2004

BICM with simplified demapping based on capacity arguments

Taylor Tsung-Ying Tsai
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Tsai, Taylor Tsung-Ying, "BICM with simplified demapping based on capacity arguments" (2004). *Master's Theses*. 2686.

DOI: <https://doi.org/10.31979/etd.ne67-avzb>

https://scholarworks.sjsu.edu/etd_theses/2686

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

BICM WITH SIMPLIFIED DEMAPPING BASED ON CAPACITY ARGUMENTS

A Thesis

Presented to

The Faculty of the Department of

Electrical Engineering

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science in Electrical Engineering

By

Taylor Tsung-Ying Tsai

December 2004

UMI Number: 1425488

Copyright 2004 by
Tsai, Taylor Tsung-Ying

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 1425488

Copyright 2005 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© 2004

Taylor Tsung-Ying Tsai

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

Dr. Robert H. Morelos-Zaragoza

Dr. Avtar Singh

Dr. Essam Marouf

APPROVED FOR THE UNIVERSITY

ABSTRACT

BICM WITH SIMPLIFIED DEMAPPING BASED ON CAPACITY ARGUMENTS

By Taylor Tsung-Ying Tsai

This thesis suggests an alternate way to handle the demapping process for efficient BICM schemes over AWGN channels. This process is simplified by examining the individual channel capacities associated with a given bit labeling. Those bits corresponding to particular levels reaching the binary capacity limit are given a maximum bit metric value since they are highly reliable. Consequently, these bit metrics are not computed. The approach refers to this as a “hard” Maximum Likelihood (ML) bit metric or “hard-demapping,” and it reduces the decoding effort. In this thesis, tasks of three aspects are combined to demonstrate the complexity reduction that is possible to achieve for BICM over an AWGN channel. Capacities for two of the M-ary QAM signal constellations are analyzed with two different labeling methods. The behaviors of log-likelihood ratio (LLR) between bit-metrics (BM) are discussed based on capacity arguments as well as error correcting codes was used for the simulations.

ACKNOWLEDGMENT

I would like to thank my advisor Dr. Robert Morelos-Zaragoza for his guidance towards this study as well as his encouragement for submitting this thesis idea to conferences. During the past two years, I received not only all his support throughout my efforts of completing this thesis, but also everything beyond the making of this thesis. I would also like to thank Dr. Essam Marouf and Dr. Avtar Singh for their patience in reading this complete version. And finally, I would like to give my thanks to my beloved parents and my beloved Jean S. Tien, for their support throughout my studies in the USA.

TABLE OF CONTENTS

1.	Overview.....	1
1.1	Introduction and Motivation	
1.2	Proposed Idea	
1.3	Thesis Outline	
2.	Signal Constellation and Mapping for BICM	6
2.1	Signal Constellation	
2.2	Gray Mapping	
2.3	Set-Partition Mapping	
2.4	Conclusion	
3.	Channel Capacities Over AWGN Channel	16
3.1	Introduction	
3.2	Constrained Channel Capacity	
3.3	Binary Channel Capacity	
3.4	Conclusion	
4.	Bit Metric and LLR for BICM	37
4.1	Introduction	
4.2	Basic Concepts for Bit Metric and Log-Likelihood Ratio	
4.3	Hard-demapping	
4.4	Conclusion	
5.	Numerical Results and Discussions.....	60

5.1	Preliminaries	
5.2	Bit Error Rate Performance	
6.	Conclusion and Final Remark	69
	References	70
	Appendices.....	72
A.	Graphical View for Gray Mapping using 64-ary QAM Constellation	72
B.	Graphical View for Set-Partition Labeling using 8PSK Constellation	73
C.	The First Set Used to Find LLR for Set-Partition Labeling using 64-ary QAM Constellation with LLR Behaviors	74

LIST OF FIGURES

Figure 1	Bit-interleaved Coded Modulation	3
Figure 2	Proposed BICM Receiver	4
Figure 3	16-ary PAM Constellation	7
Figure 4	Signal Constellations for 32- and 64-ary QAM	8
Figure 5	Gray Labeling for 32-ary QAM Constellation	10
Figure 6	Gray Labeling for 64-ary QAM Constellation	11
Figure 7	The Base of Gray Labeling Scheme	12
Figure 8	Idea of Set-Partition Mapping Using QPSK	13
Figure 9	Set-Partition Labeling for 32-ary QAM Constellation	13
Figure 10	Set-Partition Labeling for 64-ary QAM Constellation	14
Figure 11	Discrete Memoryless Channel over AWGN	18
Figure 12	Probability Density Function Set between M-ary Inputs and Outputs	19
Figure 13	Constrained Channel Capacity for 32-ary Signal Constellation	22
Figure 14	Constrained Channel Capacity for 64-ary Signal Constellation	23
Figure 15	The Signal Set used for Finding Modulation-constrained Capacity	25
Figure 16	Sub-subsets used for Computing Binary Channel Capacity for 1 st Bit	26
Figure 17	Sub-subsets used for Computing Binary Channel Capacity for 2 nd Bit	27
Figure 18	Sub-subsets used for Computing Binary Channel Capacity for 3 rd Bit	28
Figure 19	Sub-subsets used for Computing Binary Channel Capacity for 4 th Bit	29

Figure 20	Binary Capacity versus SNR for 32 QAM using Gray Mapping	31
Figure 21	Binary Capacity versus SNR for 32 QAM using Set-Partition Mapping	32
Figure 22	Binary Capacity versus SNR for 64 QAM using Gray Mapping	32
Figure 23	Binary Capacity versus SNR for 64 QAM using Set-Partition Mapping	34
Figure 24	Signal Sets used to find LLR for QPSK	41
Figure 25	Signal Sets used to find LLR(0) for 32-ary QAM using Gray Mapping	43
Figure 26	Signal Sets used to find LLR(0) for 64-ary QAM using Gray Mapping	44
Figure 27	Signal sets that used to find LLRs for 4-th bit of 64 QAM	45
Figure 28	LLR Behaviors vs. In-Phase Projection for 4-th bit of 64 QAM	47
Figure 29	Signal sets that used to find LLRs for 5-th bit of 64 QAM	48
Figure 30	LLR Behaviors vs. In-Phase Projection for 5-th bit of 64 QAM	49
Figure 31	Signal sets that used to find LLRs for the 5-th bit of 32 QAM	49
Figure 32	LLR Behaviors vs. In-Phase Projection for 5-th bit of 32 QAM	50
Figure 33	Actual vs. Quantized LLRs for the 5-th bit of 32 QAM at SNR = 20 dB	53
Figure 34	Actual vs. Quantized LLRs for the 5-th bit of 64 QAM at SNR = 30 dB	54
Figure 35	Actual vs. Quantized LLRs for the 4-th bit of 64 QAM at SNR = 30 dB	54
Figure 36	Hard-demapping used for 32-ary QAM at BICM Receiver	55
Figure 37	Hard-demapping used for 64-ary QAM at BICM Receiver	55
Figure 38	The Complexity inside the Iterative Decoder vs. SNR by applying Hard- demapping on different levels for 32-ary QAM	56
Figure 39	The Complexity inside the Iterative Decoder vs. SNR by applying Hard- demapping on different levels for 64-ary QAM	58

Figure 40	Total Constrained Soft-decision Capacity vs. Total Constrained Soft/Hard- decision Capacity for 32-ary QAM	62
Figure 41	Total Constrained Soft-decision Capacity vs. Total Constrained Soft/Hard- decision Capacity for 64-ary QAM	63
Figure 42	Simulation Result for the 32-ary BICM vs. the Simulation with Hard Demapping	64
Figure 43	Simulation Result for the 64-ary BICM vs. the Simulation with Hard Demapping	65
Figure 44	The Error Propagates through the Codeword for Multilevel Codes	66
Figure 45	The Error Propagates through the Codeword for BICM	67
Figure 46	The Complexity for Iterative Decoder over SNR for 32 QAM with and without Hard-demapping on MSB Level.....	67
Figure 47	The Complexity for Iterative Decoder over SNR for 64 QAM with and without Hard-demapping on the 4-th and the 5-th level.....	68

1. OVERVIEW

1.1 Introduction and Motivation

Bit-interleaved coded modulation (BICM) is an efficient coding scheme that is capable of reaching good error performance with powerful codes used. For many years, there has been a significant research effort devoted to the improvement of BICM efficiency. The main consideration here is to examine the signal constellation within the modulator portion. This is because the constellation labeling directly affects the efficiency of a BICM scheme. The use of lower density signal constellation sets over AWGN channel for a bandwidth-efficient BICM is discussed (Le Goff 310). In that paper, it is concluded that the best choice of the labeling technique for low-density signal constellations (e.g., 8PSK and 16QAM) is Gray mapping. Moreover, it also suggests higher-density constellation labeling methods as a good research problem whereby optimal power-efficient BICM schemes can be designed to fit almost all constellations (Le Goff 313). To handle the signal constellation, a key element is the channel capacity, as it reflects the scheduling of the decoding process as well as the error performance of the system.

In the idea of multilevel coding, to look at the capacities in a separate manner helps to decode the binary sequences independently so that, when using staged decoding, the overall decoding complexity can be reduced (Washmann et. al. 1389-1391). Even though in both references it is suggested to consider the signal constellation and capacity to reduce the decoding complexity for a BICM over AWGN, there is still another aspect

of this complexity called demapping, i.e., the computation of bit metrics for demodulation, where the complexity increases as the constellation density for the BICM goes higher (Caire et. al. 929). Again, the significance of the capacity towards the constellation labeling is mentioned in the same paper for the purpose of showing the better performance of bit-wise coded modulation schemes.

A BICM scheme contains seven main blocks, as depicted in Figure1. The main concern throughout this thesis focuses on the modulation and demodulation blocks. As indicated in the above paragraph, the cause of degradation in decoding efficiency is related to signal constellation, labeling, and demapping. Instead of improving the error performance based only on constellation and its mapping or inside the decoding block, focus is on the demapping process. For BICM, the demapping process within the demodulator is another area affecting the decoding pace. Based on the known ideas mentioned in articles written by Le Goff, Washmann et. al., and Caire et. al., the motivation here is to reduce the BICM decoding complexity from a different point of view. By coming up with an alternate demapping algorithm, observing the behaviors of the binary channel capacities, characteristics can be identified regarding to the reliability issue.

1.2 Proposed Idea

In this work, the ideas representing in those previously discussed papers are extended with employing both Gray and Set-Partition (SP) mappings for both 32- and 64-ary constellations in BICM over an AWGN channel, and find the corresponding channel

Figure 1. Bit-interleaved Coded Modulation; π denotes interleaving at the bit level.

capacities as well as their component binary channel capacities. From these capacities, a decoding algorithm for this scheme is determined by comparing the individual binary channel capacities. Whenever individual bits from the constellation corresponding to particular levels reach the binary capacity limit, a maximum bit metric value is assigned to those bits since they are highly reliable. Consequently, these bit metrics are not computed. This is defined as “hard demapping” and this approach can reduce the decoding effort.

To distinguish from the idea of multilevel coding, we demonstrate serial operation message decoding for BICM. In other words, the system considers a stream of message bits without having to separate them into different levels. The purpose of considering the binary channel capacities is to keep track of every bit-metric for demapping at each level and for this stage only. It may be the case that some messages feeding into the decoder require no further computation and can be given maximum reliability levels. To look at the capacities in different levels helps to identify which of the binary channel capacities are reaching the maximum. Figure 2 roughly depicts the idea of hard demapping at the

receiving side of the BICM. In Figure 2, LLR values are computed and delivered at the output of the bit metric generator. Those bits in correspondence to a highly reliable level will be given quantized values equal to the largest value available. All the LLR values are then passed to an iterative decoder. In this work, we consider Low-Density Parity-Check (LDPC) codes with iterative decoding. Decoding complexity can be further reduced inside the decoder: Code nodes associated with the hard decisions do not perform message computation but rather send always the maximum value. We refer to the LLRs associated with the hard decision levels as “hard” bit-metrics. The other LLR values are directly passed to the decoder. The error performance of this algorithm can be explained by considering the capacity.

Figure 2. Proposed BICM Receiver.

1.3 Thesis Outline

This thesis presents a demapping algorithm for power-efficient BICM schemes over AWGN channels. This hard demapping approach was achieved by simplifying the bit metric and LLR computations. This idea is in relation to channel capacity and constellation labeling; therefore, the study is divided into two portions and is further

described below.

The first portion including two chapters describes and analyzes theoretical concepts. For BICM over AWGN, both Gray and SP labeling techniques for 32- and 64-ary signal constellations are included in Chapter 2. The modulation constrained and binary channel capacities for the corresponding signal constellations are discussed in Chapter 3. Chapter 4 briefly restates the BM and LLR so the proposed hard demapping idea can be represented. Finally the simulation results are shown and compared in Chapter 5 toward the error performance so that the complexity reducing issue is further discussed.

2. SIGNAL CONSTELLATION AND MAPPING FOR BICM

2.1 Signal Constellation

This chapter deals with the constellation labeling techniques for BICM scheme. Different constellation labelings have influence on BICM since symbol energies are involved. It is necessary to understand and analyze various constellation mapping techniques. The signal constellation mapping process happens inside the modulator referring back to Figure 1. The main task within the modulator is to take each arriving signal point from a transmitted binary sequence and treat it as one vector in an N -dimensional signal space, where N is the number of orthonormal basis functions required for those transmitted signal sets. Therefore, as indicated in (Haykin 322-323), all the representations of these vectors form a set on this N -dimensional signal space are called the signal constellation.

There are several types of signal constellation such as Pulse Amplitude Modulation (PAM) or Quadrature Amplitude Modulation (QAM). PAM constellation deals with one dimensional signal space and QAM is for two-dimensional space. The size is defined as $M = 2^m$. Figures 3 and 4 depict QAM and PAM constellation structures using $M = 16, 32$, and 64 as examples.

In these figures, every point on either coordinate is given a value to indicate its position. For N -dimensional M -ary signal constellation, the labeling of signal points is based on orthonormal basis function:

$$\phi_i(t) = \frac{g_i(t)}{\sqrt{\int_0^T g_i^2(t) dt}}, \quad i = 1, 2, \dots, N,$$

where $g_i(t)$ is the intermediate function for those energy signals transmitted to the modulator. For PAM constellation, each signal $x_n(t)$ is given a value on the axis represented as:

$$x_n(t) = \begin{cases} (2n-1-M)\phi_1(t), & 0 \leq t \leq T \\ 0, & \text{elsewhere} \end{cases}$$

where $n = 1, 2, \dots, M$ indicates number of input signal point (Haykin 344-373). Therefore, for 16-ary PAM constellation shown in Figure 3, the general format for $n = 1, 2, \dots, 16$ signal vector is: $x_n = 2n-1-16$. For example, if $n = 3$ and 14, then $x_3 = 2*3-1-16 = -11$ and $x_{14} = 2*14-1-16 = +11$.

Figure 3. 16-ary PAM Constellation.

Similarly, signal points for M-ary QAM constellation are labeled based on the same orthonormal basis function. Contracting to PAM, the QAM uses a two-dimensional basis function as indicated in (Haykin 360-361); namely, the input waveform of QAM is the combination of two PAM signals in quadrature shown as below:

$$x_n(t) = \begin{cases} x_{n1}\phi_1(t) + x_{n2}\phi_2(t), & 0 \leq t \leq T \\ 0, & \text{elsewhere} \end{cases}$$

where the bases function for $N = 2$ are represented as:

$$\begin{aligned}\phi_1(t) &= \sqrt{\frac{2}{T}} \cos \omega_0 t \\ \phi_2(t) &= \sqrt{\frac{2}{T}} \sin \omega_0 t, \quad 0 \leq t \leq T\end{aligned}$$

each signal point is a vector set presented as $x_n = (x_{n1}, x_{n2})$, $n = n1 = n2 = 1, 2 \dots M$. For two-dimensional signal constellation such as 32- and 64-ary constellations depicted in Figure 4-(a) and 4-(b), each signal point is labeled using the above format.

Figure 4. Signal Constellations for 32- and 64-ary QAM.

Instead of using 32-ary square QAM, the 32-ary cross QAM is used in order to maintain the uniform Euclidean distance between neighboring points. As explained above, the symbol energy (E_s) for every signal differs with its position which affects the values of channel capacity. For instance, if there is a signal point with a coordinates of $(-3, -5)$, then the symbol energy for this point is the product of the shortest distance toward the origin

with a factor d ; namely, $Es = 9d^2 + 25d^2 = 34d^2$. The value d is set to be 1 in this thesis for the convenient purpose. The following sections describe two different mapping techniques along with examples of using 32- and 64-ary signal constellations.

2.2 Gray Mapping

When a two-dimensional M-ary signal constellation ($M = 2^m$) is labeled, the first thing to consider is that each symbol contains m bits of information. Consequently, every single binary bit for each position in the label affects both channel capacity and decoding. As described in (Caire et. al. 928-929), in the modulator, the signal constellation structure is made by mapping the transmitted code sequence onto the new signal sequence by the labeling map defined as $\mu: \{0,1\}^m \rightarrow X$, where X is the transmitted code sequence. And each code sequence has a unique labeling shown as $X \in X_b^i$, where i indicate the bit position out of the m bits of the corresponding symbol and b stands for its binary values shown in the following statement. For a signal constellation of size $M = 2^m$ with minimum Euclidean distance $d_{\min}$, for all $i = 1, 2 \dots m$ and $b \in \{0,1\}$, if each X belonging to the map μ has at most one $z \in X_b^i$ at distance $d_{\min} = 1$, then this binary map is a Gray Labeling. For the AWGN channel, the key feature of this type of signal mapping is to minimize and maintain this Euclidean distance while assigning neighboring signal points onto the map. Therefore, the value of the variance is kept in its minimum.

Figures 5 and 6 demonstrate labeling scheme for 32- and 64-ary constellation using Gray labeling technique. For figures involving different M-ary ($M = 2^m$)

places in the most and the second most significant bits in this thesis. Then the higher density signal constellations can be obtained by expanding more points over the four quadrants. A term known as Quasi-Gray for 32-ary QAM is mentioned in (Wesel et. al. 2427) and it addresses a situation that some neighboring signal sets do not always satisfy the Gray condition. Although there maybe several M-ary Gray configurations other than what are shown in Figures 6 and 7, they have identical capacity behaviors as long as the AWGN channel is involved.

Figure 6. Gray Labeling for 64-ary QAM Constellation.

2.3 Set-Partition Mapping

There is another labeling method called Set-Partition (SP). As mentioned in (Ungerboeck 57-60), in contrast to Gray Mapping technique, the SP labeling technique is

Figure 7. The Base of Gray Labeling Scheme.

to partition signal set successively with increasing minimum distances between subset signals. Therefore the variance is maximized. The main idea about this labeling technique is to break the original set into several subsets with equal subset size.

Furthermore, the next step is to split these subsets into several other subsets. And then the following steps are recursively repeat the previous steps until the whole constellation is decomposed into single point subsets. This implies that subsets are defined for numbers of levels depending on how many bit positions are taking into account. Each subset has unique binary position value jointly labeled with subset signal labels from previous levels. Figure 8 illustrates this idea using simple constellation (i.e., Quadrature Pulse Shift Keying) as an example. As indicated in this figure, QPSK is divided into two levels according to the bit positions. And each level is assigned a term X^0 or X^1 and contains two subsets distinguished by its binary value. For the first level, X^0 , the diagonal distance between subset signal points is considered as the maximum subset distance. For level 1, the maximum distance is the uniform Euclidean distance

mentioned previously. This characteristic reflects the behavior of component channel capacities focusing on each level independently.

Figure 8. Idea of Set-Partition Mapping Using QPSK.

The SP labeling for 32- and 64-ary QAM constellations can be constructed in a similar fashion and is depicted in Figures 9 and 10 where the format of each point is represented in the same way as described in section 2.2.

Figure 9. Set-Partition Labeling for 32-ary QAM Constellation.

Figure 10. Set-Partition Labeling for 64-ary QAM Constellation.

Appendix B illustrates the steps for SP labeling methods in detail by using one of the examples from Ungerboeck's paper (58-59).

2.4 Conclusion

This chapter briefly describes the main feature of handling transmitted messages within the BICM modulator. It offers a little bit of a flavor for the definition of the M-ary signal constellation as well as two different labeling techniques (Gary and SP). It is very significant to introduce background knowledge towards the channel capacity estimation and to illustrate two particular signal constellations (i.e., 32 and 64 QAM). Since channel capacity is the key element to handle a signal constellation, this chapter plays a transition

role to approach the central idea in this study. In addition, there exist several other structures of 32 and 64-ary constellations related to Gray and SP labelings. This fact may result in different error performances but there is no connection between the complexities reducing issue as to the decoding process. In order to approach the main idea of this thesis, the next step is to introduce the AWGN channel efficiency toward Gray and SP mapping. The following chapter focuses on the concept and computation of the channel constrained capacity and its component binary capacities over AWGN channel. Again, 32- and 64-ary signal constellations are used as examples to demonstrate this concept.

3. CHANNEL CAPACITIES OVER AWGN CHANNEL

3.1 Introduction

If there is a sequence X transmitted over a noisy channel such as an AWGN channel, then the message sequence output from the channel is Y and given by the relation:

$Y = X + n$, where n is Gaussian noise. Channel capacity is the measure of the bandwidth efficiency of the channel and is measured by selecting the largest mutual information over X and Y (McIllree 5-6). The equation below shows the general format for mutual information, $I(X;Y)$,

$$E[I(X;Y)] = E \left[\log_2 \left(\frac{P(X|Y)}{P(X)} \right) \right],$$

and it will be revisited later.

The channel efficiency depends heavily on the signal constellations and can be considered in two different representations: Constrained and binary channel capacities. The binary channel capacity relates to the bit metric as transmitting the binary messages. To reach this idea, the main task is to analyze the individual capacities from different signal constellations. The constrained channel capacity is found as the total sum of the individual channel capacities. The next section will first introduce the constrained channel capacity followed by the binary channel capacities in section 3.3. Conclusions will be made based on the analytical results obtained for those corresponding channel capacities computed in a separate manner.

3.2 Constrained Channel Capacity

As indicated in the previous section, constrained channel capacity is the measurement of the channel efficiency and relates to the mutual information between input and output sequences. In general, the capacity is denoted by “C” and measures how many bits per symbol can be sent through the channel. The unit of capacity is bits/symbol-channel. For M-ary ($M = 2^m$) signal constellation, there are m levels of binary bit positions for labeling every bit and the constrained capacity is a function of the Signal to Noise Ratio (SNR). Its upper bound is equal to the number of the levels of the constellation. For example, for 32- and 64-ary QAM signal constellations, the maximum constrained capacities are equal to 5 and 6, respectively.

Before discussing the constrained capacity, it is necessary to introduce some preliminaries related to AWGN channel. There are three concepts to be described: probability density function, relation between noise variance and SNR, and mutual information. Between input X and output Y , there are two expressions regarding to probability density functions involving with noise variance, σ^2 :

$$p(y|x=0) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y+\frac{\sigma^2}{2}\right)^2}{2\sigma^2}}$$
$$p(y|x=1) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y-\frac{\sigma^2}{2}\right)^2}{2\sigma^2}}$$

The above equation is called the conditional probability where the AWGN variance, σ^2 can be found from SNR. The definition for SNR is the symbol energy divided by twice of the variance for AWGN in this thesis and is given by:

$$2\sigma^2 = 10^{-\frac{SNR}{10}}, \text{ or}$$

$$SNR = -10\log_{10}(2\sigma^2).$$

As indicated in the above relations, the probability density functions are represented for the output Y conditional on occurrence of input X with binary value 1 or 0. They are also the characteristic equations for the discrete memoryless channel over AWGN as depicted in

Figure 11. Discrete Memoryless Channel over AWGN.

the figure above. For M-ary signal constellation, there is more than one input symbol X such that the probability density functions become more complicated and it is the concatenated version of the binary case illustrated in Figure 12.

Figure 12. Probability Density Function Set between M-ary Inputs and Outputs.

Based on Figure 12, the probability density function set can be expressed as:

$p(Y|X) = p(Y_1 Y_2 \dots Y_M | X_1 X_2 \dots X_M)$. This function has the following relations according to

Bayes' Theorem that can be used to compute the mutual information:

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X,Y) = p(X|Y)p(Y) = p(Y|X)p(X)$$

Because mutual information heavily depends on probability density function, the explicit form of mutual information between X and Y is defined as shown in section 3.1. The constrained channel capacity is defined as the largest average mutual information obtaining by finding the maximum product of probability density function set for each transmitted signal point and the sum of the average mutual information (McIllree 13). The constrained channel capacity expression is stated below as indicates in McIllree's Thesis:

$$C = \max_{p(X_1) \dots p(X_M)} \sum_{m=1}^M I(X; Y) ,$$

where the average mutual information is:

$$I(X;Y) = \int \cdots \int_{n\text{-fold}} p(Y|X) p(X) \log_2 \left(\frac{p(Y|X)}{p(Y)} \right) dY.$$

As described previously, the probability density function $p(Y|X_m)$ means the probability of Y occurs under the total m possible inputs of X where $m = 1, 2, \dots, M$. For M-ary signal constellation over AWGN in this thesis, the above expression regarding to the probability function can be represented as:

$$\begin{aligned} p(Y|X_m) &= \prod_{n=1}^2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-d_E^2}{2\sigma^2}\right) \\ &= \prod_{n=1}^2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(Y - X_m)^2}{2\sigma^2}\right). \end{aligned}$$

The variable $d_E^2 = (Y - X_m)^2$ is the Euclidean distance square between the input message points from X and output points from Y . The above equation is called conditional probability and it collects the total probability density function between transmitted and received sequences over the $N = 2$ -dimensional signal set. When M output message points Y are mapped onto the signal space, there might be a chance of the landing error related to the distance between the reference point and the actual received point so that we can calculate how accurate the output is on the expected position with the existence of the noise.

The symbol energy regarding to the signal point on the constellation map is mentioned in section 2.1. It is a significant element in computing the channel capacity as it affects the Euclidean distances. Therefore, it is necessary to normalize the symbol

energy while computing the Euclidean distance for the probability density function included inside the maximum average mutual information expression (McIllree 6). Consider a signal point $(-3, -5)$ used in section 2.1 again, the coordinate after normalizing the symbol energy becomes $\left(\frac{-3}{\sqrt{34}}, \frac{-5}{\sqrt{34}}\right)$. Instead of normalizing the individual symbol energy, we need to consider the total average symbol energy for calculating the constrained channel capacity since the expressions listed above collect the probability for the whole signal set. The average symbol energy is to sum up the total Euclidean distance for every single point and divide it by the number of points contained in M-ary constellation. For 32- and 64-ary QAM constellations, the average symbol energies are 20 and 42 respectively. Finally, the constrained channel capacity for 32- and 64-ary QAM constellations can be found by applying the above formulas. As mentioned in McIllree's thesis, it is possible to use Gauss-Hermite quadrature to approximate the integration portion for finding the mutual information (33-38). Figures 13 and 14 illustrate the constrained channel capacities for 32- and 64-ary QAM signal constellations. For both Gray and Set-Partition labelings, the channel capacity has identical behavior for the same M-ary signal constellation due to the normalized symbol energies.

Figure 13. Constrained Channel Capacity for 32-ary Signal Constellation.

As Figures 13 and 14 indicate, when the SNR increases, the capacity also increases until the SNR reach some point roughly beyond 20 dB, the curve reaches to its channel capacity limit. This means that noise no longer affects capacity. Also, note that the limit of the channel capacity equal the number of bits of the M -ary constellation (i.e., channel capacity limit is 5 for 32-ary constellation and 6 for 64 QAM). This channel capacity limit has a term called the Capacity Boundary according to (Haykin 602).

Figure 14. Constrained Channel Capacity for 64-ary QAM Signal Constellation.

3.3 Binary Channel Capacity

After realizing the boundary for the channel capacity limit for two different cases, this section introduces another view toward this measurement: the component channel capacity or binary channel capacity. The binary channel capacities are obtained by decomposing the constrained channel capacity into several levels. In addition, instead of considering the total normalized symbol energy, the normalized symbol energy for single point must take into account for binary channel capacity while encountering different situations.

The total modulation-constrained capacity is the sum of the component binary channel capacities. To compute the binary channel capacity, several steps of capacity computation for separate levels are required (Washmann et. al. 1365-1377). For a two dimensional M -ary constellation, there are $M = 2^m$ signal points with m digits for each signal point and the capacity computation can be divided into m levels. Thus for the k -th level, $k = \{0 \dots (m-1)\}$, there are 2^k subsets with $2^{(k+1)}$ sub-subsets and each point within a subset are sorted by the associated label bits. For convenience, each sub-subset can be labeled $C(b[k]b[k-1] \dots b[0])$ where each $b[k] \in \{0, 1\}$. For instance, one group of subset two for M -ary constellation could be labeled as $C(010)$. In addition, the average symbol energies for signal points are different between sub-subsets. The labeling methods can also make the symbol energies equal between different levels due to symmetrical characteristic for a signal constellation. All these facts influence the steps for computing the subset capacities.

As mentioned in (Washmann et. al. 1376-1378), the main idea of obtaining the binary channel capacity is described as the following; for the lower levels, subset capacities are combined together and from them subtracted higher-level capacities. Thus this step is operated repeatedly for the $k-1$ levels. The binary channel capacity for the most significant bit level is equal to the sub-subset capacity collected from the $(k-1)$ -th level. The equation mentioned in section 3.2 is used again in relation to the sub-subset capacities calculations; unfortunately, there is no explicit formula to estimate the binary channel capacities due to the normalized average symbol energies from different levels.

One example for either 32- and 64-ary QAM cases is described in detail with the graphical aids in appendix. The following expressions only demonstrate the necessary equations used to find the binary channel capacities for both 32- and 64-ary QAM constellation using both mappings. Each binary channel capacity for M-ary signal constellation is given a label as $CG[i]$, $i = \{0...m\}$ for the Gray mapping and $CSP[i]$, $i = \{0...m\}$ for Set-Partition labeling. By applying the concept mentioned in (Washmann et. al. 1376-1378) for computing the channel capacities individually, the following material demonstrates the details on how to find the binary channel capacities for M-ary signal constellation using a 32-ary QAM constellation as example with Quasi-Gray mapping.

The very first element is the modulation-constrained channel capacity, C , and it is computed using the equation from section 3.2 referring to the configuration below.

Figure 15. The Signal Set used for Finding Modulation-constrained Capacity.

The above structure has average symbol energy of 20, and it is used to normalize the single symbol energy for every signal point from the constellation. Then the

modulation constrained capacity C is obtained. The next step is to find channel capacities in each subset for the particular bit level. Therefore, the first level for 32 QAM has two subset capacities, $C(0)$ and $C(1)$ and is shown below:

Figure 16. Sub-subsets used for Computing Binary Channel Capacity for 1st Bit.

For each subset signal containing bit value of 1 for the first bit (i.e., LSB) is represented in black dot from above figure where the white dot stands for binary value of 0 for the first bit. Because the structures of the two subsets are symmetrical, the average symbol energies for both of them are equal. The subset channel capacity is again obtained by applying the constrained channel formula. Also, the capacities from two subsets are identical. Thus, the channel capacity at level 0 for 32 QAM using the idea mentioned in the paper written by (Washmann et. al. 1376-1378) is:

$$CG[0] = C - \frac{1}{2}(C(0) + C(1)) = C - C(0)$$

The next step is to find the channel capacity for the second bit position. To do that, we need to consider subsets corresponding to this level referring to the figure below.

Figure 17. Sub-subsets used for Computing Binary Channel Capacity for 2nd Bit.

According to the above figure, there are four sub-subset channel capacities for the second level of 32 QAM (i.e., $C(00)$, $C(01)$, $C(10)$, $C(11)$). The average symbol energies for all sub-subsets signals are identical due to the symmetrical characteristic. Then the binary channel capacity for this level is represented as:

$$\begin{aligned}
 CG1 &= \frac{1}{2}(C(0) + C(1)) - \frac{1}{4}(C(00) + C(01) + C(10) + C(11)) \\
 &= C(0) - C(00)
 \end{aligned}$$

For analyzing the binary channel capacity for the third bit position, we also observe the sub-subsets for this particular level shown below.

Figure 18. Sub-subsets used for Computing Binary Channel Capacity for 3rd Bit.

For this level, the average symbol energies for subsets no longer matched with each other. The average symbol energy for the subsets containing binary values of *000*, *001*, *010*, and *100* is 16 where the average symbol energy is 24 for sub-subsets of *100*, *101*, *110*, and *111*. Therefore, the binary channel capacity for this level becomes as

$$\begin{aligned}
 CG[2] &= \frac{1}{4}(C(00) + C(01) + C(10) + C(11)) \\
 &\quad - \frac{1}{8} \left(C(000) + C(001) + C(010) + C(011) + \right. \\
 &\quad \left. C(100) + C(101) + C(110) + C(111) \right) \\
 &= C(00) - \frac{1}{2}(C(000) + C(100))
 \end{aligned}$$

The figure below shows the sub-subsets used to find the binary channel capacity for bit level four and there are six different average symbol energies involved within the computations.

Figure 19. Sub-subsets used for Computing Binary Channel Capacity for 4th Bit.

Referring back to the figure above, there are 16 sub-subsets represented by different symbols. The average symbol energy for sub-subsets *0000*, *0001*, *0100*, and *0101* is 14 where sub-subsets *0010* and *0011* have average symbol energy of 6. The sub-subsets *0110* and *0111* have average symbol energy of 22. The sub-subset symbol energies for the cases on the right hand side of the figure are listed as follow: $E_s = 18$ for sub-subsets of *1000* and *1001*. $E_s = 26$ for sub-subsets of *1010*, *1011*, *1110*, and *1111*. And finally, the sub-subsets *1100* and *1101* have average $E_s = 34$. Thus, the sub-subset capacity at

this level is $\frac{1}{16} \left(4C(0000) + 2C(0010) + 2C(0110) + 2C(1000) + 4C(1010) + 2C(1100) \right)$.

The binary channel capacity can be written as:

$$CG[3] = \frac{1}{2}(C(000) + C(100)) - \frac{1}{8} \left(\begin{array}{l} 2C(0000) + C(0010) + C(0110) \\ + C(1000) + 2C(1010) + C(1100) \end{array} \right).$$

Finally, the binary channel capacity for the MSB level is just the same as the sub-subset capacity found in the previous level, so it is shown as:

$$CG[4] = \frac{1}{8} \left(\begin{array}{l} 2C(0000) + C(0010) + C(0110) \\ + C(1000) + 2C(1010) + C(1100) \end{array} \right).$$

Therefore, the final revised formulas used to find the binary channel capacities for M -ary QAM associating with two labeling methods are obtained and the following sets of expressions illustrate the binary channel capacities computations for both 32- and 64-ary QAM using both mapping techniques.

$$\begin{aligned} CG[0] &= C - C(0) \\ CG[1] &= C(0) - C(00) \\ CG[2] &= C(00) - \frac{1}{2}(C(000) + C(100)) \\ CG[3] &= \frac{1}{2}(C(000) + C(100)) - \frac{1}{8} \left(\begin{array}{l} 2C(0000) + C(0010) \\ + C(0110) + C(1000) \\ + 2C(1010) + C(1100) \end{array} \right) \\ CG[4] &= \frac{1}{8} \left(\begin{array}{l} 2C(0000) + C(0010) + C(0110) \\ + C(1000) + 2C(1010) + C(1100) \end{array} \right) \end{aligned}$$

Where C is the constrained capacity described in section 3.2 and is used for the rest of the cases. Next, formulas used to find the binary channel capacity for the same number of QAM constellation in associates with Set-Partition mapping is listed on the bottom of the next page. Figure 20 and 21 illustrate the results of obtaining the binary channel capacities curves for 32-ary QAM using Gray and Set-Partition labelings.

Figure 20. Binary Capacity versus SNR for 32 QAM using Gray Mapping.

$$CSP[0] = C - C(0)$$

$$CSP[1] = C(0) - C(00)$$

$$CSP[2] = C(00) - \frac{1}{2}(C(000) + C(100))$$

$$CSP[3] = \frac{1}{2}(C(000) + C(100)) - \frac{1}{4}(3C(0000) + C(0010))$$

$$CSP[4] = \frac{1}{4}(3C(0000) + C(0010))$$

Figure 21. Binary Capacity versus SNR for 32 QAM using Set-Partition Mapping.

Figure 22. Binary Capacity versus SNR for 64 QAM using Gray Mapping.

The formulas used to find the binary channel capacities for 64 QAM using Gray labeling.

$$\begin{aligned}
CG[0] &= C - \frac{1}{2}(C(0) - C(1)) \\
CG[1] &= \frac{1}{2}(C(0) - C(1)) - \frac{1}{4}(C(00) + C(01) + C(10) + C(11)) \\
CG[2] &= \frac{1}{4} \begin{pmatrix} C(00) + C(01) \\ +C(10) + C(11) \end{pmatrix} - \frac{1}{8} \begin{pmatrix} C(000) + 2C(001) \\ +C(010) + C(011) \\ +C(101) + C(110) + C(111) \end{pmatrix} \\
CG[3] &= \frac{1}{8} \begin{pmatrix} C(000) + 2C(001) \\ +C(010) + C(011) \\ +C(101) + C(110) + C(111) \end{pmatrix} - \\
&\quad \frac{1}{16} \begin{pmatrix} 2C(0000) + 2C(0001) \\ +2C(0010) + 3C(0011) \\ +2C(0101) + C(0110) \\ +2C(1000) + C(1001) + C(1100) \end{pmatrix} \\
CG[4] &= \frac{1}{16} \begin{pmatrix} 2C(0000) + 2C(0001) \\ +2C(0010) + 3C(0011) \\ +2C(0101) + C(0110) \\ +2C(1000) + C(1001) + C(1100) \end{pmatrix} - \\
&\quad \frac{1}{32} \begin{pmatrix} 4C(00000) + 4C(00001) \\ +4C(00010) + 6C(00011) \\ +4C(00101) + 2C(00110) \\ +4C(01000) + 2C(01001) + 2C(01100) \end{pmatrix} \\
CG[5] &= \frac{1}{32} \begin{pmatrix} 4C(00000) + 4C(00001) \\ +4C(00010) + 6C(00011) \\ +4C(00101) + 2C(00110) \\ +4C(01000) + 2C(01001) + 2C(01100) \end{pmatrix}
\end{aligned}$$

The formulas used to find the binary channel capacities for 64 QAM using Set-Partition labeling.

$$CSP[0] = C - C(0)$$

$$CSP[1] = C(0) - C(00)$$

$$CSP[2] = C(00) - C(000)$$

$$CSP[3] = C(000) - \frac{1}{4}(C(0000) + C(0010) + 2C(0001))$$

$$CSP[4] = \frac{1}{4} \left(\begin{matrix} C(0000) + \\ C(0010) + \\ 2C(0001) \end{matrix} \right) - \frac{1}{4} \left(\begin{matrix} C(00000) + \\ C(00010) + \\ 2C(00001) \end{matrix} \right)$$

$$CSP[5] = \frac{1}{4}(C(00000) + C(00010) + 2C(00001))$$

Figure 23. Binary Capacity versus SNR for 64 QAM using Set-Partition Mapping.

For the computation of the binary channel capacity for the most significant levels associating with all the cases, there is no need to include $C(b[m] = 0)$ or $C(b[m]=1)$ since it is identical to the subset capacities found for level $(m-1)$, due to the symmetric characteristics of the mapping. For instance, while considering Gray mapping for 64-ary constellation, there are 31 sub-subsets with different average symbol energies. Because more sub-subsets are involved to calculate the unconstrained capacities, each term within the regarding equations listed previously becomes significant (i.e., the numbers of symmetric cases between sub-subsets are less than the one using SP labeling for 64 QAM case). Furthermore, the process used to find the binary channel capacities for 32-ary constellation is relatively easier. Figures 20 through 23 are important references help to explain the proposed idea in a later chapter.

3.4 Conclusion

This chapter presents the basic concept of computing capacities for 32 and 64-ary signal constellations over an AWGN channel. They are divided into constrained and component binary channel capacities. These capacities are analyzed for two mapping techniques, Gray and Set-Partition mapping. Referring to the results obtained from calculating capacities, we realize this quantity could be varied since the average symbol energies are taken into account. Different labeling techniques do not have much effect on the behaviors for constrained capacity. However, they have a great deal of influence on binary channel capacity. Take Figure 20 as example, with Gray mapping applied, curves for $CG[4]$ through $CG[0]$ are randomly distributed due to the variance for AWGN where

the curves in Figure 21 has a distribution in decreasing order from left to right due to the maximum variance. Also, these differing results are also because of the labeling position based on the mapping and causing the difference of the symbol energies affecting the capacity computation. The binary channel capacities for the most significant bit positions reach the capacity limit at lower SNR comparing to other levels for the constellations illustrated in Figures 20 to 23. When the binary capacity reaches its capacity limit, it means that the transmitted bit reaches a reliable status. This fact suggests keeping track of every bit metric within the demodulator such that log-likelihood ratio computations (demapping) can be performed. The concept of binary channel capacity and its relation to bit metric computations are significant and will be revisited in the next chapter.

4. BIT METRIC AND LLR FOR BICM

4.1 Introduction

Referring back to Figure 1 in Chapter 1, when the transmitted messages arrive at the receiving side of the BICM, the first block they encountered is the demodulator. The main function for this block is to perform demapping including two steps; bit-metric (BM) generation and log-likelihood ration (LLR) computations. The BMs are obtained by calculating conditional probabilities between the received transmitted bits and the LLR is computed from the BMs. In contrast with the above statements, the demodulator described in (Caire et. al. 929) generates maximum likelihood (ML) bit metrics that form a series of branch matrices. In order to pass the appropriate message bits into the decoder, the BM and LLR values become significant.

The way to handle demapping process is similar to the way to compute the binary channel capacities describe in previous chapter. The probability density functions are again heavily involved. So the demapping process is also to consider the conditional probability for receiving message bit in Y given the occurrences of the message bit in X been transmitted through the AWGN channel. Even though both channel capacity or BM and LLR are considered separately for their behaviors, all these messages corresponding to each step are finally combined together to form a long sequence and then are transmitted to the associating devices in the case of BICM in this thesis.

Demodulation and decoding in the demodulator cause complexity as they reflect the decoding schedule. The number of operations inside the decoder depends not only on

levels of the signal constellation, but also on the number of LLR generated along with the BM in several levels within the constellation. To reduce the complexity, as mentioned in chapter one, the decoding process needs to be modified. Therefore, this thesis proposes an idea called hard-demapping that provides an alternate way to handle the demapping process. The key is to avoid several computation steps for LLRs by assigning fixed BMs.

In the following section, the computation of BM between transmitted messages bits and receiving message bits is presented. The detailed descriptions on obtaining the log-likelihood ratios corresponding to those bit metrics are also mentioned. Consequently, in section 4.3, the proposed hard-demapping is introduced referring to section 4.2 and all these ideas are concluded in section 4.4.

4.2 Bit Metrics

As stated previously, the demodulator generates the BM, denoted by $\lambda(i, b)$, where i indicates the i -th level of the signal constellation and b represents the binary values for all the possible positions in that particular level, i.e., $b \in \{0, 1\}$. This quantity has an explicit formula according to (Caire et. al. 929):

$$\lambda(i, b) = \begin{cases} \log \sum_{x \in X_b^i} P_\theta(Y|X) \\ \log \sum_{x \in X_b^i} p(Y|X) \end{cases} .$$

In a BICM scheme, the upper portion of the equation shown above is the logarithm value of total *a priori* probability defined in Chapter 3 with perfect channel state information (CSI), θ , and the lower portion is the one without the CSI. Specifically, the conditional

probability is set to have no CSI in this thesis to keep simplicity of computation. In this paper, i appearing from the last page have range defined as the following: $i \in \{0...4\}$ for the case of 32-ary QAM and $i \in \{0...5\}$ for 64-ary QAM.

In the case of no CSI for BICM, when the signal sequence arrives at the demodulator, where demapping takes place, the computation of the set of BM can be simplified as follow:

$$\lambda(i, b) = \log \sum \left(p(y | x(i, b)) \right).$$

The above equation produces a set of BMs, $\lambda(i, b=0)$ and $\lambda(i, b=1)$, for each bit position corresponding to each receiving message Y . There are m levels for M-ary constellation, so m sets of BMs can be obtained from the constellation. For an error correcting code of length n , the total numbers of BMs generated from each level is $2n/m$. Finally, the demapping process is completed by outputting the LLR value $\lambda(i, b)$, where the LLR is defined as (Schulze 121):

$$LLR(i) = \log \left(\frac{\sum \left(p(y | ax(i, b=0)) p(x(i, b=0)) \right)}{\sum \left(p(y | ax(i, b=1)) p(x(i, b=1)) \right)} \right).$$

In practice, the LLR can also be approximated by finding the maximum value of BM and subtracting with each other shown below:

$$LLR(i) \approx \max \log \left(p(y | ax(i, b=0)) p(x(i, b=0)) \right) - \max \log \left(p(y | ax(i, b=1)) p(x(i, b=1)) \right).$$

These two expressions obtained from (Schulze 121-122) are for the case of Rician Fading Channel; therefore, there is a fading amplitude, a , associates with the LLR computation.

However, for the case of AWGN channel, there is no fading amplitude involved and the transmitted X has equal probability of having the binary value of 0 or 1 (i.e., $p(x(i,b=0))=p(x(i,b=1))$) in our case. By combining the ideas regarding to the LLR computation in both (Caire et. al. 929) and (Schulze 121), the equation for finding this quantity is then finalized as shown below:

$$LLR(i) = \log \left(\frac{\sum (p(y | x(i,b=1)))}{\sum (p(y | x(i,b=0)))} \right),$$

This formula can be written as:

$$LLR(i) = \lambda(i,b=1) - \lambda(i,b=0).$$

Instead of taking the ratio for $\lambda(i,b=0)$ over $\lambda(i,b=1)$, we consider the reversal ratio to distinguish from the case mentioned in (Schulze 121). For a code with length n associating with M-ary constellation, there are n/m numbers of LLR collected from each level. For any M-ary QAM, the BM and LLR for level 0 can be found by applying the above expression and can be written as:

$$\begin{aligned} \lambda(0,b=0) &= \log \sum_j \left(p_j(y | x_j(0,1)) \right) \\ \lambda(0,b=1) &= \log \sum_j \left(p_j(y | x_j(0,1)) \right), \\ LLR(0) &= \lambda(0,b=1) - \lambda(0,b=0) \end{aligned}$$

where $j=0 \dots 2^{m-1}$. The number of values $LLR(0)$ equals the length of the received sequence Y divided by the number of bit positions.

Using Quadrature Pulse Shift Keying (QPSK) constellation as an example, Figure 24 illustrates the signal set that is used to find the BMs and LLRs for receiving sequence

points. In this figure, let the received point from messages Y is represented as a white circle, and the associated transmitted signal points are represented as white squares. To calculate the BM for level 0, it is necessary to find the Euclidean distances for Y to points 00 and 10 for collecting the values of conditional probabilities. Then take the logarithm for the sum of the two conditional probabilities to find $\lambda(0,b)$. Same procedure can be applied for the case where the Euclidean distances are obtained to the points 01 and 11 . Finally the LLR is found by taking $\lambda(0,b=1)$ found from the upper right structure and subtracted that by $\lambda(0,b=0)$ from the upper left corner. Similarly, the LLR for level 1 can be computed in the same way. QPSK have two sets of bit metrics for each bit position level and two LLRs found from these levels. The equations used for computing

Figure 24. Signal Sets used to find LLR for QPSK.

the $LLR[0]$, $LLR[1]$ for QPSK are explicitly shown in next page. From equation below,

$$\lambda(0,0) = \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \left(\exp \left(\frac{-((y_I - 1)^2 + (y_Q - 1)^2)}{2\sigma^2} \right) + \exp \left(\frac{-((y_I + 1)^2 + (y_Q + 1)^2)}{2\sigma^2} \right) \right) \right]$$

$$\lambda(0,1) = \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \left(\exp \left(\frac{-((y_I + 1)^2 + (y_Q - 1)^2)}{2\sigma^2} \right) + \exp \left(\frac{-((y_I - 1)^2 + (y_Q + 1)^2)}{2\sigma^2} \right) \right) \right]$$

$$LLR[0] = \lambda(0,1) - \lambda(0,0)$$

the received message bit Y is represented as (Y_I, Y_Q) , and the four points used to find all LLRs have in-phase and quadrature values of $(+1, +1)$, $(+1, -1)$, $(-1, -1)$, $(-1, +1)$.

Similarly, $\lambda(1,0)$, $\lambda(1,1)$, and $LLR[1]$ are represented as the following:

$$\lambda(1,0) = \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \left(\exp \left(\frac{-((y_I + 1)^2 + (y_Q + 1)^2)}{2\sigma^2} \right) + \exp \left(\frac{-((y_I - 1)^2 + (y_Q + 1)^2)}{2\sigma^2} \right) \right) \right]$$

$$\lambda(1,1) = \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \left(\exp \left(\frac{-((y_I + 1)^2 + (y_Q - 1)^2)}{2\sigma^2} \right) + \exp \left(\frac{-((y_I - 1)^2 + (y_Q - 1)^2)}{2\sigma^2} \right) \right) \right]$$

$$LLR[1] = \lambda(1,1) - \lambda(1,0)$$

For 32-ary QAM constellation, the demodulator produces five sets of LLR values and six sets for 64-ary QAM. These values are then processed by the iterative binary decoder to estimate the original message.

The error correcting code used in this thesis is a powerful low-density parity check (LDPC) codes with length $n = 4095$ and rate $R = 0.82$. Then the total number of LLRs collected from all the bit positions is 4095. Figures 25 and 26 again demonstrate the signal sets used to find BM and LLR for the first bit position for 32- and 64-ary QAM associating with Gray labeling as the examples. In both figures the white square stands

for the first bit position with binary value of 1 and each black dot represents the first bit with binary value of 0.

Figure 25. Signal Sets used to find LLR(0) for 32-ary QAM using Gray Mapping.

Figure 26. Signal Sets used to find LLR(0) for 64-ary QAM using Gray Mapping.

Since the conditional probability is a function of the signal-to-noise ratio, SNR, it decides the LLR behavior. As the SNR is increasing, the LLR is reaching close to very large value. To consider the LLR values over the In-Phase projection axis from the M -ary signal constellation provides the graphical evidence to this characteristic. This indicates that at high SNR, the LLR tends to reach its maximum (i.e., $\lambda(i, b=1) - \lambda(i, b=0) = \pm\infty$). In other words, the probability of getting a binary value of either 1 (or 0) corresponding to that particular bit position of the received message is approaching zero (or one).

Therefore, highly reliable LLR values between different levels for a constellation imply the waste of soft-demapping, to consider every LLR or BM for every received and transmitted point is unnecessary. This factor is supported by the binary channel capacity behavior from chapter 3. Some levels from the constellation are reliable at high SNR such that the LLRs from those levels take a very high value. The following figures illustrate this idea; Figures 27 through 30 show the LLR behavior over the in-phase axis for 64-ary QAM from SNR = 10 dB to SNR = 30 dB. Along with these figures, we insert graphical structures indicating bit position number 4 and 3 as examples to demonstrate this important factor. Some more examples regarding to LLR behaviors for 32- and 64-ary QAM will be shown again later.

Figure 27. Signal sets that used to find LLRs for 4-th bit of 64 QAM.

In Figure 27, for 64 QAM, the underlined bit position $M\{b[5]b[4]\underline{b[3]}b[2]b[1]b[0]\}$ considered for BM and LLR calculations are indicated in gray circles. For $b[3] = 0$, the symbol positions have the values of -7, -5, +5, and +7 over the in-phase axis. For those symbols containing binary values of 1 take the other of the available positions over the in-phase axis (i.e., -3, -1, +1, and +3). One of the LLRs can be represented as:

$$LLR_{In-Phase,j} = \log \left\{ \frac{\exp \left[-\frac{|Y_{In-Phase,j} + 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} + 1|^2}{2\sigma^2} \right]}{\exp \left[-\frac{|Y_{In-Phase,j} - 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 1|^2}{2\sigma^2} \right]} \right\} - \log \left\{ \frac{\exp \left[-\frac{|Y_{In-Phase,j} + 7|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} + 5|^2}{2\sigma^2} \right]}{\exp \left[-\frac{|Y_{In-Phase,j} - 7|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 5|^2}{2\sigma^2} \right]} \right\},$$

where $LLR_{In-Phase,j}$, and $Y_{In-Phase,j}$ denote the in-phase values for the j -th receiving bit to the transmitted bit positions containing different binary values. For the received message Y over the in-phase axis from -8 to +8, we compute the LLR over the range of SNR values 10dB, 15dB, 20dB, 25dB, and 30dB. Figure 28 in the following page demonstrates the computation results for LLR over the in-phase axis between different SNR values. When SNR = 10 dB, the LLRs representing the binary value of 0 are around 0 to -80, where the LLR values are within the range between 0 to +80 for those receiving bits containing binary value of 1. As the values of SNR increase, the probabilities of getting the binary value of 0 or 1 become relatively large or small; therefore, the LLR will reach closely to positive or negative infinity as shown in Figure

28 at SNR equal to 30dB. According to this figure, at high SNR, the most likely cases for the receiving bits from message Y have in-phase values of -5, -3, +3, and +5.

Figure 28. LLR Behaviors vs. In-Phase Projection for 4-th bit of 64 QAM.

Figure 29. Signal sets that used to find LLRs for the 5-th bit of 64 QAM.

Figure 29 illustrates the signal sets with their LLR behaviors for the fifth bit position using 64 QAM. And the LLR are calculated as following:

$$LLR_{In-Phase,j} = \log \left\{ \frac{\exp \left[-\frac{|Y_{In-Phase,j} + 7|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} + 5|^2}{2\sigma^2} \right]}{\exp \left[-\frac{|Y_{In-Phase,j} + 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} + 1|^2}{2\sigma^2} \right]} \right\} - \log \left\{ \frac{\exp \left[-\frac{|Y_{In-Phase,j} - 7|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 5|^2}{2\sigma^2} \right]}{\exp \left[-\frac{|Y_{In-Phase,j} - 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 1|^2}{2\sigma^2} \right]} \right\}$$

Again, the LLR behaviors are shown in Figure 30, at relatively high SNR (i.e., SNR = 30 dB), where the most likely case for receiving message Y containing the binary value 0 or 1 have values of -1 and +1 over the in-phase axis.

The figure above demonstrates the signal sets used for computing the LLRs for the case of 32 QAM. To consider the MSB for 32 QAM, the LLRs can be calculated regardless of some signal points containing different binary values within the same region and it can be represented as the following:

$$LLR_{In-Phase,j} = \log \left\{ \exp \left[-\frac{|Y_{In-Phase,j} + 5|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 5|^2}{2\sigma^2} \right] \right\} - \log \left\{ \exp \left[-\frac{|Y_{In-Phase,j} + 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} + 1|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 3|^2}{2\sigma^2} \right] + \exp \left[-\frac{|Y_{In-Phase,j} - 1|^2}{2\sigma^2} \right] \right\},$$

so that the LLR behaviors are shown in Figure 32. The most likely regions that the received messages occur over the in-phase axis are from -5 to -3 and from 3 to 5 with the LLR values greater than absolute value of 600.

Figure 32. LLR Behaviors vs. In-Phase Projection for 5-th bit of 32 QAM.

4.3 Hard-demapping

As mentions previously, the demapping is another factor influencing the complexity of a BICM demodulator. The demapping process that returns the exact LLR values is called here soft LLR. The demapping process is also known as soft-demapping and it determines the decoding schedule of an iterative decoder for BICM. To reduce the decoding steps for BICM, some LLR values from some levels do not need to be calculated. Instead, due to their high reliability, the LLRs from some levels are assigned a fixed value and the variables corresponding to those bits are called the hard-decision LLRs. The demapping algorithm for those highly reliable levels is called Hard-demapping, and constitutes the main idea of this thesis.

This alternate demapping algorithm helps not only to improve the decoding effort but this is also related to the analytical results obtaining from sections 3.3 and 4.2. Before this idea is defined, it is necessary to revisit the binary channel capacity. Referring back to section 3.3, the binary channel capacity shows different behaviors for every single bit position associated with M-ary constellation with a given labeling. When the binary channel capacity limit is reached at certain SNR (i.e., the binary channel capacity is equal to 1 bit/symbol-channel), the transmitted message bit for that particular bit position becomes highly reliable. Monitoring the binary channel capacities for bit position 4 in Figure 20 and bit positions 4 and 5 in Figure 22 allows finding a particular value of SNR as a reference point for BM calculation.

By looking at the LLR behaviors shown in Figures 28, 30, and 32, the curves for LLR values vs. the in-phase projection at level 4 for both 32 and 64 QAM are reaching

closely to the positive or negative infinity as SNR increases. The characteristic for these curves also shows every j -th bit received at these levels are highly reliable (i.e., $p_j(y|x_j(b=1 \text{ or } b=0)) \approx 0$). Therefore, the LLR values for those bits in correspondence to highly reliable levels are given a largest value available equivalent to positive or negative infinity as if making hard-decisions for these bit levels. We refer to the LLRs associated with these hard-decision levels as hard-demapping. The binary channel capacity behaviors in Figures 20 and 22 can help to determine which levels are suitable for hard-demapping associating with the LLR behaviors. Referring to those figures, we apply hard-demapping on level 4 (i.e., the 5-th bit position) for 32 QAM on levels 4 and 5 for 64 QAM, since these are the most reliable levels compared to others. Thus the hard-demapping process is to assign a fixed value of LLR for level 4 of 32 QAM and for levels 4 and 5 of 64 QAM.

The next thing is to determine what value is better suited to be assigned for these reliable bit position levels. When the LLR reaches an infinite value at certain high SNR, it is almost impossible to send this value to the decoder since to evaluate the infinite number makes no sense to the iterative decoder at the receiver. It is convenient to rather assign a value replacing the positive and negative infinities. There are two steps to come up with a number; we first revisit the LLR behaviors obtained from Figures 28, 30 and 32. At SNR = 30 dB, we take the quantized values for the LLR curve for each case. Then the positive and negative boundary (i.e., +1 and -1 corresponding to the binary values of 0 or 1) appears as a guide line toward the different binary value for the corresponding high reliable bit positions. The second step is to take product of these quantized values with

an integer 30. Note that this number is just right for the hard-demapping since the quantity $e^{\pm 30}$ is larger enough to represent the LLR. However, this number can be modified to a more precise or to be proven as a perfect number to fit this case. Thus, the figures below demonstrate some configurations for assigning this hard-demapping value using 32- and 64-ary QAM constellations from different bit positions. Appendix C illustrates another example of LLR behavior at SNR equals to 30 dB for 64 QAM associates with SP labeling for the purpose of comparison.

Figure 33. Actual vs. Quantized LLRs for the 5-th bit of 32 QAM at SNR = 30 dB.

Figure 34. Actual vs. Quantized LLRs for the 5-th bit of 64 QAM at SNR = 30 dB.

Figure 35. Actual vs. Quantized LLRs for the 4-th bit of 64 QAM at SNR = 30 dB.

Similar to Figure 2, Figures 36 and 37 illustrate the hard-demapping idea regarding to BICM for the cases of 32- and 64-ary QAM.

Figure 36. Hard-demapping used for 32-ary QAM at BICM Receiver.

Figure 37. Hard-demapping used for 64-ary QAM at BICM Receiver.

Referring to the above figures, the LLR values are computed and delivered at the output of the bit metric generator. As mentioned before, those bits from a highly reliable level are assigned values equal to -30 or +30 (or the largest value available). The way to come up with this number on these figures was discussed previously. At the same time,

other LLRs are combined together to form a series of sequence and then passed to an iterative decoder. In this work, we consider LDPC codes with iterative decoding; such that the decoding complexity can be further reduced inside the decoder: Code nodes associated with the hard decisions (i.e., hard-demapping) do not perform message computation but rather send always the maximum value (+30 or -30 in our case). These LLRs associated with the hard decision levels are defined as “hard” LLRs. The other LLR values are directly passed to the decoder. The number of operations run by the iterative decoder was measured and shown in the following figures. In addition, the error performance of this algorithm can be explained by considering the comparisons between soft- and hard-decision capacities, as discussed in the next chapter.

Figure 38. The Complexity inside the Iterative Decoder vs. SNR by applying Hard-demapping on different levels for 32-ary QAM.

In Figure 38, when there is no hard-demapping on any level for 32 QAM, the average number of operations within the decoder using LDPC codes is roughly 3.4 million per codeword. Operations are defined as additions and multiplications between code nodes and LLRs. Note that when the SNR goes to higher values, the complexity starts to reduce due to the appearance of highly reliable bits as described before. For example, at SNR = 20 dB, the number of operations for 32 QAM decoder without hard-demapping is about 3.353 million and it reduces to 3.335 million when SNR reaches 30 dB. The improvement is about 0.54%. However, at 20 dB, while hard-demapping is applying on the most significant bit level (i.e., b4) for 32 QAM, the complexity is further reduced by 20%. Referring to Figure 38, the more levels associate with hard-demapping, the more improvement for the decoding complexity can be made. Figure 39 illustrating the evidence supports this idea in the case of 64-ary QAM. If we consider the number of operations per iteration at SNR = 25 dB, then the decoder complexity that the hard-demapping used on 3 levels can be reduced by 44% compared to soft-demapping applied on all levels. Again, the more levels involving the hard-demapping algorithm, the more efficient that the iterative decoder performs.

Figure 39. The Complexity inside the Iterative Decoder vs. SNR by applying Hard-demapping on different levels for 64-ary QAM.

4.4 Conclusion

The way to handle the demapping process at a BICM receiver and a proposed alternate demapping process are introduced in this chapter. There are two steps in the demapping process, bit metric generation and log-likelihood ratio computations. This concept was discussed in section 4.2 where the bits metric was obtained by applying the conditional probability between received and transmitted sequences and are related to the capacity estimation. The LLR can be found from the BM.

After discussing the demapping process, the idea of the hard-demapping was defined in section 4.3 since this idea helps to reduce the decoding effort such that the

complexity of demodulation and decoding in BICM can be further reduced. The key for hard-demapping is to assign a fixed LLR value and passed that to the iterative decoder so that some of the messages processed in this device require no further computation.

Examples were depicted in figures to demonstrate how to compute the maximum LLR value by analyzing the LLR behavior in the cases of 32- and 64-ary QAM. To support this idea, the complexity measured in number of decoding operations versus signal to noise ratio was illustrated in several figures. Those figures indicate that with hard-demapping, the decoder complexity can be reduced dramatically by 20% and 40% in contrast to the soft-demapping algorithm on all levels for M-ary signal constellations.

However, the binary channel capacity behaviors determine which levels are suitable for applying hard-demapping. Therefore, in section 4.3, the selection of levels associated with hard-demapping for both 32- and 64-ary QAM constellations was discussed. Again the complexity versus SNR for those bit levels using the hard-demapping were demonstrated in Figures 38 and 39. The application of hard-demapping may reduce the complexity for BICM, but it might have a trade off with respect to error performance. To explore the range of this loss with different approaches, the hard-decision and soft-decision capacities must be involved. Also, some preliminaries for applying this alternative will be discussed in the following by illustrating simulation results on bit error rate performance. This is considered in the next chapter.

5. NUMERICAL RESULTS AND DISCUSSION

5.1 Preliminaries

In this chapter, the error performances with and without hard-demapping are presented. Before the simulation results are shown, comparisons are presented regarding the loss from the view of capacities BICM without and with hard-demapping applied. To distinguish from the idea of multilevel coding, the decoding process for BICM handles messages in series. In other words, the system considers a stream of message bits without having to separate them into different levels or to divide the codeword in individual blocks. By considering the loss using the approach of capacity, lower bounds for the trade off regarding to the proposed demapping are obtained. Thus, for both 32 and 64 QAM cases, we plot the total modulation-constrained capacities (i.e., the sum of the component binary channel capacities for all levels) associating with all soft-decision versus the total capacity with soft/hard mixed decisions. If this difference is small, then to achieve a relatively small loss in error performance is possible.

The error correcting code used for BICM simulation with hard-demapping is an LDPC code with the code length $n = 4095$ and code Rate $R = 0.82$. Similar to comparisons presented in (Washmann et. al. 1380), Figures 40 and 41 illustrate the total modulation-constrained capacities between soft and soft/hard-decisions for both 32 and 64 QAM modulations with Gray mapping. As indicated in (Washmann et. al. 1367, 1380), the hard-decision capacity can be interpreted as finding the channel capacity in binary symmetric channel and it can be given by

$$CH[m]=1-H_2(\varepsilon_m),$$

Where the function H_2 is the binary entropy function and the variable ε_m is the bit error probability of the equivalent m -th binary symmetric channel. The binary entropy function given as:

$$H_2(x)=-x\log_2 x-(1-x)\log_2(1-x).$$

The hard-decision binary channel capacities for level 4 in 32 QAM and levels 4 and 5 in 64 QAM are obtained using above two equations. Then these hard-decision channel capacities are added to the soft-decision channel capacities for the first four levels (i.e., bit 0 to bit 3 for both 32 and 64 QAM) found from chapter three to obtain the total soft/hard-decision channel capacities.

The notations for both figures are as follows: C(S) stands for total modulation-constrained capacity using soft-decisions for all levels, where C(S/H) represents modulation-constrained capacity using soft-decision mixed with hard-decision decoding. Each component binary capacity using soft-decision is labeled as CS[m], where m is the number of levels as same as defined previously. Referring to Figures 40 and 41, the bit position using soft-decision capacities are CS0, ..., CS3 for both QAM cases. Each binary channel capacity using hard-decision has a label CH[m]. In this thesis, hard-decision capacity is used for level 4 in 32 QAM (CH4) and levels 4 to 5 for 64 QAM (CH4, CH5).

Figure 40. Total Constrained Soft-decision Capacity vs. Total Constrained Soft/Hard-decision Capacity for 32-ary QAM.

For the soft- and hard-decision capacities comparison in 32-ary QAM case, every fifth bit (MSB) for 32 QAM using both labeling scheme is performed a hard decision at the capacity of $(5\text{bits/symbol}) \times (0.82\text{info bits/coded bits}) = 4.1\text{info bits/symbol}$, the expected loss with the hard decision decoding for the fifth level is about 0.7 dB for Gray mapping. As for the case of 64-ary signal constellation, at the capacity that corresponding to $(6\text{bits/symbol}) \times (0.82\text{info bits/coded bits}) = 4.92\text{info bits/symbol}$, the expected loss with the hard-decision decoding for the fifth and sixth levels is about 0.6 dB for 64-ary QAM using Gray labeling.

Figure 41. Total Constrained Soft-decision Capacity vs. Total Constrained Soft/Hard-decision Capacity for 64-ary QAM.

5.2 Bit Error Rate Performance

The model with an LDPC code was used to run the simulation for the hard demapping for 32- and 64-ary QAM BICM schemes as mentioned several times throughout the thesis. In addition, the length for interleaver and the deinterleaver within the BICM is 3 and with the LLR values for the corresponding bit positions set to +30 and -30 (i.e., every fifth bit has LLR of +30 or -30 for 32 QAM, and every fifth and sixth bit has LLR with the same value in the case of running the simulation for 64 QAM). The following figures illustrate the comparisons between simulations results toward the error

performances of running the ordinary BICM without hard-demapping versus the one with the hard demapping applied.

Figure 42. Simulation Result for the 32-ary BICM vs. the Simulation with Hard Demapping.

The above figure shows the bit error rate comparisons between the BICM with and without the hard-demapping applied for the case of 32-ary QAM constellation. While considering the bit error rate of 10^{-4} , the trade off is approximately 2.2 dB. This loss comparing to what was shown in Figure 42, the BER difference based on this figure could be $10^{(2.2-0.7)/10} = 10^{0.15}$. Meanwhile, for the case of 64QAM shown in Figure 43, the gap between the BICM without and with hard demapping is 2.6 dB at the point of BER equal to 10^{-4} .

Figure 43. Simulation Result for the 64-ary BICM vs. the Simulation with Hard Demapping.

The simulation results shown in this section indicate that the losses for hard-demapping of roughly from 2.2 to 2.6 dB. Even though the losses are larger than what are shown in Figures 42 and 43. The gaps in capacities with soft-decision plus hard-decision predict the performance of multilevel codes with the LDPC code; however, for the BICM in this case, these gaps from Figures 40 and 41 are lower bounds. Namely, errors in hard-decision bits are mixed with those soft-decision bits so that the whole codeword is affected. Figures 44 and 45 illustrate this idea between multilevel codes and BICM. In Figure 44, the bits gathering from hard-demapping are presented in gray and they are equally distributed within the m -th channel. Therefore, the errors do not

propagate with other bit transmitted between other levels into the decoder. In Figure 45, the hard-demapping bits are again presented as boxes in gray color and transmitted into the decoder mixing with other soft-demapping bits so that the chances of getting more errors are larger than the one shown in Figure 44. This is the main reason explaining losses from the view of BER are larger than the lower bonds demonstrating in Figures 40 and 41. In other words, these tradeoffs of the error performances are within a reasonable range.

Figure 44. The Error Propagates through the Codeword for Multilevel Codes.

Figure 45. The Error Propagates through the Codeword for BICM.

Figure 46. The Complexity for Iterative Decoder over SNR for 32 QAM with and without Hard-demapping on MSB Level.

Figure 47. The Complexity for Iterative Decoder over SNR for 64 QAM with and without Hard-demapping on the 4-th and the 5-th level.

The complexities inside iterative decoder versus SNR between BICM with and without hard-demapping are again illustrated in Figures 46 and 47 for 32 and 64 QAM.

Summarizing the observed results from Figures 42, 43, 46, and 47, the 32-ary constellation using hard-demapping on MSB level helps to reduce the decoding complexity by 20% in average with a loss of 2.2 dB for error performance. While in 64 QAM case with hard-demapping technique on levels 4 and 5, there is a 2.6 dB trade off in BER and the reduction in decoding complexity is 30%.

6. CONCLUSION AND FINAL REMARK

For BICM scheme over AWGN, the process of demapping associated with the bit-metric computation was introduced. The cases considered in this work were BICM using a 32-ary cross constellation and BICM using a 64-ary square constellation for transmission over an AWGN channel. First the total unconstrained capacity was computed; so that levels can be selected for which hard-decision (HD) decoding can be applied with a relatively small penalty in performance. The demapper is required to compute every single bit-metric value to produce the LLR values. However, the LLR values corresponding to those highly reliable levels where HD can be applied are set directly to the largest possible value, with the sign defined by a hard decision value (i.e., +1 or -1). For the other bits, the LLR values need to be computed and passed to the decoder.

The advantage of this approach is that both the complexity of the metric computation and the complexity within the decoding process can be reduced. By comparing of total constrained capacity and binary channel capacities the loss in performance can be predicted relatively small. Thus, with the utilization of the proposed demapping algorithm, a power-efficient BICM scheme over AWGN channel can be implemented together with powerful ECC schemes such as LDPC codes. A future task is to focus on the values of LLR so that the hard-demapping can be improved by considering more than two values.

REFERENCES

- Caire, Giuseppe, Giorgio Taricco, and Ezio Biglieri. "Bit-Interleaved Coded Modulation." IEEE Transaction on Information Theory 44.3 (1998) : 927-946.
- Gallager, Robert G. Low Density Parity Check Codes, Monograph. Cambridge: MIT Press, 1963.
- Haykin, Simon. Communication System. 4th ed. New York: John Wiley & Sons, Inc., 2001.
- Le Goff, Stéphane Y. "Signal Constellation for Bit-Interleaved Coded Modulation." IEEE Transaction on Information Theory 49.1 (2003) : 307-313.
- McIllree, Philip Edward. Channel Capacity Calculations for M-ary N-dimensional Signal Sets. Thesis. The University of South Australia, School of Electronic Engineering, 1995.
- Schulze, Henrik. "System Design for Bit Interleaved Coded Square QAM with Iterative Decoding in a Rician Fading Channel." European Transactions on Telecommunications 14.2 (2003) : 119-129.
- Tsai, Taylor T. and Robert H. Morelos-Zaragoza. "On Channel Capacity, Constellation Labeling and Demapping for BICM using 32-ary Constellation". *Proceedings of the Seventh International Symposium on Wireless Personal Multimedia Communications*. 12-15 Sept. 2004. Ed. T. Tsai & R. H. Morelos-Zaragoza. Abano Treme, Italy: Wireless Personal Multimedia Communications, 2004. V3-31-V3-33.

- Tsai, Taylor T. and Robert H. Morelos-Zaragoza. "On Channel Capacity, Constellation Labeling and Demapping for BICM using 32- and 64-ary Constellations". *Proceedings of the eighth International Symposium on Information Theory and its Applications*. 10-13 Oct. 2004. Ed. T. Tsai & R. H. Morelos-Zaragoza. Parma, Italy: The Society of information Theory and its Applications, 2004. 1479-1484.
- Ungerboeck, Gottfried. "Channel Coding with Multilevel/Phase Signals." IEEE Transaction on Information Theory IT-28.1 (1982) : 55-67.
- Washmann, Udo, Robert F.H. Fischer, and Johannes B. Huber. "Multilevel Codes: Theoretical Concepts and Practical Design Rules." IEEE Transaction on Information Theory 45.5. (1999) : 1361-1391.
- Wesel, Richard D., Xueting Liu, John M. Cioffi, and Christos Komninakis. "Constellation Labeling for Linear Encoders." IEEE Transaction on Information Theory 47.6 (2001) : 2417-2431.

APPENDICES

APPENDIX A. Graphical View for Gray Mapping using 64-ary QAM constellation.

64 QAM Constellation

APPENDIX B. Graphical View for Set-Partitioning Labeling using 8PSK Constellation.

APPENDIX C. The First Set used to Find LLR for Set-Partitioning Labeling using 64-ary QAM Constellation with LLR Behaviors.