbf{E} = \left[\Theta_0^T\Theta_1^T\cdots\Theta_{M-1}^T\right]^T$; the pair of data symbols on the i^{th} subcarrier of the m^{th} OFDM symbol as $\mathbf{X}_{m,i} \triangleq [X_{A,m,i}X_{B,m,i}]^T$; and the vector containing all the received frequency-domain OFDM samples as $\mathbf{R} \triangleq \left[\mathbf{R}_0^T\mathbf{R}_1^T\cdots\mathbf{R}_{M-1}^T\right]^T$. We assume that the channels $\{\mathbf{H}_u\}_{u=\{A,B\}}$ are already known at the relay, which can be achieved via estimation using the preamble symbols.

EM-BP is an iterative framework where the k^{th} iteration consists of a BP algorithm for channel decoding and an EM algorithm for phase tracking.

D.3.1 BP for virtual channel decoding

With the estimate of phase drift $\widehat{\mathbf{E}}^{(k-1)} = \left[\hat{\Theta}_0^{(k-1)T}\hat{\Theta}_1^{(k-1)T}\cdots\hat{\Theta}_{M-1}^{(k-1)T}\right]^T$ from the $k-1^{th}$ iteration, we perform BP for channel decoding to find $p\left(\mathbf{X}_{m,i}\middle|\mathbf{R},\widehat{\mathbf{E}}^{(k-1)},C^2\right)$ for all m and all i, where C^2 is the code constraint imposed by a 'virtual channel encoder' that takes the original information source symbols from nodes A and B $\{b_{A,j},b_{B,j}\}$ as inputs, and outputs $\{\mathbf{X}_{m,i}\}$ as coded symbols. The BP algorithm for virtual channel decoding applies the sum-product [17] rule on the factor graph that incorporates

the constraints imposed by virtual channel encoding, which models the simultaneous transmissions by the two terminal nodes [16, 21, 22]. The BP virtual channel decoding is initialized with the probabilities $p\left(R_{m,i}\left|\mathbf{X}_{m,i},\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right)$ for all m and all i as inputs. Here, $p\left(R_{m,i}\left|\mathbf{X}_{m,i},\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right)$ has a Gaussian form:

$$\begin{aligned} &p\left(R_{m,i}\left|\mathbf{X}_{m,i},\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right) \\ &\propto \exp\left(-\left\|R_{m,i}-\sum_{u\in\{A,B\}} e^{j\widehat{\Theta}_m^{(k-1)}}X_{u,m,i}H_{u,i}\right\|^2\Big/\sigma_W^2\right) \end{aligned} \tag{D.4}$$

The decoding results of the BP algorithm is then fed to the EM algorithm for phase tracking.

D.3.2 EM for Phase Tracking

We update the k^{th} estimate for the phase drifts $\widehat{\mathbf{E}}^{(k)}$ according to the EM algorithm consisting of an E-step and an M-step [24].

E-Step:

We compute the Q function of the phase drifts $\mathbf{E}$ given the last estimate $\widehat{\mathbf{E}}^{(k-1)}$:

$$\begin{aligned} &Q\left(\mathbf{E}\left|\widehat{\mathbf{E}}^{(k-1)}\right.\right) \\ &=\sum_{\mathbf{X}_A,\mathbf{X}_B}\log p\left(\mathbf{R}\left|\mathbf{X}_A,\mathbf{X}_B,\mathbf{E}\right.\right)p\left(\mathbf{X}_A,\mathbf{X}_B\left|\mathbf{R},\widehat{\mathbf{E}}^{(k-1)},C^2\right.\right) \\ &=\sum_m\sum_i\sum_{\mathbf{X}_{m,i}}\log p\left(R_{m,i}\left|\mathbf{X}_{m,i},\mathbf{E}\right.\right)p\left(\mathbf{X}_{m,i}\left|\mathbf{R},\widehat{\mathbf{E}}^{(k-1)},C^2\right.\right) \end{aligned} \tag{D.5}$$

where $\mathbf{X}_u \triangleq \left[\mathbf{X}_{u,0}^T\mathbf{X}_{u,1}^T\cdots\mathbf{X}_{u,M-1}^T\right]^T$ is the vector containing all the symbols transmitted by node u.

Note that in (D.5) the outputs from the BP channel decoding $\left\{p\left\{\mathbf{X}_{m,i}\ \middle|\ \mathbf{R}, \widehat{\mathbf{E}}^{(k-1)}, C^2\right)\right\}$ are used to assist EM phase tracking. To allow "decoupled" computation of (D.5), we make a simplification to ignore the relationship of the phase drift over symbols: that is, we assume $\mathbf{\Theta}_m$ and $\mathbf{\Theta}_{m'}$ are independent for $m \neq m'$. With this assumption, we can rewrite (D.5) as

$$Q\left(\mathbf{E}\left|\widehat{\mathbf{E}}^{(k-1)}\right.\right) = \sum_m Q_m\left(\mathbf{\Theta}_m\left|\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right) \tag{D.6}$$

where $Q_m(\cdot)$ is the symbol-wise Q function of the OFDM symbol:

$$\begin{aligned} &Q_m\left(\mathbf{\Theta}_m\left|\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right) \\ &= \sum_i \sum_{\mathbf{X}_{m,i}} \log p\left(R_{m,i}\left|\mathbf{X}_{m,i}, \mathbf{\Theta}_m\right.\right) p\left(\mathbf{X}_{m,i}\left|\mathbf{R}, \widehat{\mathbf{E}}^{(k-1)}, C^2\right.\right) \\ &\propto -\sum_i \sum_{\mathbf{X}_{m,i}} \left\| R_{m,i} - \sum_{u\in\{A,B\}} e^{j\Theta_{u,m}} X_{u,m,i} H_{u,i} \right\|^2 \times \\ &\qquad\qquad p\left(\mathbf{X}_{m,i}\left|\mathbf{R}, \widehat{\mathbf{E}}^{(k-1)}, C^2\right.\right) \end{aligned} \tag{D.7}$$

This decoupled expression for Q function can simplify the computation in the M-step of EM.

M-Step:

The objective of M-step is to find the variable that maximizes the Q function defined as in (D.5). Thanks to the simplification by (D.6), it is equivalent to finding

$$\widehat{\mathbf{\Theta}}_m^{(k)} = \arg\max_{\mathbf{\Theta}_m} Q_m\left(\mathbf{\Theta}_m\left|\widehat{\mathbf{\Theta}}_m^{(k-1)}\right.\right) \tag{D.8}$$

for each m. This is a symbol-by-symbol phase tracking process. To solve (D.8), we use a particle-filtering [83] type method to locate the argument that maximizes the Q function calculated in (D.7). We

use a list of samples (called particles) and the associated weights (the function values of these particles) to represent the Q function. According to the weights, we re-compute a new set of particles that are adaptively closer to the peak location, and then iterate. This method is proposed in [84] to enable a practical message passing algorithm for the phase estimation in a single-user single-carrier system. Here we modify it only slightly for our purpose. The pseudo-code for particle-filtering type method that solves (D.8) can be found in Algorithm 2, where P is the iteration number of particle-filtering, L^2 is the number of particles and ε is the forgetting factor. In our simulations and experiments in section D.4, we set $P = 4$, $L = 10$ and $\varepsilon = 0.1$.

Algorithm 2 Particle-filtering method for solving the M-step of EM

1: **Input:** $\mathbf{R}_m$, $\{\mathbf{H}_u\}$, $\left\{p\left(\mathbf{X}_{m,i}|\mathbf{R}, \widehat{\mathbf{E}}^{(k-1)}, C^2\right)\right\}$;
2: **Output:** $\widehat{\mathbf{\Theta}}_m^{(k)}$
3: initialize the list of L^2 samples as a $L \times L$ matrix $\mathbf{\Theta}^{(0)}$ whose $(p,q)^{th}$ element is given by $\mathbf{\Theta}_{p,q}^{(0)} = \begin{bmatrix} p2\pi/L & q2\pi/L \end{bmatrix}^T$ for $0 \le p, q \le L-1$;
4: **For** $l = 1$ to P **do**
5: compute the weights $\widetilde{\omega}_{p,q} = Q_m\left(\mathbf{\Theta}_{p,q}^{(l-1)} \middle| \widehat{\mathbf{\Theta}}_m^{(k-1)}\right)$, $\omega_{p,q} = \gamma\widetilde{\omega}_{p,q}$ for $0 \le p, q \le L-1$ where $\gamma^{-1} = \sum_p \sum_q \widetilde{\omega}_{p,q}$;
6: update the list of samples according to $\mathbf{\Theta}_{p,q}^{(l)} = (1-\varepsilon)\,\mathbf{\Theta}_{p,q}^{(l-1)} + \varepsilon\overline{\mathbf{\Theta}}^{(l-1)}$ for $0 \le p, q \le L-1$, where ε is a forgetting factor and $\overline{\mathbf{\Theta}}^{(l-1)} = \sum_p \sum_q \omega_{p,q}\mathbf{\Theta}_{p,q}^{(l-1)}$;
7: **End**
8: $\widehat{\mathbf{\Theta}}_m^{(k)} = \arg\max_{\mathbf{\Theta}_{p,q}^{(P)}} Q_m\left(\mathbf{\Theta}_{p,q}^{(P)} \middle| \widehat{\mathbf{\Theta}}_m^{(k-1)}\right)$;
9: **Return** $\widehat{\mathbf{\Theta}}_m^{(k)}$.

We carry out the above EM phase tracking symbol by symbol. After updating the phase drifts of all the OFDM symbols, we obtain $\widehat{\mathbf{E}}^{(k)} = \left[\widehat{\Theta}_0^{(k)T}\widehat{\Theta}_1^{(k)T}\cdots\widehat{\Theta}_{M-1}^{(k)T}\right]^T$ and then iterate to perform the next

BP channel decoding iteration.

D.3.3 Initialization and Termination of BP-EM iteration

The EM mechanism can usually find the maximum likelihood (ML) estimate of the phase upon convergence of the iterations. However, convergence is predicated on a good initial point for the EM iterations [47]. In this appendix, we use the least square (LS) estimation [77] of the phases from pilot tones in the m^{th} OFDM symbol as the initial point $\widehat{\mathbf{\Theta}}_m^{(0)}$ for all m. We denote the set that contains the indexes of pilot tones assigned to node u by P_u. Node u just transmits known symbols on the pilot tones indexed by P_u and null its signals on the other pilot tones. The initial phase estimates by LS pilot-based estimation are

$$\hat{\Theta}_{u,m}^{(0)} = \angle\left(\sum_{i\in P_u} X_{u,m,i}^{*} R_{m,i}\right) \tag{D.9}$$

for $u \in \{A, B\}$ and $m = 0, 1, \cdots, M-1$, where $\angle(\cdot)$ is the angle of a complex signal, $(\cdot)^*$ is the conjugate operator.

We repeat BP channel decoding and EM phase tracking iteratively. When the number of iterations reaches a preset maximum limit K, we terminate the BP-EM algorithm after obtaining the final phase estimate $\widehat{\mathbf{E}}^{(K)} = \left[\hat{\Theta}_0^{(K)T}\hat{\Theta}_1^{(K)T} \ \cdots \hat{\Theta}_{M-1}^{(K)T}\right]^T$. Substituting $\widehat{\mathbf{\Theta}}_m^{(K)}$ into (D.4) to replace $\widehat{\mathbf{\Theta}}_m^{(k-1)}$ for all m and all i, we carry out a final round of BP channel decoding to obtain $p\left(b_{A,j}, b_{B,j} \middle| \mathbf{R}, \widehat{\mathbf{E}}^{(K)}, C^2\right)$ for all j.

Then, the network-coded source message is obtained by

$$\begin{aligned} &\widehat{b_{A,j} \oplus b_{B,j}} \\ &= \arg\max_{b} \sum_{\substack{b_{A,j}, b_{B,j}: \\ b_{A,j} \oplus b_{B,j} = b}} p\left(b_{A,j}, b_{B,j} \left| \mathbf{R}, \widehat{\mathrm{E}}^{(K)}, C^2 \right.\right) \end{aligned} \tag{D.10}$$

After that, the relay channel-encodes the network-coded source message and broadcasts the channel-coded message to nodes A and B in the downlink phase, as illustrated in Fig. D.1

D.4 Simulations and Experimental Results

This section presents computer simulations and experimental results to evaluate the proposed algorithm. The frame format used is the one proposed in [15], a slightly modified version of 802.11 frame format. The DFT size is $N = 64$. The CP length is $N_{cp} = 16$. One OFDM symbol includes $N_d = 48$ data tones and $N_p = 4$ pilot tones. Each terminal node transmits known symbols on two of the four pilot tones, and nulls the signal on the other two pilot tones.

We adopt BPSK and QPSK modulations and the regular Repeat Accumulate (RA) channel code [49] with code rate 1/3. We adapt the virtual channel decoder for PNC developed in [22] for our purpose here (see section D.3). For each round of virtual channel decoding, we perform 20 BP iterations within the channel decoder. The signal-to-noise ratio (SNR) is defined as E_b/N_0, where E_b is the energy per source bit.

We evaluate the mean square error (MSE) of $e^{j\widehat{\Theta}_{u,m}}$: $E\left\|e^{j\hat{\Theta}_{u,m}} -\right.$

$e^{j\Theta_{u,m}}\Big\|^2$, and the BER of the network-coded messages. We investigate the performance of the proposed EM-BP method for joint phase tracking and channel decoding, benchmarked against traditional disjoint pilot-based phase tracking and channel decoding. The phase tracking of the traditional method employs the LS estimation given by (9).

D.4.1 Simulation Results

We investigate two channel models. The first is the flat Rayleigh fading channel. The second is a frequency-selective Rayleigh fading channel. We model the frequency-selective channel as a tapped delay line channel $h_u(t) = \sum_{l=0}^{L_u-1} \alpha_{u,l} \times \delta(t - lT_s)$, where L_u is the number of multipaths, $\alpha_{u,l}$ is the multipath gain. We assume that all the multipath gains are induced by independent Rayleigh fading, and that the power delay profile of the paths satisfies $E\left(|\alpha_{u,l}|^2\right) \propto \exp(-cl)$ and $\sum_{l=0}^{L_u-1} E\left(|\alpha_{u,l}|^2\right) = 1$, where c is the power decay factor. The power decay factor determines the envelope of the frequency-domain channel responses. The larger the power decay factor, the flatter the frequency-domain channel responses (as $c \to \infty$, the channel is reduced to flat fading). In our simulations, we change the value of c to investigate its impact on performance. We set the numbers of multi-paths for nodes A and B to $L_A = L_B = 4$. The channel gains are perfectly known in the simulations. For every pair of frames, the normalized CFOs that cause phase drifts on our signals are generated from a uniform distribution over the range $[-0.5\delta, 0.5\delta]$, where δ is

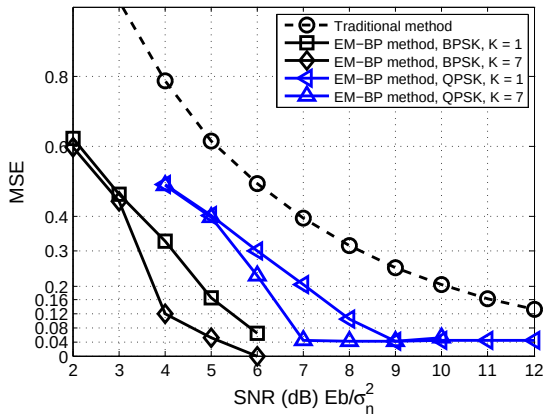


Figure D.2: The simulated MSE results of EM-BP in frequency-flat channel.

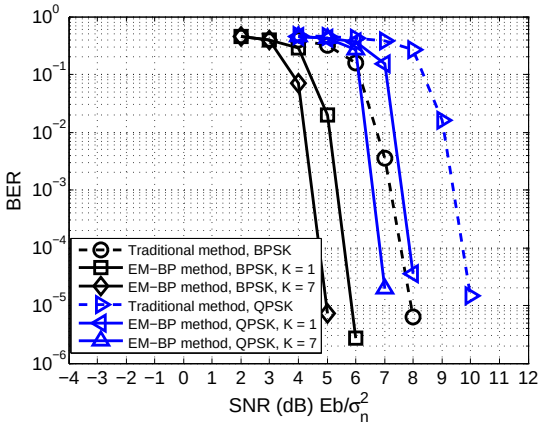


Figure D.3: The simulated BER results of EM-BP in frequency-flat channel.

the CFO attenuation factor. Since we focus on the residual CFO after precoding, we set a small value of $\delta = 0.1$.

Fig. D.2 presents the MSE results and Fig. D.3 presents the BER results under the flat fading channel. From the MSE results for the flat fading channel in Fig. D.2, we clearly see that the EM-BP algorithm gives more accurate channel estimation than the traditional pilot-based phase tracking. The estimation accuracy in EM-BP improves progressively with the number of iterations. As for the BER results of flat fading channel in Fig. D.3, we see that there is a 2 dB gain by EM-BP PNC just after the first EM iteration (K = 1). There is a 3dB gain after EM has converged (K = 7).

Fig. D.4 and Fig. D.5 respectively present the MSE results and the BER results of the frequency-selective channel with the power decay factor $c = 1$. We see that the performance trends for the frequency-selective channel are similar to that for the flat fading channel.

We also investigate the impact of the power decay factor. The

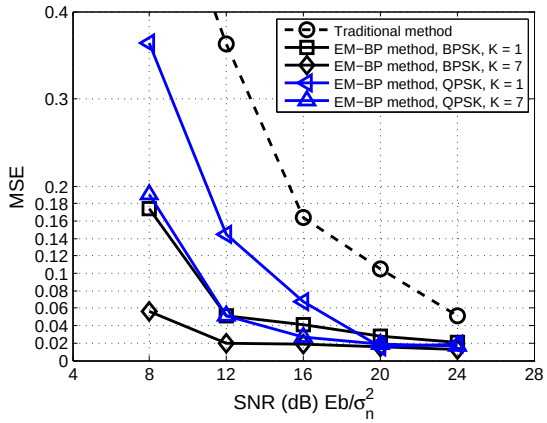


Figure D.4: The simulated MSE results of EM-BP in frequency-flat channel.

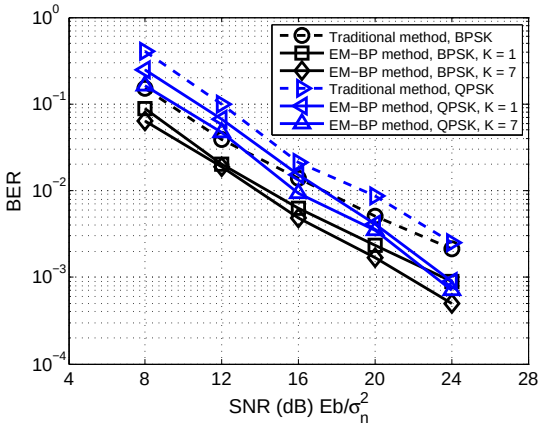


Figure D.5: The simulated BER results of EM-BP in frequency-selective channel.

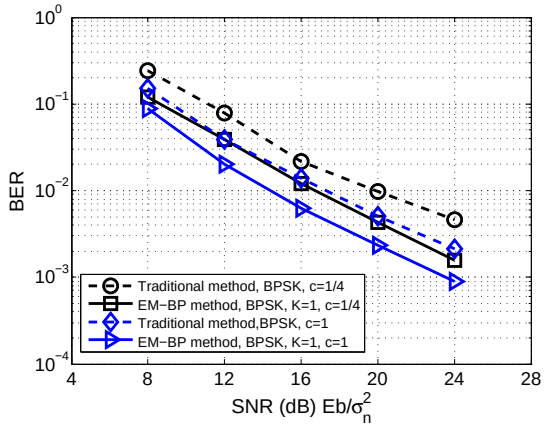


Figure D.6: The simulated BER results of EM-BP in frequency-selective channel.

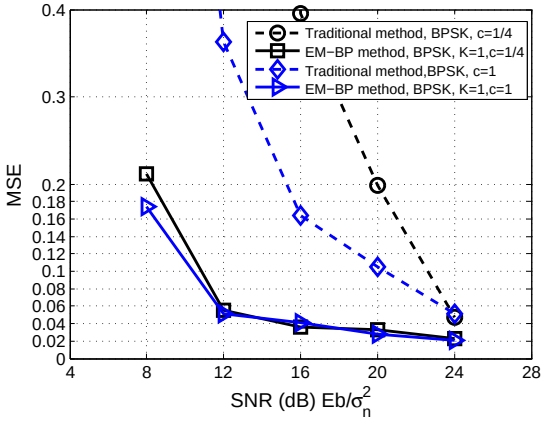


Figure D.7: The simulated MSE results of EM-BP in frequency-selective channel.

BER and MSE results in the frequency selective channel with different power decay factor ($c = 1/4$ and $c = 1$) are presented in Fig. D.6 and Fig. D.7, respectively. From Fig. D.6, we see that the BERs of the traditional method and the EM-BP method in the channel with larger c are better, since the channel now is flatter. Regardless of c, the gain in BER by EM-BP still holds. One interesting observation for the MSE results in Fig. D.7 is that the MSE of EM-BP is more robust against smaller c. With smaller c, the channel is more frequency-selective. If the frequency-domain channels on the pilot tones are in deep fades, the traditional pilot-based phase-tracking will not be accurate. Since the EM-BP method employs both data and pilots tones for phase-tacking, it is less affected by deep fades of pilot tones.

D.4.2 Experiment Results

Going beyond simulations, we also evaluate the performance of the proposed method experimentally. We use the data collected from a prototype of the OFDM PNC system [70]. The prototype is built on the USRP N210 hardware [79] and the GNU Radio software with the UHD hardware driver [80]. To emulate a TWRN system, we deployed three sets of USRP N210 with XCVR2450 boards [80] in our lab. The relay node R uses 802.11 channel 1 (2.412GHz) to poll the two end nodes to transmit together at channel 11 (2.462GHz). The system bandwidth is 4MHz. The use of 4 MHz bandwidth rather than the 20MHz full 802.11 bandwidth is due to the limitation of the USRP hardware.

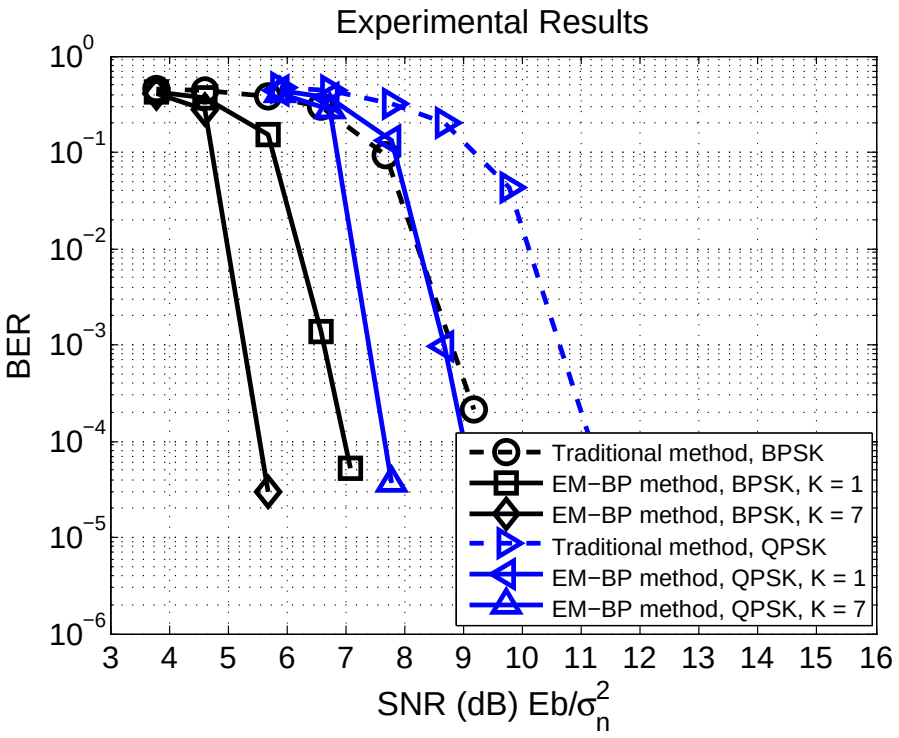


Figure D.8: The experimental BER results.

We performed controlled experiments for different SNRs. The receive powers of packets from nodes A and B at the relay were adjusted to be balanced (power imbalance within 1dB). The relay transmits 100 beacons to trigger 100 simultaneous uplink transmissions for each fixed SNR. After the terminal nodes received the beacon from the relay, they estimated the CFO from the beacon (beacons consist of 2 long training symbols as defined in 802.11 format). Then, they performed CFO precoding on the signals before the uplink transmissions. Finally, the relay received the simultaneous transmissions from the two nodes and converted it to digital data to be processed by the proposed method. In the experiments, the

channels between the relay and the two nodes were estimated using the orthogonal preambles of the packets [15, 70].

The experimental BER results are shown in Fig. D.8. In general, we observe similar performance trends as our simulation results: 2-3 dB gain by the EM-BP method. In particular, the BER performances of our experiments are closer to the flat-fading channel simulations results. The reason is that the bandwidth used in our experiments (4 MHz) is not large enough for frequency selectivity to come into play.

D.5 Conclusion

We have investigated the use of an EM-BP algorithm to solve the problems of phase tracking and channel decoding in OFDM PNC systems jointly. The main principle of our method is to use the soft information on the data produced by channel decoding to improve the performance of phase tracking, and to use the better phase tracking results to improve channel decoding, in an iterative manner. Our simulation and real experiment results show that the proposed method can obtain 2 dB gain after the first iteration, and around 3 dB gain after convergence of the iterations.

□ **End of chapter.**

Bibliography

[1] J. G. Proakis, "Digital Communications," *McGraw-Hill, New York*, 1995.

[2] S. Khurana, A. Kahol, and A. P. Jayasumana, "Effect of hidden terminals on the performance of IEEE 802.11 MAC protocol," in *Proc. IEEE LCN'98*, pp. 12–20, IEEE, 1998.

[3] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," in *Proc. ACM SIGCOMM'06*, pp. 39–50, IEEE, 2006.

[4] S. Zhang, S. Liew, and P. Lam, "Hot topic: Physical-layer network coding," in *Proc. ACM Mobicom '06*, (Los Angeles, CA, USA), pp. 358–365, Sep. 2006.

[5] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Proc. IEEE ICC '06*, vol. 9, pp. 3885–3890, IEEE, 2006.

[6] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network

information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[8] S. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: Tutorial, survey, and beyond," *Physical Communication*, vol. 6, pp. 4–42, Mar. 2013.

[9] S. Verdú, "Minimum probability of error for asynchronous gaussian multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 85–96, 1986.

[10] S. Verdú, *Multiuser Detection*. Cambridge University Press, 1998.

[11] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[12] B. Suard, G. Xu, H. Liu, and T. Kailath, "Uplink channel capacity of space-division-multiple-access schemes," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1468–1476, 1998.

[13] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communications–a tutorial," *IEEE Trans. Commun.*, vol. 30, no. 5, pp. 855–884, 1982.

[14] P. Li, L. Liu, K. Wu, and W. K. Leung, "Interleave division multiple-access," *IEEE Trans. Wireless Commun.*, vol. 5, no. 4, pp. 938–947, 2006.

[15] L. Lu, T. Wang, S. C. Liew, and S. Zhang, "Implementation of physical-layer network coding," *Physical Communication*, 2012.

[16] S. Zhang and S. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 788–796, May 2009.

[17] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[18] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.

[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[20] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[21] D. Wubben and Y. Lang, "Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems," in *Proc. IEEE Glob. Telecomm. Conf. 2010*, pp. 1–5, Dec. 2010.

[22] L. Lu and S. Liew, "Asynchronous physical-layer network coding," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 819–831, Feb. 2012.

[23] F. Lehmann, "Joint channel estimation and decoding for trellis-coded MIMO two-way relay networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 8, pp. 1455 – 1468, Aug. 2013.

[24] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[25] M. Gupta and Y. Chen, *Theory and Use of the EM Algorithm*, vol. 3. Now Pub, 2011.

[26] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.

[27] J. Dauwels, S. Korl, and H. Loeliger, "Expectation maximization as message passing," in *Proc. IEEE Int. Symp. on Inf. Theory 2005*, pp. 583–586, Sep. 2005.

[28] J. Dauwels, A. Eckford, S. Korl, and H. Loeliger, "Expectation maximization as message passing-part I: Principles and Gaussian messages," *Arxiv preprint arXiv:0910.2832*, pp. 1–14, 2009.

[29] C. Georghiades and J. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 300–308, Mar. 1997.

[30] E. Chiavaccini and G. Vitetta, "MAP symbol estimation on frequency-flat Rayleigh fading channels via a Bayesian EM algorithm," in *Proc. IEEE Int. Conf. Comm. 2011*, pp. 1057–1061, Jun. 2001.

[31] C. Cozzo and B. Hughes, "Joint channel estimation and data detection in space-time communications," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1266–1270, Aug. 2003.

[32] N. Noels, V. Lottici, A. Dejonghe, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "A theoretical framework for soft-information-based synchronization in iterative (turbo) receivers," *EURASIP Journal on Wireless Communications and Networking*, no. 2, pp. 117–129, 2005.

[33] D. Torrieri, E. Ustunel, S. Min, and D. Kang, "Iterative CDMA receiver with EM channel estimation and turbo decoding," in *Proc. IEEE Military Communications Conference 2006 (MILCOM'06)*, pp. 1–6, Oct. 2006.

[34] C. Herzet, V. Ramon, and L. Vandendorpe, "A theoretical framework for iterative synchronization based on the sum–product and the expectation-maximization algorithms," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1644–1658, May 2007.

[35] Q. Guo and D. Huang, "EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 4030–4035, Aug. 2011.

[36] J. Ylioinas and M. Juntti, "Iterative joint detection, decoding, and channel estimation in turbo-coded MIMO-OFDM," *IEEE Trans. Vehi. Tech.*, vol. 58, no. 4, pp. 1784–1796, Apr. 2009.

[37] E. Panayirci, H. Senol, and H. Poor, "Joint channel estimation, equalization, and data detection for OFDM systems in the presence of very high mobility," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4225–4238, Aug. 2010.

[38] U. Fawer and B. Aazhang, "A multiuser receiver for code division multiple access communications over multipath channels," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 1556–1565, Feb./Mar./Apr. 1995.

[39] A. Kocian and B. Fleury, "EM-based joint data detection and channel estimation of DS-CDMA signals," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1709–1720, Oct. 2003.

[40] S. Wu, U. Mitra, and C. Kuo, "Iterative joint channel estimation and multiuser detection for DS-CDMA in frequency-selective fading channels," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3261–3277, Jul. 2008.

[41] A. Assra, W. Hamouda, and A. Youssef, "EM-based joint channel estimation and data detection for MIMO-CDMA systems," *IEEE Trans. Vehi. Tech.*, vol. 59, no. 3, pp. 1205–1216, Mar. 2010.

[42] M. Kobayashi, J. Boutros, and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation,"

IEEE J. Sel. Areas Commun., vol. 19, no. 8, pp. 1450–1460, Aug. 2001.

[43] A. Kocian, I. Land, and B. H. Fleury, "Joint channel estimation, partial successive interference cancellation, and data decoding for DS-CDMA based on the SAGE algorithm," *IEEE Trans. Commun.*, vol. 55, no. 6, pp. 1231–1241, Jun. 2007.

[44] M. O. Pun, M. Morelli, and C.-C. J. Kuo, "Iterative detection and frequency synchronization for OFDMA uplink transmissions," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 629–639, Feb. 2007.

[45] C. Tan and N. Beaulieu, "On first-order Markov modeling for the Rayleigh fading channel," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2032–2040, Dec. 2000.

[46] T. M. Cover and J. A. Thomas, *Elements of information theory*. Wiley-interscience, 2012.

[47] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[48] M. Biguesh and A. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[49] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo-like codes," in *Proc. 36th Annual Allerton Conf. on Communication, Control and Computing*, pp. 201–210, Sept. 1998.

[50] X. Wang and V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[51] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[52] G. Cocco, C. Ibars, D. Gunduz, and O. del Rio Herrero, "Collision resolution in slotted ALOHA with multi-user physical-layer network coding," in *Proc. IEEE 73rd Vehi. Tech. Conf. 2011*, pp. 1–4, May 2011.

[53] J. Goseling, M. Gastpar, and J. H. Weber, "Physical-layer network coding on the random access channel," in *Proc. IEEE Int. Symp. on Inf. Theory 2013*, pp. 2339–2343, Jul. 2013.

[54] Y. Zhu, D. Guo, and M. Honig, "A message-passing approach for joint channel estimation, interference mitigation, and decoding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 6008–6018, Dec. 2009.

[55] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the

channel," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 933–946, Mar. 2000.

[56] P. Li, "Interleave-division multiple access and chip-by-chip iterative multi-user detection," *IEEE Commun. Mag.*, vol. 43, no. 6, pp. S19–S23, 2005.

[57] K. Kusume, G. Bauch, and W. Utschick, "IDMA vs. CDMA: analysis and comparison of two multiple access schemes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 78–87, 2012.

[58] Y. G. Li and G. L. Stuber, *Orthogonal frequency division multiplexing for wireless communications*. Springer Science & Business Media, 2006.

[59] P. Li, Q. Guo, and J. Tong, "The OFDM-IDMA approach to wireless communication systems," *IEEE Wireless Commun.*, vol. 14, no. 3, pp. 18–24, 2007.

[60] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM: I," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1668–1677, 1999.

[61] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[62] N. Wiberg, *Codes and decoding on general graphs*. Ph.D. dissertation, Linköping University, Linköping, Sweden, 1996.

[63] T. Peng, Y. Xiao, X. He, and S. Li, "Improved detection of

uplink OFDM-IDMA signals with carrier frequency offsets," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 646–649, 2012.

[64] J. Dang, F. Qu, Z. Zhang, and L. Yang, "Experimental results on OFDM-IDMA communications with carrier frequency offsets," in *OCEANS'12*, pp. 1–5, IEEE, 2012.

[65] D. D. Huang and K. Letaief, "An interference-cancellation scheme for carrier frequency offsets correction in OFDMA systems," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1155–1165, 2005.

[66] S. Manohar, D. Sreedhar, V. Tikiya, and A. Chockalingam, "Cancellation of multiuser interference due to carrier frequency offsets in uplink OFDMA," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2560–2571, 2007.

[67] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[68] J. Liu and J. Li, "Parameter estimation and error reduction for OFDM-based wlans," *IEEE Trans. Mobile Comput.*, vol. 3, no. 2, pp. 152–163, 2004.

[69] M. Morelli, "Timing and frequency synchronization for the uplink of an OFDMA system," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 296–306, 2004.

[70] L. Lu, L. You, Q. Yang, T. Wang, M. Zhang, S. Zhang, and S. C. Liew, "Real-time implementation of physical-layer

network coding," in *Proc. the second workshop on Software Radio Implementation Forum (SRIF'13)*, pp. 71–76, 2013.

[71] L. Rugini and P. Banelli, "BER of OFDM systems impaired by carrier frequency offset in multipath fading channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2279–2288, 2005.

[72] K. Sathananathan and C. Tellambura, "Forward error correction codes to reduce intercarrier interference in OFDM," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 4, pp. 566–569, 2001.

[73] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: a unifying theory and numerical methods for characterization," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, vol. 47, no. 5, pp. 655–674, 2000.

[74] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the em algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 4, pp. 477–489, 1988.

[75] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[76] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes.* McGraw-Hill,New York, 2002.

[77] S. M. Kay, *Fundamentals of Statistical signal processing, Volume 1: Estimation theory.* Prentice Hall PTR, 1998.

[78] IEEE 802.11 Working Group, "IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE 802.11 LAN Standards 2007*, 2007.

[79] Ettus Inc., "Universal software radio peripheral." `http://www.ettus.com`.

[80] G. FSF. , "GNU Radio - Gnu FSF Project.." `http://gnuradio.org/redmine/projects/gnuradio`.

[81] F. Rossetto and M. Zorzi, "On the design of practical asynchronous physical layer network coding," in *Proc. IEEE SPAWC'09*, pp. 469–473, 2009.

[82] T. Wang and S. Liew, "Joint channel estimation and channel decoding in physical-layer network coding systems: An EM-BP factor graph framework," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2229–2245, Apr. 2014.

[83] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[84] J. Dauwels and H.-A. Loeliger, "Phase estimation by message passing," in *Proc. IEEE ICC '04*, pp. 523–527, 2004.