Performance Evaluation of Optimized Physical Layer Network Coding

Cheng Chen
Doctor of Philosophy

University of York
Electronic Engineering

September 2017

"Research is what I am doing when I don't know what I am doing."

Wernher von Braun

Abstract

With the development of the wireless networks these years, devices are required to have an even higher data rate, and thus the interference between devices remains a significant problem to be solved. In order to secure a reliable transmission without wasting network capacity, we would necessarily select a proper transmission scheme. The Physical layer Network Coding (PNC) , is one of the optimum choices as it can increase the efficiency largely, especially in a system with 2 users and 1 relay (TWRC). Generally speaking, we may divide our research into two aspects: channel and channel coding.

From the channel coding's view, we build the system of uncoded QPSK modulated PNC and convolutional coded PNC (again modulated by QPSK). Due to the feature of the superimposed QPSK constellation, we introduce 3 different fixed mappings (plus the adaptive mapping) in order to decode the superimposed symbols at the relay more preciously. Also, we derive the theoretical upper bound of the error rate of the superimposed symbols at the relay. For uncoded PNC, we derive the SER; for coded PNC, we derive the BER, and we focus on an finite accumulation of the terms so as to obtain a tight error bound which is closer to the BER curve of the symbol-level simulation.

From the channel's view, we build the system of QPSK modulated PNC on single-carrier channel and the PNC on OFDM channels. For PNC on single-carrier channels, we assume the transmission channels are all flat fading channels. Nevertheless, it is quite difficult for it to cope with the frequency selective channels and time delay channels. As a result, we introduce OFDM technique which can largely resist their interferences. We develop different adaptive mappings on OFDM inside the sub-band to further improve the performance.

The theoretical upper bound for the error rate of the superimposed signals received at the relay will be vital in the system-level simulation, which is the part of the project DIWINE that my research benefits from.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to gratefully acknowledge my supervisor Professor Alister Burr for offering me great help during my research study. I have benefit a lot from his motivation, knowledge, enthusiasm and patience.

Also, I would like to thank Dr Dong Fang and Dr Mehdi Molu for valuable advice which helped me strengthen my understanding of the research topic.

Further thanks go to all staff and colleagues of the Communication group, for their help and support throughout my research ...

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Chapter 1

Introduction

1.1 Overview

With the recent development of wireless networks, system structures have become so complicated that it is impossible to simulate the entire network (all nodes and all links). This has stimulated the application of system-level simulation [1] which can analyse the overall performance of a network based on link level abstraction. This approach has already been widely applied in the conventional networking paradigm, but its application to cooperative relaying networks, such as that introduced by the DIWINE project [2], is less well understood. DIWINE considers wireless communication in a dense relay / node scenario where PNC (Physical-layer Network Coding) messages are flooded via dense massively air-interacting nodes in a self-contained cloud of wireless devices. More details of the DIWINE paradigm, and the approach so far taken to system-level simulation in the project, are given in [3]. PNC [4] recovers a network coded combination of the transmitted data symbols directly from the superimposed signal at a relay and thus it is more efficient because it avoids treating combined signals as interference.

The system-level simulation could be applied to common multi-node transmission system, which consist of multiple transmitors, multiple relays and 1 final destination. These will be very important in the Internet of Things (IoT) [5] such as

smart family.

Among all PNC models, the simplest example may be the Two-way Relay Channel (TWRC) [6] which involves 2 users and 1 relay. Other multi-hop systems can be treated as the combinations of multiple TWRC which means we may estimate the overall performance (in terms of packet error rate (PER)) from the error rate at a single relay (system-level simulation). Our main focus is the building and the performance analysis (the theoretical upper bound of the error rate at the relay in terms of SER or BER) of the PNC TWRC at all stages (uncoded, coded, on the Orthogonal Frequency Division Multiplexing [7](OFDM) channel, etc) so that the system-level simulation in the project DIWINE can be applied based on these result.

Our research is progressive and evolutionary. If we consider the channel properties, we may divide the research into two parts: PNC on single carrier system and PNC on OFDM system. Else, if we take the view of the application of the channel coding, our research can also be divided between uncoded PNC and channel coded PNC. Thus, there are 4 different systems which are combinations of either system of the above two categories. We start our research from the basic QPSK modulated TWRC model, then give the solution of the mapping under different channel fading states. There are some relative values of the fade coefficients of the channels to one relay (phase and amplitude) which make recovery of individual source data symbols difficult. These are called "singular fade states". Typically, we derive the theoretical SER or BER at the relay across all channel states, and compare it with the simulated result from the symbol-level simulation to prove that the theoretical upper bound is applicable. Based on this, we build the system-level simulation of a multi-hop system as required by the project DIWINE, and the theoretical SER at the relay is applied to estimate the overall PER. Next, the convolutional code is applied to the system to further boost the performance. Again, the building and the performance analysis are both given. Finally, we apply the OFDM channels to our system, as it is commonly used in the modern communication system to cope with frequency selective channel and time delaying channels, to which the single-carrier channel is vulnerable to deal with.

1.2 Contribution

The major contributions in this thesis can be stated as follows:

- The theoretical upper bound (SER) for uncoded QPSK modulated PNC at the relay is derived. There are three fixed mappings available (the bit-wise XOR, the reversed XOR and the anti-$\frac{\pi}{4}$) to deal with different singular fading between two transmitting channels, and we introduce the adaptive mapping which always picks the best mapping under the current fading states. Adaptive mapping helps to build a unique constellation of possible received superimposed symbols in which points with small Euclidean distance have the same network code value, or conversely for points with different network code value have large Euclidean distance. Particularly, we obtain the SER under specified singular fading as it is require at the system-level simulation. The averaged SER, however, is obtained by taking the Monte-Carlo simulation over all fading states.

- The theoretical upper bound (BER) for the convolutional coded PNC at the relay is obtained. Unlike some previous research which focuses on the decoding of the individual data sent from both users, we are interested in the joint decoding which decodes the received signals at the relay into superimposed symbols. We start the derivation from the basic pairwise-error probability, to the event error probability, and finally obtain the upper bound of the BER at the relay.

- The introduction of the sub-band concept for PNC on OFDM channels. Similar to the PNC on single-carrier channels, we introduce the adaptive mapping which adapt all pairs of sub-channels (from both users), and name it 'fully adaptive mapping'. However, this will be quite difficult to realize if we apply a linear channel code to it. As a trade-off, the sub-band is introduced which consists of several adjacent sub-carriers. Also, we will introduce

two adaptive mapping schemes in each sub-band: the 'majority vote' and the 'local optimum' mapping schemes.

1.3 Thesis Outline

The structure of the thesis is as follows:

- In Chapter 2, a literature review of PNC is discussed, including system model, transmission and decoding techniques, and mapping problems associated with modulation. Also, the concept of linear channel coding and OFDM channels are discussed. These three concepts will be combined in our project to obtain the best performance in terms of the error rate at the relay.

- In Chapter 3, an uncoded PNC model on flat fading channels is given, which is modulated by QPSK. We present different mappings to deal with different relative fading states, elaborate the detailed transmission and decoding process (e.g. condense the mapping application into a matrix operation), and derive the theoretical upper bound for the error rate across all singular fadings in terms of the total Euclidean distance. Simulation results are then presented to show the comparison between the theoretical error rate and the result averaged by the symbol-level simulation.

- In Chapter 4, the system-level simulation is given to show the application of the error rate derived in Chapter 3. By dividing the transmission models into different phases, we can elaborate the transmission process in terms of multiple vector/matrix operations. We assume that the transmission channel fading states are known to the destination nodes, and the transmission nodes are 'informed' by the destination nodes of the best applicable mapping (in terms of the error rate at the relay and the rank of the combined

mapping matrix) to be chosen. The system-level simulation is then applied to estimate the overall packet error rate (PER). Simulation results are presented to show the comparison of PER between the system-level simulation and the symbol-level simulation.

- In Chapter 5, we introduce convolutionally coded PNC on the single-carrier channel. In addition to the elaboration of the system model, we again focus on the derivation of the upper bound of the error rate at the relay across all fading states. The derivation process is much more complicated than that for the uncoded PNC. We start from the pairwise error probability, the event error probability, and finally generalize them to the bit error rate. Simulation results are again given for comparison.

- In Chapter 6, we proceed to the application of PNC on OFDM channels in order to solve the time delay and frequency selective channels. Due to the limitation of OFDM channels, we introduce the sub-band and the adaptive mapping/modulation (the majority vote and the best effort). We build the system for both uncoded and coded PNC, and compare their performance with that on the single-carrier channels.

1.4 Publication list

Journal Papers

- Cheng Chen and Alister Burr. "System-level Simulation of the QPSK Modulated Physical Layer Network Coding," Submitted to IET Journals

- Cheng Chen and Alister Burr. "Performance Analysis for the Convolutional-Coded Physical-Layer Network Coding," In preparation

Conference Papers

- A. Burr, Cheng Chen, M. Molu, K. Ramantas and J. S. Vardakas, "System-level simulation of multihop wireless networks using physical-layer network coding," *2015 European Conference on Networks and Communications (EuCNC)*, Paris, 2015, pp. 285-289.

- A. G. Burr and Cheng Chen, "Low energy wireless relay networks using physical layer network coding," *2015 1st URSI Atlantic Radio Science Conference (URSI AT-RASC)*, Gran Canaria, Spain, 2015, pp. 1-1.

Chapter 2

Literature Review

2.1 PNC

2.1.1 Introduction

To the best knowledge of the author, the first appearance of PNC is [8]. As this reference describes, at the physical layer [9] of wireless networks, which is the bottom level of the 7-layer of the Open Systems Interconnection model (OSI) [10], all data are transmitted in the form of electromagnetic (EM) waves. Also, Multiple-Input and Multiple-Output (MIMO) [11] [12] [13] techniques have been widely used in the industry including 4G and WIFI, in which transmission of the signal from a sender/user is often received by more than one node (relay or receiver). Likewise, a receiver/relay may receive signals from multiple transmitters simultaneously.

These characteristics may cause interference among signals, and the interference has a negative effect on wireless networks (e.g. it distorts the wanted received signal and degrades the decoding/de-mapping accuracy). The traditional way of dealing with such situation is to treat the unwanted signal as noise and filter it out through receiver design or transmission scheduling. e.g. Code Division Multiple Access (CDMA) [14].

Instead of treating interference as additive noise to be avoided, we can on the other hand accept interference and use it to improve the overall throughput. To do so in a multi-hop network, we require the system to have:

- A relay node which must be able to convert simultaneously received signals into interpretable output signals to be relayed to their final destinations.

- A destination which must be able to extract the information addressed to it from the relayed signals.

Network coding, which is able to combine and extract information through simple Galois field GF(2^n) additions [15] [16], provides a good foundation to meet such goals. However, network coding arithmetic is generally only applied on bits that have already been detected. Specifically, it cannot be used to resolve the interference of simultaneously arriving EM signals at the receiver. So, criterion 1 above cannot reliably be met.

Thus, we aim to introduce Physical-layer Network Coding (PNC). The main idea of PNC is similar to network coding, but at the lower physical layer that deals with EM signal reception and modulation. As a result, sums of EM signals can be mapped to GF(2^n) sum of digital bit streams through a proper modulation-and-demodulation technique at relay nodes, which is called de-mapping in this thesis. In this way, the interference from the unwanted signal may become part of the arithmetic operation in network coding.

2.1.2 Two-Way Relay Channel

We start from a model consisting of 2 users and 1 relay, called two-way relay channel (TWRC). This three-node wireless network is a basic unit for cooperative transmission being introduced in [17] [18] [19] [20]. In cooperative transmission,

the relay node can choose different transmission strategies in forwarding to the destination node, such as Amplify-and-Forward or Decode-and-Forward [17], according to different Signal-to-Noise (SNR) situations. We focus on the Decode-and-Forward (DNF) strategy, and consider frame-based communication in which a time slot is defined as the time required for the transmission of one fixed-size frame. Each node is equipped with an omni-directional antenna, and the channel is half duplex so that transmission and reception at a particular node must occur in different time slots.

Before introducing the PNC transmission scheme, we first describe the traditional transmission scheduling scheme and the straightforward network-coding scheme for mutual exchange of a frame in the three-node network [20] [21] or the TWRC.

2.1.2.1 Traditional Transmission in a TWRC

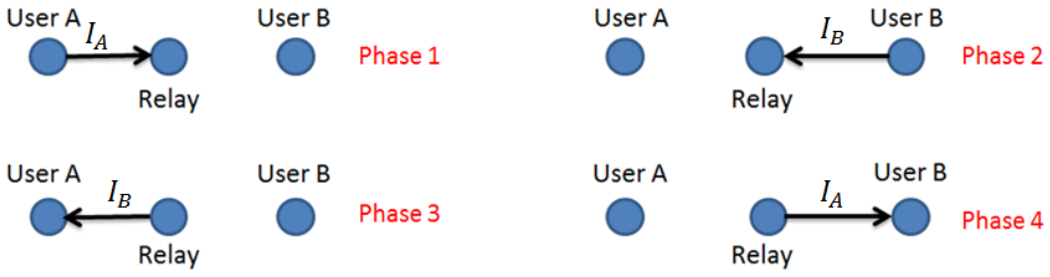

FIGURE 2.1: Traditional transmission in a TWRC

The traditional way for user A and B to exchange their information takes 4 time slots as fig 2.1 shows. In time slots 1 and 2, user A and B send their information I to the relay separately. This is because in the traditional transmission approach interference is usually avoided by prohibiting the overlapping of signals from user A and B to the relay in the same time slot.

After that, the relay sends the superimposed information back to both users in time slots 3 and 4. Thus, a total of four time slots are needed for the exchange of two frames in opposite directions.

In practice, a better order might be Phase 1 − > Phase 4 − > Phase 2 − > Phase 3, as it reduces the amount of data required to be stored at the relay as well as the end to end delay.

2.1.2.2 Straightforward Network Coding Scheme

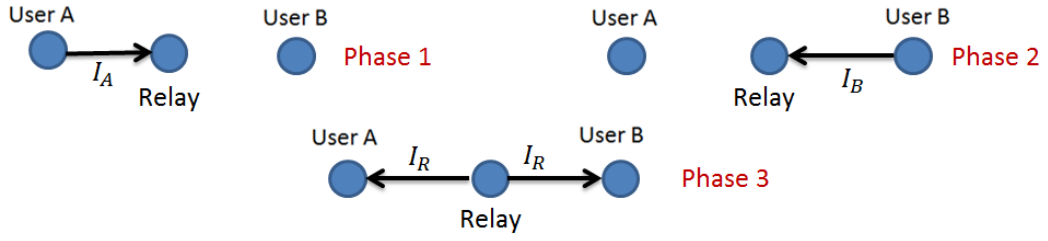

FIGURE 2.2: Straightforward network coding in a TWRC

Ref. [21] and [20] outline a straightforward way of applying network coding in the three-node wireless network, and Fig 2.2 illustrates the idea. First, user A sends frame I_A to relay at time slot 1 and then user B sends frame I_B to the relay at time slot 2. After receiving I_A and I_B, relay combines the frames into one as follows:

$$I_R = I_A \oplus I_B \tag{2.1}$$

where $\oplus$ denote the bitwise exclusive OR operation being applied over the entire length of frames I_A and I_B. The relay then broadcasts I_R to both S_A and S_B at the same time. When user A receives I_R, it extracts I_B from I_R using the local information I_A, as follows:

$$\begin{aligned} I_A &= I_A \oplus I_R \\ &= I_A \oplus (I_A \oplus I_B) \\ &= I_B \end{aligned} \tag{2.2}$$

Thus, a total of three time slots are needed for the exchange of two frames in opposite directions, which increases the transmission efficiency by 33%. We can further optimize it by using a PNC model.

2.1.2.3 Physical-Layer Network Coding(PNC)

We now introduce PNC. In PNC, we require the system to have:

- symbol-level synchronization.
- the use of power control.

Thus, S_A and S_B arrive in the relay at the same time with the same frame/packet length as well as amplitude. In this way, we can develop the PNC model shown in Fig 2.3.

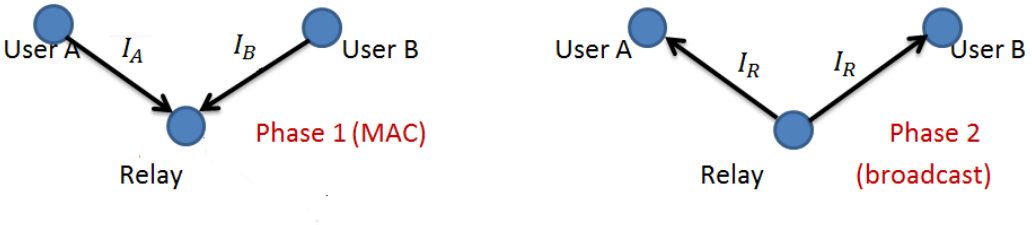

FIGURE 2.3: PNC

Assuming perfect synchronization, users could transmit their symbols/signal to the relay at the same time without considering the mutual interference in time slot 1, and the relay broadcast S_R to both users like the straightforward network coding scheme. We call them MAC phase and broadcast phase respectively. Thus a total of only 2 time slots are needed, which doubles the transmission efficiency comparing with the traditional way.

2.1.2.4 BPSK Modulation

These information-theoretic approaches are about finding the capacity bound - they are not really meant to be practical. But in this thesis we are concerned with practical approaches and their performance, rather than the information-theoretic limits.

Let us initially assume the use of BPSK modulation in all the nodes [6]. The BPSK example uses symbol-level de-mapping or per-symbol denoising, while the information-theoretic approaches such as [22] [23] [24] use codeword-level de-mapping. However, these information-theoretic approaches are about finding the capacity bound rather than practical approach. In this thesis we are concerned with practical approaches and their performance, rather than the information-theoretic limits. The motivation is:

- Symbol-level de-mapping is closer to practical communications, as it can be applied for packets of small length .

- the peculiar operation at the relay gives rise to interesting problems related to the modulation design in terms of choice of the signal constellation and the network coding used at the relay.

Assume that the channel gains are 1 (flat fading), so the transmitted symbols from A and B satisfy: $S_A, S_B \in \{-1, 1\}$ which correspond to $I_A, I_B \in \{0, 1\}$. The received signal at R is $R_S = S_A + S_B + S_N$, where S_N is the noise at the relay. If this process is noiseless ($S_N = 0$), then the possible received symbols are $R_s \in \{-2, 0, 2\}$. Then we decode R_s into the superimposed binary sequence I_R, and finally re-modulate it to S_R in order to broadcast it. We may now form table 2.1

Clearly, if $R_S = 0$, then it is impossible to decode the separate symbol from A and B even in a noiseless case (in this case we know one user has send a '1' and the

I_A	S_A	I_B	S_B	R_S	I_R	S_R
0	-1	0	-1	-2	0	-1
0	-1	1	1	0	1	1
1	1	0	-1	0	1	1
1	1	1	1	2	0	-1

TABLE 2.1: BPSK modulated PNC

other a '0', but we cannot tell which user sent which). Instead, we can still de-map (denoise) it to a network coded form: the relay will be fully aware that the symbol that both users transmitted have opposite polarity, and broadcast $S_R = 1$ (here we re-modulate I_R to S_R) to both users. In this way users can de-map the received symbol to I_R using table 2.1, and finally extract the wanted information using formula 2.2. This special modulation/demodulation mapping scheme, referred to as PNC mapping in this thesis, can be applied to obtain the equivalence of GF(2) summation of bits from both users at the physical layer ($I_A \oplus I_B$). Also, this PNC mapping reduces the overall complexity, since the relay does not need to decode the separate message but only the network coded one, and is a major advantage over traditional transmissions.

2.1.2.5 QPSK Modulation

Now we move on to the use of QPSK modulation in all the nodes. Recall that a QPSK data stream can be considered as two BPSK data streams: an in-phase stream and a quadrature-phase stream. Thus every transmitted symbol I can be divided into an in-phase (I) and a quadrature-phase (Q), represented as $I(1)$ and $I(2)$, which means the 1^{st} and the 2^{nd} bit of the transmitted symbols. Alternatively, we can simply make I a 2-row matrix, with the 1^{st} row acting as $I(1)$ and the 2^{nd} as $I(2)$. Thus, we can conclude that $I_A, I_B \in Z_{2^2}$(a length 2 vector of binary symbols), where $Z = \{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \}$.

Let M be a QPSK constellation mapper used at each user, then after modulation, we obtain QPSK symbols S_A and S_B as $S_A = M(I_A)$ and $S_B = M(I_B)$. If we

again assume the channel gains are 1 and the channels are noiseless, then at the relay, we have:

$$\begin{aligned} R_S(I) &= S_A(I) + S_B(I) \\ R_S(Q) &= S_A(Q) + S_B(Q) \end{aligned} \tag{2.3}$$

Again the relay cannot extract the individual information transmitted by both users (I_A or I_B). As long as the relay can de-map (In general cases, we cannot ignore channel fading or the noise, and we will have to map the received signal against a look-up table. These will be further explained in chapter 3), re-modulate and resend the superimposed symbol sequence S_R back to both users, the users could still subtract the wanted information from S_R, and the end-to-end delivery of information will be successful, thus we have:

$$\begin{aligned} I_R(1) &= I_A(1) \oplus I_B(1) \\ I_R(2) &= I_A(2) \oplus I_B(2) \end{aligned} \tag{2.4}$$

For simplicity, we define the QPSK modulated symbol as $S \in \{\pm 1 \pm j\}$, and thus we can make a similar PNC mapping to it as table 2.2 shows:

2.1.2.6 Mapping Selection and Error Calculation

Now we no longer assume that channels are noiseless or the gain are 1. We already know that the received signal at the relay R over the MAC channel is written as $R_S = h_A S_A + h_B S_B + S_N$, where h_A and h_B are the channel coefficients from user A and B, respectively. Here, S_N is the Gaussian noise with a variance of σ^2. We assume slow block fading and perfect channel estimation at the receivers we then

I_A	S_A	I_B	S_B	R_S	I_R	S_R
0 0	$-1-j$	0 0	$-1-j$	$-2-2j$	0 0	$-1-j$
0 0	$-1-j$	0 1	$-1+j$	-2	0 1	$-1+j$
0 0	$-1-j$	1 0	$1-j$	$-2j$	1 0	$1-j$
0 0	$-1-j$	1 1	$1+j$	0	1 1	$1+j$
0 1	$-1+j$	0 0	$-1-j$	-2	0 1	$-1+j$
0 1	$-1+j$	0 1	$-1+j$	$-2-2j$	0 0	$-1-j$
0 1	$-1+j$	1 0	$1-j$	0	1 1	$1+j$
0 1	$-1+j$	1 1	$1+j$	$2j$	1 0	$1-j$
1 0	$1-j$	0 0	$-1-j$	$-2j$	1 0	$1-j$
1 0	$1-j$	0 1	$-1+j$	0	1 1	$1+j$
1 0	$1-j$	1 0	$1-j$	$2-2j$	0 0	$-1-j$
1 0	$1-j$	1 1	$1+j$	2	0 1	$-1+j$
1 1	$1+j$	0 0	$-1-j$	0	1 1	$1+j$
1 1	$1+j$	0 1	$-1+j$	$2j$	1 0	$1-j$
1 1	$1+j$	1 0	$1-j$	2	0 1	$-1+j$
1 1	$1+j$	1 1	$1+j$	$2+2j$	0 0	$-1-j$

TABLE 2.2: QPSK modulated PNC

draw up a look-up table of all possible received superimposed QPSK symbols at the relay.

In the de-mapping stage, we first consider a de-mapping function consisting of a look-up table L and a mapper C, preceded by maximum-likelihood (ML) detection. The ML detection performs:

$$X_R = \underset{(I_A, I_B) \in Z \times Z}{\operatorname{argmin}} |R_S - (h_A M(I_A) + h_B M(I_B))|^2 \tag{2.5}$$

to obtain the estimated I_A as $I_A{}'$ and I_B as $I_B{}'$, where $(h_A M(I_A) + h_B M(I_B)) \in L$, and $X_R = \begin{bmatrix} I_A{}' \\ I_B{}' \end{bmatrix}^T$. The mapper C generates a network coded data matrix I_R from the ML estimates as $I_R = C(I_A{}', I_B{}')$. Using the constellation mapper M_R, the superimposed QPSK symbol to be broadcast is then given as $S_R = M_R(I_R)$. Note: the ML detection is used just for obtaining the quantized version of the received

signal. The communication throughput through the MAC channel is inevitably restricted by the weaker link of the two terminals.

The relaying node R selects the best de-mapping method from a well-designed finite mapping book based on the channel condition. This require the de-mapped product to be unique. According to [6], for successful decoding we have:

- $C(I_A, I_B) \neq C(I_A^{'}, I_B)$ for any $I_A \neq I_A^{'} \in Z$ and $I_B \in Z$

- $C(I_A, I_B) \neq C(I_A, I_B^{'})$ for any $I_B \neq I_B^{'} \in Z$ and $I_A \in Z$

where C is an arbitrary mapping. In the previous section, we make :

$$C(I_A, I_B) = I_A \oplus I_B \tag{2.6}$$

which is called "the XOR mapping". However, this mapping may be vulnerable when the relative phase between two channels satisfies some special conditions. As can be seen from Fig 2.4 and Fig 2.5, this mapping may result in a situation that closest points have different network coding value (I_R) and thus have more chance to have a decoding error. As a result, we introduce another 2 mappings against different fadings (including phase and amplitude fading), which are called "reversed XOR mapping" (similar to [25]) and "anti-$\frac{\pi}{4}$ mapping" (We have a 5-QAM mapping in [6], but it is not linear). The solid dividing lines are the decoding/demapping thresholds for adjacent points, and the numbers on the graph correspond to the I_R values in table 2.2. The mappings and combinations will be further explained in the next chapter.

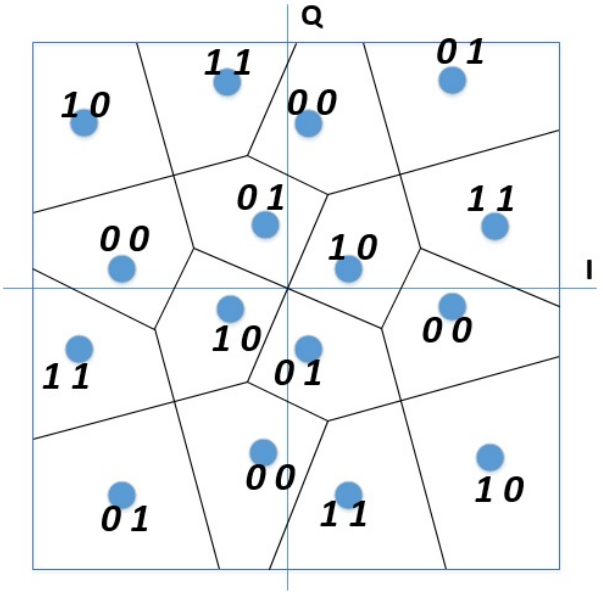

FIGURE 2.4: The XOR mapping, the relative phase is around 90^o

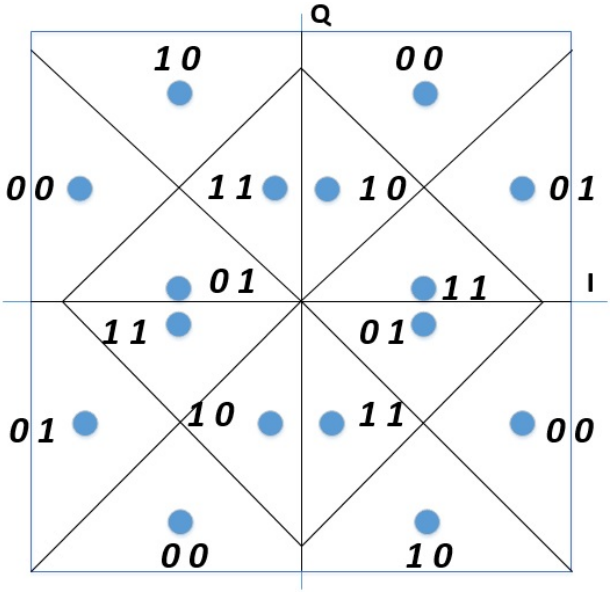

FIGURE 2.5: The XOR mapping, the relative phase is around 45^o

2.2 Convolutional Code

Convolutional code, as a linear channel code with error correcting capability, generates the codeword by feeding the data stream into a sliding application of a boolean polynomial function. The word‘convolution’ comes from the The sliding

application of the encoder over the data, and this code is thus called 'convolutional coding'. In the decoding stage, a time-invariant trellis is applied thanks to the sliding nature of the convolutional codes. With some reasonable complexity, this time invariant trellis decoding allows us to apply maximum-likelihood soft-decision method. [26]

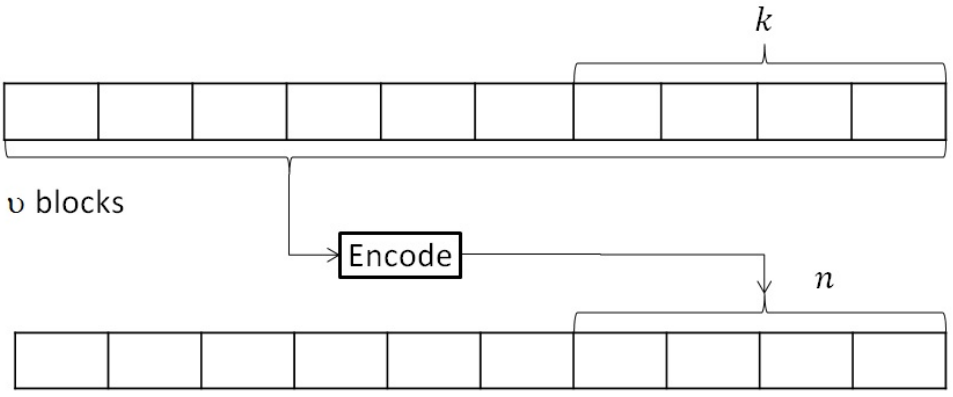

FIGURE 2.6: Structure of a general convolutional code

The structure of the convolutional code is shown in Fig 2.6. The braces in the figure can be treated as a sliding window which shifts along the input data sequence one block (k symbols) at a time as discussed in [27]. Thus we can define three important variables:

- n as the current block of code symbols
- k as the current block of data symbols corresponding to code symbols n
- ν as the total number of input blocks within the sliding windows, known as the *constraint length* of the code.

The term $\nu - 1$ may also be called the *memory order* of the code, and we name the term $k(\nu - 1)$ as the memory of the code (the capability of the encoder to store

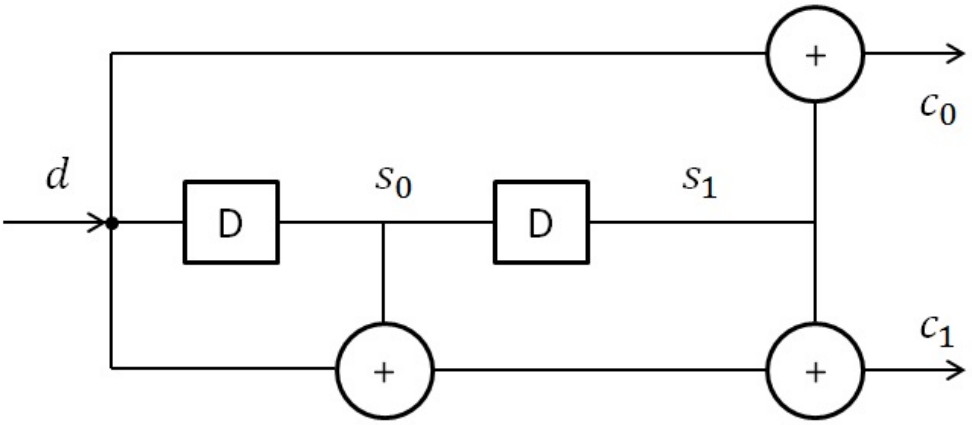

FIGURE 2.7: Encoder for a (2,1,3) convolutional code

the number of previous data symbols). The code is then called a (n, k, ν) code. We are interested in the normal case that both data and code symbols are binary.

In our research, we apply a (2,1,3) convolutional code with a constraint length of 3 and rate 1/2, shown in Fig 2.7, where d is the current data input, s_0 and s_1 are the shifted delayed input, and c_0 and $c1$ are output bits. As can be seen, there is a two-tap delay in the system. Thus, the output codeword $G(D)$ can be represented as:

$$[G(D)] = \left[1 + D^2 \quad 1 + D + D^2\right] \tag{2.7}$$

in which the '1' term in the polynomial means the input bit, D means the input bit with a delay of 1 time slot, and D^2 with 2 time slots, etc. The addition is modulo-2, equivalent to the exclusive OR operation that outputs a binary result. This is the typical convolutional code we apply in our research. If we replace D with 2 in 2.8 (and treat the addition as the convolutional addition), then we get :

$$[G(D)] = \begin{bmatrix} 5 & 7 \end{bmatrix} \tag{2.8}$$

Thus, this can also be called a (5,7) convolutional code.

2.2.1 Trellis diagram

The *trellis diagram* was first introduced by Forney in [28], which can be tread as an simplified version of a tree diagram in [29]. From the diagram, we define that:

- Columns/length of the diagram as information block periods
- Nodes as the state of the encoder after each data bit
- Branches as state transitions
- Branch labels as code symbols to be output
- Path as code sequence

If we input a sequence 10000 into a (5,7) convolutional encoder, then we will have the trellis diagram shown in 5.2, there the black numbers represent the current state, and the purple numbers are the output code bits. With the help of the trellis diagram, it is easy to see that the encoded data is 11011100.

2.2.2 Decoding

Decoding is always the most difficult and complicated process in the implementation of convolutional codes. It is further complicated when we compare it with

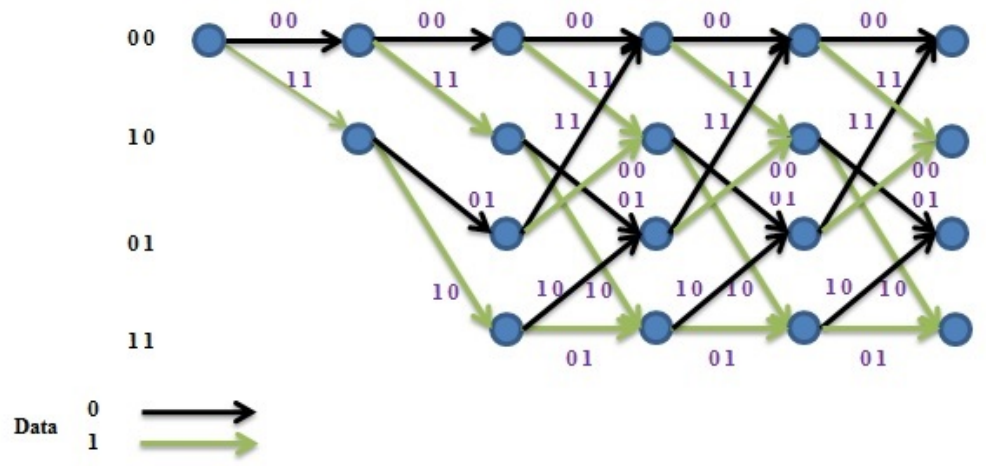

FIGURE 2.8: Trellis diagram for a (2,1,3) convolutional code

block codes because there are no distinct codewords, only potentially infinite code sequences. This means that, in principle, the decoder might have to wait an unlimited time before it is able to decide between two possible code sequences.

There are two basic techniques for decoding of convolutional codes: maximum likelihood sequence detection (MLSD), and sequential decoding. These have also been described as *breadth first* and *depth first*, respectively. The first is also better known as Viterbi decoding, which is more commonly used. The words *breadth first* and *depth first* comes from their different decoding methods. For the two paths going into the same node, Viterbi decoding will always delete the one with larger Hamming distance, which focuses on the breath of the trellis diagram, while the other method focus on the total length instead.

2.2.2.1 Viterbi decoding

The word Viterbi inside the Viterbi algorithm for MLSD in convolutional code is named after Andrew Viterbi in [30] in which the Viterbi algorithm was first applied. Through careful comparison, it aims to find a predicted path inside the code trellis that is most similar to the received signal sequence with one code block

at a time in proceeding. The predicted most similar path is called the survivor path. Thus, the algorithm may be summarized in a pseudo-code form as:

Algorithm 1 The Viterbi decoding [[27],pp.186]

For each data block period (column on trellis diagram) {
For each final state (node of right of column) {
For each branch leading top this node {
Calculate distance metric of received sequence from branch label
Add to metric talky kept at initial node (on left of trellis column)
}
Select branch with smallest distance metric. Store in a list of survivor paths
Delete the other paths from the list of survivors, and if this leaves some earlier paths 'floating', delete them too (leaving one survivor path to each final state node)
}
if the deletion process leaves only one survivor path over some earlier data period, the corresponding data block can be output.

The algorithm searches every path through the trellis. However, by noting that only one path to a given node may be the correct one, it restricts the number of paths to be considered to a finite level. Therefore it is guaranteed to find the required closest path as well as a maximum-likelihood decoder (within the decoding metric being used).

	stage 1	stage 2	stage 3	stage 4	stage 5
received sequence	11	1101	110101	1101010	1101011011
best path	11	1101	110111 110100	11010010	1110011011 1101110011 1101001010 1101001001
distance	0	0	1	1	2

	stage 6	stage 7
received sequence	110101101111	11010110111100
best path	111001101111 110100101011	11100110111100
distance	2	2

TABLE 2.3: Decoding of the received code sequence for a (5,7) convolutional code

Table 2.3 shows the application of this algorithm to the trellis of the (5,7) convolutional code in decoding a received code sequence 11010110111100, starting from the zero state. The best path (with smallest Hamming distance with the received code sequence) may shift between different trellis branches in the different decoding stages. Note that the distance in the calculation is the Hamming distance, and we apply hard decision decoding to deal with it. At the end of the process, the survivor path is 11010010101100, which has a Hamming distance of 2 compared with the received code sequence. We may encounter the situation that there is a 'tie' (the word 'tie' comes from [27]) where two branches merge at one node as a destination. This require the the decoder to make an arbitrary decision of which branch to choose. Here we choose the higher of the two branches although the choice could equally well be made randomly.

At the end of the received sequence the survivor paths follow the same path in the first trellis period only. Hence, the algorithm can only decode the 1_{st} data block. Later bits may be decoded in the following stages, and it is obvious that a delay of more than one block period is necessary, and we cannot be sure how many subsequent periods it will take before a given information bit may be decoded. Thus we introduce a *truncation window* to secure a fixed, limited delay. This technique limits the length of the survivor path to a certain level [31]. A decision is made to eliminate the unwanted data blocks at each decoding period, and the final survivor path is the one with the lowest distance metric.

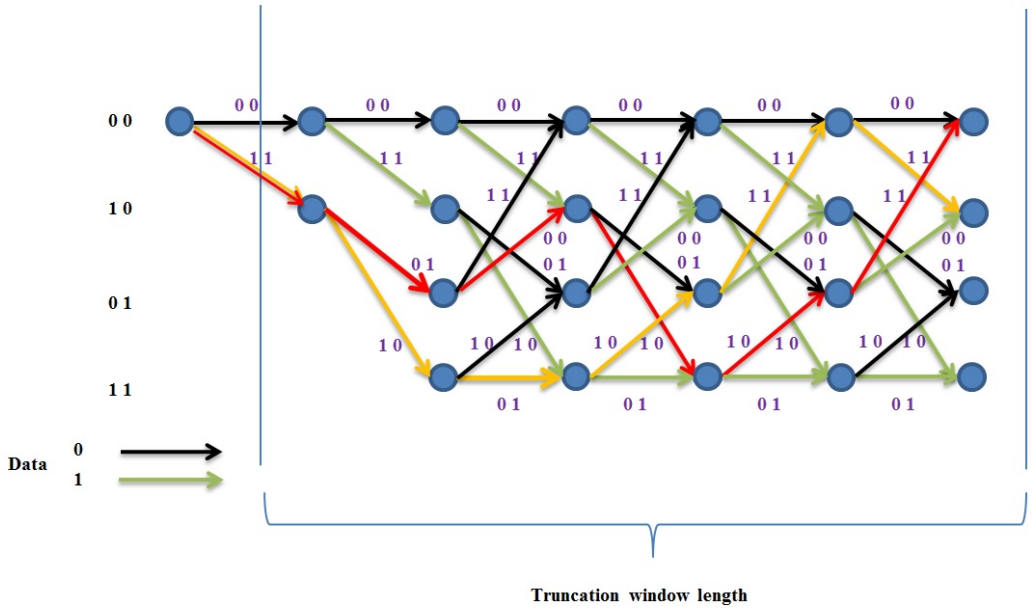

FIGURE 2.9: Truncated window apply to Viterbi decoding

Now we focus on the last 2 periods of decoding with a truncated window of length 5 block as Fig 2.9 shows, where the yellow path is the current path in the window, and the red path is the original decoded path. In the second decoding stage, there is a 'tie' between the two survivor paths in the two sub-figures from which they diverge. To decode the message, we trace the trellis back to the beginning (from the first node) where the two paths are the same, and it corresponds to a data '1' (the downward from the first node). Thus we output that bit. Next, according to the current survivor path , we output a '1' again. We note that the path is not the same as the previous result, and is sub-optimum (one data error has occurred). This shows how a truncation window could possibly degrade the

decoding performance. In practice, the degradation is negligible if the truncation window length is five or six times the constraint length.

In practice, the implementation of the decoder with truncation window will require the survivor path to be stored in some form temporarily in each window, and the decoder need to 'track back' each time when the window move forward, which makes the implementation rather complex. Still, we have two basic techniques that meet the requirements: *register exchange* and *trace back* [32], which means the survivor is stored in a register or as a linked list, and can be interchanged or traced during the decoding process.

Also, for soft decision decoding, the decoder must be fed with finite-precision soft information, which may reduce the decoding accuracy compared with using the true soft information input. This may require a quantization, typically 8 levels, which corresponds to 3-bit representation.

We have up to now focused on hard decision decoding, and Hamming distance is used as the metric. However, we can use any appropriate metric in the algorithm as it is completely general. This means, if we consider Euclidean distance as the metric, then soft-decision decoding is also available. If we consider the White Gaussian noise channel, then the soft decision decoding is optimum [27] given that the full Viterbi algorithm is applied to the maximum-likelihood decoding.

2.2.3 Performance Analysis

The bit error probability P_{eb} of convolutional codes decoded using the Viterbi algorithm was published in [31] [33]. Here we aim to obtain P_{eb} of convolutional codes using the Viterbi soft decision decoding techniques. [34]

The bit error probability P_{eb} has an upper bound in the form of a summation as:

$$P_{eb} \leq \sum_{i=1}^{\infty} \sum_{d=d_H}^{\infty} ia(d,i)P_{ep} \tag{2.9}$$

where $a(d, i)$ is the number of paths which divert from the correct path with information weight i (number of errors in sequence data) and distance d (errors in code sequence), P_{ep} is the probability that a given erroneous path is chosen by the decoder instead of the correct one, called *pairwise error probability*, and d_H is the minimum free Hamming distance of the code. If the sequence fed into the decoder is not binary, then the Euclidean distance is calculated here instead of Hamming distance. We will further explain these in chapter 5.

However, this upper bound, in the form of general expression, may have a large gap with the simulation result in the low SNR region because it is an infinite summation. We aim to find a more precise estimation of P_{eb}. [34] provides a calculation of the upper bound of P_{eb} so that it is tighter at error rate of 10^{-2} and below for some commonly used codes by rearranging equation 2.10 as:

$$P_{eb} \leq \sum_{d=d_H}^{\infty} e(d) P_{ep} \tag{2.10}$$

where

$$e(d) = \sum_{i=1}^{\infty} i a(d, i) \tag{2.11}$$

$e(d)$ is called *data-weighted distance spectrum*, and it can be obtained by looking up the table in [27]. For the (5,7) code we applied in our research, $e(d) = 1, 4, 12, 32...$ which corresponds to $i = 1, 2, 3, 4...$

[34] makes this bound tighter by summing not to infinity but to some smaller number of terms. Thus, it is then not strictly an upper bound

When we apply soft-decision decoding, if the code is binary, then the pairwise error probability can be expressed by the Hamming distance of the code. Thus, for every value of d in (2.11), we have:

$$P_{ep} = Q\left[\sqrt{2dR\frac{E_b}{N_0}}\right] \tag{2.12}$$

where R is the code rate (for a (5,7) code, $R = \frac{1}{2}$). This will also be further explained in chapter 5 to obtain a tight bound.

There are two possible ways that binary codes may be applied to PNC: the multilayer PNC [35] and the multilevel coded PNC [36]. The multilayer PNC use an outer FEC code with an inner network code (which means a networking coding layer and a channel coding layer.), while multilevel coded PNC propose a scheme in which separate encoders are applied to each bit of the constellation label. Our research are mainly based on the multilayer PNC. This will be further explained in chapter 5.

2.2.4 Coded modulation

Coded modulation (CM) [37] is a general concept for a system in which an encoder is combined with a higher order modulator (e.g. QPSK) with more than one bit per symbol to increase the spectral efficiency. Examples of encoders can be simple linear block codes or convolutional codes. Further more, some more advanced encoders such as turbo codes or low-density parity-check (LDPC) codes can also be applied. For commonly used higher order modulation schemes, we may also introduce quadrature-amplitude modulation (QAM), and phase-shift keying (PSK).

Previous research [38] has precisely described some important process of coded modulation as:

- Combine an error correcting channel code (convolutional code, block code, turbo code) and a signal constellation.

- Signal constellation aggregates the coded bits as to map them into points in a way that enhances the distance properties of the code (typically Euclidean distance).

- Thus, a codeword can be seen as a vector of signal points.

- If we assume the channel is AWGN, the decoder will map the codeword to the closest received vector in terms of Hamming distance (hard-decision decoding), or map the squared Euclidean distance to the LLR (Log Likelihood Ratio) value (soft decision-decoding).

Our project benefits from the LLR and the mapping method mentioned above, although the structure in our system is not a strict coded modulation.

Nowadays, more advanced coded modulation systems are applied including bit-interleaved coded modulation (BICM) which inserts a bit interleaver between the encoder and the modulator. Also, the CM system can also be extended to MIMO systems with several antennas at both the transmitter and the receiver. Coded modulation has been widely applied in the industry today such as IEEE 802.11a/g/n/ac/p, 4G (HSDPA and HSUPA), 3GPP Long Term Evolution (LTE), WiMAX, and the latest DVB standards (DVB-T2, DVB-S2, and DVB-C2).

Coded modulation could achieve high coding gain within the given bandwidth by means of combining coding and modulation. Thanks to the mapping inside the linear modulation process, coded modulation could increase the constellation size and accuracy by adding additional redundancy to the trellis-coded signal. More importantly, coded modulation can be decoded with a soft-decision Viterbi decoding to have a coding gain (for a two-dimensional coded modulation over a White Gaussian channel) of about $3dB$ for a BER of 10^{-5}.

2.3 OFDM

Orthogonal frequency division multiplexing (OFDM) [39]has been widely implemented in modern communication standard with more than 30 years of research and development, including WIFI and 4G. In the early days, the massive complex computation and high speed memory still remain a problem for the realization of OFDM. These problems have been solved thanks to the advances of digital signal processing (DSP) and very large scale integrated circuit (VLSI) technologies. Also, sinusoidal generators and coherent demodulation require additional arrays, and the use of the fast Fourier Transform (FFT) helps to eliminate them. Furthermore, the recent favour of OFDM is due to its excellent performance which has been proved in [40] [41].

2.3.1 Parallel data transmission and multiple carriers

OFDM is a parallel data transmission system. In a conventional serial data system, the symbols may occupy the entire allocated bandwidth, and are transmitted in turns (sequentially). A parallel data transmission system is capable of overcoming the problem that such a serial system may encounter due to high bandwidth. A parallel system has several sequential streams of data which are transmitted simultaneously (several serial system operating at the same time), so that the throughput is boosted by the number of sequential streams. As a result the allocated bandwidth is divided, and each stream takes only a small part of it.

When there is a frequency selective fading, parallel transmission can spread the channels over several frequency bandwidth. In the traditional serial system, adjacent symbols maybe seriously distorted by the fading or impulsive interference in the channel. However, parallel system adapt this situation that many symbols are only slightly distorted (while some symbols remain heavily distorted), making the reconstruction/decoding of the symbols easier even without forward error correction (FEC). Also, as we divide the entire bandwidth into a number of sub-channels/sub-carriers, each sub-channel is relatively flat (the original bandwidth

may be frequency selective). Another advantage is that equalization at each sub-band becomes easier as OFDM channels can be equalised just by compensating for the phase shift and attenuation of each sub-band separately (if the system has no ISI or ICI). Alternatively, [42] shows that the implementation of different encoding may even avoid such equalization by proper phase estimation.

We know that the total frequency bandwidth is divided into N frequency sub-channels (In OFDM, these sub-channels can be overlapping to each other). It is very important to separate them in a proper way. OFDM separates the sub-channels by making them orthogonal to each other, as the orthogonal signals can be separated at the receiver by correlation. This also eliminates the inter carrier interference (ICI). To further adapt the system, we can choose the carrier spacing by letting it be equal to the reciprocal of the symbol period. Thus, OFDM can be regarded as a form of multi-carrier modulation with carefully selected carrier spacing so that each sub-carrier is orthogonal to the others.

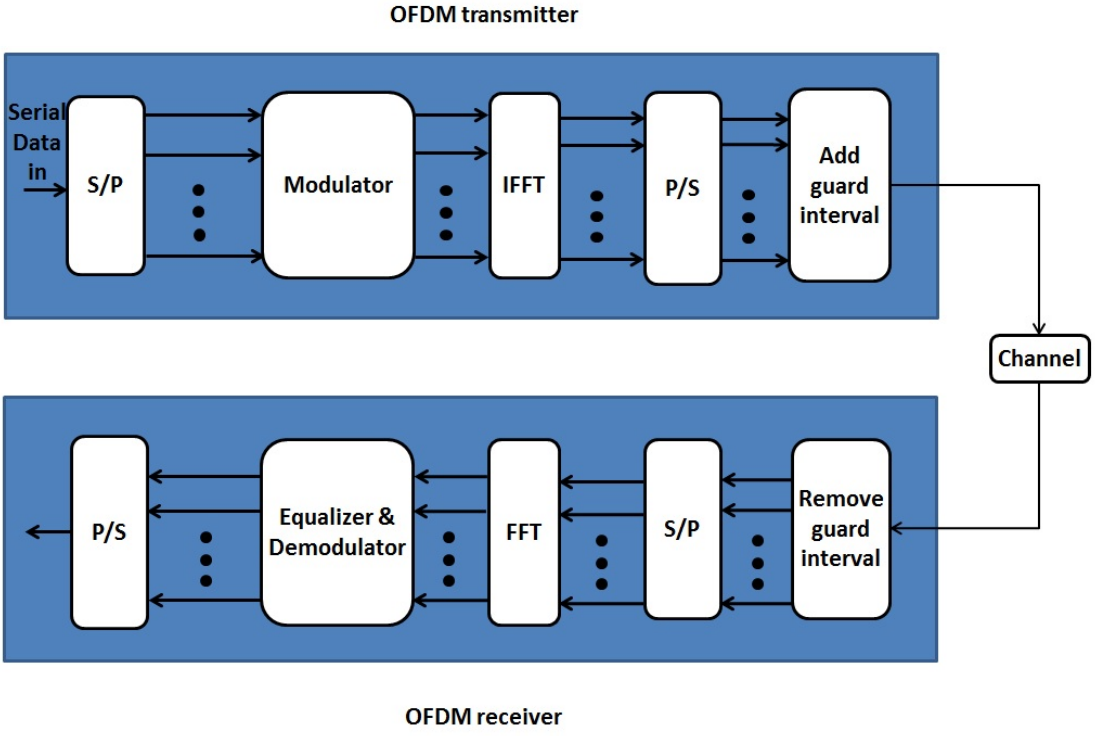

FIGURE 2.10: General OFDM structure

Fig 2.10 shows the general structure of a typical FFT-based OFDM system.The incoming serial data is first converted to parallel form and is modulated on the

sub-carriers, with X bits on each sub-carrier, where X is the signal constellation of the modulation on sub-carrier. e.g, for QPSK modulation, $X = 2$. Then the modulated complex signal is coonverted to the time domain in the baseband by the inversed FFT (IFFT).

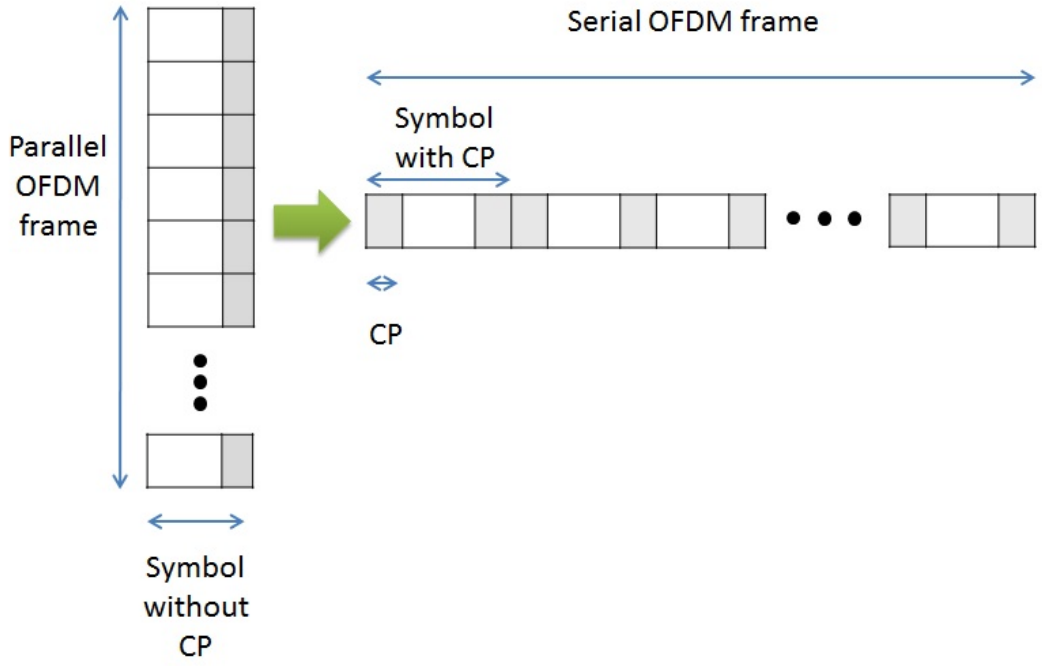

FIGURE 2.11: The cyclic prefix

When the above ISI or inter-carrier interference (ICI) are eliminated, FFT will help the receiver to separate the sub-channels and maintain their orthogonality. However, factors such as the multipath of the channel can still result in additional ISI. To solve this, a simple solution is to increase the symbol duration by introducing a guard period to make the distortion insignificant. There are several ways of doing that, and among them the cyclically extended guard interval [41] is commonly chosen. The word 'cyclic' shows that this interval is repeated in the sequence, and is also called cyclic prefix, where each OFDM symbol is extended by a period extension of the signal itself. Taking Fig 2.11 as an example. the cyclic prefix comes from the signal of the last N_{cp} samples and is added at the beginning of the entire serial OFDM symbol, where N_{cp} is the length of the cyclic prefix. Note: in order to dissolve time delay effectively, N_{cp} needs to be larger or

equal to the maximum delay of the channel, expressed in samples. We choose the cyclic prefix so that:

- The carrier synchronization at the receiver can be maintained, so signal rather than a long time of silence is always transmitted.

- The transmission system could be modeled by applying the cyclic convolution between the OFDM signal and the channel response. [7]

Thus the total duration of the symbol may become

$$T_{total} = T_{cp} + T \tag{2.13}$$

where T_{cp} is the guard interval (duration of the cyclic prefix) and T is the useful symbol duration. As long as T_{cp} is longer than the channel impulse response or the mutipath delay, then the ISI can be eliminated. However, the guard interval/cyclic prefix will reduce data throughput, and the limit of T_{cp} varies according to different applications. In the industry, T_{cp} is usually less than $T/4$. (We may have a larger T_{cp} if the delay is huge, and this will require more sub-carriers in order to maintain the transmission efficiency)

After this, the serial OFDM symbols are convolved with the channel impulse response and passed to the receiver. Upon receiving the signal, the receiver performs the inverse of the operations of the transmitter to recover the original data. A one-tap equalizer maybe used to eliminate the channel distortion, and its coefficients are obtained from the channel information.

2.3.2 Coded OFDM (COFDM)

OFDM is able to transmit data in a frequency selective channel by means of time and frequency diversity. However, it cannot change the channel's fading status.

Thus, due to the position of the individual sub-carrier, it could be deeply affected by the fading on its allocated sub-channel. Thus, we may consider the use of a channel code or a coded modulation to further improve the transmission accuracy. An example is trellis coded modulation (TCM) [43][44], which can be treated as the combination of a linear modulation (e.g QPSK modulation) and a linear code with trellis decoding (e.g convolutional code). Fig 2.12 shows the structure of COFDM.

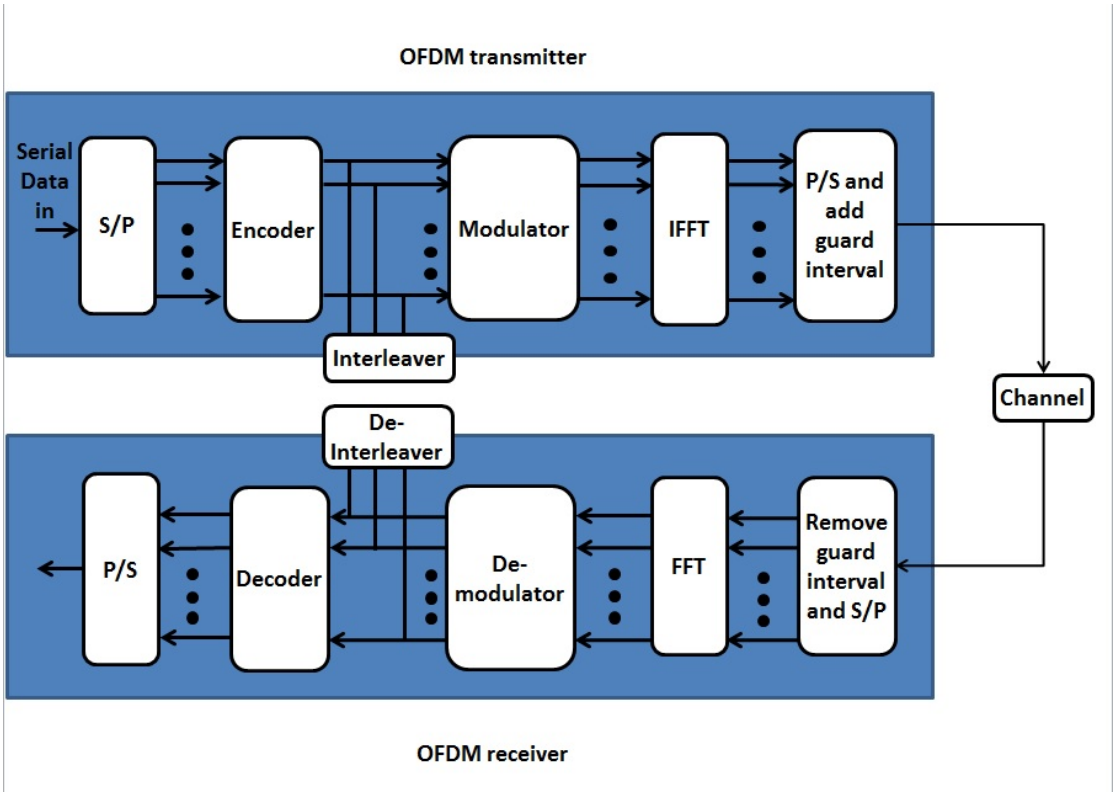

FIGURE 2.12: TCM on OFDM

We know that the major advantage of OFDM over traditional serial transmission is that OFDM can convert a wide-band frequency selective channel into several narrow-band relatively flat channels by means of parallel and multi-carrier transmission. This enables the uses of linear channel coding on OFDM.

Although the trellis code could improve the BER effectively, it is invulnerable towards impulsive [45] or burst noise [46], which may be caused by co-channel interference and phase noise that lead to data-dependent crosstalk. The transmission errors have a strong time/frequency correlation. Thus, we need to avoid the

adjacent data being transmitted on the same sub-carrier. To do this, we introduce an interleaver that breaks the correlation after the encoding step. As 2.12 shows, it is inserted between encoder and modulator.

From [47], we know that with a properly designed guard interval (cyclic prefix), interleaving and channel coding, COFDM is able to solve the impact of strong echoes. Also, computer simulation and laboratory demonstrations [48] [49] have shown the BER improvement resulting from this. Also, [39] suggests that COFDM may enable the use of omni-directional antennas in urban areas and mobiles reception with a high carrier-to-noise (C/N) ratio, because COFDM is more resist to multipath.

2.3.3 Sub-band and adaptive modulation

In COFDM, the BER of different sub-carriers depends on the noise and the frequency response of the sub-channels, and each sub-carrier is attenuated individually by the sub-channel's fast-fading and time delay. Thus, the BER of each sub-carrier may be very different. If the system is non-adaptive, then the overall BER is mainly determined by the few severely faded sub-carriers. Instead of employing higher order modulation or higher rate channel codings across all sub-carriers sub-carriers, we may identify the sub-carriers with high bit errors and modulate/encode them separately.

We may merge several sub-carriers into a sub-band [50] [51], and apply adaptive modulation [52] to each sub-band to optimize the overall SNR before the IFFT block. This technique on OFDM channels could be traced back to 1989 by Kalet, and was further developed by [53] and [54]. Importantly, [53] shows that the required SNR for a BER of 10^{-3} can be reduced by $5dB$ to $15dB$ if we apply adaptive modulation instead. Furthermore, [55] exploit the capacity of fading channel (refer to time variant Shannonian channel).

According to these previous researches, we group the adjacent sub-carriers (with similar frequency response) into sub-band, and apply the adaptive mappings (introduced in section 2.1) to each sub-band. In this way, all sub-carriers in the same sub-band [50] will be mapped in the same way. Different sub-bands, however, can be mapped or modulated differently providing that each sub-band is independent to the others.

Fig 2.13 shows the structure of adaptive modulation in OFDM. To realize this, we group the sub-carriers into sub-bands, and calculate their instantaneous SNR under each mode of modulation. Then the average instantaneous SNRs are compared with threshold values (which should be pre-defined) to choose the mode. This process is where the word 'adaptive' from. Finally, the information about modulation (and coding) to be used are sent to the transmitter using a feedback channel (assumed to be flat and noiseless). This procedure is used not only in the OFDM channel, but also the system-level simulation in our research.

As Fig 2.13 shows, the coded frames in different sub-bands have different sizes, which shows that different channel codes (with different code rates) can be applied to different sub-bands according to the sub-channels' fading states. This is followed by the adaptive modulation on each sub-band which outputs the modulated frame with a same length.

Fig 2.14 shows the transmission block structure of the sub-band in COFDM suggested by [56], in which the OFDM frames is divided into several sub-bands, and the linear coded frames combines a number of OFDM symbols. In this scheme, the modulated symbols in each coded frame after encoding should be the same, as the number of sub-carriers in each sub-band is the same.

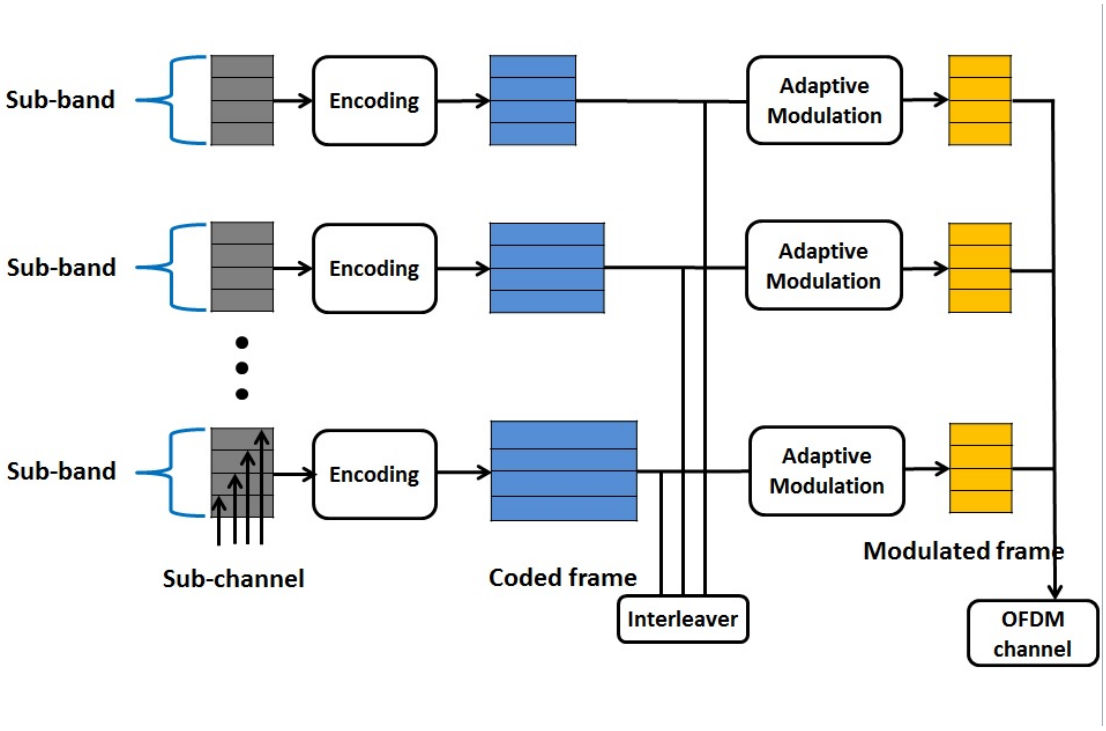

FIGURE 2.13: Adaptive OFDM

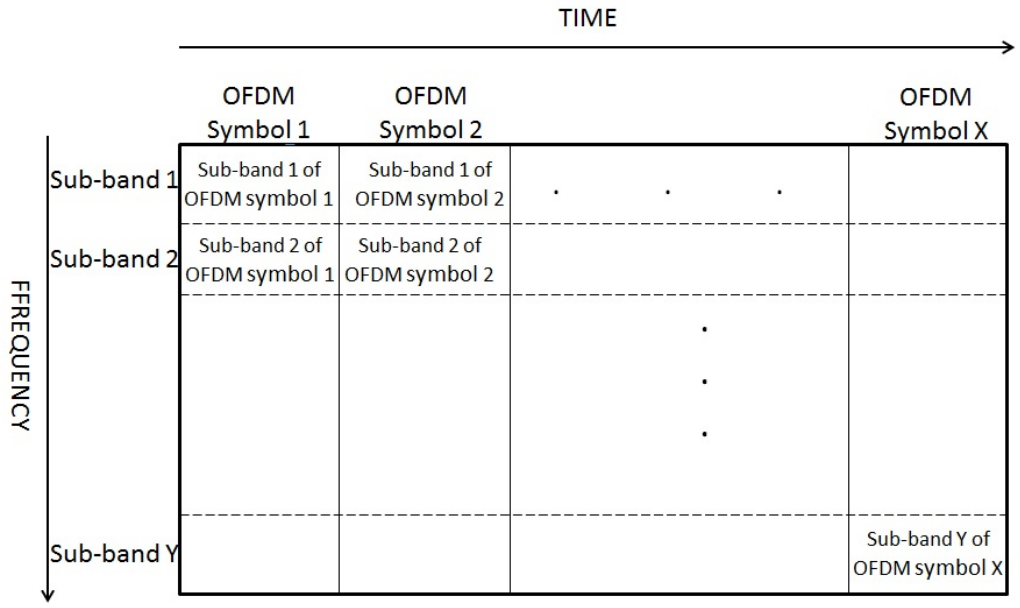

FIGURE 2.14: Transmission block structure of sub-band

Chapter 3

Un-coded PNC

There a re a lot of previous similar work related to this topic. [57] derives the SER for the AWGN channel where the singular fading is either 0 or $\frac{\pi}{2}$ (fixed fading). [58] analyses the case where the magnitude of the channels has a Rayleigh distribution, but the phases are the same (a fading of 0). Furthermore, [59] derives the error rate for fading channels, but the model is not a PNC model, and the bits from two users are being detected individually.

Based on the above research, [60] gives the BER analyse of the BPSK modulated PNC over Rayleigh fading channels by putting the Euclidean distance and the angles between constellation points into the formula suggested in [61]. [62] further improves the theoretical error rate with the ML decoding. Finally, [63] extends the research to QPSK/M-QAM modulation. However, it exams the error rate when the relative fading is 0. Also, it focuses on the XOR mapping only.

In the literature review we assume that channels are noiseless and the gains are 1. These are all ideal, and in reality we would expect noisy Rayleigh fading channels. As a result, in QPSK modulated PNC, there is a problem that adjacent points may have different network coded value (I_R), which may lead to a de-mapping failure. Thus we introduce mapping selection and error rate calculation.

In this section, we describe the QPSK modulated un-coded PNC MAC model, and all the mappings used under different channel fading states.

3.1 System Model

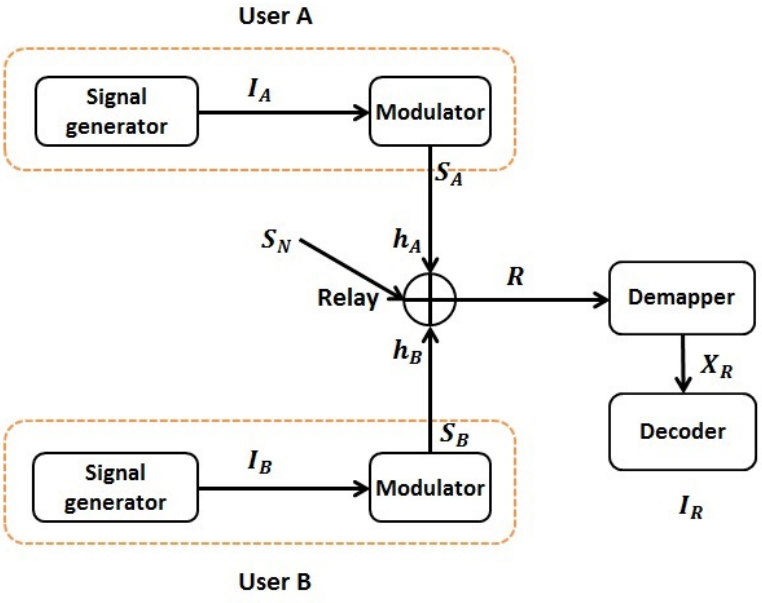

FIGURE 3.1: General model of MAC phase for PNC

The system model of the PNC MAC phase is shown in Fig 3.1 above. In this scheme, we assume that two users, A and B, transmit data to the relay with the same frame/packet length simultaneously. We assume all transmitting processes are perfectly synchronized. Our focus is the error rate of the superimposed information at the relay, assuming that the relay knows the channel information. Note that a variety of more complicated multihop wireless relay networks, such as that illustrated in Fig 3.2, may be built up from a set of these two-user MAC

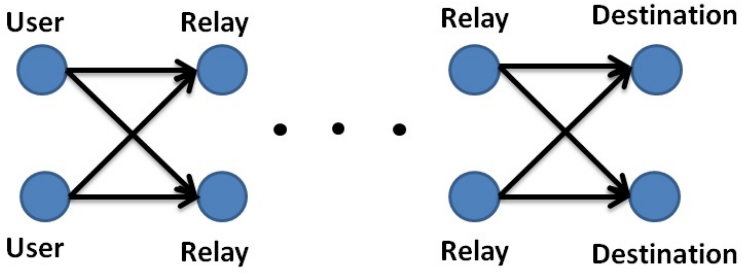

FIGURE 3.2: Example multi-hop network

phase models, since every relay in Fig 3.2 receives signals from two sources (or prior relays) simultaneously.

We assume that the source nodes A and B transmit QPSK modulated symbols S_A, S_B taking the complex values$\{\frac{\sqrt{2}}{2}(\pm 1, \pm j)\}$, in which the symbol energy is normalised to 1. We also assume that conventional Gray-coded binary mapping is used. Thus we can have the relationship between S and I in Table 3.1 .

QPSK symbol (S)	binary symbol (I)
$\frac{\sqrt{2}}{2}(-1-j)$	0 0
$\frac{\sqrt{2}}{2}(-1+j)$	0 1
$\frac{\sqrt{2}}{2}(1-j)$	1 0
$\frac{\sqrt{2}}{2}(1+j)$	1 1

TABLE 3.1: Constellation points and network coding

There are 2 users in the PNC, and each of them could transmit 4 possible symbols to the relay. As a result, there are 16 possible received symbol combinations at the relay, which we refer to 16 constellation points. Next, we compare the received superimposed complex symbol with the look-up table, and calculate the Euclidean distances between the symbol and every member in the look-up table. In this way, we could de-map the complex symbols into network coded ones by taking the related network coded value of the constellation point which has the shortest Euclidean distance with the received symbol. For example, if there are two constellation point (1,0) and (0,0) with network coded value 1 and 0 respectively, then a received symbol (0.6,0) will be de-mapped as (1,0) if we define the threshold as (0.5,0), and will be further de-mapped as a network coded 1.

After the de-map stage, we form the symbol into a row vector of size 4. If we define $I_A(1)$ as the 1_{st} bit of the network coded symbol sent from user A, then the de-mapped vector (consist of the separate binary value) is defined as:

$$X_R = \{I_A(1) \quad I_A(2) \quad I_B(1) \quad I_B(2)\}$$

And we can refer it to the constellation as table 3.2 shows:

Constellation (R)	de-mapped vector (X_R)	Constellation (R)	de-mapped vector (X_R)
$\frac{\sqrt{2}}{2}\big(h_A(-1-j)+h_B(-1-j)\big)$	0 0 0 0	$\frac{\sqrt{2}}{2}\big(h_A(1-j)+h_B(-1-j)\big)$	1 0 0 0
$\frac{\sqrt{2}}{2}\big(h_A(-1-j)+h_B(-1+j)\big)$	0 0 0 1	$\frac{\sqrt{2}}{2}\big(h_A(1-j)+h_B(-1+j)\big)$	1 0 0 1
$\frac{\sqrt{2}}{2}\big(h_A(-1-j)+h_B(1-j)\big)$	0 0 1 0	$\frac{\sqrt{2}}{2}\big(h_A(1-j)+h_B(1-j)\big)$	1 0 1 0
$\frac{\sqrt{2}}{2}\big(h_A(-1-j)+h_B(1+j)\big)$	0 0 1 1	$\frac{\sqrt{2}}{2}\big(h_A(1-j)+h_B(1+j)\big)$	1 0 1 1
$\frac{\sqrt{2}}{2}\big(h_A(-1+j)+h_B(-1-j)\big)$	0 1 0 0	$\frac{\sqrt{2}}{2}\big(h_A(1+j)+h_B(-1-j)\big)$	1 1 0 0
$\frac{\sqrt{2}}{2}\big(h_A(-1+j)+h_B(-1+j)\big)$	0 1 0 1	$\frac{\sqrt{2}}{2}\big(h_A(1+j)+h_B(-1+j)\big)$	1 1 0 1
$\frac{\sqrt{2}}{2}\big(h_A(-1+j)+h_B(1-j)\big)$	0 1 1 0	$\frac{\sqrt{2}}{2}\big(h_A(1+j)+h_B(1-j)\big)$	1 1 1 0
$\frac{\sqrt{2}}{2}\big(h_A(-1+j)+h_B(1+j)\big)$	0 1 1 1	$\frac{\sqrt{2}}{2}\big(h_A(1+j)+h_B(1+j)\big)$	1 1 1 1

TABLE 3.2: The PNC constellation and the separate network coding value/ look-up table

This table is also the look-up table at the relay. However, it is not the optimum choice to leave X as the final decoding result, because the noise could distort the transmitted symbol closer on to a neighbouring constellation point with different network coded value. One way to solve it is to do phase synchronization as [64] gives, which synchronize the phase of both channels to a same value (relative phase shift = 1).

In our research, instead, we would like to introduce three mappings which enlarge the distance to result in an error. To realize this, we choose a mapping that gives the same network code for source symbol combinations that result in the same signal, if the values of h_A and h_B are close to a given fading state. The mapping is attributed to each user in the modulator. There are several mappings available for this QPSK modulated PNC. Among them the bit-wise XOR mapping performs a modulo-2 linear combination in both the 1_{st} and the 2_{nd} bit of the network coding value, represented as:

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

which indicates that the 1^{st} bit of the mapped symbol label is formed as the

modulo-2 sum of the bits from the two sources corresponding to the real parts, while the second is the modulo-2 sum of the imaginary part bits. Then we form a matrix multiplication of the decoded symbol (the 4 column vector) and the mapping matrix so as to recover the decoded superimposed data as the decoding stage:

$$I_R = X_R M_1$$

This mapping performs well when the relative phase between the two channels is around a multiple of π. Fig 3.3 shows the decision region(we use black line as the dividing line that is used for de-map) and the network coded values (in black) for this mapping. Note: in all the following figures, the pairs of the 2-bit numbers represent the network coded values for the superimposed symbol S. As can be seen, in Fig 3.3 the closest points have the same network coding, which, in general, results in the lowest symbol error rate in decoding the network coded symbols.

Finally, we obtain the SER by compare I_R with the superimposed binary symbols of I_A and I_B, merged by the mapping chosen.

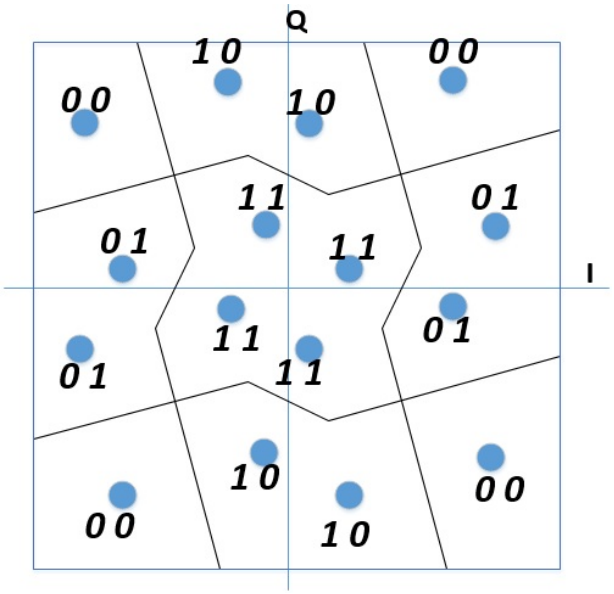

FIGURE 3.3: The XOR mapping, the relative phase is around 0

Unfortunately the XOR mapping is not the optimum selection when the relative phase between the two channels is around an odd multiple of $\frac{\pi}{2}$. Fig 3.4 shows the

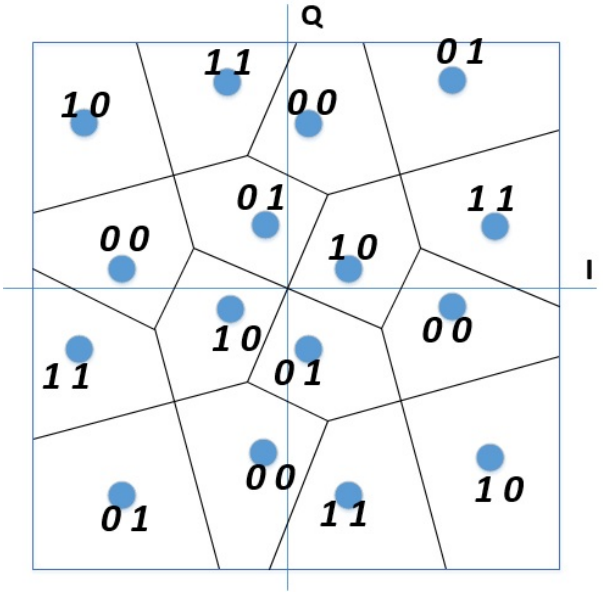

FIGURE 3.4: The XOR mapping, the relative phase is around 90^o

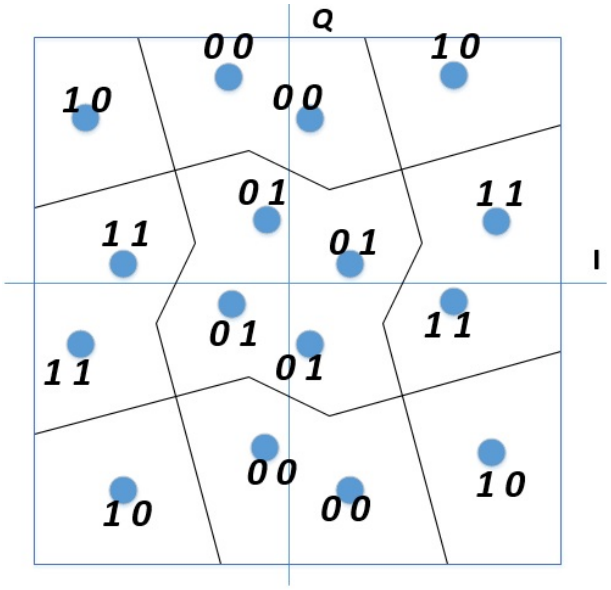

FIGURE 3.5: The reversed XOR mapping, the relative phase is around 90^o

result of the XOR mapping in this situation: the closest pairs of points have different network coding, and hence there will be a much higher error probability for the network coded symbol. Note: again the solid dividing lines are the decoding/demapping thresholds for adjacent points, and the numbers on the graph correspond to the I_R values. This setting applys to all similar figures in this chapter. So instead we apply a bit-reversed XOR mapping named as the reversed XOR mapping, which is similar in effect to the corresponding mapping mentioned in [6].

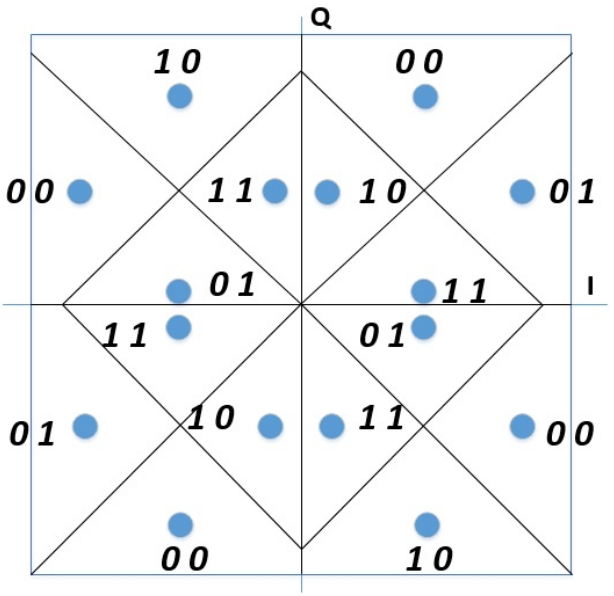

FIGURE 3.6: The XOR mapping, the relative phase is around 45^o

The mapping matrix is:

$$M_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

and the resulting constellation mapping under the relative phase of $\frac{\pi}{2}$ is shown in Fig 3.5. Like we described in literature review, the bold number in the figure shows the network coded value (I_R) of every point. Thus, we diffrentiate the two mappings against a particular relative phase $\frac{\pi}{4}$ (or its odd multipliers).

However, neither mappings may be the optimum choice when the relative phase is around an odd multiple of $\frac{\pi}{4}$ as Fig 3.6 shows the example.

We have previous research [6] noted that there are no linear mappings in this case which fulfills the exclusive law when the relay is used in the TWRC, but here we are considering more general relay network configurations such as that of Fig 3.1, for which different criteria for end-to-end decodability apply. (see [3] and references therein). However the mapping matrix

$$M_3 = \begin{bmatrix} 0 & 1, \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix},$$

(which we will refer to as the "anti-$\frac{\pi}{4}$ mapping matrix") results in the constellation shown in Fig 3.7, which again results in the closest pairs of points sharing the same network coded symbol label. The threshold for choosing this mapping will be introduced in the next section.

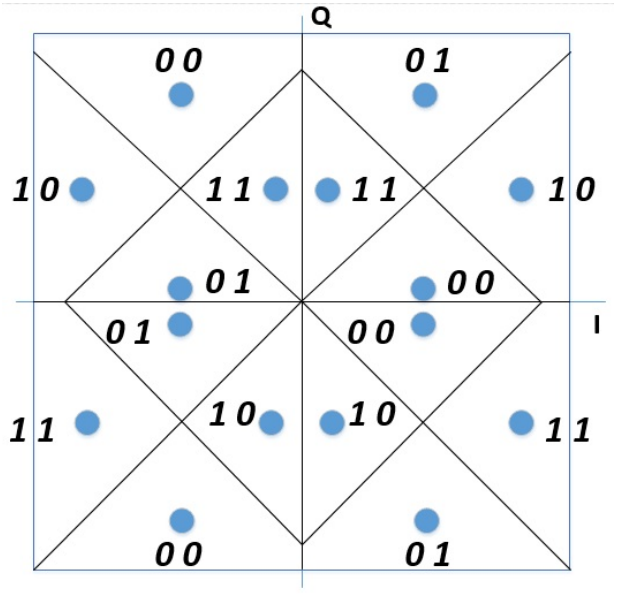

FIGURE 3.7: The anti $\frac{\pi}{4}$ mapping, the relative phase is around 45^o

And again, this mapping is not a cure-all, neither is it the optimum mapping when the relative phase is either around 0 (shown in Fig 3.8) or $\frac{\pi}{2}$ (these two cases have the same error rate due to the mapping symmetry)either. We will analyse these in the next section.

3.2 Analysis of Symbol Error

In this section, we derive the theoretical SER for the QPSK modulated PNC. We build different constellation diagrams according to different channel fading states (phase and amplitude), and calculate the theoretical SER based on the

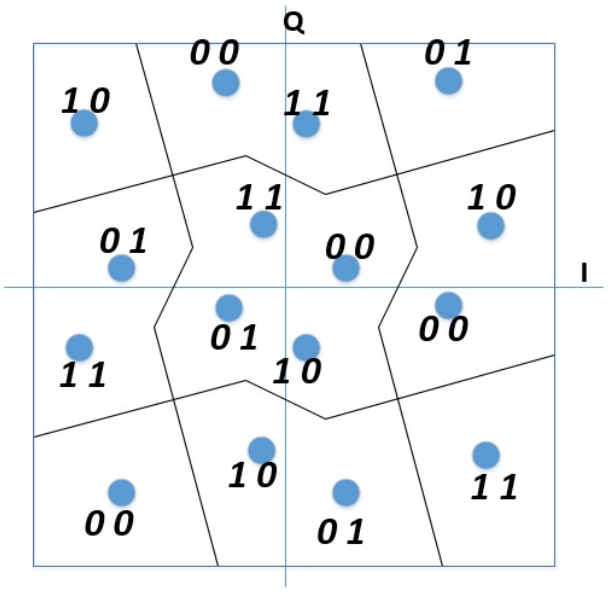

FIGURE 3.8: The anti $\frac{\pi}{4}$ mapping, the relative phase is around 0

diagrams. We will analyse the three fixed mappings and the adaptive channel method separately, and compare their performance. A comparison between the average theoretical error bound and the result of Monte Carlo simulation is given. For convenience, we define several variables according to the different mapping:

- $d_1 = \sqrt{2}|j(h_B - h_A) + h_B|$, to be used in the XOR and the reversed XOR mapping.
- $d_2 = \sqrt{2}|j(h_B)| = \sqrt{2}|h_B|$, to be used in all mapping.
- $d_3 = \frac{\sqrt{2}}{2}|2h_A| = \sqrt{2}|h_A|$, to be used in all mapping.
- $d_4 = \frac{\sqrt{2}}{2}|2\left(j\left(h_A\right) + j\left(h_B\right)\right)| = \sqrt{2}|h_A + h_B|$, to be used in all mappings .
- $d_5 = \frac{\sqrt{2}}{2}|2(j(h_A) - h_B)| = \sqrt{2}|jh_A - h_B|$, to be used in the anti-$\frac{\pi}{4}$ mapping only.

3.2.1 The XOR/Reversed XOR Mapping

We combine the consideration of the XOR and the reversed XOR mappings in order to exploit their symmetry.

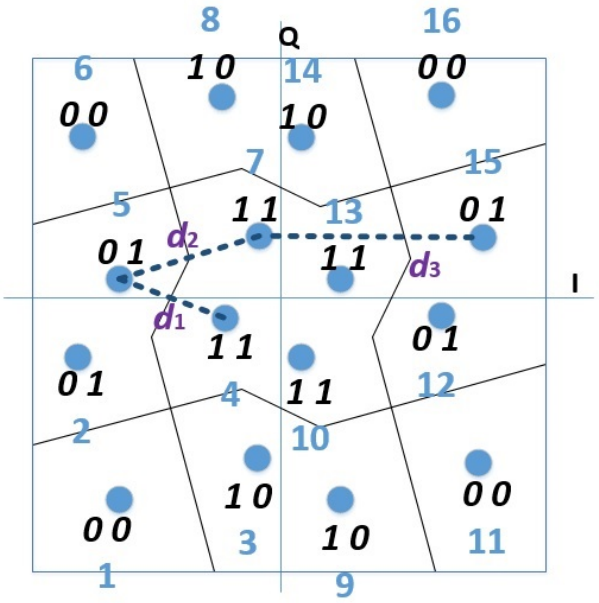

FIGURE 3.9: The XOR mapping, the relative phase is around 0

According to Fig 3.9 (the mapping situation when XOR mapping is chosen and the relative fading is around 0, which will have the same SER with the situation that the reversed XOR mapping is chosen and the relative fading is around π), we can divide all constellation points into 3 groups as table 3.3 shows:

label	Error Distances
1, 6, 11, 16	$\{d_2, d_2\}$
2, 8, 9, 15	$\{d_2, d_2, d_3\}$
4, 7, 10, 13 3, 5, 12, 14	$\{d_2, d_2, d_1, d_3\}$

TABLE 3.3: Constellation points and network coding for the XOR mapping

where blue numbers in Fig 3.9 from 1 to 16 representing different constellation points as shown in table 3.4, and SNR is the signal (at each transmitter) to noise (at the relay) ratio. Thus we have the symbol error rate P_S as:

$$P_S = \frac{1}{16}\left\{2 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_1}{2}\sqrt{2\text{SNR}}\right) + 12Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right)\right\} \tag{3.1}$$

$$= 2Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|j(h_B - h_A) + h_B|\sqrt{\text{SNR}}\right) + \frac{3}{4}Q\left(|h_A|\sqrt{\text{SNR}}\right) \tag{3.2}$$

X_R	label	superimposed NC value	X_R	label	superimposed NC value
0 0 0 0	1	0 0	1 0 0 0	9	1 0
0 0 0 1	2	0 1	1 0 0 1	10	1 1
0 0 1 0	3	1 0	1 0 1 0	11	0 0
0 0 1 1	4	1 1	1 0 1 1	12	0 1
0 1 0 0	5	0 1	1 1 0 0	13	1 1
0 1 0 1	6	0 0	1 1 0 1	14	1 0
0 1 1 0	7	1 1	1 1 1 0	15	0 1
0 1 1 1	8	1 0	1 1 1 1	16	0 0

TABLE 3.4: The superimposed network coding value for the XOR mapping

Similarly, according to table 3.5, we could have the constellation diagram as shown in Fig 3.10 (the mapping situation when the XOR mapping is chosen and the relative fading is around $\frac{\pi}{2}$ has the same error rate). Thus, we can obtain the error distances as table 3.6 shows:

X_R	label	superimposed NC value	X_R	label	superimposed NC value
0 0 0 0	1	0 0	1 0 0 0	9	1 0
0 0 0 1	2	1 0	1 0 0 1	10	0 0
0 0 1 0	3	0 1	1 0 1 0	11	1 1
0 0 1 1	4	1 1	1 0 1 1	12	0 1
0 1 0 0	5	0 1	1 1 0 0	13	1 1
0 1 0 1	6	1 1	1 1 0 1	14	0 1
0 1 1 0	7	0 0	1 1 1 0	15	1 0
0 1 1 1	8	1 0	1 1 1 1	16	0 0

TABLE 3.5: The superimposed network coding value for the reversed XOR mapping

label	Error Distances
1, 6, 11, 16	$\{d_2, d_2\}$
2, 8, 9, 15	$\{d_2, d_2, d_3\}$
3, 5, 12, 14	$\{d_2, d_2, d_1, d_3\}$
4, 7, 10, 13	$\{d_2, d_2, d_1, d_3, d_4\}$

TABLE 3.6: Constellation points and network coding for the reversed XOR mapping

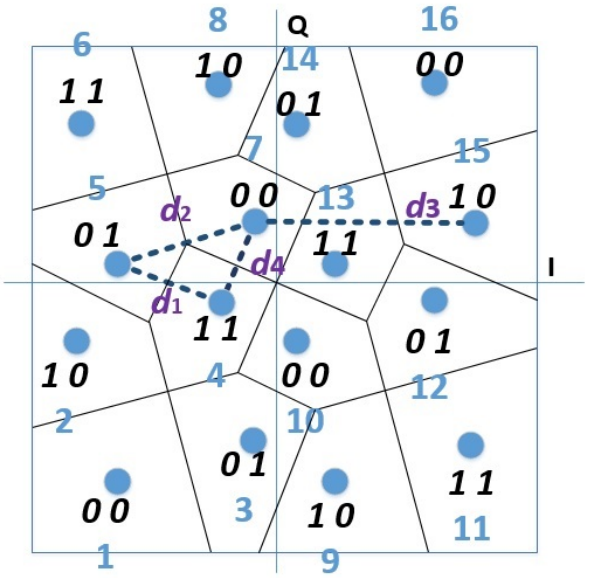

FIGURE 3.10: The reversed XOR mapping, the relative phase is around 0

Now we can work out the error rate for the reversed XOR mapping in this particular channel fading state:

$$P_S = \frac{1}{16}\Bigg\{2\times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right)+8Q\left(\frac{d_1}{2}\sqrt{2\text{SNR}}\right)+12Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right)$$
$$+2\times 4Q\left(\frac{d_4}{2}\sqrt{2\text{SNR}}\right)\Bigg\} \tag{3.3}$$
$$=2Q\left(|h_B|\sqrt{\text{SNR}}\right)+\frac{1}{2}Q\left(|j(h_B-h_A)+h_B|\sqrt{\text{SNR}}\right)+\frac{3}{4}Q\left(|h_A|\sqrt{\text{SNR}}\right)$$
$$+\frac{1}{2}Q\left(|h_A+h_B|\sqrt{\text{SNR}}\right) \tag{3.4}$$

Finally we analyse the situation when the relative phase is around $\frac{\pi}{4}$ according to Fig 3.11 (the situation when the reversed XOR mapping is chosen at this fading state) as table 3.7 shows:

label	Error Distances
1, 2, 6, 8 9, 11, 15, 16	$\{d_2, d_2, d_3\}$
3, 4, 5, 7 10, 12, 13, 14	$\{d_2, d_2, d_1, d_3\}$

TABLE 3.7: Constellation points and network coding for the reversed XOR mapping

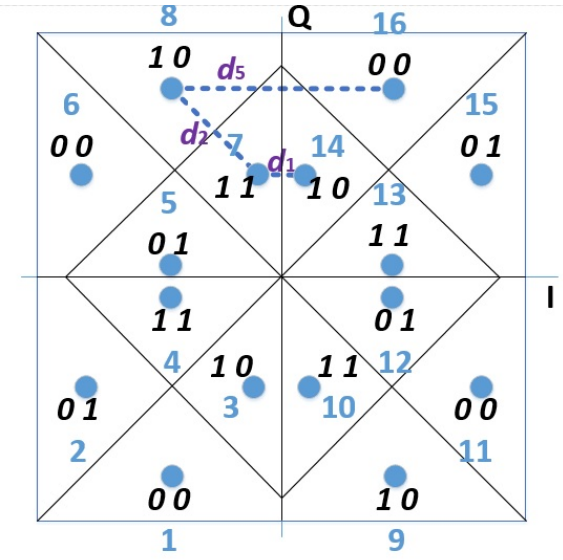

FIGURE 3.11: The XOR mapping, the relative phase is around 45^o

Then the SER for the $\frac{\pi}{4}$ case is:

$$P_S = \frac{1}{16}\left\{2 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_1}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right)\right\} \tag{3.5}$$

$$= 2Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|j(h_B - h_A) + h_B|\sqrt{\text{SNR}}\right) + + \frac{1}{2}Q\left(|h_A|\sqrt{\text{SNR}}\right) \tag{3.6}$$

3.2.2 The Anti-$\frac{\pi}{4}$ Mapping

In this section, we take a look at the constellation of the anti-$\frac{\pi}{4}$ mapping when the relative phase is around $\frac{\pi}{4}$ and 0 (the case of $\frac{\pi}{2}$ is not considered here because it will have the same theoretical SER as the 0 case due to symmetry)

First, with the help of table 3.8, we could analyse the situation when the relative phase is around $\frac{\pi}{4}$, which gives rise to Fig.3.12

X_R	label	superimposed NC value	X_R	label	superimposed NC value
0 0 0 0	1	0 0	1 0 0 0	9	0 1
0 0 0 1	2	1 1	1 0 0 1	10	1 0
0 0 1 0	3	1 0	1 0 1 0	11	1 1
0 0 1 1	4	0 1	1 0 1 1	12	0 0
0 1 0 0	5	0 1	1 1 0 0	13	0 0
0 1 0 1	6	1 0	1 1 0 1	14	1 1
0 1 1 0	7	1 1	1 1 1 0	15	1 0
0 1 1 1	8	0 0	1 1 1 1	16	0 1

TABLE 3.8: Constellation points and network coding for the an-$\frac{\pi}{4}$ mapping

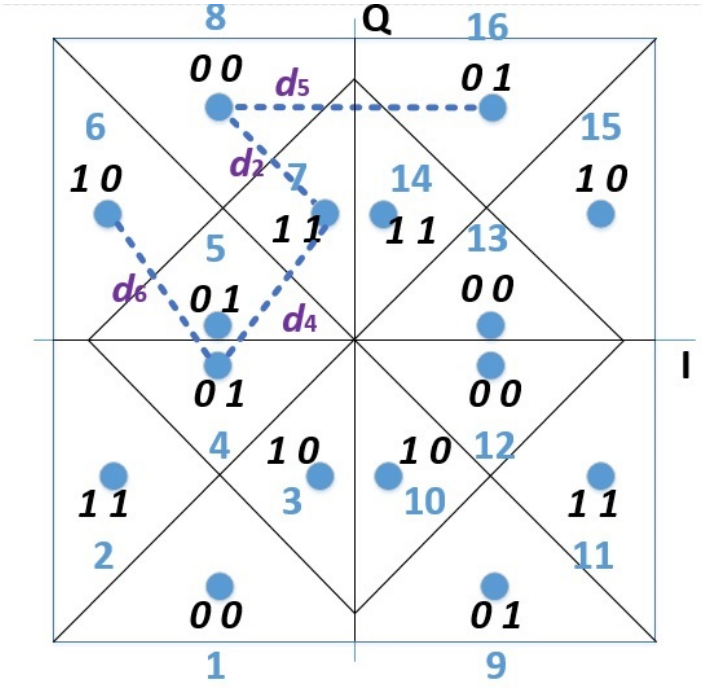

FIGURE 3.12: The anti-$\frac{\pi}{4}$ mapping when the relative phase is around 45^o

Thus, we can obtain the error distances as table 3.9 shows:

label	Error Distances
1, 2, 6, 8 9, 11, 15, 16	$\{d_2, d_2, d_3\}$
3, 4, 5, 7 10, 12, 13, 14	$\{d_2, d_2, d_4, d_3\}$

TABLE 3.9: The anti-$\frac{\pi}{4}$ mapping when the relative phase is around 45^o

According to table 3.9, the SER for the anti-$\frac{\pi}{4}$ mapping is

$$\begin{aligned} P_S = \frac{1}{16}\Bigg\{ & 3 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right) \\ & + 8Q\left(\frac{d_4}{2}\sqrt{2\text{SNR}}\right) + 2 \times 8Q\left(\frac{d_5}{2}\sqrt{2\text{SNR}}\right)\Bigg\} \\ & = 3Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|h_A|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|h_A + h_B|\sqrt{\text{SNR}}\right) \\ & + Q\left(|jh_A - h_B|\sqrt{\text{SNR}}\right) \end{aligned} \tag{3.7}$$

label	Error Distances
1, 6, 11, 16	$\{d_2, d_2\}$
4, 7, 10, 13	$\{d_2, d_2, d_4, d_5\}$
2, 8, 9, 15 3, 5, 12, 14	$\{d_2, d_2, d_4\}$

TABLE 3.10: The anti-$\frac{\pi}{4}$ mapping, the relative phase is around 0

Next, according to the look-up table 3.10, we can build the constellation shown in Fig 3.13:

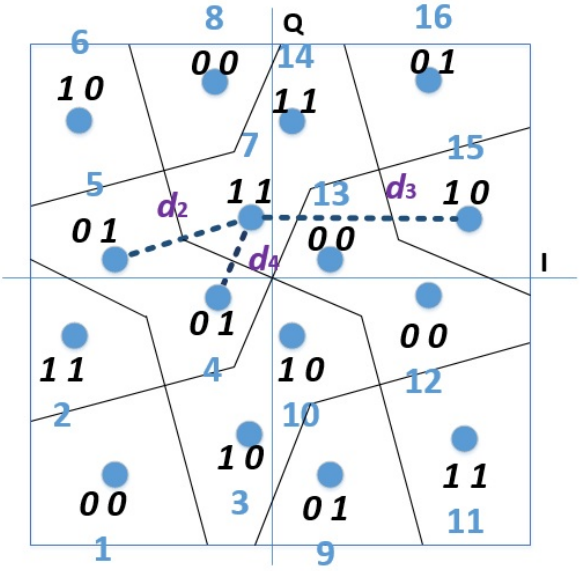

FIGURE 3.13: The anti-$\frac{\pi}{4}$ mapping, the relative phase is around 0

When the relative phase is about 0,

$$P_S = \frac{1}{16}\left\{3 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 16Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right)\right\}$$
$$= 3Q\left(|h_B|\sqrt{\text{SNR}}\right) + Q\left(|h_A + h_B|\sqrt{\text{SNR}}\right) \quad (3.8)$$

3.2.3 The Adaptive Mapping

The adaptive mapping method selects the optimum mapping according to the channel fading states. So, the XOR mapping is chosen when the relative phase is around 0, and the reversed XOR mapping is chosen when the relative phase is around $\frac{\pi}{2}$. The threshold which differentiates them is the phase fading of $\frac{\pi}{4}$.

The anti-$\frac{\pi}{4}$ mapping may be optimum when the relative phase is around $\frac{\pi}{4}$. However, this mapping eliminates error between the most adjacent pairs of points at the cost of introducing more errors between the other adjacent points labels within the same constellation, which may increase the BER. Thus, we need to ensure that:

$$P_{S-adaptive} = P_S(anti\frac{\pi}{4}) < P_S(XOR) \quad (3.9)$$

where $P_{S-adaptive}$ is the SER for the adaptive mapping, $P_S(anti\frac{\pi}{4})$ is the SER for the anti $\frac{\pi}{4}$ mapping, and $P_S(XOR)$ is the SER for the XOR mapping (the expression will be the same if we choose the reversed XOR mapping). We here ignore the influence of d_3 (actually d_3 can be cancel out in the equation) and d_5 as they are far larger than d_1 or d_2. Then

$$3Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|h_A|\sqrt{\text{SNR}}\right) < 2Q\left(|h_B|\sqrt{\text{SNR}}\right)$$
$$+ \frac{1}{2}Q\left(|h_A|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|h_A + jh_B|\sqrt{\text{SNR}}\right) \quad (3.10)$$

or

$$\begin{aligned}
&\frac{1}{16}\left\{3\times 16Q\left(\frac{d_2}{2}\sqrt{2\mathrm{SNR}}\right)+8Q\left(\frac{d_3}{2}\sqrt{2\mathrm{SNR}}\right)\right\}\\
&<\frac{1}{16}\left\{2\times 16Q\left(\frac{d_2}{2}\sqrt{2\mathrm{SNR}}\right)+8Q\left(\frac{d_1}{2}\sqrt{2\mathrm{SNR}}\right)\right.\\
&\left.+8Q\left(\frac{d_3}{2}\sqrt{2\mathrm{SNR}}\right)\right\}
\end{aligned} \tag{3.11}$$

As can be seen, d_1 is far less than d_4 or d_5, so we can ignore their terms without losing a great accuracy. Thus the trade-off is:

$$P_{S-adaptive}=P_{S-XOR}+\frac{16Q\left(\frac{d_2}{2}\sqrt{2\mathrm{SNR}}\right)-8Q\left(\frac{d_1}{2}\sqrt{2\mathrm{SNR}}\right)}{16} \tag{3.12}$$

If we take the approximate value, then:

$$d_2\approx 2d_1 \tag{3.13}$$

which is the threshold for picking this particular mapping. Thus the SER in this situation ($d_2<2d_1$) is:

$$\begin{aligned}
P_S&=\frac{1}{16}\left\{3\times 16Q\left(\frac{d_2}{2}\sqrt{2\mathrm{SNR}}\right)+8Q\left(\frac{d_3}{2}\sqrt{2\mathrm{SNR}}\right)\right\}\\
&=3Q(|h_B|\sqrt{S})+\frac{1}{2}Q\left(|h_A+h_B|\sqrt{\mathrm{SNR}}\right)
\end{aligned} \tag{3.14}$$

For all other amplitude fading situations, when the relative phase fading is around 0 or $\frac{\pi}{2}$, the SER is:

$$P_S = \frac{1}{16}\left\{2 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_1}{2}\sqrt{2\text{SNR}}\right)\right\} \tag{3.15}$$

$$= 2Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|\,(h_A + h_B) + jh_B|\sqrt{\text{SNR}}\right) \tag{3.16}$$

and when the relative phase is around $\frac{\pi}{4}$, the SER is:

$$P_S = \frac{1}{16}\left\{2 \times 16Q\left(\frac{d_2}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_1}{2}\sqrt{2\text{SNR}}\right) + 8Q\left(\frac{d_3}{2}\sqrt{2\text{SNR}}\right)\right\} \tag{3.17}$$

$$= 2Q\left(|h_B|\sqrt{\text{SNR}}\right) + \frac{1}{2}Q\left(|h_A + h_B|\sqrt{\text{SNR}}\right) + + \frac{1}{2}Q\left(|h_A + jh_B|\sqrt{\text{SNR}}\right) \tag{3.18}$$

Since the expressions are different, the estimation of the SER behaviour over all fading channels could only be averaged using the Monte-Carlo simulation. However, we don't necessarily need the average SER in the process of the later system-level simulation. The averaged error rate is only needed for an rough estimation of the final error rate at the destination.

3.3 Simulation result

We introduce the simulation result as well as the comparison of the simulation and the theory here. Across all figures in this section, we can find that all theoretical curves may stay apart from its related simulated curves at lower SNR, but will merge with them at higher SNR, showing that our theory give rise to an upper bound of the performance analysis.

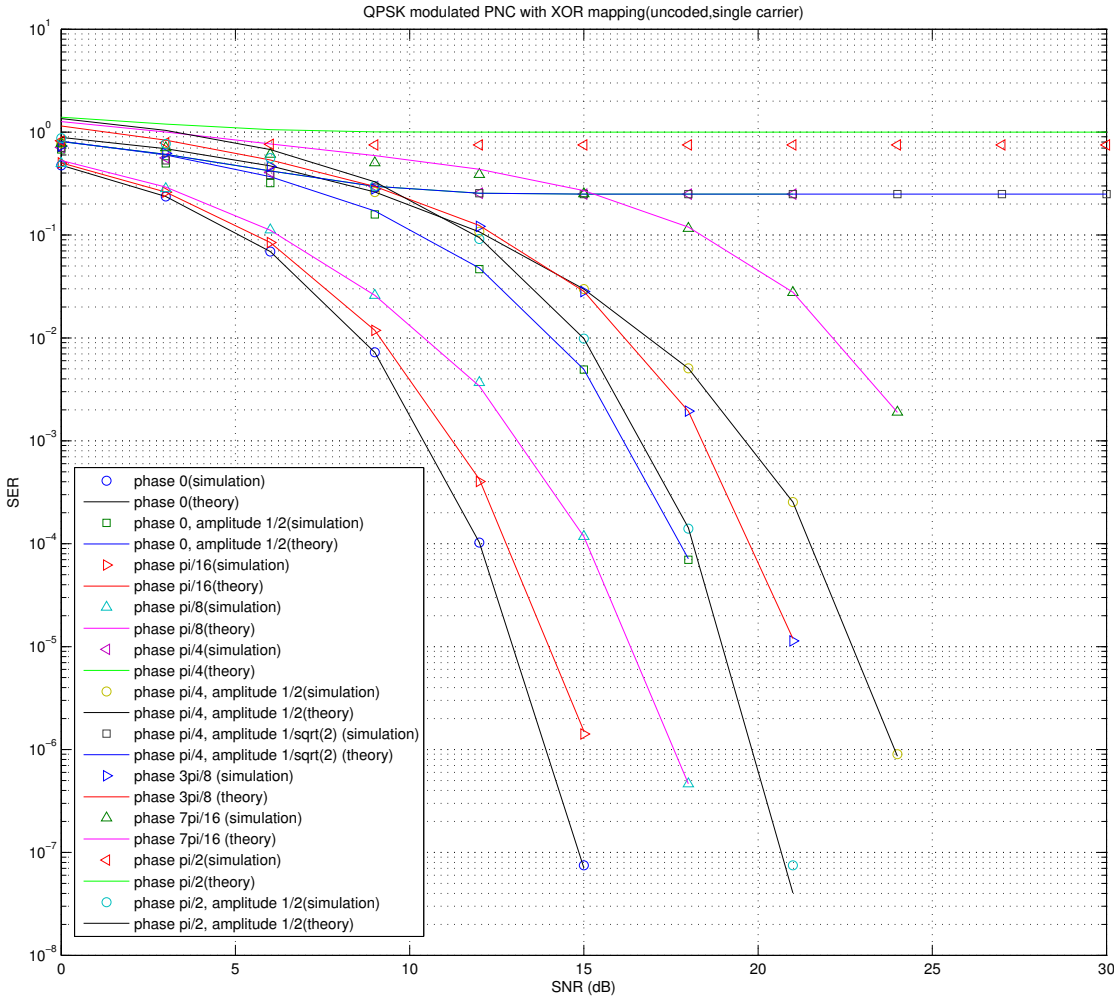

FIGURE 3.14: The XOR mapping

The performance of the XOR mapping over several singular fadings can be seen from Fig 3.14. Note: unless specified, the amplitude fading is set to 1 as default.

This mapping perform quite well when phase is around 0, and the performance is getting worse along with the increase of phases (up to $\frac{\pi}{2}$). Interestingly, the SER performance gets better when amplitude fading gets lower (when phase shift is $\frac{\pi}{2}$, the SER of the curve with an amplitude fading of 1/2 perform better than that of 1, which is the default value). This is because distance d_1 and d_3 become far larger than d_2 when amplitude fading gets lower, and can be ignored.

At lower SNR(SNR$<$ 3dB), the theoretical curve locate above the simulation points, and finally they merge at higher SNR, showing that the theoretical SER is an upper bound.

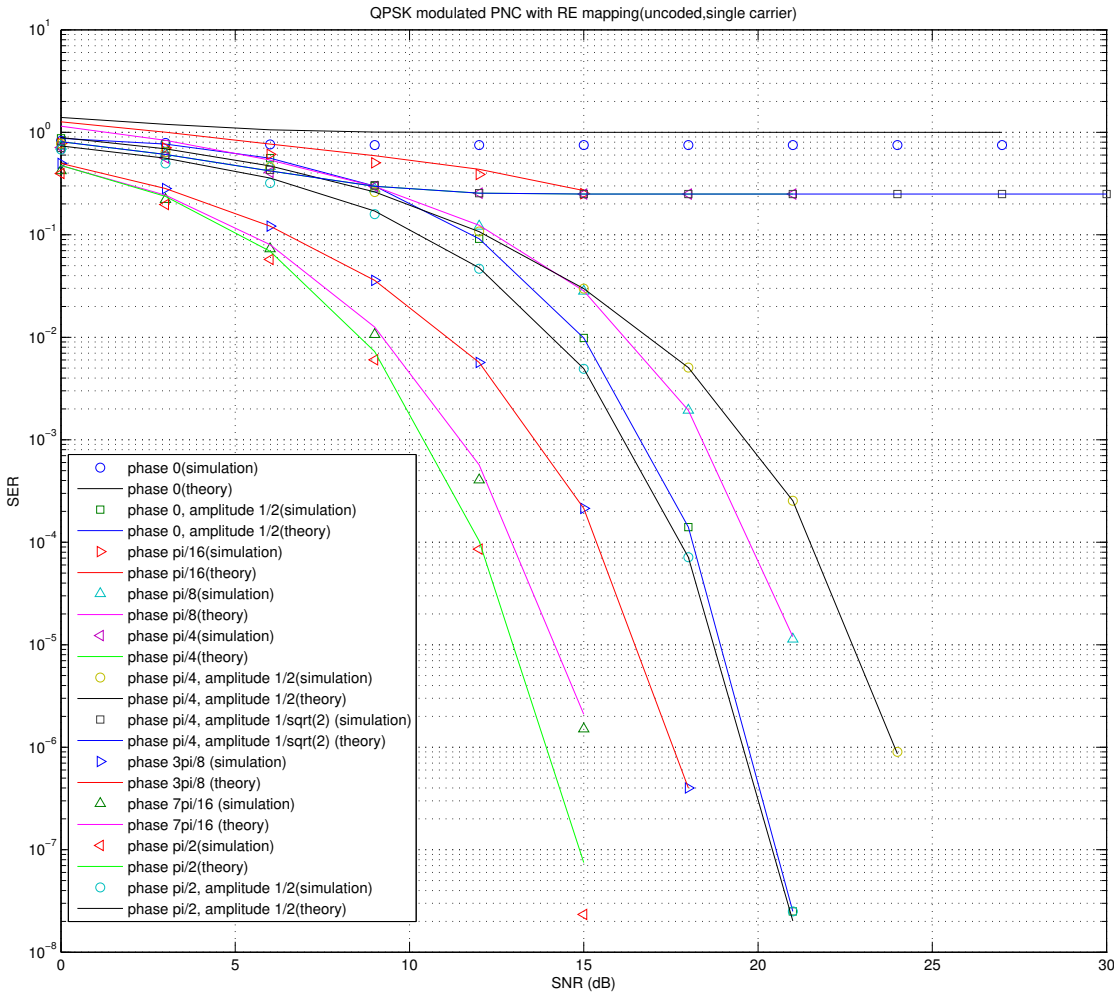

FIGURE 3.15: The reversed XOR mapping

The performance of the reversed XOR mapping is shown in Fig 3.15 above. If we compare it with Fig 3.14, then we can conclude that the performance resulted from the reversed XOR mapping and the XOR mapping is symmetric about $\frac{\pi}{4}$.

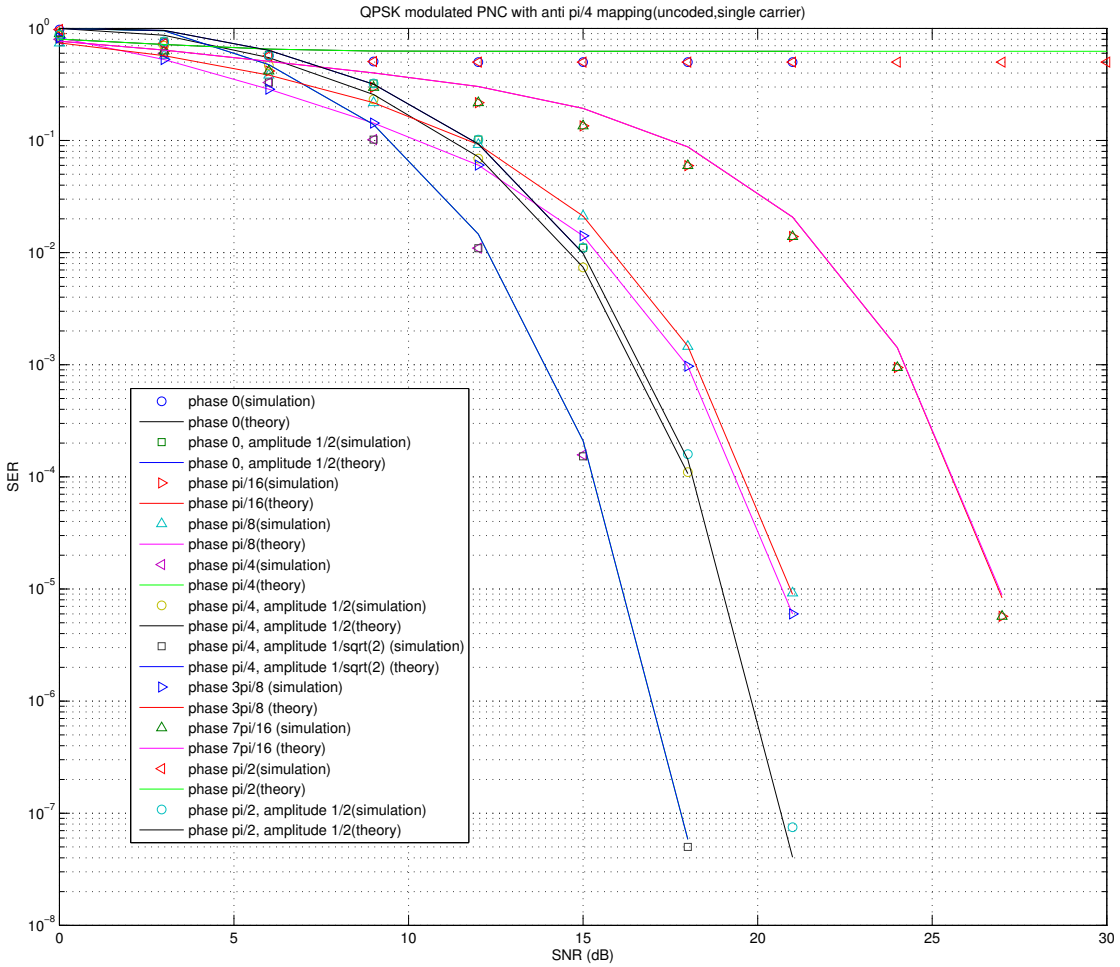

FIGURE 3.16: The anti-$\frac{\pi}{4}$ mapping

The performance of the anti-$\frac{\pi}{4}$ is shown in Fig 3.16. As can be seen, its SER performance reaches optimum when the the phase is $\frac{\pi}{4}$. But when the amplitude fading gets smaller, the performance becomes worse, because the addition error with distance d_2 will become more important than d_1, which is explained in the section "Adaptive mapping".

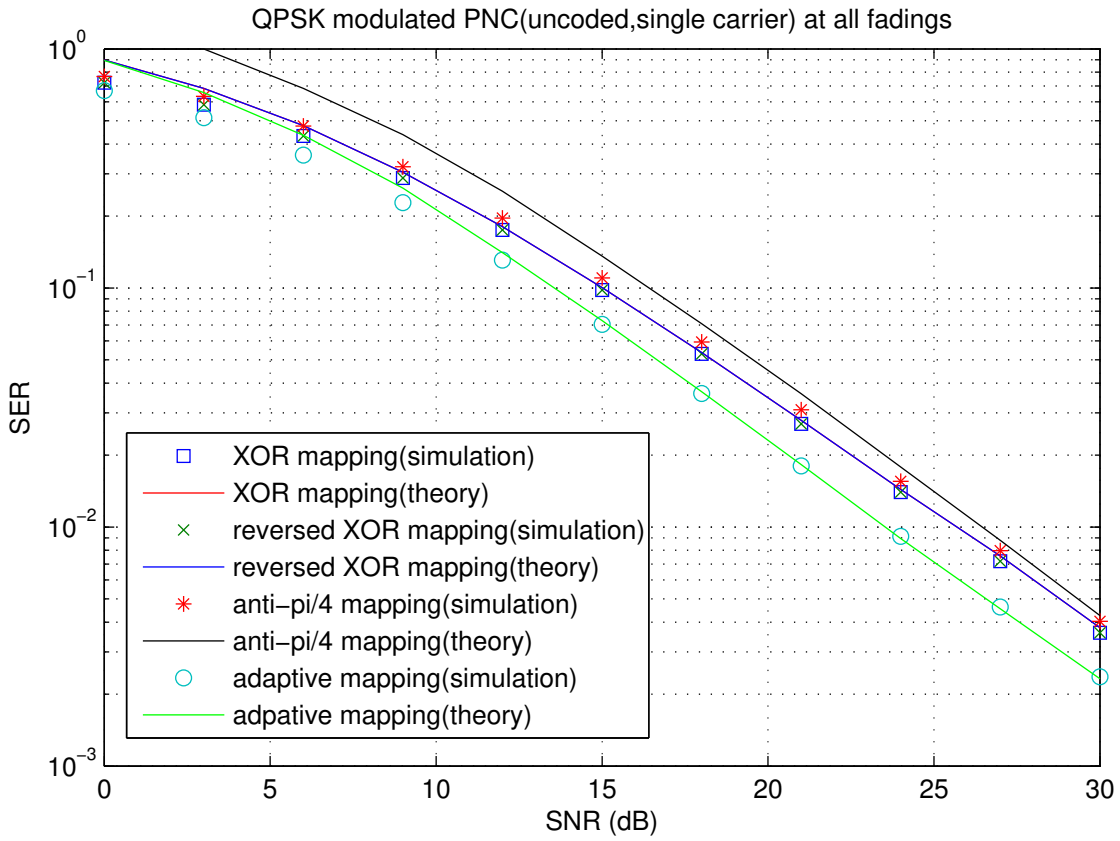

FIGURE 3.17: All mappings and all fadings

Finally, we simulate the situation across all fading states by means of Monte-Carlo simulation shown in Fig 3.17. We notice that :

- The adaptive mapping perform the best (at least 2dB better than any fixed mappings) as expected.
- The XOR/reversed XOR mapping performs the same across all singular fadings due to symmetry.
- The anti-$\frac{\pi}{4}$ mapping performs the worst across all fadings, showing that it is very vulnerable outside its particular favoured fading state.

3.4 Summary

We build the three mappings for the QPSK modulated PNC MAC phase, and derive their SER upper bound across all relative phases and amplitude. Also,

we have simulate and calculate the theoretical SER performance of the adaptive mapping with the help of the Monte-Carlo simulation.These upper bounds are not strictly mean to be the information-theoretic limits, but are practical in our project, especially in the lator system-level simulation.

Chapter 4

System-level Simulation

In this section we aim to perform a system-level simulation of a multi-node system. We start from a 5-node system (2 users and 2 relays) shown in Fig. 4.1 and further extend it to a 13-node system (4 users and 8 relays) shown in Fig. 4.2. Note: all dash arrows represent the noisy wireless transmission channels (AWGN channels), and all solid arrows represent the point-to-point noise-less wired link channels.

4.1 System Model

4.1.1 5-node system

To simplify some explanations, we define the vertical and horizontal concatenation of matrices here:

- if $A = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and
- $B = \begin{bmatrix} 2 & 2 \end{bmatrix}$ then

- $\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ and
- $[A\ B] = \begin{bmatrix} 1 & 1 & 2 & 2 \end{bmatrix}$

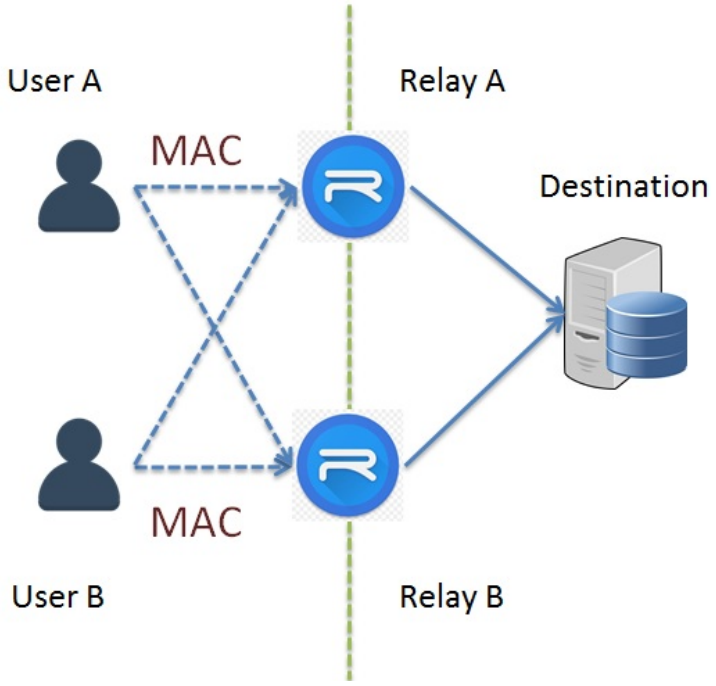

FIGURE 4.1: Example system for system-level simulation: 5 nodes

We start our research from the 5-node system shown in Fig. 4.1. As can be seen, the left side of the dividing line perform similar to the system model of the previous chapter except users transmit to 2 relays rather than 1, and the relays will decode the superimposed binary symbol as I_{RA} and I_{RB} respectively with the help of the mapping they applied on each relay (here we assume the mapping matrix are M_A and M_B).

Then the process going into the right side of the dividing line. I_{RA} and I_{RB} are transmit to the final destination one by one through noiseless wired channel which we assume that there are no attenuations on these channels.

Then the destination will perform a matrix multiplication to retrieve the original binary symbol I_A and I_B. Instead of the superimposed symbol at the relays,

the final destination will recover a concatenated symbol I. This symbol will be compared with $\begin{bmatrix} I_A & I_B \end{bmatrix}$ to obtain the overall error rate. This is performed by a matrix multiplication as:

$$I = X_D G^{-1} \tag{4.1}$$

Where $X_D = \begin{bmatrix} I_{RA} \\ I_{RB} \end{bmatrix}$ and $G = \begin{bmatrix} M_A & M_B \end{bmatrix}$

However, we need to make sure that the generator matrix G is full rank in order to perform the inverse operation. Thus, we need to choose the mappings wisely. In other word, we choose the best realizable pairs of mappings. As we introduced in the last chapter, each relay has 3 mappings to choose (the XOR, the reversed XOR and the anti-$\frac{\pi}{4}$), and thus there are altogether $3^2 = 9$ combinations of mapping matrices, which is enormous. Recalling the settings in the last chapter(M_1 is the XOR mapping, M_2 is the reversed XOR mapping, and M_3 is the anti-$\frac{\pi}{4}$ mapping), we could analyse their ranks separately in order to filter out the non-realizable combinations (the combinations which are not full-rank):

combination metrics	rank
$\begin{bmatrix} M_1 & M_1 \end{bmatrix}$	2
$\begin{bmatrix} M_1 & M_2 \end{bmatrix}$	3
$\begin{bmatrix} M_1 & M_3 \end{bmatrix}$	4
$\begin{bmatrix} M_2 & M_1 \end{bmatrix}$	2
$\begin{bmatrix} M_2 & M_2 \end{bmatrix}$	3
$\begin{bmatrix} M_2 & M_3 \end{bmatrix}$	4
$\begin{bmatrix} M_3 & M_1 \end{bmatrix}$	4
$\begin{bmatrix} M_3 & M_2 \end{bmatrix}$	4
$\begin{bmatrix} M_3 & M_3 \end{bmatrix}$	2

TABLE 4.1: Rank of all mapping combination matrices

Observing from table 4.1 above, the full-rank ($rank = 4$) combination metrics are: $[M_1; M_3]$, $[M_2; M_3]$, $[M_3; M_1]$ and $[M_3; M_2]$, which means we only need to choose the best mapping among these 4 without doing anymore filtering or looping.

As a result, the destination will deal with the calculation of total error rate and feedback the relays of the best realizable mapping selection. We will further discuss about how these are realized in the next section.

4.1.2 13-node system

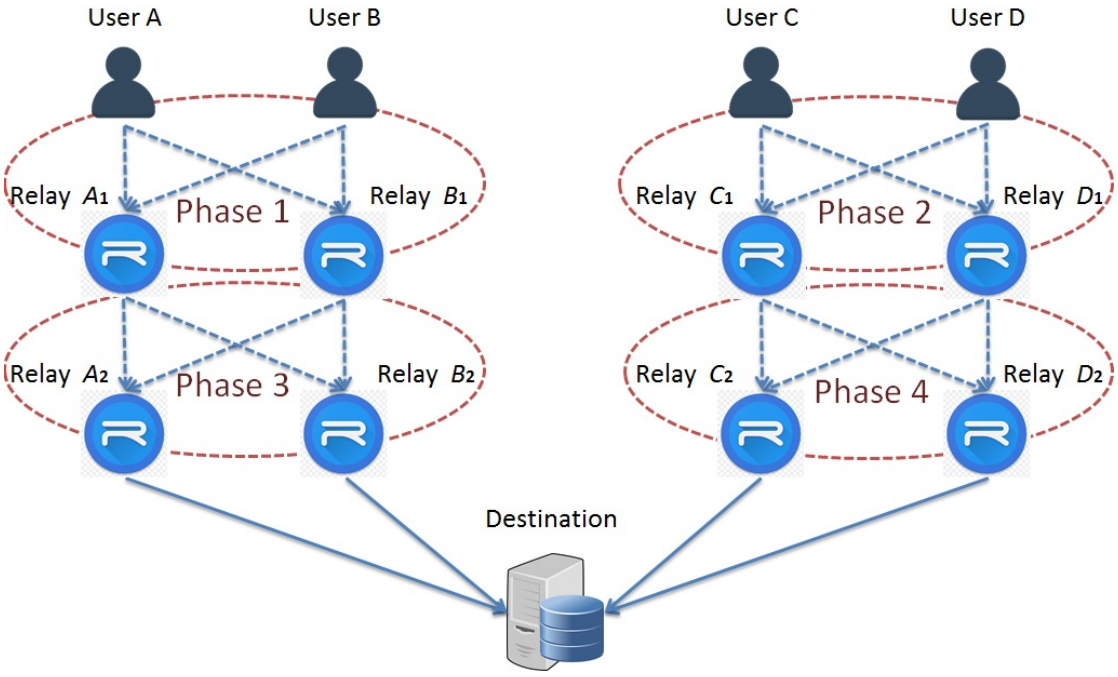

FIGURE 4.2: Example system for system-level simulation: 13 nodes

We divide the 13-node system into different phases. The word phase correspond to different stages of transmission. There are 4 wireless transmission stages in the system model, and we name them from phase 1 to 4 respectively. In each phase, the two source nodes (user/relay act as transmitter) transmit their own sequences to two separate destination nodes (relays that act as the destination) on independent channels simultaneously. Relays can either be source nodes or destination nodes. Take relay A_1 as example: In phase 1 it is defined as a destination node, but in phase 3 it is the source node/transmitter.

We assign different time slots for different phases in order to avoid the interferences between phases, thus the transmitting sequence is from phase 1 to 4 in turns. In each phase, both sources transmit a QPSK stream (an $N \times 1$ matrix, where N is the total symbols to be transmitted) to the relays like the PNC. Then the relays build the mapping matrix based on the fading state of the pairs of channels and then decode the received symbols into the concatenated vectors X accordingly. This is the same as that defined in the system model.

The mapping matrices are all 4×2. As introduced in the previous section, there are 3 mappings: M_1 (the XOR mapping), M_2 (the reversed XOR mapping), and M_3 (the anti-$\frac{\pi}{4}$ mapping). In phase 1, if M_{A1} represents the mapping that destination relay A_1 chooses, and X_1 is the de-mapped concatenated vector, then we can define a matrix multiplication to recover the superimposed symbol as:

$$I_{A1} = X_{A1} M_{A1} \tag{4.2}$$

Both relays then pass then decoded superimposed symbols to the relays (A_2 and B_2) in a subsequent phase (phase 3) together with their mapping matrices. Phase 2 proceeds in the same way as phase 1, and similarly in phases 3 and 4. Relays A_1 and B_1, and C_1 and D_1 forward the superimposed symbols to relays A_2 and B_2, C_2 and D_2, which again choose appropriate mappings to recover new superimposed symbols. We name the mappings applied on these relays as M_{A2}, M_{B2}, M_{C2}, M_{D2} respectively.

If we ignore the distortion of noise, then the expected de-mapped concatenated vectors are the same in the same phase as:

$$X_{A1} = X_{B1} \tag{4.3}$$

and

$$X_{C1} = X_{D1} \tag{4.4}$$

The relays in phase 1 and 2 will act as transmitters in phase 3 and 4. Thus, the transmitted sequences will be like:

$$I_{A1} = I_{B1} = X_{A1} \begin{bmatrix} M_{A1} & M_{B1} \end{bmatrix} \tag{4.5}$$

and

$$I_{C1} = I_{D1} = X_{C1} \begin{bmatrix} M_{C1} & M_{D1} \end{bmatrix} \tag{4.6}$$

Thus in phase 3 and 4, with the same assumption, we have:

$$X_{A2} = X_{B2} = X_{A1} \begin{bmatrix} M_{A1} & M_{B1} \end{bmatrix} \tag{4.7}$$

and

$$X_{C2} = X_{D2} = X_{C1} \begin{bmatrix} M_{C1} & M_{D1} \end{bmatrix} \tag{4.8}$$

Similar to equation (4.7) and (4.8), we have :

$$I_{A2} = I_{B2} = X_{A2} \begin{bmatrix} M_{A2} & M_{B2} \end{bmatrix} \tag{4.9}$$

and

$$I_{C2} = I_{D2} = X_{C2} \begin{bmatrix} M_{C2} & M_{D2} \end{bmatrix} \tag{4.10}$$

Finally, the 4 symbols are transmitted to the destination one by one. We concatenate these signals into an $N \times 8$ matrix (because each of them is $N \times 2$ in size)

X:

$$\begin{aligned} X_D &= \begin{bmatrix} I_{A2} & I_{B2} & I_{C2} & I_{D2} \end{bmatrix} \\ &= \begin{bmatrix} X_{A2}M_{A2} & X_{A2}M_{B2} & X_{C2}M_{C2} & X_{C2}M_{D2} \end{bmatrix} \\ &= \begin{bmatrix} X_{A2}(M_{A2} & M_{B2}) & X_{C2}(M_{C2} & M_{D2}) \end{bmatrix} \end{aligned} \tag{4.11}$$

Using the result of (4.9) and (4.10), we can deduce a generator matrix (formed by the combination of the mappings at each relay) as:

$$G = \begin{bmatrix} [M_{A1}\ M_{B1}] \times [M_{A2}\ M_{B2}] & 0_{4,4} \\ 0_{4,4} & [M_{C1}\ M_{D1}] \times [M_{C2}\ M_{D2}] \end{bmatrix} \tag{4.12}$$

Where $0_{4,4}$ is a zero matrix of size 4×4. So the final step is to recover the original information from each user as a concatenated matrix $X_{A\ D}$. This is realized by a matrix multiplication:

$$I = X_D G^{-1} \tag{4.13}$$

However, we need to assume that the final generator matrix is full-rank so as to perform a inverse matrix operation. In order to achieve that, we need to ensure that every concatenated mapping matrix is full rank, which means all the 4×4 matrices :

$$[M_{A1}\ M_{B1}], [M_{A2}\ M_{B2}], [M_{C1}\ M_{D1}], [M_{C2}\ M_{D2}]$$

are full-rank. In this way, we just need to do the mapping selection 4 times given that all these matrices are independent to the others (because all channels are independent). In other word, we have successfully reduced the overall complexity.

4.2 Mapping Selection

This system aims to pick the realistically optimum mapping at all phases. As a result, the pairs of relays act as the receiver will calculate the theoretical SERs for all three mappings under the current fading states, and pass them to one of destination nodes in another phase. Again take phase 1 as example: relay A_1 and B_1 will calculate the all 3 SER values using the theory we derived in last chapter, and pass them to either A_2 or B_2. Then the destination will form a set of SER combinations (error occurs when either destination relay decodes its superimposed symbol, which is close to the direct summation of the SER at both relays), and sort it in the order of low to high. Note the exhaustive search is needed here to distinguish the optimum mapping combinations whose related mapping matrix is full rank. As a result, we design a flow chart to explain the mapping selection in Fig 4.3:

In the flow chart, we can see that an exhaustive search is needed in each destination node, which is represented in a loop. From the 5-node system, we already know that there are altogether 4 possible realizable mapping combos to be select in each phase. Thus, we can solve the exhaustive search in algorithm 2.

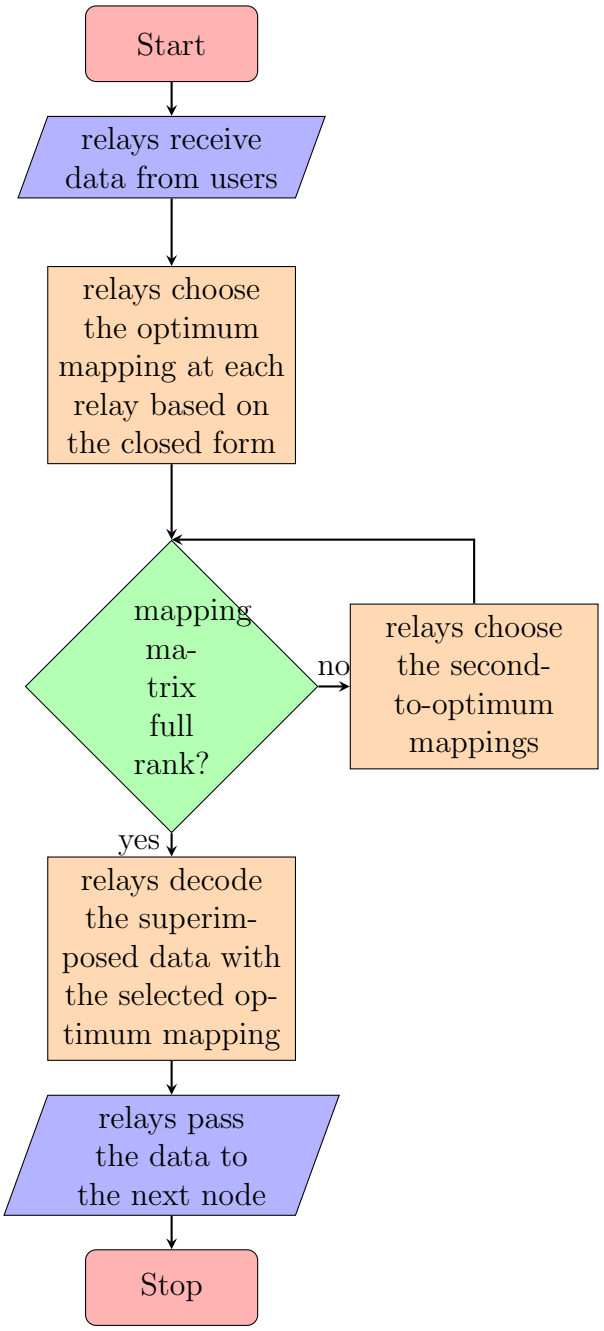

FIGURE 4.3: System scenario in flow chart

In algorithm 2, we have $P_{[M_1,M_3]}$ as the combine SER when relay 1 choose the XOR mapping, and relay 2 choose the $anti-\frac{\pi}{4}$ mapping, which applies to other mapping combinations as well. S_d is a boolean variable showing whether the destination received the mapping matrices from the relays. S_r is a boolean variable showing whether the relays receives signal from the tranmitters, $P_{S(1)[M_1]}$ represent the SER at one relay in a sinlge phase if the XOR method is applied, and this is similar for

Algorithm 2 The improved mapping scenario

while $S_r == 1$ **do**
  for $i = 1, i < 2, i++$ **do**
    $P_S(i,1) \leftarrow P_{S(i)[M_1]}$
    $P_S(i,2) \leftarrow P_{S(i)[M_2]}$
    $P_S(i,3) \leftarrow P_{S(i)[M_3]}$
  end for
  while $S_d == 1$ **do**
    $P_{[M_1\ M_3]} \leftarrow P_{S(1)[M_1]} + P_{S(2)[M_3]}$
    $P_{[M_2\ M_3]} \leftarrow P_{S(1)[M_2]} + P_{S(2)[M_3]}$
    $P_{[M_3\ M_1]} \leftarrow P_{S(1)[M_3]} + P_{S(2)[M_1]}$
    $P_{[M_3\ M_2]} \leftarrow P_{S(1)[M_3]} + P_{S(2)[M_2]}$
  end while
  SERM $= sort[P_{[M_1\ M_3]}, P_{[M_2\ M_3]}, P_{[M_3\ M_1]}, P_{[M_3\ M_2]}]$
  loop
    if SERM(1) $== P_{[M_1\ M_3]}$ **then**
      $M = [M_1\ M_3]$
    else if SERM(1) $== P_{[M_2\ M_3]}$ **then**
      $M = [M_2\ M_3]$
    else if SERM(1) $== P_{[M_3\ M_1]}$ **then**
      $M = [M_3\ M_1]$
    else
      $M = [M_3\ M_2]$
    end if
  end loop
end while

the other relay and the other mappings. $SERM$ is a vector of all the combinations of SER, f is a function which can fetch the mapping matrix from the concatenated SER, and M is the optimum realizable mapping.

This pre-filtered exhaustive search may introduce more complexities in higher-order modulations, but we still prefer using it here as the maximum numbers of search is still acceptable to the devices.

4.3 PER calculation

In this section, we present a symbol-level simulation over the two-user-two-relay model for the end-to-end PER. The system settings are the same as for the BER simulation except the output is the PER instead of BER, and the packet size N

is set to be 100 (100 bits in one packet/frame). Also, with the help of the closed form derived early, we can estimate the PER value as:

$$P_P = 1 - (1 - P_{S-A})^N \tag{4.14}$$

$$\approx P_{S-A} \times N \tag{4.15}$$

where P_P is the end-to-end PER, P_{S-A} is the averaged SER at either relay with the adaptive mapping, and N is the packet size 100. Again, we notice that the SER used here is averaged over Monte-Carlo simulation. However, this is not necessary as the system-level simulation does not require the average SER, and this theoretical closed form is only used to estimate the system behaviour.

4.4 Simulation result

For the simulation of PER, which is shown in Fig 4.4, the theoretical curve acts as the upper bound as expected, showing that the system-level simulation works effective for complicated multi-node systems.

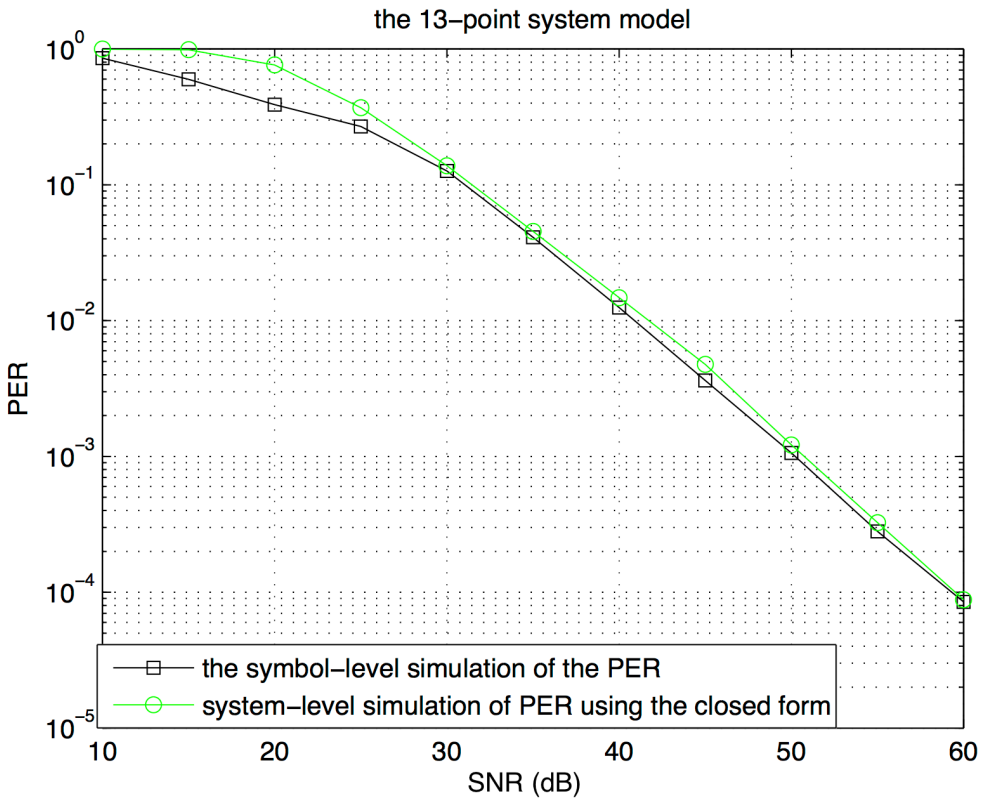

FIGURE 4.4: PER for the multi-node network

4.5 Separate PER simulation

We have up to now assume that for all PNC MAC phase, the SNR on each users are the same, and both channels have the same average amplitude. However, this may not be easily realized in industry. Instead, we may have different channel to noise (at the relay) ratio. We simulate its PER in Fig 4.5 .

From Fig 4.5, we can see that when the amplitude of one channel is far smaller than the other's, the PER maintains high regardless of the CNR(channel-to-noise ratio) of the better channel. Thus, we may even ignore the data sent via the worse channel, and make the TWRC into a point-to-point model. As a result, we may have two additional mappings as:

$$M_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$M_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

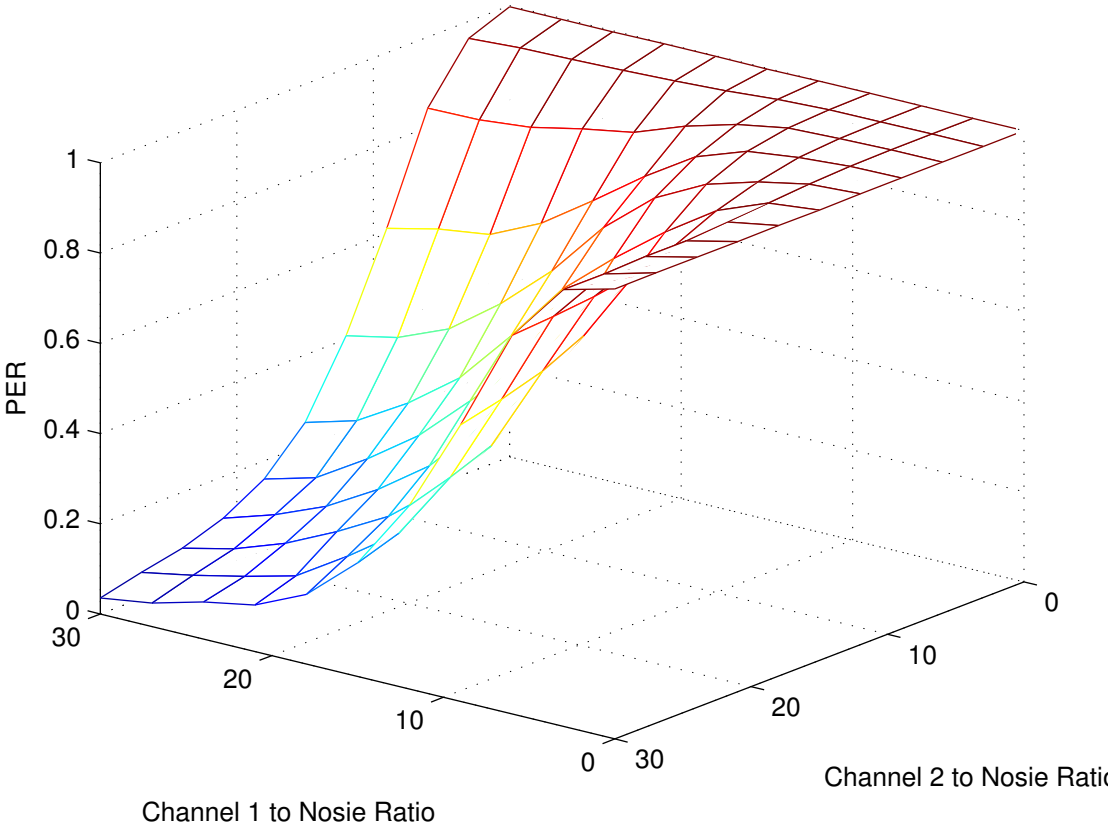

FIGURE 4.5: PER for the uncoded PNC with different channel to noise ratio in dB

These are also shown in the DIWINE project paper [3], and are simulated by the author.

4.6 Summary

We build two general multi-node transmission models in order to practice the system-level simulation, and we also design the algorithm for all relays to choose a realizable best mapping. Thus , the system-level simulation of the overall PER can be derived from the SER at any relay node in the system. This will enable the use of system-level simulation in most general multi-hop system.

Chapter 5

Convolutional Coded PNC

There are a lot of previous research about the coded PNC, one of them is [65], in which it decodes the individual information sent from both transmitters as joint channel-decoding network coding (Jt-CNC). This model is an evolutionary product based on many researches including [66] [67] [68] [69] [70] [71]. As we can see, it claims to have these advantages:

- 1. Jt-CNC decoder is optimal in terms of BER performance.
- 2. Jt-CNC aloow the users to have different channel codings.
- 3. Jt-CNC allow asynchronous linear coded PNC.

Based on the above advantages and the achievement in [72], further research of asynchronous linear coded PNC like[73] have sprung out. However, Jt-CNC will make the trellis very large if the code length reaches a certain amount, as the trellis expands exponentially with the increase of the code length.

Thus, we aim to build the joint decoding instead, and the PNC system is again synchronized [74] like the model of the uncoded PNC. Also, we assmue that both

users apply the same channel coding. In this situation, joint decoding will be much less complicated in the encoding and decoding than Jt-CNC without a significant loss of BER performance.

5.1 System model

The convolutionally coded PNC MAC phase is similar to the uncoded PNC in which user A and B transmit data to the relay simultaneously (we need to assume it is perfectly synchronized) with the same frame/packet length at time slot 1. Thus, the relay receives the symbol-by-symbol superimposed symbols instead of two separate data streams. After de-mapping and decoding, the symbols are decoded into a linear combined superimposed binary sequence (I_R) based on mapping. We may compare it with a linear combination of the sequences of the original binary data from both users with the same mapping mapping ($C(I_A, I_B)$) to obtain the error rate. We are typically interested in the MAC phase only, because it deals with the superimposed symbol and is thus more complicated. Like the uncoded PNC, we have 3 mappings available: the XOR, the reversed XOR and the anti- $\frac{\pi}{4}$ mapping. Likewise, the received symbol at the relay is again:

$$R_S = h_A S_A + h_B S_B + S_N \tag{5.1}$$

where R_S is the received signal at the relay, S_A and S_B are the modulated signals sent from each user, and S_N is the noise at the relay, h_A and h_B representing the channels' frequency response. Also, we assume that both channels are flat fading channels (the frequency response is a constant complex value during one packet time), because we will further extend our research onto the OFDM channel, whose narrow-band sub-channels can be treated as flat fading channels.

As long as we apply a convolutional code to the PNC, we need to assume that both transmitters apply the same convolutional code before the baseband information

being fed into the encoders. This is because we do joint decoding which decodes the received signal into a linear superimposed binary sequences rather than the separate information form the both users. Figure 5.1 shows the structure of a convolutional coded PNC model in the MAC phase.

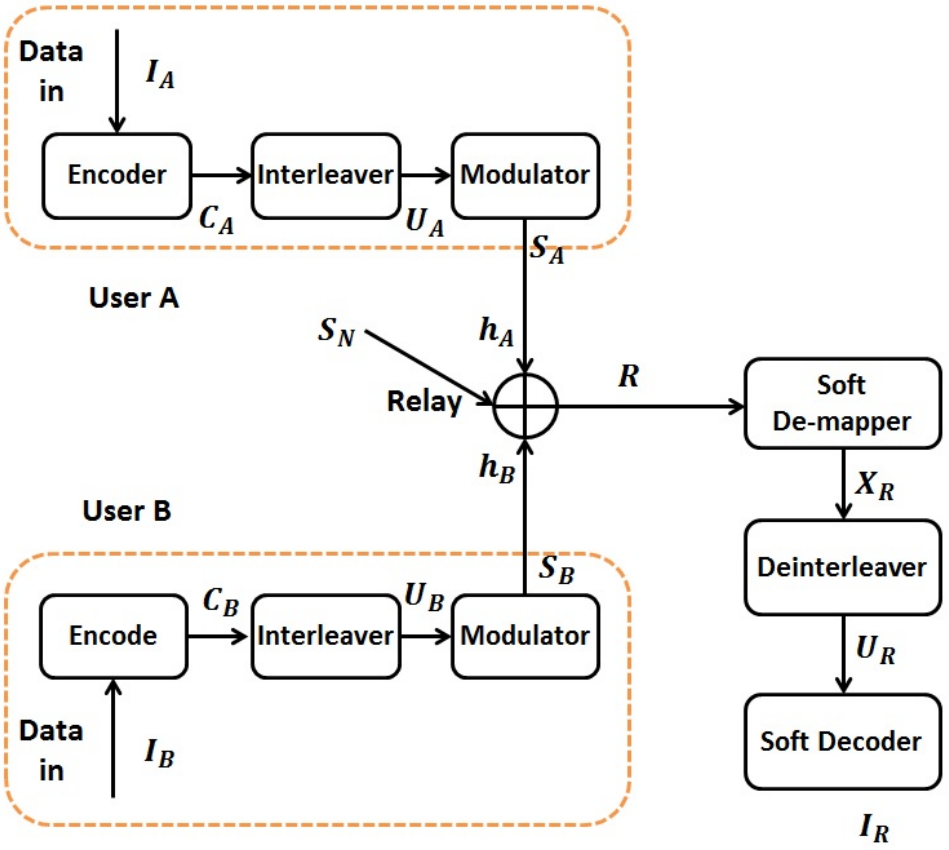

FIGURE 5.1: Structure of the convolutional coded PNC MAC phase

In figure 5.1 we have a simple block interleaver locate between encoder and modulator at each user in order to improve the overall error rate.

In the de-mapping stage, every received symbol in the signal sequence is compared with every element in a look-up table. This look-up table is formed by all constellations according to the channel fading state, and the 'Network Coding' term refer to the related network coding value of each constellation point in the look-up table. Take the XOR mapping as example: its 1^{st} bit correspond to the real part, and the 2^{nd} bit to the imaginary part of the network coding of the noiseless superimposed symbol if we assume the channel gains are 1. These settings are the same as for un-coded PNC.

However, in the the uncoded system, we de-map the received symbols by finding the term inside the look-up table that has the smallest Euclidean distance against the received symbol (which is similar to a hard-decision decoding). Now in the coded PNC, we calculate the accumulated Euclidean distances in the exponential form, and take the LLR value accordingly (corresponding to a soft-decision decoding) instead. To do this, we first refer to table 3.2 which shows the relationship between the constellation (R) and de-mapped vector (X_R). Then, we link the constellation to the superimposed network coded value (I_R) through table 3.4, 3.5 or 3.8 depend on mapping. The 1^{st} and the 2^{nd} bit of the network coding are calculated separately. Take the 1^{st} bit as an example:

$$U_R^{'}(1) = log\frac{\sum_{n:N_C(R(n))=1X} exp(-(M_n))}{\sum_{n':N_C(R(n'))=0X} exp(-(M_{n'}))} \tag{5.2}$$

where N_C function returns the network coding value of the target, X denotes a 'don't care' state (either 0 or 1). $U_R^{'}(1)$ is the maximum likelihood (ML) ratio of the 1^{st} bit. If we define the matrix of the lookup table as $R(n)$, then function M as the ratio of squared Euclidean distance over noise standard deviation can be written:

$$M_n = \frac{|R_s - R(n)|^2}{2\sigma^2} \tag{5.3}$$

where σ is the noise standard deviation.

Next, after the de-interleaving stage, the LLR is fed into a conventional Viterbi decoder and decoded to the binary information. The decoded information is the superimposed network-coded information rather than the individual information from both transmitters.

However, traditional Viterbi decoding cannot decode the LLR easily even the both transmitters are encoded with the same convolutional encoder. As [75] shows, the

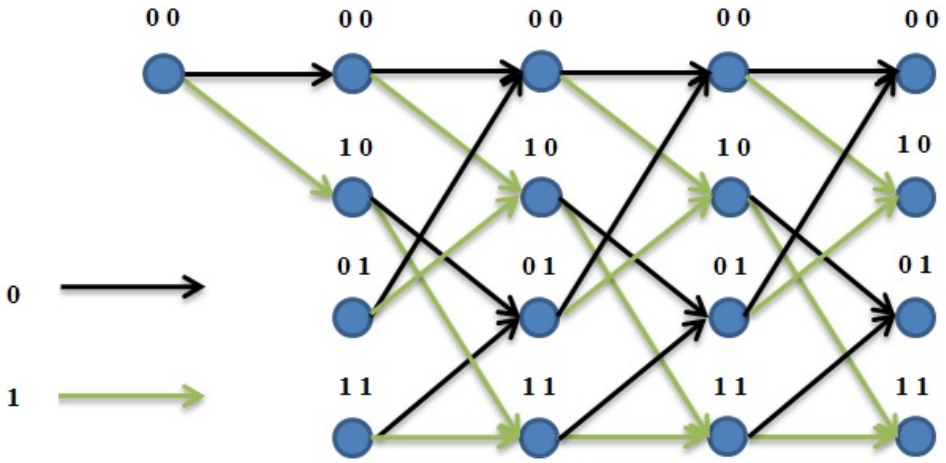

FIGURE 5.2: Combined trellis of a (5,7) convolutional code

LLR is fed into a "full-state" Viterbi algorithm decoder which is combined with the trellis from both users. Thus, finding the superimposed symbol at the relay will require exhaustive search for the most-likely path from the both trellis. Thus, this search actually decodes the separate information held by both users rather than the superimposed symbol directly. To improve that, [75] compresses the full-state trellis to a reduced state trellis suggested in Fig 5.2 (arrows represent the superimposed codeword it will be decoded as, and numbers represent the combined (bit-wise XOR) states), and shows that this compression does not lose much generality or accuracy in terms of the superimposed BER at the relay.

As [76] defines, the HDF has a layered strauctures, in which the inner layer provides the XOR property, while outer layer provides the error correction capability. Similarly in our research, the outer layer is the linear convolutional code responsible for error correction, while the inner layer closer to the channel symbols perform a linear operation depend on the mapping chosen. From [27] we know the linear combination of 2 linear codeword is another codeword, thus the de-noised superimposed codeword at the relay must be a superimposed codeword from both users, upon which our derivation of the BER is based.

Next we consider the constellation diagram at the relay shown in Fig 5.3 and

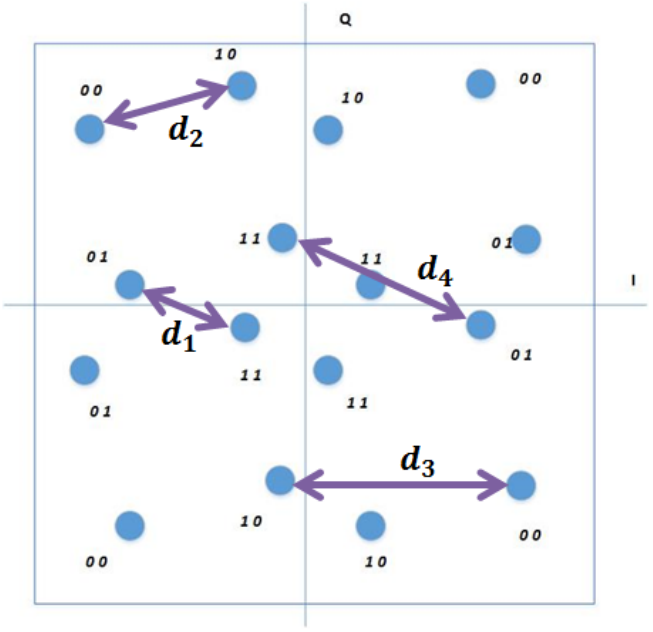

FIGURE 5.3: distance d_1, d_2, d_3 and d_4

Fig 5.4. The modulation applied with the PNC is QPSK modulation, thus the transmitted symbols and the channels are all complex. I If we apply the XOR mapping, for simplicity, we consider here the 1st bit of the network coding value corresponding to the constellations: we assume that the symbols in which the correct and the erroneous sequences differ in only the real part without a significant loss of generality, since a QPSK is equivalent to 2 orthogonal BPSKs. However, this does not apply to the anti$\frac{\pi}{4}$ mapping which we need to consider both real and imaginary part of the constellation. For convenience, we define 6 error distances according to the plots in Fig 5.3 and Fig 5.4:

- $d_1 = \sqrt{2}|jh_A + h_B|$;
- $d_2 = \sqrt{2}|h_B|$;
- $d_3 = \sqrt{2}|h_A|$;

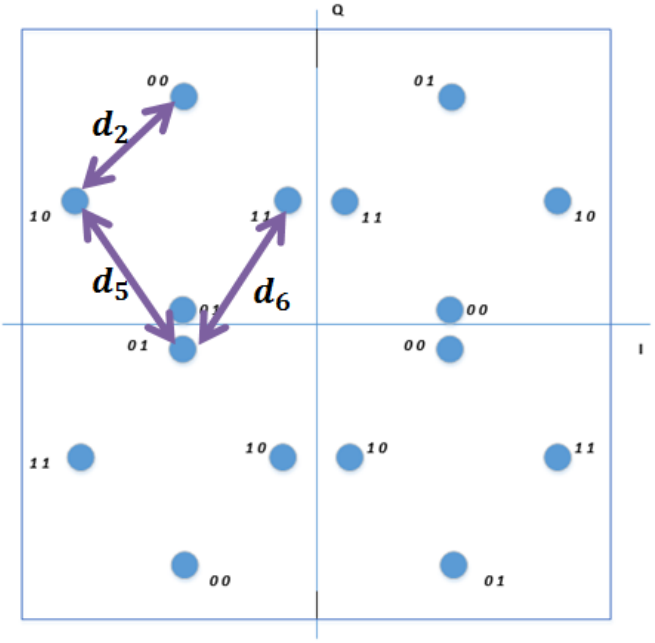

FIGURE 5.4: d_5 and d_6

- $d_4 = \sqrt{2}|h_A(1-j) + jh_B|$;
- $d_5 = \sqrt{2}|jh_A - h_B|$;
- $d_6 = \sqrt{2}|jh_A - jh_B| = \sqrt{2}|h_A - h_B|$;

5.2 BER analysis

In this section we calculate the superimposed BER at the relay with ML decoding by means of the error-event probability. We calculate it mainly through the binomial expansion and the total Euclidean distance.

5.2.1 Equal channel

We start our research from the equal channel case, in which the two channels frequency response are equal and are both normalised to 1 .

5.2.1.1 Pairwise Error Probability

For convenience, we define the expected correct superimposed sequence as s, which is equal to the linear combinition (with regard to mapping) of the binary information that the two users hold, denoted as $s = S_A h_A + S_B h_B$, where h is the channel gain. Accordingly, in the de-mapping stage, the erroneous superimposed sequences at the relay after de-noising is defined as s'.

In addition to these, we further introduce the correct codeword sequences as $c = C_A \oplus C_B$ (binary codeword sequence in between the encoding and the modulation stage). Similarly, we define the erroneous codeword at the relay as c'. Again, c' doesn't physically exist in the system, but is a virtual by-product in the decoding stage.

Next, we define some important terms below:

- Event $E(c, c')$: Event that a data sequence that should be decoded as a network coded sequence c is instead decoded as an erroneous sequence c;

- Pairwise error-event probability $P_{ep}(c, c')$: the probability of $E(c, c')$, written by:

$$P_{ep}(c, c') = P(E(c, c')) \tag{5.4}$$

- Union of the event $E(c, c')$ defined as $\bigcup_{c,c'} E(c, c')$.

Since there are many $s \equiv c$ and $s' \equiv c'$($\equiv$ symbol means 'decoded as' here), thus the pairwise error probability becomes:

$$P_{ep}(c, c') = \sum_{s \equiv c} P(s|c) \, P\left(\bigcup_{s' \equiv c'} E(s, s') \right) \tag{5.5}$$

where $P(s|c)$ denotes the probability of the superimposed signal sequences s given c. Since we use linear QPSK modulation, we can assume that all s give rise to c are equally probable, thus:

$$P_{ep}(s, s') = P(E(s, s')) = P(||R_s - s||^2 > ||R_s - s'||^2) = Q\left(\frac{d(s, s')}{2\sigma}\right) \quad (5.6)$$

here $d(s, s') = \sqrt{||s - s'||^2}$ denotes the Euclidean distance between the superimposed signal sequence s and the erroneous sequence s given that $||x||^2 = \sum_i {x_i}^2$ and σ is the noise standard deviation per dimension. According to the union bound:

$$P\left(\bigcup_{s' \equiv c'} E(s, s')\right) \leq \sum_{s' \equiv c'} P(E(s, s')) = \sum_{s' \equiv c'} P_{ep}(s, s') \quad (5.7)$$

Hence the pairwise error probability for the binary coding sequence c and c is:

$$P_{ep}(\mathbf{c}, \mathbf{c}') \leq \sum_{\mathbf{s} \equiv \mathbf{c}} P(\mathbf{s}|\mathbf{c}) \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \quad (5.8)$$

Now we can look at a particular example. Normally we could assume that the correct sequence is $c = 000000$ for a (5,7) convolutional code (becasue the convolutional code is linear, which means every codeword appear at the same probability. Thus, we can make an all-0 sequence as the correct sequence for simplicity, as every '1' in the erroneous sequence marks an error) and the erroneous sequence is $c' = 111011$ (the closest sequence in Hamming distance). Also, we assume that two users apply QPSK modulation, the two channels are both equal to 1, and we can just look at the 1_{st} bit of the network coding (in the XOR mapping, it corresponds to the real part) without loss of generality. Thus, the superimposed sequence s takes the values $-\sqrt{2}$,0,$\sqrt{2}$. Then for $c = 000000$ there are 2^6 sequences of s taking

the value of $\{\pm\sqrt{2}, \pm\sqrt{2}, \pm\sqrt{2}, \pm\sqrt{2}, \pm\sqrt{2}, \pm\sqrt{2}\}$, and they are all equally probable with $P(s|c) = \frac{1}{64}$. For $c' = 110111$, there are 2 erroneous sequences s' taking the values $\{0, 0, \pm\sqrt{2}, 0, 0, 0\}$. Thus, we can calculate that there are two values of total Euclidean distances $d(s, s')$ between them: the smaller one takes the values $\sqrt{10}$ (when the 3rd bits of s and s' have same polarities), and the other one takes the value $\sqrt{2 \times 5 + \left(2\sqrt{2}\right)^2} = 3\sqrt{2}$(when they are different), which makes $P_{ep}(c, c')$ as:

$$\begin{aligned} P_{ep}(000000, 110111) &= 64 \times \frac{1}{64}\left(Q\left(\frac{\sqrt{10}}{2\sigma}\right) + Q\left(\frac{3\sqrt{2}}{2\sigma}\right)\right) \\ &= Q\left(\frac{\sqrt{10}}{2\sigma}\right) + Q\left(\frac{3\sqrt{2}}{2\sigma}\right) \end{aligned} \tag{5.9}$$

At high SNR, the term $Q\left(\frac{3\sqrt{2}}{2\sigma}\right)$ which corresponds to a Euclidean distance of $s\sqrt{2}$ will be asymptotically negligible compared to the term $Q\left(\frac{\sqrt{10}}{2\sigma}\right)$ with a Euclidean distance of $\sqrt{10}$.

Now we move on to the general case for the equal channel model. Assume the total length of the sequence s and s' are both n bits long, and there are l 1s in c (correspond to l 0s in s). c'/s', however, differs from c/s in d_H places as the following form shows:

$$\begin{aligned} \mathbf{s} &= \left\{\underbrace{0, 0 \ldots 0, 0, 0, 0 \ldots 0, 0, 0}_{l}, \underbrace{\pm\sqrt{2}, \pm\sqrt{2} \ldots \pm\sqrt{2}, \ldots \pm\sqrt{2}}_{n-l}\right\} \\ \mathbf{s}' &= \left\{\underbrace{0, 0 \ldots 0}_{l-k}, \underbrace{\pm\sqrt{2} \ldots \pm\sqrt{2}}_{k}, \underbrace{0, 0, 0 \ldots 0, 0, 0}_{d_H-k}, \underbrace{\pm\sqrt{2}, \ldots \pm\sqrt{2}}_{n-l-d_H+k}\right\} \end{aligned}$$

Observing from the above form that there are 2^{n-l} sequences s (which are again equiprobable) and 2^{n-l-d_H+2k} sequences s'. In 2^k of the letter (those in which both s and s' are 0 in $l-k$ places, and both s and s' take the value $\pm\sqrt{2}$ with the same polarity in $n-l-d_H$), the Euclidean distance is $\sqrt{2d_H}$, which is minimum. Once

again we ignore the term with higher Euclidean distance. Then the pairwise error probability is approximately:

$$P_{ep}(c, c') \simeq 2^k Q\left(\frac{\sqrt{d_H}}{\sqrt{2}\sigma}\right) \tag{5.10}$$

Still we cannot treat this as a proper upper bound because we have neglected some terms (terms corresponding to the Euclidean distance of $3\sqrt{2}$ in the previous example). So consider the situation where s and s' takes opposite polarities in p bits out of the $n - l - d_H + k$ places. Then the total Euclidean distance becomes $d(s, s') = \sqrt{2d_H + 8p}$, and there are $2^k \ {}^{(n-l-d_H+k)}C_p$ such sequences, making it:

$$P_{ep}(\mathbf{c}, \mathbf{c}') \simeq 2^k \sum_{p=0}^{n-l-d_H+k} {}^{(n-l-d_H+k)}C_p Q\left(\frac{\sqrt{d_H + 4p}}{\sqrt{2}\sigma}\right) \tag{5.11}$$

Once again, the term inside the Q fucntion in the expansion has an increment of Hamming distance 4, which means it is negligible for higher SNR values. Since the distribution of k is binomial (k is the number of places containing 1s in d_H places). Thus the average pairwise error probability over all pairs of c, c' which differs in d_H places is:

$$\overline{P_{ep}}(d_H) \simeq \sum_{k=0}^{d_H} \frac{{}^{d_H}C_k}{2^{d_H}} 2^k Q\left(\frac{\sqrt{d_H}}{\sigma}\right) = \left(\frac{3}{2}\right)^{d_H} Q\left(\frac{\sqrt{d_H}}{\sqrt{2}\sigma}\right) \tag{5.12}$$

5.2.1.2 Event Error Probability and BER

For a linear convolutional channel code, an error event is the event that any erroneous sequence is chosen which diverts from the correct path from a given point in the sequence (the start of the sequence) and returns to it at some later point, denoted as $E(c) = \bigcup_{c'} E(c, c')$. We assume that erroneous sequence c' differs from c from the first symbol and later converges with c. Thus, according

to the union bound, the error event probability P_{ev} is:

$$P_{ev}(\mathbf{c}) = P(E(\mathbf{c})) \leq \sum_{\mathbf{c}'} P_{ep}(\mathbf{c}, \mathbf{c}') \tag{5.13}$$

Substituting from (5.8) we have:

$$P_{ev}(\mathbf{c}) \leq \sum_{\mathbf{c}'} \sum_{\mathbf{s} \equiv \mathbf{c}} P(\mathbf{s}|\mathbf{c}) \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) = \sum_{\mathbf{s} \equiv \mathbf{c}} P(\mathbf{s}|\mathbf{c}) \sum_{\mathbf{c}'} \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \tag{5.14}$$

We take into account the number of network coded bit errors that arise from a given erroneous coded sequence c', denoted as $e_b(c')$. Then the BER of the superimposed symbol at the relay is:

$$P_{eb}(\mathbf{c}) \leq \sum_{\mathbf{s} \equiv \mathbf{c}} P(\mathbf{s}|\mathbf{c}) \sum_{\mathbf{c}'} e_b(\mathbf{c}') \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \tag{5.15}$$

Then the average BER is given by taking the average over all possible coded sequences:

$$\overline{P_{eb}} \leq \sum_{\mathbf{c}} P(\mathbf{c}) \sum_{\mathbf{s} \equiv \mathbf{c}} P(\mathbf{s}|\mathbf{c}) \sum_{\mathbf{c}'} e_b(\mathbf{c}') \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \tag{5.16}$$

Again we look back to the example in which the correct network coded sequence is an all-0 sequence (a network coded 0 corresponds to a $\pm\sqrt{2}$) with length n:

$$s = \left\{ \underbrace{\pm\sqrt{2}, \pm\sqrt{2} \ldots \pm\sqrt{2}, \ldots \pm\sqrt{2}}_{n} \right\}$$
$$s' = \left\{ \underbrace{\pm\sqrt{2} \ldots \pm\sqrt{2}}_{n-d_H}, \underbrace{0, 0, 0 \ldots 0, 0, 0}_{d_H} \right\}$$

There are 2^n such sequences s, and they are assumed to be equiprobable. And again, we look only at the minimum Euclidean distance between s and s', which is $\sqrt{2d_H}$. This is because the next-to-minimum distance has an increment of $2\sqrt{2}$, and hence can be ignored. Thus P_{ev} is:

$$\begin{aligned} P_{ev}(\mathbf{0}) &\leq 2^n \frac{1}{2^n} \sum_{\mathbf{c}'} \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) = \sum_{\mathbf{c}'} \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \\ &= \sum_{d_H \geq d_{min}} \sum_{\mathbf{c}' : d(\mathbf{c}, \mathbf{c}') = d_H} \sum_{\mathbf{s}' \equiv \mathbf{c}'} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) = \sum_{d_H \geq d_{min}} A(d_H) \sum_{\mathbf{s}' \equiv \mathbf{c}' : d(\mathbf{c}, \mathbf{c}') = d_H} Q\left(\frac{d(\mathbf{s}, \mathbf{s}')}{2\sigma}\right) \\ &\simeq \sum_{d_H \geq d_{min}} A(d_H) Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \end{aligned} \tag{5.17}$$

On the 2^{nd} and the 3^{rd} line we group the erroneous sequences according to their Hamming distance from c (Hamming distances between their related c and c') denoted as $s' \equiv c' : d(c, c') = d_H$, because the pairwise error probability (the inner summation) depends primarily on the Hamming distances. We denote the number of sequences c' which differ from c at a Hamming distance of d_H as $A(d_H)$. Finally we notice that for each c' there is only 1 s' that meets the requirement, and the minimum Euclidean distance is $\sqrt{2d_H}$. Taking into account the number of bit errors that arise from c', the BER is:

$$P_{eb}(\mathbf{0}) \simeq \sum_{d_H \geq d_{min}} e(d_H) Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \tag{5.18}$$

where $e(d_H)$ is the error-weighted distance spectrum defined as the number of sequences at Hamming distance d_H weighted according to the average number of bit errors that occur on the paths. Also, it is the average value of $e(c')$ over all sequences of c' at a Hamming distance d_H from 0. Now let us return to a general correct sequence c. We can still write:

$$\begin{aligned} P_{ev}(c) &\leq \sum_{d_H \geq d_{min}} \sum_{c':d(c,c')=d_H} \sum_{s' \equiv c'} Q\left(\frac{d(s,s')}{2\sigma}\right) \\ &\simeq \sum_{d_H \geq d_{min}} \sum_{c':d(c,c')=d_H} \sum_{s' \equiv c':d(s \equiv c,s')=\sqrt{2d_H}} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \\ &= \sum_{d_H \geq d_{min}} \sum_{c':d(c,c')=d_H} 2^k_{c,c'} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \end{aligned} \tag{5.19}$$

Since the value of the pairwise error probability (which is the inner summation) depends on d_H only, also we know from the forms that the number of erroneous signal sequences s' is given by 2^k (k is the number of places in which the network coded sequence c corresponding to s contains 1 and the $c' \equiv s'$ contains 0, denoted as $k_{(c,c')}$. Thus, the average error event probability over all correct sequences is:

$$\begin{aligned} \overline{P_{ev}} &\simeq \sum_c P(c) \sum_{d_H \geq d_{min}} \sum_{c':d(c,c')=d_H} 2^{k_{c,c'}} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \\ &= \sum_{d_H \geq d_{min}} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \sum_c P(c) \sum_{c':d(c,c')=d_H} 2^{k_{c,c'}} \\ &= \sum_{d_H \geq d_{min}} A(d_H) \left(\frac{3}{2}\right)^{d_H} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) \end{aligned} \tag{5.20}$$

In the last chapter, we derive the overall PER from SER at the relay. In the convolutional coded PNC, however, we cannot do it this way as the outcome of the convolutional decoder is bit stream rather than symbols. And this is why we would finally derive the BER instead.

However, there is another error rate called 'word error probability' (WER) which is the error rate inside a single codeword, given by:

$$P_W = 1 - (1 - P_{ev})^N \tag{5.21}$$

where P_W is the WER, and N is the packet length If we are doing the system-level simulation on the convolutional coded PNC, we coudl still obtain the estimated overall PER as:

$$P_P = 1 - (1 - P_{W-A})^N \tag{5.22}$$

$$\approx P_{W-A} \times N \tag{5.23}$$

where P_{W-A} is the averaged WER at the relay. However, we still foucs on the derivation of BER as this is the direct outcome that can be measured. If we assume all correct sequences c are equiprobable as well as using the result of (5.12) that $\frac{{}^{d_H}C_k}{2^{d_H}} 2^k = \left(\frac{3}{2}\right)^{d_H}$. Thus, taking into account the number of bit errors for each erroneous sequence we have the average BER as:

$$\overline{P_{eb}} \simeq \sum_{d_H \geq d_{min}} e\left(d_H\right) \left(\frac{3}{2}\right)^{d_H} Q\left(\frac{\sqrt{2d_H}}{2\sigma}\right) = \sum_{d_H \geq d_{min}} e\left(d_H\right) \overline{P_{ep}\left(d_H\right)} \tag{5.24}$$

As a result, the theoretical average BER of the superimposed symbol at the relay depends on $\overline{P_{ep}}\left(d_H\right)$ only.

5.2.2 The real channel

We introduce the fixed real fading channels in order to calculate the number of erroneous sequences in terms of the binomial expansion in this section. First

we look at the comparison of the expected noiseless superimposed sequences and the erroneous superimposed sequences that being detected as s'. Here we define $h_A = 1 + \delta$ and $h_B = 1 - \delta$, which δ is real, then:

$$\mathbf{s} = \left\{ \underbrace{\pm\sqrt{2}\delta \ldots \pm\sqrt{2}\delta \ldots\ldots \ldots \pm\sqrt{2}\delta \ldots \pm\sqrt{2}\delta}_{l}, \underbrace{\pm\sqrt{2} \ldots \pm\sqrt{2} \ldots \pm\sqrt{2} \ldots \pm\sqrt{2}}_{n-l} \right\}$$

$$\mathbf{s}' = \left\{ \underbrace{\pm\sqrt{2}\delta \ldots \pm\sqrt{2}\delta}_{l-k}, \underbrace{\mp\sqrt{2} \ldots \mp\sqrt{2}}_{p}, \underbrace{\pm\sqrt{2} \ldots \pm\sqrt{2}}_{k-p}, \underbrace{\pm\sqrt{2}\delta \ldots \pm\sqrt{2}\delta}_{d_H-k}, \underbrace{\pm\sqrt{2} \ldots \pm\sqrt{2}}_{n-l-d_H+k} \right\}$$

For the correct sequence s, there are l bits that take a value $\pm 2\delta$ (network coded 1), and thus $n - l$ bits of ± 2 (network coded 0). We define d_H as the number of places that S and S' differ from in terms of their related network coding value, and k is the number of bits among d_H places that s takes a network coded 1 while s' takes a network coded 0. Among k we further define variable p such that in this p places, the symbol of s and s' have different polarity.

Our main focus is on the d_H places, but we cannot ignore the remaining $n - d_H$ places when s and s' have different polarities. However, in d_H places the distance to cause an error is either $\sqrt{2}(1 - \delta)$ or $\sqrt{2}(1 + \delta)$, while the distance to cause an error in the remaining $n - d_H$ places is $2\sqrt{2}$, which is negligible compared with $\sqrt{2}(1 - \delta)$ or $\sqrt{2}(1 + \delta)$(especially in higher SNRs).

Now we calculate the total numbers of erroneous sequences The length of s is n, so there are 2^n possible sequences, and we assume that they are equally probable. Thus the possibility of choosing a particular sequence is $\frac{1}{2^n}$. For each sequence s, there are d_H places that differ from s' as we defined. Inside d_H places we have another variable k ranging form 0 to d_H, and p from 0 to k, showing that both k and p are all binomial. thus, the total erroneous sequences N at such d_H places is :

$$N = {}^{d_H}C_k \, {}^{k}C_p \tag{5.25}$$

Then we move on to the Euclidean distances $d(s, s')$ lying between s and s', which is:

$$\begin{aligned} d(s, s') &= \sqrt{2(d_H - p)(1-\delta)^2 + 2p(1+\delta)^2} \\ &= \sqrt{2d_H(1-\delta)^2 + 8p\delta} \end{aligned} \tag{5.26}$$

Note that both k and p are binomial, thus the average pairwise error probability over all possible network coding sequence c (correct) and c' (erroneous) which differ in d_H places is:

$$\begin{aligned} \overline{P_{ep}(d_H)} &= \sum_{k=0}^{d_H} \frac{N}{2^{d_H}} Q\left(\frac{d(s, s')/2}{\sigma}\right) \\ &= \sum_{k=0}^{d_H} \frac{{}^{d_H}C_k \, {}^{k}C_p}{2^{d_H}} Q\left(\frac{\sqrt{2d_H(1-\delta)^2 + 8p\delta}}{\sigma}\right) \end{aligned} \tag{5.27}$$

5.2.3 The complex channel

We now move on to the complex channels which is more close to the real situation. In this situation, we not only need to calculate the number of erroneous sequences by means of the binomial expansion, but also need to calculate the number of the correct sequence in the same way.

5.2.3.1 The XOR mapping

The XOR mapping is the most appropriate mapping when the relative phase shift σ is around 0 as shown in Fig. 5.5. For convenience, we could define the channels as $h_A = 1+\delta$ and $h_B = 1-\delta$, and δ is in general complex. We are most interested in the case where $|\delta|$ is small, because otherwise the constellation points will be divided into 4 cluster of 4 aggregated points each, which means the error distances will just be within the clusters only. Also, we could assume that δ lies in the first quadrant without a loss of generality. This is because we could still obtain the the equivalent result by taking its conjugate if δ lies in the second quadrant. Thus, the error distances are:

$$
\begin{aligned}
d_1 &= \sqrt{2}|1-(1-2j)\delta| \\
d_2 &= \sqrt{2}|h_B| = 2|1-\delta| \\
d_3 &= \sqrt{2}|h_A| = 2|1+\delta| \\
d_4 &= \sqrt{2}|1+(1+2j)\delta|
\end{aligned}
$$

Putting $\delta = \delta_0 \exp(j\theta_\delta)$, these error distances will further become:

$$
\begin{aligned}
d_1 &= \sqrt{2\left(1+5{\delta_0}^2-2\delta_0 cos(\theta_\delta)-4\delta_0 sin(\theta_\delta)\right)} \\
d_2 &= \sqrt{2\left(1+{\delta_0}^2-2\delta_0 cos(\theta_\delta)\right)} \\
d_3 &= \sqrt{2\left(1+{\delta_0}^2+2\delta_0 cos(\theta_\delta)\right)} \\
d_4 &= \sqrt{2\left(1+5{\delta_0}^2+2\delta_0 cos(\theta_\delta)-4\delta_0 sin(\theta_\delta)\right)}
\end{aligned}
$$

Note if $\theta_\delta \leq \arctan\left(\frac{\delta_0}{\sqrt{1-{\delta_0}^2}}\right)$, the minimum distance is the minimum of d_1 and d_2. this is similar for the case that d_3 should be used instead of d_4

According to Fig. 5.5, we can divide all points into 5 sets from a to e, with points in a same sets having a same total error distances :

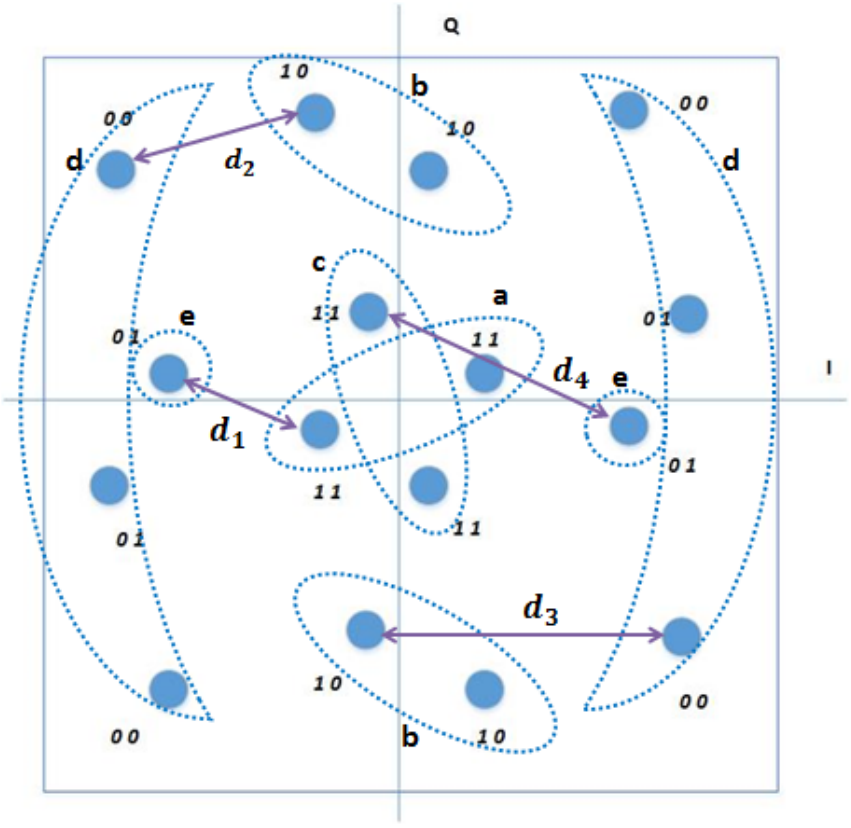

FIGURE 5.5: Constellation diagram

- a: 1 d_1 and 1d_3 (a d_2 is covered by a d_1 in this situation, because d_1 is smaller in this case);
- b: 1 d_2 and 1d_3;
- c: 1 d_2 and 1d_3;
- d: 1 d_2;
- e: 1 d_1.

Among them, constellation points in sets d and e have only 1 error distance, and the rest have 2. From the network coding view, there are 2 ways for a network coded 0 to become a network coded 1, but only 1 way for a network coded 1 to become a network coded 0. This result in introducing extra erroneous sequences. Taking set a as example: a network coded 1 may become a '0' by being distorted at a distance of either d_2 or d_1. If there are altogether a_1 d_1s and a_2 d_2s, then its

easy to see that the distribution of both a_1 and a_2 are binomial. Unfortunately both the correct sequence and the erroneous sequences need a further division. Here we define the correct sequence as c, and the erroneous sequence as c'. Also, we assume that k bits out of d_H places that c takes a network coded 1 (c' takes a network coded 0 at the same position). These variables will help to sum up the total number of correct sequences c as:

For c taking a network coded 1, the distribution of k is:

- p bits in set a;
- q in set b;
- $r = k - p - q$ in set c.

Likewise, for c taking a network coded 0, the distribution of $d_H - k$ is:

- s in set e;
- $t = d_H - k - s$ in set d.

All the above variables are binomial distributed. Let $s \equiv c$ denote some noiseless superimposed signal sequences s at the receiver which is decoded as the network coded sequence c, then its probability $P(s|c)$ is:

$$P(s|c) = \frac{{}^{k}C_{p}\ {}^{k-p}C_{q}\ {}^{d_H-k}C_{s}}{2^{d_H}} \tag{5.28}$$

Now we look at the erroneous sequences $s' \equiv c'$, and further divide the variables.

- In set a, there are $\bar{p}$ out of p points locate at an error distance of d_1, and $p - \bar{p}$ at d_3s;
- $\bar{q}$ of q at d_2 and $q - \bar{q}$ at d_3 In set b;
- $\bar{r}$ of r at d_2 and $r - \bar{r}$ at d_4 in set c,
- all t at d_1 in set d,
- all s at d_2 in set e,

Again they are binomial distributed. Thus the number of such sequences is:

$$N(s' : d(s, s') = d_e) = {}^{p}C_{\bar{p}}\, {}^{q}C_{\bar{q}}\, {}^{k-p-q}C_{\bar{r}} \tag{5.29}$$

Where $d_e \in d(s, s')$ is the set of Euclidean distance values between s and s', and $N(s' : d(s, s') = d_e)$ represent the number of erroneous superimposed sequences s' at a distance d_e from s.

Next we calculate the total Euclidean distance d:

$$
\begin{aligned}
d(s,s')) = \Bigg(& \bar{p}{d_1}^2 + (p-\bar{p}){d_3}^2 + \bar{q}{d_2}^2 + (q-\bar{q}){d_3}^2 + \bar{r}{d_2}^2 \\
& + (r-\bar{r}){d_4}^2 + s{d_1}^2 + t{d_2}^2 \Bigg)^{\frac{1}{2}} \\
= & \Bigg((\bar{p}+s){d_1}^2 + (\bar{q}+\bar{r}+t){d_2}^2 \\
& + (p-\bar{p}+q-\bar{q}){d_3}^2 + (r-\bar{r}){d_4}^2 \Bigg)^{\frac{1}{2}} \\
\\
= & \sqrt{2} \Bigg(d_H + (d_H + 4(k-p+\bar{p}-q-\bar{r}+s)){\delta_0}^2 \\
& - 2(d_H - 2(k-\bar{p}-\bar{q}-\bar{r}))\delta_0 cos(\theta_\delta) \\
& - 4(k-p+\bar{p}-q-\bar{r}+s)\delta_0 \Bigg)^{\frac{1}{2}} \\
\\
= & \sqrt{2} \Bigg(d_H(1+{\delta_0}^2 - 2\delta_0 cos(\theta_\delta)) + 4k({\delta_0}^2 + \delta_0 cos(\theta_\delta)) \\
& - 4(p+q)({\delta_0}^2 - \delta_0 sin(\theta_\delta)) \\
& + 4\bar{p}({\delta_0}^2 - \delta_0 cos(\theta_\delta) - \delta_0 sin(\theta_\delta)) - 4\bar{q}\delta_0 cos(\theta_\delta) \\
& - 4(\bar{r}({\delta_0}^2 + \delta_0 cos(\theta_\delta) - \delta_0 sin(\theta_\delta)) + 4s({\delta_0}^2 - \delta_0 sin(\theta_\delta)) \Bigg)^{\frac{1}{2}}
\end{aligned}
\tag{5.30}
$$

Next, we calculate the pairwise error probability P_{ep}. We define E as the event, and $\bigcup_x E_x$ as the union of event E_x, so:

$$
\begin{aligned}
P_{ep}(c,c') &= \sum_{s\equiv c} P(s|c) P\Big(\bigcup_{s'\equiv c'} E(s,s') \Big) \\
&\leq \sum_{s\equiv c} P(s|c) \sum_{s'\equiv c'} P(E(s,s')) \\
&= \sum_{s\equiv c} P(s|c) \sum_{s'\equiv c'} Q\Big(\frac{d(s,s')}{2\sigma}\Big) \\
&= \mathop{\mathbb{E}}_{s\equiv c} \Big[\sum_{s'\equiv c'} P(s|c) Q\Big(\frac{d(s,s')}{2\sigma}\Big) \Big] \\
&= \mathop{\mathbb{E}}_{s\equiv c} \Big[\sum_{d_e \in d(s,s')} P(s|c) N(s' : d(s,s') = d_e) Q\Big(\frac{d_e}{2\sigma}\Big) \Big] \\
&= \sum_{d_e \in d(s,s')} \mathop{\mathbb{E}}_{s\equiv c} \Big[P(s|c) N(s' : d(s,s') = d_e) Q\Big(\frac{d_e}{2\sigma}\Big) \Big] \qquad (5.31)
\end{aligned}
$$

Where $s \equiv c$ denotes that the correct code sequence c is decoded as the correct superimposed sequence s, $\mathbb{E}$ is the expectation, $d_e \in d(s,s')$ denote that d_e is in the set of Euclidean distance values between s and s'. $P(s|c)$, $N(s' : d(s,s') = d_e)$, $Q(\frac{d_e}{2\sigma})$ have been calculated in equation 5.28, 5.29 and 5.30. Next we calculate the average pairwise error probability over all c,c' which differ in d_H places as:

$$
\overline{P_{ep}(d_H)} \leq \sum_{k=0}^{d_H} \frac{{}^{d_H}C_k}{2^{d_H}} P_{ep}(c,c') \qquad (5.32)
$$

Where $d_H \in (c,c')$ is the hamming distances between the selected correct sequence and the erroneous sequences.

Finally we form the average superimposed BER at the relay $\overline{P_{eb}}$ by the pairwise error probability and the error-weighted distance spectrum $e(d_H)$:

$$
\overline{P_{eb}} \leq \sum_{d_H \geq d_{min}} e(d_H) \overline{P_{ep}(d_H)} \qquad (5.33)
$$

5.2.3.2 The reversed XOR mapping

Now we move on to the reversed XOR mapping when the relative phase shift σ is around $\frac{\pi}{2}$, which is shown in Fig.5.6 below. As we can see, the 1_{st} bit of the network coding for this revered XOR mapping is the revered value (change form 1 to 0 or the otherwise) of the that for the XOR mapping, while the 2_{nd} bit maintain the same. This change does not affect the calculation of the BER due to symmetry. Thus, the average BER for the reversed XOR mapping under a $\frac{\pi}{2}$ channel fading state.

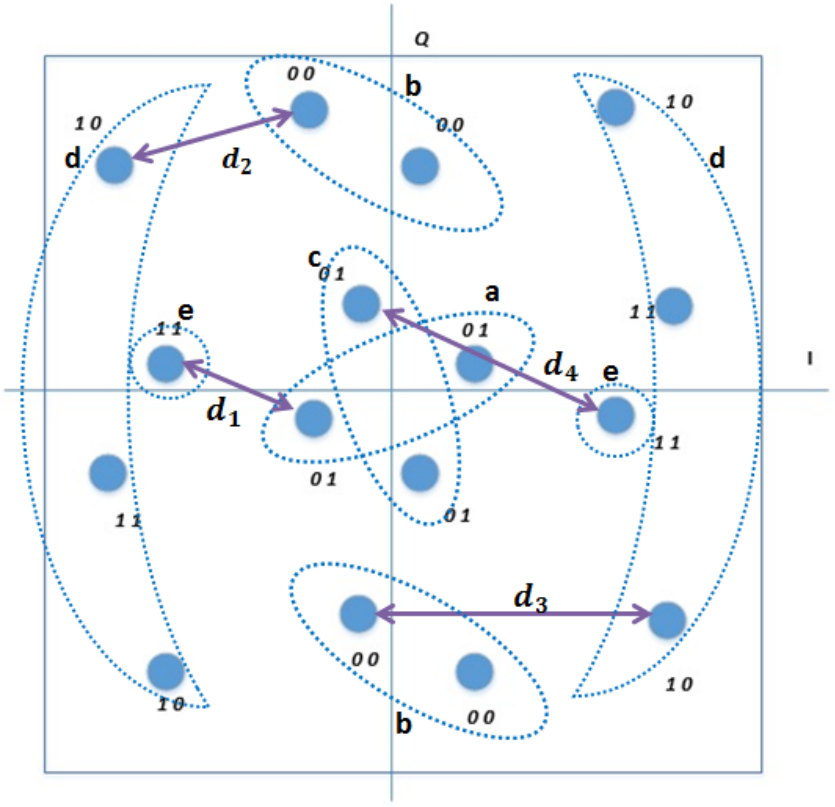

FIGURE 5.6: $\frac{\pi}{2}$ case with reversed XOR mapping

Again for convenience, we could define the channels as $h_A = j + \delta$ and $h_B = j - \delta$, where δ is in general complex and its amplitude is small. Thus we can obtain the error distances as:

$$d_1 = \sqrt{2}|(\delta + 1)(j - 1)|$$
$$d_2 = \sqrt{2}|jh_B| = 2|1 + j\delta|$$
$$d_3 = \sqrt{2}|h_A| = 2|j + \delta|$$
$$d_4 = \sqrt{2}|1 + (1 + 2j)\delta|$$

5.2.3.3 The anti-$\frac{\pi}{4}$ mapping

Again for convenience, we could define the channels as $h_A = \frac{1+j}{\sqrt{2}} + \delta$ and $h_B = \frac{1+j}{\sqrt{2}} - \delta$ (σ), where δ is in general complex and its amplitude is small. Thus we can obtain the error distances as:

$$d_1 = \sqrt{2}|\frac{1 + j}{\sqrt{2}} - \delta|$$
$$d_5 = \sqrt{2}|j(\delta + 1) - \sqrt{2}|$$
$$d_6 = 2\sqrt{2}|\delta|$$

The anti-$\frac{\pi}{4}$ mapping is different from either the XOR mapping or the reversed XOR mapping as it is not a linear mapping. Thus, it doesn't maintain the same symmetry as the other 2 mappings. In this case, we have to analyse the real and the imaginary part (or the 1_{st} and the 2_{nd} network coding value) separately. To start with, we take a look at the 1_st bit of the network coding value. We can divide all points into 4 sets from a to d , with points in a same sets having a similar error distances with neighbours as can be seen for Fig 5.7. Like the XOR mapping, we assume that c and c' differs in d_H places, and there are k bits out of d_H that take a network coded 1. Thus we could assign:

- p bits in set a;
- $k - p$ in set c;

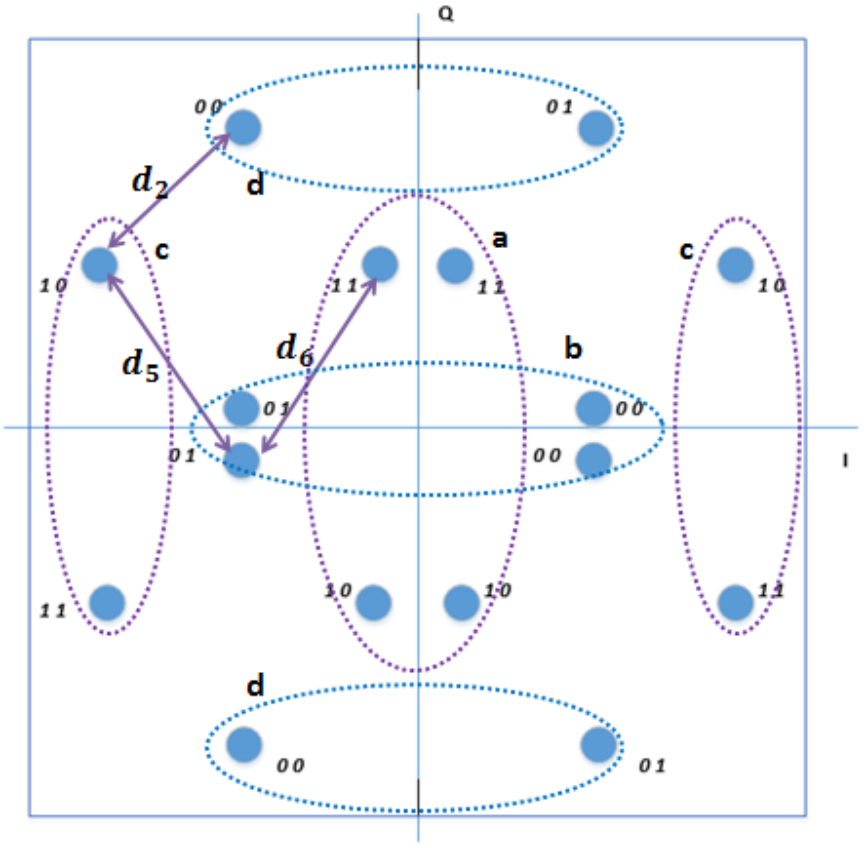

FIGURE 5.7: The 1_{st} bit

Likewise, for c taking a network coded 0, the distribution of $d_H - k$ is:

- q in set b;
- $d_H - k - q$ in set d.

All the above variables are binomial distributed. Thus for $s \equiv c$, we can derive its probability $P(s|c)$ as:

$$P(s|c) = \frac{{}^{k}C_p \, {}^{d_H-k}C_q}{2^{d_H}} \tag{5.34}$$

Now we look at the erroneous sequences $s' \equiv c'$, and further divide the variables.

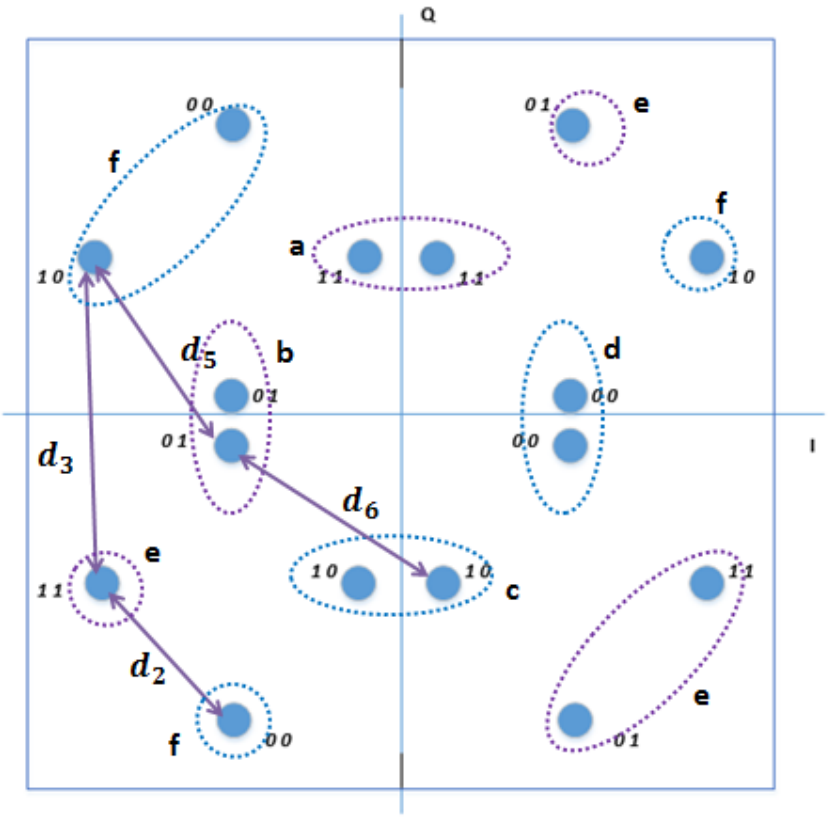

FIGURE 5.8: The 2_{nd} bit

- p_1 of p at d_5, p_2 of $p - p_1$ at d_6 and $p - p_1 - p_2$ at d_2 In set a;
- q_1 of q at d_5, q_2 of $q - q_2$ at d_6 and $q - q_1 - q_2$ at d_2 In set b;
- all $k - p$ at d_2 in set c,
- all $d_H - k - q$ at d_2 in set d,

Again they are binomial distributed. Thus the number of the erroneous sequences is:

$$N(s' : d(s, s') = d_e) = {}^{p}C_{p_1}\ {}^{p_1}C_{P_2}\ {}^{q}C_{q_1}\ {}^{q_1}C_{q_2} 2^{d_H - p_1 - p_2 - q_1 - q_2} \tag{5.35}$$

because all points have 2 d_2s. Next we calculate the total Euclidean distance d.

$$d(s,s') = \sqrt{(d_H - p_1 - p_2 - q_1 - q_2){d_2}^2 + (p_1 + q_1){d_5}^2 + (p_2 + q_2){d_6}^2} \tag{5.36}$$

Filling into (11), we can get the average pairwise error probability for the sequence formed by the 1_st bit of the network coding $\overline{P_{ep(1)}(d_H)}$, and also $\overline{P_{eb(1)}}$.

Next, we analyse the 2_{nd} bit of the constellation points. As Fig 5.8 shows, we divide them into 6 sets as the inner set a d and the outer set e and f. Again we consider the sequence $s \equiv c$ which in the d_H places where c differs from c'. Like the analysis of the 1_st bit of the network coding, there are k bits out of d_H that take a network coded 1. Thus we could assign:

- p in set a.
- q bits in set b;
- $r = k - p - q$ in set e;

Likewise, for c taking a network coded 0, the distribution of $d_H - k$ is:

- s in set c.
- t bits in set d;
- $u = d_H - k - s - t$ in set f;

All the above variables are binomial distributed. Thus for $s \equiv c$, we can derive its probability $P(s|c)$ as:

$$P(s|c) = \frac{{}^{k}C_p \, {}^{k-p}C_q \, {}^{d_H-k}C_s \, {}^{d_H-k-s}C_t}{2^{d_H}} \tag{5.37}$$

Now we consider the case $s' \equiv c'$ in which:

- p' of p at d_6 and $p - p'$ at d_2 In set a;
- q' of q at d_5 and $q - q'$ at d_2 In set b;
- s' of s at d_6 and $s - s'$ at d_2 In set c;
- t' of t at d_5 and $t - t'$ at d_2 In set d;
- r' of r at d_3 and $k - p - q - r'$ at d_2 In set e;
- u' of q at d_3 and $d_H - k - s - t - u'$ at d_2 In set f;

such a sequence has the Euclidean distance from s as:

$$\begin{aligned} d(s, s') = \Big(& (d_H - p' - q' - s' - t'){d_2}^2 + (r' + u'){d_3}^2 \\ & + (q' + t'){d_5}^2 + (p' + s'){d_6}^2 \Big)^{-\frac{1}{2}} \end{aligned} \tag{5.38}$$

And the number of such sequences is:

$$N(s' : d(s, s') = d_e) = {}^{p}C_{p'}\ {}^{q}C_{q'}\ {}^{s}C_{s'}\ {}^{t}C_{t'}\ {}^{k-p-q}C_{r'}\ {}^{d_H-k-s-t}C_{u'} \tag{5.39}$$

Finally, we take the average value of the BER derived from both the 1_{st} and the 2_{nd} bit of the network coding, as they are encoded separately and are independent to each other.

5.3 Simulation result

In this section, we give the simulation and the theoretical curve of the convolutional coded PNC. Note: In reality, the theoretical curve act as the upper bound,

and is accumulated from $d_H = d_{min}$ to infinity. The more accumulation operation, the worse BER the performance curve will have. However, we aim to find the curve which is more converge with the simulation one, thus we take an finite addition of the term (e.g. from 5 to 6) without losing any generality. Because the addition of items affect less on the curve at high SNR, but much more at lower SNR.

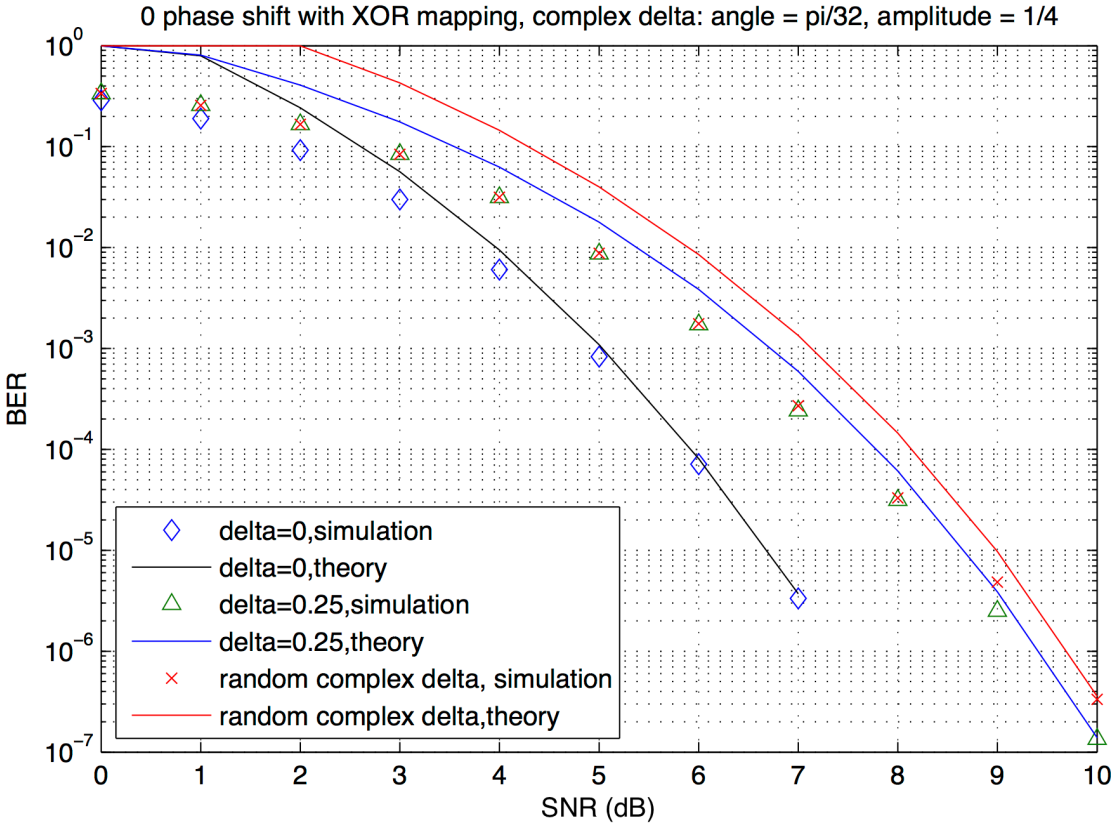

FIGURE 5.9: Fading near 0

In Fig 5.9, we compare the performance of the simulation and the union bound at a singular fading around 0. We assume that the system apply the adpative mapping, thus the XOR mapping is chosen in this situation. Recall that two channels are set as $1+\delta$ and $1-\delta$. We simulate from the equal channel ($\delta = 0$) to real fading channel ($\delta = 0.25$), and finally the complex channel ($\delta = 0.25\angle\frac{\pi}{32}$). Typically, the addition of terms in which we calculate BER stops at a Hamming distance of 6, which correspond to 2 errors in a codeword.

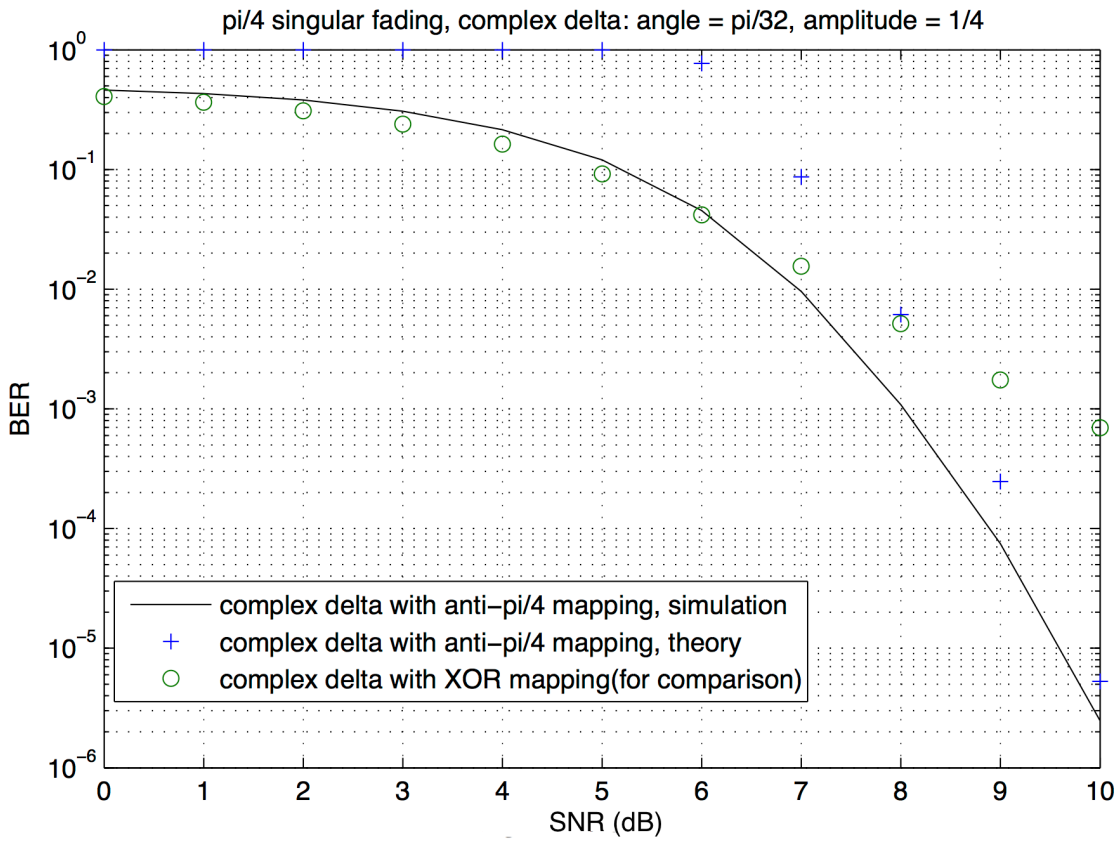

FIGURE 5.10: Fading near $\frac{\pi}{4}$

In Fig 5.10, we compare the performance of the simulation and the union bound at a singular fading around $\frac{\pi}{4}$. Again the addition of terms in which we calculate BER stops at a Hamming distance of 6, which correspond to 2 errors in a codeword.Assuming that the anti-$\frac{\pi}{4}$ mapping is chosen, we simulate the complex channel ($\delta = 0.25\angle\frac{\pi}{32}$) for both the simulation and the union bound. Also, we introduce the performance the XOR mapping is chosen, which is far worse than the anti-$\frac{\pi}{4}$ mapping. However, in the coded PNC, the difference between the performance of the above 2 mappings at a singular fading of $\frac{\pi}{4}$ is not that huge. This is because the code has improved the performance of the system by a lot, and on the other hand amplifies the advantages of the optimum mapping at the current singular fading over other fixed mappings.

5.4 Summary

With the help of the pairwise error probability and the event error probability, we derive the upper bound of the superimposed BER at the relay for joint decoding. We derive not only the XOR mapping, but also the revesed XOR mapping and the anti-$\frac{\pi}{4}$ mapping, showing that the performance could be estimated with the appropriate upper bound.

Chapter 6

PNC on OFDM channels

6.1 Uncoded PNC on OFDM channels

So far we have based the PNC on single carrier channels. The channels we choose are flat fading channels whose frequency response remains flat during one packet period. However, in real situations, we may face frequency selective/fast fading channels whose response cannot be regarded as 'flat'. Also, this system is vulnerable to time response channels which cause ISI. Thus, we introduce the OFDM method. As introduced in the section 1.3, OFDM will have advantages that allow us to optimize our system:

- By dividing the channel into narrowband flat fading sub channels, OFDM is more resistant to frequency selective fading than single carrier systems are.
- Eliminates ISI through use of a cyclic prefix.
- Using adequate channel coding and interleaving one can recover symbols lost due to the frequency selectivity of the channel. Thus we may apply a linear convolutional code to the system.
- It is possible to use maximum likelihood decoding with reasonable complexity, which is similar to the coded PNC on single-carrier system.

- OFDM is computationally efficient through using FFT techniques to implement the modulation and demodulation functions.
- Provides good protection against co-channel interference and impulsive parasitic noise.

Also, all of our previous systems are based on the assumption that the relay receives the information from two transmitters at the same time (perfect synchronization). [77] and [78] helps to achieve the theoretical work of synchronization, and [79] even moves forward to its implementation. However, perfect synchronization is quite difficult to achieve due to phase errors and noise [80]. Thus, we introduce the guard interval/cyclic prefix on OFDM channels to solve the problem at the cost of losing some affordable capacity.

6.1.1 System model

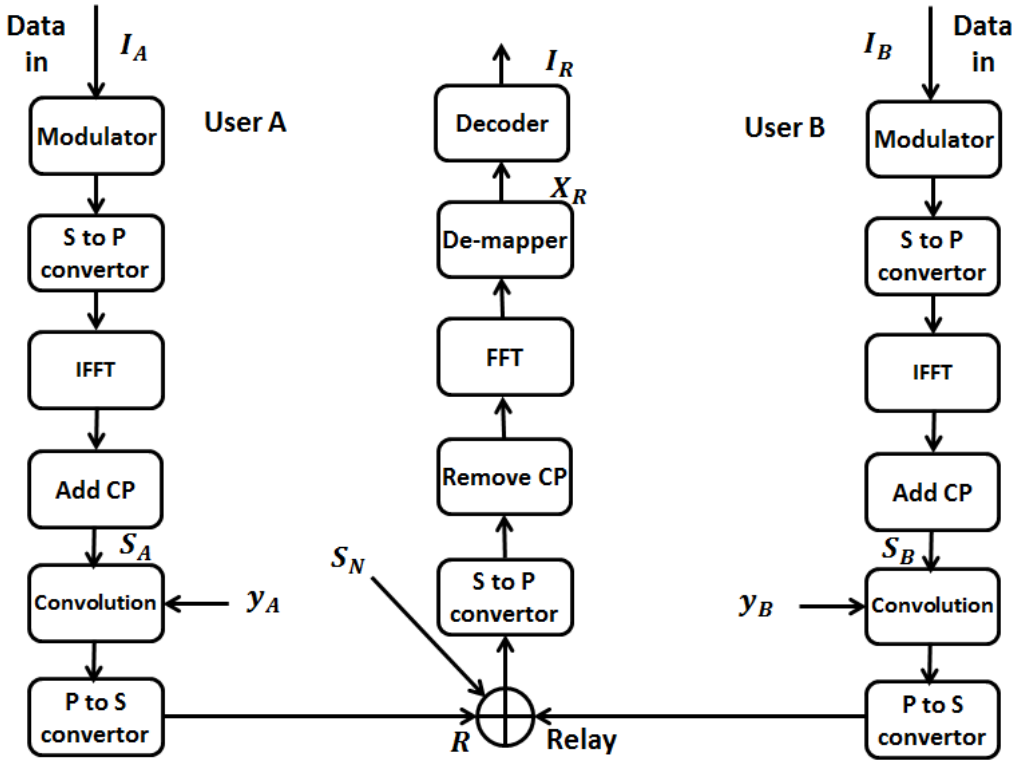

FIGURE 6.1: Simulation model of PNC on OFDM channels

We start our reseach on OFDM with un-coded PNC. The simulation model of un-coded PNC on OFDM channels is shown in Fig 6.1. After linear QPSK modulation, the sequence of data I sent from either user is fed into a serial-to-parallel convertor so as to be allocated onto the OFDM sub-carriers. The convertor converts the data sequence to a matrix whose number of row is equal to the number of sub-carriers. Typically, we make the number of sub-carriers 64 as this is the commonly applied in the industry. To do this, we set the length of the original data sequence I to be a multiple of 64 so as to fill all sub-carriers evenly (We can still fill the empty spaces on some sub-carriers with all '0's as padding if the length is not a multiple of 64).

Then, we take the inverse Fast Fourier Transform (IFFT) of the symbols on all sub-carriers to transform all symbols into the time domain.

In an OFDM system, we typically add cyclic prefix (CP) in order to counter the ISI. The size of the CP is decided by the maximum delay of the multipath channel. We define the multipath channel in Matlab as:

h = (randn(1,$L+1$) + 1i*randn(1,$L+1$))/sqrt(2*($L+1$));

where L is the maximum multipath delay, typically we set this equal to N_{cp}, so that it does not cause intersymbol or intercarrier interference. Note that the factor $\sqrt{L+1}$ is required to normalise the channel so that (on average) it does not change the signal power. In this way we can compare its performance with PNC on single-carrier. We set the number of signal samples in the cyclic prefix as 8 ($N_{CP} = L = 8$). We typically copy the time-domain samples from sample 56 to 64, and duplicate them before all the samples as the CP. Also, this channel is defined in time domain rather than frequency domain.

After that, the time-domain symbols are convoluted with the time delay channel in time domain.

Finally, symbols from both users are transformed back to serial form and accumulated at the relay together with noise. We assume that R_s is the received signal at the relay, h_A and h_B represent the channels' frequency responses, S_A and S_B

are the modulated signal sent from both users, and S_N is the noise symbol at the relay. Then, for PNC on single-carrier, we already know that the relay will receive

$$R_s = h_A S_A + h_B S_B + S_N \tag{6.1}$$

in which QPSK symbols are transmitted in frequency domain and are directly multiplied by the channel frequency response.

However, the symbol transmission of PNC on OFDM channels is in the time domain. Thus, symbols on each sub-carrier are convolved with the related sub-carrier's channel impulse response separately rather than multiplied by the channel frequency response. As a result, the relay receives:

$$R_s = y_A * S_A + y_B * S_B + S_N \tag{6.2}$$

where $*$ denotes the convolution operation, and y_A and y_B denotes the two channel's impulse responses corresponding to the frequency response h. By considering parallel transmission in OFDM, we divide the channels into 64 sub-channels, with the impulse response y_A^1 to y_A^{64} and y_B^1 to y_B^{64}. Based on that, we can obtain the sub-channels' frequency response h_A^1 to h_A^{64} and h_B^1 to h_B^{64} respectively.

At the relay, the superimposed sequence of symbols are transformed to parallel form first. We then remove the CP which is "polluted" by the ISI, and truncate the OFDM frames on each sub-carrier by N_{CP}. After that, we apply the fast Fourier Transform (FFT) on all samples to transform them back to the frequency domain. This is followed by de-mapping and decoding so as to decode the received symbols into linear superimposed symbols.

The de-mapping process on each sub-carrier is similar to that of un-coded PNC in a single-carrier system in which we calculate the Euclidean distance against each element in the look-up table. However, the realization of de-mapping is much more complicated here. In the single-carrier system, we define the channels' frequency

responses directly as h_A and h_B, and the look-up table has a size of $1 \times (16 \times 2)$ for each pair of transmission channels (h_A and h_B), as shown in table 3.2. In our OFDM system, we have 64 sub-carriers, and each of them have a such look-up table. These will be merged to a combined look-up table of size $64 \times (16 \times 2)$ shown in table 6.1.

Constellation (R)	de-mapped vector (X_R)	until	Constellation (R)	de-mapped vector (X_R)
$\frac{-h_A^1(1+j)-h_B^1(1+j)}{\sqrt{2}}$	0 0 0 0	until	$\frac{h_A^1(1+j)+h_B^1(1+j)}{\sqrt{2}}$	1 1 1 1
$\frac{-h_A^2(1+j)-h_B^2(1+j)}{\sqrt{2}}$	0 0 0 0	until	$\frac{h_A^2(1+j)+h_B^2(1+j)}{\sqrt{2}}$	1 1 1 1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$\frac{-h_A^{64}(1+j)-h_B^{64}(1+j)}{\sqrt{2}}$	0 0 0 0	until	$\frac{h_A^{64}(1+j)+h_B^{64}(1+j)}{\sqrt{2}}$	1 1 1 1

TABLE 6.1: The look-up table for PNC on OFDM channel

Note: in the single-carrier system, all transmissions are assumed to be in the frequency domain only. However, in OFDM, the symbols are convolved with channels' impulse response in time domain. Thus, we will have to transfer the impulse response y associate with each sub-carrier into frequency response h so as to fill them in the look-up table. Thus, for each received OFDM frame, the de-mapped symbol X_R becomes a matrix of size 64×4:

$$X_R^n = \{I_A^n(1) \quad I_A^n(2) \quad I_B^n(1) \quad I_B^n(2)\}$$

where X_R^n is the de-mapped symbol on the n^{th} sub-carrier, and $I_A^n(1)$ is the 1_{st} bit of the network coded symbol sent from user A on the n^{th} sub-carrier.

6.1.2 Mapping selection

Recall that in the QPSK modulated PNC on single-carrier channels, we assign different mappings to the relay in order to counter different singular fadings which are

formed by the frequency response of both channels (h_A and h_B). These mappings are also available for PNC on the OFDM channel as well. However, the OFDM channel involves multiple sub-channels, and accordingly we assign mappings to each sub-carrier/pair of sub-channel (here he word pair means the sub-carrier at both users with the same position, e.g : h_A^1 and h_B^1). As a result, relay may have different mappings associated with different sub-carriers (adaptive mapping), or the same mappings for sub-carriers which may not be the optimum mapping under the singular fading (fixed mapping).

As we introduced in the previous chapters, there are 3 basic mappings available for QPSK modulated PNC, described as the XOR mapping, the reversed XOR mapping, and the anti-$\frac{\pi}{4}$ mapping. All other mappings are the different combinations of these three.

6.1.2.1 Adaptive mapping

The adaptive mapping on OFDM is very similar to that applied to the single carrier channels. The relay treats each sub-carrier as an independent singular fading and feedback the best mapping (in terms of SER) on each sub-carrier to both transmitters separately. The expression of SER under different singular fadings has been introduced in chapter 3 already.

For PNC on single carrier channels, the mappings remain the same within each packet for a given singular fading as we assume that all channels are invariant during one packet time. Likewise, for PNC on OFDM channels, we can make a similar assumption that all sub-channels are invariant during one sub-carrier packet time.

However, due to the parallel-to-serial conversion at both transmitter, the transmitted OFDM packets are converted from the parallel sub-carrier packets, and the adjacent symbols in the OFDM packets comes from different sub-carriers. As a result, if different mappings have been assigned to adjacent sub-carriers, we then need to apply different mappings in the decoding of adjacent symbols in OFDM

frames. This is possible in uncoded PNC, but if we consider the application of a linear codeword to the system, we will immediately find difficulty in decoding the superimposed symbols to a linear combination of the original binary sequence. This is because the decoding of a linear codeword require a sequence of data rather than decoding the information bit-by-bit. Instead, we will expect a scheme in which a fixed mapping is associated with all/some sub-carriers, or at least inside each sub-band. These optimized mappings will be introduced in the next section.

6.1.2.2 Fixed mapping

We consider applying a certain fixed mapping to all sub-carriers as we wish to further apply a linear channel code to the system. As a result, we need to make the system choose the best mapping among all 3 fixed mappings.

The majority vote mapping is one of the possible solutions for the PNC on the OFDM channels. In this mapping, a "vote" is held in the relay to determine which fixed mapping is going to be applied to sub-carriers. Its algorithm can be seen from the flow chart in Fig 6.2

To realize it, the relay will examine the fading associated with sub-carriers against the thresholds between different mappings. The thresholds has been introduced in the section 'adaptive mapping' in chapter 1. In this way, relay can quickly determine which mapping is the optimum for a certain sub-carrier. Next, relay gives a "vote" or a "token" to that chosen mapping, and will move on to the next sub-carrier. When voting is finished, relay will count the "token" held by each mapping, and the mapping with the most tokens win. Once the result is "published", all sub-carriers will accept that winning mapping as a compromise.

However, in some cases, 2 different mappings may hold same number of tokens, which will make the voting result more complicated. To solve this, we define the priority of these mappings as an invisible rule, which is: $M_1 > M_2 > M_3$. We know that M_3 (the anti-$\frac{\pi}{4}$ mapping) performs the worst across all fading states, thus we give it the lowest priority. The other two mappings-M_1: the XOR mapping

and M_2: the reversed XOR mapping-have the same SER performance across all singular fading. However for simplicity, we give the XOR mapping the highest priority over the reversed XOR mapping. We express all these rules in algorithm 3.

Algorithm 3 The majority vote mapping

```
t_1 = t_2 = t + 3 = 0
while S_r == 1 do
  for i = 1, i < 64, i++ do
    if f_i ∈ th_3 then
      t_3++
    else if f_i ∈ th_2 then
      t_2++
    else
      t_1++
    end if
  end for
  if t_1 > t_2 then
    loop
      if t_1 > t_3 then
        M = M_1
      else
        M = M_3
      end loop
    else if t_2 > t_3 then
      M = M_2
    else
      M = M_3
    end if
  end loop
end while
```

where f_i is the singular fading of the current sub-carrier, t_1 to t_3 and th_1 to th_3 are the tokens and threshold associate with the three mappings. In the for loop, we start from th_3, because the threshold of the anti-$\frac{\pi}{4}$ is the amplitude, while th_1 and th_2 are all phases.

We can further adapt this mapping into the local optimum mapping which is inspired by the system-level simulation. This mapping calculates the three average SER in which the three fixed mappings are applied to all sub-carriers. The best effort mapping is better than the majority vote mapping in the cases that a particular fixed mapping has the most vote, but the application of it to all sub-carriers

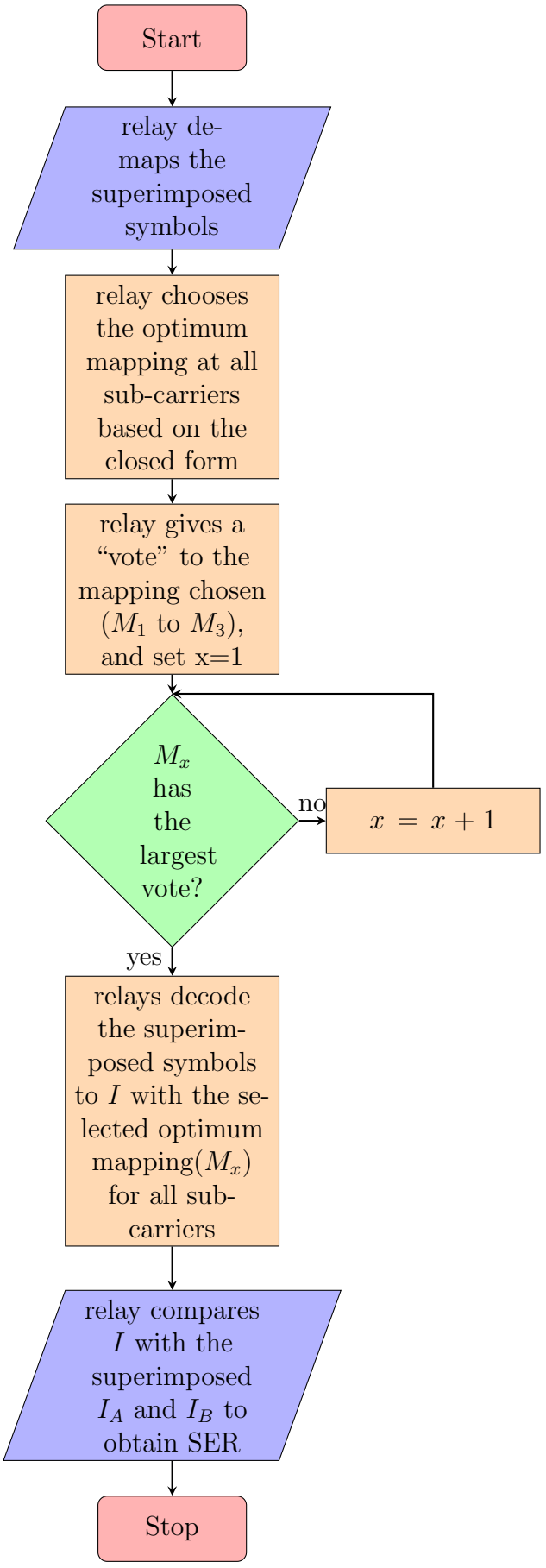

FIGURE 6.2: The majority vote mapping in flow chart

may not result in the lowest overall SER compared with the SER obtained from the application of another fixed mapping to sub-carriers. Its algorithm can be seen from the flow chart in Fig 6.3, and is further explained in algorithm 3.

Algorithm 4 The local optimum mapping

```
while S_r == 1 do
  for i = 1, i < 64, i++ do
    for j = 1, j < 3, j++ do
      P_sj = P_sj + P_sj(i)
    end for
  end for
  if P_s1 > P_s2 then
    loop
      if P_s1 > P_s3 then
        M = M_1
      else
        M = M_3
      end loop
    else if P_s2 > P_s3 then
      M = M_2
    else
      M = M_3
    end if
  end loop
end while
```

where P_{sj} is the accumulated overall SER for mapping M_j, and $P_{sj}(i)$ is the SER for the application of mapping M_j on the ith sub-carrier. Again we give the priority sequence as $M_1 > M_2 > M_3$.

The choice of mappings is highly dependent on the type of system. Overall, adaptive mapping is the best if we consider the un-coded PNC on OFDM channels. If we require the system to be upgraded to a linear coded system, then the best effort mapping would be favoured over the majority vote mapping at the cost of introducing more complexity.

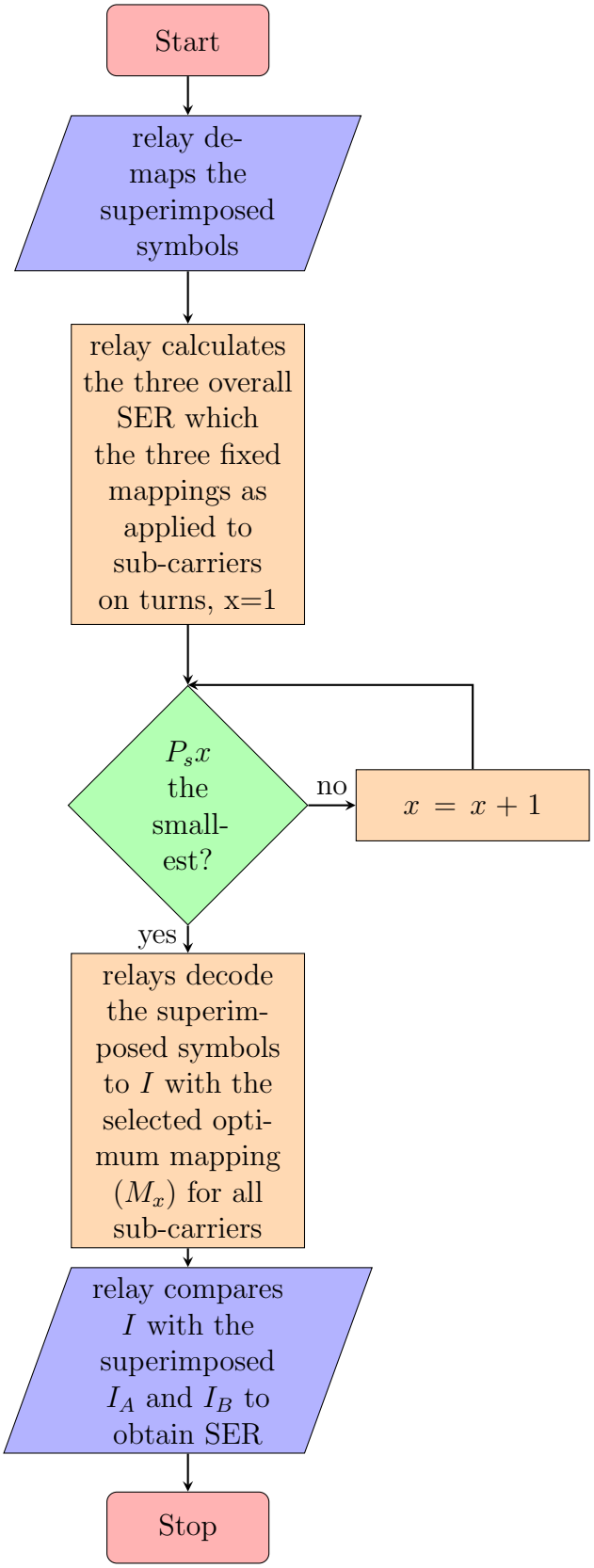

FIGURE 6.3: The local optimum mapping in flow chart

6.1.2.3 The sub-band

So far we have introduced the situations which the two optimized mappings are applied to all sub-carriers. These method may have a large degradation of the overall SER compared with the adaptive mapping. As a result, we consider applying either the majority vote or the best effort mapping in a number of consecutive sub-carriers, which we name as "sub-band". If the size of the sub-band is reduce to 1, then it is the same as the adaptive mapping. On the other hand, if the size is large as 64, then the mapping result will be the same as that in the last section. The smaller sub-band size, the more accurate (the word accurate here means either the lower the SER, or the closer its performance to the fully adaptive case) the system behaves. However, in order to adapt the system to the linear coded PNC on OFDM channels, we need to set the minimum limit for the cluster. We could apply one sub-carrier at a time though, but the problem then would be complexity. Because if we do so, we would need 64 encoders and decoders, and also that the coded frames would need to be quite long to fit a reasonable length codeword. Thus, we set the size of sub-band as 8 sub-carriers, as the maximum channel impulse response length is 8, and the coherence bandwidth is also likely to be 8. This setting will be further explained in the next section which we further apply a linear channel code to the system.

We may even consider applying unbalanced sub-band which the size of the sub-band may not be fixed. This method is typically useful when there are consecutive channels that is optimum for one particular mapping. e.g. If the best mapping for the initial 16 sub-carrier is:

$$M_1, M_1, M_1, M_1, M_1, M_2, M_2, M_2$$
$$M_2, M_3, M_3, M_3, M_1, M_1, M_1, M_1$$

Then, instead of having two sub-bands as

$\{M_1, M_1, M_1, M_1, M_1, M_1, M_1, M_1\}$ and $\{M_1, M_1, M_1, M_1, M_1, M_1, M_1, M_1\}$

we may have 4 sub-bands as: $\{M_1, M_1, M_1, M_1, M_1\}$, $\{M_2, M_2, M_2, M_2, M_2\}$, $\{M_3, M_3, M_3\}$ and $\{M_1, M_1, M_1, M_1\}$

This optimization will help the SER performance of the system approach closer to that of the adaptive channel method, but adds more complexity. Furthermore, it is vulnerable to the situation that there are too many individual sub-carriers that have different singular fading with their neighbors.

6.2 Convolutional coded PNC on OFDM channels

Finally we combine all the optimization methods onto the PNC system which includes the mapping choosing, the convolutional code, and the OFDM channel (sub-band included).

Like the uncoded PNC on OFDM channels, we set the number of sub-carriers as 64 with a cyclic prefix of 8. Taking into account that the sub-band will help improve the performance of the system, we introduce 8 sub-bands with a bandwidth of 8 adjacent sub-carriers to the model.

However, if we apply different mappings to different sub-bands or sub-carriers, we would need to use separate encoders/decoders on each of them. As a result, we could either make each user have only 1 encoder, or separate encoders assigned to each sub-band.

6.2.1 System model

6.2.1.1 Single encoder

The system model of the coded OFDM with 1 encoder is shown in Fig 6.4, and the relay part is shown in Fig 6.5. As can be seen, the encoder module (consisting

of encoder, interleaver and modulator) is added to the system ahead of the OFDM module.

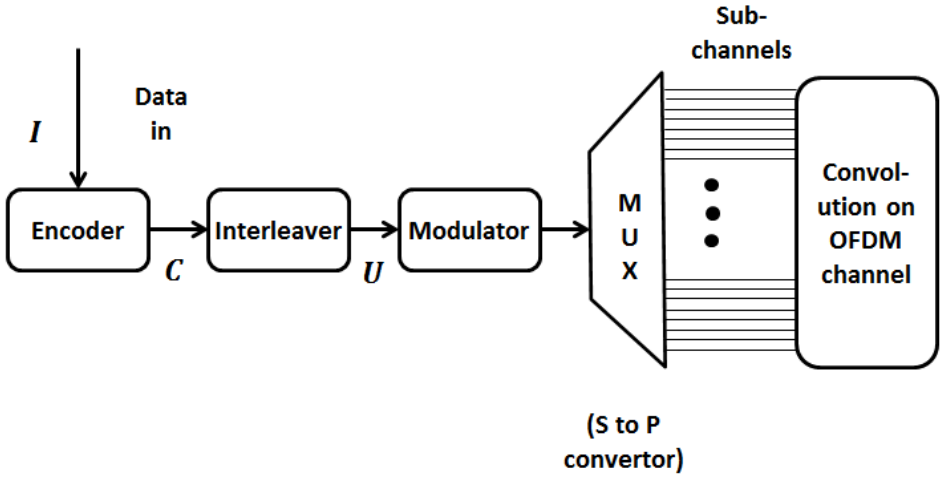

FIGURE 6.4: System model of coded PNC on OFDM channels (1 encoder)

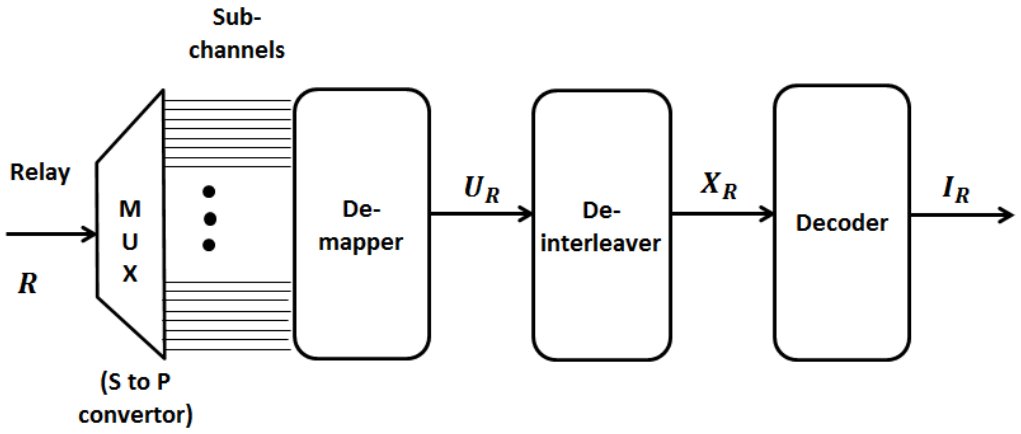

FIGURE 6.5: System model of coded PNC on OFDM channels (1 decoder)

Like coded PNC on single-carrier system, we have a block interleaver locate in between encoder and modulator, and a block de-interleaver between encoder and modulator. These prevent the adjacent bits in the same codeword to 'suffer' from the same channel fading so as to improve the overall performance.

This model is comparatively less complicated as there is only 1 encoder at each user (which also means only 1 decoder is required at the relay). As a result, 1 particular mapping should be applied to all of the sub-carriers, which means the sub-bands are no longer needed here. Thus, symbols on some sub-carriers would be heavily faded if the mapping is not the optimum mapping for its fading state.

To solve this, we make some sub-bands have different mappings with their neighbours. However, the data stream feed into the interleaver is a linear codeword. If the sub-bands have multiple mappings, then the correpondent codeword message will also change, which may makes the correspondent code sequence of the actual transmitted symbols maynot be a linear codeword. Thus, the PNC on the OFDM channels with one encoder cannot have the sub-band settings.

6.2.1.2 Multiple encoder

We thus use OFDM channels with multiple encoders as shown in Fig 6.6. As can be seen, the symbols feed into the serial to parallel convertor is the binary sequence, which is same as the system we introduced in the last chapter.

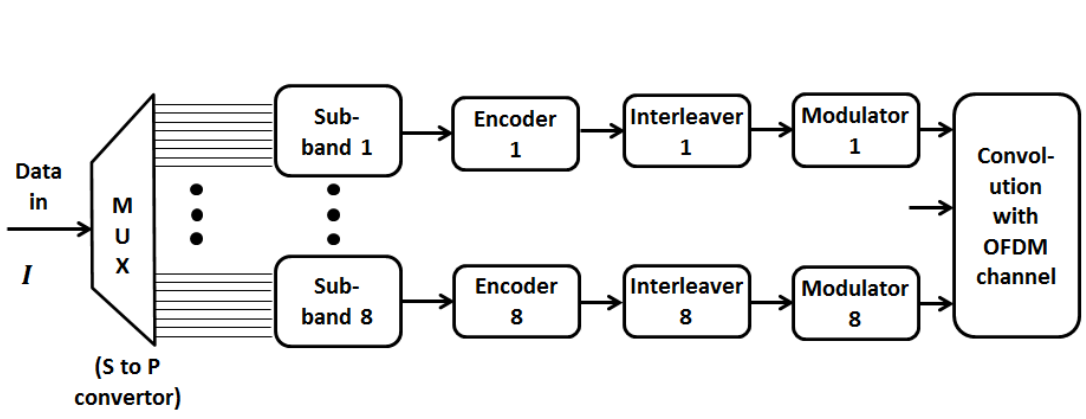

FIGURE 6.6: System model of coded PNC on OFDM channels(multiple encoder)

At the cost of introducing some more complexity ($(N-1)$ times more encoder units, where N is the number of sub-bands), we can now assign different mappings for each sub-band according to the relative fading state of the sub-carriers. If we still

apply a (5,7) convolutional encoder to all sub-carrier, then the decoding section will consist of 8 (5,7) decoders.

We can further adapt the system by applying different linear codes (with different rates) to different sub-bands. This includes the Convolutionl code with longer codeword(e,g, (11,13)), and other linear channel code (Turbo, LDPC, etc) as long as the adaptive modulation could make the modulated sequence on each sub-carrier have the same length.

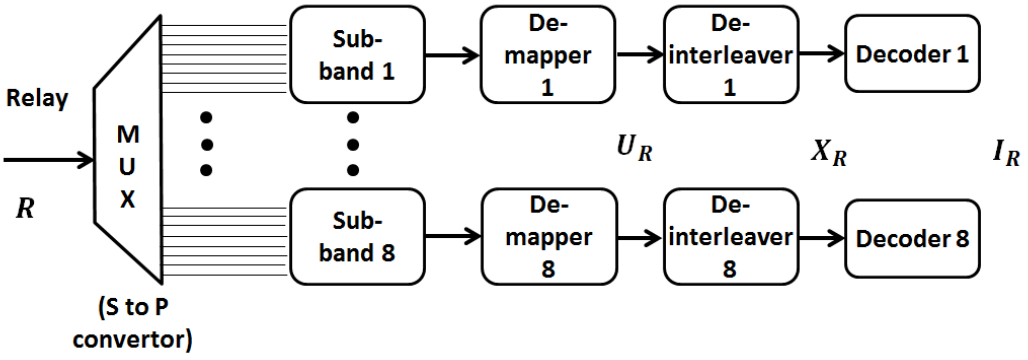

FIGURE 6.7: System model of coded PNC on OFDM channels (multiple decoder)

As a result, we have multiple linear decoders at the receiver corresponding to different sub-bands as Fig 6.7 shows, and the decoding is carried out in each sub-band independently.

6.3 Simulation result

Figure 6.8 and 6.9 shows the SER of the uncoded PNC on OFDM channels with majority vote and local optimum mappings in various sub-bands respectively. We simulate across different Rayleigh fading channels with multipath delays. As predicted, the larger the sub-band size, the poorer SER performance the system have, and the performance curve movers closer to the case where no sub-band is applied. On the other hand, the smaller the sub-band size, the better SER performance, and the performance curve movers closer to that where the adaptive channel method is applied.

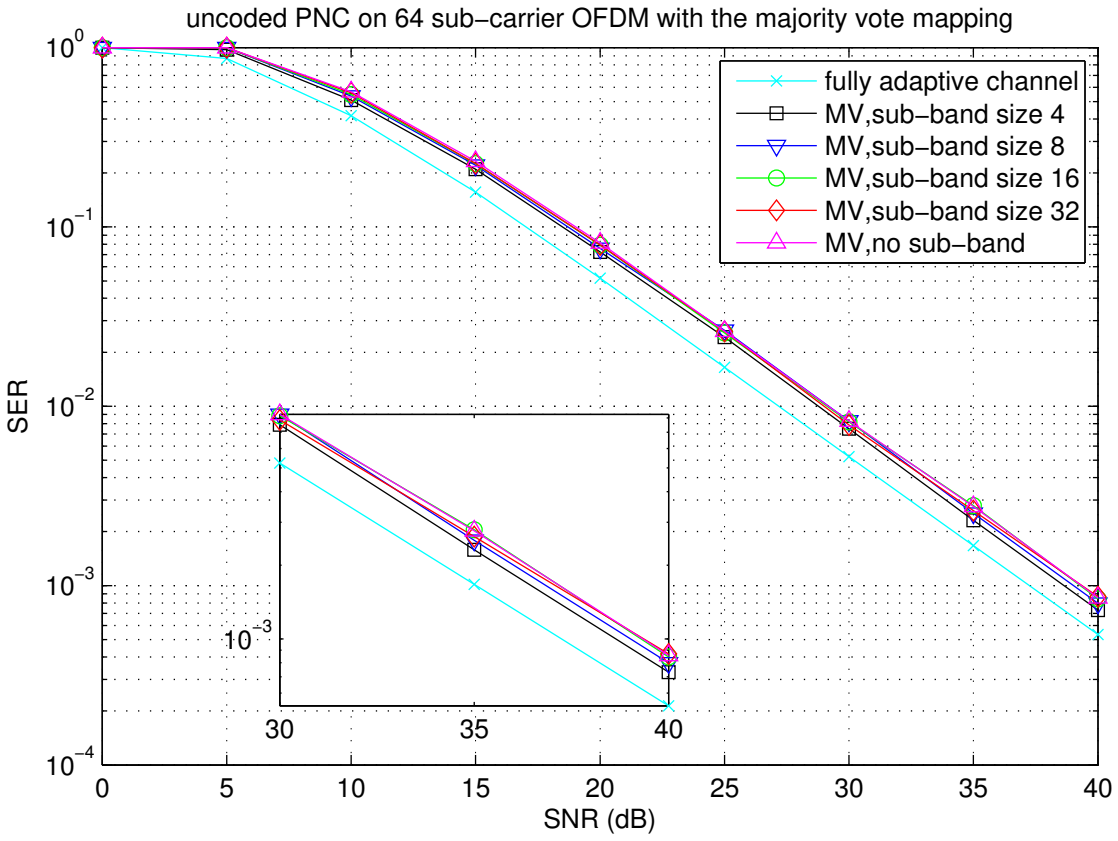

FIGURE 6.8: Majority vote mapping with sub-band

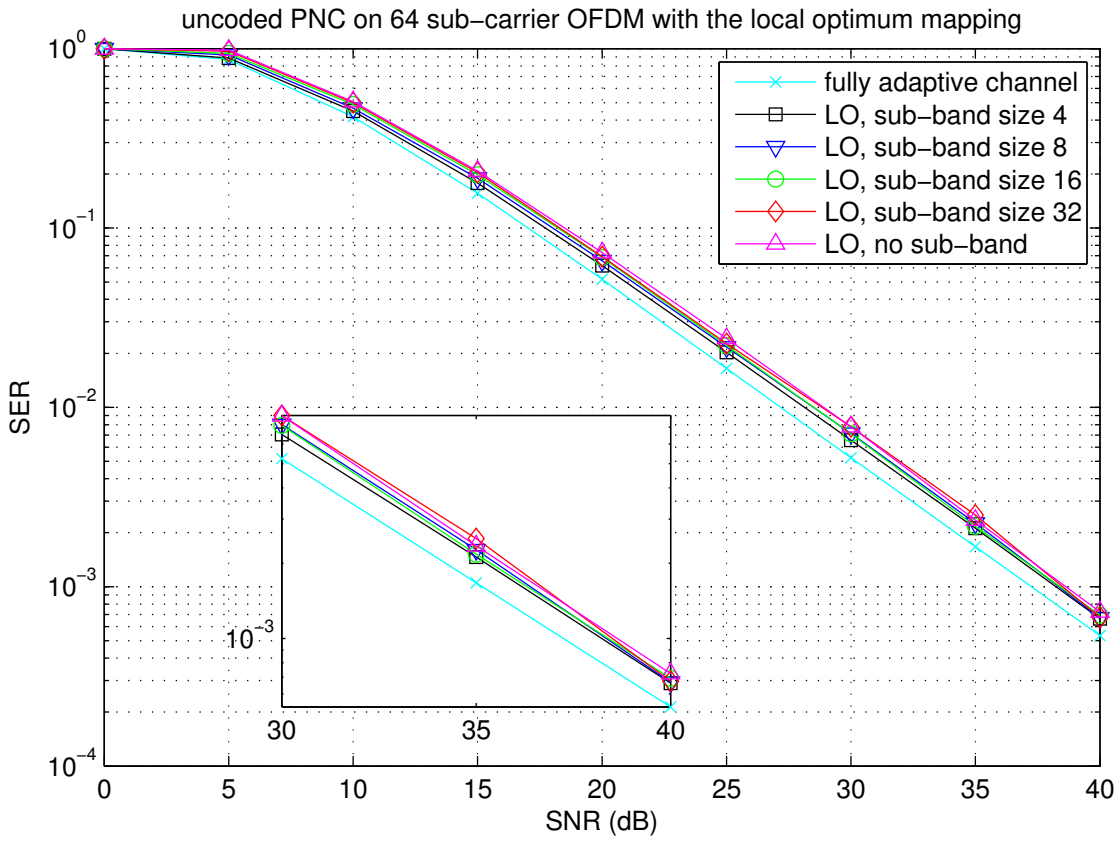

FIGURE 6.9: Local optimum mapping with sub-band

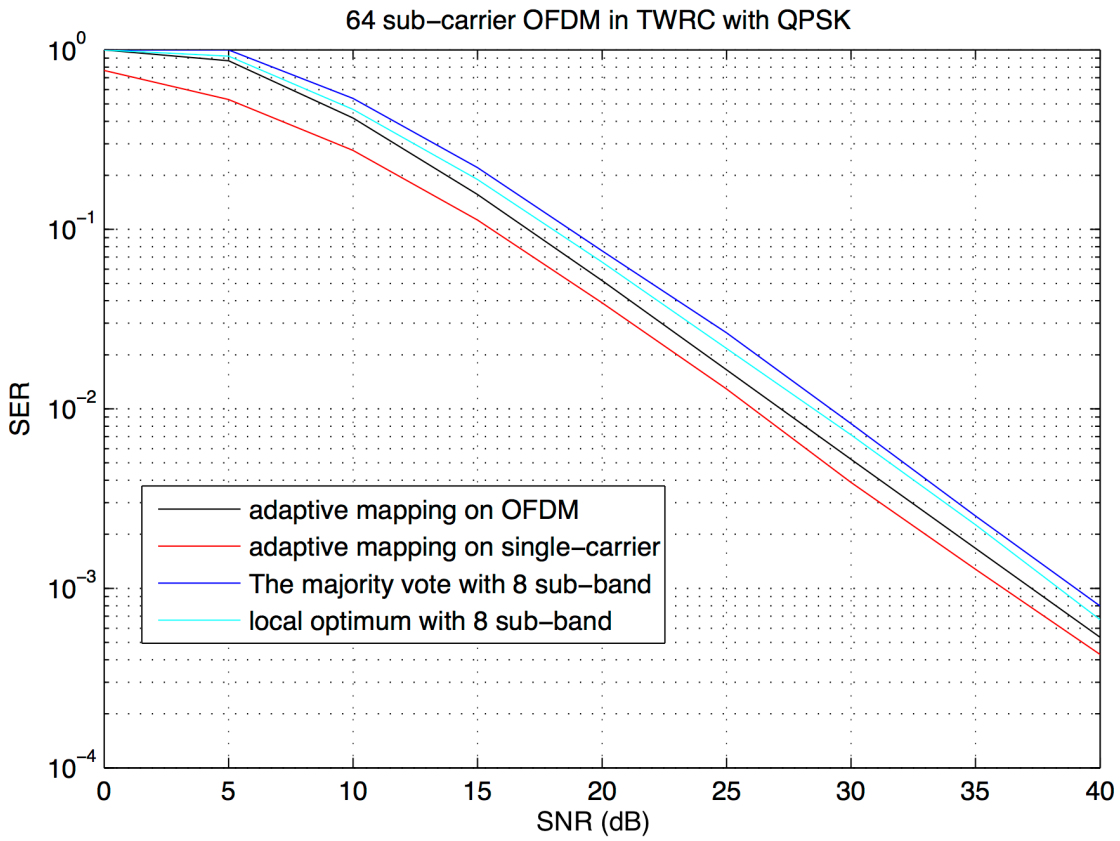

FIGURE 6.10: Performance comparison for uncoded PNC

Figure 6.10 shows the comparison of SER performance of the adaptive mapping (fully adaptive channel) on OFDM channel, the adaptive mapping on single-carrier channel, the 8 sub-band majority vote mapping, and the 8 sub-band best effort mapping. It can be seen that with the same sub-band size, the best-effort mapping performs slightly better than majority vote (by approximately $0.5dB$) at the cost of introducing some complexity. Also, we note that for adaptive mapping, the performance on the OFDM channel is worse than that on the single-carrier channel. This results from the loss of energy in the cyclic prefix.

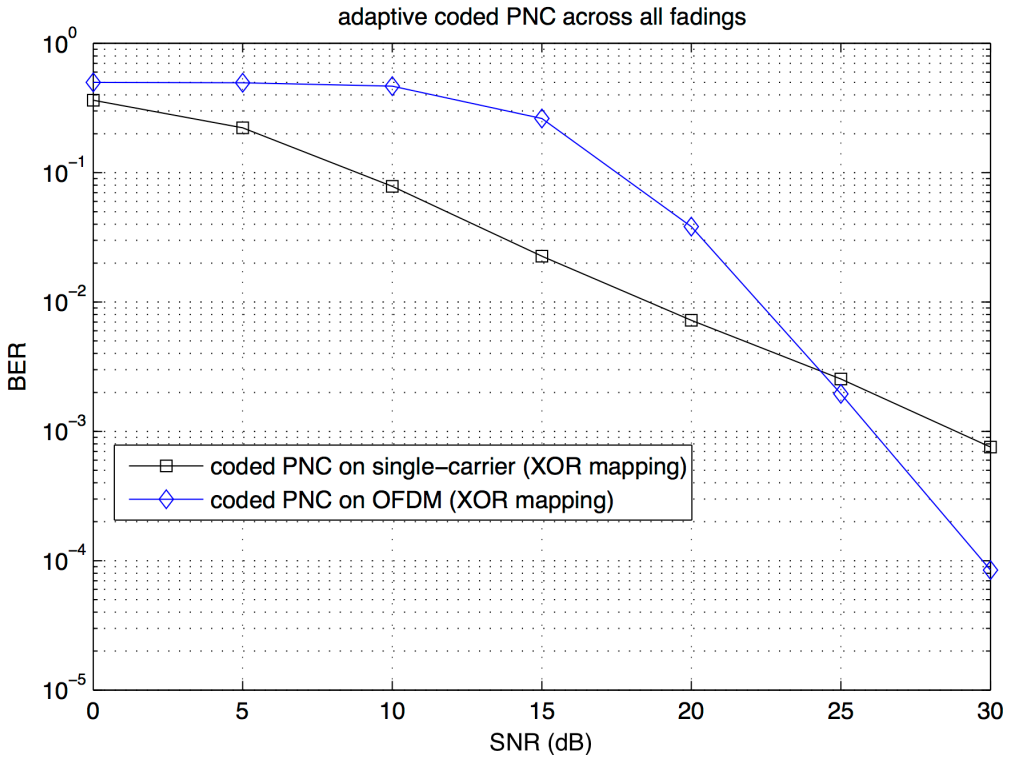

FIGURE 6.11: Performance comparison for coded PNC

Figure 6.11 shows the comparison between the coded PNC on OFDM (1 encoder, local optimum mapping) and the uncoded PNC on single-carrier channel across all singular fading states (average BER). As can be seen, the coded PNC on OFDM performs poorer in lower SNR, which may be caused by the energy loss in the CP. Still, the BER curve of the coded PNC on OFDM has a reasonable diversity and surpass that on single-carrier channel as it benefits from the tight coherent bandwidth.

6.4 Summary

We simulate the PNC on OFDM channels in this chapter, and introduce the mappings (fixed, fully adaptive) across all sub-carriers. Furthermore, we introduce the mappings inside every sub-band in order to realize the convolutional coded PNC on OFDM channels. This includes both the multi-encoder and single-encoder models.

Chapter 7

Conclusion and Further Work

7.1 Conclusion

In this thesis, our research aim is to build a system-level simulation of multi-hop (multiple PNC) networks. To do this, we focus on the derivation of the error rate of the superimposed signals received at the relay. The error rate is a tight theoretical upper bound, and can be used for feeding back to each relay the best mapping. For uncoded PNC, we calculate the superimposed SER at the relay; for channel coded PNC, we calculate the superimposed BER at the relay.

In Chapter 3, we introduce the QPSK modulated PNC model, and three fixed mappings for different relative fading states (in both amplitude and phase). Also, we introduce the adaptive mapping which 'intelligently' chooses the best fixed mapping according to the different fading states. In addition to these, we derive the theoretical upper bound of the superimposed SER at the relay for all the mappings under all fadings, and simulate the SER graph which compares the upper bound with the performance of the symbol-level simulation of the system. According to the result, we are confident to show that the derived upper bound is tight.

In Chapter 4, we build two models that apply system-level simulation. In these models, The SER of each mapping (out of 3) is calculated at the node in next

phase, and feedback the best realizable mapping (the mapping that result in the lowest SER) to the current phase. Typically, we design the algorithm for these progress shown in the flow chart. Finally, we obtain the overall PER through symbol-level simulation, and the theoretical upper bound of PER as the system-level simulation. Again the upper bound is tight.

In Chapter 5, we equip our PNC model with a linear channel code (convolutional code), as it can boost the performance of the transmission (in terms of error rate). Instead of separate decoding (called Jt-CNC), we still use joint decoding to obtain the superimposed BER after taking the size of the trellis into account. We then focus on the derivation of the superimposed BER at the relay from the pairwise-error probability, the event-error probability, and finally accumulate the terms into an upper bound on the superimposed BER. We take an finite summation of terms as we want a tight upper bound, which is shown in the simulation result.

In Chapter 6, we introduce the OFDM channel to reduce the possible consequences resulting from the time delay channels and frequency selective channels. These consequences may reduce the performance of the single-carrier system. In addition, we introduce the sub-band which enables the adaptive mapping within each sub-band, and the application of linear channel coding. By comparison, we determine that PNC on an OFDM channel performs as good as that on the single-carrier channel despite slightly poorer error performance due to the CP.

7.2 Further Work

- We have so far assume that the relay have a perfect channel estimation. However, this is quite difficult in the industry (e.g. the affection of Doppler effect). Thus we may consider a proper channel estimation which is able to estimate the channel dynamically.

- We may consider higher order modulation to be applied in our PNC system. e.g. 64 QAM, and the possibility of extending the current BER theory to

it. Due to the complexity of 64 QAM, we can no longer apply exhaustive search to it. thus, we may consider some other algorithms.

- We may apply some other channel coding which are more powerful. e.g. turbo code and LDPC code. At the cost of introducing more complexity, we could expect to have a better BER performance.

- For the coded PNC on OFDM channel, we may apply some other interleaving skills. e.g. interleave within the sub-band. We will compare coded interleaved OFDM without subbands with the subband based approach. Also, we may consider some other interleavers other than the simplest block interleaver so as to further improve the performance.

Glossary

PNC	**P**hysical layer **N**etwork **C**oding
TWRC	**T**wo **W**ay **R**elay **C**hannel
DIWINE	**D**ense cooperat**I**ve **WI**reless cloud **NE**twork
BER	**B**it **E**rror **R**ate
SER	**S**ymbol **E**rror **R**ate
PER	**P**acket **E**rror **R**ate
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
CDMA	**C**ode **D**ivision **M**ultiple **A**ccess
MIMO	**M**ultiple-**I**nput and **M**ultiple-**O**utput
MAC	**M**ultiple **A**ccess **C**hannel
dB	Decibel
SNR	**S**ignal to **N**oice **R**atio
ML	**M**aximum **L**ikelihood
LLR	**L**og **L**ikelihood **R**atio
HDF	**H**ierachical **D**ecode and **F**orward

Bibliography

[1] Y. Wang, J. Xu, and L. Jiang, "Challenges of system-level simulations and performance evaluation for 5g wireless networks," *IEEE Access*, vol. 2, pp. 1553–1561, 2014.

[2] "diwine project," http://diwine-project.eu/public/, accessed: 2017-06-04.

[3] A. Burr, C. Chen, M. Molu, K. Ramantas, and J. S. Vardakas, "System-level simulation of multihop wireless networks using physical-layer network coding," in *IEEE EuCNC*. IEEE, 2015, pp. 285–289.

[4] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proc. IEEE*, vol. 99, no. 3, pp. 438–460, 2011.

[5] R. H. Weber and R. Weber, *Internet of things*. Springer, 2010, vol. 12.

[6] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE Trans. Commun.*, vol. 27, no. 5, pp. 773–787, Jun 2009.

[7] Y. Wu, B. Ledoux, A. Bergeron, and B. Caron, "OFDM for digital television terrestrial distribution over channels with multipath and non-linear distortions," 1995.

[8] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 358–365.

[9] "Physical layer," https://en.wikipedia.org/wiki/Physical_layerl, accessed: 2017-06-04.

[10] "Osi model," https://en.wikipedia.org/wiki/OSI_model, accessed: 2017-06-04.

[11] A. Kaye and D. George, "Transmission of multiplexed PAM signals over multiple channel and diversity systems," *IEEE Trans. Commun. Technol.*, vol. 18, no. 5, pp. 520–526, Oct. 1970.

[12] L. H. Brandenburg and A. D. Wyner, "Capacity of the gaussian channel with memory: The multivariate case," *The Bell System Technical Journal*, vol. 53, no. 5, pp. 745–778, May 1974.

[13] W. van Etten, "Maximum likelihood receiver for multiple channel transmission systems," *IEEE Trans. Commun.*, vol. 24, no. 2, pp. 276–283, Feb 1976.

[14] S. Moshavi, "Multi-user detection for ds-cdma communications," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 124–136, 1996.

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

[16] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow." *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000. [Online]. Available: http://dblp.uni-trier.de/db/journals/tit/tit46.html#AhlswedeCLY00

[17] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec 2004.

[18] P. C. Ng and S. C. Liew, "Offered load control in ieee 802.11 multi-hop ad-hoc networks," in *IEEE Cat. No.04EX975*, Oct 2004, pp. 80–89.

[19] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 572–584, September 1979.

[20] Y. Wu, P. A. Chou, S.-Y. Kung *et al.*, "Information exchange in wireless networks with network coding and physical-layer broadcast," MSR-TR-2004, Tech. Rep., 2005.

[21] C. Fragouli, J. yves Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM CCR*, 2006.

[22] S. J. Kim, N. Devroye, P. Mitran, and V. Tarokh, "Comparison of bidirectional relaying protocols," in *IEEE SARNOFF*, Apr 2008, pp. 1–5.

[23] S. J. Kim, P. Mitran, and V. Tarokh, "Performance bounds for bidirectional coded cooperation protocols," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5235–5241, Nov 2008.

[24] P. Popovski and H. Yomo, "Physical network coding in two-way wireless relay channels," in *IEEE ICC*, June 2007, pp. 707–712.

[25] D. Fang and A. Burr, "Rotationally invariant coded modulation for physical layer network coding in two-way relay fading channel," in *in IEEE EW*, April 2012, pp. 1–6.

[26] "Convolutional code," https://en.wikipedia.org/wiki/Convolutional_code, accessed: 2017-06-04.

[27] A. Burr, *Modulation and coding: for wireless communications*.

[28] G. Forney, "Convolutional codes i: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. 16, no. 6, pp. 720–738, Nov 1970.

[29] E. A. Lee and D. G. Messerschmitt, "Digital communication," 1994.

[30] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[31] D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.

[32] T. K. Truong, M. T. Shih, I. S. Reed, and E. H. Satorius, "A vlsi design for a trace-back viterbi decoder," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 616–624, Mar 1992.

[33] A. J. Viterbi and J. K. Omura, *Principles of digital communication and coding.* Nabu Press, 2011.

[34] A. Burr, "Bounds and approximations for the bit error probability of convolutional codes," *Electronics Letters*, vol. 29, no. 14, pp. 1287–1288, July 1993.

[35] M. Thejovathi and E. N. Rao, "Multilayer coding mechanisms for broadcasting over mimo networks," *IJACSI Commum Networks*, vol. 3, no. 2, p. 84, 2013.

[36] D. Fang and A. Burr, "Multilevel coded linear physical-layer network coding with extended mapping in galois field for rayleigh fading two-way relay channels," in *IEEE PIMRC*. IEEE, 2013, pp. 95–99.

[37] "Coded modulation," https://www.chalmers.se/en/projects/Pages/Coded-modulation-EN.aspx, accessed: 2017-06-04.

[38] J. B. Anderson and A. Svensson, *Coded modulation systems.* Springer Science & Business Media, 2003.

[39] W. Y. Zou and Y. Wu, "Cofdm: an overview," *IEEE Trans. Broadcast.*, vol. 41, no. 1, pp. 1–8, Mar 1995.

[40] I. Kalet, "The multitone channel," *IEEE Trans. Commun.*, vol. 37, no. 2, pp. 119–124, Feb 1989.

[41] A. Ruiz, J. M. Cioffi, and S. Kasturia, "Discrete multiple tone modulation with coset coding for the spectrally shaped channel," *IEEE Trans. Commun.*, vol. 40, no. 6, pp. 1012–1029, Jun 1992.

[42] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 628–634, Oct 1971.

[43] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part ii: State of the art," *IEEE Commun. Mag.*, vol. 25, no. 2, pp. 12–21, February 1987.

[44] E. Biglieri, D. Divsalar, M. K. Simon, P. J. McLane, and J. Griffin, *Introduction to trellis-coded modulation with applications.* Prentice-Hall, Inc., 1991.

[45] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Electromagn. Compat.*, vol. 44, no. 1, pp. 249–258, 2002.

[46] W. Card and P. Chaudhari, "Characteristics of burst noise," $IEEE_{ME}MC, vol.\ 53, no.\ 6, pp.652 - -653, 1965.$

[47] Y. Wu and W. Y. Zou, "Performance simulation of cofdm for tv broadcast application," *SMPTE Journal*, vol. 104, no. 5, pp. 258–265, May 1995.

[48] *Task Force on the COFDM of the Transmission Expert Group*, Jan 1994.

[49] J. J. Nicolas, "Investigation of coding and equalization for the digital HDTV terrestrial broadcast channel," Ph.D. dissertation, Massachusetts Institute of Technology, 1994.

[50] L. Hanzo, C. H. Wong, and M.-S. Yee, *Adaptive wireless transceivers: turbo-coded, turbo-equalised and space-time coded TDMA, CDMA, MC-CDMA and OFDM systems.* Wiley, 2002.

[51] D. Marelli and M. Fu, "Subband methods for ofdm equalization," in *IEEE ICC*, vol. 4. IEEE, 2003, pp. 2350–2354.

[52] J. Faezah and K. Sabira, "Adaptive modulation for OFDM systems," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 1, no. 2, 2009.

[53] A. Czylwik, "Adaptive OFDM for wideband radio channels," in *Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, vol. 1, Nov 1996, pp. 713–718 vol.1.

[54] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 773–775, Feb 1995.

[55] R. Steele and W. T. Webb, "Variable rate QAM for data transmissions over rayleigh fading channels at proceedings of wireless'91." *IEEE*, pp. 1–14, 1991.

[56] L. Ye and A. Burr, "Adaptive modulation and code rate for turbo coded ofdm transmissions," in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, Apr 2007, pp. 2702–2706.

[57] K. Lu, S. Fu, Y. Qian, and H.-H. Chen, "Ser performance analysis for physical layer network coding over awgn channels," in *IEEE GLOBECOM*. IEEE, 2009, pp. 1–6.

[58] E. C. Y. Peh, Y.-C. Liang, and Y. L. Guan, "Power control for physical-layer network coding in fading environments," in *IEEE PIMRC*. IEEE, 2008, pp. 1–5.

[59] A. Y.-C. Peng, S. Yousefi, and I.-M. Kim, "On error analysis and distributed phase steering for wireless network coding over fading channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, p. 5639, 2009.

[60] M. Park, I. Choi, and I. Lee, "Exact ber analysis of physical layer network coding for two-way relay channels," in *IEEE VTC Spring*. IEEE, 2011, pp. 1–5.

[61] J. W. Craig, "A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations," in *IEEE MILCOM*. IEEE, 1991, pp. 571–575.

[62] M. Ju and I.-M. Kim, "Error performance analysis of bpsk modulation in physical-layer network-coded bidirectional relay networks," *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2770–2775, 2010.

[63] Y. Huang, Q. Song, S. Wang, and A. Jamalipour, "Symbol error rate analysis for m-qam modulated physical-layer network coding with phase errors," in *IEEE PIMRC*. IEEE, 2012, pp. 2003–2008.

[64] Y. Huang, Q. Song, S. Wang, and A. Jamalipour, "Phase-level synchronization for physical-layer network coding," *IEEE GLOBECOM*, pp. 4423–4428, 2012.

[65] Q. Yang and S. C. Liew, "Optimal decoding of convolutional-coded physical-layer network coding," in *IEEE WCNC*. IEEE, 2014, pp. 364–369.

[66] I. Committee *et al.*, "Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std*, vol. 802, p. 50, 1997.

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

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

[69] L. Lu, T. Wang, S. C. Liew, and S. Zhang, "Implementation of physical-layer network coding," *Physical Communication*, vol. 6, pp. 74–87, 2013.

[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 *Proceedings of the second workshop on Software radio implementation forum*. ACM, 2013, pp. 71–76.

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

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

[73] Q. Yang and S. C. Liew, "Asynchronous convolutional-coded physical-layer network coding," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1380–1395, 2015.

[74] S. Zhang, S.-C. Liew, and P. P. Lam, "On the synchronization of physical-layer network coding," in *IEEE ITW*. IEEE, 2006, pp. 404–408.

[75] D. To and J. Choi, "Convolutional codes in two-way relay networks with physical-layer network coding," vol. 9, no. 9, pp. 2724–2729, 2010.

[76] J. Sykora and A. Burr, "Layered design of hierarchical exclusive codebook and its capacity regions for HDF strategy in parametric wireless 2-WRC," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3241–3252, Sept 2011.

[77] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, 2007.

[78] Y. Yang and R. S. Blum, "Phase synchronization for coherent mimo radar: Algorithms and their analysis," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5538–5557, 2011.

[79] S. Sigg, R. M. El Masri, and M. Beigl, "Feedback-based closed-loop carrier synchronization: A sharp asymptotic bound, an asymptotically optimal approach, simulations, and experiments," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1605–1617, 2011.

[80] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.