國 立 交 通 大 學

電信工程學系碩士班

碩 士 論 文

在混合自動重傳機制中設計

位元交錯編碼調變及位元交錯編碼調變迭代解碼之

適應性多重傳輸位元映射規則

Adaptive Multiple Transmissions Mapping Design

for BICM and BICM-ID in HARQ

研 究 生：林哲群

指導教授：沈文和 教授

中 華 民 國 九 十 七 年 九 月

在混合自動重傳機制中設計
位元交錯編碼調變及位元交錯編碼調變迭代解碼之
適應性多重傳輸位元映射規則

Adaptive Multiple Transmissions Mapping Design for BICM and BICM-ID in HARQ

研 究 生 ： 林哲群　　Student : Che-Chun Lin

指導教授 ： 沈文和 博士　　Advisor : Dr. Wern-Ho Sheen

國 立 交 通 大 學

電 信 工 程 學 系 碩 士 班

碩 士 論 文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Communication Engineering

September 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年九月

在混合自動重傳機制中設計
位元交錯編碼調變及位元交錯編碼調變迭代解碼之
適應性多重傳輸位元映射規則

研究生：林哲群　　指導教授：沈文和 教授

國立交通大學

電信工程學系碩士班

摘要

在文獻中已經指出，相較於使用同一位元映射規則於同一封包的多次傳輸，改變多次傳輸的位元映射規則可以提供更高的吞吐量。根據位元交錯編碼調變的通道容量分析，重傳時的位元映射規則必須隨著所操作的通道訊雜比的改變而改變。因此我們提出了使用基因演算法找出對每個訊雜比而言最好的位元映射規則。而當位元交錯編碼調變迭代解碼用在混合自動重傳機制時，由外來資訊轉換圖的分析也可得知，對同一編碼方式而言，每個不同的訊雜比也應該有不同的位元映射規則。因此我們在無限長度封包與無限迭代次數的假設下，針對每個訊雜比設計出最佳的位元映射規則。而在實際有限長度封包的應用下，我們畫出了解映射器與解碼器轉換圖相對於平均值的變動，同時建議在解映射器與解碼器的轉換圖之間保留一些空間來設計有限長度封包的位元映射規則。

Adaptive Multiple Transmissions Mapping Design for BICM and BICM-ID in HARQ

Student: Che-Chun Lin **Advisor: Dr. Wern-Ho Sheen**

Department of Communication Engineering

National Chiao Tung University

Abstract

It has been shown in the literature that varying bit-to-symbol mappings for multiple transmissions of the same packet provides better throughput performance compared to using the same bit-to-symbol mapping throughout all the transmissions. The analysis of BICM capacity suggests that the retransmission mapping scheme should be adaptive to the operating SNR. Hence we propose using a genetic algorithm to find the optimal mapping for each SNR. When BICM with iterative decoding (BICM-ID) is applied in HARQ systems, the analysis of EXIT-chart also suggests that the suitable mapping for each SNR for the same code should be different. Therefore, we find optimal mapping for each SNR under the assumption of infinite block length and unlimited iteration number. In the real application of finite block length, we plot the variation of demapper and decoder transfer curve relative to the averaged one and suggest that margins between the demapper and the decoder transfer curve should be preserved for the finite block length design.

誌謝

本篇論文的完成首先要特別感謝的是我的指導教授 沈文和教授的細心指導，在每週定期的集會中不斷的提出問題，檢視每個想法中邏輯的缺失與漏洞，指出思考上的盲點，並提出有用的建議與指導，使本研究得以完善。

同時也要感謝的是昌龍學長，在繁忙的工作之餘，還抽空一起討論研究，提出核心的關鍵問題，指出有價值的研究方向，並在遇到困難的過程中建議解決方法，是本研究能夠前進很大的助力。此外也要感謝實驗室其他的學長與同學們，在課業上互相討論研究，也提供了研究上的一些想法與啟發。

當然也必須要感謝的是我的父母，提供一個舒適無慮的環境，使我能專心在課業上學習，不需擔憂生活，非常謝謝你們。

最後，要感謝的是上帝，那所有慈悲、愛與恩典的泉源。

民國九十七年九月

研究生林哲群謹誌於交通大學

Content

List of Tables

List of Figures

Chapter 1: Introduction

One of the main challenges in wireless communication is the fluctuation of signal amplitude caused by fading. Many efforts have been put to mitigate this adverse effect. Trellis coded modulation (TCM), originally proposed by Ungerboeck for bandwidth-efficient communication over the additive white Gaussian noise (AWGN), has shown some drawbacks when transmitting over fading environment. In the design of TCM, modulation and coding is combined as an entity to improve the performance. The design goal is to maximize the minimum free Euclidean distance, and therefore it is often optimized over AWGN channel. However, when transmitting over fading channels, its performance is significantly degraded since the diversity order is usually low. To combat the adverse effect of fading channel, symbol interleaver is added and parallel transitions in the trellis should be avoided. However, since the minimum number of distinct symbols between two codewords limits the diversity order, the constraint length should be increased. The increased constraint length further results in exponentially increased decoding complexity which is unacceptable.

In [2], Zehavi proposed an alternative approach called bit-interleaved coded modulation (BICM) to increase the diversity order to the minimum Hamming distance of the code. By placing a bit-wise interleaver at the encoder output, this allows large diversity order with moderate system complexity. In [4], Li and Ritcey showed that the performance of BICM can be further improved by iterative decoding between the demapper and the decoder, a scheme called bit-interleaved coded modulation with iterative decoding (BICM-ID). It has been shown in the literature that the design of the demapper is crucial to achieve a high coding gain over iteration. In [6], EXIT chart was proposed to describe the iterative decoding behavior through a decoding trajectory between the transfer curve of the demapper and decoder.

On the other hand, error control is also a main issue for data communication. Combined with the advantage of automatic-repeat-request (ARQ) mechanisms and forward-error-correction (FEC) schemes, HARQ is often adopted to achieve high reliability and high system throughput. In HARQ schemes, additional redundant parity bits are appended to the original message for both error correction and detection. When the presence of errors is detected, the receiver first tries to correct the erroneous bits. If the number of errors is beyond the designed error-correcting capability of the code, a retransmission request is send to the transmitter. The retransmitted packets can be exactly the same as the initial one or contain some extra redundant bits. When a new packet is received, the newly received packet can be decode alone or jointly decode with the previous ones.

In this thesis we are concerned about the case that retransmission carry identical bits and all the received packets are combined together for decoding. Furthermore, BICM and BICM-ID in conjunction with HARQ is considered.

It is known that the performance will be significantly improved by introducing packet combing. Chase combining [12] is the well known ML combing technique. It combines arbitrary number of coded packets into a single coded packet with lower code rate, thus improves the error-correcting capability of the code. However, Wengerter [15] showed that different bit to symbol mapping for retransmission can further improves the system performance. By simply swapping or taking logic inversion on the modulation bits to average out the unequal bit reliabilities, a method called constellation rearrangement, significant improvement has been observed. However, no optimality can be claimed on this method. In [16], an optimization criterion base on the BER upper bound has been proposed. The main deficiency of the mapping found by the minimization of the BER upper bound is that the upper bound is only tight at high SNR, the performance at low SNR can not be guaranteed. Murthy

[17] further suggested to change the criterion to maximize the sum of the magnitudes of the LLR of the bits forming the M-QAM symbols in different retransmissions. However, the maximization is made on the sum not on the individual bits LLR, no optimality can be guaranteed. Another criterion based on the augmented signal space after retransmission is to maximize the minimal accumulated (over transmission) squared Euclidean distance [18]. Gidlund [19] also aimed at increasing the Euclidean distance between signal points, thus applying the idea of set-partition in TCM to spread the signal points well in the augmented signal space. These designs ignore the relationship of the number of bit differences between nearest symbols; however, the number of bit differences is a crucial parameter for the design of BICM mappings. Hence is also not optimized for BICM systems.

Those mapping designs described above are all independent of SNR. However, an analysis based on the BICM capacity under multiple transmission [14] showed that one single mapping can not be optimal for the whole interested SNR range. It was showed that constellation rearrangement (CoRe) outperforms the mapping obtained by the minimization of BER upper bound (MBER) at low SNR. However, at high SNR, MBER exhibits better performance than CoRe. Since there are different operating SNR region at different code rate, this paper suggests that mapping should be adaptive considering the targeted spectral efficiency (code rate). Although adaptive mapping scheme has been proposed, mappings that are optimized for each SNR is still an open problem. Hence we aim to find these optimal mappings.

When iterative decoding is applied (BICM-ID), mapping design is especial crucial for obtaining large iterative decoding gain even for single transmission. Various mapping design methods have been proposed for single transmission. However, very few have addressed the issue of multiple transmissions mapping design. In [21], Roberson designed the retransmission mappings that optimize the uncoded zero prior

pairwise error probability and the uncoded ideal prior pair-wise error probability. Since the uncoded pair-wise error probability is independent of the underlying coding scheme, the design is not optimized for a particular code and may cause large performance degradation. By the analysis of the EXIT chart, the first intersection of the demapper transfer curve and the decoder transfer curve should be as high as possible. Since different mapping and coding have different transfer curve, their first intersection will be different. Therefore, a mapping that is good for a particular code may not be good for another one as well. Guided by the EXIT chart, mapping design should be dependent on the outer code. Furthermore, the dependency of the demapper transfer function on SNR also suggests that different mappings should be designed for the same code on different SNR. Hence we propose a method jointly considering the outer code and the operating SNR to design the retransmission mappings based on the EXIT chart.

Chapter 2: Overview of HARQ

A major concern in data communication is how to control transmission errors caused by the channel noise so that error-free data can be delivered to the user. There are two basic error-control schemes for data communication: automatic-repeat-request (ARQ) schemes and forward-error-correction (FEC) schemes.

In an ARQ error-control system, some parity bits are appended to the original information bits for error detection. When a codeword is received, the receiver computes its syndrome and determines if there is any erroneous bit. If the presence of errors is detected, the receiver discards the erroneously received codeword and requests a retransmission of the same codeword via a feed back channel.

In an FEC error-control system, an error-correcting code is used for combating transmission errors. When the receiver detects the presence of errors in the received codeword, it attempts to correct them. After the error correction has been performed, the decoded codeword is then delivered to the users. If the receiver failed to detect the presence of errors or the number of erroneous bits exceeds the error-correcting capability of the code, a decoding error is committed.

The advantages of ARQ scheme is simple and provides high system reliability. However, the throughput of ARQ system falls rapidly with increasing channel error rate. On the contrary, the FEC schemes maintain constant throughput (equal to the code rate) but is less reliable since the decoded message has to be delivered to the user regardless of whether it is correct or not. Thus to overcome the drawbacks in both ARQ and FEC schemes, a combination of these two called hybrid automatic-repeat-request (HARQ) scheme is proposed.

A HARQ system consists of FEC subsystem contained in an ARQ system. When

errors are detected by the receiver, the receiver tries to correct the erroneous bits. If the receiver fails to correct all of them, a retransmission request is delivered to the transmitter. The system throughput is increased by correcting the error patterns that occur most frequently. The system reliability is increased by requesting a retransmission rather than passing the unreliably decoded message to the user. As a result, a proper combination of FEC and ARQ provides higher throughput than FEC system and higher reliability than ARQ system.

There are three types of HARQ scheme, type I, type II and type III. In type I HARQ scheme the uncorrectable error packets are simply discarded and the receiver requests a retransmission of the same packet. Type I scheme is suitable for fairly static channel conditions since the error-correcting code can be designed specifically for this constant noise level. However, in applications with fluctuating channel conditions, type I has some drawbacks. When the bit error rate is small such that only small error correction capability is needed, the redundancy bits carried for correction of large bit errors represent a waste. When the channel is very noisy, the possibility of inadequate error-correcting capability will increase the frequency of retransmission and hence reduces the system throughput.

To overcome the drawbacks of the type I scheme, *incremental redundancy* HARQ (IR-HARQ) scheme is proposed. The basic idea is to transmit the additional redundancy bits only when they are needed. When the channel condition is good, only a small fraction of redundancy bits are transmitted to correct small bit errors. When a retransmission request is delivered to the transmitter, additional redundancy bits are transmitted to the receiver. The receiver than combine the newly received packet and the previous ones to form a more powerful code with lower effective code rate. These schemes offer higher throughput efficiency since the error correcting code redundancy is adapted to the varying channel conditions. Depending on whether each retransmitted

packets are self decodable or not, IR-HARQ can be further categorized into two classes: type II and type III schemes. Type II scheme is also referred to as *full incremental redundancy* scheme where only incremental redundancy bits to the initial transmission are retransmitted. Type III scheme is also referred to as *partial incremental redundancy* scheme where partly identical bits and partly incremental bits to the initial transmission are retransmitted. The main drawback of type II scheme is that the decoder has to rely on both the previous received packets as well as the newly received one to decode. In situation where a packet may be lost, it is not possible to use previous packet and recover the original message. Thus it is desirable to have a scheme where incremental coded bits are self decodable.

Another advantage that makes type II and type III HARQ scheme more attractable than type I is that instead of simply discard the erroneous packets, type II and type III scheme combine the previous received packets and jointly decode them. Although damaged by the channel noise, these packets still carry useful information that is beneficial for decoding. Therefore, decoding with packet combining often performs better than decoding without packet combining.

Chapter 3: System Model

3.1 Transmitter

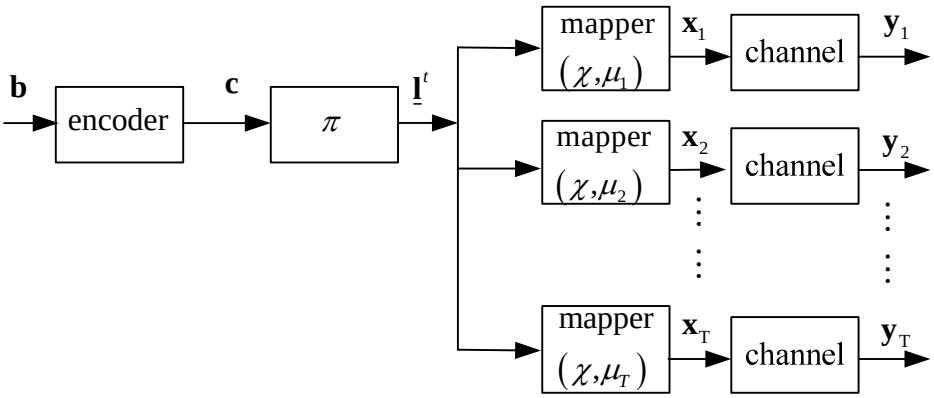

Fig 3.1 BICM in HARQ transmitter model

Consider the system model described by Fig3.1. A vector of binary bits $\mathbf{b}=[b_1,b_2,...,b_{N_I}]$ with length N_I is encoded to binary codewords $\mathbf{c}=[c_1,c_2,...,c_{N_c}]$ with length N_c. The encoded codewords are then fed into bit interleaver π. The interleaved codewords are denoted as $\underline{\mathbf{l}}^t=[\mathbf{l}_1^t,\mathbf{l}_2^t,...,\mathbf{l}_{N_L}^t]$, where $\mathbf{l}_i^t=[l_{i,1}^t,l_{i,2}^t,...,l_{i,n_s}^t]$ $,i=1,....,N_L$ is a group of n_s bits that will be mapped to a complex symbol. The sequence of 2^{n_s} -ary complex symbols are denoted as $\mathbf{x_i}=[x_{i,1},x_{i,2},\ldots,x_{i,N_L}]$. The first subscript indicates the i-th transmission $,i=1,....,T$, and T is the maximum allowed transmission number. Since different bit to symbol mappings can be adopted while retransmission, we denote the i-th transmission mapper as μ_i and $x_{i,k}=\mu_1\left(\mathbf{l}_k^t\right)\in\chi$.

In our notation convention, we write random variables using upper case letters and their realizations by the corresponding lower case letters. Bold case letters represent vectors and underscore is used to represent a sequence of vectors. The same terminology will be used throughout this thesis.

3.2 Channel

The signal $\mathbf{x_i}$ is send over the channel and $\mathbf{y_i} = \mathbf{h_i}\mathbf{x_i^T} + \mathbf{n_i}$ is the received one, where $\mathbf{n}_i = [n_{i,1}, n_{i,2}, ..., n_{i,N_L}]$ is AWGN and $\mathbf{h_i} = [h_{i,1}, h_{i,2}, ..., h_{i,N_L}]$ is the channel fading gain. Each elements in $\mathbf{n}_i$ is an iid complex Gaussian Random variable with zero mean and variance $\frac{N_0}{2}$ in real and imaginary part. When the channel is modeled as a frequency non-selective fast fading channel, $h_{i,k}$ is an iid complex Gaussian random variable with zero mean and variance $\frac{1}{2}$ in real and imaginary part. When the channel is modeled as an AWGN channel, $h_{i,k}$ is 1 for all i and k. Finally, we assume that SNR is the same for each retransmission.

3.3 Receiver

Here we denote $\underline{\mathbf{L}}(u,v) = \left(\mathbf{L}_1(u,v), \mathbf{L}_2(u,v), ..., \mathbf{L}_{N_L}(u,v)\right)$ as a sequence of log likelihood ratio (LLR), where $u \in \{D, A, E\}, v \in \{\phi, \varphi\}$. D stands for Detection, A for a priori, and E for extrinsic. ϕ stands for the demapper and φ for the decoder. $\mathbf{L}_k(u,v) = \{L_{k,1}(u,v), L_{k,2}(u,v), ..., L_{k,n_s}(u,v)\}$ is a sequence of LLRs belong to the k-th symbol $\{x_{1,k}, x_{2,k}, ..., x_{T,k}\}$ (Since the same coded bits are transmitted in each retransmission, each k-th symbol carries the same coded bits.)

3.3.1 BICM Receiver Model

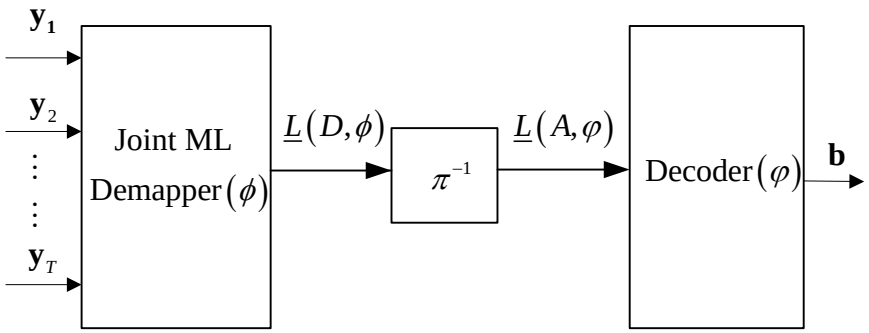

Fig 3.3.1 BICM in HARQ receiver model

As shown in Fig 3.3.1, the joint ML demapper ϕ receives a sequence of symbols $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_\mathbf{T}$ in all T transmissions and jointly detects them to compute the coded bit LLRs $\underline{\mathbf{L}}(D,\phi)$ for the decoder. After passing through the deinterleaver π^{-1} to restore the original bit order, the deinterleaved LLR $\underline{\mathbf{L}}(A,\phi)$ is fed into the decoder. The original information bits $\mathbf{b}$ are then decoded by the decoder.

3.3.2 BICM-ID Receiver Model

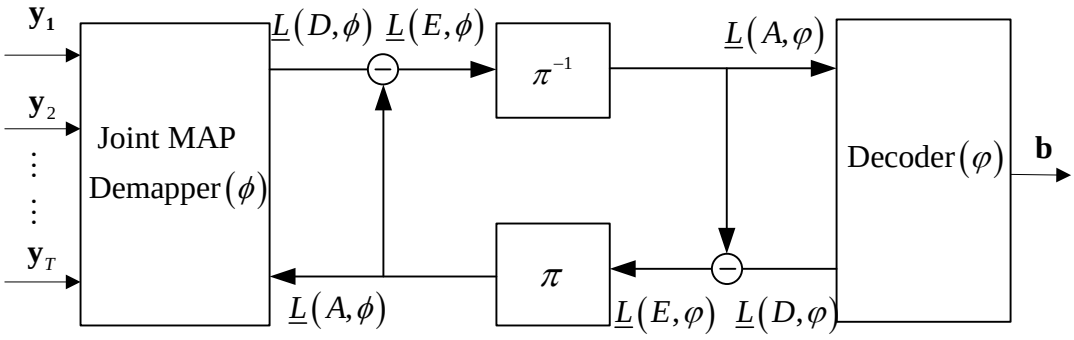

Fig 3.3.2 BICM-ID in HARQ receiver model

When iterative decoding is applied, the demapper not only receives the T transmission symbols $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_\mathbf{T}$ but also the a priori information $\underline{\mathbf{L}}(A,\phi)$ generated by the decoder. The demapper then apply MAP detection algorithm to

compute the coded bit LLRs $\underline{\mathbf{L}}(D,\phi)$. Before passing to the decoder, the demapper subtracts the a prior information $\underline{\mathbf{L}}(A,\phi)$ to produce the extrinsic information $\underline{\mathbf{L}}(E,\phi)$. The decoder applies similar pricinple to generate the extrinsic information $\underline{\mathbf{L}}(E,\varphi)$ and feed back to the decoder. Thus the signal is iteratively decoded by mutually exchanging soft information between inner demapper and outer decoder. This iteration process continues until a prescribed number of iteration is reached.

3.3.3 Joint MAP Demapper

The demapper computes the a posteriori probability for coded bits. Since the modulation is memoryless, only the k-th symbol is concerned when detecting the i-th bit in the k-th symbol. Therefore, for simplicity, we drop the subindex k in the k-th label $\mathbf{l}_k^t$ and define $\mathbf{l}^t = \left(l_1^t, l_2^t, ..., l_{n_s}^t\right) \triangleq \mathbf{l}_k^t = \left(l_{k,1}^t, l_{k,2}^t, ..., l_{k,n_s}^t\right) \in \Lambda$. For the i-th bit LLR in the k-th symbol, define $L_i(u,v) \triangleq L_{k,i}(u,v)$. Similarly, we define $x_i \triangleq x_{i,k}, y_i \triangleq y_{i,k}, h_i \triangleq h_{i,k}, n_i \triangleq n_{i,k}$, and therefore $y_i = h_i x_i + n_i$. Perfect channel state information is assumed at the receiver, thus the LLR for the i-th bit is calculated as

$$L_i(D,\phi) = \log \frac{p\left(l_i^t = 1 \mid y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right)}{p\left(l_i^t = 0 \mid y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right)}$$

$$= \log \frac{p\left(l_i^t = 1\right)}{p\left(l_i^t = 0\right)} + \log \frac{p\left(h_1, h_2, ..., h_T \mid l_i^t = 1\right)}{p\left(h_1, h_2, ..., h_T \mid l_i^t = 0\right)} + \log \frac{p\left(y_1, y_2, ..., y_T \mid l_i^t = 1, h_1, h_2, ..., h_T\right)}{p\left(y_1, y_2, ..., y_T | l_i^t = 0, h_1, h_2, ..., h_T\right)}$$

$$= \log \frac{p\left(l_i^t = 1\right)}{p\left(l_i^t = 0\right)} + \log \frac{p\left(h_1, h_2, ..., h_T\right)}{p\left(h_1, h_2, ..., h_T\right)} + \log \frac{p\left(y_1, y_2, ..., y_T \mid l_i^t = 1, h_1, h_2, ..., h_T\right)}{p\left(y_1, y_2, ..., y_T | l_i^t = 0, h_1, h_2, ..., h_T\right)}$$

$$= L_i(A,\phi) + \log \frac{\sum_{\mathbf{l}^t \in \Lambda_i^1} p\left(y_1, y_2, ..., y_T, \mathbf{l}^t \mid l_i^t = 1, h_1, h_2, ..., h_T\right)}{\sum_{\mathbf{l}^t \in \Lambda_i^0} p\left(y_1, y_2, ..., y_T, \mathbf{l}^t \mid l_i^t = 0, h_1, h_2, ..., h_T\right)}$$

$$= L_i(A,\phi) + \log \frac{\sum_{\mathbf{l}^t \in \Lambda_i^1} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - h_k \mu_k(\mathbf{l}^t) \|^2}{2\sigma^2} \right) \prod_{\substack{j=1 \\ j \neq i}}^{n_s} \frac{p(l_j^t = c_j)}{p(l_j^t = 0)}}{\sum_{\mathbf{l}^t \in \Lambda_i^0} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - h_k \mu_k(\mathbf{l}^t) \|^2}{2\sigma^2} \right) \prod_{\substack{j=1 \\ j \neq i}}^{n_s} \frac{p(l_j^t = d_j)}{p(l_j^t = 0)}}$$

$$= L_i(A,\phi) + \log \frac{\sum_{\mathbf{l}^t \in \Lambda_i^1} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - h_k \mu_k(\mathbf{l}^t) \|^2}{2\sigma^2} \right) \exp\left(\sum_{\substack{j=1 \\ j \neq i}}^{n_s} c_j \cdot L_j(A,\phi) \right)}{\sum_{\mathbf{l}^t \in \Lambda_i^0} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - h_k \mu_k(\mathbf{l}^t) \|^2}{2\sigma^2} \right) \exp\left(\sum_{\substack{j=1 \\ j \neq i}}^{n_s} d_j \cdot L_j(A,\phi) \right)}$$

Chapter 4: Symbol Mapping Diversity in HARQ

4.1 Constellations under joint detection

When the same mappings are used during further retransmissions, the performance improvement comes only from SNR gain. If the same signal is transmitted T times, 3T dB gain will obtained at the receiver. However, when the retransmission mappings are changed, the receiver will have the potential to get not only 3T dB gain but also additional gain by the benefit of symbol mapping diversity. This can be well explained by the enlarged signal space under joint detection. For the simplicity of exposition, consider 4-PAM transmission in the following:

Fig 4.1.1 constellations with Gray mapping for the first transmission

Fig 4.1.2 constellations with the same mapping as the first transmission for the second transmission

In our first example (Fig 4.1.1 and Fig 4.1.2), Gray mapping is transmitted at the first time and the same mapping is adopted at the second transmission. Since the receiver receives two signal and jointly detect them, the signal space under joint detection is enlarged to two times the dimension of single transmission. As shown in Fig 4.1.3, the signal space is now a two dimensional one instead of just one dimension.

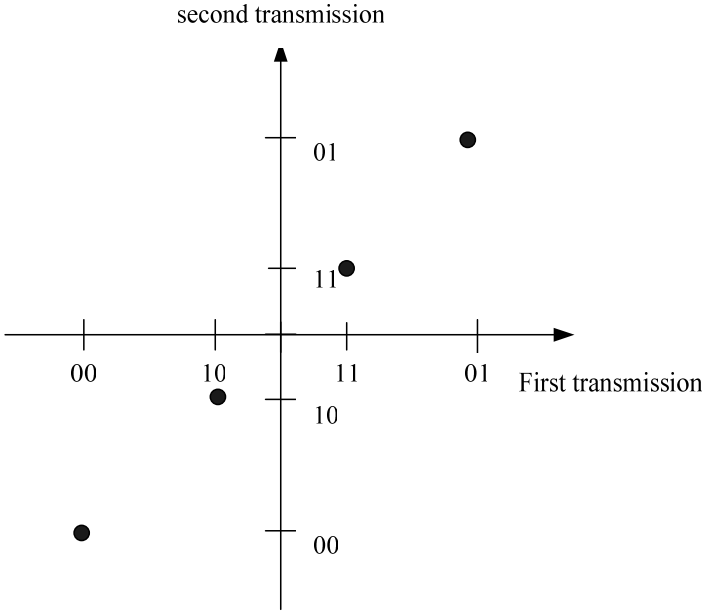

Fig 4.1.3 constellation under joint detection

The abscissa in Fig 4.1.3 represents the constellation for the first transmission and the ordinate represents the one for the second transmission. Let us denote the minimum Euclidean distance between the constellation points in single transmission as d_1. Then the minimum Euclidean distance after the second transmission will be $d_2 = \sqrt{d_1^2 + d_1^2} = \sqrt{2}d_1$ Therefore the SNR is doubled after retransmission. However, despite of the enlarged signal space dimension under joint detection, the constellation are still aligned in one dimension. This implies the inefficiency of utilizing the same mapping while retransmission.

Consider the case of re-mapping the second transmission as shown in Fig 4.1.4.

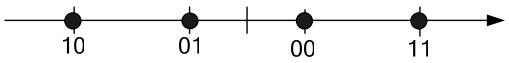

Fig 4.1.4 re-mapping for the second transmission

Thus the second mapping is not Gray anymore. The constellation after re-mapping is shown in Fig 4.1.5.

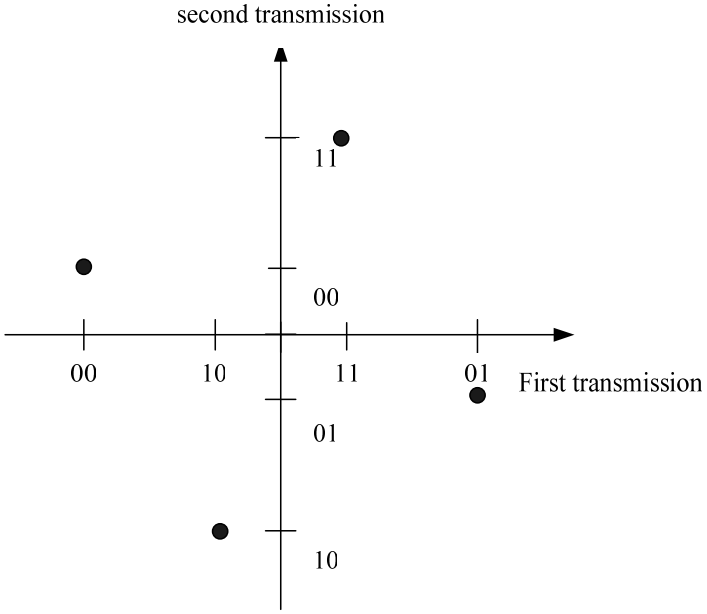

Fig 4.1.5 constellation after re-mapping

Observe that by the simple re-mapping technique, the constellation is now augmented to two dimensional signal space, and the minimum Euclidean distance is enlarged to $d_2 = \sqrt{(2d_1)^2 + d_1^2} = \sqrt{5}d_1$. Comparing to $\sqrt{2}d_1$, the utilization of all the available dimensions provides larger minimum Euclidean distance than just doubling the signal power. This comes at no additional power or bandwidth cost. For further transmissions larger than two, similar arguments apply.

The possibility of larger Euclidean distance between signal points is similar to the concept of binary coding. For a coding scheme with code rate $R = \frac{K}{N}$, N coded bits are used to transmit the original K information bits and $N \geq K$. For length N coded bits, there are 2^N codewords available, while only 2^K codewords are needed to represent the original message. Thus it is possible to assign 2^K codewords appropriately such that they are spaced far apart from each other to obtain larger Hamming distance. Similarly, for a 2^{n_s}-ary modulation with T transmissions, there are 2^{Tn_s} constellation points available, while the source signal only requires 2^{n_s} constellation points (Since type I HARQ is considered, the same bits are retransmitted.

Hence only 2^{n_s} constellation points are required despite of the number of transmissions). Larger Euclidean distance can be provided by the extra $2^{Tn_s} - 2^{n_s}$ constellation points available. Therefore, retransmitting the same bits can be thought of as a form of coding in symbol domain. Analogous to binary coding, retransmission without mapping change can be considered as a form of repetition code which is not efficient in terms of enlarging the Hamming distance. Hence appropriate design of retransmission mapping is essential.

4.2 Coded Modulation Capacity

The benefits of mapping change can also be evaluated analytically from the CM (coded modulation) capacity. A proper change of retransmission mappings will boost the CM capacity. In our evaluation, we normalize the CM capacity with respect to T to take into account the increased number of channel uses after T transmission. For AWGN channel, the coded modulation capacity under uniform input constraint is evaluated as

$$
\begin{aligned}
C_{CM\_HARQ} &= \frac{1}{T} I\left(X_{1,} X_2, ..., X_T; Y_1, Y_2, ..., Y_T\right) \\
&= \frac{1}{T} I\left(\mathbf{L}^t, \mathbf{L}^t, ..., \mathbf{L}^t; Y_1, Y_2, ..., Y_T\right) \\
&= \frac{1}{T} I\left(\mathbf{L}^t; Y_1, Y_2, ..., Y_T\right) \\
&= \frac{1}{T} \{H\left(\mathbf{L}^t\right) - H\left(\mathbf{L}^t \mid Y_1, Y_2, ..., Y_T\right)\} \\
&= \frac{1}{T} \{\log_2 2^{n_s} - \sum_{\mathbf{l}^t = \lambda_j \in \Lambda} \int_{y_1=-\infty}^{\infty} ... \int_{y_T=-\infty}^{\infty} p\left(\mathbf{l}^t = \lambda_j, y_1, y_2, ..., y_T\right) \log_2 \frac{p\left(y_1, y_2, ..., y_T\right)}{p\left(\mathbf{l}^t = \lambda_j, y_1, y_2, ..., y_T\right)} dy_1 ... dy_T \}
\end{aligned}
\tag{4.1}
$$

where

$$
\begin{aligned}
& p\left(\mathbf{I}^t = \lambda_j, y_1, y_2, ..., y_T\right) \\
&= p\left(y_1, y_2, ..., y_T \mid \mathbf{I}^t = \lambda_j\right) p\left(\mathbf{I}^t = \lambda_j\right) \\
&= p\left(y_1 \mid \mathbf{I}^t = \lambda_j\right) ... p\left(y_T \mid \mathbf{I}^t = \lambda_j\right) p\left(\mathbf{I}^t = \lambda_j\right) \\
&= p\left(y_1 \mid \mathbf{I}^t = \lambda_j\right) ... p\left(y_T \mid \mathbf{I}^t = \lambda_j\right) \frac{1}{2^{n_s}} \\
&= p\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_j\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{I}^t = \lambda_j\right)\right) \frac{1}{2^{n_s}}
\end{aligned} \tag{4.2}
$$

$$
\begin{aligned}
& p\left(y_1, y_2, ..., y_T\right) \\
&= \sum_{\mathbf{I}^t = \lambda_i \in \Lambda} p\left(y_1, ..., y_T \mid \mathbf{I}^t = \lambda_i\right) p\left(\mathbf{I}^t = \lambda_i\right) \\
&= \sum_{\mathbf{I}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mathbf{I}^t = \lambda_i\right) p\left(y_2 \mid \mathbf{I}^t = \lambda_i\right) ... p\left(y_T \mid \mathbf{I}^t = \lambda_i\right) p\left(\mathbf{I}^t = \lambda_i\right) \\
&= \sum_{\mathbf{I}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mathbf{I}^t = \lambda_i\right) p\left(y_2 \mid \mathbf{I}^t = \lambda_i\right) ... p\left(y_T \mid \mathbf{I}^t = \lambda_i\right) \frac{1}{2^{n_s}} \\
&= \sum_{\mathbf{I}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_i\right)\right) p\left(y_2 \mid \mu_2\left(\mathbf{I}^t = \lambda_i\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{I}^t = \lambda_i\right)\right) \frac{1}{2^{n_s}}
\end{aligned} \tag{4.3}
$$

With (4.2) and (4.3), (4.1) becomes

$$
\begin{aligned}
& C_{CM\_HARQ} \\
&= \frac{1}{T}\{n_s - \sum_{\mathbf{I}^t = \lambda_j \in \Lambda} \int_{y_1=-\infty}^{\infty} ... \int_{y_T=-\infty}^{\infty} p\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_j\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{I}^t = \lambda_j\right)\right) \frac{1}{2^{n_s}} \cdot \\
& \log_2 \frac{\sum_{\mathbf{I}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_i\right)\right) ... p\left(y_T \mid \mu_1\left(\mathbf{I}^t = \lambda_i\right)\right)}{p\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_j\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{I}^t = \lambda_j\right)\right)} dy_1 ... dy_T\} \\
&= \frac{1}{T}\{n_s - \sum_{\mathbf{I}^t = \lambda_j \in \Lambda} \int_{y_1=-\infty}^{\infty} ... \int_{y_T=-\infty}^{\infty} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - \mu_k\left(\mathbf{I}^t = \lambda_j\right) \|^2}{2\sigma^2}\right) \frac{1}{2^{n_s}} \cdot \\
& \log_2 \frac{\sum_{\mathbf{I}^t = \lambda_i \in \Lambda} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - \mu_k\left(\mathbf{I}^t = \lambda_i\right) \|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T} \frac{\| y_k - \mu_k\left(\mathbf{I}^t = \lambda_j\right) \|^2}{2\sigma^2}\right)} dy_1 ... dy_T\}
\end{aligned} \tag{4.4}
$$

Observe that the first term $\frac{n_s}{T}$ in the coded modulation capacity is constant

regardless of the mapping scheme, hence it remains to minimize the term

$$A(\mu_1,...,\mu_T)=\sum_{\mathbf{l}^t=\lambda_j\in\Lambda}\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_j)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\cdot$$

$$\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_j)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

for the maximization of CM capacity. Therefore, it is only required to compare the term $A(\mu_1,\mu_2,...,\mu_T)$ to determine which mapping scheme has larger CM capacity.

Consider the two different mapping schemes μ_k and μ_k' , $k=1,...,T$. Suppose that the relationship of the two mapping scheme can be expressed as $\mu_k'(\lambda_i)=\mu_k(\lambda_i)+\alpha_{k,i}$, $i=1,...,M$ and $k=1,...,T$, where $\alpha_{k,i}$ is a complex number and M-ary modulation is considered. Then the for the first mapping scheme $\mu_k, k=1,...,T$

$A(\mu_1,\mu_2,...,\mu_T)$

$$=\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_1)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_1)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

$$+\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_2)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_2)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

$$\vdots$$

$$+\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_M)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_M)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

We further define

$$a_j(\mu_1,\mu_2,...,\mu_T)=$$

$$\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\lambda_j)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\lambda_j)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

Consider, for example, the first term in $A(\mu_1,\mu_2,...,\mu_T)$,

$$a_1(\mu_1,\mu_2,...,\mu_T)=$$

$$\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_1)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu_k(\mathbf{l}^t=\lambda_1)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

For the second mapping scheme $\mu'_k, k=1,...,T$, the first term in $A(\mu'_1,\mu'_2,...,\mu'_T)$ is

$$a_1(\mu'_1,\mu'_2,...,\mu'_T)$$

$$=\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu'_k(\lambda_1)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu'_k(\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-\mu'_k(\lambda_1)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

$$=\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-(\mu_k(\lambda_1)+\alpha_{k,1})\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-(\mu_k(\lambda_i)+\alpha_{k,i})\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-(\mu_k(\lambda_1)+\alpha_{k,1})\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

$$=\int_{y_1=-\infty}^{\infty}...\int_{y_T=-\infty}^{\infty}\exp\left(-\sum_{k=1}^{T}\frac{\|(y_k-\alpha_{k,1})-\mu_k(\lambda_1)\|^2}{2\sigma^2}\right)\frac{1}{2^{n_s}}\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|(y_k-\alpha_{k,i})-\mu_k(\lambda_i)\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T}\frac{\|(y_k-\alpha_{k,1})-\mu_k(\lambda_1)\|^2}{2\sigma^2}\right)}dy_1...dy_T$$

Let $y'_k = y_k - \alpha_{k,1}, k = 1,...,T$, then

$$a_1\left(\mu'_1, \mu'_2, ..., \mu'_T\right)$$

$$= \int_{y'_1=-\infty-\alpha_{k,1}}^{\infty-\alpha_{k,1}} ... \int_{y'_T=-\infty-\alpha_{k,1}}^{\infty-\alpha_{k,1}} \exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right) \frac{1}{2^{n_s}} \cdot$$

$$\log_2 \frac{\exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right) + \sum_{\mathbf{l}^t=\lambda_i\in\Lambda, i\neq 1} \exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \left(\mu_k\left(\lambda_i\right) + a_{k,i}\right) + a_{k,1}\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right)} dy'_1 ... dy'_T$$

$$= \int_{y'_1=-\infty}^{\infty} ... \int_{y'_T=-\infty}^{\infty} \exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right) \frac{1}{2^{n_s}} \cdot$$

$$\log_2 \frac{\exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right) + \sum_{\mathbf{l}^t=\lambda_i\in\Lambda, i\neq 1} \exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \left(\mu_k\left(\lambda_i\right) + a_{k,i}\right) + a_{k,1}\|^2}{2\sigma^2}\right)}{\exp\left(-\sum_{k=1}^{T} \frac{\| y'_k - \mu_k\left(\lambda_1\right)\|^2}{2\sigma^2}\right)} dy'_1 ... dy'_T$$

Compare $a_1\left(\mu_1, \mu_2, ..., \mu_T\right)$ with $a_1\left(\mu'_1, \mu'_2, ..., \mu'_T\right)$, the only difference is the term $\sum_{\mathbf{l}^t=\lambda_i\in\Lambda} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - \mu_k\left(\mathbf{l}^t=\lambda_i\right)\|^2}{2\sigma^2}\right)$ .The same argument applies to other $a_i\left(\mu_1, ..., \mu_T\right), i = 2,...,M$ as well. Hence to maximize the coded modulation capacity, the term $\sum_{\mathbf{l}^t=\lambda_i\in\Lambda} \exp\left(-\sum_{k=1}^{T} \frac{\| y_k - \mu_k\left(\mathbf{l}^t=\lambda_i\right)\|^2}{2\sigma^2}\right)$ should be minimized. Observing the exponent term in this expression, the retransmission mapping should be designed so that the combined Euclidean distance between signal points will be maximized. Intuitively, two nearby signal points in the first transmission mapping should be

assigned to signal points that are far apart from each other in the second transmission mapping. Hence mappings designed to maximize the minimum Euclidean distance between constellation points often achieves high CM capacity.

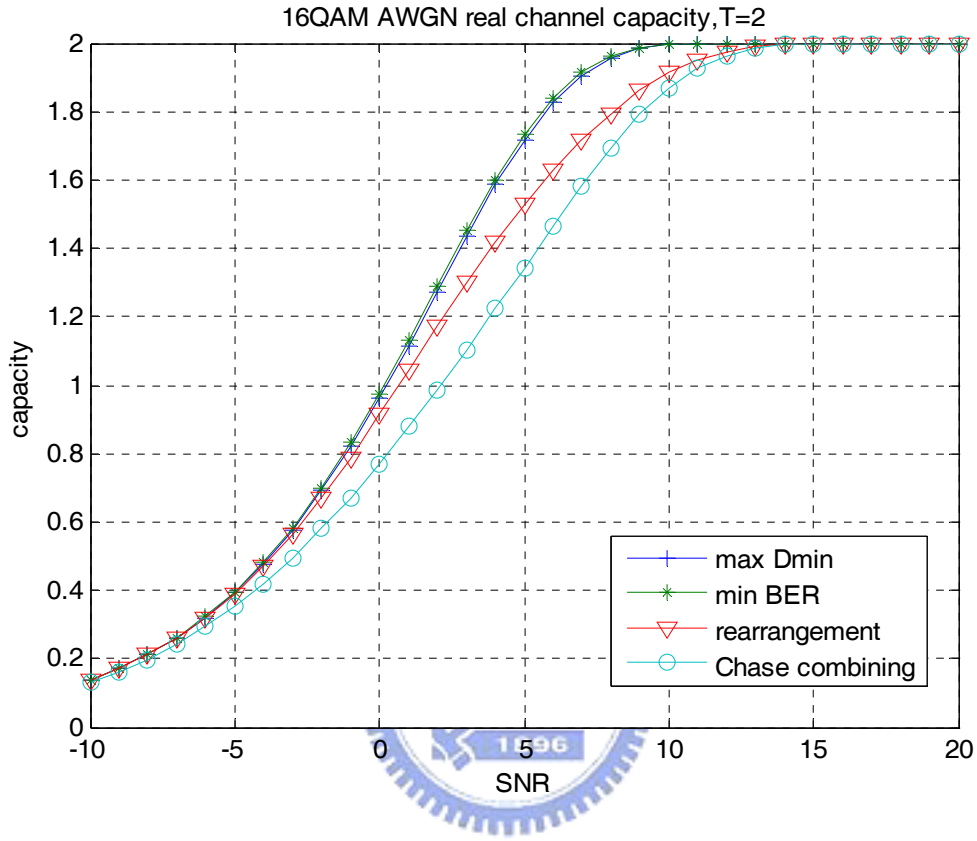

Fig 4.2.1 CM capacity of different mapping schemes

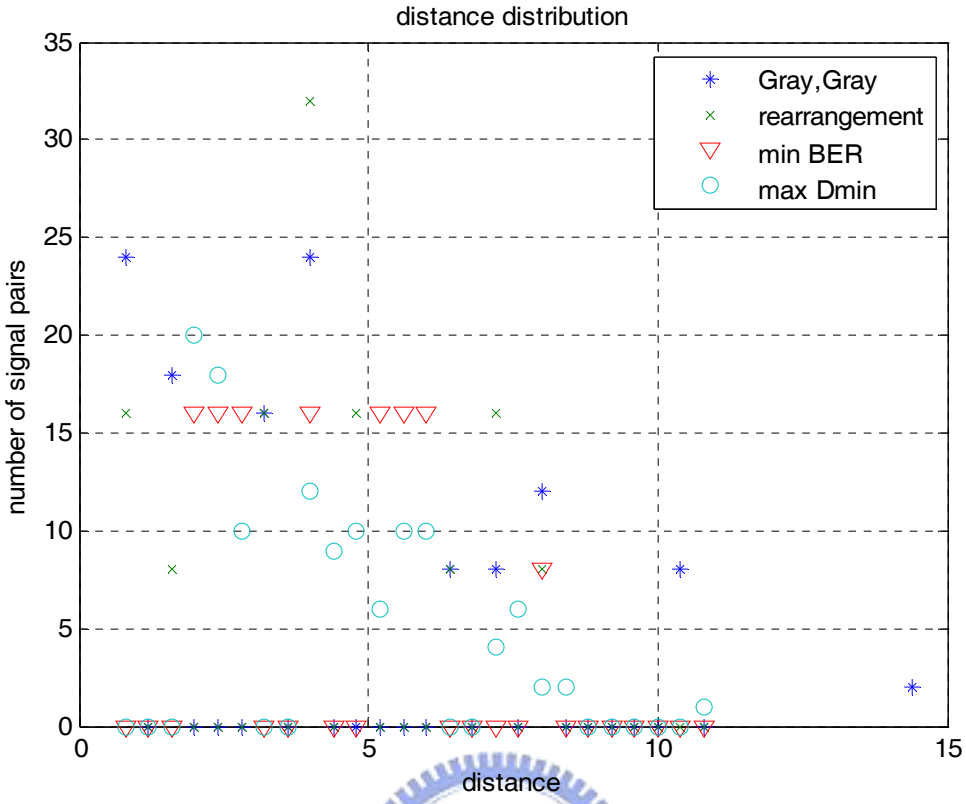

Fig 4.2.2 distance distribution of different mapping schemes

Fig 4.2.1 compared the CM capacity for 16QAM in AWGN channels for some different mapping schemes in the literature and Fig 4.2.2 showed their corresponding distance distribution between signal pairs. Mapping designed for maximizing the minimum Euclidean distance (MDMIN) has high CM capacity since the nearest signal points have been pulled far apart. Similar behavior has been presented in the mapping designed for minimizing the BER upper bound (MBER) since the minimum Euclidean distance dominates the BER upper bound. The distance distribution has confirmed that they have largest minimum Euclidean distance. For constellation rearrangement, although the minimum Euclidean distance has not been enlarged compared to Chase combining, the number of signal pairs that have smallest distance have decreased. Therefore its CM capacity is larger than Chase combining.

The above argument applied to frequency none-selective fast fading channel as well. We assume that the receiver have perfect channel state information, the capacity

derivation is quite similar to AWGN channel.

$$C_{CM\_HARQ} = \frac{1}{T} I\left(X_{1,} X_2, ..., X_T; Y_1, Y_2, ..., Y_T \mid H_1, H_2, ..., H_T\right)$$

$$= \frac{1}{T} I\left(\mathbf{L}^t; Y_1, Y_2, ..., Y_T \mid H_1, H_2, ..., H_T\right)$$

$$= \frac{1}{T}\{H\left(\mathbf{L}^t \mid H_1, H_2, ..., H_T\right) - H\left(\mathbf{L}^t \mid Y_1, Y_2, ..., Y_T, H_1, H_2, ..., H_T\right)\}$$

$$= \frac{1}{T}\{H\left(\mathbf{L}^t\right) - H\left(\mathbf{L}^t \mid Y_1, Y_2, ..., Y_T, H_1, H_2, ..., H_T\right)\}$$

$$= \frac{1}{T}\{n_s - \sum_{\mathbf{l}^t = \lambda_j \in \Lambda} \int_{h_1=-\infty}^{\infty} ... \int_{h_t=-\infty}^{\infty} \int_{y_1=-\infty}^{\infty} ... \int_{y_T=-\infty}^{\infty} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right), h_1\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right), h_T\right) p\left(h_1\right) ... p\left(h_T\right) \cdot$$

$$\frac{1}{2^{n_s}} \log_2 \frac{\sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_i\right), h_1\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_i\right), h_T\right)}{p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right), h_1\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right), h_T\right)} dy_1 ... dy_T dh_1 ... dh_T\} \quad (4.5)$$

4.3 Bit-Interleaved Coded Modulation Capacity

The above coded modulation capacity analysis applies for the ideal case that coding and modulation are combined together and the channel code is powerful enough. This analysis shows the performance limit when joint coding and modulation scheme is applied. However, when BICM scheme is adopted, which separates coding and modulation, the capacity will be different from the CM capacity. Similar to CM capacity, the BICM capacity is also affected by the choice of retransmission mappings. The derivation of BICM capacity with multiple transmissions is a direct extension of the BICM capacity with single transmission [3].

Fig 4.3.1 shows the equivalent parallel channel model for BICM under the assumption of ideal interleaving. S is the random variable whose outcome determines the switch position and is i.i.d. uniformly distributed over $\{1, 2, ..., n_s\}$.

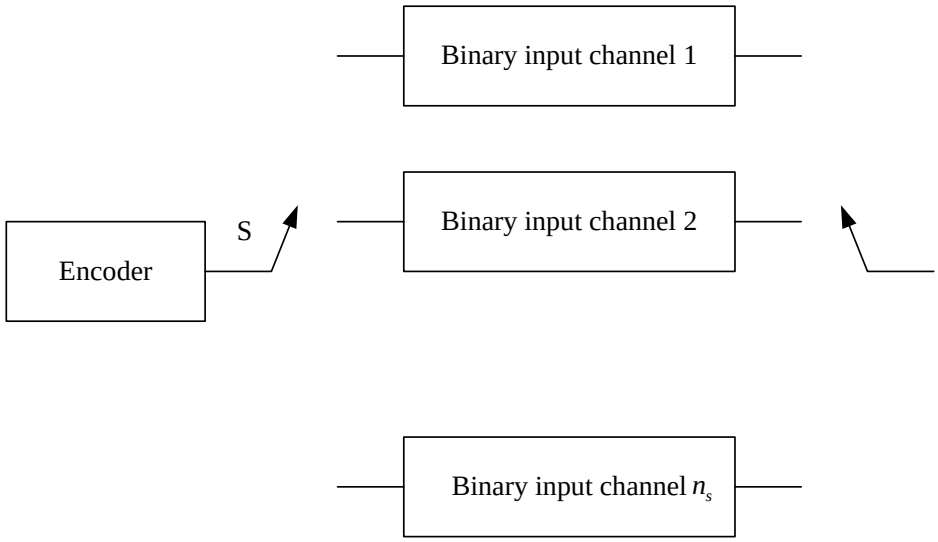

Fig 4.3.1 Equivalent parallel channel model for BICM in the case of ideal interleaving

The BICM capacity with perfect CSI is given by (normalized with respect to T)

$$C_{BICM\_HARQ} = \frac{1}{T} n_s I\left(c; Y_1, Y_2, ..., Y_T \mid H_1, H_2, ..., H_T, S\right)$$

$$= \frac{1}{T} n_s \left(\frac{1}{n_s} \sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T \mid H_1, H_2, ..., H_T, s = k\right) \right)$$

$$= \frac{1}{T} \{ \sum_{k=1}^{n_s} \left(H\left(L_k^t \mid H_1, H_2, ..., H_T\right) - H\left(L_k^t \mid Y_1, Y_2, ..., Y_T, H_1, H_2, ..., H_T\right) \right) \}$$

$$= \frac{1}{T} \sum_{k=1}^{n_s} \left(H\left(L_k^t\right) - H\left(L_k^t \mid Y_1, Y_2, ..., Y_T, H_1, H_2, ..., H_T\right) \right)$$

$$= \frac{1}{T} \sum_{k=1}^{n_s} \left(1 - \sum_{l_k^t = b \in \{0,1\}} \int_{y_1, ..., y_T} \int_{h_1, ..., h_T} p\left(l_k^t = b, y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right) \log_2 \frac{p\left(y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right)}{p\left(l_k^t = b, y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right)} dy_1 ... dy_T \right)$$

(4.6)

where

$$p\left(l_k^t = b, y_1, y_2, ..., y_T, h_1, ..., h_T\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda} p\left(y_1, y_2, ..., y_T, h_1, ..., h_T, \mathbf{l}^t = \lambda_j, l_k^t = b\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} p\left(y_1, y_2, ..., y_T, h_1, ..., h_T, \mathbf{l}^t = \lambda_j\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} p\left(y_1, y_2, ..., y_T \mid h_1, h_2, ..., h_T, \mathbf{l}^t = \lambda_j\right) p\left(h_1, h_2, ..., h_T \mid \mathbf{l}^t = \lambda_j\right) p\left(\mathbf{l}^t = \lambda_j\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} p\left(y_1, y_2, ..., y_T \mid h_1, h_2, ..., h_T, \mathbf{l}^t = \lambda_j\right) p\left(h_1, h_2, ..., h_T\right) p\left(\mathbf{l}^t = \lambda_j\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} p\left(y_1 \mid h_1, \mathbf{l}^t = \lambda_j\right) p\left(y_2 \mid h_2, \mathbf{l}^t = \lambda_j\right) ... p\left(y_T \mid h_T, \mathbf{l}^t = \lambda_j\right) p\left(h_1, h_2, ..., h_T\right) \frac{1}{2^{n_s}}$$

$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} p\left(y_1 \mid h_1, \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) p\left(y_2 \mid h_2, \mu_2\left(\mathbf{l}^t = \lambda_j\right)\right) ... p\left(y_T \mid h_T, \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) p\left(h_1, h_2, ..., h_T\right) \frac{1}{2^{n_s}} \quad (4.7)$$

$$p\left(y_1, y_2, ..., y_T, h_1, h_2, ..., h_T\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1, ..., y_T, h_1, h_2, ..., h_T, \mathbf{l}^t = \lambda_i\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1, ..., y_T \mid h_1, h_2, ..., h_T, \mathbf{l}^t = \lambda_i\right) p\left(h_1, h_2, ..., h_T \mid \mathbf{l}^t = \lambda_i\right) p\left(\mathbf{l}^t = \lambda_i\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1 \mid h_1, \mathbf{l}^t = \lambda_i\right) p\left(y_2 \mid h_2, \mathbf{l}^t = \lambda_i\right) ... p\left(y_T \mid h_T, \mathbf{l}^t = \lambda_i\right) p\left(h_1, h_2, ..., h_T\right) p\left(\mathbf{l}^t = \lambda_i\right)$$

$$= \sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1 \mid h_1, \mathbf{l}^t = \lambda_i\right) p\left(y_2 \mid h_2, \mathbf{l}^t = \lambda_i\right) ... p\left(y_T \mid h_T, \mathbf{l}^t = \lambda_i\right) p\left(h_1, h_2, ..., h_T\right) \frac{1}{2^{n_s}}$$

$$= \sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1 \mid h_1, \mu_1\left(\mathbf{l}^t = \lambda_i\right)\right) p\left(y_2 \mid h_2, \mu_2\left(\mathbf{l}^t = \lambda_i\right)\right) ... p\left(y_T \mid h_T, \mu_T\left(\mathbf{l}^t = \lambda_i\right)\right) p\left(h_1, h_2, ..., h_T\right) \frac{1}{2^{n_s}} \quad (4.8)$$

with (4.7) and (4.8) , (4.6) becomes

$$= \frac{1}{T}\{\sum_{k=1}^{n_s}(1-\sum_{l_k^t=b\in\{0,1\}}\int_{y_1,y_2,...,y_T}\int_{h_1,...,h_T}\sum_{\mathbf{l}^t=\lambda_j\in\Lambda_k^b} p\left(y_1\mid h_1,\mu_1\left(\mathbf{l}^t=\lambda_j\right)\right)...p\left(y_T\mid h_T,\mu_T\left(\mathbf{l}^t=\lambda_j\right)\right)p\left(h_1,h_2,...,h_T\right)\frac{1}{2^{n_s}}\cdot$$

$$\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1\mid h_1,\mu_1\left(\mathbf{l}^t=\lambda_i\right)\right)...p\left(y_T\mid h_T,\mu_T\left(\mathbf{l}^t=\lambda_i\right)\right)p\left(h_1,h_2,...,h_T\right)\frac{1}{2^{n_s}}}{\sum_{\mathbf{l}^t=\lambda_j\in\Lambda_k^b} p\left(y_1\mid h_1,\mu_1\left(\mathbf{l}^t=\lambda_j\right)\right)...p\left(y_T\mid h_T,\mu_T\left(\mathbf{l}^t=\lambda_j\right)\right)p\left(h_1,h_2,...,h_T\right)\frac{1}{2^{n_s}}} dy_1...dy_T dh_1,...dh_T\} \quad (4.9)$$

$$= \frac{1}{T}\{\sum_{k=1}^{n_s}(1-\sum_{l_k^t=b\in\{0,1\}}\int_{y_1,...,y_T}\int_{h_1,...,h_T}\sum_{\mathbf{l}^t=\lambda_j\in\Lambda_k^b}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-h_k\mu_k\left(\mathbf{l}^t=\lambda_j\right)\|^2}{2\sigma^2}\right)p\left(h_1,...,h_T\right)\frac{1}{2^{n_s}}\cdot$$

$$\log_2\frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-h_k\mu_k\left(\mathbf{l}^t=\lambda_i\right)\|^2}{2\sigma^2}\right)}{\sum_{\mathbf{l}^t=\lambda_j\in\Lambda_k^b}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-h_k\mu_k\left(\mathbf{l}^t=\lambda_j\right)\|^2}{2\sigma^2}\right)} dy_1...dy_T dh_1,...dh_T\} \quad (4.10)$$

Different from the CM capacity which is only determined by the configuration of constellation points in the enlarged signal space, the BICM capacity also depends on the number of bit differences between symbols. The term $\sum_{\lambda_j\in\Lambda_k^b}\exp\left(-\sum_{k=1}^{T}\frac{\|y_k-h_k\mu_k\left(\lambda_j\right)\|^2}{2\sigma^2}\right)$, which is summed over those symbols whose k-th bit is b, apparently depends strongly on the labeling on symbols. Since mappings designed to have large CM capacity often aim to enlarge the Euclidean distance between constellation points and ignore the effect of the number of bit differences between symbols, they do not necessary achieves high BICM capacity.

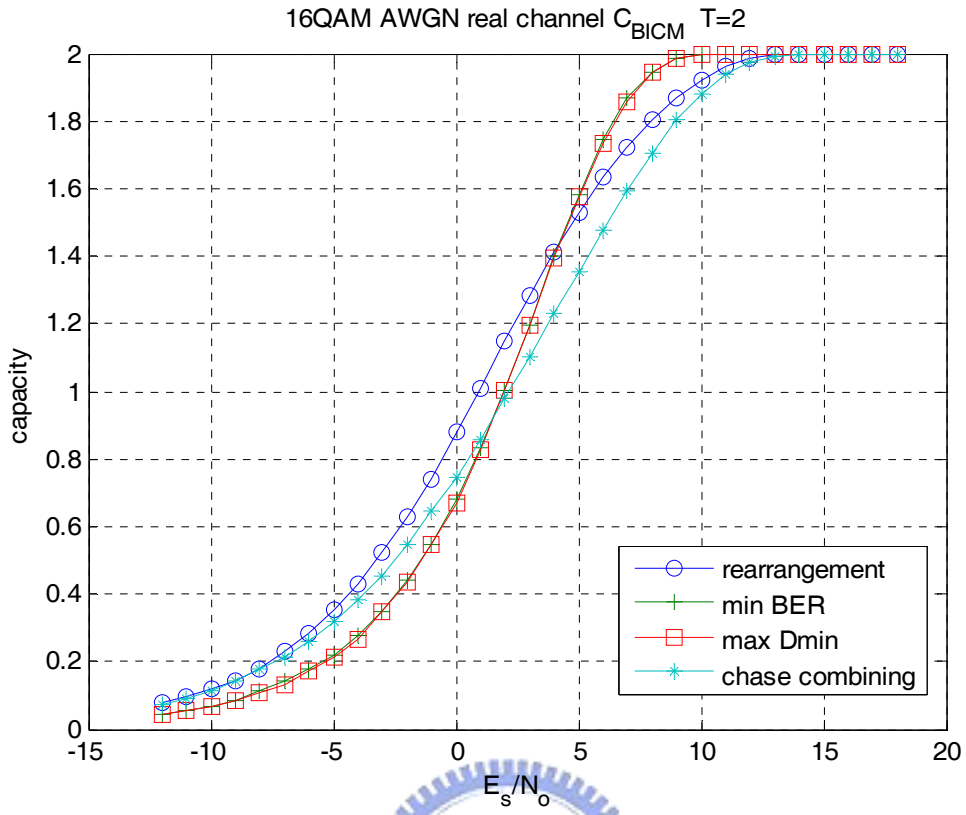

Fig 4.3.2 BICM capacity of different mapping schemes

As shown in Fig 4.3.2, MBER and MDMIN have low BICM capacity at low SNR region, they outperform constellation rearrangement and Chase combining only at high SNR region. Constellation rearrangement, on the other hand, has higher BICM capacity than Chase Combining over the entire SNR range. Hence the behavior of BICM capacity is different from CM capacity and the design criterion for BICM systems should be different from CM systems.

Chapter 5: Extrinsic Information Transfer Chart

Extrinsic information transfer chart (EXIT chart) was first proposed by S. ten Brink [7] to visualize the convergence behavior of iterative demapping and decoding. Mutual information between extrinsic log-likelihood ratio and coded bit are used to describe the exchange of soft information between the demapper and the decoder. The iterative processing is visualized as a decoding trajectory in the EXIT chart. Although primary used to describe the behavior of iterative processing, it turns out to be useful as well for the mapping design in BICM without iterative decoding.

5.1 Transfer characteristics

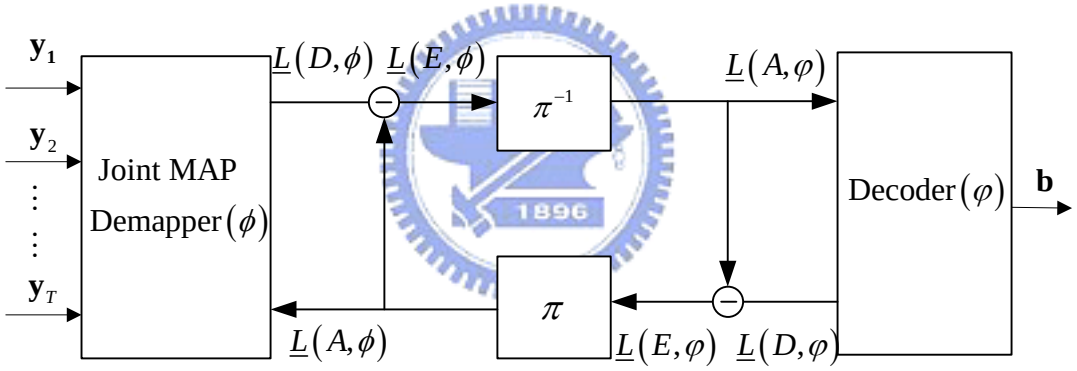

Fig.5.1.1 iterative demapping and decoding model

Fig 5.1.1 shows the iterative demapping and decoding model which is the same as in Fig 3.3.2.We follow the same notation convention defined in chapter 3, where $\underline{\mathbf{L}}(u,v)=\left(\mathbf{L}_1(u,v),\mathbf{L}_2(u,v),...,\mathbf{L}_{N_L}(u,v)\right)$ $u\in\{D,A,E\},v\in\{\phi,\varphi\}$ and $\mathbf{L}_k(u,v)$ $=\{L_{k,1}(u,v),L_{k,2}(u,v),....,L_{k,n_s}(u,v)\}$. Similarly, the subindex k is left out and we define $L_i(u,v)\triangleq L_{k,i}(u,v)$.

Consider a BPSK signal transmitted over AWGN channel, thus the received signal is $y=x+n$ where $x\in\{\pm1\}$ and n is AWGN with zero men and variance σ_n.

The conditional probability density function writes as $p(y \mid x) = \frac{e^{-\frac{(y-x)^2}{2\sigma_n^2}}}{\sqrt{2\pi}\sigma_n}$.

Thus the corresponding log-likelihood ratio is calculated as

$$L = \ln \frac{p(y \mid x=1)}{p(y \mid x=-1)} = \ln \frac{\frac{e^{-\frac{(y-1)^2}{2\sigma_n^2}}}{\sqrt{2\pi}\sigma_n}}{\frac{e^{-\frac{(y+1)^2}{2\sigma_n^2}}}{\sqrt{2\pi}\sigma_n}} = \frac{2}{\sigma_n^2} y = \frac{2}{\sigma_n^2}(x+n) \tag{5.1}$$

From the above equation, the L can be further modeled as $L = \mu_L \cdot x + n_L$

with

$$\mu_L = \frac{2}{\sigma_n^2} \tag{5.2}$$

and n_L being Gaussian distributed with zero mean and variance

$$\sigma_L = \frac{4}{\sigma_n^2} \tag{5.3}$$

Thus, the mean and variance of L satisfy the following relationship

$$\mu_L = \frac{\sigma_L^2}{2} \tag{5.4}$$

We make use of the following observation for modeling a priori information.

1. For large interleavers the a priori $L_i(A;v), v \in \{\phi, \varphi\}$ values remain fairly uncorrelated from the respective channel observations $\mathbf{y} = [y_1, ..., y_T]$ over many iterations.
2. The probability density functions of the extrinsic output values $L_i(E;v), v \in \{\phi, \varphi\}$ approach Gaussian-like distributions with increasing number of iterations.

Observation 1 and 2 suggest that the a priori input $L_i(A;v), v \in \{\phi, \varphi\}$ can be modeled as an independent Gaussian random variable with mean $\mu_A = \frac{\sigma_A^2}{2}$ and

variance σ_A^2 in conjunction with the known transmitted bits x.

$$L_i(A,v) = \mu_A \cdot x + n_A, v \in \{\phi, \varphi\} \tag{5.5}$$

Therefore, the conditional probability density function for $L_i(A,v), v \in \{\phi, \varphi\}$ is given by

$$P_{L_i(A,v)}(\eta \mid X = x) = \frac{1}{\sqrt{2\pi}\sigma_A} \exp\left(-\frac{\left(\eta - \frac{\sigma_A^2}{2}x\right)^2}{2\sigma_A^2} \right) \tag{5.6}$$

With (5.6), we can calculate the mutual information between the transmitted bits and the prior LLR.

$$\begin{aligned} & I_A \\ &= I\left(X; L_i(A,v)\right) \\ &= \frac{1}{2}\sum_{x=-1.1} \int_{-\infty}^{\infty} p_{L_i(A,v)}(\eta \mid X = x) \log_2 \frac{p_{L_i(A,v)}(\eta \mid X = x)}{\frac{1}{2} p_{L_i(A,v)}(\eta \mid X = -1) + \frac{1}{2} p_{L_i(A,v)}(\eta \mid X = 1)} d\eta \end{aligned} \tag{5.7}$$

with (5.6),(5.7) becomes

$$I_A = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_A} e^{-\frac{\left(\eta - \frac{\sigma_A^2}{2}\right)^2}{2\sigma_A^2}} \log_2\left(1 + e^{-\eta}\right) d\eta \tag{5.8}$$

We define $J(\sigma_A) \triangleq I_A$. Note that (5.8) can not be expressed in closed form; however, it is monotonically increasing in σ_A and therefore reversible.

$$\sigma_A = J^{-1}(I_A) \tag{5.9}$$

We can set up a table for the relationship of I_A and σ_A, thus generating a priori LLR for a specific value of I_A according to the Gaussian distribution $N\left(\frac{\sigma_A^2}{2}x, \sigma_A\right)$

Mutual information is also used to quantify the extrinsic information output of the demapper or decoder.

$$
\begin{aligned}
& I_E \\
&= I\left(X; L_i\left(E, v\right)\right) \\
&= \frac{1}{2} \sum_{x=-1.1} \int_{-\infty}^{\infty} p_{L_i(E,v)}\left(\eta \mid X = x\right) \log_2 \frac{p_{L_i(E,v)}\left(\eta \mid X = x\right)}{\frac{1}{2} p_{L_i(E,v)}\left(\eta \mid X = -1\right) + \frac{1}{2} p_{L_i(E,v)}\left(\eta \mid X = 1\right)} d\eta
\end{aligned} \tag{5.10}
$$

Note that no Gaussian assumption is imposed on the extrinsic output distribution $p_{L_i(E,v)}\left(\eta \mid X = x\right)$. The conditional PDF of $L_i\left(E, v\right)$ is obtained by Monte Carlo simulation (histogram measurements).

The joint map demapper takes not only the a priori information $L_i\left(A, \phi\right)$ but also the T channel outputs $Y_1, Y_2, ..., Y_T$ which is dependent on the E_s / N_o value for each transmission. Furthermore, the mapping scheme $\mu_1, \mu_2, ..., \mu_T$ also affects the extrinsic output. Therefore, we define the demapper transfer characteristics as $I_E = T_{de}\left(I_A, E_s / N_o, \mu_1, \mu_2, ..., \mu_T\right)$. For the decoder, only the a priori information is taken to compute the extrinsic information, thus we define the transfer characteristics for the decoder as $I_E = T_{dc}\left(I_A\right)$

5.2 Transfer Characteristics of the Demapper

5.2.1 Demapper Transfer Function

To simplify our notation, we define the a priori and extrinsic information for the i-th symbol as $\mathbf{l}^a = (l_1^a, l_2^a, ..., l_{n_s}^a) \triangleq L_i\left(A, \Phi\right)$ and $\mathbf{l}^e = (l_1^e, l_2^e, ..., l_{n_s}^e) \triangleq L_i\left(E, \Phi\right)$. $\mathbf{l}_{[k]}^a$ denotes the a priori information without the k-th bit prior $\mathbf{l}_{[k]}^a = (l_1^a, \ldots, l_{k-1}^a, l_{k+1}^a .., l_{n_s}^a)$ The demapper output mutual information is then computed by:

$$I_E = \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; L^e\right)$$
$$= \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T, \mathbf{L}_{[k]}^a\right) \quad (5.11)$$

Where the validity of the second step has been proved to be true in [8] for the case of decoders and the same proof can be used for demappers. In (5.11)

$$I\left(L_k^t; Y_1, Y_2, ..., Y_T, \mathbf{L}_{[k]}^a\right)$$
$$= H\left(L_k^t\right) - H\left(L_k^t \mid Y_1, Y_2, ..., Y_T, \mathbf{L}_{[k]}^a\right)$$
$$= 1 - \sum_{l_k^t = b \in \{0,1\}} \int_{y_1, y_2, ..., y_T} \int_{\mathbf{l}_{[k]}^a} p\left(l_k^t = b, y_1, y_2, ..., y_T, \mathbf{l}_{[k]}^a\right) \log_2 \frac{p\left(y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a\right)}{P\left(l_k^t = b, y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a\right)} dy_1 .. dy_T d\mathbf{l}_{[k]}^a \quad (5.12)$$

where

$$P\left(l_k^t = b, y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a\right)$$
$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda} P\left(\mathbf{l}^t = \lambda_j, l_k^t = b, y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a\right)$$
$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(\mathbf{l}^t = \lambda_j, y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a\right)$$
$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1, y_2, .., y_T, \mathbf{l}_{[k]}^a \mid \mathbf{l}^t = \lambda_j\right) p\left(\mathbf{l}^t = \lambda_j\right)$$
$$= \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1, y_2, .., y_T \mid \mathbf{l}^t = \lambda_j\right) p\left(\mathbf{l}_{[k]}^a \mid \mathbf{l}^t = \lambda_j\right) \frac{1}{2^{n_s}}$$
$$= \frac{1}{2^{n_s}} \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1 \mid \mathbf{l}^t = \lambda_j\right) P\left(y_2 \mid \mathbf{l}^t = \lambda_j\right) ... P\left(y_T \mid \mathbf{l}^t = \lambda_j\right) \prod_{i=1, i \neq k}^{n_s} p\left(l_i^a \mid l_i^t\right)$$
$$= \frac{1}{2^{n_s}} \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) P\left(y_2 \mid \mu_2\left(\mathbf{l}^t = \lambda_j\right)\right) ... P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i \neq k}^{n_s} p\left(l_i^a \mid l_i^t\right) \quad (5.13)$$

Similarly,

$$p\left(y_1, y_2, .., y_T, \mathbf{l}^a_{[k]}\right)$$

$$= \sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1, y_2, .., y_T, \mathbf{l}^a_{[k]}, \mathbf{l}^t = \lambda_i\right)$$

$$= \sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1, y_2, .., y_T, \mathbf{l}^a_{[k]} \mid \mathbf{l}^t = \lambda_i\right) p\left(\mathbf{l}^t = \lambda_i\right)$$

$$= \sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1, y_2, .., y_T \mid \mathbf{l}^t = \lambda_i\right) p\left(\mathbf{l}^a_{[k]} \mid \mathbf{l}^t = \lambda_i\right) p\left(\mathbf{l}^t = \lambda_i\right)$$

$$= \frac{1}{2^{n_s}} \sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1 \mid \mathbf{l}^t = \lambda_i\right) p\left(y_2 \mid \mathbf{l}^t = \lambda_i\right) ... p\left(y_T \mid \mathbf{l}^t = \lambda_i\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)$$

$$= \frac{1}{2^{n_s}} \sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_i\right)\right) p\left(y_2 \mid \mu_2\left(\mathbf{l}^t = \lambda_i\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_i\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right) \qquad (5.14)$$

With (5.13) and (5.14), (5.12) becomes

$$I\left(L^t_k; Y_1, Y_2, ..., Y_T, \mathbf{L}^a_{[k]}\right)$$

$$= 1 - \sum_{l^t_k=b\in\{0,1\}} \int_{y_1, y_2, ..., y_T} \int_{\mathbf{l}^a_{[k]}} \{\frac{1}{n_s} \sum_{\mathbf{l}^t=\lambda_j\in\Lambda^b_k} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) ... P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)$$

$$\cdot \log_2 \frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_i\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_i\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)}{\sum_{\mathbf{l}^t=\lambda_j\in\Lambda^b_k} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) ... P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)} \} dy_1 .. dy_T d\mathbf{l}^a_{[k]} \qquad (5.15)$$

Thus

$$I_E = \frac{1}{n_s} \sum_{k=1}^{n_s} I\left(L^t_k; L^e\right)$$

$$= \frac{1}{n_s} \sum_{k=1}^{n_s} I\left(L^t_k; Y_1, Y_2, ..., Y_T, \mathbf{L}^a_{[k]}\right)$$

$$= 1 - \frac{1}{n_s} \sum_{k=1}^{n_s} \sum_{l^t_k=b\in\{0,1\}} \int_{y_1, y_2, ..., y_T} \int_{\mathbf{l}^a_{[k]}} \{\frac{1}{n_s} \sum_{\mathbf{l}^t=\lambda_j\in\Lambda^b_k} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) ... P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right) \cdot$$

$$\cdot \log_2 \frac{\sum_{\mathbf{l}^t=\lambda_i\in\Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_i\right)\right) ... p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_i\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)}{\sum_{\mathbf{l}^t=\lambda_j\in\Lambda^b_k} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) ... P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l^a_i \mid l^t_i\right)} \} dy_1 .. dy_T d\mathbf{l}^a_{[k]} \qquad (5.16)$$

Equation (5.16) shows the demapper transfer function' s dependency on the a priori information, SNR , transmission number and the mapping scheme for the subsequent transmissions.

The demapper transfer function can also be obtained by assuming the a prior information $L_i(A,\phi)$ to be Gaussian distributed and applying equation (5.10).

5.2.2 Properties of the demapper transfer function

The properties of the demapper transfer function for single transmission [8] apply to the case of multiple transmissions. We show these extensions in the following.

5.2.2.1 Area property

To simplify the analysis of the area property of the demapper transfer function, the a priori input is modeled as the output of the binary erasure channel (BEC) (Fig 5.2.2.1).

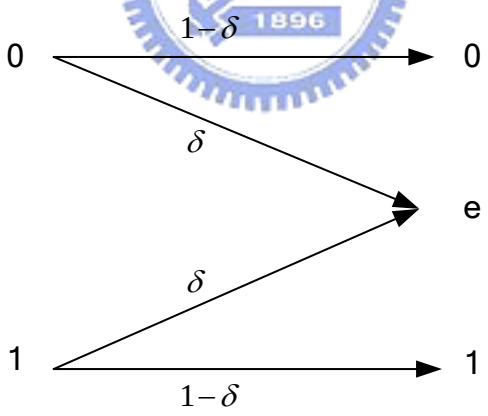

Fig 5.2.2.1 Binary Erasure Channel

For single transmission, it has been proved that when using the BEC model for the a priori input, the area under the demapper transfer function is $\frac{C_{CM}}{n_s}$. The same result can be applied for multiple transmissions as well. For single transmission,

$$
\begin{aligned}
A &= \int_0^1 I_E dI_A \\
&= \int_0^1 \frac{1}{n_s} \sum_{k=1}^{n_s} I\left(L_k^t; Y, \mathbf{L}_{[k]}^a\right) dI_A \\
&= \int_0^1 \frac{1}{n_s} \sum_{k=1}^{n_s} \left(H\left(L_k^t\right) - H\left(L_k^t \mid Y, \mathbf{L}_{[k]}^a\right)\right) dI_A \\
&= 1 - \frac{1}{n_s} \sum_{k=1}^{n_s} \int_0^1 H\left(L_k^t \mid Y, \mathbf{L}_{[k]}^a\right) dI_A \\
&= 1 - \frac{1}{n_s} H\left(\mathbf{L}^t \mid Y\right) \\
&= \frac{H(X) - H(X \mid Y)}{n_s} \\
&= \frac{I(X;Y)}{n_s} = \frac{C_{CM}}{n_s}
\end{aligned}
\tag{5.17}
$$

where the fifth step have been proved to be true in [8]. Observing that this derivation is independent of the received symbol Y, we can extend (5.17) to the case of multiple transmission.

$$
\begin{aligned}
A &= \int_0^1 I_E dI_A \\
&= \int_0^1 \frac{1}{n_s} \sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T, \mathbf{L}_{[k]}^a\right) dI_A \\
&= \int_0^1 \frac{1}{n_s} \sum_{k=1}^{n_s} \left(H\left(L_k^t\right) - H\left(L_k^t \mid Y_1, Y_2, ...Y_T, \mathbf{L}_{[k]}^a\right)\right) dI_A \\
&= \frac{H(X) - H(X \mid Y_1, Y_2, ..., Y_T)}{n_s} \\
&= \frac{T\frac{1}{T} I(X; Y_1, Y_2, ..., Y_T)}{n_s} = \frac{TC_{CM\_HARQ}}{n_s}
\end{aligned}
\tag{5.18}
$$

Although in both cases the area depend on the CM capacity, only in the case of multiple transmissions do the mapping scheme affects the area. Observe the CM capacity for single transmission:

$$C_{CM}$$

$$= n_s - \sum_{\lambda_i \in \Lambda} \int_{y=-\infty}^{\infty} p\left(y \mid \mu\left(\lambda_i\right)\right) \frac{1}{2^{n_s}} \log_2 \frac{\sum_{\lambda_j \in \Lambda} p\left(y \mid \mu\left(\lambda_j\right)\right)}{p\left(y \mid \mu\left(\lambda_i\right)\right)} dy$$

$$= n_s - \sum_{\lambda_i \in \Lambda} \int_{y=-\infty}^{\infty} p\left(y \mid \mu\left(\lambda_i\right)\right) \frac{1}{2^{n_s}} \log_2 \sum_{\lambda_j \in \Lambda} p\left(y \mid \mu\left(\lambda_j\right)\right) dy + \sum_{\lambda_i \in \Lambda} \int_{y=-\infty}^{\infty} p\left(y \mid \mu\left(\lambda_i\right)\right) \frac{1}{2^{n_s}} \log_2 p\left(y \mid \mu\left(\lambda_i\right)\right) dy$$

$$= n_s - \int_{y=-\infty}^{\infty} [\frac{1}{2^{n_s}} \log_2 \sum_{\lambda_j \in \Lambda} p\left(y \mid \mu\left(\lambda_j\right)\right)] \sum_{\lambda_i \in \Lambda} p\left(y \mid \mu\left(\lambda_i\right)\right) dy + \int_{y=-\infty}^{\infty} \sum_{\lambda_i \in \Lambda} p\left(y \mid \mu\left(\lambda_i\right)\right) \frac{1}{2^{n_s}} \log_2 p\left(y \mid \mu\left(\lambda_i\right)\right) dy$$

The second term and the third term at the last line is independent of mapping changes, thus the CM capacity for single transmission is determined solely by the channel E_b / N_o. This implies that for a fix channel condition, the area under the demapper transfer function is fixed.

For the case of multiple transmissions,

$$C_{CM\_HARQ}$$

$$= \frac{1}{T} \{ n_s - \sum_{\lambda_i \in \Lambda} \int_{y_1=-\infty}^{\infty} \cdots \int_{y_T=-\infty}^{\infty} p\left(y_1 \mid \mu_1\left(\lambda_i\right)\right) \cdots p\left(y_T \mid \mu_T\left(\lambda_i\right)\right) \frac{1}{2^{n_s}} \log_2 \frac{\sum_{\lambda_j \in \Lambda} p\left(y_1 \mid \mu_1\left(\lambda_j\right)\right) \cdots p\left(y_T \mid \mu_T\left(\lambda_j\right)\right)}{p\left(y_1 \mid \mu_1\left(\lambda_i\right)\right) \cdots p\left(y_T \mid \mu_T\left(\lambda_i\right)\right)} dy_1 \ldots dy_T \}$$

The term $p\left(y_1 \mid \mu_1\left(\lambda_j\right)\right) \cdots p\left(y_T \mid \mu_T\left(\lambda_j\right)\right)$ is affected by different combinations of mapping scheme for each transmission, thus the area is not only determined by the channel E_b / N_o but also the transmission mappings.

5.2.2.2 Zero prior characteristics

Comparing the BICM capacity (4.9) (for the special case of AWGN channel) with the demapper transfer function (5.16), they have quite similar form except that there are weightings $\prod_{i=1, i \neq k}^{n_s} p\left(l_i^a \mid l_i^t\right)$ in the summation of the conditional pdf. $P\left(y_1 \mid \mu_1\left(\mathbf{I}^t = \lambda_j\right)\right) \cdots P\left(y_T \mid \mu_T\left(\mathbf{I}^t = \lambda_j\right)\right)$. In fact, it is the priori information that gives weightings contributes to the increasing of the demapper transfer function. The

similarity implies the close relationship of the BICM capacity and the demapper transfer function. Consider the case of zero prior input, the demapper transfer function is now given by

$$\begin{aligned} I_E\left(I_A=0, E_s/N_o, \mu_1, ..., \mu_T\right) &= \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; L^e\right) \\ &= \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T\right) \\ &= \frac{T}{n_s}\frac{1}{T}\sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T\right) \\ &= \frac{T}{n_s} C_{BICM\_HARQ} \end{aligned} \tag{5.19}$$

In the second step, the prior LLR $\mathbf{L}_{[k]}^a$ is just omitted due to the lack of a priori information. (5.19) shows the zero prior characteristics is just proportional to the BICM capacity; therefore, the BICM capacity can be interpreted as the demapper output mutual information when the demapper has zero prior input. This observation gives us a guideline to the design of the retransmission mapping in BICM.

5.2.2.3 Summaries of the Properties of the Demapper Transfer Curve

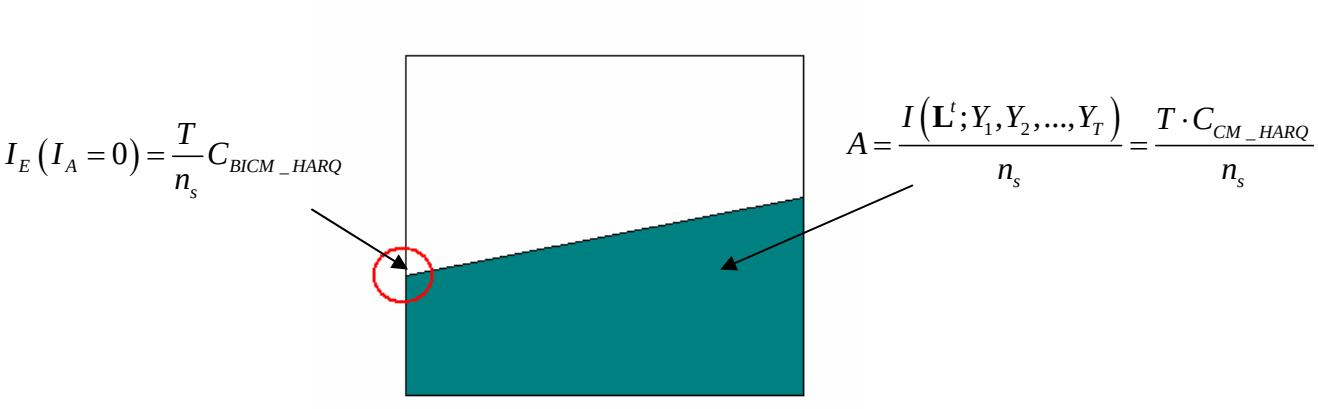

Fig 5.2.2.2 Properties of the Demapper Transfer Curve

Here we summarize the properties of the demapepr transfer curve (Fig 5.2.2.3). The left end point of the demapper transfer curve is directly proportional to the BICM

capacity and the area under the demapper transfer curve is directly proportional to the CM capacity.We need to emphasis that the area property holds true only if the a priori input is modeled as the output from the BEC channel.

5.2.2.4 Some Examples of Demapper Transfer Curve

Fig 5.2.2.4.1 shows some example of different demapper transfer functions for single transmission when 16QAM modulation and AWGN channel is assumed. Although different demappers have different transfer curves, the area under them are approximately the same. When the channel SNR is changed as shown in Fig 5.2.2.4.2, the shape stays roughly the same and the demapper transfer function simply shifts up or down.

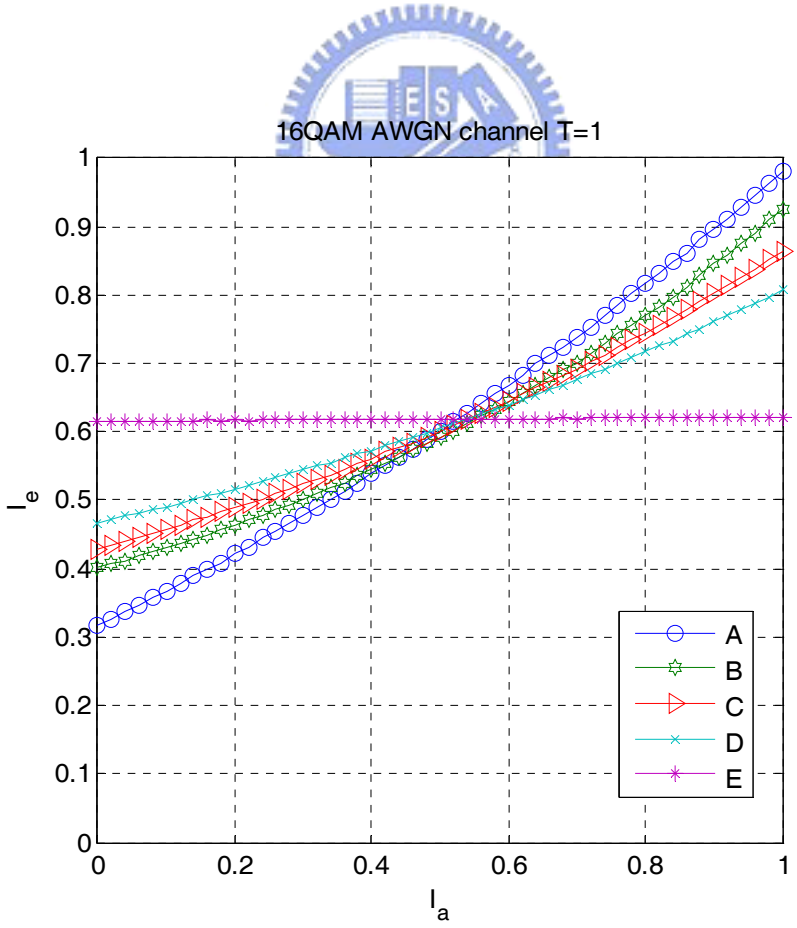

Fig 5.2.2.4.1 Various demapper transfer curve at the same SNR (T=1)

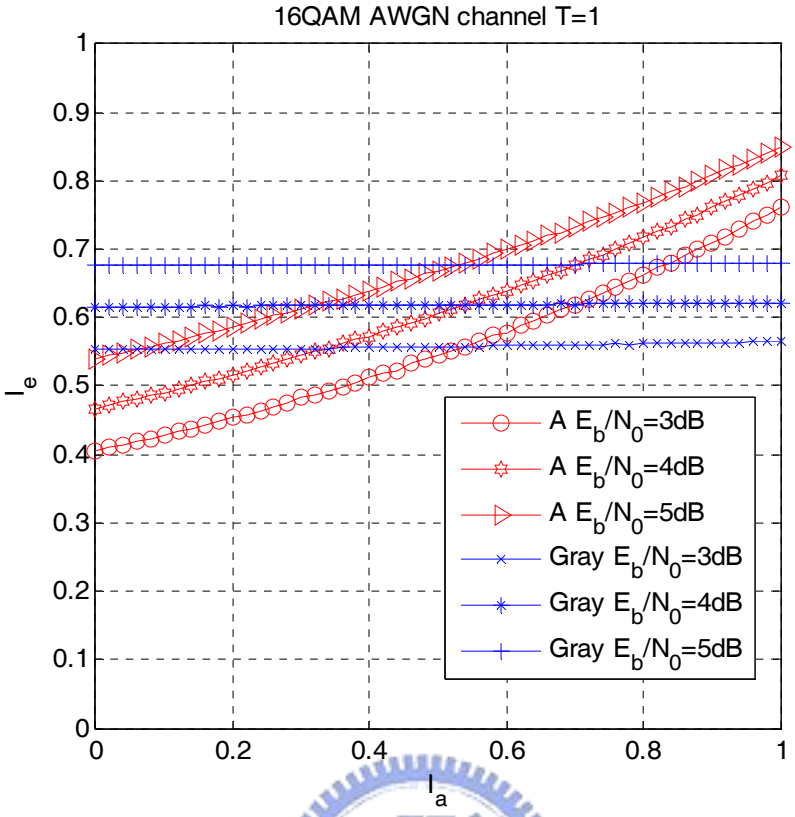

Fig 5.2.2.4.2 demapper transfer curve at different SNR (T=1)

When there are multiple transmissions, the area under the demapper transfer curve can not be assumed to be the same and will be varied by the different mapping combinations. Fig 5.2.2.4.3 illustrates that the choice of retransmission mappings not only affects the shape of the joint demapper transfer curve but also the area under it. Also, the benefits of the change of the retransmission mappings can be seen by the potential larger area under the demapper transfer curve compared to chase combining.

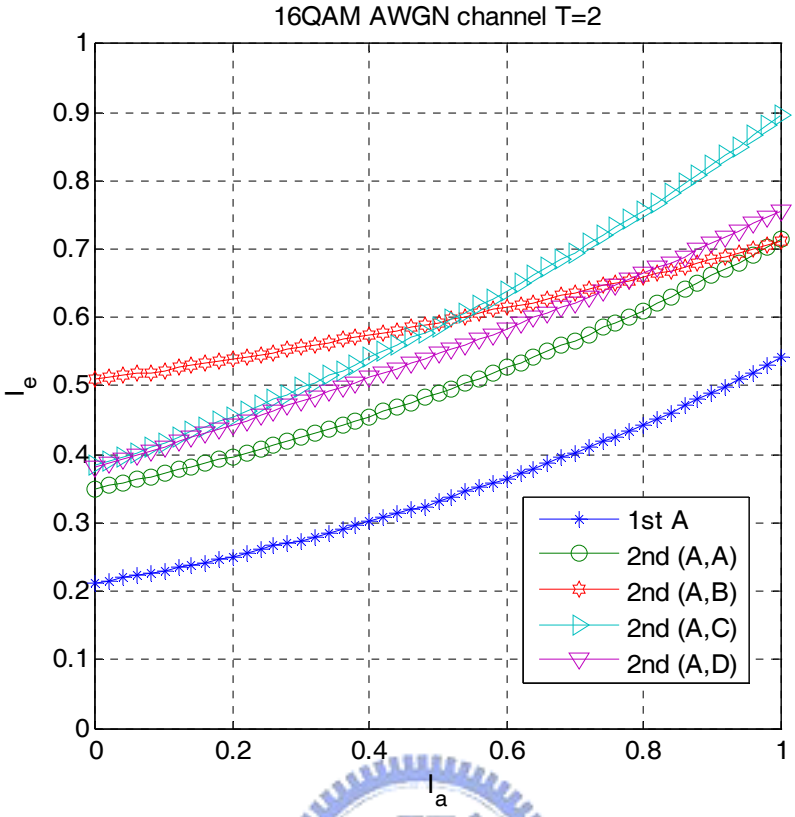

Fig 5.2.2.4.3 Various demapper transfer curve at the same SNR(T=2)

5.3 Transfer Characteristics of Decoder

For the decoder transfer curve, the a priori input I_a is plotted on the ordinate and the extrinsic output I_e is on the abscissa. Fig 5.3.1 shows some commonly used decoder curve transfer curve. The generator polynomials for convolutional code are represented as octal numbers, with the most significant bit corresponding to the generator connection on the very left side of the shift register. As can be observed, different codes with different error correcting capability exhibit different transfer characteristics. A less powerful code has a smooth rise in the middle while a more powerful code behaves like a step function. This suggests that a less powerful code should be equipped with a demapper with steeper transfer curve to benefit from the iterative decoding gain, while a more powerful code should be designed with a demapper with a flatter transfer curve to avoid early intersection.

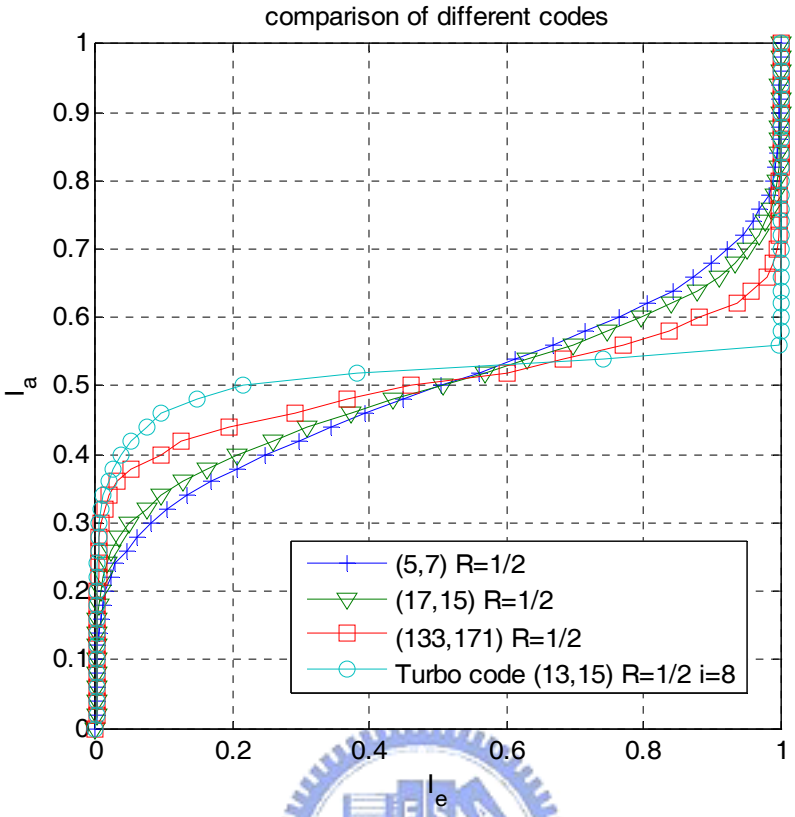

Fig 5.3.1 decoder transfer curve of different codes

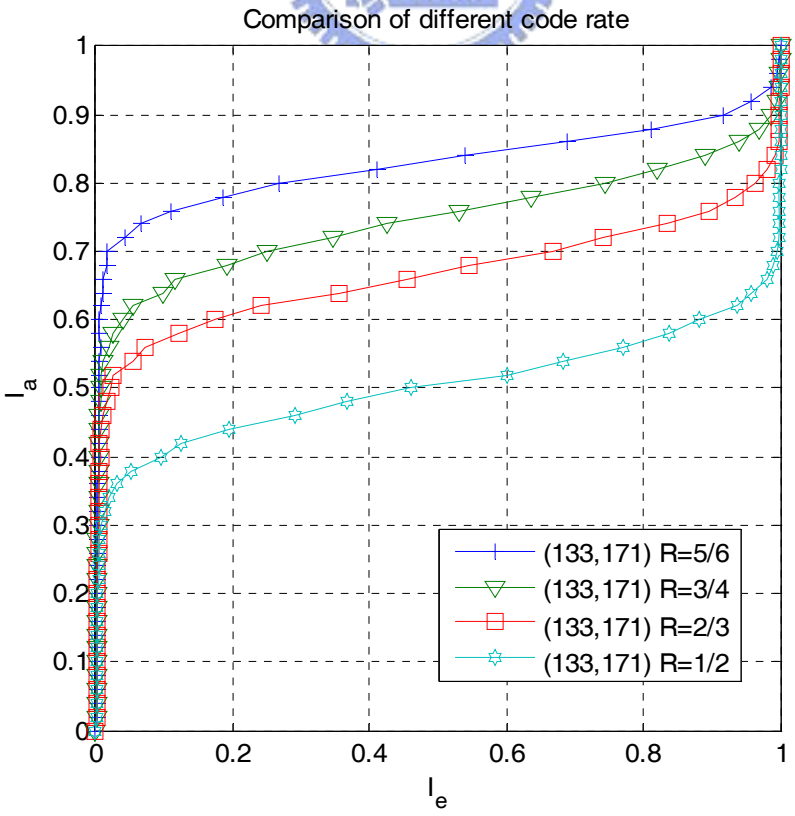

Fig 5.3.2 decoder transfer curve of different code rate

Fig 5.3.2 shows the transfer curve of the same code punctured to different code rate. In [8], it has been proved that the area under the decoder transfer curve is equal to the code rate under the assumption of BEC a priori input. Thus a code with higher code rate has a large area, this requires higher SNR for the tunnel between the demapper and the decoder transfer curve to open.

When a code has inner iteration between the component decoders, the number of inner iterations also affects the decoder transfer curve. Fig 5.3.3 plots the transfer curve of a turbo code with different inner iteration number. As can be observed, higher inner iteration number exhibits better transfer characteristics. For the iteration number above 8, no significant further improvement has been observed, which is in agreement with the BER simulation.

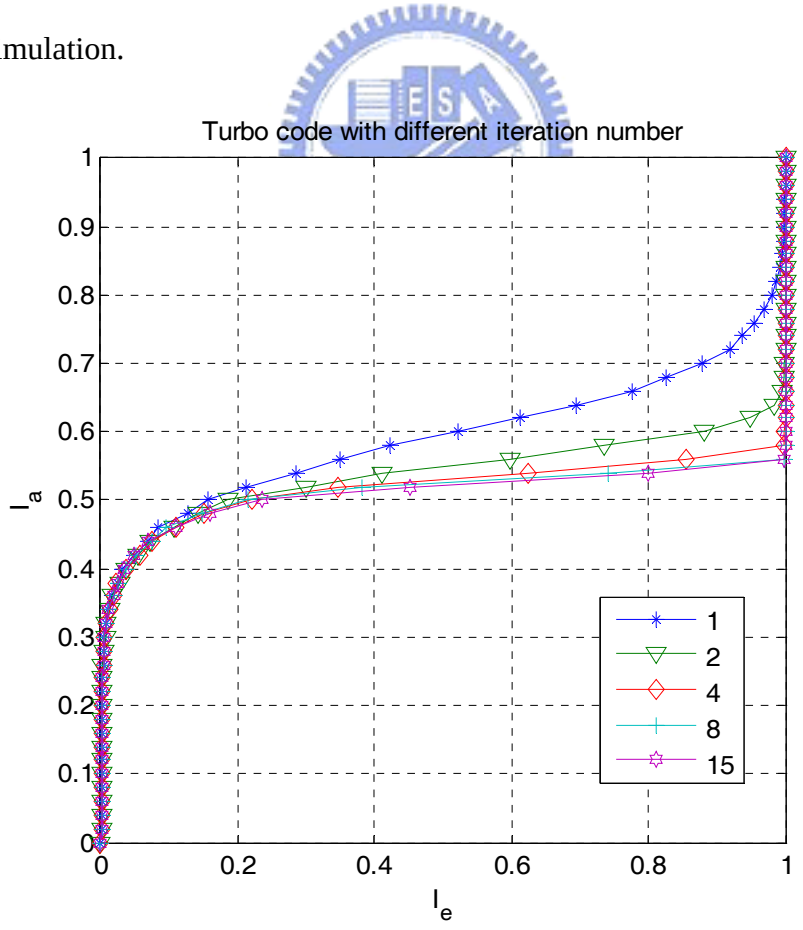

Fig 5.3.3 decoder transfer curve of turbo code with different iteration number

5.4 Extrinsic Information Transfer Chart (EXIT Chart)

To visualize the exchange of extrinsic information, we plot demapper and decoder characteristics into a single diagram which is referred to as Extrinsic Information Transfer Chart. The abscissa is the a priori input I_{A_1} for the demapper and the ordinate is the demapper extrinsic output I_{E_1}. However, the abscissa becomes the decoder extrinsic output I_{E_2} and the ordinate is the a priori input I_{A_2} for the decoder.

Let n be the iteration index. For a fixed SNR, transmission number and mapping scheme, the decoding trajectory starts at the left end point of the demapper transfer curve since no a priori information $I_{A_1,1}=0$ is outputted from the decoder. The demapper then passes extrinsic information $I_{E_1,1}=T_{de}\left(I_{A_1,1}=0\right)$ to the decoder. The decoder takes $I_{E_1,1}$ as a priori information $I_{A_2,1}$ and computes the extrinsic information $I_{E_2,1}=T_{dc}\left(I_{A_2,1}\right)$ for the demapper. The demapper then takes $I_{E_2,1}$ as $I_{A_1,2}$ and outputs $I_{E_1,2}=T_{de}\left(I_{A_1,2}\right)$ to the decoder for the next iteration. This iteration process continues until the trajectory reaches the intersection of the two transfer curve. For large interleaver block size, the simulated decoding trajectory matches well with the transfer curve. However, for small interleaver block size, the rapid increasing of the correlation of the extrinsic information makes the decoding trajectory deviate from the predicted one.

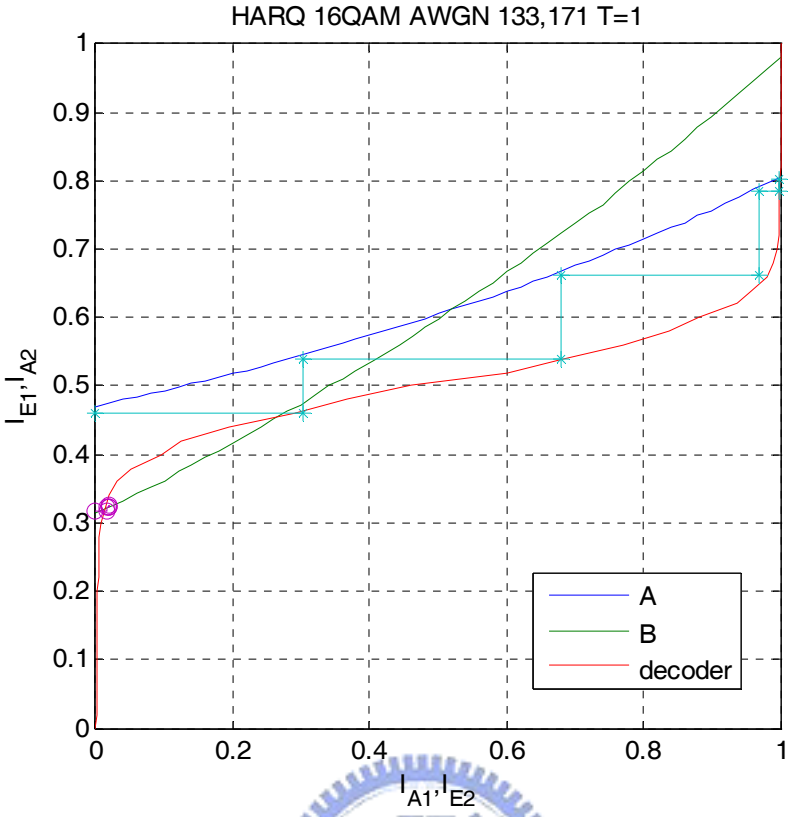

Fig 5.4 Examples of decoding trajectory

As Fig 5.4 illustrates, the iteration process stops at the intersection of the demapper and the decoder transfer curve. This intersection determines the final decoder output mutual information. The larger the decoder output mutual information, the better the BER performance. In our example, mapping A is a better choice for this decoder at this SNR, since the tunnel is opened and the first intersection is higher than mapping B. Hence it is essential to design the mapping scheme so that the intersection will be as right as possible.

Chapter 6: Mapping Design Criterion for BICM in HARQ

6.1 Motivation

We have already showed in chapter 3 that mappings should be changed for retransmissions. The additional benefit gained from the mapping diversity is the potential to increase the minimum squared Euclidean distance between the joint constellation points. Mapping diversity gain can also be evaluated from the coded modulation capacity and the bit-interleaved coded modulation capacity. A proper design of the retransmission mapping will increase either CM capacity or BICM capacity compared to chase combining. This comes at no extra power or bandwidth cost.

Although both the CM capacity and BICM capacity can reflect the effects of mapping change, BICM capacity is a better performance measurement for the HARQ system that adopt BICM scheme. This can be well explained from the analysis of the EXIT chart. BICM without iterative decoding can be seen as a special case of BICM-ID with only one iteration number. Therefore the decoder output mutual information is only determined by the demapper output mutual information with zero prior input. The higher the demapper input mutual information to the decoder, the higher the decoder output mutual information. The higher the decoder output mutual information, the lower the BER will be. Thus the demapper output mutual information with zero prior input will determine the final performance of this BICM system. As already proved in the previous chapter, the output of the demapper transfer function with zero prior input is equal to $\frac{T}{n_s} C_{BICM\_HARQ}$. Therefore the BICM capacity is a good performance measurement to evaluate different mapping schemes.

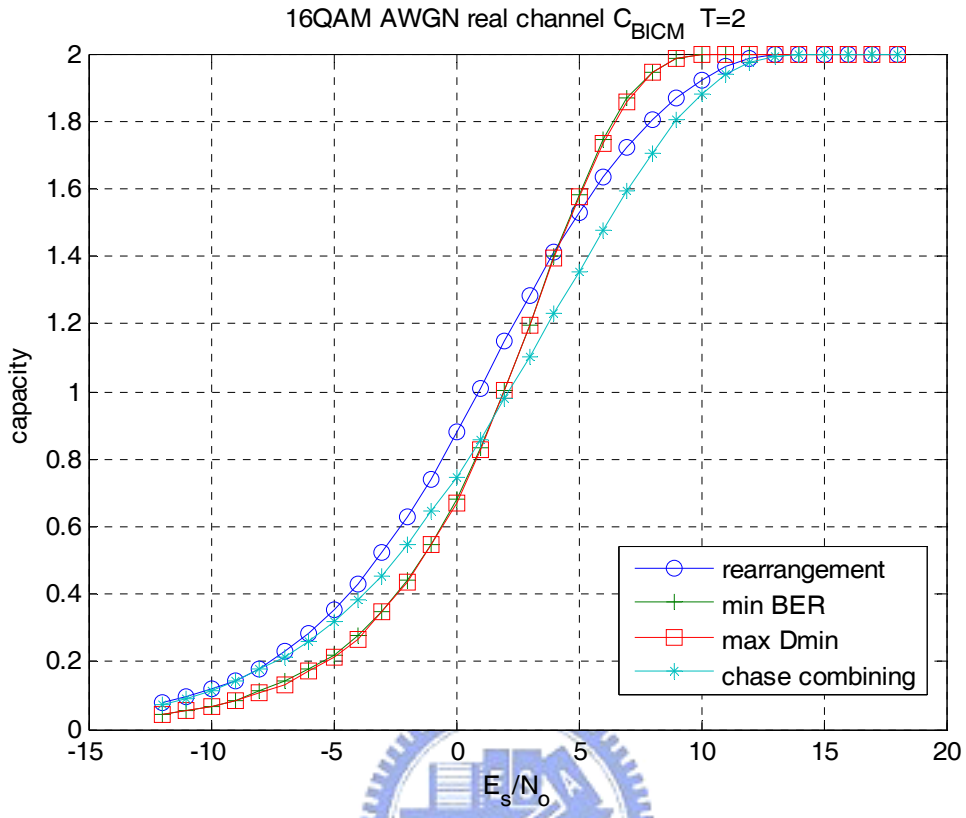

Fig 6.1 BICM capacity for various mapping schemes

Fig 6.1 plots the BICM capacity for different 16QAM mapping schemes with 2 transmissions. Three different mapping schemes proposed by previous work have been plotted together to compare their relative performance. As already being observed by [14], three distinct operation regions can be identified according to the relative performance order of different mappings. In the first operation region corresponds to SNR less than 1 dB, rearrangement [15] outperforms all the other schemes. While minimum BER upper bound (min BER) [16] and maximum minimum Euclidean distance (max Dmin) [18] perform even worse than Chase combining (no mapping change). In the second operation region defined by SNR range from 1dB to 4dB, min BER and max Dmin starts to surpass Chase combining. For SNR larger than 4dB, min BER and max Dmin scores best among rearrangement and chase combining. From the

observations above, mappings that perform well at low SNR region does not necessary perform well at high SNR region.

Conversely, mappings that are good at high SNR region are not necessary good at low SNR region as well. This suggests that mapping should be changed according to different channel code and code rate, since the operating SNR region will be different. There will have inevitable performance loss if only single mapping scheme is adopted.

6.2 Design Criterion

Motivated by the pervious observation, we state our design problem and propose the optimization criterion as follows:

Problem Statement:

Given a specific SNR and the previous 1,2,…, T-1 mappings, we want to find the T-th transmission mapping such that the BER is as small as possible.

Since there is a close relationship between BICM capacity and BER, we can state our optimization criterion as following:

Optimization Criterion:

$$\max_{\mu_T} C_{BICM\_HARQ}\left(SNR, \mu_1, ..., \mu_T\right)$$
$$= \max_{\mu_T} \frac{T}{n_s} C_{BICM\_HARQ}\left(SNR, \mu_1, ..., \mu_T\right)$$
$$= \max_{\mu_T} I_E\left(I_A = 0, SNR, \mu_1, ..., \mu_T\right)$$

under a prescribed SNR and $\mu_1, \mu_2, ..., \mu_{T-1}$

Chapter 7:Mapping Design Criterion for BICM-ID in HARQ

7.1 Motivation

Base on the analysis of the EXIT chart, the process of iterative decoding can be visualized as a decoding trajectory between the demapper transfer curve and the decoder transfer curve. The final output mutual information is determined by the first intersection of the demapper transfer curve and the decoder transfer curve. The higher the first intersection, the higher the decoder output mutual information. Since there will have different decoder transfer curve for different code and code rate, it is essential to design mapping specifically for a give code and code rate such that the first intersection is the highest. Mapping that is good for a specific code is not necessary good for another code. Fig 7.1.1 shows two sets of demapper and decoder transfer curve at a specific SNR.

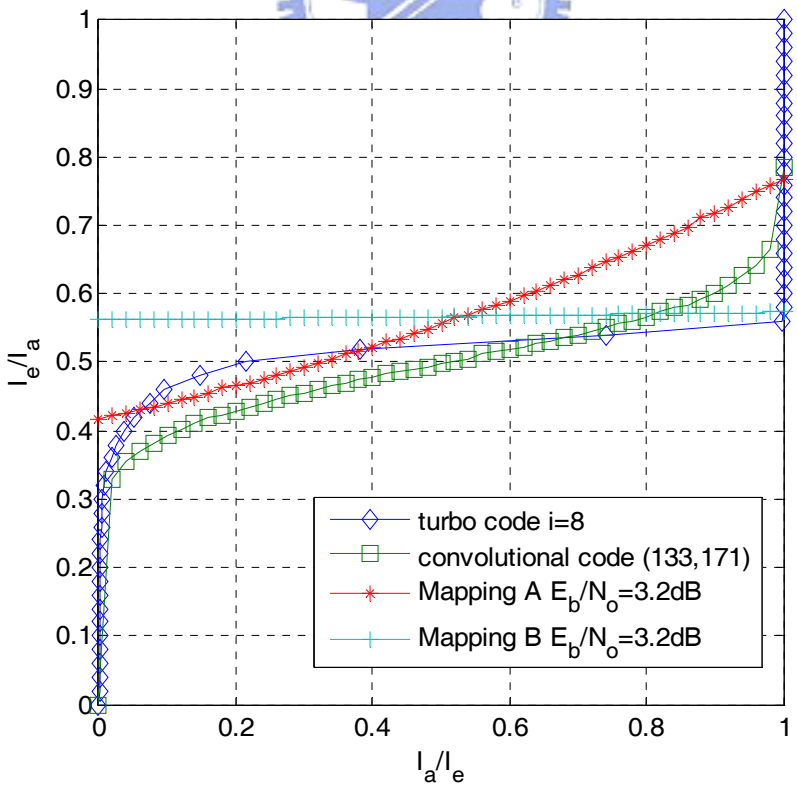

Fig 7.1.1 different decoder and demapper transfer curve

Comparing the two different mappings A and B, mapping A is more suitable for convolutional code than mapping B in terms of the first intersection point. Conversely, mapping B is more suitable for turbo code than mapping A. Thus mapping should be adapted for different code to obtain better iterative decoding performance.

The demappers not only have to be designed for different code and code rate, they should also be adapted for different SNR. As shown in Fig 7.1.2, mapping A is better than mapping C at 3.2 dB since higher SNR allows steeper demapper transfer curve to obtain higher intersection point. However, lower SNR requires flatter demapper transfer curve to avoid early intersection, thus mapping C is more suitable for low SNR. This implies different mappings should be chosen for different SNR.

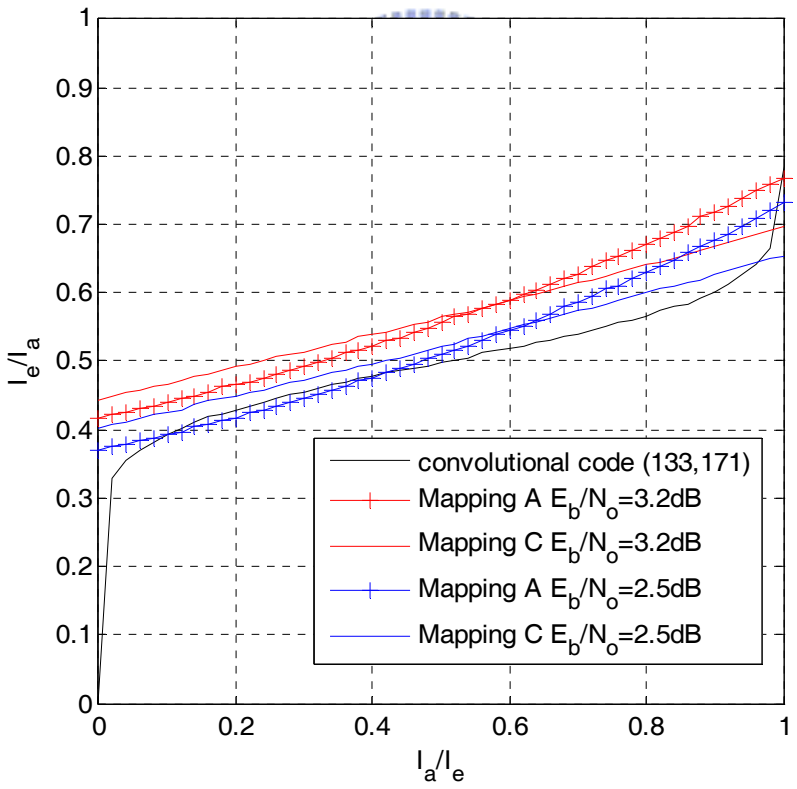

Fig 7.1.2 mappings at different SNR

7.2 Design Criterion For infinite Block Length

The transfer curve in the EXIT chart is based on the assumption that the block length is infinite so that the decoding trajectory matches quite well with the transfer curve. Hence we can predict the final decoding performance by the transfer curve. Furthermore, infinite iteration number is assumed so that the trajectory will get thorough even when a small tunnel is opened. Based on these assumptions, we state our design problem and suggest the optimization criterion in the following.

Problem Statement:

Given a specific SNR and the previous 1,2,…, T-1 mappings, we want to find the T-th transmission mapping such that the BER.is as small as possible.

We define the demapper transfer curve as $f_{\mu_1,\mu_2,...,\mu_T,SNR}(I_A)$ and the inverse decoder transfer curve as $g^{-1}(I_A)$. The first intersection point between the decoder transfer curve and the demapper transfer curve is I_A^*. I_A^* can be defined formally: $f_{\mu_1,\mu_2,...,\mu_T,SNR}(I_A) > g^{-1}(I_A), \forall I_A < I_A^*$ and $f_{\mu_1,\mu_2,...,\mu_T,SNR}(I_A^*) = g^{-1}(I_A^*)$. Since I_A^* determines the final decoder output mutual information, the optimization criterion can be stated as:

Optimization Criterion:

$$\max_{\mu_T} I_A^*$$

$$= \max_{\mu_T} I_E^* \text{.....since } g^{-1}(I_A) \text{ is none-decreasing function}$$

where $I_E^* = g^{-1}(I_A^*)$

under a prescribed SNR and $\mu_1, \mu_2, ..., \mu_{T-1}$

In the second step we applied the none-decreasing property of the decoder transfer

function. The reason to switch from the decoder final output to the decoder final input is that the mutual information output of the decoder at high SNR is difficult to discern. The scale is quite small and requires a huge amount of simulations to obtain high numerical precision. Therefore we aim to maximize the final decoder input for ease of the demand of numerical precision.

7.3 EXIT chart for Finite Block Length

In fact, the block length of a packet can not be infinite in real applications. Typically, the block length is finite and short in terms of the requirement of long block length for the assumption of EXIT chart to hold true. For short block length, the real transfer curve will deviate from the infinite one and the increased correlation in the iteration process will cause the decoding trajectory to die out at middle.

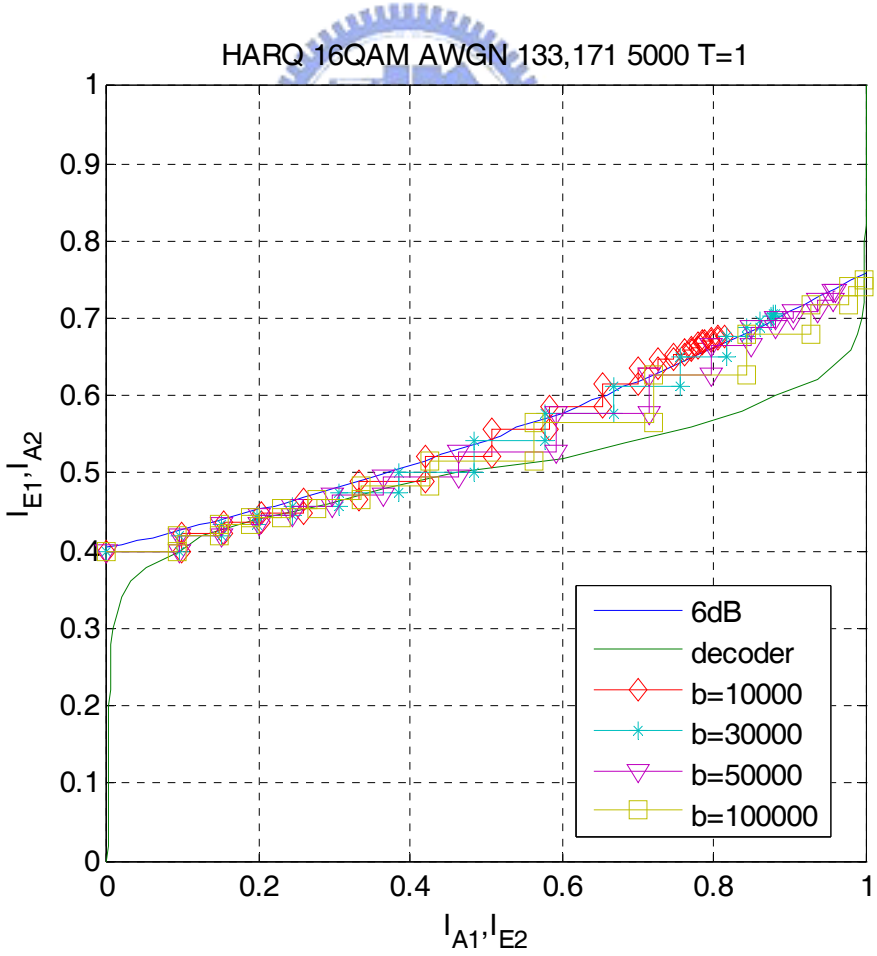

Fig 7.3.1 Decoding trajectory for finite block length

Fig 7.3.1 shows the decoding trajectory for different block length. As can be observed, the convergence point of the trajectory of finite block length codes disagree with the intersection of the demapper and decoder transfer curve. The shorter the block length is, the earlier the trajectory converges.

Since the original transfer curve of the EXIT chart is accurate only for large block length. For practical short block length design, we need to characterize the behavior of transfer curve. Instead of obtaining histogram from the extrinsic LLRs of many blocks, we measure the histogram only by one single block. After calculating the individual I_e for each block, we average them and obtain an averaged transfer curve.

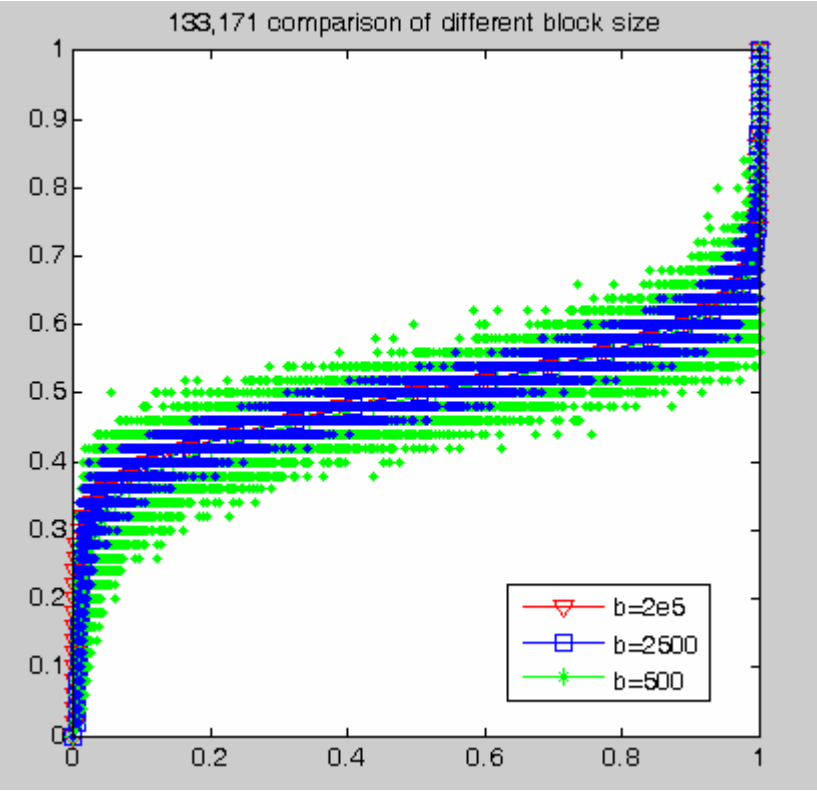

Fig 7.3.2 Variation of decoder transfer curve for different block length

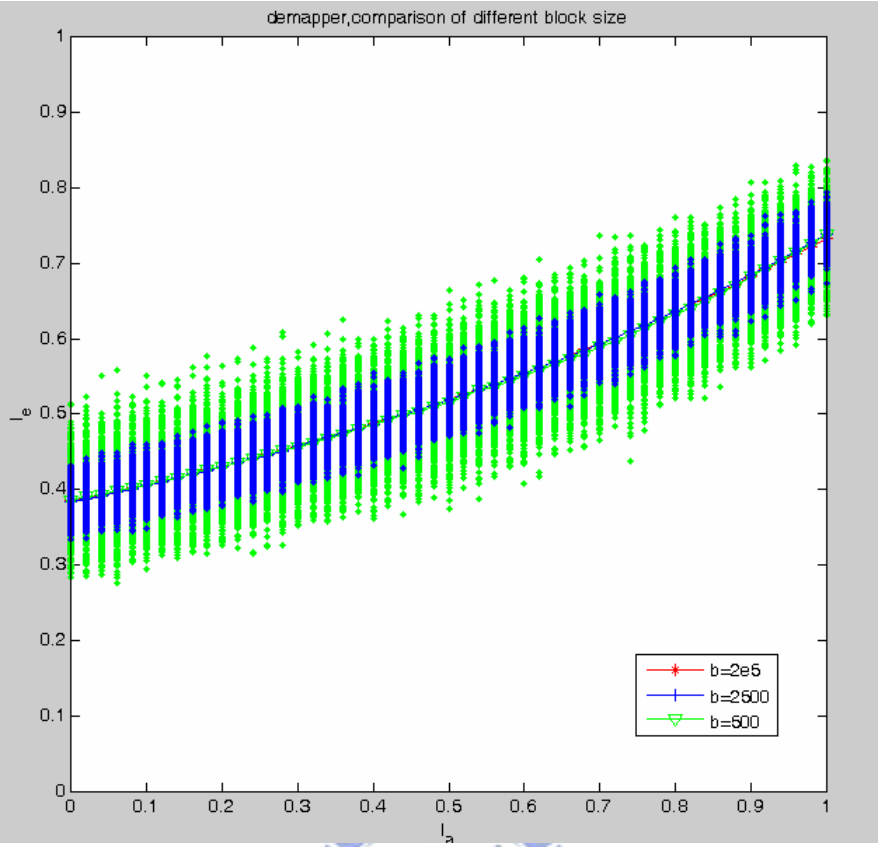

Fig 7.3.3 Variation of demapper transfer curve for different block length

The effect of finite block length can be seen clearly from these figures. A finite block length decoder or demapper will cause the transfer curve to swing back and forth around its mean value. The variation is larger for short block length. Although different block length exhibit different variations, their mean value are approximately the same as the transfer curve of very long block length.

7.4 Distribution of Output Extrinsic Mutual Information

We further plot the histogram of I_e corresponds to different I_a and compared them with normal distribution. The normal distribution is generated with the same mean and variance as the original data.

Distribution of demapper I_e

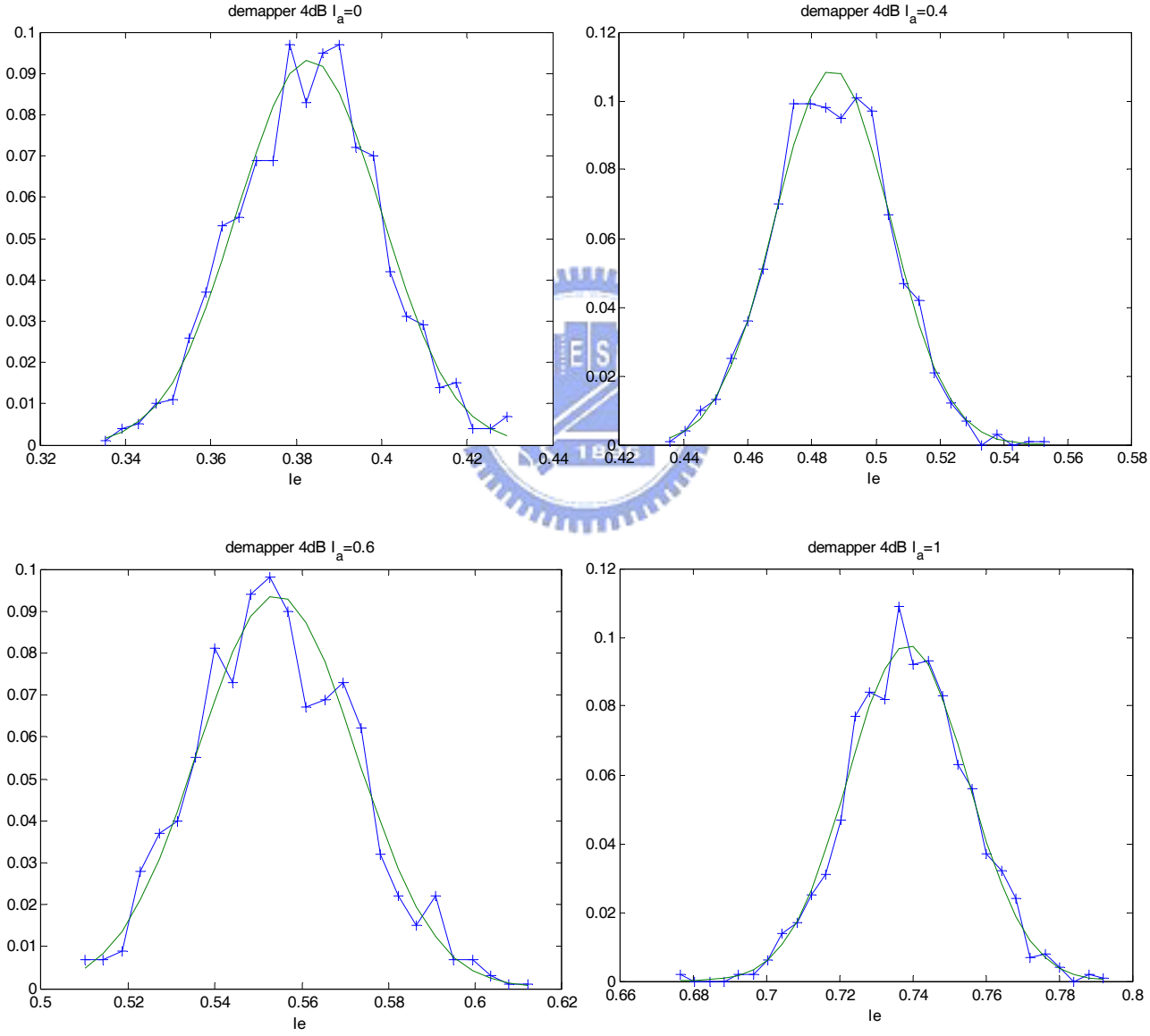

Fig. 7.4.1 Distribution of demapper transfer curve

Distribution of decoder I_e

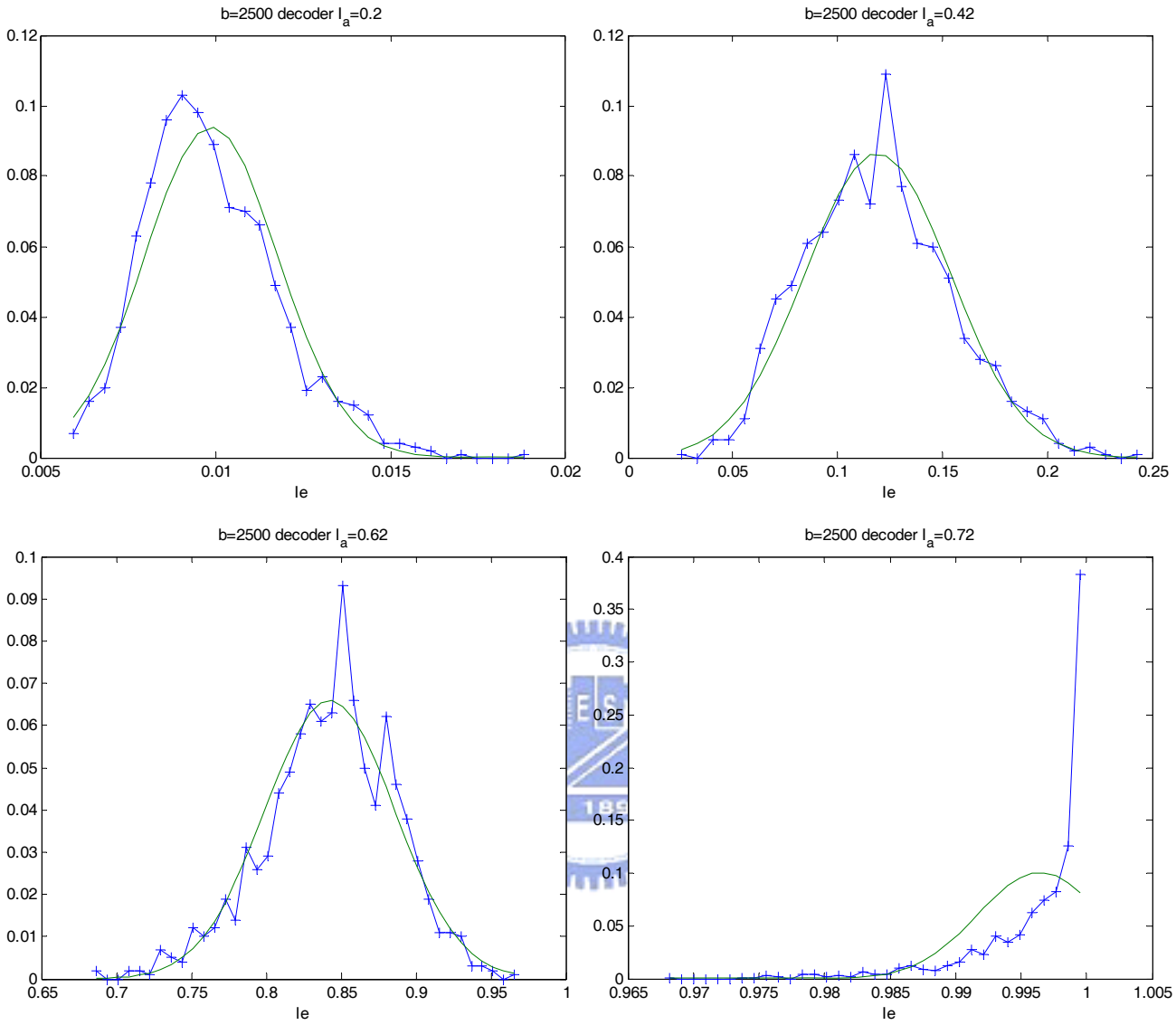

Fig 7.4.2 Distribution of decoder transfer curve

Observe that the histogram of I_e is very similar to normal distribution, except for decoder I_e with large a priori input I_a. Hence the demapper I_e and most part of the decoder I_e can be approximated as a Gaussian random variable.

7.5 Design Criterion for finite block length

Since the transfer curve of short bock length swings, the first intersection between the demapper and decoder transfer curve as predicted by the infinite one will no longer hold. If the transfer curve of demapper shifts down or the decoder shifts up for some realization, they will intersect each other too early. Hence this suggests that some margin between the demapper and decoder transfer curve should be preserved for tunnel to open in most realizations. Hence we modify the original criterion for infinite block length as following:

$$\max_{\mu} I_A^*$$

$$= \max_{\mu} I_E^* \text{.....since } g^{-1}\left(I_A\right) \text{ is none-decreasing function}$$

$$\text{subject to } \forall I_A < I_A^* - \sigma,\ f_{\mu,SNR}\left(I_A\right) > g^{-1}\left(I_A\right) + \delta$$

Where I_A^* is defined to be the first intersection as before. The only modification is the additional margin constraint. We limit the minimum distance between the decoder and the demapper transfer curve to be δ. Note that the minimum distance constraint is applied for I_A before $I_A^* - \sigma$ only. σ is preserved so that the demapper and decoder will intersect with each other.

Since the demapper and decoder I_e can be approximated as a Gaussian random variable, we can calculate the probability that the demapper or decoder will be in some particular region. Therefore, the margin δ can be chosen so that the tunnel will open for some percentage of realizations.

Chapter 8: Search Algorithm

8.1 Simplified Model

To evaluate the BICM capacity requires the computation of multiple integral which grows linearly with the transmission number T. Furthermore, the large number of intermediate mapping results during searching process demands huge amount of simulation which is unmanageable, thus it is required to simply the calculation of the demapper transfer function. In [9], a simplified model is proposed and a closed-form demapper transfer function is obtained for single transmission. Here we extend the result of [9] to the case of multiple transmission to facilitate the searching process.

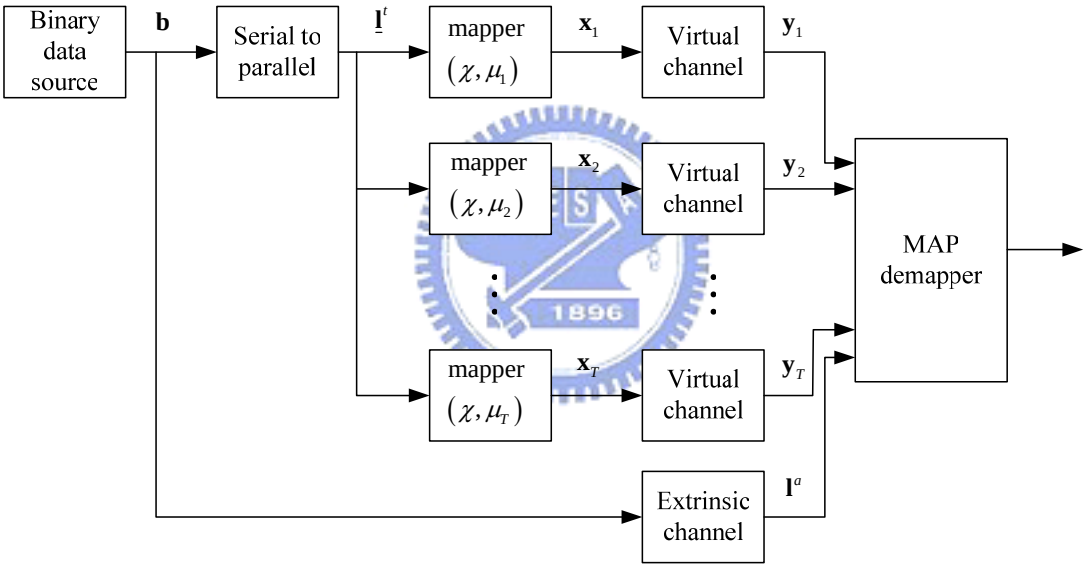

Fig.8.1 simplified channel model

8.1.1 Hard-decision Virtual Channel

The virtual channel is defined as a hard-decision channel that has the same capacity as the real channel in the simplified model (Fig 8.1). The output of the virtual channel $y_i \in \chi, i = 1,...,T$ is hard-decisioned to one of the transmitted signal point $x_i \in \chi, i = 1,...,T$. Therefore, the output alphabet is the same as the input alphabet. The virtual channel can be characterized by a sequence of transmission matrix

$\mathbf{Q} = [Q_1, Q_2, ..., Q_T]$. Each elements q_{ij}^k in the matrix Q_k denotes the transition probability with input signal s_i^k and output signal s_j^k, $q_{ij}^k = p\left(y_k = s_j^k \mid x_k = s_i^k\right),\ s_i^k, s_j^k \in \chi,\ i, j = 1, 2, ..., M, k = 1, 2, ..., T$. Since SNR is the same for each retransmission, the transmission matrix is the same : $Q_1 = Q_2 = ... = Q_T$. The virtual channel SNR γ_v is chosen to ensure that the capacity of the virtual channel $C_v\left(r_v\right)$ is equal to the capacity of the real channel $C_r\left(\gamma_r\right)$ for a fixed real channel SNR γ_r.

Let $l^{r_i} \in \Lambda, i = 1, 2, ..., T$ denotes the label corresponds to the received signal y_i, so that $l^{r_i} = \mu_i^{-1}\left(y_i\right)$, Given a sequence of transmission matrix $\mathbf{Q}$, the capacity of the virtual channel is computed as:

$$
\begin{aligned}
C_v\left(\gamma_v\right) &= \frac{1}{T} I\left(X_1, X_2, ..., X_T; Y_1, Y_2, ..., Y_T\right) \\
&= \frac{1}{T} I\left(\mathbf{L}^t, \mathbf{L}^t, ..., \mathbf{L}^t; \mathbf{L}^{r_1}, \mathbf{L}^{r_2}, ..., \mathbf{L}^{r_T}\right) \\
&= \frac{1}{T} I\left(\mathbf{L}^t; \mathbf{L}^{r_1}, \mathbf{L}^{r_2}, ..., \mathbf{L}^{r_T}\right) \\
&= \frac{1}{T}\left(H\left(\mathbf{L}^t\right) - H\left(\mathbf{L}^t \mid \mathbf{L}^{r_1}, \mathbf{L}^{r_2}, ..., \mathbf{L}^{r_T}\right)\right) \\
&= \frac{1}{T}\left(\log_2 2^{n_s} + \sum_{\mathbf{l}^t \in \Lambda} \sum_{\mathbf{l}^{r_1} \in \Lambda} ... \sum_{\mathbf{l}^{r_T} \in \Lambda} p\left(\mathbf{l}^{r_1}, ..., \mathbf{l}^{r_T} \mid \mathbf{l}^t\right) p\left(\mathbf{l}^t\right) \log_2 \frac{p\left(\mathbf{l}^t, \mathbf{l}^{r_1}, ..., \mathbf{l}^{r_T}\right)}{p\left(\mathbf{l}^{r_1}, ..., \mathbf{l}^{r_T}\right)}\right) \\
&= \frac{1}{T}\left(n_s + \sum_{\mathbf{l}^t \in \Lambda} \sum_{\mathbf{l}^{r_1} \in \Lambda} ... \sum_{\mathbf{l}^{r_T} \in \Lambda} p\left(\mathbf{l}^{r_1} \mid \mathbf{l}^t\right) ... p\left(\mathbf{l}^{r_T} \mid \mathbf{l}^t\right) p\left(\mathbf{l}^t\right) \log_2 \frac{p\left(\mathbf{l}^{r_1}, ..., \mathbf{l}^{r_T} \mid \mathbf{l}^t\right) p\left(\mathbf{l}^t\right)}{\sum_{\mathbf{l}^t \in \Lambda} p\left(\mathbf{l}^{r_1}, ..., \mathbf{l}^{r_T} \mid \mathbf{l}^t\right) p\left(\mathbf{l}^t\right)}\right) \\
&= \frac{1}{T}\left(n_s + \sum_{\mathbf{l}^t \in \Lambda} \sum_{\mathbf{l}^{r_1} \in \Lambda} ... \sum_{\mathbf{l}^{r_T} \in \Lambda} p\left(\mathbf{l}^{r_1} \mid \mathbf{l}^t\right) ... p\left(\mathbf{l}^{r_T} \mid \mathbf{l}^t\right) \frac{1}{2^{n_s}} \log_2 \frac{p\left(\mathbf{l}^{r_1} \mid \mathbf{l}^t\right) ... p\left(\mathbf{l}^{r_T} \mid \mathbf{l}^t\right)}{\sum_{\mathbf{l}^t \in \Lambda} p\left(\mathbf{l}^{r_1} \mid \mathbf{l}^t\right) ... p\left(\mathbf{l}^{r_T} \mid \mathbf{l}^t\right)}\right)
\end{aligned}
$$

$$= \frac{1}{T}\left(n_s + \sum_{i=1}^{M}\sum_{j_1=1}^{M}\cdots\sum_{j_T=1}^{M} p\left(s_{j_1}^1 \mid s_i^1\right)\cdots p\left(s_{j_T}^T \mid s_i^1\right)\frac{1}{2^{n_s}}\log_2 \frac{p\left(s_{j_1}^1 \mid s_i^1\right)\cdots p\left(s_{j_T}^T \mid s_i^1\right)}{\sum_{i=1}^{M} p\left(s_{j_1}^1 \mid s_i^1\right)\cdots p\left(s_{j_T}^T \mid s_i^1\right)} \right)$$

$$= \frac{1}{T}\left(n_s + \sum_{i=1}^{M}\sum_{j_1=1}^{M}\cdots\sum_{j_T=1}^{M} q_{ij_1}q_{ij_2}\cdots q_{ij_T}\frac{1}{2^{n_s}}\log_2 \frac{q_{ij_1}q_{ij_2}\cdots q_{ij_T}}{\sum_{i=1}^{M} q_{ij_1}q_{ij_2}\cdots q_{ij_T}} \right) \tag{8.1}$$

Fig 8.1.1 shows some examples of the virtual channel capacity and the real channel capacity.

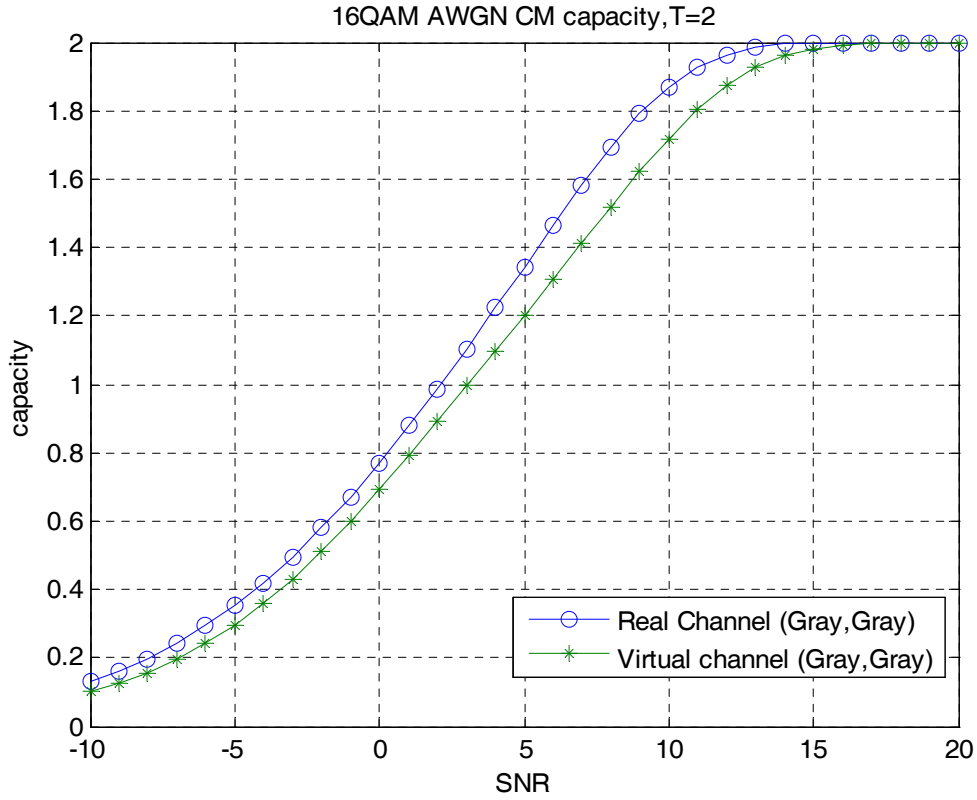

Fig 8.1.1 CM capacity of real channel and virtual channel (T=2)

As shown in Fig 8.1.1, the capacity of the virtual channel is always less than the real channel since some information is lost in the process of making hard-decision. Thus the SNR of the virtual channel should be raised so that its capacity will be the same as the real one. Since the capacity $C_v(\gamma_v)$ is a monotonically increasing function of

SNR, we can define its inverse function and find the corresponding SNR $\gamma_v = C_v^{-1}\left(C = C_r\left(\gamma_r\right)\right)$ to compute the transmission matrix $\mathbf{Q}$.

8.1.2 Extrinsic Channel

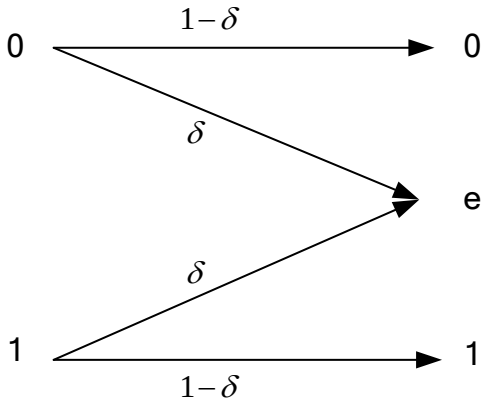

Fig 8.1.2 binary erasure channel

The a priori input to the demapper is modeled as the output of the BEC channel with erasure probability δ. Let $\Lambda_{[k]}^a$ be the set of all the possible values of the a priori information with the k-th bit prior omitted and $\mathbf{l}_{[k]}^a \in \Lambda_{[k]}^a$. The cardinality of $\Lambda_{[k]}^a$ for BEC is $|\Lambda_{[k]}^a| = 3^{n_s}$. The transition probability is

$$p\left(l_i^a \mid l_i^t\right) = \begin{cases} \delta & l_i^a = e \\ 1-\delta & l_i^t = l_i^a, l_i^a \neq \mathrm{e} \\ 0 & l_i^t \neq l_i^a, l_i^a \neq \mathrm{e} \end{cases} \tag{8.2}$$

The mutual information between the binary input and the erasure channel output is then given by $I_A = 1-\delta$. Thus, for a specific value of I_A, the corresponding value of δ can be determined.

8.1.3 Closed Form Demapper Transfer Function

With the help of the simplified hard decision virtual channel and the BEC extrinsic channel, we can obtain a closed form demapper transfer curve. The derivation is quite similar to section 5.2.1 except that the original integration is replaced by discrete summation.

$$
\begin{aligned}
I_E &= \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; L^e\right) \\
&= \frac{1}{n_s}\sum_{k=1}^{n_s} I\left(L_k^t; Y_1, Y_2, ..., Y_T, \mathbf{L}_{[k]}^a\right) \\
&= 1 - \frac{1}{n_s}\sum_{k=1}^{n_s} \sum_{l_k^t = b \in \{0,1\}} \sum_{y_1 \in \Lambda} \cdots \sum_{y_T \in \Lambda} \sum_{\mathbf{l}_{[k]}^a \in \Lambda_{[k]}^a} \frac{1}{n_s} \sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) \cdots P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l_i^a \mid l_i^t\right) \cdot \\
&\cdot \log_2 \frac{\sum_{\mathbf{l}^t = \lambda_i \in \Lambda} p\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_i\right)\right) \cdots p\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_i\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l_i^a \mid l_i^t\right)}{\sum_{\mathbf{l}^t = \lambda_j \in \Lambda_k^b} P\left(y_1 \mid \mu_1\left(\mathbf{l}^t = \lambda_j\right)\right) \cdots P\left(y_T \mid \mu_T\left(\mathbf{l}^t = \lambda_j\right)\right) \prod_{i=1, i\neq k}^{n_s} p\left(l_i^a \mid l_i^t\right)}
\end{aligned}
\tag{8.3}
$$

Here we plot some example of the virtual and real demapper transfer curve to confirm the accuracy of the virtual channel simplification.

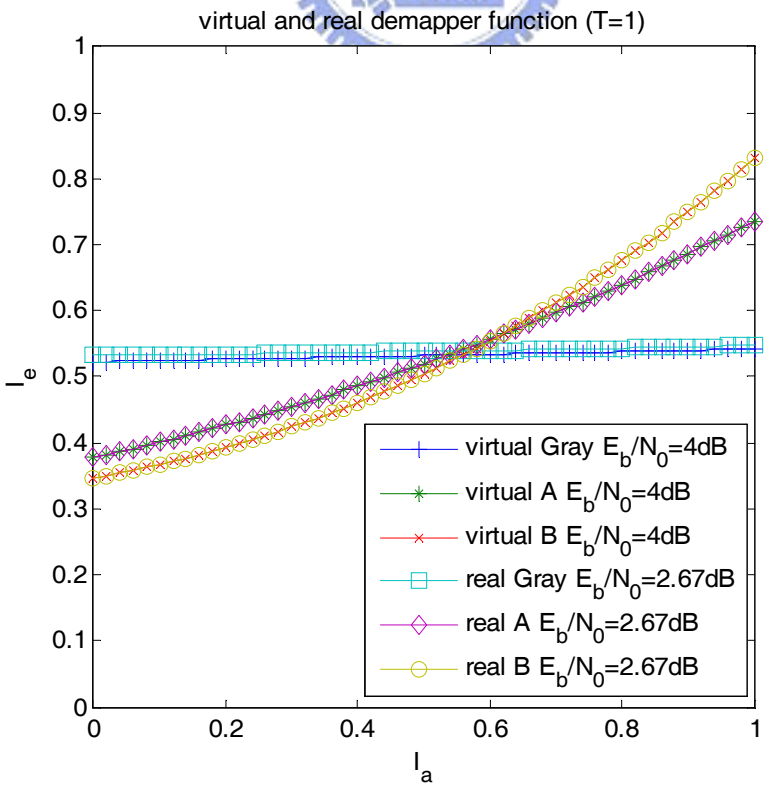

Fig 8.1.3.1 Various virtual and real channel demapper (T=1)

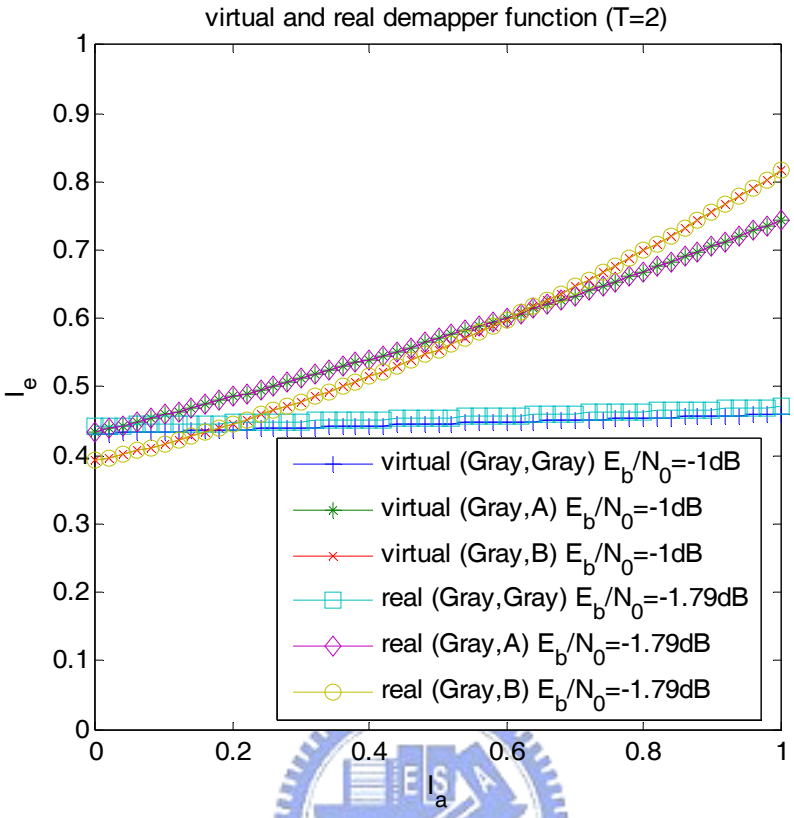

Fig 8.1.3.2 Various virtual and real channel demapper (T=2)

8.2 Genetic Algorithms

8.2.1 Introduction

We propose using genetic algorithm to find the optimal mapping. It is proposed in light of its robustness over a variety of optimization problems. Genetic algorithms were first developed by John Holland in the late 1960s and early 1970s. Since its conception, genetic algorithms have been used widely as a tool in computer programming and artificial intelligence, optimization, neutral training, and many other areas.

Genetic algorithms are search algorithms base on the mechanics of natural selection and natural genetics. Through the mechanics of selection, the fittest ones survive while

the worse ones die out. By crossover and mutation, useful parts are exchanged and occasionally new parts are tried for good measure. They efficiently exploit historical information to speculate on new search points for better performance.

Genetic algorithms surpass other conventional search methods by its robustness. They are different in four ways:

1. GAs work with a coding of the parameter set, not the parameter themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (object function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

GAs operate on the coding of the parameter set to exploit coding similarities, as a result, they are largely unconstrained by the limitations of other methods (continuity, derivative existence, single peak , and so on).

Many optimization methods suffer from the dangerous of location false peaks in many-peaked search spaces since only single point is tried for optimization. Instead, GAs work with many points simultaneously in the search space. Thus the probability of finding a false peak is reduced by climbing many peaks in parallel.

No need for other auxiliary information except the objective function values makes GAs applicable for wide variety of problems. After all, every search problem has a metric to optimize; however, different search problems have vastly different forms of auxiliary information. Thus the sole dependence on the objective function values makes GAs robust.

Taken these four features together, genetic algorithms are robust and advantageous over other commonly used techniques.

8.2.2 General Procedure

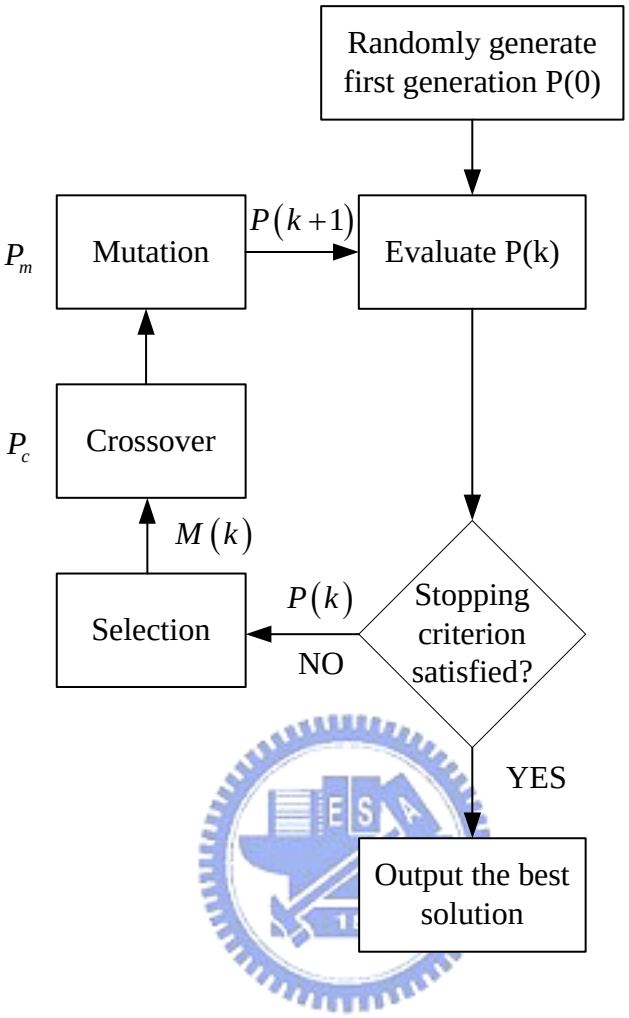

Fig 8.2.2 Flow chart for genetic algorithm

First we have to encode the points in the search space as strings of symbols. These strings are called chromosomes. Each symbol in a string is a gene. At the beginning of the genetic algorithm, we randomly generate a set of chromosomes $p(0)$ with size N in the search space. During the k-th iteration of the process, we evaluate the objective function for each chromosome in $p(k)$. If the stopping criterion is not satisfied, we continue to the selection process. At the selection process, each chromosome in $p(k)$ is selected according to its value in the objective function. The better the fitness measure in the objective function, the higher the probability of survival. After selection, a

mating pool $M(k)$ with the same size as the initial population is formed for the crossover operation. Each chromosome in the mating pool has a probability of P_c to be chosen for crossover. The crossover operation takes a pair of chromosomes, called the parents, from the mating pool and gives a pair of offspring chromosomes. The operation involves exchanging substrings of the two parent chromosomes in the hope of giving birth to a pair of better chromosomes. Next, we apply the mutation operation. The mutation takes each chromosome from the mating pool and randomly exchanges each symbol of the chromosome with a give probability p_m. Typically, the value of p_m is very small, so that only a few chromosomes will undergo a change due to mutation. After applying the crossover and mutation operation to the mating pool $M(k)$, we obtain the new population $p(k+1)$. These procedures of selection, crossover and mutation are applied iteratively until certain stopping criterion is reached. Here we stop the iteration process when the best-so-far chromosome does not change significantly form iteration to iteration. Note that throughout the entire process, the population size N is always kept the same.

8.2.3 Representation Scheme

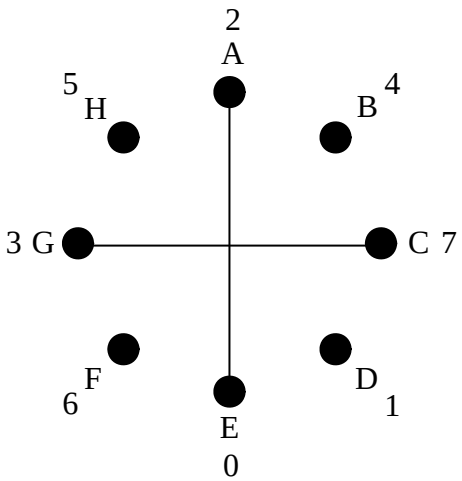

Fig 8.2.3 8 PSK constellation

Let the alphabets $A, B, C, \ldots$ denote the constellation points in the signal space. We encode the mapping as a string of decimal digits $L_1, L_2, \ldots, L_M$ which are the corresponding labels in the constellation points $A, B, C, \ldots.$. Take 8PSK mapping as an example, as shown in Fig 8.2.3, 2 4 7 1 0 6 3 5 is an encoding of this mapping.

8.2.4 Selection

To improve the convergence rate of the algorithm, we adopt a scheme called elitism. In the elitist strategy, we copy the best-so-far I chromosomes directly, and the rest N-I chromosomes are selected from the original population according to the roulette-wheel scheme. In the roulette-wheel scheme, chromosomes are selected into mating pool with probability proportional to their fitness. Let $f(x_i)$ be the objective function and $x_i \; i = 1, 2, \ldots., N$ be the i-th chromosomes. A chromosome x_i will be selected with probability equal to $\frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)}$. Thus the better the fitness measure the higher the chance of this chromosome will be selected. The resulting population after selection is called mating pool for the following crossover operation.

8.2.5 Crossover

Apart from the best-so-far I chromosomes which are preserved according to elitism, we randomly choose two parent from the mating pool and perform crossover operation with crossover probability Pc. If there is no crossover, we just copy parents directly. For crossover operation, we first generate two distinct uniformly distributed random numbers a, b with values range from 0 to L-1, where L is the number of genes in a chromosome. In our example, L is just equal to the number of labels in a mapping. On pair of parent can give birth to two children. For the first child, the labels in parent 1 in the position from a to b is copied directly to the first child. For the rest labels not

assigned to the first child, they are assigned with the same order as in the second parent. In this way, the labels from a to b in parent 1 is preserved in their absolute position while the rest labels from parents 2 is preserved only in their relative position. For the second child, the role of parent 1 and parent 2 are just exchanged.

As shown is Fig 8.2.5, the two random number a and b is 2 and 5. Labels from position 2 to 5 in parent 1 is copied directly to child 1, while the rest labels 0,6,1,7 are assigned according to their orders in parent 2. Child 2 is created in the same way, except that labels in position 2 to 5 is copied from parent 1 and the rest are arranged in the same order as in parent 1.

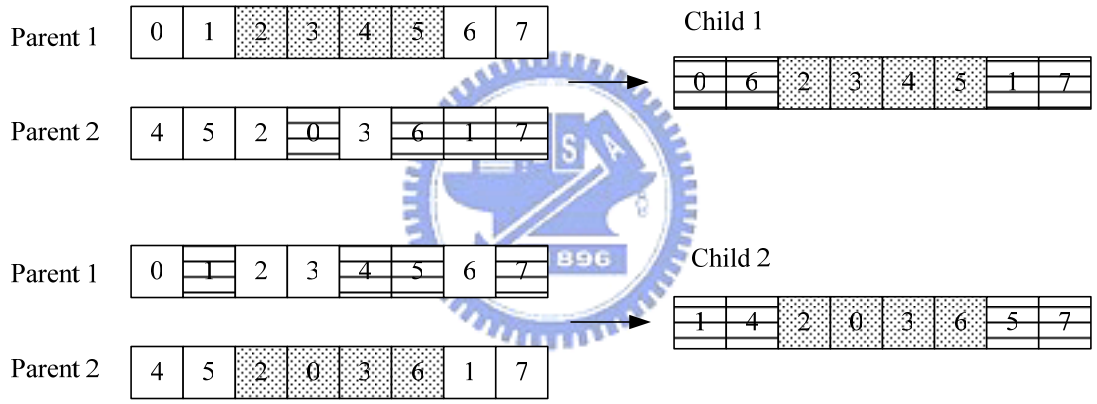

Fig 8.2.5 crossover operation

8.2.6 Mutation

Except for the best-so-far I chromosomes, each chromosome has a probability of p_m to mutate. The mutation operation is quite simple; we just randomly pick two labels in a chromosome and swap them. Fig 8.2.6 shows an example, the two positions 4 and 7 are picked and the corresponding labels 0 and 1 are just swapped.

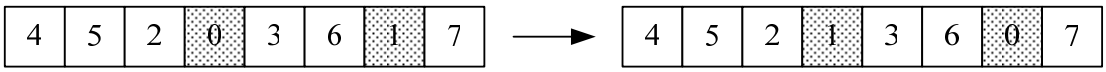

Fig. 8.2.6 mutation operation

8.2.7 Suggested GA Parameters

There many parameters that should be assigned properly for GA to work well. Parameters such as population size, elite number, crossover probability and mutation probability are important for a better convergence behavior. Here we have experimented some different values and suggested a set of proper parameter values.

In our experiments, we use Gray as the first transmission mapping and design the second mapping for BICM. For each parameter test, the other parameters are fixed to isolate the effects of the testing parameter. The best values in the objective function are plotted against iteration number to observe the convergence behavior.

8.2.7.1 Population Size

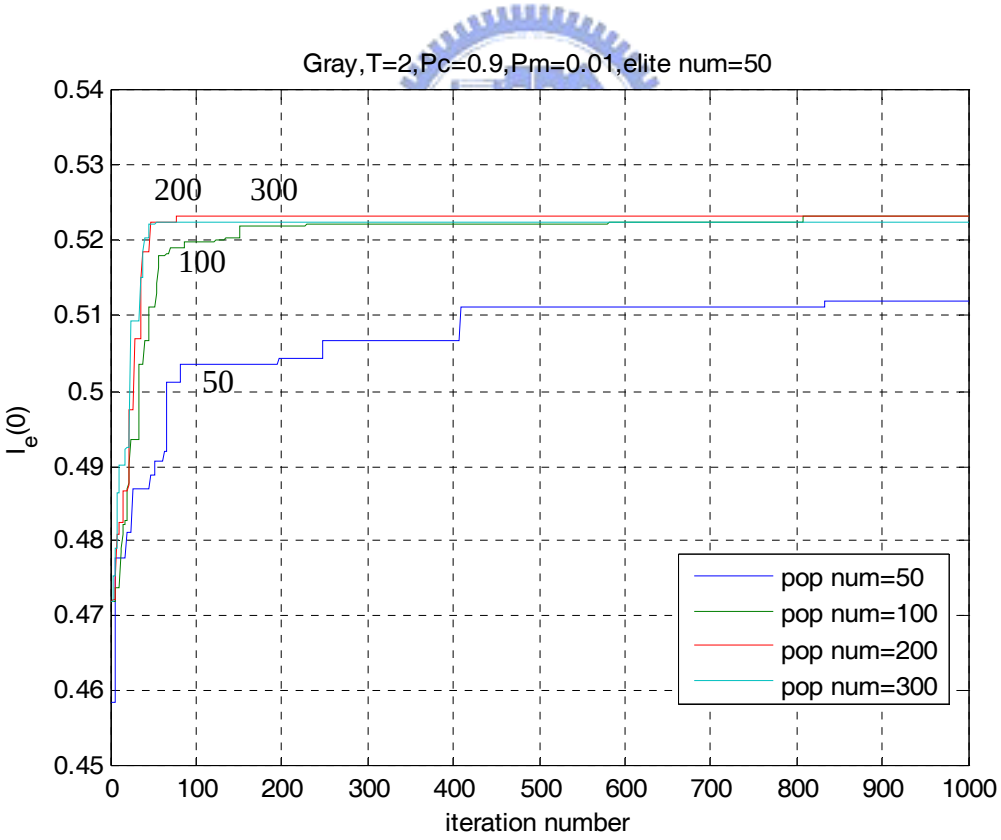

Fig 8.2.7.1 population size comparison

One of the features in GA is that it searches with multiple points in the search space instead of moving from one single point to another. This feature makes it robust in

problems with many-peaked search spaces since the possibility of locating false peak is reduced by climbing many hills simultaneously. On the other way, tying many possible solutions at the same time also increases the chance of finding the best solution. Therefore, it seems that the larger the population size the higher the rate of convergence. As Fig 8.2.7.1 indicates, increasing the population size from 50 to 200 accelerates the convergence rate. However, increasing the population size from 200 to 300 exhibits similar behavior. This suggests that there is a threshold such that no further improvements will be gained after passing this threshold. Thus we suggest choosing population size to 100 as a proper value since the convergence rate is similar to the one with size 200 and 300.

8.2.7.2 Elite Number

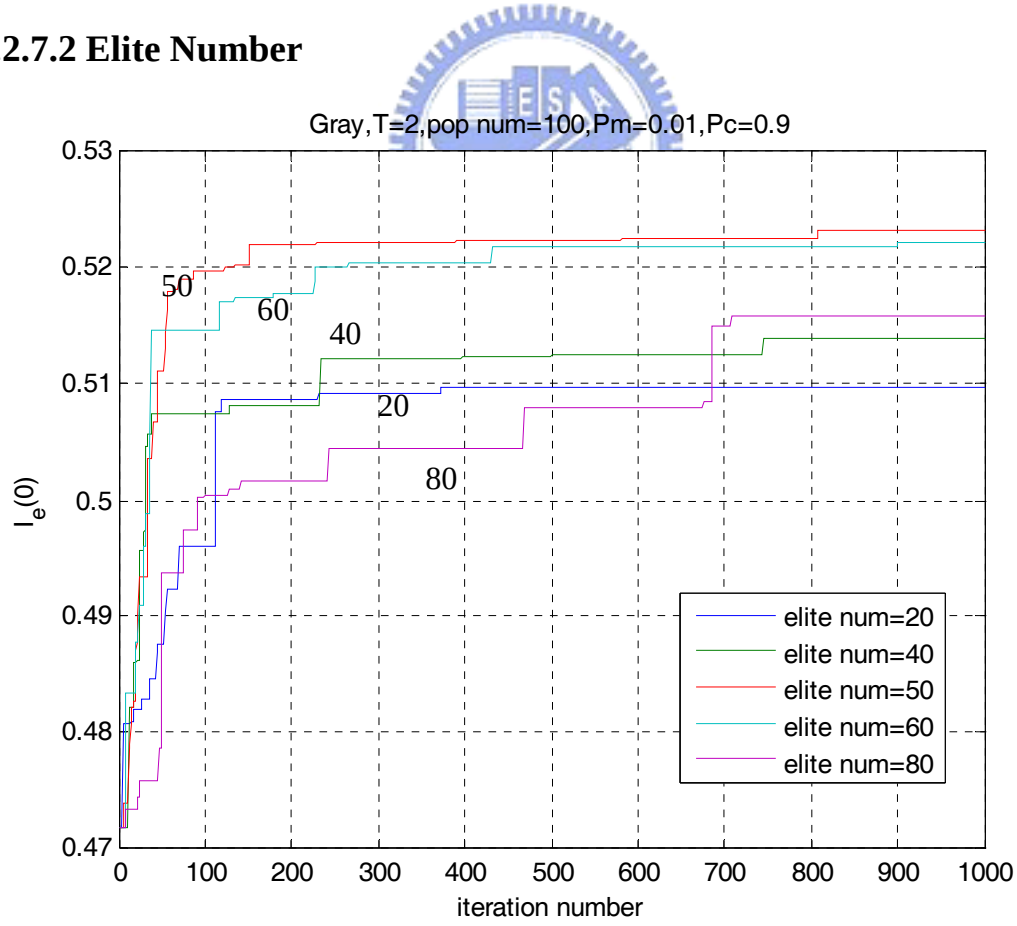

Fig 8.2.7.2 elite number comparison

Some researchers in the GA field have point out that by reproducing always the best

individuals in the current population do leads to a faster convergence of the algorithm. However, there is also some dangerous of “premature convergence”, making the algorithm converges to a local solution. Since the initial population is generated randomly due to the blindness of the location of the global optimum, it will happen that a significant part of the individuals represents points of the search space which are far from the global optimum Thus these best individuals may only represent the regions where local optimum is located. This suggests the number of elites that will be preserved should be chosen properly to avoid premature convergence.

Fig 8.2.7.2 shows the effects of different elite number. The increasing of the elite number accelerates the convergence rate can be seen for elite number ranging from 20 to 50. However, increasing the elite number to 80 leads to premature convergence. Thus we suggest elite number 50 as a proper value.

8.2.7.3 Crossover Probability

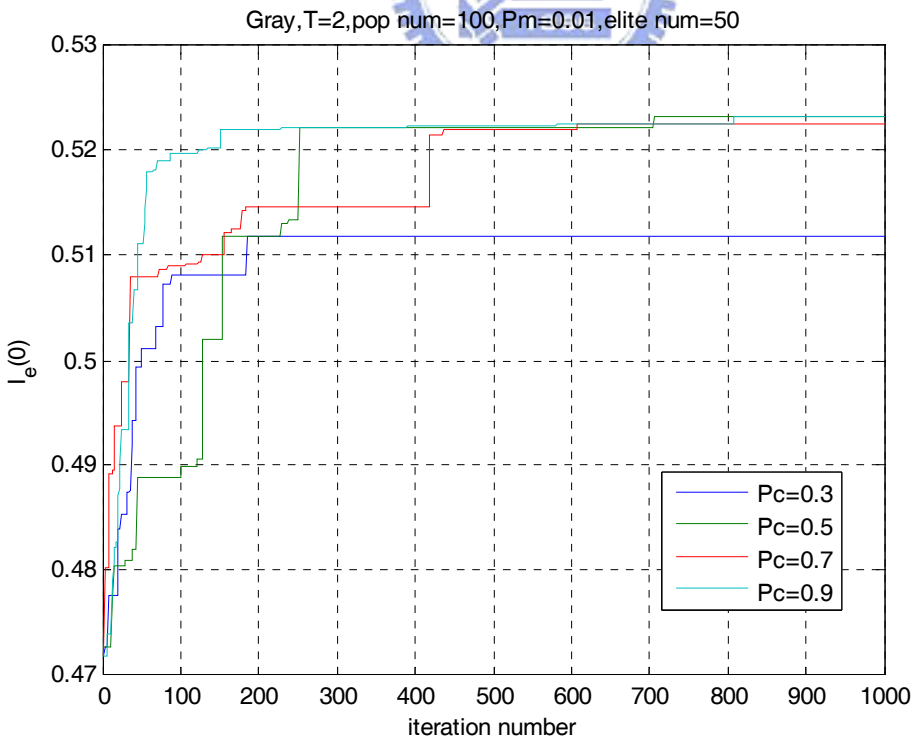

Fig 8.2.7.3 crossover probability comparison

Crossover is the main operation to explore the search space. Crossover combines parts of the genes from the mating parents in the hope to produce a child which inherits both the good traits from parents. Generally, the crossover probability should be large for the whole population to evolve. If the crossover probability is too small, the population almost stays the same and no new solutions are explored. Fig 8.2.7.3 indicates that higher crossover probability contributes to higher convergence. Thus crossover probability 0.9 is recommended for better convergence rate.

8.2.7.4 Mutation Probability

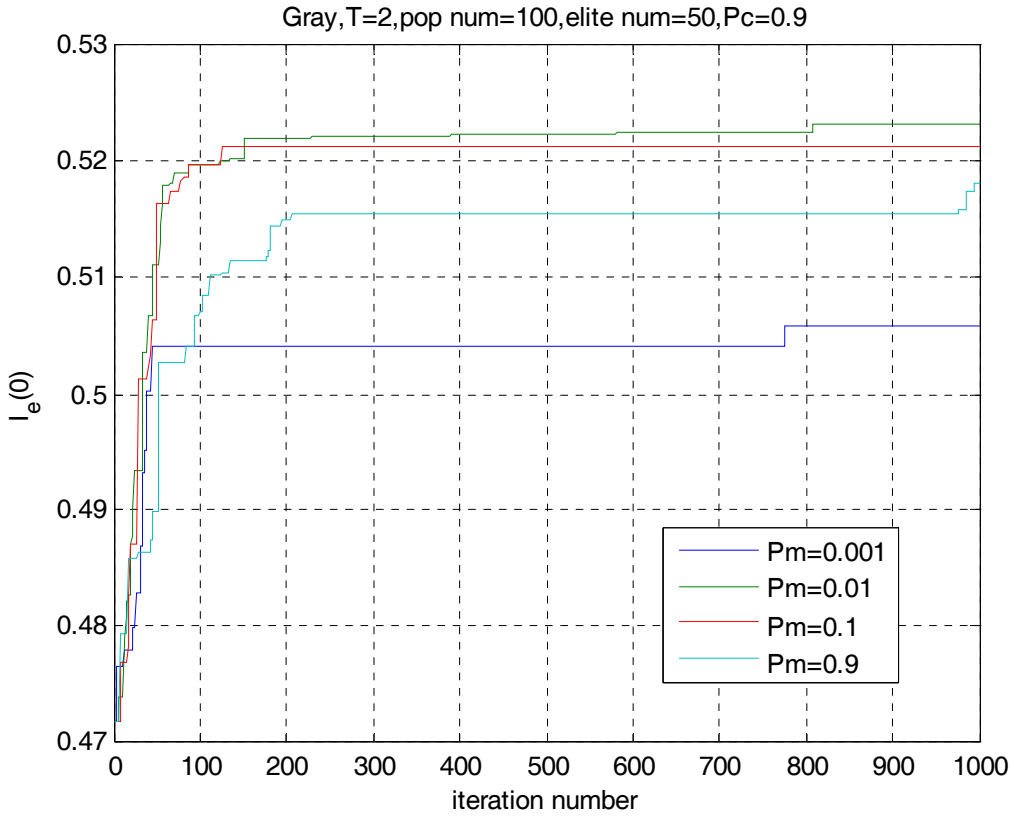

Fig 8.2.7.4 mutation probability comparison

Although the crossover operator is a very potent means of exploring search spaces, it does have a disadvantage. Ideally, good genes should be preserved and be combined with the other good genes to form a new chromosome. However, as the random natural of the crossover operator, it is possible that the good genes will be eliminated after the

crossover operation. Since there is no way that the crossover operator can bring back the lost genes, the mutation operator is created to reintroducing those missing genes back into the genetic pool. Since the operator proceeds by performing a random modification on the individual, the mutation probability should be small to avoid ruining the evolution mechanism introduced by crossover. As shown in Fig 8.2.7.4, too small or large mutation probability results in premature convergence, hence we advise using mutation probability 0.01.

8.2.7.5 Summary of the Suggested GA Parameters

Here we summarized the suggested GA parameters in the following.

Population size	Elite number	Crossover probability	Mutation probability
100	50	0.9	0.01

Chapter 9: Mapping Search Results

9.1 Mappings for BICM

In [3], it is showed that the BICM capacity is upper bounded by the CM capacity, and the BICM capacity of Gray mapping is quite close to its corresponding CM capacity over the entire SNR region. Our search results also confirmed that Gray mapping is the optimal one. Hence we adopt Gray mapping as the first transmission mapping. For the second transmission mappings, we searched the label from $\frac{E_s}{N_o} = -4dB$ (virtual channel SNR) to $\frac{E_s}{N_o} = 12dB$, and the increment is 1 dB. The results are presented in the following.

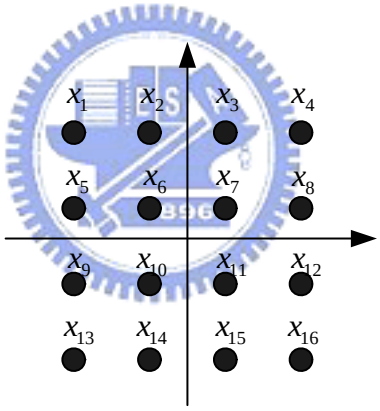

Fig 9.1.1 16QAM constellation

Table 9.1.1 BICM mappings (T=2):

SNR (virtual)	SNR (real)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
-4	-4.8	13	5	15	7	9	1	11	3	12	4	14	6	8	0	10	2
-3	-3.8	13	5	15	7	9	1	11	3	12	4	14	6	8	0	10	2
-2	-2.7	13	5	15	7	9	1	11	3	12	4	14	6	8	0	10	2
-1	-1.7	10	14	3	11	2	6	7	15	8	12	13	9	0	4	5	1
0	-0.7	2	6	11	3	10	14	15	7	8	12	13	9	0	4	5	1
1	0.3	13	5	3	7	9	1	11	15	12	4	14	6	8	0	10	2
2	1.2	13	5	3	7	9	1	11	15	12	4	14	6	8	0	10	2
3	2.2	15	7	14	10	11	3	6	2	13	9	12	8	5	1	4	0
4	3.1	4	0	5	1	12	8	9	13	6	10	7	3	14	2	15	11
5	4	13	9	12	8	1	5	4	0	15	11	6	10	7	3	14	2
6	4.8	8	12	2	10	0	4	14	6	5	11	3	15	1	13	7	9
7	5.7	13	9	5	8	6	3	15	2	12	11	7	10	4	1	14	0
8	6.6	8	7	10	0	3	13	2	15	12	9	5	11	6	14	1	4
9	7.5	13	8	15	2	9	7	3	10	6	14	11	5	12	1	4	0
10	8.6	12	9	6	8	2	14	3	13	10	5	11	7	4	1	15	0
11	9.6	5	8	15	0	9	3	4	11	13	7	14	6	2	10	1	12
12	10.7	4	1	14	2	11	15	5	9	7	8	3	12	0	13	10	6

To compare each mapping's relative performance, we also plot the BICM capacity of each mapping. In our legend, the real channel SNR is labeled on the left, and the virtual channel SNR is labeled on the right.

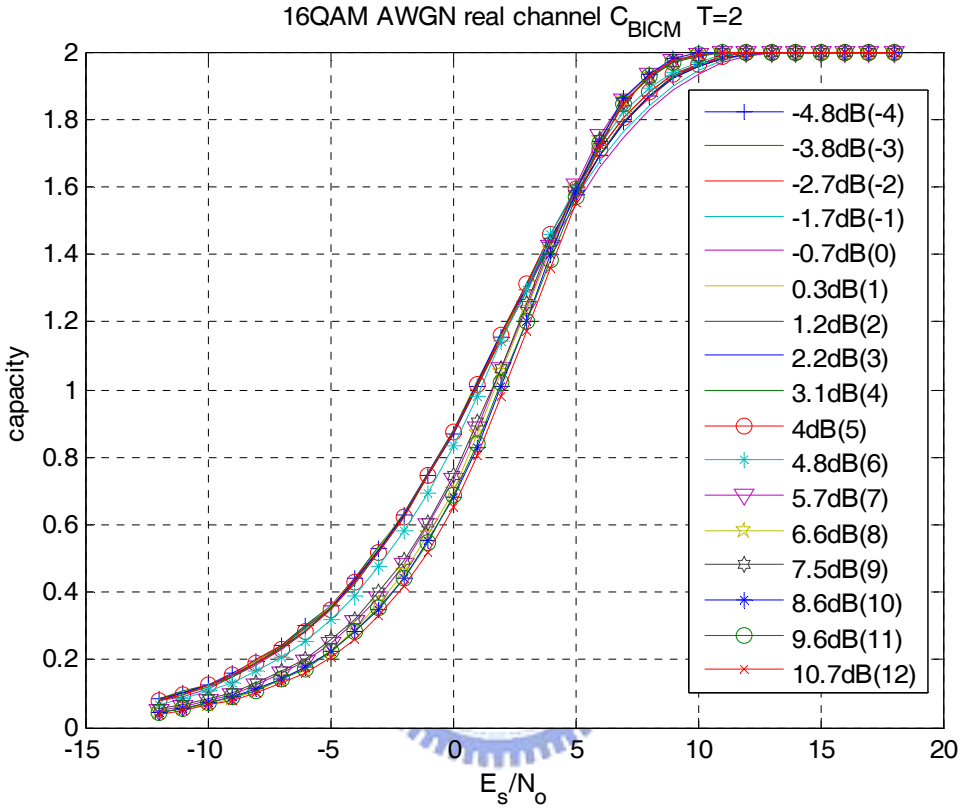

Fig 9.1.2 BICM capacity of mappings designed for different SNR (T=2)

As can be predicted, the mappings designed for low SNR achieves high capacity at low SNR; however, the capacity decrease at high SNR, and vice versa.

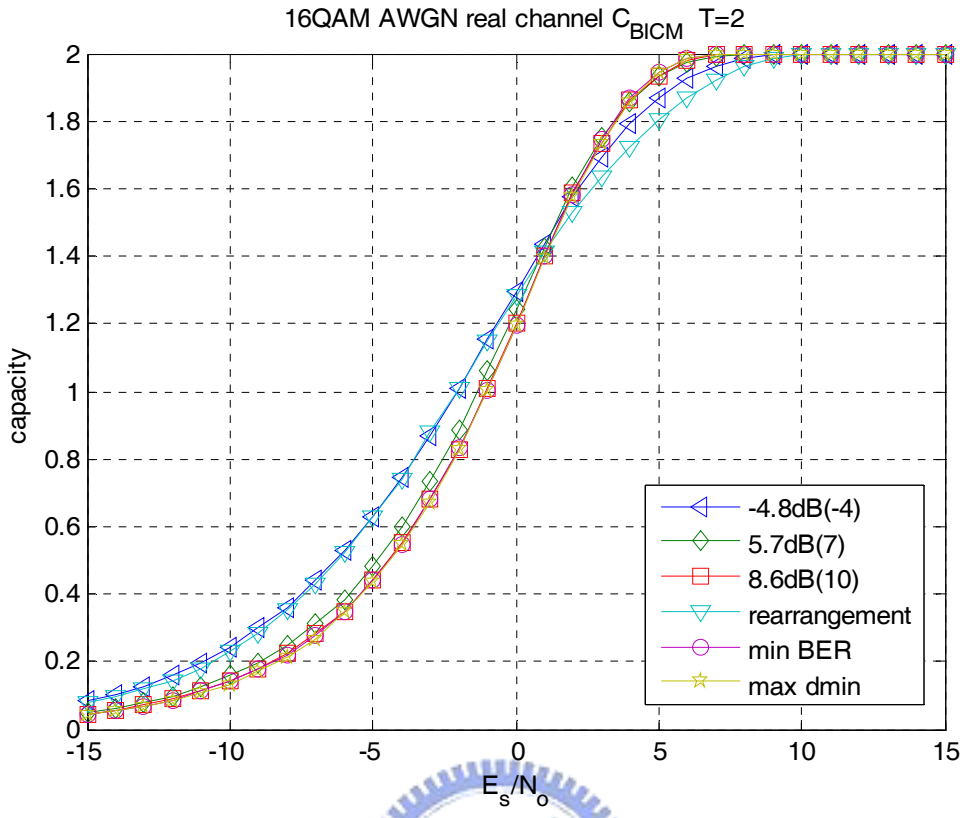

Fig 9.1.3 Comparison of BICM capacity of different mapping schemes (T=2)

We also compared some of our designed mappings with the previous mappings in the literature. It is shown that the constellation rearrangement has similar behavior as the mapping designed for low SNR, hence the capacity decreases at high SNR. Mapping designed to minimize the BER upper bound or maximize the minimum Euclidean distance have similar capacity as mappings designed for high SNR. Thus they are suitable for the applications that operate at high SNR and not suitable at low SNR.

Table 9.1.2 BICM mappings (T=3)

SNR (virtual)	SNR (real)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
-3	-3.8	3	1	5	0	9	7	2	4	11	13	8	6	15	10	14	12
-2	-2.7	5	0	4	12	1	13	8	6	9	7	2	14	3	11	15	10
-1	-1.7	14	4	12	13	6	0	8	9	2	10	11	5	15	7	3	1
0	-0.7	12	14	6	4	8	10	7	0	9	15	3	2	13	11	1	5
1	0.3	1	15	7	14	5	9	11	6	4	13	3	12	0	2	8	10
2	1.2	1	0	5	4	15	7	11	6	8	13	3	12	9	10	2	14
3	2.2	12	6	4	14	13	10	11	3	8	7	5	2	9	15	0	1
4	3.1	0	2	15	8	10	1	11	12	7	3	14	13	4	5	6	9
5	4	0	1	10	5	13	2	8	9	14	7	15	3	12	6	11	4
6	4.8	12	3	5	14	11	7	8	6	15	2	10	1	0	13	9	4
7	5.7	9	0	15	6	4	13	11	2	8	7	3	10	1	14	5	12
8	6.6	13	9	0	5	11	4	6	8	2	3	14	15	12	10	7	1
9	7.5	10	1	3	14	9	13	5	6	0	12	11	8	7	15	2	4

The behavior of BICM capacity for transmit 3 times is similar to the behavior of BICM apacity for transmit 2 times. Mappings that have high capacity at high SNR have low capacity at low SNR, and vice versa. For the comparison to other mappings, similar to transmit 2 times, the capacity of constellation rearrangement is high only at low SNR, and the capacity of MBER and MDMIN is only high at high SNR.

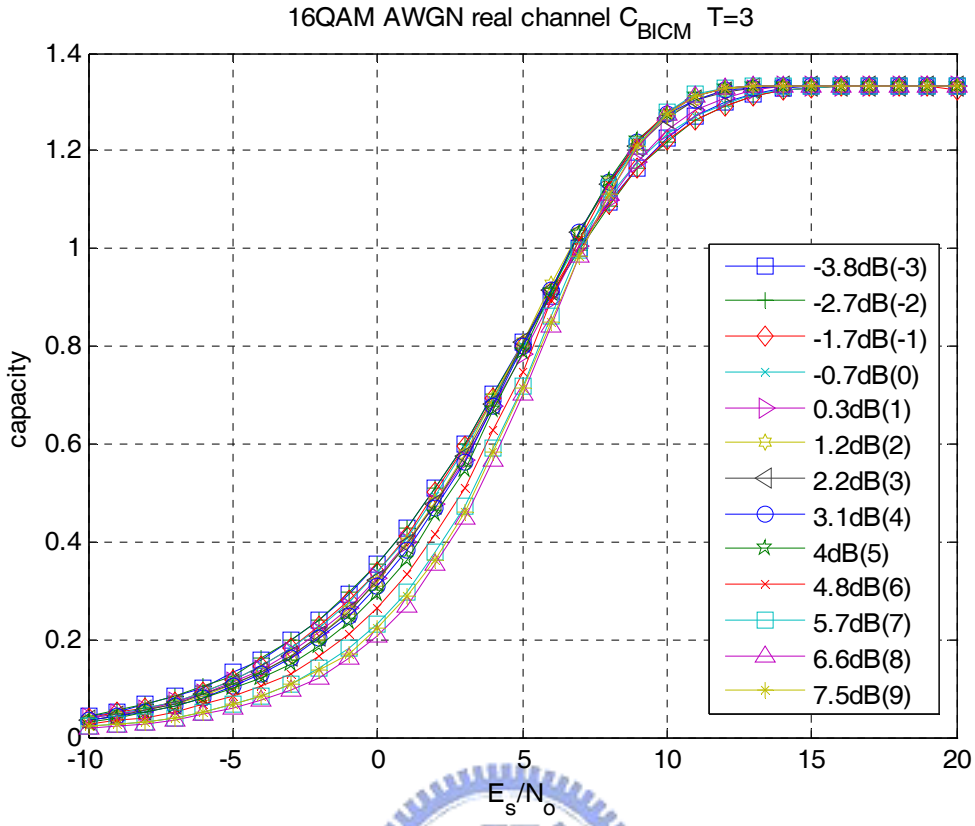

Fig 9.1.4 BICM capacity of mappings design for different SNR (T=3)

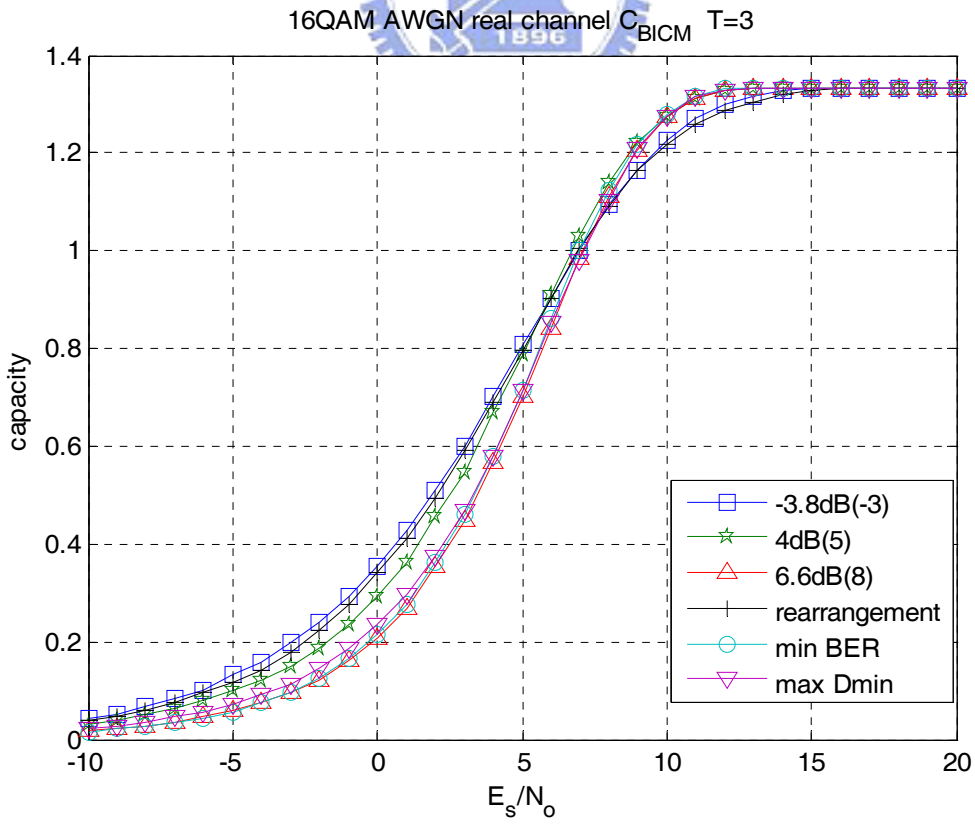

Fig 9.1.5 Comparison of BICM capacity of different mapping schemes (T=3)

9.2 Mappings for BICM-ID with infinite block length

Table 9.2.1 BICM-ID mappings for infinite block length (T=2, 1st)

Virtual SNR	Real SNR	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
0	-0.7	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
1	0.3	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
1.5	0.8	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
2	1.2	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
3	2.2	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
4	3.1	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
5	4	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
6	4.8	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
7	5.24	0	4	8	9	12	5	13	14	15	6	7	11	10	2	1	3
7.5	6	11	2	6	10	14	3	15	12	13	4	5	0	7	1	9	8
8	6.6	10	6	15	9	0	12	5	3	4	7	14	2	1	13	11	8
9	7.5	9	5	10	15	12	0	6	3	7	11	13	8	2	14	1	4
10	8.6	2	1	11	8	4	7	13	14	9	10	0	3	15	12	6	5
11	9.6	7	2	13	14	4	1	8	11	10	15	6	5	9	12	3	0

Table 9.2.2 BICM-ID mappings for infinite block length (T=2, 2st)

Virtual SNR	Real SNR	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
0	-0.7	0	8	1	9	4	12	5	13	2	10	3	11	6	14	7	15
1	0.3	0	13	9	12	4	11	6	2	5	15	3	14	1	10	7	8
1.5	0.8	4	8	13	6	14	2	11	1	7	3	10	12	9	15	5	0
2	1.2	6	5	3	8	0	15	11	14	12	9	2	4	10	7	1	3
3	2.2	9	12	0	5	10	15	3	6	2	7	11	14	4	1	13	8
4	3.1	0	10	12	6	9	3	15	5	13	7	11	1	14	4	2	8
5	4	1	8	13	14	4	7	11	6	2	15	3	0	10	9	12	5
6	4.8	0	6	14	13	3	5	11	8	12	15	2	1	9	10	7	4
7	5.24	13	4	7	2	3	8	14	11	6	10	15	1	5	12	0	9
7.5	6	3	0	9	5	6	10	12	15	13	14	7	2	8	11	1	4
8	6.6	0	6	5	12	11	8	10	15	2	13	9	3	7	14	1	4
9	7.5	6	5	12	11	3	15	2	1	10	9	4	8	0	7	14	13
10	8.6	7	11	12	0	2	8	5	6	1	14	15	9	13	4	3	10
11	9	1	12	13	6	2	11	7	10	8	14	9	0	15	4	5	3

Since the block length is assumed to be infinite, the trajectory will get through even when only a very small tunnel is opened. Hence no margin is preserved between the demapper and the decoder transfer curve.

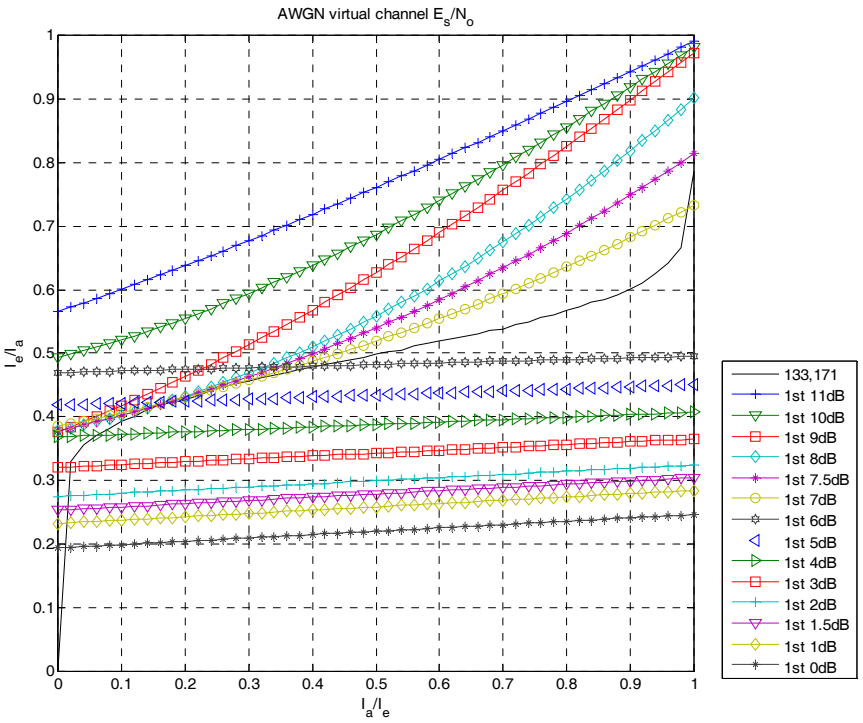

Fig 9.2.1 demapper transfer function for inifinite block length (T=1)

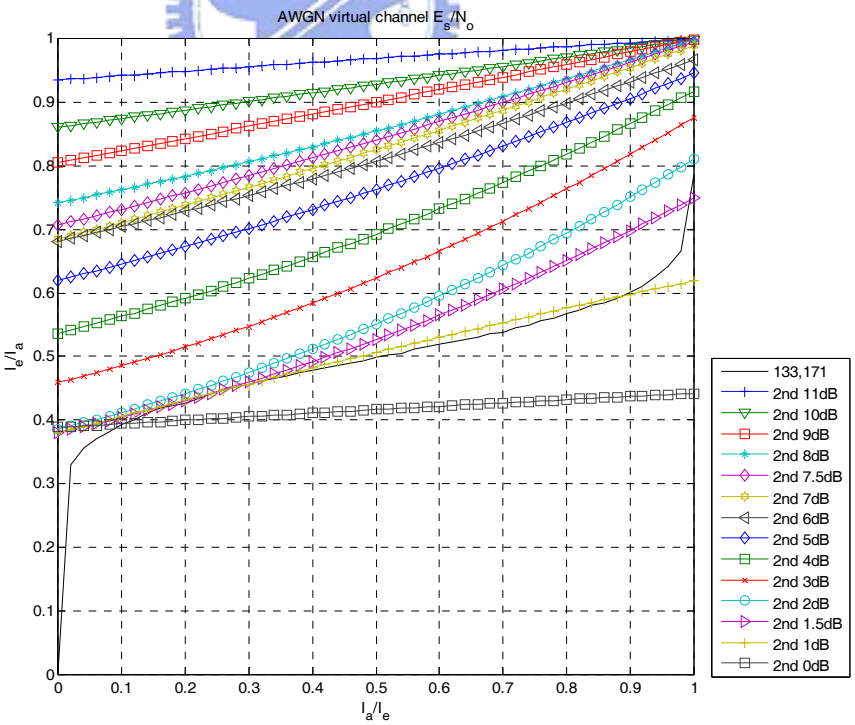

Fig 9.2.2 demapper transfer function for inifinite block length (T=2)

9.3 Mappings for BICM-ID with finite block length

Transmit one time

Here we choose the block length to be 2500 and margin of two times the standard deviation is preserved in both the demapper and the decoder to allow the tunnel to have about a chance of 97.7% to open. Therefore $\delta = 2 \cdot std\_demapper + 2 \cdot std\_decoder$, and σ is properly chosen to be $\sigma = 0.1$, where σ, δ is the same as that defined in chapter 7.

Table 9.3.1 BICM-ID mappings for finite block length (T=1)

Virtual SNR	Real SNR	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
7	5.24	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7
7.5	6	12	14	1	9	13	0	3	8	4	5	2	10	6	7	15	11
8	6.6	9	12	8	13	5	4	1	10	0	2	15	11	7	6	14	3

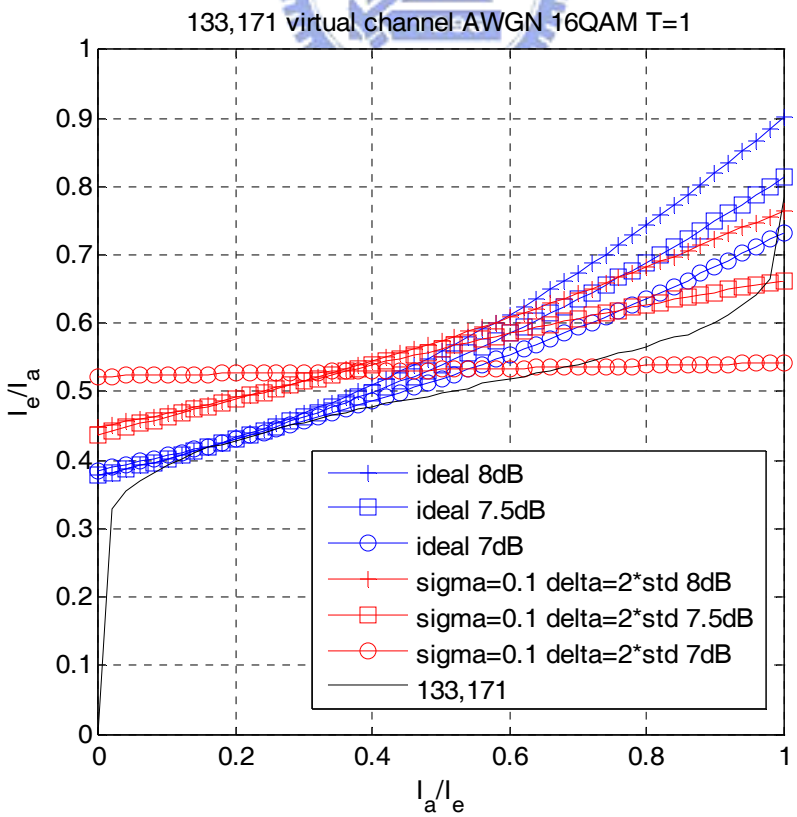

Fig 9.3.1 demapper transfer curve for finite block length (T=1)

Transmit 2 times

Table 9.3.2 BICM-ID mappings for finite block length (T=2,1st)

Virtual SNR	Real SNR	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
1.5	0.8	11	9	1	3	10	8	0	2	14	12	4	6	15	13	5	7

Table 9.3.3 BICM-ID mappings for finite block length (T=2,2nd)

Virtual SNR	Real SNR	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
1.5	0.8	12	6	9	8	10	3	15	7	0	14	4	1	5	2	11	13

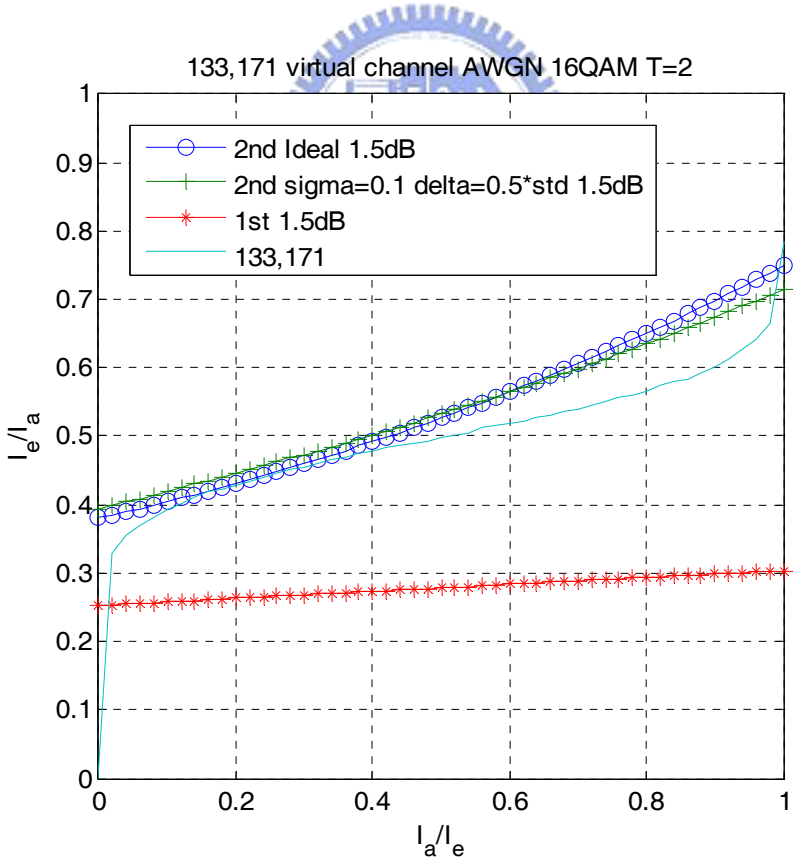

Fig 9.3.2 demapper transfer curve for finite block length (T=2)

Chapter 10: Simulation Results

Since a HARQ system is packet oriented, one of the performances metric is its throughput. Throughput is defined as the number of successful received information bits per second. Suppose n is the number of coded bits in a packet, r is the code rate and R is the packet rate (packet/sec). Define N_t to be the total number of transmitted packets and N_r to be the total number of successful received packets. Then the throughput is defined as $\frac{N_r \cdot n \cdot r}{N_t / R} = \frac{N_r}{N_t} \cdot n \cdot r \cdot R$. Since n, r and R is fixed for a given MCS, we define $\frac{N_r}{N_t}$ as its throughput performance metric.

10.1 BICM

The simulation parameter is as follows

Code	**Code rate**	**Block length**	**Modulation**	**Channel**
Convolutional code (133,171)	$\frac{1}{3}$	14400	16QAM	AWGN
	$\frac{1}{2}$	9600		
	$\frac{3}{4}$	6400		
Turbo code (13,15)	$\frac{1}{3}$	15012		
	$\frac{1}{2}$	10012		
	$\frac{3}{4}$	6680		

To verify whether the throughput performance of each mapping is the same as suggested by the BICM capacity, we simulated the throughput of each mapping designed for a specific SNR. We are interested in those SNR regions which correspond to throughput range from 0 to 1. Capacity approaching turbo code and commonly used convolutional code is simulated. Since the operating SNR regions are different, we also experiment codes with different code rate obtained from puncturing the same mother code

Comparing the relative throughput performance of each mapping, the results match quite well with the prediction from the BICM capacity. Each mapping performs best or at least not worse than others at their designed SNR. Also, a mapping that has high throughput at high SNR does not necessary has high throughput at low SNR, and vice versa. This confirms the idea that mappings should be adaptive to SNR.

Since different codes with different code rate have different operating SNR region, mapping should be designed specifically for different code. The simulation results show that different codes have their different best mapping scheme. Furthermore, although theoretically mapping scheme should be adaptive for different SNR, the simulation results shows that it is possible to design a single mapping scheme that is suitable for the whole interested SNR region. Comparing with the best throughput contour, mappings that are designed for the SNR region where throughput is 0 to 0.5 usually perform well at the whole SNR range. Hence this suggests that one mapping scheme can be designed for a specific MCS (modulation coding scheme).

We also compared the available mapping schemes in the literature. As can be observed, constellation rearrangement achieves high throughput for codes with code rate $\frac{1}{3}$ when transmit two times (Fig 10.1.1.2 and Fig 10.1.2.2). However, when

applying codes with higher code rate such as rate $\frac{1}{2}$ (Fig 10.1.1.4 and Fig 10.1.2.4) and rate $\frac{3}{4}$ (Fig 10.1.1.6 and Fig 10.1.2.6), the throughputs fall below the best throughput contour. The performance degradation is even more prominent when the rate is raised to $\frac{3}{4}$. For transmitting three times, constellation rearrangement performs poor even for rate $\frac{1}{3}$ (Fig. 10.1.1.8). Therefore the behavior of constellation rearrangement is quite similar to the mappings designed for low SNR and is only suitable for certain SNR region.

For the other two mapping schemes MBER (minimize the BER upper bound) and MDMIN (maximize the minimum Euclidean distance), they are both similar to the behavior of mappings designed for high SNR. The throughputs are far below best contour at code rate $\frac{1}{3}$ (Fig 10.1.1.2 and Fig 10.1.2.2). The throughput gaps between MBER (or MDMIN) and the best contour increases with the increasing code rate. However, they are close to optimal only when rate $\frac{3}{4}$ convolutional code is applied (in our case). Hence MBER and MDMIN are only suitable for the application of high code rate that operates at high SNR region.

Note that in the legend in the following, the real channel SNR (E_s / N_o) is labeled first and the virtual channel SNR is the next. .

10.1.1 Turbo Codes

Transmit 2 times, Code Rate 1/3

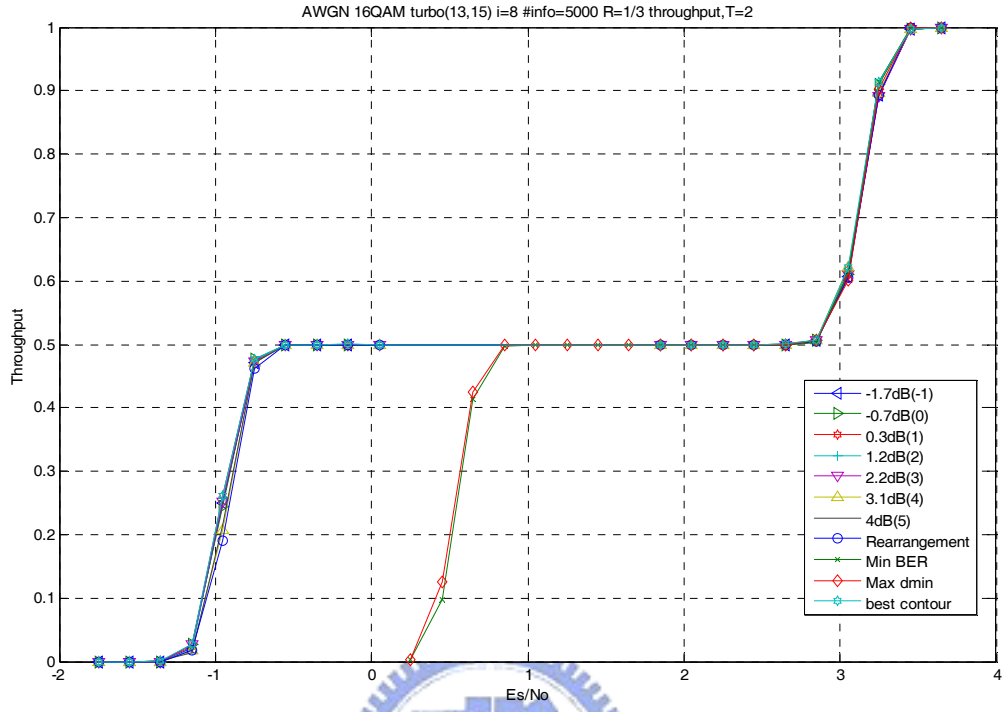

Fig 10.1.1.1. Throughput of turbo code with code rate=$\frac{1}{3}$, T=2

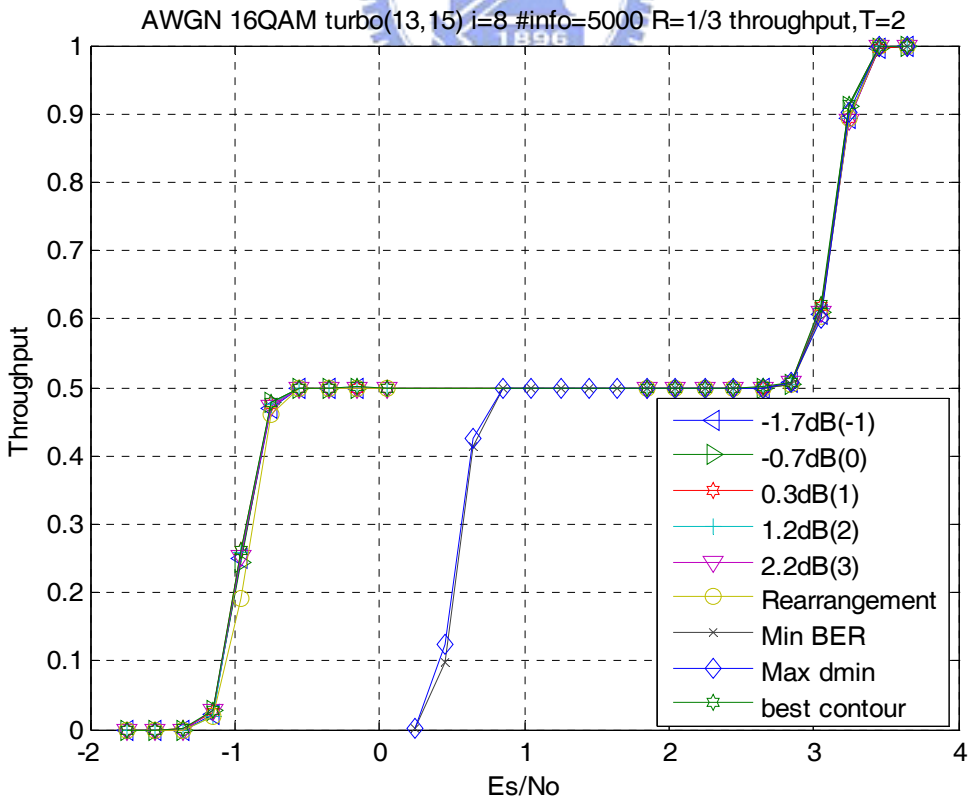

Fig 10.1.1.2 Throughput of turbo code with code rate=$\frac{1}{3}$, best candidates, T=2

Transmit 2 times, Code Rate 1/2

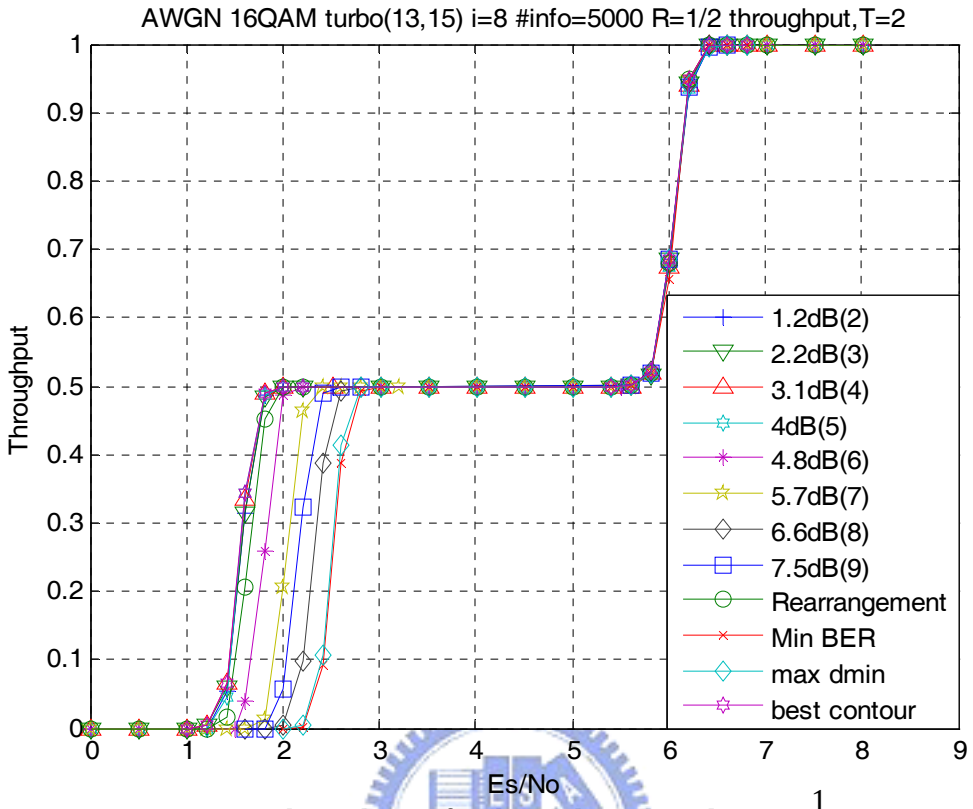

Fig 10.1.1.3 Throughput of turbo code with code rate=$\frac{1}{2}$, T=2

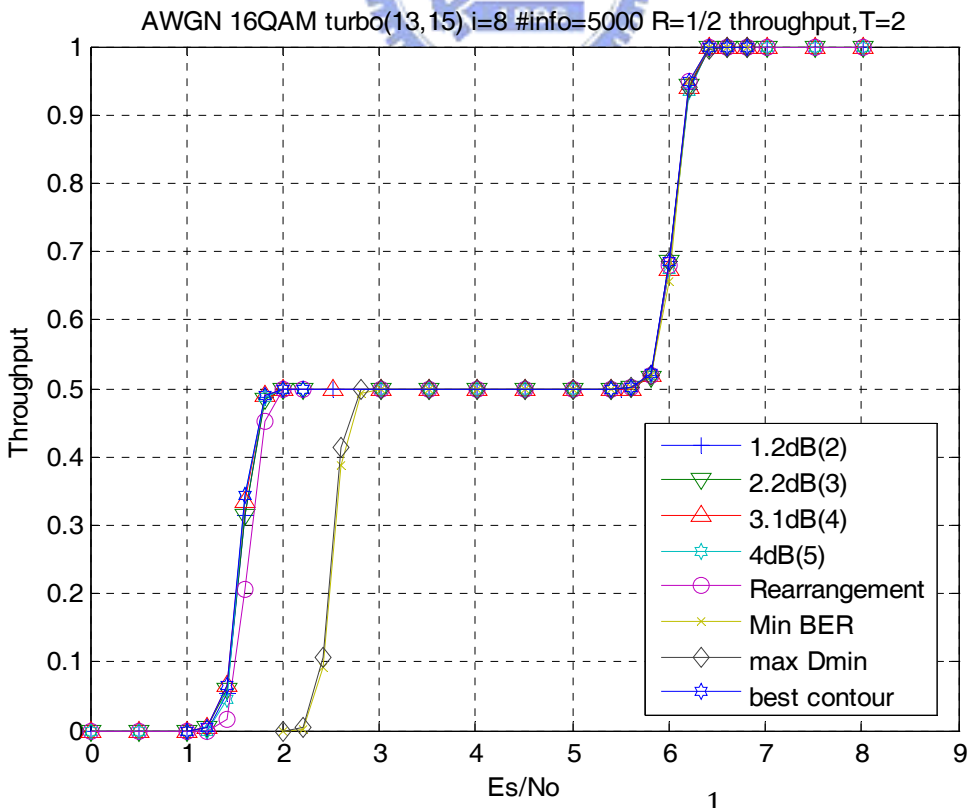

Fig 10.1.1.4 Throughput of turbo code with code rate=$\frac{1}{2}$, best candidates, T=2

Transmit 2 times, Code Rate 3/4

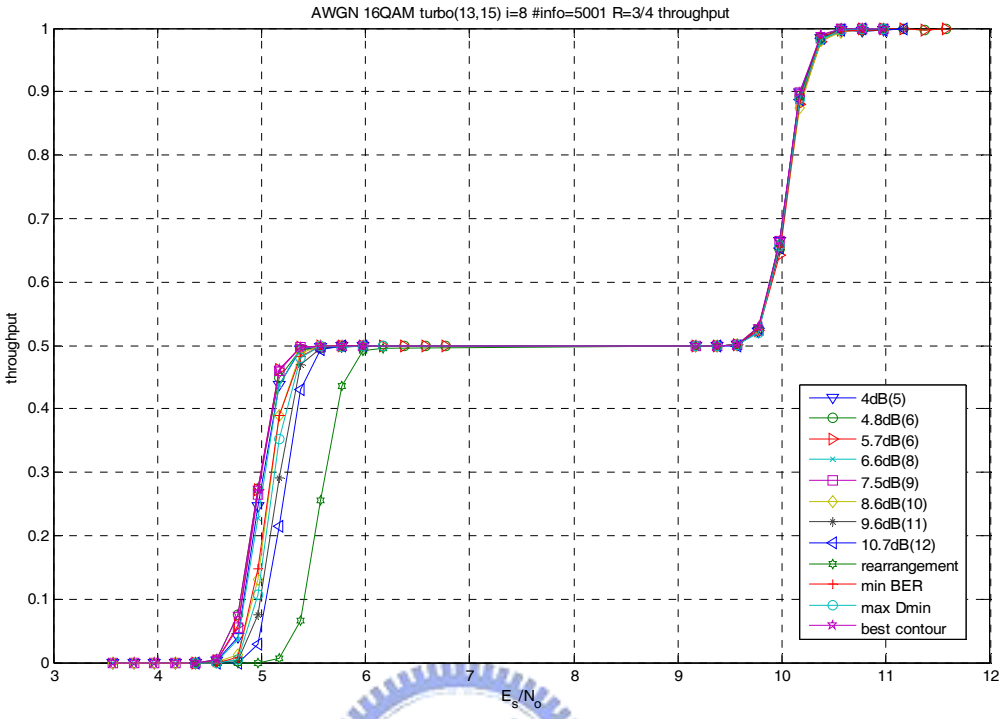

Fig 10.1.1.5 Throughput of turbo code with code rate=$\frac{3}{4}$, T=2

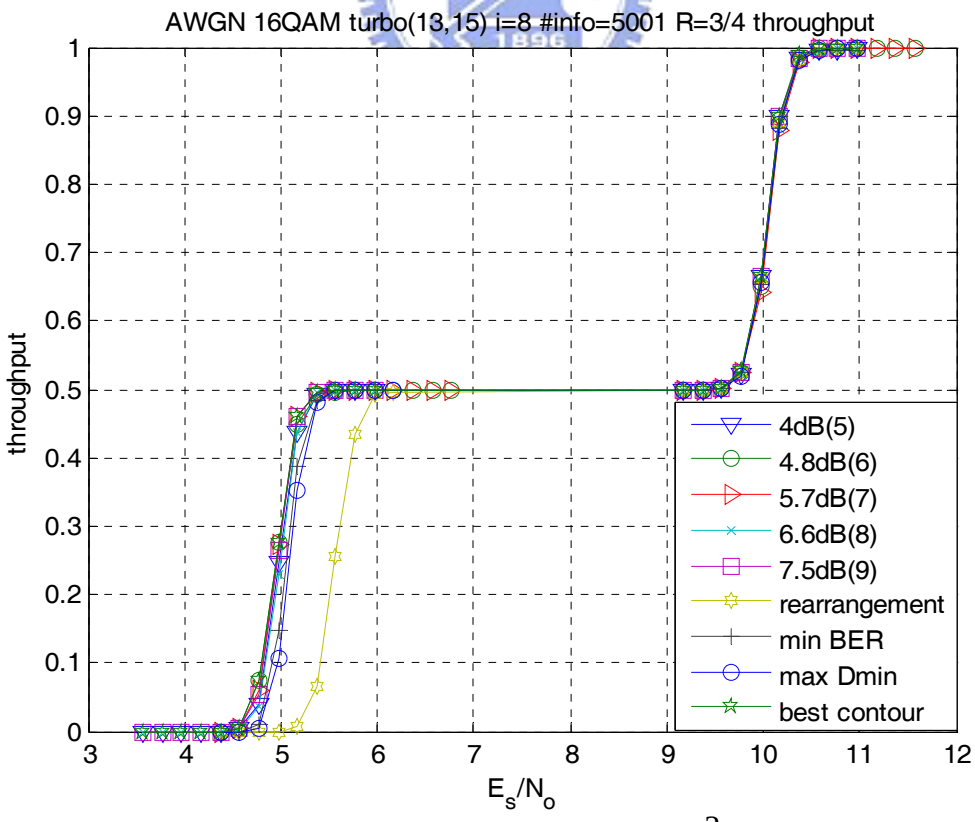

Fig 10.1.1.6 Throughput of turbo code with code rate=$\frac{3}{4}$, best candidates, T=2

Transmit 3 times, Code Rate 1/3

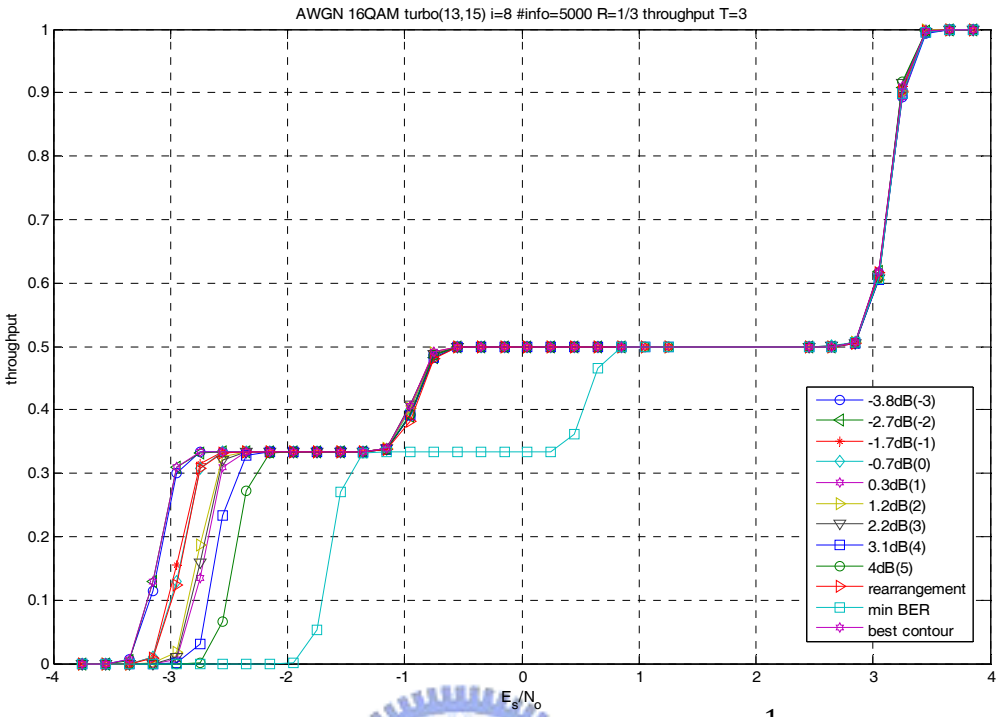

Fig 10.1.1.7 Throughput of turbo code with code rate=$\frac{1}{3}$, T=3

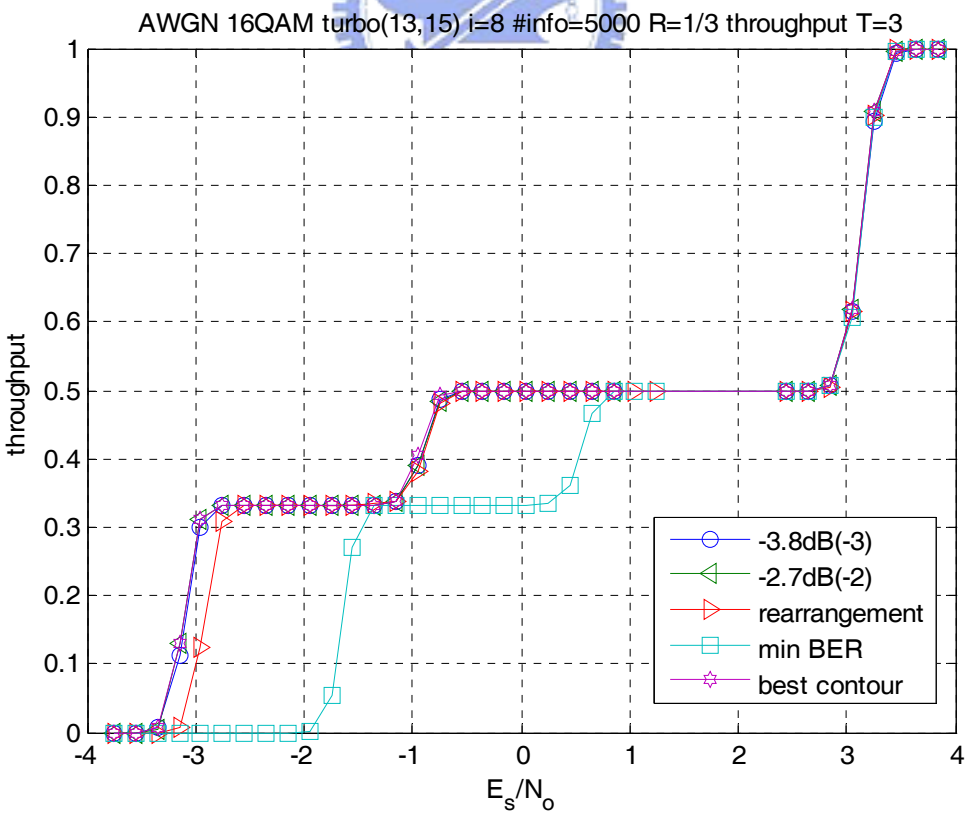

Fig 10.1.1.8 Throughput of turbo code with code rate=$\frac{1}{3}$, best candidates T=3

Transmit 3 times, Code Rate 3/4

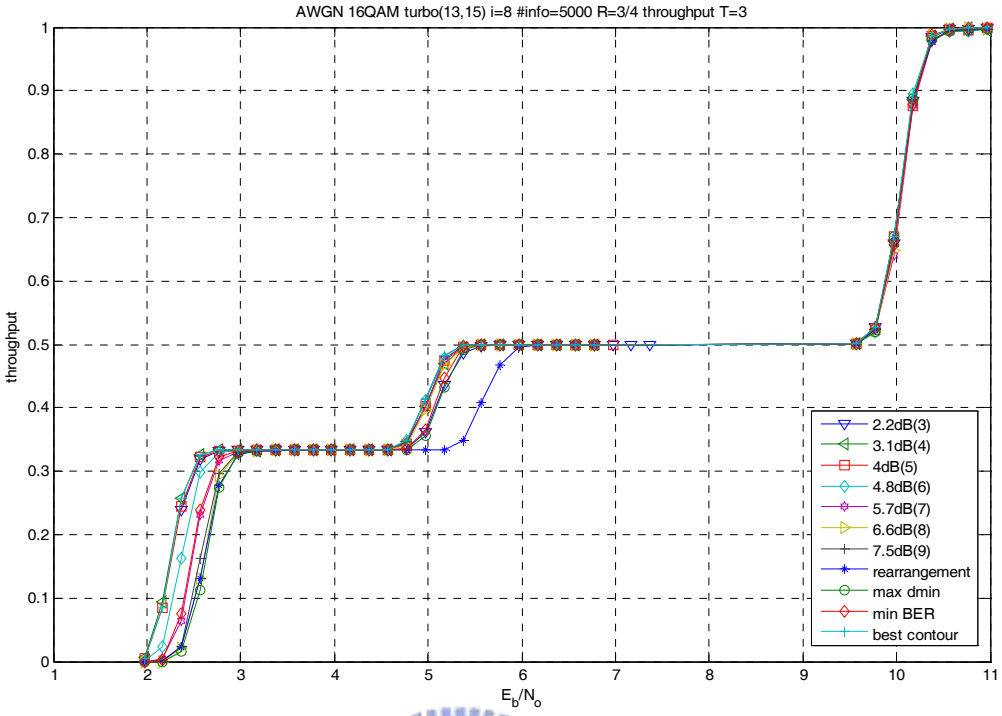

Fig 10.1.1.9 Throughput of turbo code with code rate=$\frac{3}{4}$, T=3

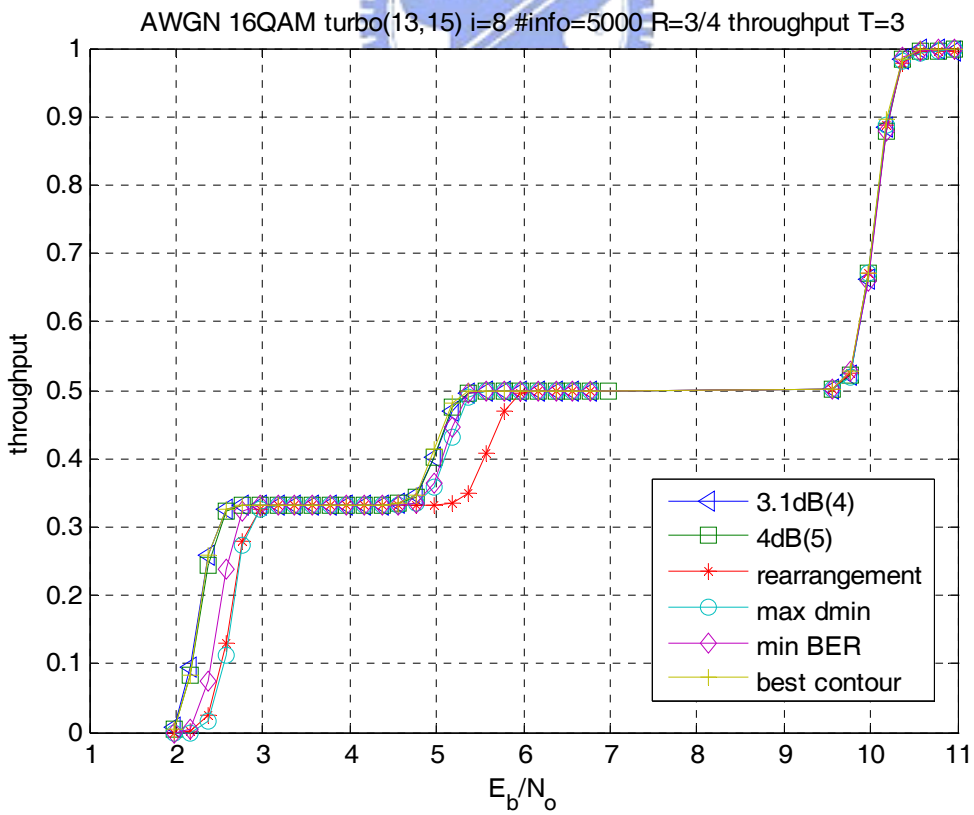

Fig 10.1.1.10 Throughput of turbo code with code rate=$\frac{3}{4}$, best candidates T=3

10.1.2 Convolutional Codes

Transmit 2 times, Code Rate 1/3

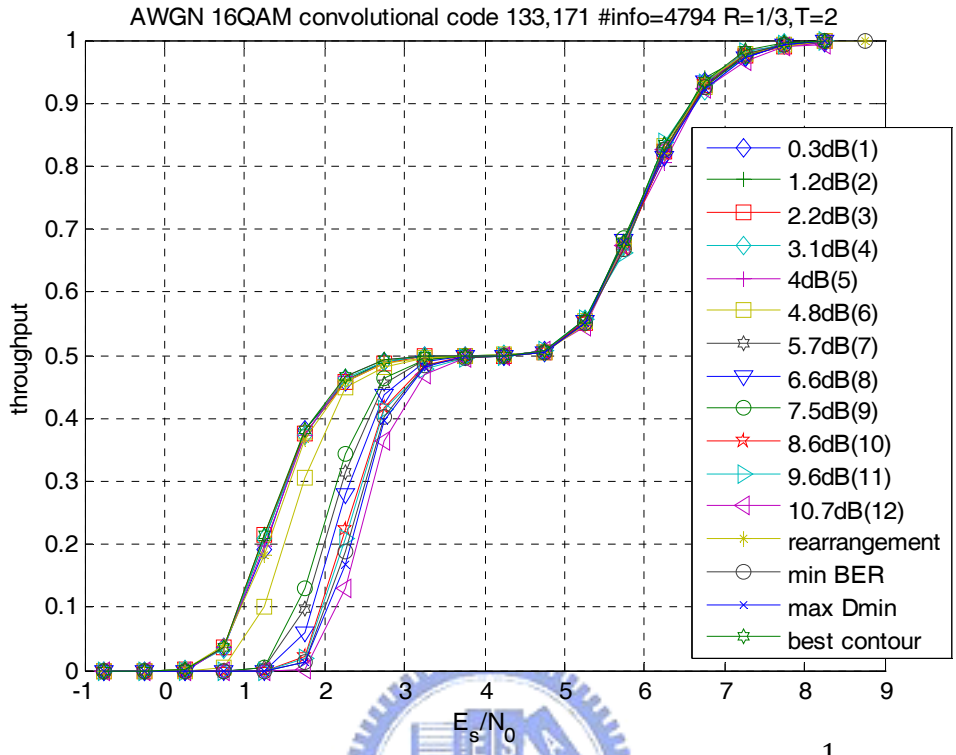

Fig 10.1.2.1 Throughput of convolutional code with code rate=$\frac{1}{3}$, T=2

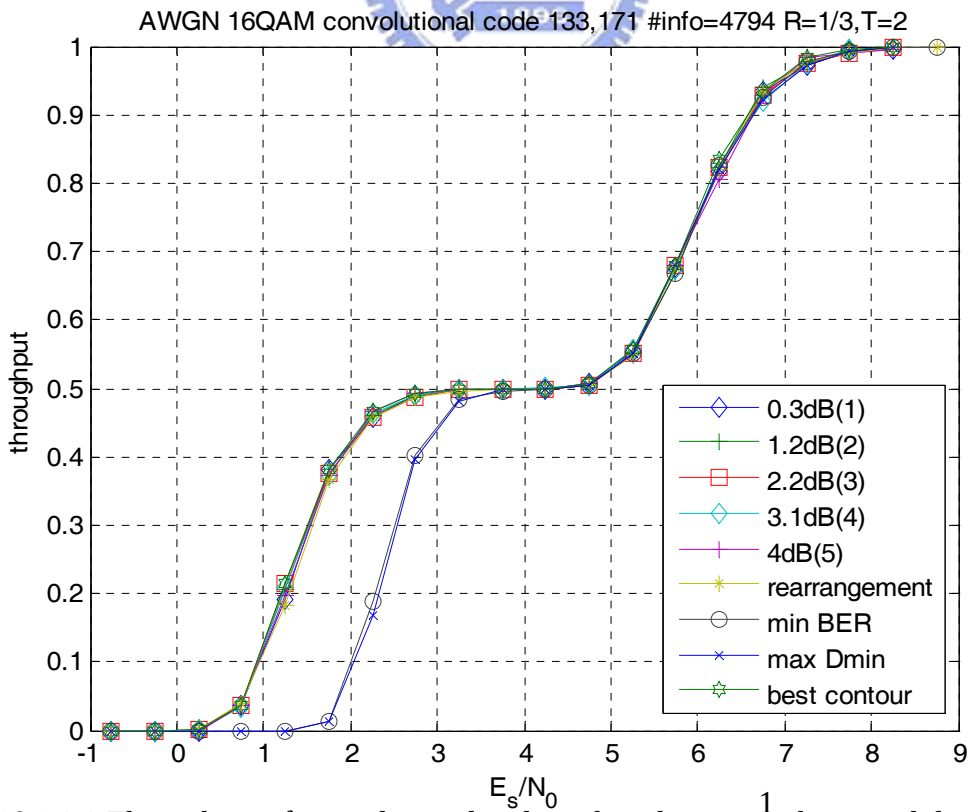

Fig 10.1.2.2 Throughput of convolutional code with code rate=$\frac{1}{3}$, best candidates, T=2

Transmit 2 times, Code Rate 1/2

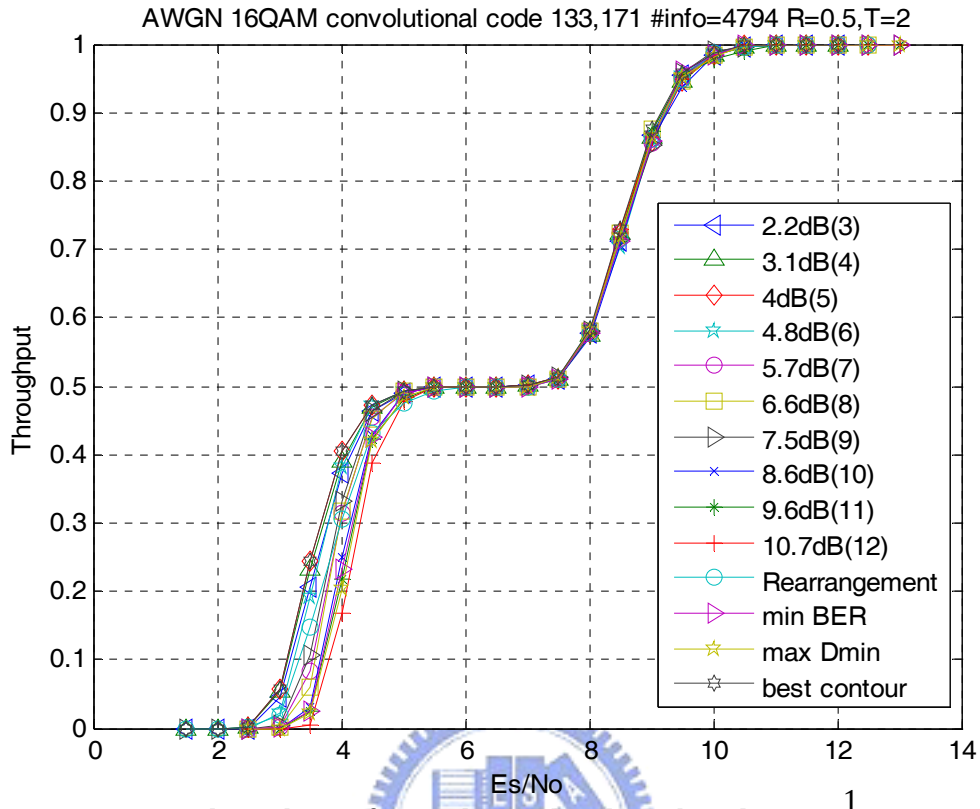

Fig 10.1.2.3 Throughput of convolutional code with code rate=$\frac{1}{2}$, T=2

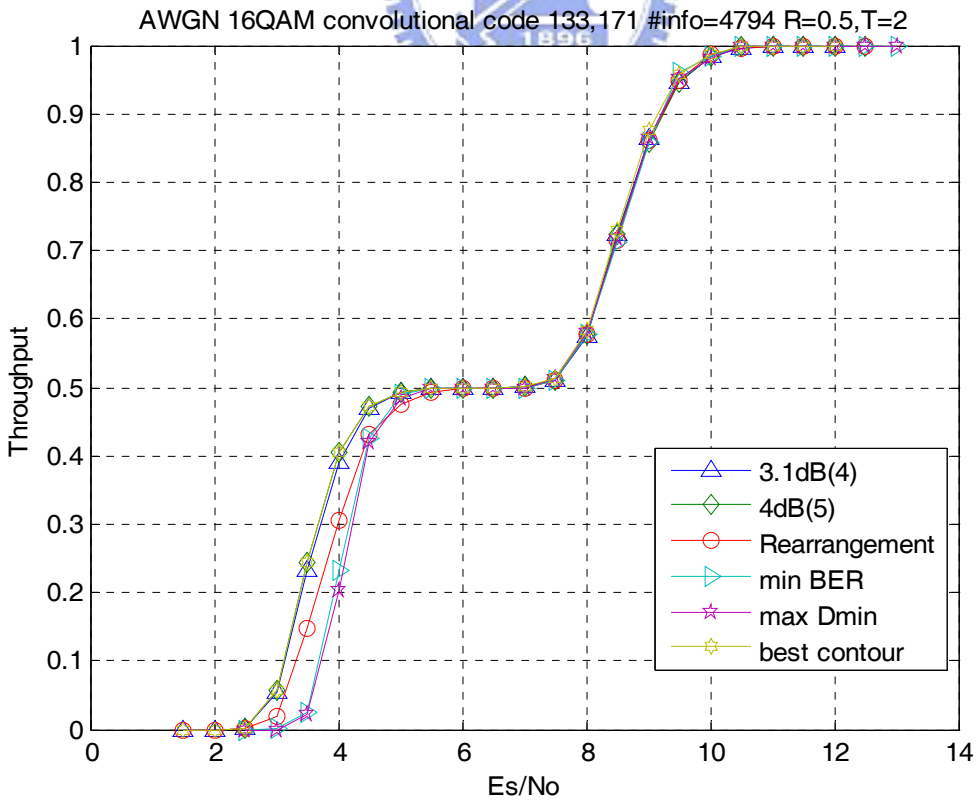

Fig 10.1.2.4 Throughput of convolutional code with code rate=$\frac{1}{2}$, best candidate, T=2

Transmit 2 times, Code Rate 3/4

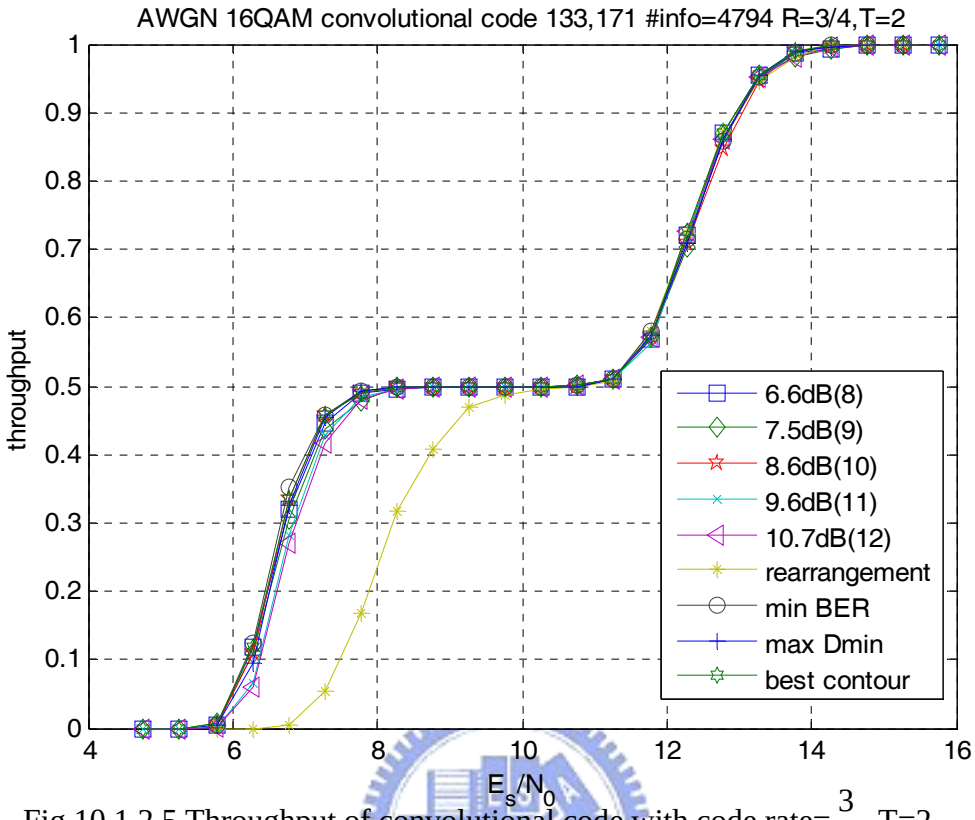

Fig 10.1.2.5 Throughput of convolutional code with code rate=$\frac{3}{4}$, T=2

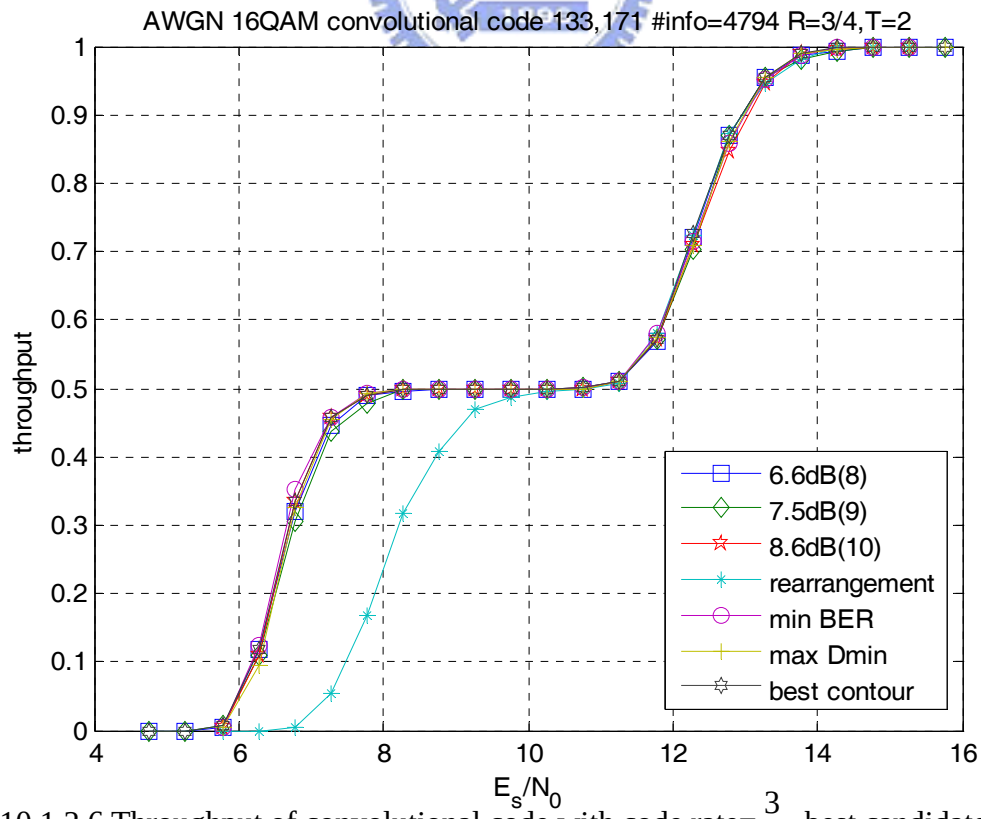

Fig 10.1.2.6 Throughput of convolutional code with code rate=$\frac{3}{4}$, best candidates, T=2

10.2 BICM-ID

In our simulations, to ensure the iteration number is large enough for the trajectory to reach the first intersection, we set the iteration number to 20.

10.2.1 Mappings for Infinite Block Length

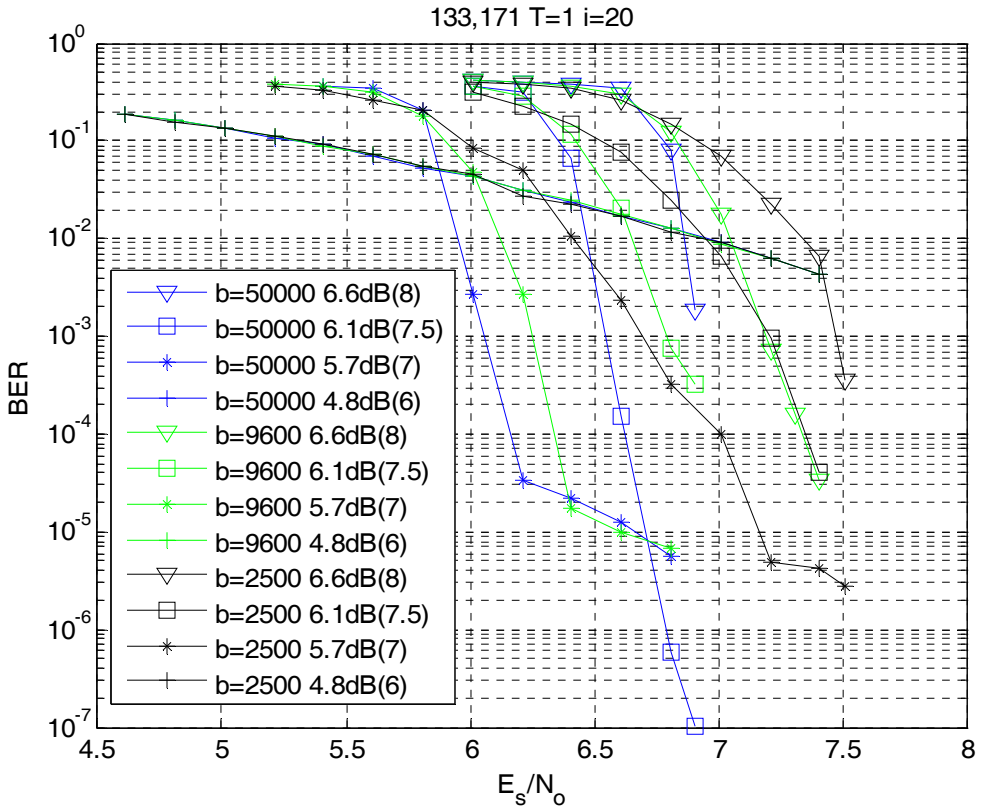

Fig 10.2.1.1 BER of different block length

As illustrates before, mappings designed under the assumption of infinite block length will suffer performance degradation when the block length is finite. The simulation results show that the BER for most of the mapping is not lowest at their designed SNR. Since shorter block length will cause greater variations of the transfer curve and increased correlation of the output extrinsic information, the performance degrades more than larger block length. Hence for the performance to close to optimal, very large block length should be applied, which is impractical for real applications.

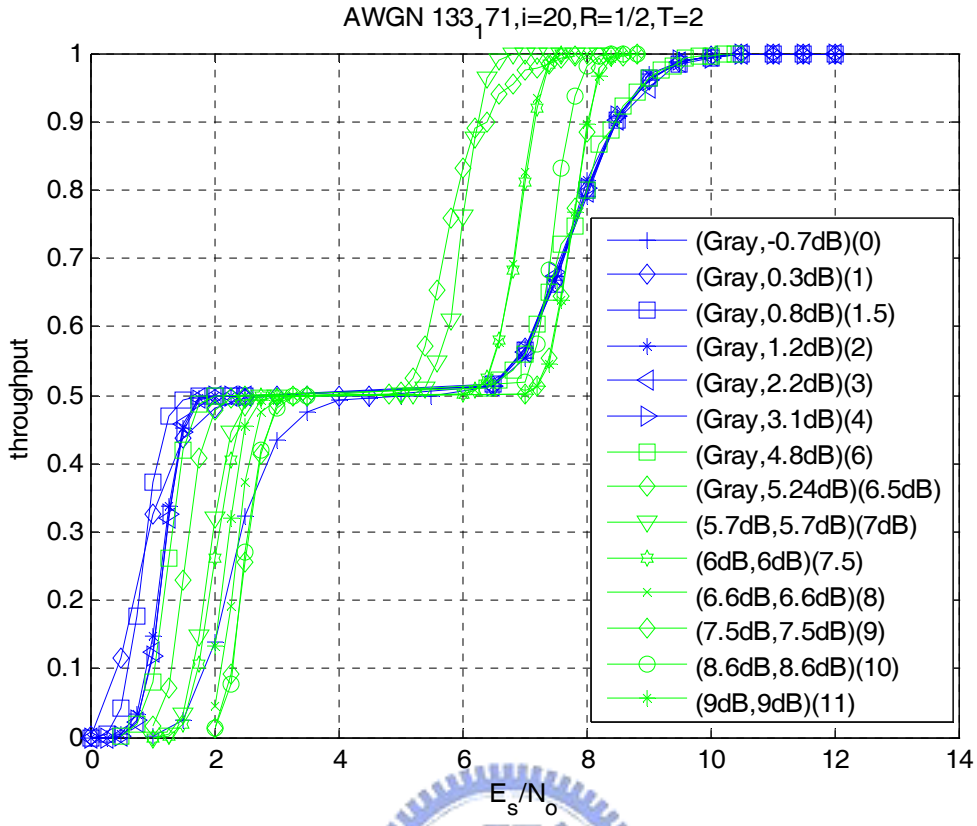

Fig 10.2.1.2 Throughput of block length =2500

The behavior of throughput performance of finite block length is similar and each mapping is not optimal under their designed SNR.

10.2.2 Mappings for Finite Block Length

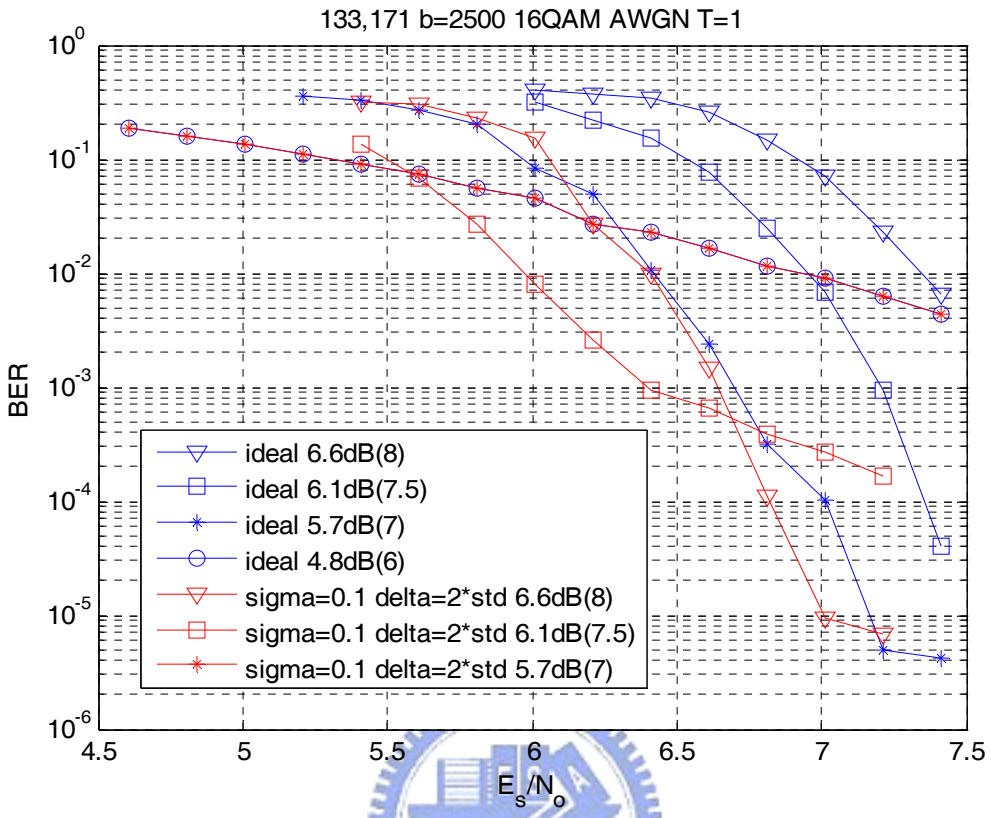

Fig 10.2.2.1 BER comparison of finite length demapper and infinite length demapper (T=1)

Although no optimality can be claimed, the BER of those demappers designed for 2500 block length have showed significant performance improvement over those designed for infinite block length. Margin of two times the standard deviation is preserved in both the demapper and the decoder to allow the tunnel to have about a chance of 97.7% to open.

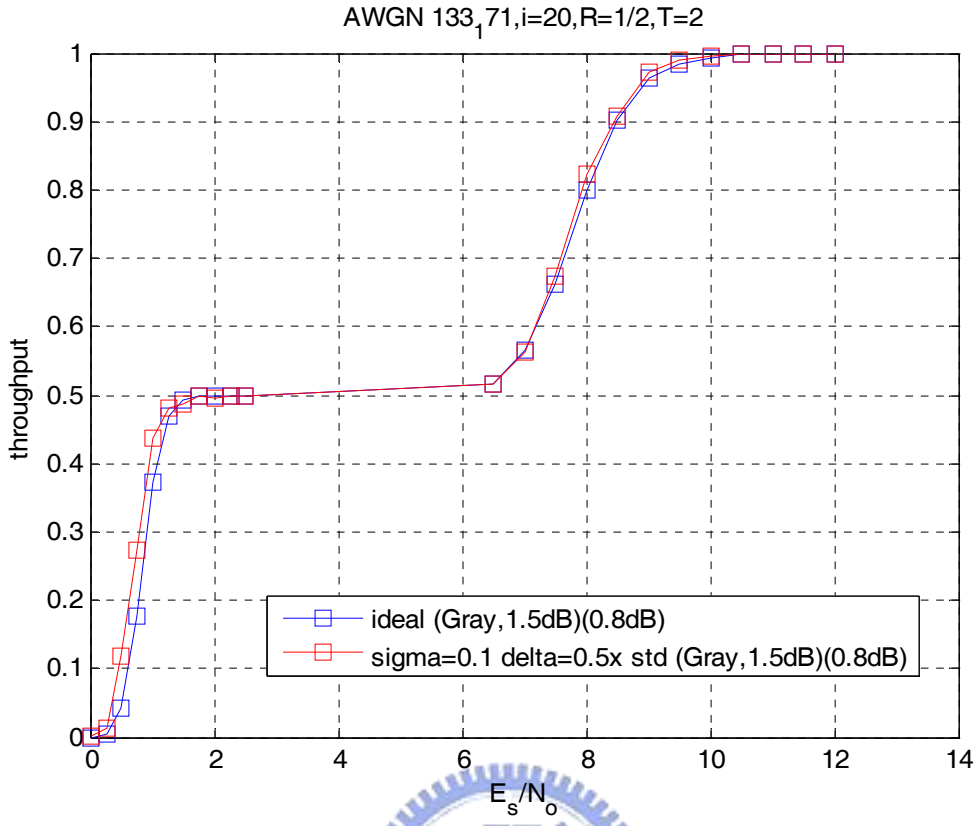

Fig 10.2.2.2 Throughput comparison of finite length demapper and infinite length demapper (T=2)

The throughput is also expected to be improved when margin is preserved to allow some percentages of tunnels to open.

Chapter 11: Conclusions

We proposed using genetic algorithm to efficiently search the optimal mappings for each SNR when BICM is adopted in HARQ. Since different modulation and coding scheme (MCS) operates on different SNR region, mapping scheme should be adaptive to different MCS and the simulation results showed performance gains over the none-adaptive one using existing mappings in the literature. Although theoretically the optimal mappings for each SNR should be designed specifically, the throughput simulation suggests that one single mapping can achieve approximately optimal performance given a specific MCS.

For the mapping design of BICM-ID in HARQ, we also proposed using genetic algorithm to find the optimal mappings for each SNR under the assumption of infinite block length and iteration number. For the practical mappings design of finite block length, some margin is advised to preserve to allow some percentages of tunnels to open. The simulation results showed significant performance gain over the design with no margin when finite block length code is applied.

Reference

[1] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 55–67, Jan. 1982.

[2] E. Zehavi, “ 8-PSK trellis codes for a Rayleigh channel,” *IEEE Trans. Commun.*, vol. 40, pp. 873-884, May 1992.

[3] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 927-945, May 1998.

[4] X. Li and J. A. Ritcey, “Trellis coded modulation with bit interleaving and iterative decoding,” *IEEE J. Select. Areas Commun.*, vol. 17, pp.715-724,1999.

[5] X. Li, A. Chindapol, and J.A. Ritcey, “Bit-interleaved coded modulation with iterative decoding and 8PSK signaling,” *IEEE Trans. Commun.*, vol.50, no.8, pp.1250–1257, Aug. 2002.

[6] S. ten Brink, “Designing iterative decoding schemes with the extrinsic information transfer chart”, *AEU Int. J. Electron Commun.*, vol. 54, no. 6, pp. 389-398, Dec. 2000.

[7] S. ten Brink, “Convergence Behavior of Iterative Decoded Parallel Concatenated Codes,” *IEEE Trans. Comm.*, vol. 49, no. 10, pp. 1727-1737, Oct. 2001.

[8] Ashikhmin,A., Hramer,G., and ten Brink,S.:” Extrinsic information transfer functions: model and erasure channel properties”, *IEEE Trans. Inf. Theory*, vol. 50, No.11,pp.2657-2673, Nov. 2004

[9] Qi, X.; Zhou, S.; Zhao, M.; Wang, J., "Design of constellation labeling maps for iteratively demapped modulation schemes based on the assumption of hard-decision virtual channels," *Communications, IEE Proc.*, vol.152, no.6, pp. 1139-1148, 9 Dec. 2005

[10] S. Lin, D. J. Costello Jr., and M. J. Miller, “Automatic-repeat-request error-control schemes,” *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5–18, Dec. 1984.

[11] S. Lin and D. J. Costello, Jr., “Error Control Coding”. Pearson Prentice Hall, 2003.

[121 D. Chase. "Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets." *IEEE Trans. Commum.* , vol. COM-33. no. *5.* pp. 385-393. May 1985.

[13] S. Y. Le Goff, “Signal constellations for bit-interleaved coded modulation,” *IEEE Trans. Inform. Theory*, vol. 49, no. 1, pp. 307–313, Jan.2003

[14] L. Szczecinski, F.-K. Diop, and M. Benjillali, “BICM in HARQ with Mapping Rearrangement: Capacity and Performance of Practical Schemes”, *GLOBECOM* pp. 1410-1415, NOV 2007

[15] Ch. Wengerter, A. Golitschek Edler von Elbwart, E. Seidel, G. Velev, M.P. Schmitt,” Advanced Hybrid ARQ Technique Employing a Signal Constellation Rearrangement,” *IEEE VTC. Porc.*, vol. 4, Fall 2002.

[16] H. Samra, Z. Ding, and P. M. Hahn “Symbol Mapping Diversity Design for Multiple Packet Transmissions”, *IEEE Transactions on Communications*, vol. 53, No.5, May 2005. pp. 810-817

[17] J. V. K. Murthy and A. Chockalingam, “Log-likelihood ratio based optimum mappings selection for symbol mapping diversity with *M*-QAM,” *Proc. NCC’2005,* Kharagpur, India, January 2005.

[18] L. Szczecinski and M. Bacic, “Constellations design for multiple transmissions : Maximizing the minimum squared Euclidean distance,” in *IEEE WCNC 2005*, New Orlean, USA, Mar. 2005, pp. 1066 – 1071

[19] M. Gidlund and Y. Xu ”An Improved ARQ Scheme with Application to Multi-Level Modulation Techniques”, ISCIT 2004, Vol. 2, pp.973 - 978 ,Oct 2004

[20] M. N. Khormuji and E. G. Larsson, “Improving collaborative transmit diversity by using constellation rearrangment,” in *Proc. IEEE Wireless Commun. and Networking Conf.*, Hongkong, Mar. 2007

[21] J. Roberson, Z. Ding, “A BICM approach to Type-II Hybrid ARQ,” *IEEE ASSP. Conf. Proc.* vol.4, May 2006

[22] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison Wesley, 1989

[23] E. K. P. Chong, S.H. Zak, “An Introduction to Optimization”, John Wiley & Sons, 2001