SIGNAL MAPPING FOR BIT-INTERLEAVED CODED MODULATION WITH ITERATIVE DECODING

by

Hassan Mohammadnavazi

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The College of Graduate Studies

(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2018

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis/dissertation entitled:

Signal Mapping for Bit-Interleaved Coded Modulation with Iterative Decoding

submitted by Hassan Mohammadnavazi in partial fulfillment of the requirements of

the degree of Doctor of Philosophy.

Dr. Md. Jahangir Hossain, School of Engineering

Supervisor

Prof. Julian Cheng, School of Engineering

Supervisory Committee Member

Dr. Thomas Johnson, School of Engineering

Supervisory Committee Member

Prof. Shawn Wang, Irving K. Barber School of Arts and Sciences

University Examiner

Dr. Nghi Tran, Electrical and Computer Engineering, University of Akron, OH, USA

External Examiner

Abstract

Due to its convenience, wireless communication systems have grown tremendously. Data rate requirements of various services/applications that are transmitted over wireless channels are increasing day by day. Bit interleaved coded modulation (BICM), which is a serial concatenation of a channel encoder, an interleaver and a symbol mapper, is a spectral efficient technique that is being used in many wireless systems. The performance of BICM can be significantly improved by using the iterative decoding technique at the receiver. This system is referred as BICM with iterative decoding (BICM-ID), and offers an improved performance over both fading and nonfading channels. It is well-known that BICM-ID performance is strongly dependent on the applied signal mapping. Signal mapping is the assignment of binary bits to complex symbols from a modulation alphabet. It is demanded to use a higher order modulation in BICM-ID to achieve a higher data rate and spectral efficiency. However, finding a mapping for a large modulation that offers an improved performance for BICM-ID is very complicated. This is because of the huge number of possible mappings for higher order modulations.

This thesis focuses on the mapping problem for BICM-ID systems. In particular, novel mapping methods are developed for higher order modulations, including two-dimensional and multi-dimensional modulations. The proposed methods in this thesis consist of (i) heuristic methods and (ii) computer search techniques. In comparison with the previously known mappings, the proposed mappings significantly improve the BICM-ID performance over the considered channels. This is confirmed by various analytical and simulation results that are investigated in this thesis.

Lay Summary

Today's wireless networks should be designed to support widely varying user needs (home, office, etc.). The obvious trend is the need of very high data transmission rates to deal with the increasing demand of multimedia communication services such as video teleconferencing and high quality media streaming. The bit interleaved coded modulation with iterative decoding (BICM-ID) system offers excellent performance on the both wired and wireless communication channels. As such, it is a great candidate to be used in future wireless systems. The technical challenge is to enable high data transmission rates for BICM-ID. One main approach is to increase the number of the unique signals that BICM-ID uses to transfer information. In this approach, the assignment of information to the unique signals is a very crucial and complex problem. This problem is addressed in this thesis such that, the system performance improves without any increase in the system complexity.

Preface

The research chapters of this thesis (Chapters 2-5) have been published in or submitted to peer reviewed journals.

A version of Chapter 2 has been published as: Hassan M. Navazi and Md. J. Hossain, "A novel symbol mapping method for BICM-ID systems for higher order signal constellations," *IEEE Commun. Lett.*, Vol. 18, no. 8, pp. 1323-1326, Aug. 2014. I conducted all the mathematical analysis and computer programming. I wrote the manuscript, and Dr. Md. J. Hossain revised and edited it to improve the presentation in the manuscript.

A version of Chapter 3 has been published as: Hassan M. Navazi and Md. J. Hossain, "Efficient multi-dimensional mapping using QAM constellations for BICM-ID," *IEEE Trans. on Wireless Commun.*, Vol. 16, no. 12, pp. 8067-8076, Dec. 2017. I conducted all the mathematical analysis and computer programming. I wrote the manuscript, and Dr. Md. J. Hossain revised and edited it to improve the presentation in the manuscript.

A version of Chapter 4 has been submitted to the journal IEEE Transactions on Wireless Communications as: Hassan M. Navazi and Md. J. Hossain, "Multi-dimensional mapping of higher order QAM constellations for BICM-ID over Rayleigh fading channels," Jan. 2018. I conducted all the mathematical analysis and computer programming. I wrote the manuscript, and Dr. Md. J. Hossain helped me improve the presentation in the manuscript by his comments.

A version of Chapter 5 has been submitted to the journal IEEE Transactions on Wireless Communications as: Hassan M. Navazi and Md. J. Hossain, "Multi-Dimensional QAM mapping for BICM-ID over AWGN channels," Feb. 2018. I conducted all the mathematical analysis and computer programming. I wrote the manuscript, and Dr. Md. J. Hossain helped me improve the presentation in the manuscript by his comments.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BICM	Bit Interleaved Coded Modulation
BICM-ID	Bit Interleaved Coded Modulation with Iterative Decoding
BSA	Binary Switching Algorithm
CSI	Channel State Information
GA	Genetic Algorithm
LLR	Log-Likelihood Ratio
MD	Multi-Dimensional
MSED	Minimum Squared Euclidean Distance
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RTS	Reactive Tabu Search
SED	Squared Euclidean Distance
SNR	Signal to Noise Ratio
χ	Signal constellation
N_{min}	Average Hamming distance between neighboring symbols
Φ	Harmonic mean of MSED
$\hat{\Phi}$	Harmonic mean of MSED after feedback
$W(\boldsymbol{l})$	Hamming wight of binary sequence $\boldsymbol{l}$
$d_H(\boldsymbol{l}_1, \boldsymbol{l}_2)$	Hamming distance between binary sequences $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$
$sgn(.)$	Sign function

Acknowledgment

I would like to thank my supervisor, Dr. Jahangir Hossain, for his excellent guidance and invaluable advice. He has always been supportive, encouraging, and patient. He is not only a professional supervisor, but also a good friend. I cannot thank him enough for the assistance he provided at all levels of my PhD study.

I also thank my thesis committee, Dr. Julian Cheng and Dr. Thomas Johnson, for their helpful comments and suggestions. My sincere gratitude also goes to Dr. Ha H. Nguyen, University of Saskatchewan and Dr. Robert G. Maunder, University of Southampton, from whom I learned priceless points regarding my research before my attending UBC. I owe particular thanks to Lori Walter, Centre for Scholarly Communication, UBC Okanagan, for her help and support.

My special thanks also go to all my family members and friends for their support.

Dedication

To my parents, Golzar and Panjali

Chapter 1

Introduction and Overview

1.1 Introduction

In general, data rate requirements of various services/applications that are transmitted over communication networks including wired and wireless networks are increasing day by day. Higher order modulations in conjunction with efficient channel coding will play a vital role to meet this high data rate requirement.

Bit interleaved coded modulation (BICM) was introduced by Zehavi in [1] as an attractive coded modulation scheme. In BICM, channel coded bits are randomly interleaved and then the interleaved coded bits are mapped to the signals at the modulator. Due to its simplicity and design flexiblity, BICM has been standardized for contemporary wireless and wired communication systems [2] and is a potential choice for future communication systems.

BICM offers a good error performance over the Rayleigh fading channel. However, the employed interleaver results in a random modulation, which degrades BICM performance over the additive white Gaussian noise (AWGN) channel. One effective way to overcome this problem is to use iterative decoding at the receiver. This system is known as BICM-ID, which is investigated in [3]-[5]. Because of the iterative decoding, BICM-ID offers an impressive performance over the AWGN channel as well as over the Rayleigh fading channel [6].

The performance of BICM-ID is significantly dependent on the employed signal mapping at the modulator. Signal mapping is defined as the assignment of binary digits to complex signals (symbols) from a modulation alphabet. Multi-dimensional (MD) mapping improves bandwidth efficiency for BICM-ID by a reasonable increase in system complexity [7]. In MD mappings, a sequence of bits is mapped to a sequence of symbols instead of a single symbol. Many research studies have been carried out to address the signal mapping problem for BICM-ID, see for examples, [7]-[24].

The two main methodologies to develop signal mappings for BICM-ID are as follows: (i) heuristic methods and (ii) computer search techniques. The existing heuristic methods construct good 2-D/MD mappings for only smaller modulations such as 2-D 32-ary quadrature amplitude modulation (QAM) [22] and MD 8-ary phase shift keying (PSK) [7]. For larger modulations, computer search techniques are usually used. However, the achieved results from the existing computer search techniques are suitable only for modulations with medium sizes such as 2-D 64-QAM [13].

Higher order modulations can be used to increase the data rate and improves the bandwidth efficiency of BICM-ID. However, developing efficient mappings of larger constellations for BICM-ID is always challenging due to the huge number of possible mappings. Even the best computer search techniques become intractable in finding good mappings of larger constellations for BICM-ID due to the high level of complexity [16]. In [16] and [25], the authors demonstrated that random mapping can lead to efficient MD mappings. Random mapping technique searches among a set of randomly generated mappings to find a mapping that improves the performance of BICM-ID. However, as the set of randomly generated mappings is very large, this technique also suffers from computational complexity. This complexity degrades the obtained random mapping's performance especially for large MD modulations. Consequently, developing efficient mappings of larger constellations (including 2-D and MD constellations) is a very important and open research question.

Motivated by the above discussions, this thesis focuses on improving the error performance of BICM-ID via developing novel and efficient signal mappings. In particular, we investigate the mapping problem for higher order signal constellations including 2-D and MD constellations. To design mapping functions throughout this thesis, two general methodologies will be followed: (i) systematic methods which generate mapping heuristically and (ii) novel and computationally fast computer search techniques.

In the rest of this chapter, we briefly review the BICM and BICM-ID concepts and discuss about the mapping problem for BICM-ID.

1.2 BICM system

A BICM system model is shown in Fig. 1.1, where the transmitter is built from serial concatenation of an encoder, a bit interleaver Π, and a modulator. A sequence of information bits $\boldsymbol{u}$ is encoded by a convolutional encoder. Then, the coded bits $\boldsymbol{c}$ are randomly interleaved, and the interleaved coded bits $\boldsymbol{v}$ are grouped in blocks of mN bits, where m and N are positive integers. For notational convenience, let us denote the t^{th} block of interleaved coded bits at the input of the modulator by $\boldsymbol{l}_t = [l_t^{(1)}, l_t^{(2)}, \cdots, l_t^{(mN)}]$. The modulator maps $\boldsymbol{l}_t$ to a vector of N consecutive 2^m-ary signals, $\boldsymbol{x}_t = [x_t^{(1)}, x_t^{(2)}, \cdots, x_t^{(N)}]$, using a mapping function $\mu : \{0,1\}^{mN} \longrightarrow \boldsymbol{\chi} = \chi^N$, where χ denotes the 2-D 2^m-ary signal set. Mathematically, we can write

$$\boldsymbol{x}_t = [x_t^{(1)}, x_t^{(2)}, \cdots, x_t^{(N)}] = \mu(\boldsymbol{l}_t). \tag{1.1}$$

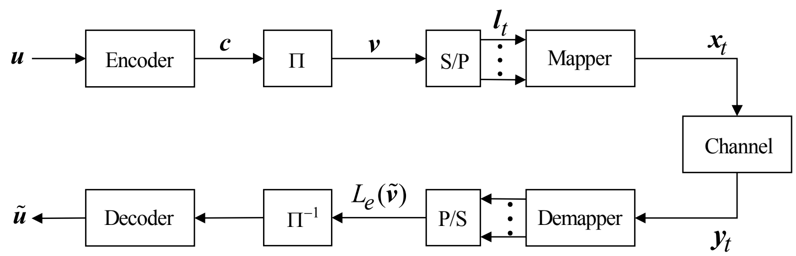

Figure 1.1: The block diagram of a BICM system.

At the receiver, the received signal-vector corresponding to the transmitted symbol-vector $\boldsymbol{x}_t$ can be expressed as

$$\boldsymbol{y}_t = \boldsymbol{h}_t^T \boldsymbol{x}_t + \boldsymbol{n}_t, \tag{1.2}$$

where $\boldsymbol{h}_t = [h_t^{(1)}, h_t^{(2)}, \cdots, h_t^{(N)}]$ is the corresponding vector of Rayleigh fading coefficients, A^T represents the transpose of A, and $\boldsymbol{n}_t$ is a vector of N additive complex white Gaussian noise samples with zero-mean and variance N_0. It is important to note that if $N = 1$, the mapping is a regular 2-D mapping; otherwise, it is referred to as an MD ($2N$-D) mapping.

At the receiver, the demapper calculates the log-likelihood ratios (LLRs) using the received signal as [4]

$$L_e(l_t^i) = \log \frac{\sum_{\boldsymbol{x}_t \in \boldsymbol{\chi}_0^i} \mathrm{P}(\boldsymbol{y}_t | \boldsymbol{x}_t)}{\sum_{\boldsymbol{x}_t \in \boldsymbol{\chi}_1^i} \mathrm{P}(\boldsymbol{y}_t | \boldsymbol{x}_t)}, \tag{1.3}$$

where $\boldsymbol{\chi}_0^i$ and $\boldsymbol{\chi}_1^i$ represent the subset of signals $\boldsymbol{x}_t \in \boldsymbol{\chi}$ whose labels have the bit value of 0 and 1, respectively, in the i^{th} bit position. $\mathrm{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is the probability density function, which is determined by the channel model. The extrinsic LLRs are then deinterleaved and fed to the decoder, where a decision is made about the transmitted bits.

The idea of bit by bit interleaving in BICM improves the diversity order of the system. Diversity order is a highly influential parameter on the bit eror rate (BER) performance of coded systems over the Rayleigh fading channel [26]. As a result, BICM offers a good error performance over the Rayleigh fading channel. However, as mentioned previously, the interleaver results in a random modulation, which degrades the BER performance of BICM over the AWGN channel.

1.3 BICM-ID system

To improve the error performance of BICM, iterative decoding has been used at the receiver [3]-[5]. The resulting system is referred to as BICM-ID. The iterative decoding technique used for the BICM-ID is similar to that of a turbo code [27]. However, BICM-ID employs only one encoder at the transmitter and only one decoder at the receiver. As such, it has considerably less complexity in comparison with turbo codes [4].

The BICM-ID transmitter is the same as the BICM transmitter, which has been explained in section 1.2. Therefore, in what follows, we describe only the receiver of BICM-ID. Fig. 1.2 illustrates the block diagram of a BICM-ID system. It is assumed that the receiver has the perfect channel state information (CSI). At the receiver, the demapper uses the received signal-vector $\boldsymbol{y}_t$ and the *a priori* LLR of the coded bits to compute the extrinsic LLR for each of the bits in the received signal-vector as [4]

$$L_e(l_t^i) = \log \frac{\sum_{\boldsymbol{x}_t \in \boldsymbol{\chi}_0^i} \mathrm{P}(\boldsymbol{y}_t|\boldsymbol{x}_t) \prod_{j=1, j\neq i}^{mN} e^{-L_a(l_t^j).l_t^j}}{\sum_{\boldsymbol{x}_t \in \boldsymbol{\chi}_1^i} \mathrm{P}(\boldsymbol{y}_t|\boldsymbol{x}_t) \prod_{j=1, j\neq i}^{mN} e^{-L_a(l_t^j).l_t^j}}, \tag{1.4}$$

where $L_a(l_t^j) = \log(\mathrm{P}(l_t^j = 0)/\mathrm{P}(l_t^j = 1))$ is the *a priori* LLR of the coded bits. After being permuted by the random deinterleaver, the extrinsic LLRs are applied to the channel decoder. The decoder then calculates the extrinsic LLR values on the coded bits. After being interleaved, these LLRs are fed back to the demapper and used as the *a priori* LLRs in the next iteration. Through this iterative process, BICM-ID achieves a significant coding gain and improves the error performance.

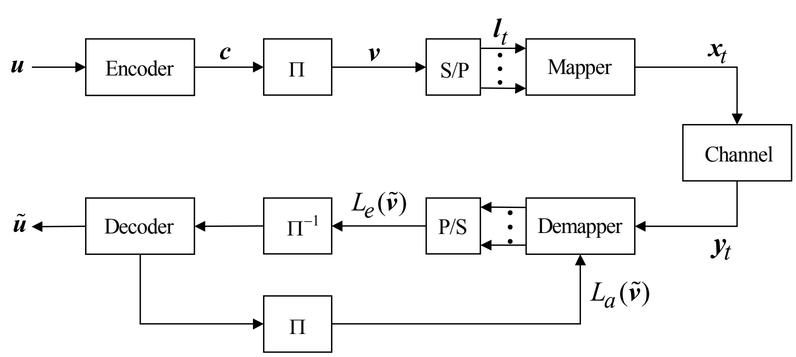

Figure 1.2: The block diagram of a BICM-ID system.

1.4 Problem statement

It is widely known that the performance of BICM-ID strongly depends on the applied signal mapping at the modulator. Signal mapping describes how to assign binary sequences to complex signals from a modulation alphabet. In fact, a constellation with M signal points has $\lambda = M!$ possible mappings, where ! denotes the factorial operation. For example, there are $8! = 40320$ mappings for 8-QAM. Table 1.1 provides the number of possible mappings for 2-D M-QAM for different values of M. It is worth noting that the number of unique mappings for a constellation is smaller than the number of possible mappings. However, finding the unique mappings among the possible mappings for large constellations is prohibitively complicated.

Table 1.1: Number of possible mappings for M-QAM.

Modulation	λ
4-QAM	24
16-QAM	2.1×10^{13}
64-QAM	1.3×10^{89}
256-QAM	8.6×10^{506}
1024-QAM	5.4×10^{2639}

As this table shows, the number of possible mappings for higher order constellations such as 1024-QAM approaches infinity, which makes finding the corresponding good/optimum mappings difficult if not impossible. As a result, exhaustive computer search techniques are not applicable to find suitable mappings of larger constellations. The well-known binary switching algorithm (BSA) [9] is the best known computer-based mapping search technique. However, the BSA becomes intractable to obtain good mappings of larger constellations for BICM-ID due to the high level of complexity [16]. In the case of MD mappings, the problem is much more severe because by increasing the dimensionality, the number of possible mappings increases exponentially. For example, as shown in Table 1.1, while the number of possible mappings for 2-D 64-QAM is 1.3×10^{89}, there are 3.6×10^{13019} possible mappings for 4-D 64-QAM. For example, using In [16] and [25], the authors demonstrated that the random mapping technique can lead to efficient higher dimensional mappings. According to the random mapping technique, computer searching is used to obtain a good mapping from a large set of randomly generated mappings. Selecting a mapping randomly/blindly from a large set and checking its efficiency makes the procedure complex. Moreover, it degrades the resulting mappings' performance especially when the MD modulation is constructed using a large 2-D modulation. Therefore, the problem of 2-D or MD mapping of higher order modulations for BICM-ID has not been solved efficiently yet.

1.5 Thesis objective

BER is the most common metric that is used to evaluate the performance of BICM-ID. Fig. 1.3 illustrates a typical BER curve of BICM-ID after a number of iterative decoding. In this figure, E_b denotes the transmitted energy per bit, N_0 represents the power of noise per Hertz, the threshold E_b/N_0 is the point where BER starts to decrease, the turbo cliff region is the region where the BER curve falls quickly, and the error-floor region is the region where the BER curve is flat at very small values. Two main goals in developing signal mappings for BICM-ID are to achieve a BER curve with (i) a lower error-floor and (ii) an earlier turbo cliff (i.e., smaller threshold E_b/N_0).

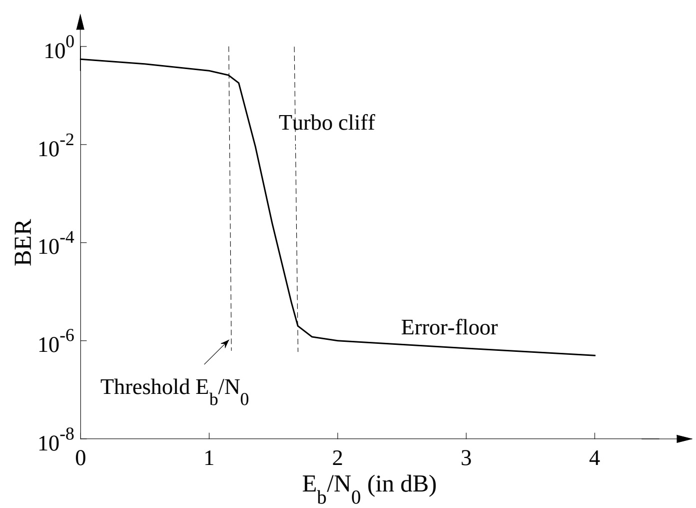

Figure 1.3: A typical BER curve for BICM-ID.

The error performance of BICM-ID at low signal to noise ratios (SNRs) depends on the BER at the first iteration. This is because at low SNRs no coding gain can be achieved from the iterative decoding process. In this case, BICM-ID is equivalent to BICM. Therefore, the optimum mapping for BICM-ID in the low SNR region corresponds to the optimum mapping for BICM. However, the mapping designed for a low SNR region usually offers a poor error-floor for BICM-ID. The mapping designed to minimize the error-floor of BICM-ID results in an extremely low BER at a very high SNR value. Hence, this mapping is not very relevant for practical communication systems, which require a BER of 10^{-3} to 10^{-6}. Moreover, finding such a mapping is computationally expensive. Consequently, designing an efficient MD mapping for BICM-ID that offers good BERs at both low and high SNR values is very desirable.

Motivated by the above discussions, the objective in this thesis is to develop efficient 2-D/MD mapping methods for BICM-ID to achieve good BER performance over AWGN and

Rayleigh fading channels in both low and high SNR regions. In what follows, we investigate mapping design guidelines for AWGN and Rayleigh fading channels. To investigate the guidelines for the low SNR region, we consider BICM-ID performance at the first iteration.

1.6 Mapping design guidelines for BICM-ID

1.6.1 AWGN channel

Low SNR region

Let $\boldsymbol{d} = \{d_1, \cdots, d_p\}$ is the set of all possible Euclidean distances between two signal points in $\boldsymbol{\chi}$, where $d_i < d_j$ if $i < j$, and p depends on the constellation. For example, p takes the value of two and five, respectively, for 2-D and 4-D QPSK. A larger value of d_1 is desired to achieve a better BER performance of BICM over the AWGN channel [28]. Moreover, in order to achieve a good asymptotic BER performance of BICM over AWGN channels, the value of N_{min} should be as small as possible [28], where N_{min} is defined as

$$N_{min} = \frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}_b^i} N(\boldsymbol{x}, i), \tag{1.5}$$

where $\boldsymbol{x} = [x^1, x^2, \cdots, x^N]$ is a $2N$-D signal point, and $N(\boldsymbol{x}, i)$ is the number of signal points at the Euclidean distance d_1 from $\boldsymbol{x}$ that are different from $\boldsymbol{x}$ in the i^{th} bit position.

High SNR region

The asymptotic performance of a mapping for BICM-ID over the AWGN channel depends on $\Phi_a(\mu, \boldsymbol{\chi})$, which is expressed as [15]

$$\Phi_a(\mu, \boldsymbol{\chi}) = \left(\frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}_b^i} \exp\left(- \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2}{4N_0} \right) \right)^{-1}, \tag{1.6}$$

where $\hat{\boldsymbol{x}} = [\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^N]$ is different from $\boldsymbol{x}$ only in the i^{th} bit position and $\|\mathbf{A}\|$ is the Euclidean norm of $\mathbf{A}$. To achieve improved asymptotic performance, a greater value of $\Phi_a(\mu, \boldsymbol{\chi})$ is desired [15]. Let us define, $\hat{d}^2_{min}$ as the minimum squared Euclidean distance (MSED) between two symbol-vectors with a Hamming distance of one bit in the applied mapping, i.e., $\hat{d}^2_{min}$ is the minimum value of $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$ in (1.6), and define

$$J(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \begin{cases} 1 & \text{if } \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \hat{d}^2_{min} \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

as an indicator function. At high SNR values, $\exp\left(-\frac{\hat{d}^2_{min}}{4N_0}\right)$ becomes the dominant term in (1.6). Therefore, (1.6) at high SNR values can be approximated as follows

$$\begin{aligned}\Phi_a(\mu, \boldsymbol{\chi}) &\simeq \left(\frac{1}{mN2^{mN}}\sum_{i=1}^{mN}\sum_{b=0}^{1}\sum_{\boldsymbol{x}\in\boldsymbol{\chi}_b^i} J(\boldsymbol{x}, \hat{\boldsymbol{x}})\exp\left(-\frac{\hat{d}^2_{min}}{4N_0}\right)\right)^{-1} \\ &= \frac{1}{\hat{N}_{min}}\exp\left(\frac{\hat{d}^2_{min}}{4N_0}\right), \end{aligned} \tag{1.8}$$

where $\hat{N}_{min}$ is defined as

$$\hat{N}_{min} = \frac{1}{mN2^{mN}}\sum_{i=1}^{mN}\sum_{b=0}^{1}\sum_{\boldsymbol{x}\in\boldsymbol{\chi}_b^i} J(\boldsymbol{x}, \hat{\boldsymbol{x}}). \tag{1.9}$$

In fact, $\hat{N}_{min}$ is the average number of signal pairs in which the two signals are at the Euclidean distance $\hat{d}_{min}$ and at the Hamming distance of one bit from each other.

According to (1.8), in the high SNR region, $\Phi_a(\mu, \boldsymbol{\chi})$ clearly depends on $\hat{N}_{min}$ and $\hat{d}^2_{min}$, which are determined by the applied mapping. In fact, $\Phi_a(\mu, \boldsymbol{\chi})$ increases as $\hat{N}_{min}$ decreases or as $\hat{d}^2_{min}$ increases. However, the effect of $\hat{d}^2_{min}$ on $\Phi_a(\mu, \boldsymbol{\chi})$ is much more significant than that of $\hat{N}_{min}$ due to the exponential relationship between $\hat{d}^2_{min}$ and $\Phi_a(\mu, \boldsymbol{\chi})$.

1.6.2 Rayleigh fading channel

The so called *harmonic mean* of the MSED [28] of a mapping is a well-known parameter that relates to the BER performance of BICM-ID on Rayleigh fading channels. In [15], the harmonic mean of the MSED is developed for $2N$-D mappings and is expressed as

$$\Phi(\mu, \boldsymbol{\chi}) = \left(\frac{1}{mN2^{mN}}\sum_{i=1}^{mN}\sum_{b=0}^{1}\sum_{\boldsymbol{x}\in\boldsymbol{\chi}_b^i} \frac{1}{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2}\right)^{-1}. \tag{1.10}$$

For the performance at the first iteration, $\hat{\boldsymbol{x}}$ refers to the nearest neighbor of $\boldsymbol{x}$ in $\boldsymbol{\chi}_b^i$, and $\Phi(\mu, \boldsymbol{\chi})$ is referred to as the harmonic mean of the MSED before feedback. For the asymptotic performance, $\boldsymbol{\chi}_b^i$ involves only one symbol-vector $\hat{\boldsymbol{x}}$, which is different from $\boldsymbol{x}$ only in the i^{th} bit position. In this case, (1.10) is referred to as the harmonic mean of the MSED after feedback, which is denoted by $\hat{\Phi}(\mu, \boldsymbol{\chi})$.

We can rewrite $\Phi(\mu, \boldsymbol{\chi})$ as

$$\Phi(\mu, \boldsymbol{\chi}) = \left(\frac{1}{mN2^{mN}} \sum_{i=1}^{p} \frac{n_i}{d_i^2} \right)^{-1}, \tag{1.11}$$

where n_i is defined as

$$n_i = \sum_{j=1}^{mN} \sum_{b=0}^{1} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}_b^j} I_i(\boldsymbol{x}, \hat{\boldsymbol{x}}); \qquad i = 1, \cdots, p, \tag{1.12}$$

where $I_i(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is an indicator function that takes the value of one if the Euclidean distance between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is equal to d_i, otherwise it takes the value of zero. In what follows, we describe the mapping design criteria for the Rayleigh fading channel in low and high SNR regions.

Low SNR region

To achieve a good performance at the first iteration, a larger value of $\Phi(\mu, \boldsymbol{\chi})$ is desired. In the low SNR region, $\hat{\boldsymbol{x}}$ in (1.10) is the nearest neighbour of $\boldsymbol{x}$ in $\boldsymbol{\chi}_b^i$. Since each signal is different with its nearest neighbor at least in one bit position, $\hat{\boldsymbol{x}}$ is at the Euclidean distance d_1 from $\boldsymbol{x}$ for some values of i. As a result, n_1 in (1.11) always has a non-zero value. Moreover, summation of n_i over all values of i is constant, i.e.,

$$\sum_{i=1}^{p} n_i = mN2^{mN}. \tag{1.13}$$

Considering (1.11) and (1.13), it is obvious that for a specific value of i, any reduction in n_i without increasing n_j, where $j < i$, yields a larger value of $\Phi(\mu, \boldsymbol{\chi})$. In particular, one can increase $\Phi(\mu, \boldsymbol{\chi})$ by decreasing n_1.

High SNR region

For the asymptotic performance of BICM-ID, $\hat{\boldsymbol{x}}$ is considered to be different from $\boldsymbol{x}$ only in the i^{th} bit position. As a result, it is possible to design a mapping in which $\hat{d}_{min} > d_i$ for some small values of i. This gives $n_i = 0$ for some small values of i and it yields a larger value of $\hat{\Phi}(\mu, \boldsymbol{\chi})$. Numerical examples show that a significant increase in $\hat{d}_{min}$ leads to a considerable increase in $\hat{\Phi}(\mu, \boldsymbol{\chi})$.

In summary, it can be concluded that a mapping that offers a small value of N_{min} and a large value of $\hat{d}_{min}^2$ is suitable to improve the error performance of BICM-ID over both AWGN and Rayleigh fading channels in low and high SNR regions. A small value of N_{min} implies that the average Hamming distance between neighbouring symbols, i.e., symbols with the Euclidean distance d_1, is small, and therefore, n_1 is small. This eventually increases $\Phi(\mu, \boldsymbol{\chi})$ at the first iteration of BICM-ID in the Rayleigh fading channel. Thus, a small value of N_{min} can improve the mapping's performance at the first iteration, i.e., in low SNR region, not only in the AWGN channel but also in the Rayleigh fading channel. On the other hand, a large value of $\hat{d}^2_{min}$ implies that n_i is equal to zero for small values of i. This increases $\hat{\Phi}(\mu, \boldsymbol{\chi})$ at high SNR values. As a result, the asymptotic BER performance of BICM-ID improves over AWGN and Rayleigh fading channels as the value of $\hat{d}^2_{min}$ increases.

1.7 Coded modulation capacity

Throughout this thesis, we use the coded modulation (CM) capacity (or constrained capacity) [28] to indicate how far the achieved performance is from the capacity. Table 1.2 lists the CM capacity for different modulations over the AWGN and Rayleigh fading channels.

Table 1.2: CM capacity for M-QAM.

Modulation	AWGN (E_b/N_0 in dB)	Rayleigh fading (E_b/N_0 in dB)
16-QAM	2.09	3.98
32-QAM	3.13	5.02
64-QAM	4.28	6.24
128-QAM	5.26	7.34
256-QAM	6.48	8.65
512-QAM	7.57	9.77
1024-QAM	8.91	11.11

1.8 Literature review of mapping methods

This section provides a brief review of the well-known methods that have been applied to find suitable mappings for BICM-ID. To the best of our knowledge, none of the heuristic mapping methods in the literature is applicable for higher order modulations. Therefore, in this section, we review only the computer-based mapping search techniques, which can be used for higher order modulations.

1.8.1 Binary switching algorithm (BSA)

One iteration of the BSA can be described in the following steps [9]:
1- A random mapping is generated for the constellation.
2- A cost function is calculated for each symbol in the constellation, and then, symbols are listed in descending cost value.
3- The label of the symbol with the highest cost value is switched with the label of another symbol such that the total cost is reduced as much as possible.
4- If a switching is done in step 3, go to step 2; otherwise, look for such a switching for the next symbol in the list (the symbol with the next highest cost value).
5- If a switching is done in step 4, go to step 2; otherwise, the algorithm ends.

Usually, the BSA is applied for a number of iterations and the best achieved mapping is selected as the BSA mapping. The BSA can be considered as the best known computer search technique to find suitable mappings for BICM-ID. However, it becomes intractable for finding suitable mappings for modulations with a large alphabet size due to the computational time constraints.

1.8.2 Random mapping

The random mapping technique searches among a set of randomly generated mappings to find a mapping that minimizes a defined cost function [16], [25]. In order to achieve a better result, the set of randomly generated mappings should be as large as possible. This makes the random mapping technique suffer from computational complexity and degrades the obtained mapping's performance.

The rest of this thesis focuses on our contributions to the signal mapping problem for BICM-ID. In particular, Chapter 2 provides our proposed mapping method for higher order 2-D modulations. Chapter 3 presents our proposed MD mapping of 16- and 64-QAM. Chapter 4 describes our proposed MD mapping of higher order modulations for BICM-ID over Rayleigh fading channels. Chapter 5 presents our proposed MD mapping method using rectangular QAMs for BICM-ID over AWGN channels. Finally, Chapter 6 concludes the thesis.

Chapter 2

Mapping for Higher Order 2-D Modulations

As mentioned earlier, finding a good mapping for higher order modulations for BICM-ID is challenging. Several mapping algorithms for 2-D modulations were proposed in the literature. The BSA was proposed in [9]. In [13], the authors used a reactive Tabu search (RTS) method to find mappings with a minimum error-floor. A genetic algorithm (GA)-based mapping optimization was proposed in [17]. All the mentioned algorithms still exhibited high computational complexity when finding suitable mappings for higher order modulations.

In this chapter, we take a heuristic approach to propose a systematic mapping method for higher order modulations, where the computer search based methods become impractical. The proposed method is a simple and explicit method and easily generates good mappings for higher order modulations. We study the resulting mappings' characteristics and compare their performance with other well-known mappings to date. Numerical results show that for a target BER of 10^{-6} and over the Rayleigh fading channel, our resulting mapping offers a gain of 0.7 dB over the RTS mapping for 64-QAM. This gain is 2.3 dB and 4.4 dB, respectively, over the well-known BSA mappings for 256- and 1024-QAM. For all these cases, our mappings exhibit a comparable error-floor with a gap of about 0.6 dB or less. On the AWGN channel, our achieved gains are even larger.

2.1 Proposed mapping

Our proposed mapping consists of a precoding process followed by an intermediate mapping as described below.

2.1.1 Precoding process

Let us denote the proposed precoding function by $\Psi : \{0,1\}^m \rightarrow \{0,1\}^m$, where m is the number of bits per symbol. According to the proposed precoding method, an arbitrary m-bit label $\boldsymbol{l}_t = [l_t^1, l_t^2, \cdots, l_t^m]$ is converted to a precoded m-bit label $\hat{\boldsymbol{l}}_t = \Psi(\boldsymbol{l}_t) =$

$[\hat{l}_t^1, \hat{l}_t^2, \cdots, \hat{l}_t^m]$ as follows:

$$\hat{l}_t^i = \begin{cases} W(\boldsymbol{l}_t) & \text{if } i = \text{chosen-index} \\ l_t^i \oplus W(\boldsymbol{l}_t) & \text{otherwise,} \end{cases} \tag{2.1}$$

where $W(\boldsymbol{x})$ is an indicator function that takes the value of one if the Hamming weight of $\boldsymbol{x}$ is odd, otherwise it is equal to zero, the chosen-index can take value from the set $\{1, 2, \cdots, m\}$[1], and $\oplus$ is the modulo-2 addition.

Example 2.1. *Suppose that the chosen-index is equal to one and label* $\boldsymbol{l}_1 = [l_1^1, l_1^2, l_1^3, l_1^4] = [1, 1, 0, 1]$. *Since the Hamming weight of* $\boldsymbol{l}_1$ *is odd, using eq. (2.1), we can write*

$$\begin{aligned} \hat{l}_1^1 &= W(\boldsymbol{l}_1) = W([1,1,0,1]) = 1, \\ \hat{l}_1^2 &= W(\boldsymbol{l}_1) \oplus l_1^2 = W([1,1,0,1]) \oplus 1 = 1 \oplus 1 = 0, \\ \hat{l}_1^3 &= W(\boldsymbol{l}_1) \oplus l_1^3 = W([1,1,0,1]) \oplus 0 = 1 \oplus 0 = 1, \\ \hat{l}_1^4 &= W(\boldsymbol{l}_1) \oplus l_1^4 = W([1,1,0,1]) \oplus 1 = 1 \oplus 1 = 0. \end{aligned}$$

Thus, using our proposed precoding process, $\boldsymbol{l}_1 = [1, 1, 0, 1]$ *results in* $\hat{\boldsymbol{l}}_1 = [1, 0, 1, 0]$. *Similarly, for* $\boldsymbol{l}_2 = [0, 1, 0, 1]$ *the Hamming weight is even; as such, eq. (2.1) yields* $\hat{\boldsymbol{l}}_2 = [0, 1, 0, 1]$.

Proposition 2.1. *Suppose that* $\boldsymbol{l}_k$ *and* $\boldsymbol{l}_n$ *are two* m*-bit labels that are different only in the* j^{th} *bit position. According to the proposed precoding in eq. (2.1), the precoded label of* $\boldsymbol{l}_k$*, i.e.,* $\hat{\boldsymbol{l}}_k$*, and the precoded label of* $\boldsymbol{l}_n$*, i.e.,* $\hat{\boldsymbol{l}}_n$*, have the Hamming distance of* m *bits if the chosen-index is equal to* j*. Otherwise, they have the Hamming distance of* $(m-1)$ *bits.*

Proof. Suppose that $\boldsymbol{l}_k = [l_k^1, l_k^2, \cdots, l_k^j, \cdots, l_k^m]$ and $\boldsymbol{l}_n = [l_n^1, l_n^2, \cdots, l_n^j, \cdots, l_n^m]$, where for all i except $i \neq j, l_k^i = l_n^i$. Using eq. (2.1), we can obtain the corresponding precoded labels for $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$, respectively, as $\hat{\boldsymbol{l}}_k = [\hat{l}_k^1, \hat{l}_k^2, \cdots, \hat{l}_k^j, \cdots, \hat{l}_k^m]$ and $\hat{\boldsymbol{l}}_n = [\hat{l}_n^1, \hat{l}_n^2, \cdots, \hat{l}_n^j, \cdots, \hat{l}_n^m]$. Without loss of generality, let us assume that the chosen-index is equal to q. Then, there are two possible cases as follows.

Case 1: The chosen-index is equal to the bit position in which $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ differ, i.e., $q = j$. Using eq. (2.1), we can write $\hat{l}_k^i = W(\boldsymbol{l}_k)$ and $\hat{l}_n^i = W(\boldsymbol{l}_n)$ if $i = q$. Moreover, the Hamming distance between $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ is equal to one bit. Therefore, we can write

$$\hat{l}_k^i \oplus \hat{l}_n^i = W(\boldsymbol{l}_k) \oplus W(\boldsymbol{l}_n) = 1. \tag{2.2}$$

Now, $\hat{l}_k^i \oplus \hat{l}_n^i = 1$ implies that $\hat{l}_k^i = \bar{\hat{l}}_n^i$, where $\bar{x}$ is the 1's complement of x.

[1]In the next subsection, we describe how to choose the chosen-index.

Similarly, using eq. (2.1), we can write $\hat{l}_k^i = l_k^i \oplus W(\boldsymbol{l}_k)$ and $\hat{l}_n^i = l_n^i \oplus W(\boldsymbol{l}_n)$ when $i \neq q$, which yields

$$\hat{l}_k^i \oplus \hat{l}_n^i = l_k^i \oplus W(\boldsymbol{l}_k) \oplus l_n^i \oplus W(\boldsymbol{l}_n). \tag{2.3}$$

Using the fact that the Hamming distance of $\boldsymbol{l}_k$ and $\boldsymbol{l}_l$ is equal to one bit and $l_k^i \oplus l_n^i = 0$, from eq. (3) we can write $\hat{l}_k^i \oplus \hat{l}_n^i = 1$, which implies that $\hat{l}_k^i = \bar{\hat{l}}_n^i$. Hence, for Case 1, $\hat{l}_k^i = \bar{\hat{l}}_n^i$ for all i, which means that $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have a Hamming distance of m bits from each other.

Case 2: The chosen-index is equal to one of the bit positions in which $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ do not differ, i.e., $q \neq j$. Similar to Case 1, using eq. (2.1), we can write

$$\hat{l}_k^i \oplus \hat{l}_n^i = W(\boldsymbol{l}_k) \oplus W(\boldsymbol{l}_n) = 1, \quad \text{if} \quad i = q, \tag{2.4}$$

which means that $\hat{l}_k^i = \bar{\hat{l}}_n^i$. However, if $i \neq q$, using eq. (2.1) we can write $\hat{l}_k^i = l_k^i \oplus W(\boldsymbol{l}_k)$ and $\hat{l}_n^i = l_n^i \oplus W(\boldsymbol{l}_n)$. Then, we have

$$\hat{l}_k^i \oplus \hat{l}_n^i = [l_k^i \oplus l_n^i] \oplus [W(\boldsymbol{l}_k) \oplus W(\boldsymbol{l}_n)]. \tag{2.5}$$

Since the Hamming distance of $\boldsymbol{l}_k$ from $\boldsymbol{l}_n$ is equal to one bit, and $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ are different in the j^{th} bit-position, then using eq. (2.5) we can write $\hat{l}_k^j \oplus \hat{l}_n^j = 0$, which implies that $\hat{l}_k^j = \hat{l}_n^j$. On the other hand, if $i \neq j$, we can write $l_k^i \oplus l_n^i = 0$, which yields eq. (2.5) to be equal to one. This implies that $\hat{l}_k^i = \bar{\hat{l}}_n^i$. Consequently, for Case 2, for all i except $i \neq j$, $\hat{l}_k^i = \bar{\hat{l}}_n^i$, which means that $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have a Hamming distance of $(m-1)$ bits from each other. $\square$

Proposition 2.2. *The proposed precoding function is bijective, i.e.,* $\boldsymbol{l}_t = \Psi^{-1}(\hat{\boldsymbol{l}}_t)$*; the original label* $\boldsymbol{l}_t$ *can be uniquely obtained from the precoded label* $\hat{\boldsymbol{l}}_t$*.*

Proof. Let us consider that the chosen-index is q, and then according to eq. (2.1), we can write

$$\hat{l}_t^q = W(\boldsymbol{l}_t) = \sum_{\forall j} l_t^j = l_t^q \oplus \sum_{j \neq q} l_t^j, \tag{2.6}$$

where $\sum$ is a modulo-2 summation. From eq. (2.6), we have

$$l_t^q = \hat{l}_t^q \oplus \sum_{j \neq q} l_t^j. \tag{2.7}$$

Now, let us find the m-bit label $\boldsymbol{l}_t = [l_t^1, l_t^2, \cdots, l_t^i, \cdots, l_t^m]$ from its precoded version, i.e., $\hat{\boldsymbol{l}}_t = \Psi(\boldsymbol{l}_t) = [\hat{l}_t^1, \hat{l}_t^2, \cdots, \hat{l}_t^m]$. There are two possible cases as follows.

Case 1: $i \neq q$. In this case, according to eq. (2.1), $\hat{l}_t^i$ is given by

$$\hat{l}_t^i = l_t^i \oplus W(\boldsymbol{l}_t), \tag{2.8}$$

and using $\hat{l}_t^q = W(\boldsymbol{l}_t)$, we can write

$$l_t^i = \hat{l}_t^i \oplus W(\boldsymbol{l}_t) = \hat{l}_t^i \oplus \hat{l}_t^q. \tag{2.9}$$

Case 2: $i = q$. In this case, using eq. (2.7), l_t^i can be obtained as

$$l_t^i = l_t^q = \hat{l}_t^q \oplus \sum_{j \neq q} l_t^j. \tag{2.10}$$

Using eq. (2.9), we simplify eq. (2.10) as

$$\begin{aligned} l_t^i &= \hat{l}_t^q \oplus \sum_{j \neq q} (\hat{l}_t^j \oplus \hat{l}_t^q) \\ &= \hat{l}_t^q \oplus \sum_{j \neq q} \hat{l}_t^j \oplus \sum_{j \neq q} \hat{l}_t^q \\ &= \sum_{\forall j} \hat{l}_t^j \oplus \sum_{j \neq q} \hat{l}_t^q \\ &= W(\hat{\boldsymbol{l}}_t) \oplus A(m) \times \hat{l}_t^q, \end{aligned} \tag{2.11}$$

where $A(m)$ is an indicator function that takes the value zero if m is odd, otherwise it is equal to one. Consequently, if q denotes the chosen-index, using eqs. (2.9) and (2.11), the reverse equation to generate $\boldsymbol{l}_t$ from $\hat{\boldsymbol{l}}_t$ can be expressed as

$$l_t^i = \begin{cases} \hat{l}_t^i \oplus \hat{l}_t^q & \text{if } i \neq q \\ W(\hat{\boldsymbol{l}}_t) \oplus A(m) \times \hat{l}_t^q & \text{if } i = q. \end{cases} \tag{2.12}$$

From eq. (2.12), it is obvious that $\boldsymbol{l}_t$ depends only on its precoded version $\hat{\boldsymbol{l}}_t$. In other words, two precoded labels, $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_l$ yield the same original label only when $\hat{\boldsymbol{l}}_k = \hat{\boldsymbol{l}}_l$. That is, $\hat{\boldsymbol{l}}_k = \hat{\boldsymbol{l}}_l$ leads to $\boldsymbol{l}_k = \boldsymbol{l}_l$. As a result, the precoding function is bijective. $\square$

2.1.2 Mapping process

Our proposed mapping uses Gray mapping as the intermediate mapping. The reason will be explained later in this section. In order to obtain the resulting mappings using the precoded label $\hat{\boldsymbol{l}}_t$ and Gray mapping, we map the symbols as follows. In the Gray labeled constellation, the symbol labeled with $\hat{\boldsymbol{l}}_t$ is mapped to $\boldsymbol{l}_t$. The proposed mapping is defined

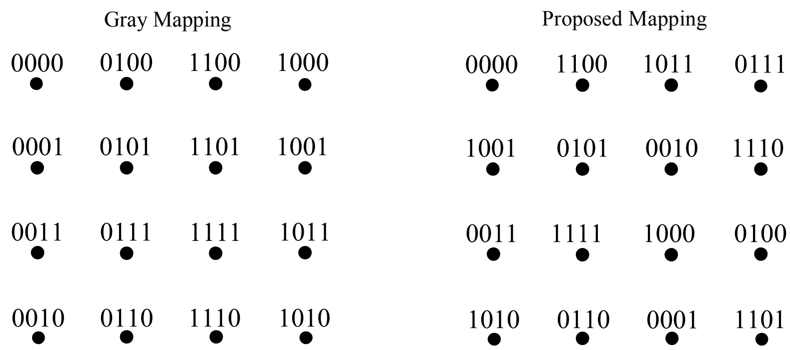

Figure 2.1: An example of the resulting mapping for 16-QAM.

as

$$\Phi(\boldsymbol{l}_t) = G(\hat{\boldsymbol{l}}_t) = G(\Psi(\boldsymbol{l}_t)), \tag{2.13}$$

where $\Phi(\cdot)$ is the proposed mapping and $G(\cdot)$ represents the Gray mapping.

Example 2.2. *For example, Fig. 2.1 shows the resulting* 16*-QAM mapping with our proposed algorithm. In what follows, we describe how the resulting* 16*-QAM mapping is obtained. Let us assume that the chosen-index is equal to one. Using eq. (2.1), the precoded version of labels* $\boldsymbol{l}_1 = [1, 1, 0, 1]$ *and* $\boldsymbol{l}_2 = [0, 1, 0, 1]$ *can be written as follows (c.f., Example 2.1):*

$$\begin{aligned} \boldsymbol{l}_1 = [1, 1, 0, 1] \quad &\rightarrow \quad \hat{\boldsymbol{l}}_1 = [1, 0, 1, 0], \\ \boldsymbol{l}_2 = [0, 1, 0, 1] \quad &\rightarrow \quad \hat{\boldsymbol{l}}_2 = [0, 1, 0, 1]. \end{aligned}$$

Now, $\boldsymbol{l}_1 = [1, 1, 0, 1]$ *is mapped to the symbol in the* 16*-QAM constellation whose label in the Gray labeled 16-QAM is* $\hat{\boldsymbol{l}}_1 = [1, 0, 1, 0]$*, and* $\boldsymbol{l}_2 = [0, 1, 0, 1]$ *is mapped to the symbol whose label in the Gray labeled 16-QAM constellation is* $\hat{\boldsymbol{l}}_2 = [0, 1, 0, 1]$*. In a similar fashion, the rest of the* 16*-QAM symbols can be mapped using our proposed mapping method.*

Proposition 2.3. *The Hamming distance between two adjacent symbols in resulting mappings is either two,* m*, or* $(m-1)$ *bits, and the fraction of adjacent symbols with the Hamming distance of two bits can be at least* $(\frac{m-1}{m})$*, which tends to be larger for higher order constellations.*

Proof. See Appendix A. $\square$

2.2 Characteristics of Gray mapping

In what follows, we go through three characteristics of Gray mappings that are beneficial in our proposed mapping method.

- Gray mapping is easy to generate. For square QAM and PSK constellations, Gray labeling can be obtained from the natural binary labeling (see [29] and [30] for details). For cross QAM constellations, pseudo Gray labeling can be obtained using the procedure described in [31].
- The Hamming distance between two adjacent symbols is one bit for Gray mappings and is at most two bits for pseudo-Gray mappings.
- Our study shows that among the well-known mappings of 2^m-ary modulations, Gray mappings have the largest average Euclidean distance between any pair of symbols with the Hamming distance of m or $(m-1)$ bits. For example, Table 2.1 shows that among the well-known mappings for 16-QAM ($m = 4$), the Gray mapping has the largest average Euclidean distance between symbols with the Hamming distance of 3 or 4 bits (please refer to [32] for the comparison of the average Euclidean distance between symbols with the Hamming distance of m or $(m-1)$ bits for other QAM constellations).

Table 2.1: Average Euclidean distance between symbols with 3 or 4-bit Hamming distances for 16-QAM mappings.

Mapping	Average Euclidean distance
Set Partitioning [33]	0.8154
Modified Set Partitioning [34]	0.7946
Mixed Laleling [34]	0.7520
Gray [34]	0.8541
MSEW-1 [11]	0.7798
MSEW-2 [11]	0.7195
MSEW-3 [11]	0.6584
$M16^a$ [9]	0.8255
$M16^r$ [9]	0.7807

2.3 Characteristics of our mapping

Since our proposed mapping uses a simple precoding and Gray mapping as an intermediate mapping, it provides a simple and efficient method for generating mappings for QAM and PSK modulations of any order. Moreover, it has two interesting characteristics as follows.

- The average Euclidean distance between the symbols with the Hamming distance of one bit is considerably increased. This can increase $\hat{d}^2_{min}$ and the harmonic mean of the MSED after feedback (denoted by $\hat{\Phi}(\mu, \boldsymbol{\chi})$), and as a result, it improves the BER performance of BICM-ID in the error-floor region.

- According to proposition 2.3, the Hamming distance between most of the adjacent symbols in our proposed mappings is two bits [2]. This decreases N_{min} and increases the harmonic mean of MSED before feedback (denoted by $\Phi(\mu, \boldsymbol{\chi})$), and therefore, it improves the BER performance of BICM-ID in the turbo cliff region.

2.4 Numerical results and discussion

In this section, we compare the performance of our proposed mappings and the best previously known mappings for BICM-ID over AWGN and Rayleigh fading channels. Table 2.2 compares various evaluation parameters for different QAM mappings. This table shows that the proposed mappings improve N_{min} and Φ, while offering significantly large values of $\hat{d}^2_{min}$ and $\hat{\Phi}$. As a consequence, it is expected that the proposed mappings offer a better error performance for BICM-ID in the low SNR region.

Table 2.2: Evaluation parameters for different QAM mappings.

Modulation	Mapping	N_{min}	$\hat{d}^2_{min}$	Φ	$\hat{\Phi}$
64-QAM	RTS-based mapping [13]	2.7500	1.6190	0.1048	2.8742
	TV mapping [14]	5.0000	1.2381	0.0952	2.2784
	Proposed mapping	2.2143	1.2381	0.1127	2.4986
256-QAM	BSA mapping [9]	3.6104	1.1765	0.0257	2.8741
	TV mapping [14]	7.0000	0.5882	0.02353	1.7876e
	Proposed mapping	2.1667	1.0588	0.0333	2.5301
1024-QAM	BSA mapping [9]	4.4899	0.5865	0.0064	2.7515
	Proposed mapping	2.1129	0.9208	0.0100	2.5617

Simulation results for the BER performance of the BICM-ID system using different mappings are shown in Fig. 2.2-2.6. We use a rate-$\frac{1}{2}$ convolutional code with the generator polynomial of $(13, 15)_8$. The length of the used interleaver is about 10000 bits. All gains are quantified at a BER of 10^-6, which is the target BER of many practical data communications. All BER curves are presented with seven iterations, and BER performance for various iterations can be found in [32].

[2]We are thankful to the anonymous Reviewer of the original article as his/her comment motivated us to investigate why our mappings perform better in the turbo cliff region and we found this characteristic.

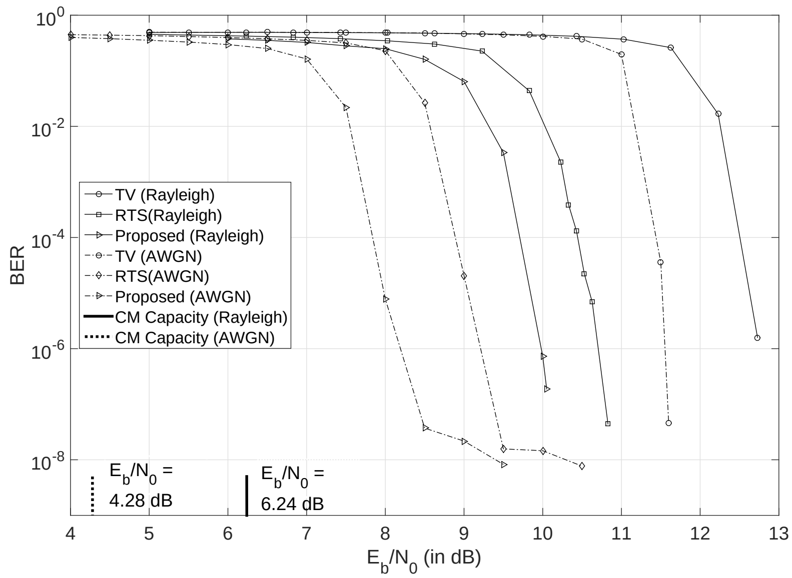

Figure 2.2: BER performance of 64-QAM.

Fig. 2.2 plots the simulation results for the BER performance of BICM-ID using our resulting 64-QAM mapping and the previously known 64-QAM mappings, i.e., the RTS-based mapping, which is optimal in the error-floor region [13], and TV mapping[3]. In this figure, we also plot the corresponding analytical bound on the error-floor using the Gauss-Chebyshev method [34]. From this figure, we can observe that our proposed mapping outperforms the RTS-based mapping in the turbo cliff region with a certain degradation of the error-floor. This is expected as our proposed mapping has a larger value of Φ while it has a comparable value of $\hat{\Phi}$, as listed in Table 2.2. As shown in Fig. 2.2, in comparison with the TV mapping, our proposed mapping significantly improves the error performance of BICM-ID in both the turbo cliff and error-floor regions. This is because the values of Φ and $\hat{\Phi}$ for our proposed mapping are large (see Table 2.2). On Rayleigh fading channels, the proposed mapping outperforms the RTS-based mapping by 0.7 dB and TV mapping by 2.7 dB. The gain on AWGN channels is even larger.

In Fig. 2.3, we compare the BER performance of BICM-ID using our proposed mapping, TV mapping, and the BSA mapping for 256-QAM. We have used the BSA mapping that is optimized for the error-floor region. By optimizing the generalized weighted BSA cost function [9] via exhaustive search, the performance of BSA mapping in the turbo cliff region can be improved at the expense of the error-floor. Moreover, the time complexity of

[3] For the constellations presented in this section, we have used the same TV mappings that are reported in [14].

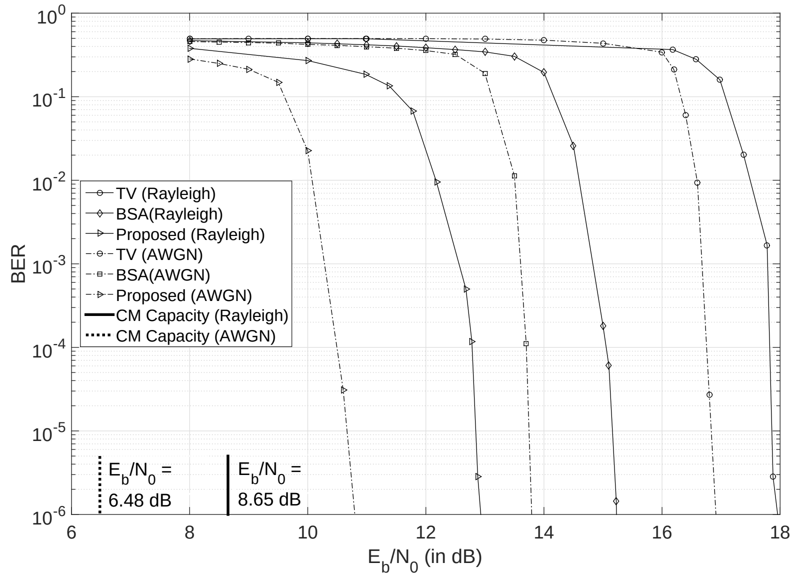

Figure 2.3: BER performance of 256-QAM.

this exhaustive search can prohibit finding a BSA mapping for higher order constellations. Fig. 2.3 shows that our mapping outperforms the BSA and TV mappings in the turbo cliff region and it achieves a gain of about 2.3 dB over the BSA mapping. In Fig. 2.4, we compare the BER performance of our proposed mapping with that of the BSA mapping for 1024-QAM. On Rayleigh fading channels, our mapping for 1024-QAM offers a gain of about 4.4 dB. This is expected as our proposed mapping has a larger value for Φ than that of the BSA mapping (see Table 2.2). The obtained gains on AWGN channels are even larger. Finally, in Fig. 2.5 and Fig. 2.6, we compare the error-floor of BICM-ID using 64-QAM and using 256- and 1024-QAM mappings, receptively on Rayleigh fading channels. These figures show that our mappings offer comparable error-floors (the gap is 0.6 dB or less) to those of the best known mappings.

It is worth noting that it takes more than a day for the BSA to complete only one round of the search algorithm for 1024-QAM. However, our method requires only a fraction of a second to obtain the corresponding proposed mapping. Moreover, our mappings improve the system performance compared to the BSA mappings. This indicates our method's efficiency compared to the BSA.

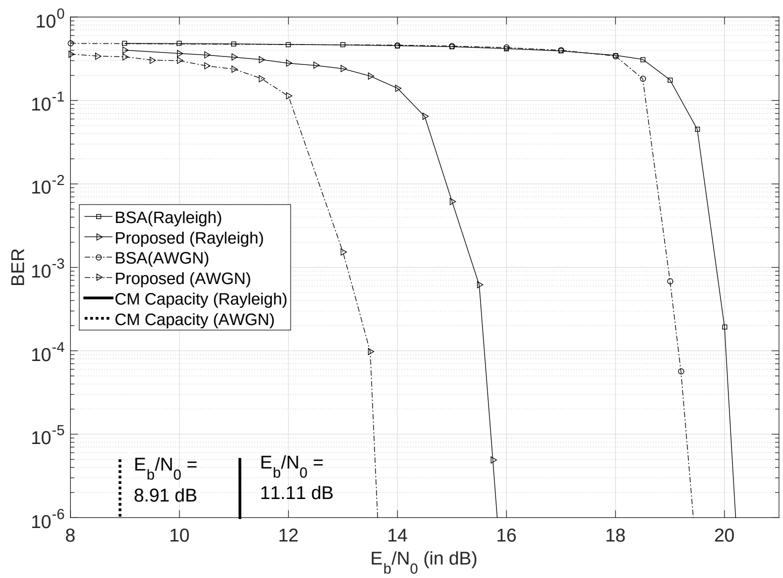

Figure 2.4: BER performance of 1024-QAM.

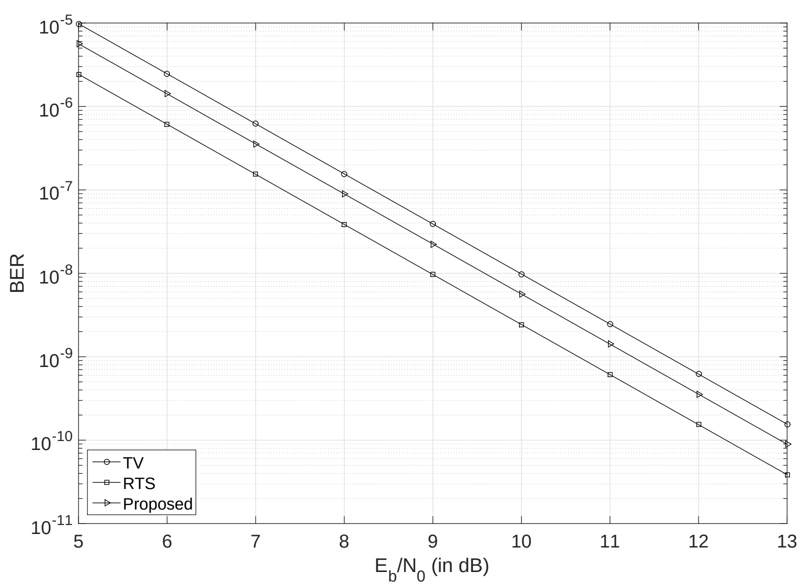

Figure 2.5: Analytical bound on the error-floor [34] of 64-QAM on Rayleigh fading channels.

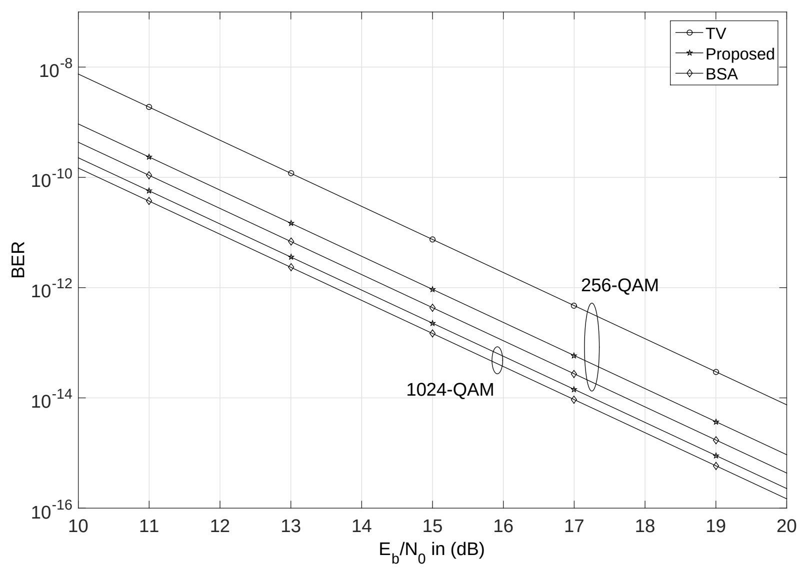

Figure 2.6: Analytical bound on the error-floor [34] of 256- and 1024-QAM on Rayleigh fading channels.

Chapter 3

Efficient Multi-Dimensional Mapping of 16-, 64-QAM for BICM-ID

Assigning a sequence of binary digits to a vector of symbols rather than a single symbol is referred to as a MD mapping. MD mapping improves system bandwidth efficiency and also offers more flexibility in generating good mappings for BICM-ID [15]. However, it tremendously increases the number of possible mappings and makes it difficult to find good/optimum mappings for MD modulations. This problem is more severe when MD modulations are constructed using higher order 2-D modulations. Suitable MD mappings are obtained by computer search techniques except for MD modulations that use a smaller modulation, e.g., BPSK, QPSK and 8-PSK, as a basic modulation.

The BSA can be considered as the best known computer search method for finding good mappings. However, to obtain suitable mappings for larger modulations such as MD modulations, the BSA becomes intractable due to its complexity [16]. In [16] and [25], the authors demonstrated that random mapping can lead to efficient MD mappings. According to the random mapping technique, computer search is used to obtain a good mapping from a large set of randomly generated mappings, which makes the procedure complex. Moreover, it degrades the resulting mappings' performance especially for larger MD modulations.

In this chapter, we propose a more efficient mapping method for MD modulations that use 2^m-QAM ($m = 4, 6$) as the basic modulation. Our goal is to obtain mappings that improve the error performance of BICM-ID at low SNR values as well as at high SNR values. A similar objective is considered in [11] where the authors used a doping technique (combining two mappings) to obtain mappings for 2-D modulations. However, instead of combining mappings, we develop a single mapping for a given MD modulation, in order to achieve a lower error rate for BICM-ID in both low and high SNR regions. Furthermore, our proposed method yields mappings for MD modulations rather than 2-D ones. The proposed method is a heuristic-based technique and does not employ any computer searching. The presented numerical results show that our approach not only efficiently generates mappings but also improves the BER performance of BICM-ID over both AWGN and Rayleigh fading channels. For example, our method can save about 3 dB of transmit signal energy for a target BER of 10^{-6} compared to the mappings found by the BSA and random mappings.

3.1 Proposed mapping method

As it is discussed in section 1.6, a mapping that offers a small value of N_{min} while it gives a large value of $\hat{d}_{min}$ is suitable to achieve good error performance of BICM-ID over both AWGN and Rayleigh fading channels in low and high SNR regions. Based on this discussion, we take a heuristic approach to construct a mapping that improves the BER performance of BICM-ID in low and high SNR regions over AWGN and Rayleigh fading channels. In particular, we apply two key techniques as follows. First, to generate a mapping with a large value of $\hat{d}_{min}$, we map binary labels with a Hamming distance of one bit to the symbol-vectors with a large Euclidean distance. This leads to improved error-floors over AWGN and Rayleigh fading channels. Second, most of the nearest neighbouring symbol-vectors are mapped to the binary labels with a Hamming distance of two bits. This results in a small value of N_{min} and yields good BER performance in a low SNR region over AWGN and Rayleigh fading channels.

3.1.1 Method description

The proposed MD mapping using 2^m-QAM symbols is constructed progressively in $(m-1)$ steps. The mappings in step i $(1 \leq i \leq (m-2))$ are intermediate mappings whereas the mapping in step $i = (m-1)$ is the final mapping. In the i^{th} step, 2^{i+1} symbols from 2^m-ary constellation are selected to be used in the mapping process. In what follows, we describe our mapping method in details.

Symbols selection

Let S_j represent the symbol with position-index j in a square QAM constellation, $j = 1, \cdots, 2^m$. We assume that j increases from left to right and from top to bottom in the constellation. The general principles in choosing 2^{i+1} symbols from a 2^m-QAM constellation in the i^{th} step are as follows: (i) by moving the set of selected symbols one can cover all symbols of the constellation such that each symbol is covered only one time. In other words, the square M-QAM constellations can be partitioned into a number of subsets where the structures/shapes formed by these subsets are congruent with one another. Thus, by moving one of the subsets and superimposing it on the remaining subsets, one can cover all symbols in the constellation such that each symbol is covered only once. (ii) The MSED between the chosen symbols is as large as possible. Without loss of generality, assume that χ_i denotes the set of 2^{i+1} chosen symbols in step i and $\boldsymbol{\alpha}_i = [\alpha_i^{(1)}, \alpha_i^{(2)}, \cdots, \alpha_i^{(2^{i+1})}]$ indicates the position-indexes of symbols in χ_i. The set of used symbols in step $(i+1)$ contains all the used symbols in step i, i.e., $\chi_i \subset \chi_{i+1}$ and $\boldsymbol{\alpha}_i \subset \boldsymbol{\alpha}_{i+1}$. As such in step $(m-1)$, all symbols in the constellation will be used to construct the MD mapping.

Mapping process

Suppose that $\boldsymbol{l} = [l^{(1)}, l^{(2)}, \cdots, l^{(mN)}]$ is an mN-bit binary label, and in step i, $\boldsymbol{a}_i = [a_i^{(1)}, a_i^{(2)}, \cdots, a_i^{((i+1)N)}]$ denotes the $(i+1)N$ least significant bits of $\boldsymbol{l}$ where $a_i^{(k)}$ is given by:

$$a_i^{(k)} = l^{(mN-(i+1)N+k)}, \qquad k = 1, 2, \cdots, (i+1)N. \tag{3.1}$$

Assume that $\boldsymbol{a}_i$ is mapped to symbol-vector $\boldsymbol{x}_i = [x_i^{(1)}, \cdots, x_i^{(N)}]$, where $x_i^{(k)} \in \chi_i$. The corresponding position-index vector for $\boldsymbol{x}_i$ is denoted by $\boldsymbol{j}_i = [j_i^{(1)}, \cdots, j_i^{(N)}]$, where $j_i^{(k)} \in \boldsymbol{\alpha}_i$ refers to the position-index of symbol $x_i^{(k)}$ in the constellation. Now, we describe the steps of the mapping process.

First step: In step $i = 1$, the selected symbol set χ_1 is equivalent to QPSK symbols in terms of intersymbol Euclidean distances. Therefore, in order to achieve a good mapping, we use the optimum MD QPSK mapping method introduced in [12]. In particular, a $2N$-bit label $\boldsymbol{a}_1$ is mapped to N consecutive QPSK symbols using the method proposed in [12]. Then, we use a conversion vector, denoted by $\boldsymbol{\gamma} = [\gamma^{(1)}, \cdots, \gamma^{(4)}]$, to convert each symbol in the achieved MD QPSK mapping to one of the symbols in χ_1. Without loss of generality, we assume the QPSK symbols are expressed as:

$$P_k = e^{j\frac{\pi k}{2}}; \quad k = 1, \cdots, 4; \quad j^2 = -1, \tag{3.2}$$

where k is the symbol position-index in the QPSK constellation. A particular QPSK symbol, P_k, is converted to one of the symbols in χ_1, as given below:

$$P_k \to S_z; \quad z = \gamma^{(k)}, \tag{3.3}$$

where S_z is the symbol with position-index z in 2^m-QAM constellation. It is important to note that $\boldsymbol{\gamma}$ converts each QPSK symbol to the corresponding symbol in the 4-ary constellation created using the four chosen 2^m-QAM symbols. As a consequence, all properties of the developed MD QPSK mapping in [12] are conserved for our MD mapping using four selected 2^m-QAM symbols.

<u>*Subsequent steps*</u>: In step i ($i = 2, 3 \cdots, m-1$), we use the intermediate mapping in the previous step to map label $\boldsymbol{a}_i$ to a vector of N symbols from χ_i. Assume that in step i, $\boldsymbol{b}_i = [b_i^{(1)}, b_i^{(2)}, \cdots, b_i^{(N)}]$ denotes the N most significant bits of $\boldsymbol{a}_i$, i.e., $b_i^{(k)} = a_i^{(k)}$ for $k = 1, \cdots, N$. Each symbol in $\boldsymbol{x}_{i-1}$ is transformed to obtain the symbol-vector in step i, $\boldsymbol{x}_i$. The transformation rule is defined by $\boldsymbol{\beta}_{i,k}$, i.e.,

$$x_{i-1}^{(k)} \xrightarrow{\boldsymbol{\beta}_{i,k}} x_i^{(k)}, \quad x_{i-1}^{(k)} \in \chi_{i-1}, \quad x_i^{(k)} \in \chi_i, \tag{3.4}$$

where $\boldsymbol{\beta}_{i,k}$ is a 2^i-tuple vector and it is determined based on the Hamming weight of $\boldsymbol{b}_i$ and the bit value of $b_i^{(k)}$.

Symbol transformation using $\boldsymbol{\beta}_{i,k}$: For given vectors $\boldsymbol{j}_{i-1}$ and $\boldsymbol{\alpha}_{i-1}$ and for a particular value of k ($k = 1, \cdots, N$), there exists a $q \in \{1, \cdots, 2^i\}$ such that $j_{i-1}^{(k)} = \alpha_{i-1}^{(q)}$. Then the position-index of the k^{th} symbol in $\boldsymbol{x}_i$, i.e., $j_i^{(k)}$, is given by:

$$j_i^{(k)} = \beta_{i,k}^{(q)}, \tag{3.5}$$

where $\beta_{i,k}^{(q)}$ is the q^{th} element of the corresponding vector $\boldsymbol{\beta}_{i,k}$. The values of $j_i^{(k)}$ determine the symbols in $\boldsymbol{x}_i$.

3.1.2 Design considerations of $\boldsymbol{\beta}_{i,k}$

There are two key ideas in designing $\boldsymbol{\beta}_{i,k}$ ($i > 1$) as follows. As discussed in Section 1.6, a large value of $\hat{d}^2_{min}$ is desired to achieve a good error performance at high SNRs over AWGN and Rayleigh fading channels. Let $\hat{d}^2_{min,i}$ be the MSED between two symbol-vectors with a Hamming distance of one bit in the i^{th} step of our proposed mapping process. As mentioned earlier, the intermediate MD mapping in the first step is equivalent to the optimum MD QPSK mapping, which is developed in [12]. Therefore, it yields the largest possible value of $\hat{d}^2_{min,1}$ for the selected four symbols from 2^m-QAM. To achieve a large value of $\hat{d}^2_{min}$, $\boldsymbol{\beta}_{i,k}$ should be designed such that $\hat{d}^2_{min,i} \geq \hat{d}^2_{min,1}$ for $i = 2, 3, \cdots, m-1$. To develop a mapping with a small value of N_{min}, $\boldsymbol{\beta}_{i,k}$ is designed such that the most of the symbol-vectors with the Euclidean distance $d_{min,i}$ in step i are mapped by binary labels with a Hamming distance of two bits, where $d_{min,i}$ is the minimum Euclidean distance between the symbols in χ_i. Based on the above discussion, we design $\boldsymbol{\beta}_{i,k}$ as discussed below.

Let $\boldsymbol{a}_i = [\boldsymbol{b}_i, \boldsymbol{a}_{i-1}]$ be a given label in step i where $\boldsymbol{b}_i$ and $\boldsymbol{a}_{i-1}$ are two binary sequences of lengths N and iN bits, respectively. Assume that $\hat{\boldsymbol{a}}_i$ is a binary sequence of $(i+1)N$ bits and it is different from $\boldsymbol{a}_i$ only in the k^{th} bit position. Then, there are two possible cases for $\hat{\boldsymbol{a}}_i$ as given below:

$$\hat{\boldsymbol{a}}_i = \begin{cases} [\hat{\boldsymbol{b}}_i, \boldsymbol{a}_{i-1}] & \text{if } k \leq N, \\ [\boldsymbol{b}_i, \hat{\boldsymbol{a}}_{i-1}] & \text{if } k > N, \end{cases} \tag{3.6}$$

where $\hat{\boldsymbol{b}}_i$ and $\hat{\boldsymbol{a}}_{i-1}$ are two binary sequences of lengths N and iN, respectively that have a Hamming distance of one bit from $\boldsymbol{b}_i$ and $\boldsymbol{a}_{i-1}$, respectively. Our first goal is to map $\boldsymbol{a}_i$ and $\hat{\boldsymbol{a}}_i$ to the symbol-vectors $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}_i$, respectively such that $\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \geq \hat{d}^2_{min,1}$.

Let $\tilde{\boldsymbol{a}}_i$ be a sequence of $(i+1)N$ bits. Also assume that $\tilde{\boldsymbol{a}}_i$ is different from $\boldsymbol{a}_i$ in the j^{th} and the k^{th} bit positions where $j < k \leq (i+1)N$. Then, $\tilde{\boldsymbol{a}}_i$ can be defined as one of the three following possible cases:

$$\tilde{\boldsymbol{a}}_i = \begin{cases} [\tilde{\boldsymbol{b}}_i, \boldsymbol{a}_{i-1}] & \text{if } j < N, k \leq N, \\ [\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{a}}_{i-1}] & \text{if } j \leq N, k > N, \\ [\boldsymbol{b}_i, \tilde{\boldsymbol{a}}_{i-1}] & \text{if } j > N, k > N, \end{cases} \tag{3.7}$$

where $\tilde{\boldsymbol{b}}_i$ is an N-bit sequence with a Hamming distance of two bits from $\boldsymbol{b}_i$ and $\tilde{\boldsymbol{a}}_{i-1}$ is an iN-bit sequence with a Hamming distance of two bits from $\boldsymbol{a}_{i-1}$. Suppose that in step i, $\tilde{\boldsymbol{x}}_i$ is an element of $\boldsymbol{\psi}_i$, i. e., $\tilde{\boldsymbol{x}}_i \in \boldsymbol{\psi}_i$, where $\boldsymbol{\psi}_i$ denotes the set of the nearest symbol-vectors to $\boldsymbol{x}_i$ in $\boldsymbol{\chi}_i = \chi_i^N$. Our second goal is to map most of the symbol-vectors in $\boldsymbol{\psi}_i$ by one of the possible cases of $\tilde{\boldsymbol{a}}_i$ in (3.7). In this way, one can have:

$$d_H(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) = 2 \tag{3.8}$$

for most cases of $\tilde{\boldsymbol{x}}_i$, where $d_H(a, b)$ denotes the Hamming distance between a and b.

The two mentioned goals are achieved via a systematic symbol transformation from step $(i-1)$ to step i using $\boldsymbol{\beta}_{i,k}$. Specifically, $\boldsymbol{\beta}_{i,k}$ depends on the Hamming weight of $\boldsymbol{b}_i$ as well as on the bit value $b_i^{(k)}$. We consider four different cases for $\boldsymbol{\beta}_{i,k}$ as follows:

$$\boldsymbol{\beta}_{i,k} = \begin{cases} \boldsymbol{\beta}_{E0} & \text{if } w_H(\boldsymbol{b}_i) \in \mathbb{E},\ b_i^{(k)} = 0, \\ \boldsymbol{\beta}_{E1} & \text{if } w_H(\boldsymbol{b}_i) \in \mathbb{E},\ b_i^{(k)} = 1, \\ \boldsymbol{\beta}_{O0} & \text{if } w_H(\boldsymbol{b}_i) \in \mathbb{O},\ b_i^{(k)} = 0, \\ \boldsymbol{\beta}_{O1} & \text{if } w_H(\boldsymbol{b}_i) \in \mathbb{O},\ b_i^{(k)} = 1, \end{cases} \tag{3.9}$$

where $\boldsymbol{\beta}_{E0}$, $\boldsymbol{\beta}_{E1}$, $\boldsymbol{\beta}_{O0}$, and $\boldsymbol{\beta}_{O1}$ are row vectors with 2^i elements and each of them is a subset of $\boldsymbol{\alpha}_i$.

3.1.3 Proposed vectors for $\boldsymbol{\alpha}_i$, $\boldsymbol{\gamma}$, and $\boldsymbol{\beta}_{i,k}$

Table 3.1 shows the position-index vector of the selected symbols in step i, i. e., $\boldsymbol{\alpha}_i$, of the proposed mapping method. The proposed conversion vector, $\boldsymbol{\gamma}$, for 16-QAM and 64-QAM is also indicated in Table 3.2. Furthermore, Tables 3.3 and 3.4 provide the proposed vectors for $\boldsymbol{\beta}_{E0}$, $\boldsymbol{\beta}_{E1}$, $\boldsymbol{\beta}_{O0}$, and $\boldsymbol{\beta}_{O1}$ for different steps in the MD mapping using 16-QAM and 64-QAM, respectively.

Table 3.1: $\boldsymbol{\alpha}_i$ in different steps of the proposed mapping method.

Basic Modulation	i	$\boldsymbol{\alpha}_i$
16-QAM	1	[1, 3, 9, 11]
	2	[1, 2, 3, 4, 9, 10, 11, 12]
64-QAM	1	[1, 5, 33, 37]
	2	[1, 3, 5, 7, 33, 35, 37, 39]
	3	[1, 3, 5, 7, 17, 19, 21, 23, 33, 35, 37, 39, 49, 51, 53, 55]
	4	[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63]

Table 3.2: Conversion vector, $\boldsymbol{\gamma}$.

Basic Modulation	$\boldsymbol{\gamma}$
16-QAM	[11, 3, 1, 9]
64-QAM	[37, 5, 1, 33]

Table 3.3: Different cases of $\boldsymbol{\beta}_{i,k}$ for 16-QAM.

$\boldsymbol{\beta}_{i,k}$	$i=2$	$i=3$
$\boldsymbol{\beta}_{E0}$	[1, 3, 9, 11]	[1, 2, 3, 4, 9, 10, 11, 12]
$\boldsymbol{\beta}_{E1}$	[2, 4, 10, 12]	[5, 6, 7, 8, 13, 14, 15, 16]
$\boldsymbol{\beta}_{O0}$	[11, 9, 3, 1]	[11, 12, 9, 10, 3, 4, 1, 2]
$\boldsymbol{\beta}_{O1}$	[12, 10, 4, 2]	[15, 16, 13, 14, 7, 8, 5, 6]

3.2 Examples

In what follows, we provide a number of examples to illustrate how to use Tables 3.1-3.4 to construct the MD mapping using 16-QAM.

Example 3.1. Fig. 3.1(a) indicates the symbols in a 16-QAM constellation. According to Table 3.1, $\boldsymbol{\alpha}_1$ equals $[1, 3, 9, 11]$, which means the set of selected symbols to be used in step 1 is $\chi_1 = \{S_1, S_3, S_9, S_{11}\}$. Similarly, using this table we have $\chi_2 = \{S_1, S_2, S_3, S_4, S_9, S_{10}, S_{11}, S_{12}\}$. The symbols in χ_1 and χ_2 are shown in black colour in Fig. 3.1(b) and Fig. 3.1(c), respectively.

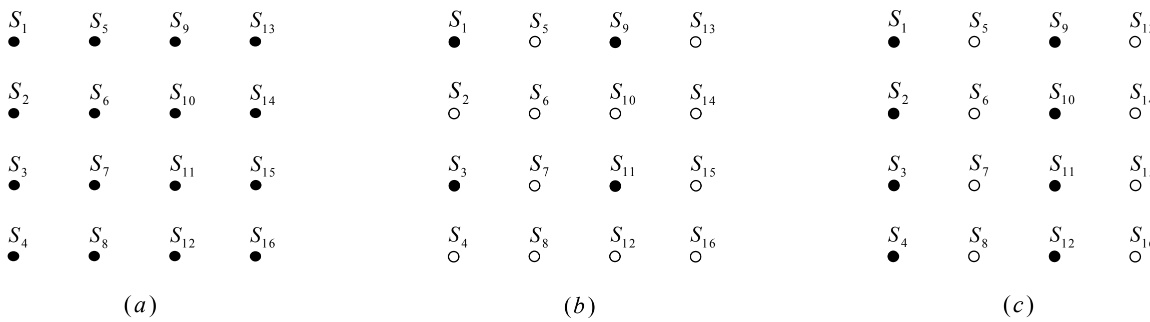

Figure 3.1: (a) A 16-QAM constellation, (b) Four selected 16-QAM symbols (dark symbols) to be used in mapping step $i = 1$, i.e., χ_1, and (c) Eight selected 16-QAM symbols (dark symbols) to be used in mapping step $i = 2$, i.e., χ_2.

Table 3.4: Different cases of $\boldsymbol{\beta_{i,k}}$ for 64-QAM.

$\boldsymbol{\beta_{i,k}}$	i	Index-vector
$\boldsymbol{\beta}_{E0}$	2	[1, 5, 33, 37]
	3	[1, 3, 5, 7, 33, 35, 37, 39]
	4	[1, 3, 5, 7, 17, 19, 21, 23, 33, 35, 37, 39, 49, 51, 53, 55]
	5	[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63]
$\boldsymbol{\beta}_{E1}$	2	[3, 7, 35, 39]
	3	[17, 19, 21, 23, 49, 51, 53, 55]
	4	[9, 11, 13, 15, 25, 27, 29, 31, 41, 43, 45, 47, 57, 59, 61, 63]
	5	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64]
$\boldsymbol{\beta}_{O0}$	2	[37, 33, 5, 1]
	3	[37, 39, 33, 35, 5, 7, 1, 3]
	4	[37, 39, 33, 35, 53, 55, 49, 51, 5, 7, 1, 3, 21, 23, 17, 19]
	5	[37, 39, 33, 35, 45, 47, 41, 43, 53, 55, 49, 51, 61, 63, 57, 59, 5, 7, 1, 3, 13, 15, 9, 11, 21, 23, 17, 19, 29, 31, 25, 27]
$\boldsymbol{\beta}_{O1}$	2	[39, 35, 7, 3]
	3	[53, 55, 49, 51, 21, 23, 17, 19]
	4	[45, 47, 41, 43, 61, 63, 57, 59, 13, 15, 9, 11, 29, 31, 25, 27]
	5	[38, 40, 34, 36, 46, 48, 42, 44, 54, 56, 50, 52, 62, 64, 58, 60, 6, 8, 2, 4, 14, 16, 10, 12, 22, 24, 18, 20, 30, 32, 26, 28]

Example 3.2. In the proposed MD mapping method, let us set $m = 4$ (16-QAM), $N = 2$, and $\boldsymbol{l} = [1, 1, 1, 0, 0, 1, 1, 1]$. For this example, $\boldsymbol{a}_1$ is made of the four least significant bits of $\boldsymbol{l}$, i.e., $\boldsymbol{a}_1 = [0, 1, 1, 1]$. In step $i = 1$, $\boldsymbol{a}_1$ is mapped to a vector of two QPSK symbols following the proposed method in [12], which results in the QPSK symbol-vector $\boldsymbol{P} = [P_4, P_2]$. Then, $\boldsymbol{P}$ is converted to the 16-QAM symbol-vector $\boldsymbol{x}_1$ using $\boldsymbol{\gamma}$. By applying

(3.3) and setting $\boldsymbol{\gamma} = [11, 3, 1, 9]$ (c.f., Table 3.2 for 16-QAM), $\boldsymbol{x}_1$ is obtained as:

$$\boldsymbol{x}_1 = [S_{\gamma^{(4)}}, S_{\gamma^{(2)}}] = [S_9, S_3]. \tag{3.10}$$

The vector of the position-indexes corresponding to the symbol-vector $\boldsymbol{x}_1$ is $\boldsymbol{j}_1 = [9, 3]$.

Example 3.3. Using example 3.2, and for step $i = 2$, $\boldsymbol{a}_2$ is equal to $[1, 0, 0, 1, 1, 1]$ and then $\boldsymbol{b}_2$ equals $[1, 0]$. Let us consider that $\boldsymbol{a}_2$ is mapped to symbol-vector $\boldsymbol{x}_2 = [x_2^{(1)}, x_2^{(2)}]$ in this step. Since $\boldsymbol{b}_2$ has an odd Hamming weight and $b_2^{(1)} = 1$, according to (3.9) $\boldsymbol{\beta}_{2,1}$ equals $\boldsymbol{\beta}_{O1}$. From Table 3.1, $\boldsymbol{\alpha}_1 = [1, 3, 9, 11]$ and from Table 3.3, $\boldsymbol{\beta}_{O1} = [12, 10, 4, 2]$. Since $\boldsymbol{j}_1$ is equal to $[9, 3]$ (see Example 3.2), then $j_1^{(1)} = \alpha_1^{(q)}$ when $q = 3$. Using (3.5) and setting $q = 3$, the result is $j_2^{(1)} = \boldsymbol{\beta}_{2,1}^{(3)} = 4$. Similarly, since $b_2^2 = 0$, then $\boldsymbol{\beta}_{2,2}$ equals $\boldsymbol{\beta}_{O0}$, where $\boldsymbol{\beta}_{O0} = [11, 9, 3, 1]$ (c.f., Table III for $i = 2$). Moreover, when $q = 2$, $j_1^{(2)} = \alpha_1^{(q)}$. Using (3.5) and setting $q = 2$, the result is $j_2^{(2)} = \boldsymbol{\beta}_{2,2}^{(2)} = 9$. As a consequence, $\boldsymbol{j}_2$ equals $[4, 9]$, which means that $\boldsymbol{x}_1$ will be transformed to $\boldsymbol{x}_2 = [S_4, S_9]$. In other words, in step $i = 2$, $\boldsymbol{a}_2$ is mapped to $\boldsymbol{x}_2 = [S_4, S_9]$.

In step $i = 3$, $\boldsymbol{a}_3$ equals $[1, 1, 1, 0, 0, 1, 1, 1]$, and then, $\boldsymbol{b}_3$ is equal to $[1, 1]$. The Hamming weight of $\boldsymbol{b}_3$ is even and both elements of $\boldsymbol{b}_3$ are equal to 1. As a result, in order to determine the elements of $\boldsymbol{j}_3 = [j_3^{(1)}, j_3^{(2)}]$, we set $\boldsymbol{\beta}_{3,1} = \boldsymbol{\beta}_{E1}$ and $\boldsymbol{\beta}_{3,2} = \boldsymbol{\beta}_{E1}$. From Table 3.3 for $i = 3$, $\boldsymbol{\beta}_{E1}$ equals $[5, 6, 7, 8, 13, 14, 15, 16]$ and from Table 3.1 for 16-QAM, $\boldsymbol{\alpha}_2$ is equal to $[1, 2, 3, 4, 9, 10, 11, 12]$. Furthermore, in step $i = 2$, $\boldsymbol{j}_2$ equals $[4, 9]$. As a result, $j_2^{(1)} = \alpha_2^{(q)}$ when $q = 4$, and $j_2^{(2)} = \alpha_2^{(q)}$ when $q = 5$. By applying (3.5), one can obtain $j_3^{(1)} = \beta_{E1}^{(4)} = 8$ and $j_3^{(2)} = \beta_{E1}^{(5)} = 13$. Consequently, $\boldsymbol{j}_3$ equals $[8, 13]$, which means that $\boldsymbol{x}_2$ will be transformed to $\boldsymbol{x}_3 = [S_8, S_{13}]$. In other words, $\boldsymbol{l}$ is finally mapped to symbol-vector $\boldsymbol{x}_3 = [S_8, S_{13}]$.

Example 3.4. Table 3.5 illustrates the proposed 4-D mapping using 16-QAM symbols. In this table, the decimal label in the $(j + 1, k + 1)^{th}$ entry is mapped to symbol-vector $\boldsymbol{x} = [S_j, S_k]$. For example, the decimal label 231, which corresponds to binary label $\boldsymbol{l} = [1, 1, 1, 0, 0, 1, 1, 1]$, is the $(9, 14)^{th}$ entry of Table 3.5 and is mapped to symbol-vector $\boldsymbol{x} = [S_8, S_{13}]$.

3.3 Numerical results and discussion

In this section, we provide a selection of select numerical examples to demonstrate the performance and advantages of our proposed MD mapping for BICM-ID systems. We compare our mappings with random mappings and also with the mappings that are optimized using well-known BSA. However, the BSA becomes intractable for MD modulations

Table 3.5: Our proposed 4-D 16-QAM mapping.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
S_1	0	17	14	31	65	80	79	94	3	18	13	28	66	83	76	93
S_2	33	48	47	62	96	113	110	127	34	51	44	61	99	114	109	124
S_3	5	20	11	26	68	85	74	91	6	23	8	25	71	86	73	88
S_4	36	53	42	59	101	116	107	122	39	54	41	56	102	119	104	121
S_5	129	144	143	158	192	209	206	223	130	147	140	157	195	210	205	220
S_6	160	177	174	191	225	240	239	254	163	178	173	188	226	243	236	253
S_7	132	149	138	155	197	212	203	218	135	150	137	152	198	215	200	217
S_8	165	180	171	186	228	245	234	251	166	183	168	185	231	246	233	248
S_9	9	24	7	22	72	89	70	87	10	27	4	21	75	90	69	84
S_{10}	40	57	38	55	105	120	103	118	43	58	37	52	106	123	100	117
S_{11}	12	29	2	19	77	92	67	82	15	30	1	16	78	95	64	81
S_{12}	45	60	35	50	108	125	98	115	46	63	32	49	111	126	97	112
S_{13}	136	153	134	151	201	216	199	214	139	154	133	148	202	219	196	213
S_{14}	169	184	167	182	232	249	230	247	170	187	164	181	235	250	229	244
S_{15}	141	156	131	146	204	221	194	211	142	159	128	145	207	222	193	208
S_{16}	172	189	162	179	237	252	227	242	175	190	161	176	238	255	224	241

with a large alphabet size, e.g., 6-D 64-QAM, due to its computational time constraints. The random mappings for AWGN and Rayleigh fading channels are obtained by selecting the best mappings from a large number of randomly generated mappings. We consider a rate-1/2 convolutional code with the generator polynomial of $(13, 15)_8$. The length of interleaver used is 10008 bits. All BER curves are presented with seven iterations, and all reported gains are measured at a BER of 10^{-6}. Also, throughout this section, SNR refers to the SNR per bit, i.e., $\frac{E_b}{N_0}$.

3.3.1 Performance over AWGN channel

As previously discussed, there are two important parameters for a mapping, i.e., N_{min} and $\hat{d}^2_{min}$, that are relevant to BER performance of BICM-ID systems over AWGN channel. In Table 3.6, we compare N_{min} and $\hat{d}^2_{min}$ values of our MD mappings using 16-QAM and 64-QAM with those of the well-known BSA mappings that are optimized for AWGN channels and random mappings. In this table, BSA MD 64-QAM mapping for higher dimension, e.g., $N = 3$, is not reported as it could not be obtained due to the computational complexity. Table 3.6 clearly shows that our mappings offer smaller values of N_{min} compared to their counterparts. So, the proposed MD mappings will improve the BER performance of BICM-ID at low SNR values over AWGN channels. This is confirmed in Fig. 3.2 which plots the BER of various mappings over an AWGN channel. As it can be observed from this figure, the proposed mappings outperform the BSA mappings by 1.5 dB, 2.5 dB, and 3.5 dB for 4-D 16-QAM, 6-D 16-QAM, and 4-D 64-QAM, respectively in the low SNR region. The error rate improvement with the proposed mappings over random mappings is even

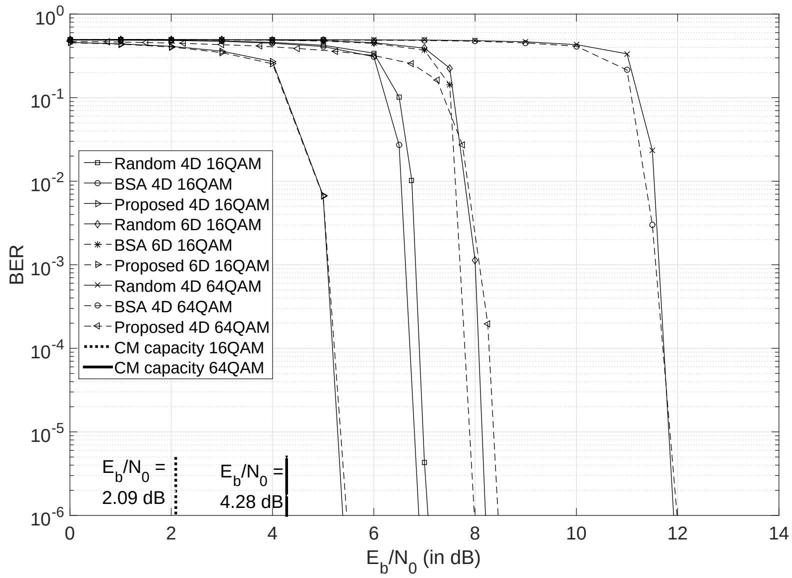

Figure 3.2: BER performance over an AWGN channel.

larger. Table 3.6 also shows that our mappings yield larger values of $\hat{d}^2_{min}$ compared to the BSA and random mappings. Thus, the proposed mappings result in improved BER performance in the high SNR region over AWGN channels. This can be observed from the plotted error-floor bounds in Fig. 3.3. It is worth noting that the gap between N_{min} for different mappings in Table 3.6, can explain the gap between the corresponding BER curves in Fig. 3.2. This is because for large constellations such as MD constellations, the impact of N_{min} on BER is more significant than that of $\hat{d}^2_{min}$.

Table 3.6: Comparison of N_{min} and $\hat{d}^2_{min}$ for different mappings.

Mapping	$N=2$		$N=3$	
	N_{min}	$\hat{d}^2_{min}$	N_{min}	$\hat{d}^2_{min}$
Random MD 16-QAM	4.0412	0.2069	6.0137	0.1333
BSA MD 16-QAM	3.7305	1.2069	5.8254	1.3333
Proposed MD 16-QAM	2.2500	2.4000	2.2778	2.6667
Random MD 64-QAM	6.0114	0.0476	8.9979	0.0317
BSA MD 64-QAM	5.8240	1.1905	-	-
Proposed MD 64-QAM	2.3214	2.2857	2.3571	2.5397

3.3.2 Performance over Rayleigh fading channels

For Rayleigh fading channels, $\Phi(\mu, \boldsymbol{\chi})$ and $\hat{\Phi}(\mu, \boldsymbol{\chi})$ are two important mapping parameters to compare the BER performance of BICM-ID systems. We compare the values of $\Phi(\mu, \boldsymbol{\chi})$

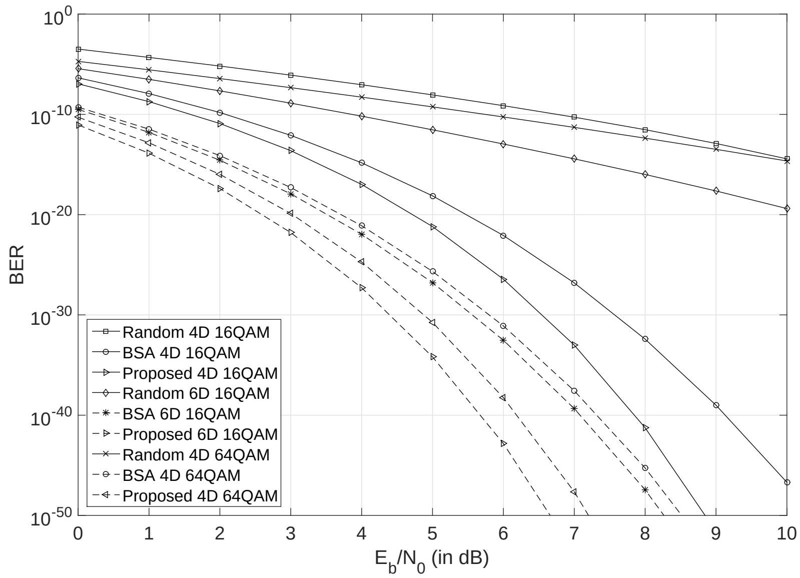

Figure 3.3: Error-floor bounds over an AWGN channel.

and $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for the mappings already considered in the previous section in Table 3.7. The proposed mappings offer larger values of $\Phi(\mu, \boldsymbol{\chi})$ in comparison with the BSA and random mappings. This results in better BER performance in the low SNR region with our mappings. The BER plots in Fig. 3.4 show that the proposed mappings offer gains of 1.5 dB, 1.6 dB, and 3 dB for 4-D 16-QAM, 6-D 16-QAM, and 4-D 64-QAM, respectively, compared to the BSA mappings that are optimized for Rayleigh fading channels. The performance gain with respect to the random mappings is larger. Similar to the AWGN channel, the gap between BER curves in Fig. 3.4 can be explained by the gap between the corresponding $\Phi(\mu, \boldsymbol{\chi})$ in Table 3.7. From the listed values of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ in Table 3.7, we observe that the proposed mappings also increase the values of $\hat{\Phi}(\mu, \boldsymbol{\chi})$. Therefore, our mappings offer improved error-floor bounds as illustrated in Fig. 3.5.

Table 3.7: Comparison of $\Phi(\mu, \boldsymbol{\chi})$ and $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for different mappings.

Mapping	$N = 2$		$N = 3$	
	$\Phi(\mu, \boldsymbol{\chi})$	$\hat{\Phi}(\mu, \boldsymbol{\chi})$	$\Phi(\mu, \boldsymbol{\chi})$	$\hat{\Phi}(\mu, \boldsymbol{\chi})$
Random MD 16-QAM	0.2012	1.4350	0.1335	1.4934
BSA MD 16-QAM	0.2026	2.5814	0.1342	2.8047
Proposed MD 16-QAM	0.2151	2.8491	0.1446	2.9741
Random MD 64-QAM	0.0478	1.1688	0.0318	1.4370
BSA MD 64-QAM	0.0481	2.6899	-	-
Proposed MD 64-QAM	0.0579	2.8166	0.0392	2.9040

Please note that the BSA requires a very long time to finish only one round of the search algorithm for 6-D 16-QAM. But, our proposed method is a heuristic method and generates the proposed mappings instantaneously. Moreover, our proposed mappings improve the system error performance compared to the BSA mappings. This shows the efficiency of our proposed method compared to the BSA.

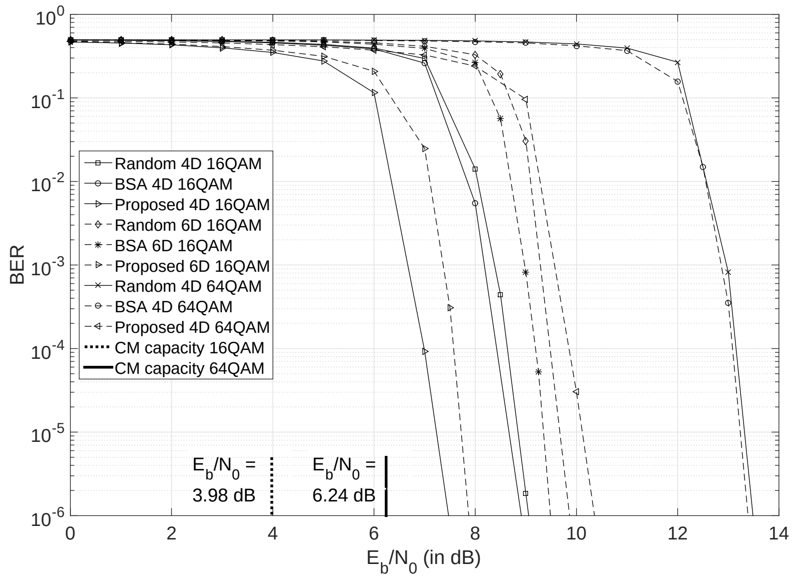

Figure 3.4: BER performance over a Rayleigh fading channel.

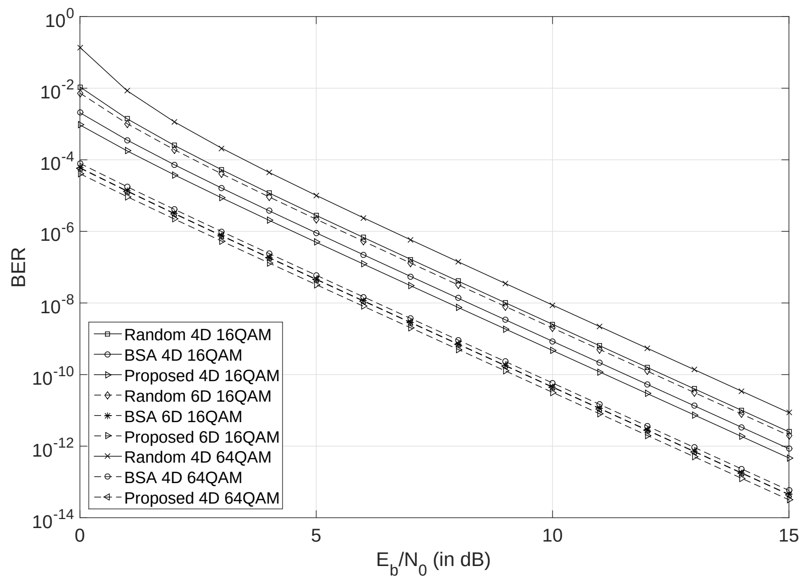

Figure 3.5: Error-floor bounds over a Rayleigh fading channel.

Chapter 4

Multi-dimensional Mapping of Higher Order QAM for BICM-ID Over Rayleigh Fading Channels

The number of possible mappings for an MD modulation increases exponentially as the order of the employed 2-D basic modulation increases. For example, there are $4! = 24$ distinct mappings for a 2-D QPSK while for a 4-D QPSK modulation, there are $16! = 2.1 \times 10^{13}$ possible mappings. For the higher order modulations such as 1024-QAM, the number of possible MD mappings approaches infinity. This makes it complicated to find good mappings for larger constellations. Indeed, the large number of possible mappings is the main pitfall of all the proposed computer search based methods. For instance, the well-known mapping search method known as the BSA [9], becomes intractable if the order of the modulation increases [7]. As a result, the BSA is not applicable to directly search for good MD mappings for higher order modulations. Analytical mapping search methods are investigated in [13] and [16] for higher order modulations. However, due to a high computational complexity, in these works, the authors have not reported results for constellations larger than 64-QAM.

Motivated by the above discussions, the present study focuses on finding good MD mappings for a wide range of modulations including higher order modulations. The objective is to improve the error performance of the BICM-ID system over Rayleigh fading channels. First, we introduce a MD mapping method employing four 2-D mappings. Then, a lower bound for the harmonic mean of the MSED [6] is derived for this MD mapping. We next develop mutual cost functions and minimize them over the 2-D mappings in order to achieve a MD mapping with a large value for the harmonic mean of the MSED. The proposed method is a low complexity approach and can be easily applied to find good MD mappings for higher order modulations such as 512- and 1024-QAM. The reported numerical results confirm the efficiency of the achieved mappings. In Chapter 2, we developed MD mappings for two specific modulations, i.e., 16- and 64-QAM. In the current chapter, we develop a generalized mapping method to construct MD mappings for a vast range of modulations that includes 2^m-QAM ($m = 4, \cdots, 10$). Moreover, the proposed method in this chapter, is not limited to QAM and it is applicable to any kind of constellation.

4.1 Proposed mapping method

As mentioned earlier, for a 2^m-ary signal constellation there are $2^m!$ possible mappings. In fact, a comprehensive computer search to find good mappings becomes intractable quickly as the modulation order increases. The well-known BSA mapping search method cannot be used directly to obtain good mappings. Therefore, we propose an efficient technique to find good MD mappings for BICM-ID systems over Rayleigh fading channels.

We consider the asymptotic performance of BICM-ID in which the ideal *a priori* information about the decoded bits is available at the demodulator. In this case, to detect a particular bit carried by the received signal, the demodulator has perfect information about the remaining $(mN-1)$-bits carried by the signal. Thus, the modulation is converted to a binary modulation and Euclidean distances can easily be increased by employing a carefully designed mapping. As it is discussed in 1.6.2, the asymptotic performance of BICM-ID over Rayleigh fading channels is influenced by the harmonic mean of the MSED after feedback, which is calculated for a given mapping function, μ, applied to signal set $\boldsymbol{\chi}$. According to 1.10, for a $2N$-D mapping of a 2^m-ary constellation, the harmonic mean of the MSED after feedback is given by

$$\hat{\Phi}(\mu, \boldsymbol{\chi}) = \left(\frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}_b^i} \frac{1}{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2} \right)^{-1}, \tag{4.1}$$

where $\boldsymbol{x} = [x^1, x^2, \cdots, x^N]$ is a $2N$-D signal point and $\boldsymbol{\chi}_b^i$ is the subset of $\boldsymbol{\chi}$ whose labels take value b at the i^{th} bit position, and $\hat{\boldsymbol{x}} = [\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^N]$ is a signal point in $\boldsymbol{\chi}_{\bar{b}}^i$ whose label is different with that of $\boldsymbol{x}$ only in the i^{th} bit position. For BICM-ID with a particular code, a larger value of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ offers a lower error floor [34]. However, maximizing $\hat{\Phi}(\mu, \boldsymbol{\chi})$ is a complicated problem even for 2-D modulations such as 64-QAM [13]. Therefore, we propose an innovative approach to generate MD mappings using 2-D mappings. Next, we develop cost functions that are optimized over the employed 2-D mappings to achieve a high value of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for the MD mapping. Our cost functions are simple and give excellent results, even for higher order modulations such as 1024-QAM.

Let $\boldsymbol{l} = [l^1, l^2, \cdots, l^{mN}]$ be an mN-bit binary label. Equivalently $\boldsymbol{l} = [\boldsymbol{l}^1, \boldsymbol{l}^2, \cdots, \boldsymbol{l}^N]$ where $\boldsymbol{l}_i$ is an m-bit binary label and is given by

$$\boldsymbol{l}_i = [l^{(i-1)m+1}, \cdots, l^{im}]; \quad i = 1, ..., N. \tag{4.2}$$

Suppose that $\mathcal{L}$ denotes the set of all mN-bit binary labels and $\mathcal{L}_e$ and $\mathcal{L}_o$ represent the subset of all $\boldsymbol{l} \in \mathcal{L}$ with even and odd Hamming weights, respectively. The MD mapping problem can be broken into four mappings in 2-D signal space as described below. According to the proposed MD mapping function, i.e., μ, label $\boldsymbol{l}$ is mapped to the $2N$-D signal point $\boldsymbol{x} = [x^1, \cdots, x^N]$ as given below

$$x_i = \begin{cases} \lambda_{el}(\boldsymbol{l}_i) & \text{if } i = 1 \ , \ \boldsymbol{l} \in \mathcal{L}_e, \\ \lambda_{ol}(\boldsymbol{l}_i) & \text{if } i = 1 \ , \ \boldsymbol{l} \in \mathcal{L}_o, \\ \lambda_{er}(\boldsymbol{l}_i) & \text{if } i \geq 2 \ , \ \boldsymbol{l} \in \mathcal{L}_e, \\ \lambda_{or}(\boldsymbol{l}_i) & \text{if } i \geq 2 \ , \boldsymbol{l} \in \mathcal{L}_o, \end{cases} \ ; \quad i = 1, ..., N, \tag{4.3}$$

where λ_{el}, λ_{ol}, λ_{er}, and λ_{or} are 2-D mapping functions, which will be discussed later in this section. In the applied mapping, let $\boldsymbol{\chi}_e$ and $\boldsymbol{\chi}_o$ represent the subset of signal points in $\boldsymbol{\chi}$ whose labels belong to $\mathcal{L}_e$ and $\mathcal{L}_o$, respectively. Without loss of generality, assume that $\boldsymbol{x} \in \boldsymbol{\chi}_e$ and $\hat{\boldsymbol{x}} \in \boldsymbol{\chi}_o$ where $\hat{\boldsymbol{x}} = [\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^N]$ is a signal point whose label is different with that of $\boldsymbol{x}$ only in one bit position. We partition the 2-D signal constellation χ into two separate subsets with equal cardinalities and denote them as χ_{el} and χ_{ol}. Then, we limit the first element in $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, i.e., x_1 and $\hat{x}_1$, to belong to χ_{el} and χ_{ol}, respectively. In (4.3), $\lambda_{el}(.)$ and $\lambda_{ol}(.)$, each map an m-bit label to a 2-D signal point chosen from χ_{el} and χ_{ol}, respectively. However, χ_{el} and χ_{ol} involve only 2^{m-1} signal points while there are 2^m distinct m-bit labels. As a consequence, each signal point in χ_{el} and χ_{ol} should be mapped by two m-bit labels simultaneously. In order to obtain a one-to-one MD mapping function, we restrict the two labels that are mapped to a particular signal point in either χ_{el} or χ_{ol} to be different in an odd number of bit positions. For simplicity, we assume that these two labels are different just in the first bit position. On the other hand, there is no constraint on $\lambda_{er}(.)$ and $\lambda_{or}(.)$ except they need to be bijective.

Proposition 4.1. *In the proposed MD mapping function, μ, there is a one-to-one correspondence between MD signal points and binary labels.*

Proof. It is obvious that the Hamming distance between two labels from a particular label-set, i.e., $\mathcal{L}_e$ or $\mathcal{L}_o$, is even. However, two labels one from $\mathcal{L}_e$ and the other from $\mathcal{L}_o$ have an odd Hamming distance from each other. Since there is no common 2-D signal point between χ_{el} and χ_{ol}, there is no common $2N$-D signal point between $\boldsymbol{\chi}_e$ and $\boldsymbol{\chi}_o$. As a result, none of the $2N$-D signal points will be mapped simultaneously by a label from $\mathcal{L}_e$ and a label from $\mathcal{L}_o$. Therefore, it is sufficient to prove that there is a one-to-one correspondence between labels from $\mathcal{L}_e$ and signal points from $\boldsymbol{\chi}_e$ and similarly between labels in $\mathcal{L}_o$ and signal points in $\boldsymbol{\chi}_o$. In what follows, we prove this for the even subsets, i.e., for labels in

$\mathcal{L}_e$ and signal points in $\boldsymbol{\chi}_e$.

Assume that $\boldsymbol{l} = [l^1, l^2, \cdots, l^{mN}]$ and $\tilde{\boldsymbol{l}} = [\tilde{l}^1, \tilde{l}^2, \cdots, \tilde{l}^{mN}]$ are two labels in $\mathcal{L}_e$ and are mapped to $\boldsymbol{x} = [x^1, \cdots, x^N]$ and $\tilde{\boldsymbol{x}} = [\tilde{x}^1, \cdots, \tilde{x}^N]$, respectively where both $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ are in $\boldsymbol{\chi}_e$. Let us define $\boldsymbol{l}_i$ and $\tilde{\boldsymbol{l}}_i$ as the i^{th} m-tuple bits of $\boldsymbol{l}$ and $\tilde{\boldsymbol{l}}$, respectively. Then $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_{mN}]$ and $\tilde{\boldsymbol{l}} = [\tilde{\boldsymbol{l}}_1, \tilde{\boldsymbol{l}}_2, \cdots, \tilde{\boldsymbol{l}}_{mN}]$. Based on the relation between $\boldsymbol{l}_i$ and $\tilde{\boldsymbol{l}}_i$ for different values of i, there are two possible cases as follows:

Case 1: There exists a value of i ($i \geq 2$) such that $\boldsymbol{l}_i \neq \tilde{\boldsymbol{l}}_i$. Let $j \geq 2$, then according to (4.3) the same one-to-one mapping function, i.e., $\lambda_{er}(.)$, is used to map $\boldsymbol{l}_j$ to x_j and $\tilde{\boldsymbol{l}}_j$ to $\tilde{x}_j$. Therefore, because $\boldsymbol{l}_j \neq \tilde{\boldsymbol{l}}_j$, we have

$$x_j \neq \tilde{x}_j \Rightarrow \boldsymbol{x} \neq \tilde{\boldsymbol{x}}. \tag{4.4}$$

Case 2: $\boldsymbol{l}_i = \tilde{\boldsymbol{l}}_i$ for all $i \geq 2$. In this case, $\boldsymbol{l}_i \neq \tilde{\boldsymbol{l}}_i$ only when $i = 1$, and as a result, the Hamming distance between $\boldsymbol{l}$ and $\tilde{\boldsymbol{l}}$ is equal to the Hamming distance between $\boldsymbol{l}_1$ and $\tilde{\boldsymbol{l}}_1$. Since $\boldsymbol{l}$ and $\tilde{\boldsymbol{l}}$ belong to $\mathcal{L}_e$, they have an even Hamming distance from each other. Consequently, the Hamming distance between $\boldsymbol{l}_1$ and $\tilde{\boldsymbol{l}}_1$ is even as well. However, the two labels that are mapped to each symbol in χ_{el} have an odd Hamming distance from each other. Therefore, because $\lambda_{el}(\boldsymbol{l}_1) \neq \lambda_{el}(\tilde{\boldsymbol{l}}_1)$, we have

$$x_1 \neq \tilde{x}_1 \Rightarrow \boldsymbol{x} \neq \tilde{\boldsymbol{x}}. \tag{4.5}$$

From (4.4) and (4.5), it is concluded that in the proposed mapping function, different labels from $\mathcal{L}_e$ are mapped to different signal points in $\boldsymbol{\chi}_e$. In a similar way, it can be proven that the different labels from $\mathcal{L}_o$ are mapped to the different signal points in $\boldsymbol{\chi}_o$. As a result, the proposed MD mapping function is bijective.

$\square$

4.2 Development and optimization of cost functions

As it is mentioned previously, maximizing $\hat{\Phi}(\mu, \boldsymbol{\chi})$ is a complicated task. In what follows, we break $\hat{\Phi}(\mu, \boldsymbol{\chi})^{-1}$ into two separated parts. Next, we derive a cost function for each part to maximize $\hat{\Phi}(\mu, \boldsymbol{\chi})$. Both cost functions operate in 2-D signal space rather than MD signal space. As such, they have much lower complexity. We use (4.1) to write

$$\hat{\Phi}(\mu,\boldsymbol{\chi})^{-1} = \frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \left(\sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \frac{1}{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2} + \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_o}} \frac{1}{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2} \right), \tag{4.6}$$

where $\boldsymbol{\chi}_e$ and $\boldsymbol{\chi}_o$ are the same size. Moreover, when a given $\boldsymbol{x}$ is in $\boldsymbol{\chi}_e$ then the corresponding $\hat{\boldsymbol{x}}$ belongs to $\boldsymbol{\chi}_o$ and vice versa. Therefore, the two parts in (4.6) are equivalent, and as a result, we have

$$\hat{\Phi}(\mu,\boldsymbol{\chi})^{-1} = 2\Omega(\mu,\boldsymbol{\chi}), \tag{4.7}$$

where $\Omega(\mu,\boldsymbol{\chi})$ is given by

$$\Omega(\mu,\boldsymbol{\chi}) = \frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \frac{1}{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2}. \tag{4.8}$$

Since $\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2 = \sum_{j=1}^{N} |x_j - \hat{x}_j|^2$, then (4.8) can be rewritten as

$$\Omega(\mu,\boldsymbol{\chi}) = \frac{1}{mN2^{mN}} \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \frac{1}{\sum_{j=1}^{N} |x_j - \hat{x}_j|^2}. \tag{4.9}$$

Proposition 4.2. *Let $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]$ is a vector of positive real numbers. Then we have*

$$\frac{1}{\sum_{i=1}^{N} y_i} \leqslant \frac{1}{N^2} \sum_{j=1}^{N} \frac{1}{y_j}. \tag{4.10}$$

Proof. As $f(y) = \frac{1}{y}$ is convex on $\mathbb{R}_+$, so

$$f(\sum_{i=1}^{N} \lambda_i y_i) \leq \sum_{i=1}^{N} \lambda_i f(y_i), \tag{4.11}$$

where $\sum_{i=1}^{N} \lambda_i = 1$. Let $\lambda_i = \frac{1}{N}$. Then

$$f(\frac{1}{N} \sum_{i=1}^{N} y_i) \leq \sum_{i=1}^{N} \frac{1}{N} f(y_i), \tag{4.12}$$

i.e.,

$$\frac{1}{\frac{1}{N}\sum_{i=1}^{N} y_i} \leq \sum_{i=1}^{N} \frac{1}{N}\frac{1}{y_i} \tag{4.13}$$
$$\Rightarrow \frac{N}{\sum_{i=1}^{N} y_i} \leq \frac{1}{N}\sum_{i=1}^{N} \frac{1}{y_i}$$
$$\Rightarrow \frac{1}{\sum_{i=1}^{N} y_i} \leq \frac{1}{N^2}\sum_{i=1}^{N} \frac{1}{y_i}.$$

□

Applying (4.10) in (4.9), we can write

$$\Omega(\mu, \boldsymbol{\chi}) \leqslant K\Psi(\mu, \boldsymbol{\chi}), \tag{4.14}$$

where K is a constant value and $\Psi(\mu, \boldsymbol{\chi})$ is defined as

$$\Psi(\mu, \boldsymbol{\chi}) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \sum_{j=1}^{N} \frac{1}{|x_j - \hat{x}_j|^2}.$$

We can decompose $\Psi(\mu, \boldsymbol{\chi})$ as

$$\Psi(\mu, \boldsymbol{\chi}) = \Psi_l(\mu, \boldsymbol{\chi}) + \Psi_r(\mu, \boldsymbol{\chi}), \tag{4.15}$$

where $\Psi_l(\mu, \boldsymbol{\chi})$ and $\Psi_r(\mu, \boldsymbol{\chi})$ are given by

$$\Psi_l(\mu, \boldsymbol{\chi}) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \frac{1}{|x_1 - \hat{x}_1|^2} \tag{4.16}$$

and

$$\Psi_r(\mu, \boldsymbol{\chi}) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{x}\in\boldsymbol{\chi}_b^i \\ \boldsymbol{x}\in\boldsymbol{\chi}_e}} \sum_{j=2}^{N} \frac{1}{|x_j - \hat{x}_j|^2}. \tag{4.17}$$

Let $\boldsymbol{l} = [l^1, l^2, \cdots, l^{mN}]$ and $\hat{\boldsymbol{l}} = [\hat{l}^1, \hat{l}^2, \cdots, \hat{l}^{mN}]$ are two mN-bit labels, which are different only in the i^{th} bit position, and are mapped to $\boldsymbol{x} = [x^1, \cdots, x^N]$ and $\hat{\boldsymbol{x}} = [\hat{x}^1, \cdots, \hat{x}^N]$, respectively. We define $\boldsymbol{l}_i$ and $\tilde{\boldsymbol{l}}_i$ respectively as the i^{th} m-tuple bits of $\boldsymbol{l}$ and $\hat{\boldsymbol{l}}$, and rewrite $\boldsymbol{l} = [\boldsymbol{l}^1, \boldsymbol{l}^2, \cdots, \boldsymbol{l}^{mN}]$ and $\tilde{\boldsymbol{l}} = [\tilde{\boldsymbol{l}}^1, \tilde{\boldsymbol{l}}^2, \cdots, \tilde{\boldsymbol{l}}^{mN}]$. Then, (4.16) is equivalent to

$$\Psi_l'(\lambda_{el}, \lambda_{ol}, \mathcal{L}) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{l} \in \mathcal{L}_b^i \\ \boldsymbol{l} \in \mathcal{L}_e}} \frac{1}{|\lambda_{el}(\boldsymbol{l}_1) - \lambda_{ol}(\hat{\boldsymbol{l}}_1)|^2}, \tag{4.18}$$

where $\mathcal{L}_b^i \in \mathcal{L}$ is the subset of labels with value b in their i^{th} bit position. For a given m-bit sequence $\boldsymbol{l}_i$, $\hat{\boldsymbol{l}}_i$ can take $(m+1)$ distinct m-bit sequences, where each one is the same as $\boldsymbol{l}_i$ or different from $\boldsymbol{l}_i$ only in one bit position. For example, if $m = 4$ and $\boldsymbol{l}_i = [0,0,0,0]$, $\hat{\boldsymbol{l}}$ can take either of the 5 labels in $\{[0,0,0,0],[0,0,0,1],[0,0,1,0],[0,1,0,0],[1,0,0,0]\}$. Let $\boldsymbol{\alpha} = [\alpha^1, \cdots, \alpha^m]$ and $\boldsymbol{\beta} = [\beta^1, \cdots, \beta^m]$ are two binary sequences, where $\boldsymbol{\beta}$ has the Hamming distance of either zero or one from $\boldsymbol{\alpha}$. The set of $(m+1)$ possibilities for $\boldsymbol{\beta}$ is denoted by $\mathcal{B}$. Assume that for a given i, $\boldsymbol{l}_i = \boldsymbol{\alpha}$ and $\hat{\boldsymbol{l}}_i = \boldsymbol{\beta}$. Then (4.18) is equivalent to

$$\psi_l(\lambda_{el}, \lambda_{ol}, \chi_{el}, \chi) = \sum_{\boldsymbol{\alpha}} \sum_{\boldsymbol{\beta} \in \mathcal{B}} \frac{a_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{(l)}}{|\lambda_{el}(\boldsymbol{\alpha}) - \lambda_{ol}(\boldsymbol{\beta})|^2} \tag{4.19}$$

where $a_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{(l)}$ is given by

$$a_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{(l)} = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{l} \in \mathcal{L}_b^i \\ \boldsymbol{l} \in \mathcal{L}_e}} [\boldsymbol{l}_1 = \boldsymbol{\alpha}, \hat{\boldsymbol{l}}_1 = \boldsymbol{\beta}], \tag{4.20}$$

where $[x]$ is an indicator function and is defined as

$$[x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

Similarly, (4.17) is equivalent to

$$\Psi_r'(\lambda_{er}, \lambda_{or}, \mathcal{L}) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{l} \in \mathcal{L}_b^i \\ \boldsymbol{l} \in \mathcal{L}_e}} \sum_{j=2}^{N} \frac{1}{|\lambda_{er}(\boldsymbol{l}_j) - \lambda_{or}(\hat{\boldsymbol{l}}_j)|^2}. \tag{4.22}$$

The m-bit elements $\boldsymbol{l}_i$ in $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_N]$ are independent from one another for all i. Then (4.22) is equivalent to

$$\psi_r(\lambda_{er}, \lambda_{or}, \chi) = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{l} \in \mathcal{L}_b^i \\ \boldsymbol{l} \in \mathcal{L}_e}} \frac{N-1}{|\lambda_{er}(\boldsymbol{l}_2) - \lambda_{or}(\hat{\boldsymbol{l}}_2)|^2}, \tag{4.23}$$

which can be rewritten as

$$\psi_r(\lambda_{er}, \lambda_{or}, \chi) = \sum_{\boldsymbol{\alpha}} \sum_{\boldsymbol{\beta} \in \mathcal{B}} \frac{a^{(r)}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(N-1)}{|\lambda_{er}(\boldsymbol{\alpha}) - \lambda_{or}(\boldsymbol{\beta})|^2}, \tag{4.24}$$

where $a^{(r)}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is computed as

$$a^{(r)}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \sum_{i=1}^{mN} \sum_{b=0}^{1} \sum_{\substack{\boldsymbol{l} \in \mathcal{L}_b^i \\ \boldsymbol{l} \in \mathcal{L}_e}} [\boldsymbol{l}_2 = \boldsymbol{\alpha}, \hat{\boldsymbol{l}}_2 = \boldsymbol{\beta}]. \tag{4.25}$$

Using (4.7), (4.14), and (4.15), a lower bound of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ can be derived as follows

$$\begin{aligned} \hat{\Phi}^{-1}(\mu, \boldsymbol{\chi}) &\leqslant 2K \left(\Psi_l(\mu, \boldsymbol{\chi}) + \Psi_r(\mu, \boldsymbol{\chi})\right) \\ &\Rightarrow \Delta \leqslant \hat{\Phi}(\mu, \boldsymbol{\chi}), \end{aligned} \tag{4.26}$$

where Δ is given by

$$\Delta = \frac{K'}{\Psi_l(\mu, \boldsymbol{\chi}) + \Psi_r(\mu, \boldsymbol{\chi})}. \tag{4.27}$$

Because $\Psi_l(\mu, \boldsymbol{\chi}) \equiv \psi_l(\lambda_{el}, \lambda_{ol}, \chi_{el}, \chi)$ and $\Psi_r(\mu, \boldsymbol{\chi}) \equiv \psi_r(\lambda_{er}, \lambda_{or}, \chi)$, we rewrite Δ as

$$\Delta = \frac{K'}{\psi_l(\lambda_{el}, \lambda_{ol}, \chi_{el}, \chi) + \psi_r(\lambda_{er}, \lambda_{or}, \chi)}. \tag{4.28}$$

Note that (4.28) operates in 2-D signal space rather than MD signal space, and as a result, optimization is much simpler.

Our objective is to maximize Δ and then to calculate the corresponding $\hat{\Phi}(\mu, \boldsymbol{\chi})$. Since $\psi_l(\lambda_{el}, \lambda_{ol}, \chi_{el}, \chi)$ and $\psi_r(\lambda_{er}, \lambda_{or}, \chi)$ are independent from each other, then the maximum value of Δ, Δ_{max} is given by

$$\Delta_{max} = K' \left[\min_{\lambda_{el}, \lambda_{ol}, \chi_{el}} \psi_l(\lambda_{el}, \lambda_{ol}, \chi_{el}, \chi) + \min_{\lambda_{er}, \lambda_{or}} \psi_r(\lambda_{er}, \lambda_{or}, \chi) \right]^{-1}. \tag{4.29}$$

4.2.1 Optimization of cost functions

The minimization of ψ_l and ψ_r in (4.29) can be done using the BSA [9]. To minimize ψ_r, two random mappings are initially considered as λ_{er} and λ_{or}. Then, the BSA is used to minimize the cost function introduced in (4.24) for λ_{er}. In fact, this is a mutual cost function where the cost value for a given symbol in λ_{er}, is computed by using $(m+1)$ corresponding symbols from λ_{or}. Our approach is to minimize this cost function by alternatingly modifying each of λ_{er} and λ_{er}. In other words, the BSA is used to modify λ_{er} to reduce the cost function. After a given number of iterations, λ_{er} and λ_{or} are exchanged. Again, the BSA is used to decrease the cost value. After a certain number of iterations, λ_{er} and λ_{or} are exchanged again and the BSA modifies λ_{er}. This procedure is repeated up to a given number of iterations.

In addition to λ_{el} and λ_{ol}, χ_{el} is another effective argument in computing ψ_l. As there is no constraint on χ_{el}, it is a complex process to optimize ψ_l. To simplify the optimization process, χ_{el} is constrained to involve only the symbols whose labels in the resulting λ_{er} take binary value b in a given bit position. In this chapter, we assume that χ_{el} involves the symbols whose labels in λ_{er} take the value zero in the first bit-position. The functions ψ_l and ψ_r are computed by considering the similar Euclidean distances between two dimensional symbols. As a result, there is a potential advantage in applying the above mentioned constraint on χ_{el} because it will be easier to find a suitable λ_{el} corresponding to a given λ_{ol}. After determining χ_{el} and χ_{ol}, two random mappings are generated as λ_{el} and λ_{ol}. As mentioned previously, two assigned labels to a given symbol by either of λ_{el} and λ_{ol} are different only in the first bit position. Similar to the previous step, the BSA is applied to minimize ψ_l by modifying λ_{el}. Then, λ_{el} is exchanged by λ_{ol} and the BSA minimizes ψ_l by modifying the new λ_{el}. This procedure is repeated up to a given number of iterations.

By executing the proposed algorithm for a certain number of iterations, a local maximum value is calculated using (4.29). The search algorithm is executed several times and each time the corresponding value for $\hat{\Phi}(\mu, \boldsymbol{\chi})$ is calculated. Finally, the modulations corresponding to the maximum obtained $\hat{\Phi}(\mu, \boldsymbol{\chi})$ are chosen. Fig. 4.1 illustrates the flowchart of the proposed algorithm.

Numerical results confirm that the proposed algorithm generates mappings with significantly large values of $\hat{\Phi}(\mu, \boldsymbol{\chi})$. As a result, the obtained mappings would improve the error performance of BICM-ID systems over Rayleigh fading channels.

4.3 Numerical results and discussion

In this section, we provide our resulting mappings and selected numerical results to illustrate the performance and advantage of our proposed MD mappings for BICM-ID.

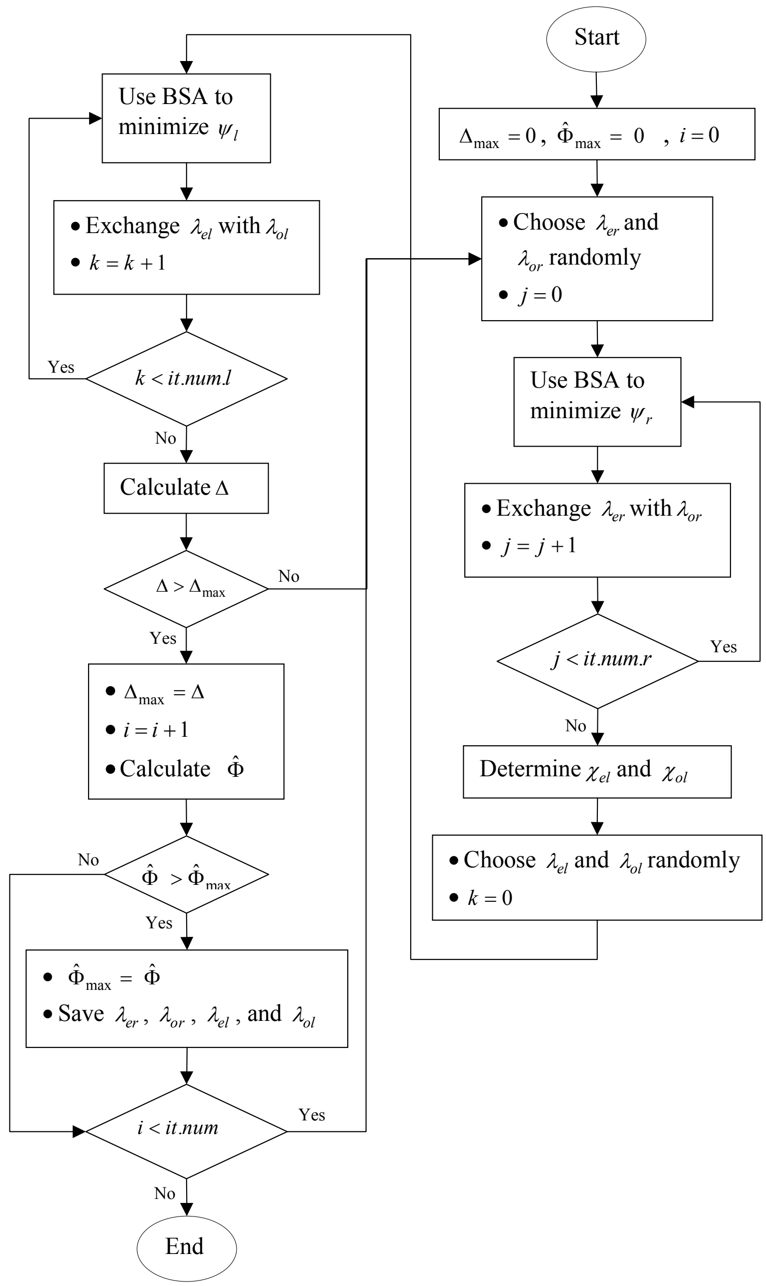

Figure 4.1: Flowchart of the proposed algorithm ($it.num.r$, $it.num.l$, and $it.num$ represent the number of iterations for different loops).

4.3.1 Resulting MD mappings of M-QAM

Our proposed algorithm is used to obtain MD mappings of 2^m-QAM for $m = 4, 5, \cdots, 10$. Tables 4.1-4.11 show the resulting 2-D mappings such as λ_{er}, λ_{or}, λ_{el}, and λ_{ol} in decimal format. In these tables, the resulting 2-D mappings for higher order constellations are indicated in multiple rows. For example in Table 4.3, λ_{er} for 64-QAM is indicated in two rows where the first element in the second row is the label of the 33^{rd} symbol in the constellation. For square QAMs, it is assumed that the symbol order starts from the top left corner in the constellation and increases from top to bottom and from left to right (see Fig. 4.2.(a) as an example for 16-QAM). For cross QAM constellations such as 32-QAM, we consider the symbol order used in [19]. In Tables 4.1-4.11, two labels in the i^{th} parentheses in λ_{el} and λ_{ol} belong to the i^{th} symbol in χ_{el} and χ_{ol}, respectively. For example, Fig. 4.2(b), (c), and (d) illustrate the 16-QAM mappings reported in Table 4.1.

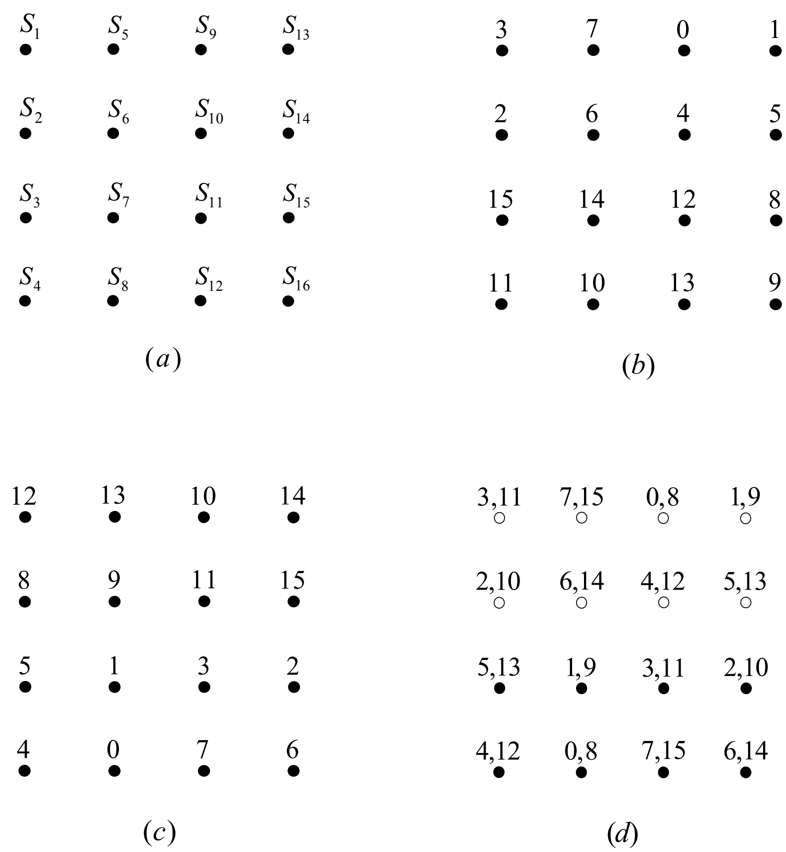

Figure 4.2: (a). Symbol's arrangement in 16-QAM, and achieved 16-QAM mappings in decimal format: (b). λ_{er}, (c). λ_{or}, and (c). λ_{el} (the light symbols), λ_{ol} (the dark symbols).

As mentioned previously, χ_{el} involves the symbols whose binary labels in λ_{er} take the value zero at the first bit position. Equivalently, χ_{el} is constructed by the symbols whose decimal label in λ_{er} is smaller than $\frac{M}{2}$. As a result, for 16-QAM, $\chi_{el} = \{S_1, S_2, S_5, S_6, S_9, S_{10}, S_{13}, S_{14}\}$, where χ_{el} is indicated by light symbols in Fig. 4.2(d).

Table 4.1: Proposed λ_{er}, λ_{or}, λ_{el}, and λ_{ol} for 16-QAM.

λ_{er}	[3	2	15	11	7	6	14	10	0	4	12	13	1	5	8	9]
λ_{or}	[12	8	5	4	13	9	1	0	10	11	3	7	14	15	2	6]
λ_{el}	[(3,	11)	(2,	10)	(7,	15)	(6,	14)	(0,	8)	(4,	12)	(1,	9)	(5,	13)]
λ_{ol}	[(5,	13)	(4,	12)	(1,	9)	(0,	8)	(3,	11)	(7,	15)	(2,	10)	(6,	14)]

Table 4.2: Proposed λ_{er}, λ_{or}, λ_{el}, and λ_{ol} for 32-QAM.

λ_{er}	[24	17	13	8	25	29	1	9	21	28	0	5	16	20	4	12,
	30	22	6	2	23	18	14	7	26	19	15	11	27	31	3	10]
λ_{or}	[7	2	30	22	6	14	18	23	10	15	19	26	3	11	27	31,
	13	9	25	24	8	1	29	21	4	0	28	20	5	12	17	16]
λ_{el}	[(13	29)	(8	24)	(1	17)	(9	25)	(0	16)	(5	21)	(4	20)	(12	28),
	(6	22)	(2	18)	(14	30)	(7	23)	(15	31)	(11	27)	(3	19)	(10	26)]
λ_{ol}	[(7	23)	(2	18)	(6	22)	(14	30)	(10	26)	(15	31)	(3	19)	(11	27),
	(13	29),	(9	25)	(8	24)	(1	17)	(4	20)	(0	16)	(5	21)	(12	28)]

The remaining 16-QAM symbols belong to χ_{ol}, which are shaded in Fig. 4.2(d). Example 4.1 clarifies how to use λ_{el}, λ_{ol}, λ_{er}, and λ_{or} to construct the proposed MD of 16-QAM.

Example 4.1. In the proposed MD mapping method, let us set $m = 4$ (16-QAM), $N = 3$ and $\boldsymbol{l} = [0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1]$. $\boldsymbol{l}$ is considered as a sequence of three 4-bit labels, i.e., $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3]$, where $\boldsymbol{l}_1 = [0, 1, 1, 0]$, $\boldsymbol{l}_2 = [1, 1, 1, 1]$, and $\boldsymbol{l}_3 = [0, 1, 1, 1]$. The mapping rule in (4.3) is used to map $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3]$ to signal point $\boldsymbol{x} = [x^1, x^2, x^3]$, as follows. The Hamming weight of $\boldsymbol{l}$ is odd, i.e., $\boldsymbol{l} \in \mathcal{L}_o$, thus $x^1 = \lambda_{ol}(\boldsymbol{l}_1)$, $x^2 = \lambda_{or}(\boldsymbol{l}_2)$, and $x^3 = \lambda_{or}(\boldsymbol{l}_3)$. The decimal format of $\boldsymbol{l}_1$, $\boldsymbol{l}_2$, and $\boldsymbol{l}_3$ are 6, 15, and 7, respectively. In Fig. 4.2(d), it can be observed that among the shaded symbols that λ_{ol} operates on, symbol S_{16} is mapped by decimal label 6. As a result, $x^1 = S_{16}$. Considering the mapping function λ_{or} indicated in Fig. 4.2(c), we also have $x^2 = \lambda_{or}((15)_2) = S_{14}$ and $x^3 = \lambda_{or}((7)_2) = S_{12}$. Consequently, $\boldsymbol{l}$ is mapped to $\boldsymbol{x} = [S_{16}, S_{14}, S_{12}]$.

Table 4.3: Proposed λ_{er}, λ_{or}, λ_{el}, and λ_{ol} for 64-QAM.

λ_{er}	[63	55	61	53	48	56	50	58	62	54	46	38	35	43	51	59,
	60	52	36	37	32	33	49	57	47	39	44	45	40	41	34	42,
	21	6	4	5	0	1	3	16	14	29	12	13	8	9	24	11,
	22	30	28	20	17	25	27	19	23	31	15	7	2	10	26	18]
λ_{or}	[0	8	17	2	7	20	13	5	1	9	25	10	15	28	12	4,
	16	24	27	26	31	30	29	21	3	11	19	18	23	22	14	6,
	49	57	59	58	63	62	60	52	34	42	51	50	55	54	47	39,
	41	33	56	48	53	61	36	44	40	32	43	35	38	46	37	45]
λ_{el}	[(14	46)	(6	38)	(4	36)	(5	37)	(0	32)	(1	33)	(3	35)	(11	43),
	(30	62)	(21	53)	(12	44)	(13	45)	(8	40)	(9	41)	(16	48)	(27	59),
	(22	54)	(29	61)	(20	52)	(7	39)	(2	34)	(17	49)	(24	56)	(19	51),
	(23	55)	(31	63)	(28	60)	(15	47)	(10	42)	(25	57)	(26	58)	(18	50)]
λ_{ol}	[(0	32)	(8	40)	(17	49)	(2	34)	(7	39)	(20	52)	(13	45)	(5	37),
	(1	33)	(16	48)	(25	57)	(10	42)	(15	47)	(28	60)	(21	53)	(4	36),
	(9	41)	(24	56)	(27	59)	(26	58)	(31	63)	(22	54)	(29	61)	(12	44),
	(3	35)	(11	43)	(19	51)	(18	50)	(23	55)	(30	62)	(14	46)	(6	38)]

Table 4.4: Proposed λ_{er}, λ_{or}, λ_{el}, and λ_{ol} for 128-QAM.

λ_{er}	[100	103	37	86	105	122	32	115	116	118	36	70	104	107	106	113,
	102	110	126	111	127	96	114	98	119	101	108	124	109	82	66	99,
	117	94	78	125	64	67	112	97	87	76	79	95	80	72	83	74,
	68	71	92	77	88	65	75	90	61	84	69	93	73	89	91	120,
	60	31	13	85	24	8	81	121	14	15	12	29	25	11	16	3,
	30	5	21	28	9	27	17	19	52	23	7	20	57	56	26	1,
	39	4	22	59	43	10	49	51	38	54	6	41	58	40	42	33,
	53	46	47	63	18	2	50	48	62	55	44	45	0	123	34	35]
λ_{or}	[17	24	121	0	44	61	85	15	16	8	81	3	60	31	13	14,
	25	9	27	56	20	21	28	29	11	26	40	59	57	4	5	12,
	1	10	49	43	41	22	7	30	19	48	58	42	6	52	53	23,
	18	35	51	33	54	39	55	47	123	2	50	34	38	46	62	37,
	107	122	32	98	102	118	36	63	104	106	115	114	126	119	70	86,
	113	97	99	96	110	100	103	116	83	112	82	66	111	101	78	117,
	72	67	64	109	127	108	94	76	65	88	80	125	124	95	79	77,
	74	89	90	120	68	71	69	87	75	73	91	105	45	84	93	92]
λ_{el}	[(37	101)	(32	96)	(36	100)	(61	125)	(60	124)	(31	95)	(13	77)	(24	88),
	(8	72)	(14	78)	(15	79)	(12	76)	(29	93)	(25	89)	(11	75)	(16	80),
	(3	67)	(30	94)	(5	69)	(21	85)	(28	92)	(9	73)	(27	91)	(17	81),
	(19	83)	(52	116)	(23	87)	(7	71)	(20	84)	(57	121)	(56	120)	(26	90),
	(1	65)	(39	103)	(4	68)	(22	86)	(59	123)	(43	107)	(10	74)	(49	113),
	(51	115)	(38	102)	(54	118)	(6	70)	(41	105)	(58	122)	(40	104)	(42	106),
	(33	97)	(53	117)	(46	110)	(47	111)	(63	127)	(18	82)	(2	66)	(50	114),
	(48	112)	(62	126)	(55	119)	(44	108)	(45	109)	(0	64)	(34	98	(35	99)]
λ_{ol}	[(17	81)	(24	88)	(0	64)	(45	109)	(60	124)	(15	79)	(16	80)	(8	72),
	(3	67)	(63	127)	(31	95)	(13	77)	(14	78)	(25	89)	(11	75)	(9	73),
	(27	91)	(57	121)	(21	85)	(28	92)	(29	93)	(56	120)	(10	74)	(40	104),
	(59	123)	(20	84)	(4	68)	(7	71)	(12	76)	(26	90)	(1	65)	(49	113),
	(43	107)	(41	105)	(22	86)	(23	87)	(5	69)	(19	83)	(48	112)	(51	115),
	(58	122)	(6	70)	(39	103)	(52	116)	(30	94)	(18	82)	(50	114)	(35	99),
	(42	106)	(54	118)	(55	119)	(62	126)	(53	117)	(2	66)	(34	98)	(33	97),
	(38	102)	(46	110)	(37	101)	(47	111)	(32	96)	(36	100)	(44	108)	(61	125)]

Table 4.5: Proposed λ_{er}, λ_{or}, λ_{el}, and λ_{ol} for 256-QAM.

λ_{er}	[206	205	222	237	236	220	252	211	227	195	194	135	215	204	231	199,
	207	254	221	141	253	200	140	243	167	226	151	210	247	83	230	198,
	238	142	201	88	217	157	216	156	131	209	242	166	193	134	213	197,
	239	255	89	158	173	232	172	188	163	130	183	225	208	150	246	214,
	223	218	174	233	249	189	248	136	162	147	241	129	165	81	245	229,
	202	219	159	191	190	137	153	152	179	145	146	182	224	149	192	133,
	203	234	250	138	154	169	185	168	177	161	240	144	181	80	244	212,
	143	175	251	170	155	187	186	184	178	176	160	128	164	148	132	228,
	91	235	139	60	171	25	24	56	49	17	51	180	35	3	115	99,
	92	124	44	28	120	40	57	59	48	16	50	34	19	2	39	67,
	12	72	29	61	121	8	41	58	32	33	113	18	55	114	98	7,
	108	125	73	104	62	9	27	26	52	112	1	54	97	38	23	66,
	93	109	13	45	105	63	10	42	43	0	53	96	22	65	82	103,
	77	94	90	30	46	122	123	11	36	20	37	21	5	6	119	196,
	76	95	126	14	127	31	106	47	116	4	117	64	118	102	85	71,
	78	79	110	111	74	75	15	107	84	101	100	68	69	86	70	87]
λ_{or}	[49	48	16	32	112	0	52	36	43	42	27	26	58	59	57	56,
	17	33	1	53	21	117	20	116	47	11	123	10	63	41	40	24,
	50	113	54	96	37	64	84	4	107	106	46	122	105	62	9	25,
	51	18	97	65	22	5	101	100	15	127	31	90	30	104	121	8,
	34	55	114	38	6	118	69	68	75	74	14	13	73	45	61	120,
	19	2	23	82	119	102	86	111	110	95	126	109	125	72	29	60,
	35	3	98	7	66	103	85	70	79	77	94	108	12	124	44	28,
	180	115	39	99	67	196	71	87	78	76	93	92	91	235	139	171,
	80	148	244	228	212	229	197	198	206	207	239	203	143	234	175	170,
	164	149	132	133	246	214	231	199	205	238	223	142	202	219	251	138,
	128	224	192	245	213	230	215	204	222	237	221	255	218	159	250	155,
	181	81	150	193	134	210	247	135	220	236	254	141	174	89	191	154,
	240	165	225	208	151	83	194	195	211	200	253	201	158	233	249	187,
	160	144	129	183	166	209	226	227	252	140	217	173	88	190	137	169,
	176	161	182	241	130	242	167	131	243	216	157	232	189	153	185	186,
	178	177	179	146	145	162	147	163	156	188	172	248	136	152	168	184]
λ_{el}	[(83	211)	(88	216)	(89	217)	(81	209)	(80	208)	(91	219)	(60	188)	(25	153),
	(24	152)	(56	184)	(49	177)	(17	145)	(51	179)	(35	163)	(3	131)	(115	243),
	(99	227)	(92	220)	(124	252)	(44	172)	(28	156)	(120	248)	(40	168)	(57	185) ,
	(59	187)	(48	176)	(16	144)	(50	178)	(34	162)	(19	147)	(2	130)	(39	167),
	(67	195)	(12	140)	(72	200)	(29	157)	(61	189)	(121	249)	(8	136)	(41	169) ,
	(58	186)	(32	160)	(33	161)	(113	241)	(18	146)	(55	183)	(114	242)	(98	226),
	(7	135)	(108	236)	(125	253)	(73	201)	(104	232)	(62	190)	(9	137)	(27	155),
	(26	154)	(52	180)	(112	240)	(1	129)	(54	182)	(97	225)	(38	166)	(23	151),
	(66	194)	(93	221)	(109	237)	(13	141)	(45	173)	(105	233)	(63	191)	(10	138),
	(42	170)	(43	171)	(0	128)	(53	181)	(96	224)	(22	150)	(65	193)	(82	210),
	(103	231)	(77	205)	(94	222)	(90	218)	(30	158)	(46	174)	(122	250)	(123	251),
	(11	139)	(36	164)	(20	148)	(37	165)	(21	149)	(5	133)	(6	134)	(119	247),
	(76	204)	(95	223)	(126	254)	(14	142)	(127	255)	(31	159)	(106	234)	(47	175),
	(116	244)	(4	132)	(117	245)	(64	192)	(118	246)	(102	230)	(85	213)	(71	199),
	(78	206)	(79	207)	(110	238)	(111	239)	(74	202)	(75	203)	(15	143)	(107	235),
	(84	212)	(101	229)	(100	228)	(68	196)	(69	197)	(86	214)	(70	198)	(87	215)]
λ_{ol}	[(49	177)	(48	176)	(32	160)	(112	240)	(0	128)	(53	181)	(43	171)	(52	180),
	(11	139)	(10	138)	(42	170)	(26	154)	(27	155)	(58	186)	(59	187)	(56	184),
	(17	145)	(33	161)	(1	129)	(54	182)	(96	224)	(21	149)	(20	148)	(36	164),
	(116	244)	(122	250)	(123	251)	(63	191)	(9	137)	(41	169)	(57	185)	(16	144),
	(113	241)	(97	225)	(37	165)	(117	245)	(84	212)	(4	132)	(47	175)	(31	159),
	(46	174)	(30	158)	(105	233)	(62	190)	(8	136)	(24	152)	(50	178)	(18	146),
	(22	150)	(5	133)	(64	192)	(101	229)	(107	235)	(106	234)	(75	203)	(127	255),
	(13	141)	(45	173)	(104	232)	(121	249)	(25	153)	(51	179)	(55	183)	(38	166),
	(65	193)	(6	134)	(118	246)	(68	196)	(100	228)	(15	143)	(74	202)	(14	142),
	(90	218)	(73	201)	(61	189)	(40	168)	(34	162)	(2	130)	(114	242)	(23	151),
	(119	247)	(102	230)	(86	214)	(69	197)	(111	239)	(110	238)	(126	254)	(109	237),
	(125	253)	(72	200)	(29	157)	(120	248)	(19	147)	(115	243)	(98	226)	(82	210),
	(66	194)	(103	231)	(85	213)	(70	198)	(79	207)	(95	223)	(77	205)	(94	222),
	(12	140)	(44	172)	(28	156)	(35	163)	(3	131)	(39	167)	(99	227)	(7	135),
	(83	211)	(71	199)	(87	215)	(78	206)	(76	204)	(93	221)	(108	236)	(92	220),
	(91	219)	(124	252)	(60	188)	(80	208)	(81	209)	(67	195)	(89	217)	(88	216)]

Table 4.6: Proposed λ_{er} and λ_{or} for 512-QAM.

λ_{er}	[397	493	479	462	477	456	335	269	162	165	455	502	468	453	439	503,
	411	334	509	494	461	472	488	285	374	342	372	402	485	501	434	385,
	172	440	408	510	495	504	491	473	262	451	498	466	452	406	386	326,
	475	174	409	511	492	507	271	366	164	481	449	486	471	390	340	260,
	463	415	399	446	398	447	410	430	420	422	464	438	388	405	389	484,
	414	412	396	474	458	442	428	394	423	421	448	384	487	436	437	404,
	413	445	459	431	270	395	429	427	416	418	480	432	401	400	496	407,
	444	506	443	393	489	286	424	426	417	483	261	358	433	403	465	391,
	392	350	284	457	380	382	300	425	419	263	327	309	277	499	482	497,
	490	505	268	173	318	301	282	264	259	257	256	258	308	356	435	387,
	287	348	302	170	319	365	280	266	295	359	294	167	279	467	166	276,
	364	317	316	281	299	330	314	298	275	288	304	273	160	292	310	324,
	441	332	267	346	315	313	312	296	291	289	320	305	306	272	293	325,
	303	383	265	362	378	10	361	297	355	290	352	370	336	353	311	274,
	283	171	328	360	169	329	381	376	371	307	354	321	3	163	369	368,
	367	379	347	8	331	344	345	377	35	339	323	375	337	0	322	373,
	168	363	58	11	42	333	40	41	19	39	32	338	17	48	161	2,
	74	26	27	45	43	24	56	57	50	34	33	38	96	16	49	1,
	126	44	123	59	122	9	25	120	115	99	113	103	97	102	53	240,
	62	90	46	106	107	104	105	121	114	51	67	65	112	64	71	227,
	201	91	47	124	75	72	125	73	83	98	18	81	36	80	52	7,
	12	28	109	29	61	60	127	89	82	66	119	37	23	20	211	21,
	219	251	31	92	249	110	111	88	117	118	55	100	243	6	241	209,
	203	153	185	94	76	108	77	93	101	87	69	54	70	68	179	147,
	136	253	235	349	217	13	248	95	85	116	22	225	193	226	4	129,
	222	351	15	184	78	232	200	79	86	242	84	146	195	178	215	181,
	191	220	236	255	152	239	252	216	194	210	245	244	246	198	148	130,
	143	159	223	204	206	238	254	207	213	197	229	212	196	214	149	133,
	234	137	218	139	508	157	237	14	177	208	183	128	470	199	230	5,
	187	205	189	478	476	154	186	30	231	341	192	151	150	180	228	145,
	155	156	140	142	141	202	250	63	224	176	144	454	500	134	131	247,
	221	188	190	158	460	138	175	233	357	278	450	135	469	182	132	343]
λ_{or}	[48	37	96	33	306	64	5	308	186	60	90	441	41	123	45	233,
	225	112	113	339	161	129	224	240	250	26	10	42	40	107	61	127,
	179	103	39	32	17	241	341	69	126	74	58	347	169	59	11	235,
	343	16	0	1	65	81	80	21	63	124	122	43	345	171	106	44,
	35	49	51	99	98	34	114	50	89	25	73	9	105	120	121	57,
	163	3	115	53	83	66	2	18	72	185	88	249	91	75	104	56,
	97	67	117	119	55	116	82	54	93	13	77	232	111	8	125	109,
	19	227	131	102	226	38	118	52	153	248	95	29	137	24	108	27,
	101	100	36	195	178	87	242	6	22	217	79	201	184	76	47	349,
	243	68	85	23	84	70	194	86	216	152	200	136	15	219	187	251,
	7	211	245	4	247	20	130	146	221	205	155	253	189	92	139	110,
	193	147	229	183	230	215	210	246	157	223	236	239	255	237	12	203,
	71	231	181	199	244	198	150	182	204	207	252	141	78	28	31	351,
	176	144	135	197	213	151	134	214	159	220	238	222	254	94	202	234,
	145	128	192	228	149	196	212	148	206	143	140	191	188	218	14	46,
	310	209	342	208	133	166	132	180	156	158	142	174	154	30	138	62,
	374	450	278	438	471	406	164	470	478	479	476	477	190	172	397	472,
	262	402	466	502	390	454	468	404	462	415	412	460	413	461	408	456,
	434	386	326	340	407	500	469	452	414	350	463	399	396	492	175	392,
	279	439	276	422	486	455	405	464	398	510	494	511	508	495	411	509,
	358	167	503	436	391	453	388	484	446	430	474	444	428	445	348	332,
	387	260	418	501	437	400	448	389	286	410	447	284	493	287	475	443,
	165	467	324	485	487	423	384	420	394	458	431	334	395	268	424	303,
	419	356	435	403	277	496	421	401	270	382	490	429	459	366	507	267,
	357	481	272	327	449	432	465	318	426	442	506	380	170	491	383	425,
	293	295	160	336	263	385	480	416	316	302	319	300	427	301	296	365,
	353	288	256	359	325	497	433	261	282	266	346	317	330	315	312	297,
	305	289	352	321	257	273	320	417	298	314	378	362	299	376	379	377,
	451	292	373	371	354	309	375	372	457	173	367	281	265	344	440	271,
	499	368	307	369	322	258	311	482	285	488	393	328	329	331	280	364,
	483	304	291	323	290	338	274	294	335	504	168	505	363	360	264	489,
	275	259	337	355	370	177	162	498	409	473	269	333	361	313	381	283]

Table 4.7: Proposed λ_{el} and λ_{ol} for 512-QAM.

λ_{el}	[(172	428)	(168	424)	(162	418)	(165	421)	(175	431)	(173	429)	(164	420)	(170	426),
	(160	416)	(167	423)	(161	417)	(42	298)	(43	299)	(169	425)	(41	297)	(57	313),
	(35	291)	(33	289)	(32	288)	(1	257)	(64	320)	(177	433)	(10	266)	(58	314),
	(171	427)	(122	378)	(59	315)	(123	379)	(40	296)	(121	377)	(51	307)	(49	305),
	(163	419)	(96	352)	(65	321)	(17	273)	(241	497)	(80	336)	(90	346)	(74	330),
	(106	362)	(45	301)	(107	363)	(56	312)	(105	361)	(120	376)	(34	290	99	355),
	(97	353)	(113	369)	(37	293)	(39	295)	(103	359)	(69	325)	(124	380)	(235	491),
	(61	317)	(11	267)	(109	365)	(75	331)	(104	360)	(9	265)	(50	306)	(98	354),
	(115	371)	(19	275)	(48	304)	(112	368)	(81	337)	(7	263)	(63	319)	(44	300),
	(127	383)	(233	489)	(125	381)	(91	347)	(73	329)	(25	281)	(89	345)	(114	370),
	(3	259)	(67	323)	(225	481)	(0	256)	(16	272)	(100	356)	(139	395)	(110	366),
	(47	303)	(27	283)	(108	364)	(8	264)	(88	344)	(72	328)	(66	322)	(83	339),
	(53	309)	(227	483)	(101	357)	(243	499)	(179	435)	(71	327)	(12	268)	(92	348),
	(251	507)	(24	280)	(111	367)	(249	505)	(185	441)	(2	258)	(117	373)	(116	372),
	(119	375)	(195	451)	(131	387)	(68	324)	(211	467)	(189	445)	(219	475)	(76	332),
	(187	443)	(137	393)	(232	488)	(77	333)	(93	349)	(18	274)	(82	338)	(55	311),
	(226	482)	(102	358)	(85	341)	(4	260)	(229	485)	(255	511)	(253	509)	(15	271),
	(201	457)	(29	285)	(95	351)	(13	269)	(54	310)	(242	498)	(118	374)	(38	294),
	(87	343)	(23	279)	(36	292)	(247	503)	(230	486)	(252	508)	(239	495)	(155	411),
	(136	392)	(184	440)	(79	335)	(248	504)	(153	409)	(52	308)	(6	262)	(178	434),
	(70	326)	(84	340)	(20	276)	(244	500)	(198	454)	(238	494)	(207	463)	(141	397),
	(205	461)	(152	408)	(200	456)	(216	472)	(217	473)	(86	342)	(22	278)	(194	450),
	(130	386)	(210	466)	(215	471)	(150	406)	(212	468)	(156	412)	(159	415)	(206	462),
	(220	476)	(236	492)	(204	460)	(223	479)	(221	477)	(157	413)	(146	402)	(246	502),
	(182	438)	(134	390)	(214	470)	(148	404)	(180	436)	(203	459)	(94	350)	(154	410),
	(142	398)	(174	430)	(14	270)	(46	302)	(234	490)	(21	277)	(176	432)	(144	400),
	(208	464)	(133	389)	(149	405)	(147	403)	(193	449)	(31	287)	(78	334)	(188	444),
	(158	414)	(218	474)	(138	394)	(250	506)	(126	382)	(129	385)	(128	384)	(228	484),
	(166	422)	(197	453)	(231	487)	(28	284)	(254	510)	(222	478)	(140	396)	(190	446),
	(202	458)	(186	442)	(26	282)	(5	261)	(240	496)	(209	465)	(135	391)	(196	452),
	(213	469)	(199	455)	(245	501)	(237	493)	(191	447)	(143	399)	(30	286)	(62	318),
	(60	316)	(224	480)	(145	401)	(192	448)	(132	388)	(151	407)	(183	439)	(181	437)]
λ_{ol}	[(225	481)	(19	275)	(163	419)	(33	289)	(64	320)	(177	433)	(60	316)	(58	314),
	(42	298)	(57	313)	(40	296)	(56	312)	(125	381)	(179	435)	(48	304)	(113	369),
	(49	305)	(161	417)	(17	273)	(241	497)	(224	480)	(126	382)	(10	266)	(90	346),
	(43	299)	(41	297)	(123	379)	(45	301)	(103	359)	(112	368)	(37	293)	(97	353),
	(1	257)	(65	321)	(80	336)	(129	385)	(234	490)	(26	282)	(74	330)	(171	427),
	(169	425)	(107	363)	(11	267)	(127	383)	(69	325)	(0	256)	(39	295)	(32	288),
	(96	352)	(81	337)	(63	319)	(124	380)	(106	362)	(122	378)	(59	315)	(61	317),
	(235	491)	(44	300)	(35	291)	(34	290)	(51	307)	(98	354)	(114	370)	(66	322),
	(89	345)	(50	306)	(185	441)	(72	328)	(25	281)	(73	329)	(9	265)	(120	376),
	(121	377)	(99	355)	(3	259)	(227	483)	(53	309)	(83	339)	(55	311)	(82	338),
	(18	274)	(2	258)	(249	505)	(88	344)	(111	367)	(8	264)	(91	347)	(104	360),
	(105	361)	(115	371)	(67	323)	(117	373)	(116	372)	(38	294)	(242	498)	(118	374),
	(13	269)	(93	349)	(77	333)	(232	488)	(137	393)	(24	280)	(27	283)	(109	365),
	(75	331)	(101	357)	(119	375)	(226	482)	(102	358)	(178	434)	(52	308)	(153	409),
	(54	310)	(248	504)	(29	285)	(95	351)	(184	440)	(108	364)	(47	303)	(233	489),
	(243	499)	(131	387)	(195	451)	(23	279)	(70	326)	(87	343)	(6	262)	(194	450),
	(217	473)	(79	335)	(201	457)	(15	271)	(76	332)	(187	443)	(16	272)	(36	292),
	(85	341)	(211	467)	(84	340)	(130	386)	(146	402)	(86	342)	(22	278)	(200	456),
	(216	472)	(136	392)	(219	475)	(12	268)	(92	348)	(251	507)	(100	356)	(71	327),
	(68	324)	(4	260)	(247	503)	(20	276)	(210	466)	(157	413)	(221	477)	(152	408),
	(155	411)	(253	509)	(189	445)	(237	493)	(139	395)	(110	366)	(7	263)	(193	449),
	(229	485)	(245	501)	(244	500)	(215	471)	(134	390)	(246	502)	(205	461)	(236	492),
	(141	397)	(239	495)	(255	511)	(28	284)	(31	287)	(203	459)	(5	261)	(21	277),
	(181	437)	(199	455)	(230	486)	(198	454)	(150	406)	(182	438)	(223	479)	(220	476),
	(207	463)	(252	508)	(254	510)	(78	334)	(94	350)	(250	506)	(240	496)	(147	403),
	(231	487)	(197	453)	(183	439)	(151	407)	(212	468)	(214	470)	(204	460)	(206	462),
	(238	494)	(191	447)	(188	444)	(218	474)	(14	270)	(46	302)	(176	432)	(128	384),
	(208	464)	(149	405)	(213	469)	(196	452)	(132	388)	(148	404)	(159	415)	(156	412),
	(222	478)	(142	398)	(190	446)	(154	410)	(202	458)	(186	442)	(209	465)	(144	400),
	(228	484)	(135	391)	(133	389)	(166	422)	(164	420)	(180	436)	(143	399)	(140	396),
	(158	414)	(174	430)	(30	286)	(138	394)	(62	318)	(145	401)	(192	448)	(167	423),
	(172	428)	(175	431)	(165	421)	(173	429)	(170	426)	(160	416)	(162	418)	(168	424)]

Table 4.8: Proposed λ_{er} for 1024-QAM.

	[674	195	226	193	197	229	67	225	129	163	161	165	97	69	203	65,
	182	54	252	222	220	180	118	176	86	151	148	150	247	246	214	212,
	194	199	227	231	131	133	200	99	234	202	37	101	1	33	235	201,
	158	156	254	22	52	470	23	20	116	183	144	146	215	242	244	213,
	192	228	224	162	66	64	3	5	35	451	483	139	485	481	449	233,
	406	190	404	407	502	223	48	50	468	112	178	87	240	245	210	208,
	196	230	130	167	205	237	450	487	453	389	385	77	72	171	137	169,
	438	188	94	436	400	159	191	500	16	255	114	179	181	84	243	694,
	198	128	98	68	232	39	448	141	138	419	355	421	74	417	75	73,
	126	30	124	432	434	402	92	184	496	18	248	119	82	147	149	662,
	134	135	103	96	239	0	136	173	104	109	13	106	323	353	107	105,
	62	278	28	342	374	127	95	186	471	498	503	55	221	80	145	209,
	132	160	71	2	455	482	170	423	359	322	45	10	11	325	357	321,
	60	310	304	476	31	368	370	439	403	435	152	464	253	19	115	211,
	166	164	7	32	480	387	168	324	320	8	456	257	461	491	9	41,
	412	510	306	308	274	276	372	343	336	340	189	157	51	250	177	241,
	70	100	4	34	452	418	327	356	495	490	42	261	493	293	43	457,
	446	508	478	272	279	307	63	56	375	371	120	154	466	218	216	83,
	102	204	207	484	386	76	354	260	40	291	488	458	289	459	393	489,
	414	444	447	408	479	511	275	472	58	338	122	405	499	469	21	117,
	238	36	143	175	384	79	111	47	259	295	397	429	427	363	361	425,
	284	286	415	440	442	410	311	504	506	24	125	401	155	467	53	85,
	6	486	454	420	12	15	352	292	258	392	426	394	395	301	331	329,
	318	380	314	348	445	413	509	305	61	26	433	437	187	501	49	113,
	236	140	391	416	78	326	263	256	463	431	424	365	269	328	265	297,
	316	280	312	376	443	411	309	477	474	273	29	93	90	88	251	17,
	38	206	388	108	358	294	460	290	288	399	268	367	271	360	299	267,
	287	350	383	315	282	378	441	475	277	507	27	339	123	497	219	81,
	673	142	172	422	14	44	494	492	428	270	364	332	303	264	266	330,
	382	319	317	283	313	379	346	409	59	373	341	369	185	153	249	217,
	581	174	390	110	46	262	396	398	430	462	302	300	333	362	298	296,
λ_{er}	351	285	381	281	347	344	377	473	505	25	337	121	91	465	702	566,
	609	745	961	681	585	617	833	553	521	969	335	873	366	777	334	809,
	829	831	349	828	830	345	798	924	57	572	574	542	89	638	918	534,
	577	713	993	649	837	869	865	555	805	1001	939	843	937	875	813	811,
	825	827	792	826	796	892	862	956	958	926	822	790	636	700	670	764,
	677	747	997	619	587	835	523	971	1003	905	845	841	781	815	779	810,
	793	797	824	799	794	894	952	927	816	818	1022	540	944	950	668	766,
	579	513	545	683	933	525	522	773	1005	941	907	877	842	776	778	808,
	893	891	795	895	888	860	959	954	1020	820	990	543	606	916	1014	732,
	741	715	651	586	929	621	557	968	973	769	801	840	879	783	812	780,
	953	859	856	863	957	890	920	784	788	786	791	854	912	948	564	734,
	611	613	549	584	897	589	970	554	807	1000	909	906	872	874	876	814,
	921	889	858	955	923	922	823	991	575	988	882	886	919	671	560	688,
	641	547	963	995	901	871	867	520	1007	803	943	938	904	936	847	782,
	861	1021	821	989	925	984	1016	819	787	880	639	914	946	735	982	630,
	705	714	965	650	931	618	834	1002	552	771	975	844	768	911	940	878,
	857	987	789	817	573	986	1018	570	568	884	604	1008	703	980	532	656,
	737	578	517	999	685	616	836	832	866	559	772	770	775	1004	908	846,
	910	1017	1019	785	536	1023	887	852	855	915	607	983	1012	562	535	660,
	675	746	515	962	749	653	935	899	839	868	804	972	802	800	974	942,
	985	569	539	538	541	883	850	848	947	951	696	1010	767	624	628	663,
	709	643	712	717	744	682	960	996	623	591	864	527	838	1006	556	774,
	885	537	571	851	945	637	634	632	669	664	698	528	567	631	598	690,
	739	707	645	551	514	994	648	680	687	898	588	870	524	806	526	558,
	849	853	881	605	913	600	917	976	701	733	765	530	596	599	658	692,
	679	704	610	576	615	751	967	512	930	964	655	896	903	590	620	633,
	601	603	635	602	949	667	666	978	760	1015	531	762	626	752	727	695,
	706	642	740	580	583	546	612	516	992	719	966	932	928	900	902	622,
	977	697	665	1009	699	979	981	728	730	563	592	659	720	722	754	759,
	738	736	743	676	708	608	644	519	544	548	716	998	934	652	654	684,
	761	729	1013	731	529	1011	565	533	627	594	657	661	691	724	756	726,
	678	672	742	710	711	646	640	647	582	614	550	518	718	748	686	750,
	593	625	763	561	597	629	595	721	689	753	723	725	755	757	693	758]

Table 4.9: Proposed λ_{or} for 1024-QAM.

λ_{or}	[829	825	827	797	793	893	891	889	856	859	921	861	987	857	1017	985,
	910	846	942	908	940	878	782	876	814	783	812	778	808	810	811	809,
	831	792	824	795	953	957	955	858	1021	821	789	1019	571	569	537	849,
	774	556	1006	974	802	911	1004	844	847	936	874	780	776	815	779	334,
	349	826	799	895	863	923	817	573	989	541	785	539	885	881	853	633,
	558	526	806	524	972	800	775	772	768	904	938	872	879	781	875	813,
	828	796	794	890	925	1016	984	986	538	536	945	851	605	635	603	697,
	601	590	620	870	838	527	864	804	770	975	771	943	840	842	843	777,
	345	888	860	922	823	1023	570	887	883	637	913	600	602	1009	665	902,
	622	900	928	903	896	588	623	866	868	559	554	803	906	941	939	366,
	830	894	920	991	819	1018	848	850	632	917	949	667	699	731	1013	977,
	654	684	934	932	655	898	591	899	839	832	552	807	1000	909	907	937,
	892	959	954	575	787	568	855	634	947	976	666	979	1011	561	729	761,
	686	652	966	719	964	687	930	680	935	836	1002	1007	970	801	845	877,
	798	784	788	880	884	882	852	701	669	978	1015	728	981	565	529	593,
	718	998	548	544	992	996	648	960	653	931	616	834	520	968	769	841,
	927	1020	791	543	639	607	915	951	664	765	733	730	533	595	763	625,
	748	518	716	516	751	967	512	682	685	901	618	871	867	557	1005	873,
	952	820	786	604	914	696	1008	1010	980	760	531	762	563	627	629	597,
	750	519	612	615	546	994	514	962	749	650	897	525	522	773	973	905,
	862	818	988	886	912	703	698	528	530	567	596	659	592	594	657	721,
	550	614	582	608	583	551	744	717	999	965	584	589	621	835	971	805,
	816	1022	990	854	946	671	983	1012	767	631	626	599	661	720	753	689,
	644	647	640	676	580	610	712	515	517	549	995	586	929	865	523	1003,
	924	822	540	636	919	948	735	560	532	624	752	658	691	722	725	723,
	646	710	711	708	645	576	746	714	547	513	963	651	933	837	555	1001,
	956	572	542	944	916	564	982	732	562	628	598	663	727	724	755	693,
	742	743	740	642	707	578	705	611	613	997	545	683	587	619	521	969,
	958	926	638	606	950	668	1014	764	734	535	656	660	754	756	759	757,
	736	672	679	704	709	643	641	579	747	715	993	961	649	585	869	553,
	57	790	89	700	918	702	670	566	766	534	630	690	692	695	758	726,
	678	738	706	739	675	737	677	673	581	609	713	745	681	617	833	398,
	25	574	121	91	465	249	219	217	81	113	688	241	662	694	208	213,
	194	674	196	230	198	166	741	6	38	577	142	390	172	110	262	430,
	505	337	369	185	153	497	49	251	17	85	117	83	209	210	242	212,
	195	192	199	228	134	132	70	102	236	238	140	174	388	14	46	396,
	473	341	93	123	90	467	501	53	218	115	177	145	211	245	247	214,
	227	226	231	224	128	135	164	7	204	486	206	454	422	108	44	428,
	373	27	339	437	187	88	469	21	19	216	80	149	243	240	215	244,
	193	197	131	162	130	98	160	4	207	36	143	391	420	78	294	492,
	59	507	433	371	401	155	499	51	250	82	181	147	146	144	148	246,
	225	229	133	64	167	68	71	100	32	452	175	384	416	358	494	462,
	409	277	29	125	122	405	466	253	248	221	179	84	183	178	176	151,
	129	67	99	163	66	103	96	239	34	484	386	418	12	326	460	288,
	475	305	273	26	120	154	157	464	55	119	114	87	116	118	86	150,
	182	161	165	200	205	232	2	455	0	480	76	111	15	292	290	270,
	377	477	474	24	338	435	189	498	503	16	255	468	112	220	22	180,
	65	97	234	202	5	237	39	448	482	168	79	354	352	263	463	302,
	346	309	61	506	375	340	186	152	471	18	50	23	20	52	222	54,
	69	1	101	37	35	3	487	136	170	423	356	260	495	256	399	335,
	379	441	509	504	58	336	439	403	184	496	500	223	48	156	254	252,
	235	203	33	483	453	450	138	141	387	327	320	40	259	258	431	364,
	313	378	411	472	307	56	372	127	92	402	191	159	470	404	158	406,
	201	449	485	451	385	389	173	355	104	324	47	291	295	392	332	300,
	347	344	443	511	311	343	276	95	370	434	400	407	436	502	188	190,
	233	481	139	171	421	109	419	106	359	322	490	488	426	367	268	303,
	281	283	445	413	442	410	275	63	274	368	31	28	94	432	438	73,
	169	137	75	72	417	77	13	10	8	456	42	397	424	365	271	333,
	381	315	282	376	415	479	440	408	279	476	374	342	310	124	30	126,
	105	107	323	353	74	45	257	261	461	289	429	394	269	360	264	362,
	285	317	383	312	314	348	380	447	272	508	308	304	306	60	278	62,
	321	357	9	325	11	491	293	493	458	427	395	328	301	299	266	298,
	351	382	319	287	350	280	318	316	284	286	478	444	510	414	446	412,
	457	41	489	43	459	425	393	361	363	331	265	329	297	267	330	296]

Table 4.10: Proposed λ_{el} for 1024-QAM.

	[(195	707))	((226	738)	(193	705)	(197	709)	(229	741)	(67	579)	(225	737)	(129	641),
	(163	675)	(161	673)	(165	677)	(97	609)	(69	581)	(203	715)	(65	577)	(182	694),
	(54	566)	(252	764)	(222	734)	(220	732)	(180	692)	(118	630)	(176	688)	(86	598),
	(151	663)	(148	660)	(150	662)	(247	759)	(246	758)	(214	726)	(212	724)	(194	706),
	(199	711)	(227	739)	(231	743)	(131	643)	(133	645)	(200	712)	(99	611)	(234	746),
	(202	714)	(37	549)	(101	613)	(1	513)	(33	545)	(235	747)	(201	713)	(158	670),
	(156	668)	(254	766)	(22	534)	(52	564)	(470	982)	(23	535)	(20	532)	(116	628),
	(183	695)	(144	656)	(146	658)	(215	727)	(242	754)	(244	756)	(213	725)	(192	704),
	(228	740)	(224	736)	(162	674)	(66	578)	(64	576)	(3	515)	(5	517)	(35	547),
	(451	963)	(483	995)	(139	651)	(485	997)	(481	993)	(449	961)	(233	745)	(406	918),
	(190	702)	(404	916)	(407	919)	(502	1014)	(223	735)	(48	560)	(50	562)	(468	980),
	(112	624)	(178	690)	(87	599)	(240	752)	(245	757)	(210	722)	(208	720)	(196	708),
	(230	742)	(130	642)	(167	679)	(205	717)	(237	749)	(450	962)	(487	999)	(453	965),
	(389	901)	(385	897)	(77	589)	(72	584)	(171	683)	(137	649)	(169	681)	(438	950),
	(188	700)	(94	606)	(436	948)	(400	912)	(159	671)	(191	703)	(500	1012)	(16	528),
	(255	767)	(114	626)	(179	691)	(181	693)	(84	596)	(243	755)	(198	710)	(128	640),
	(98	610)	(68	580)	(232	744)	(39	551)	(448	960)	(141	653)	(138	650)	(419	931),
	(355	867)	(421	933)	(74	586)	(417	929)	(75	587)	(73	585)	(126	638)	(30	542),
	(124	636)	(432	944)	(434	946)	(402	914)	(92	604)	(184	696)	(496	1008)	(18	530),
	(248	760)	(119	631)	(82	594)	(147	659)	(149	661)	(134	646)	(135	647)	(103	615),
	(96	608)	(239	751)	(0	512)	(136	648)	(173	685)	(104	616)	(109	621)	(13	525),
	(106	618)	(323	835)	(353	865)	(107	619)	(105	617)	(62	574)	(278	790)	(28	540),
	(342	854)	(374	886)	(127	639)	(95	607)	(186	698)	(471	983)	(498	1010)	(503	1015),
	(55	567)	(221	733)	(80	592)	(145	657)	(209	721)	(132	644)	(160	672)	(71	583),
	(2	514)	(455	967)	(482	994)	(170	682)	(423	935)	(359	871)	(322	834)	(45	557),
	(10	522)	(11	523)	(325	837)	(357	869)	(321	833)	(60	572)	(310	822)	(304	816),
	(476	988)	(31	543)	(368	880)	(370	882)	(439	951)	(403	915)	(435	947)	(152	664),
	(464	976)	(253	765)	(19	531)	(115	627)	(211	723)	(166	678)	(164	676)	(7	519),
	(32	544)	(480	992)	(387	899)	(168	680)	(324	836)	(320	832)	(8	520)	(456	968),
	(257	769)	(461	973)	(491	1003)	(9	521)	(41	553)	(412	924)	(510	1022)	(306	818),
	(308	820)	(274	786)	(276	788)	(372	884)	(343	855)	(336	848)	(340	852)	(189	701),
λ_{el}	(157	669)	(51	563)	(250	762)	(177	689)	(241	753)	(70	582)	(100	612)	(4	516),
	(34	546)	(452	964)	(418	930)	(327	839)	(356	868)	(495	1007)	(490	1002)	(42	554),
	(261	773)	(493	1005)	(293	805)	(43	555)	(457	969)	(446	958)	(508	1020)	(478	990),
	(272	784)	(279	791)	(307	819)	(63	575)	(56	568)	(375	887)	(371	883)	(120	632),
	(154	666)	(466	978)	(218	730)	(216	728)	(83	595)	(102	614)	(204	716)	(207	719),
	(484	996)	(386	898)	(76	588)	(354	866)	(260	772)	(40	552)	(291	803)	(488	1000),
	(458	970)	(289	801)	(459	971)	(393	905)	(489	1001)	(414	926)	(444	956)	(447	959),
	(408	920)	(479	991)	(511	1023)	(275	787)	(472	984)	(58	570)	(338	850)	(122	634),
	(405	917)	(499	1011)	(469	981)	(21	533)	(117	629)	(238	750)	(36	548)	(143	655),
	(175	687)	(384	896)	(79	591)	(111	623)	(47	559)	(259	771)	(295	807)	(397	909),
	(429	941)	(427	939)	(363	875)	(361	873)	(425	937)	(284	796)	(286	798)	(415	927),
	(440	952)	(442	954)	(410	922)	(311	823)	(504	1016)	(506	1018)	(24	536)	(125	637),
	(401	913)	(155	667)	(467	979)	(53	565)	(85	597)	(6	518)	(486	998)	(454	966),
	(420	932)	(12	524)	(15	527)	(352	864)	(292	804)	(258	770)	(392	904)	(426	938),
	(394	906)	(395	907)	(301	813)	(331	843)	(329	841)	(318	830)	(380	892)	(314	826),
	(348	860)	(445	957)	(413	925)	(509	1021)	(305	817)	(61	573)	(26	538)	(433	945),
	(437	949)	(187	699)	(501	1013)	(49	561)	(113	625)	(236	748)	(140	652)	(391	903),
	(416	928)	(78	590)	(326	838)	(263	775)	(256	768)	(463	975)	(431	943)	(424	936),
	(365	877)	(269	781)	(328	840)	(265	777)	(297	809)	(316	828)	(280	792)	(312	824),
	(376	888)	(443	955)	(411	923)	(309	821)	(477	989)	(474	986)	(273	785)	(29	541),
	(93	605)	(90	602)	(88	600)	(251	763)	(17	529)	(38	550)	(206	718)	(388	900),
	(108	620)	(358	870)	(294	806)	(460	972)	(290	802)	(288	800)	(399	911)	(268	780),
	(367	879)	(271	783)	(360	872)	(299	811)	(267	779)	(287	799)	(350	862)	(383	895),
	(315	827)	(282	794)	(378	890)	(441	953)	(475	987)	(277	789)	(507	1019)	(27	539),
	(339	851)	(123	635)	(497	1009)	(219	731)	(81	593)	(142	654)	(172	684)	(422	934),
	(14	526)	(44	556)	(494	1006)	(492	1004)	(428	940)	(270	782)	(364	876)	(332	844),
	(303	815)	(264	776)	(266	778)	(330	842)	(382	894)	(319	831)	(317	829)	(283	795),
	(313	825)	(379	891)	(346	858)	(409	921)	(59	571)	(373	885)	(341	853)	(369	881),
	(185	697)	(153	665)	(249	761)	(217	729)	(174	686)	(390	902)	(110	622)	(46	558),
	(262	774)	(396	908)	(398	910)	(430	942)	(462	974)	(302	814)	(300	812)	(333	845),
	(362	874)	(298	810)	(296	808)	(351	863)	(285	797)	(381	893)	(281	793)	(347	859),
	(344	856)	(377	889)	(473	985)	(505	1017)	(25	537)	(337	849)	(121	633)	(91	603),
	(465	977)	(335	847)	(366	878)	(334	846)	(349	861)	(345	857)	(57	569)	(89	601)]

Table 4.11: Proposed λ_{ol} for 1024-QAM.

λ_{ol}	[(349 861)	(334 846)	(366 878)	(345 857)	(473 985)	(398 910)	(335 847)	(505 1017),
	(57 569)	(121 633)	(91 603)	(89 601)	(465 977)	(249 761)	(217 729)	(81 593),
	(113 625)	(209 721)	(208 720)	(212 724)	(195 707)	(196 708)	(198 710)	(134 646),
	(102 614)	(38 550)	(142 654)	(390 902)	(422 934)	(46 558)	(396 908)	(430 942),
	(377 889)	(25 537)	(337 849)	(185 697)	(153 665)	(497 1009)	(219 731)	(49 561),
	(17 529)	(85 597)	(83 595)	(211 723)	(243 755)	(210 722)	(213 725)	(214 726),
	(194 706)	(199 711)	(230 742)	(135 647)	(132 644)	(166 678)	(70 582)	(6 518),
	(236 748)	(140 652)	(174 686)	(110 622)	(14 526)	(262 774)	(294 806)	(302 814),
	(409 921)	(341 853)	(369 881)	(123 635)	(88 600)	(501 1013)	(53 565)	(251 763),
	(117 629)	(177 689)	(145 657)	(241 753)	(245 757)	(247 759)	(242 754)	(246 758),
	(226 738)	(192 704)	(228 740)	(128 640)	(160 672)	(71 583)	(164 676)	(204 716),
	(238 750)	(206 718)	(388 900)	(172 684)	(108 620)	(44 556)	(494 1006)	(428 940),
	(346 858)	(373 885)	(339 851)	(93 605)	(187 699)	(499 1011)	(467 979)	(218 730),
	(216 728)	(115 627)	(181 693)	(149 661)	(240 752)	(215 727)	(244 756)	(193 705),
	(227 739)	(231 743)	(224 736)	(130 642)	(162 674)	(68 580)	(100 612)	(7 519),
	(36 548)	(486 998)	(454 966)	(420 932)	(78 590)	(460 972)	(492 1004)	(462 974),
	(475 987)	(59 571)	(27 539)	(433 945)	(90 602)	(155 667)	(51 563)	(21 533),
	(19 531)	(82 594)	(80 592)	(84 596)	(147 659)	(146 658)	(148 660)	(150 662),
	(225 737)	(197 709)	(229 741)	(131 643)	(167 679)	(103 615)	(4 516)	(32 544),
	(207 719)	(484 996)	(391 903)	(416 928)	(12 524)	(326 838)	(290 802)	(270 782),
	(347 859)	(507 1019)	(29 541)	(437 949)	(401 913)	(469 981)	(466 978)	(253 765),
	(250 762)	(221 733)	(179 691)	(114 626)	(144 656)	(151 663)	(86 598)	(118 630),
	(67 579)	(129 641)	(133 645)	(66 578)	(98 610)	(96 608)	(239 751)	(34 546),
	(455 967)	(143 655)	(175 687)	(384 896)	(358 870)	(292 804)	(288 800)	(268 780),
	(379 891)	(277 789)	(273 785)	(125 637)	(122 634)	(405 917)	(157 669)	(464 976),
	(55 567)	(119 631)	(112 624)	(87 599)	(183 695)	(176 688)	(180 692)	(182 694),
	(161 673)	(163 675)	(99 611)	(64 576)	(3 515)	(205 717)	(232 744)	(2 514),
	(452 964)	(386 898)	(418 930)	(76 588)	(15 527)	(263 775)	(258 770)	(399 911),
	(344 856)	(309 821)	(61 573)	(26 538)	(371 883)	(120 632)	(154 666)	(503 1015),
	(255 767)	(248 760)	(16 528)	(468 980)	(178 690)	(116 628)	(220 732)	(22 534),
	(65 577)	(165 677)	(200 712)	(234 746)	(5 517)	(237 749)	(39 551)	(0 512),
	(480 992)	(168 680)	(327 839)	(79 591)	(354 866)	(256 768)	(463 975)	(364 876),
	(411 923)	(441 953)	(305 817)	(506 1018)	(24 536)	(338 850)	(189 701)	(152 664),
	(496 1008)	(498 1010)	(18 530)	(23 535)	(20 532)	(52 564)	(222 734)	(203 715),
	(97 609)	(69 581)	(101 613)	(202 714)	(487 999)	(450 962)	(448 960)	(482 994),
	(170 682)	(423 935)	(111 623)	(260 772)	(352 864)	(47 559)	(431 943)	(300 812),
	(313 825)	(443 955)	(477 989)	(474 986)	(58 570)	(375 887)	(340 852)	(435 947),
	(471 983)	(184 696)	(50 562)	(223 735)	(502 1014)	(254 766)	(252 764)	(54 566),
	(201 713)	(1 513)	(37 549)	(483 995)	(35 547)	(138 650)	(141 653)	(136 648),
	(387 899)	(324 836)	(356 868)	(40 552)	(259 771)	(392 904)	(424 936)	(332 844),
	(283 795)	(376 888)	(410 922)	(504 1016)	(472 984)	(56 568)	(336 848)	(403 915),
	(439 951)	(186 698)	(500 1012)	(191 703)	(48 560)	(470 982)	(158 670)	(156 668),
	(235 747)	(33 545)	(485 997)	(451 963)	(453 965)	(389 901)	(173 685)	(104 616),
	(359 871)	(320 832)	(495 1007)	(490 1002)	(291 803)	(426 938)	(367 879)	(271 783),
	(381 893)	(378 890)	(413 925)	(509 1021)	(275 787)	(343 855)	(372 884)	(370 882),
	(127 639)	(92 604)	(402 914)	(159 671)	(407 919)	(400 912)	(404 916)	(233 745),
	(481 993)	(406 918)	(190 702)	(139 651)	(385 897)	(419 931)	(109 621)	(355 867),
	(322 834)	(8 520)	(456 968)	(295 807)	(488 1000)	(365 877)	(269 781)	(333 845),
	(317 829)	(281 793)	(282 794)	(445 957)	(311 823)	(307 819)	(63 575)	(276 788),
	(95 607)	(368 880)	(432 944)	(434 946)	(94 606)	(124 636)	(436 948)	(449 961),
	(137 649)	(438 950)	(171 683)	(417 929)	(72 584)	(421 933)	(106 618)	(13 525),
	(45 557)	(261 773)	(42 554)	(397 909)	(394 906)	(301 813)	(303 815)	(264 776),
	(383 895)	(315 827)	(314 826)	(348 860)	(442 954)	(408 920)	(511 1023)	(274 786),
	(308 820)	(31 543)	(374 886)	(342 854)	(28 540)	(30 542)	(73 585)	(188 700),
	(169 681)	(75 587)	(107 619)	(74 586)	(323 835)	(77 589)	(10 522)	(257 769),
	(493 1005)	(461 973)	(429 941)	(427 939)	(395 907)	(265 777)	(360 872)	(298 810),
	(285 797)	(319 831)	(280 792)	(312 824)	(286 798)	(440 952)	(479 991)	(279 791),
	(272 784)	(304 816)	(476 988)	(306 818)	(310 822)	(60 572)	(105 617)	(62 574),
	(278 790)	(126 638)	(353 865)	(357 869)	(11 523)	(325 837)	(491 1003)	(293 805),
	(289 801)	(458 970)	(393 905)	(331 843)	(328 840)	(299 811)	(362 874)	(296 808),
	(351 863)	(382 894)	(287 799)	(350 862)	(380 892)	(318 830)	(284 796)	(415 927),
	(447 959)	(478 990)	(444 956)	(508 1020)	(457 969)	(510 1022)	(41 553)	(9 521),
	(321 833)	(412 924)	(489 1001)	(446 958)	(414 926)	(43 555)	(425 937)	(361 873),
	(459 971)	(363 875)	(329 841)	(297 809)	(316 828)	(267 779)	(266 778)	(330 842)]

4.3.2 Performance comparison

The proposed mappings are compared to the MD mappings that are optimized for Rayleigh fading channels using the BSA. Typically, the BSA is the best known computer search algorithm to find good mappings for BICM-ID. However, it becomes intractable to obtain MD mappings with a higher alphabet size, e.g., 6-D 64-QAM. In our simulations, we consider a rate-1/2 convolutional code with the generator polynomial of $(13, 15)_8$. The length of the used interleaver is about 10000 bits. All BER curves are presented with seven iterations, and all gains reported in this section are measured at a BER of 10^{-6}.

Table 4.12 indicates the values of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for different $2N$-D ($N = 2, 3$) 2^m-QAM mappings. Note that in the case of 6-D ($N = 3$) mappings using 2^m-QAM ($m > 4$), the BSA result could not be obtained due to the computational time constraints. However, our proposed method yields efficient $2N$-D mapping of 2^m-QAM for any value of N and for m ($m \leq 10$). Table 4.12 shows that for all considered values of N and m, the resulting mappings with our proposed method offer greater values of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ in comparison with those of the BSA mappings. Moreover, for a given code and modulation, the mapping with a greater value of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ achieves a lower error floor [34]. As a consequence, the proposed mappings are expected to offer lower error-floors than those of the BSA mappings. The value of $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for the proposed 4-D mapping using higher order 2^m-QAM ($m = 7, \cdots, 10$) is listed in Table 4.13.

Table 4.12: Comparison of the harmonic mean of MSED, $\hat{\Phi}(\mu, \boldsymbol{\chi})$, for different mappings.

Mapping	$N = 2$	$N = 3$
BSA MD 16-QAM	2.5814	2.8047
Proposed MD 16-QAM	3.1622	3.3105
BSA MD 32-QAM	2.8574	-
Proposed MD 32-QAM	3.1677	3.3089
BSA MD 64-QAM	2.8047	-
Proposed MD 64-QAM	3.1683	3.2870

Table 4.13: $\hat{\Phi}(\mu, \boldsymbol{\chi})$ for proposed 4-D mapping using higher order QAMs.

Basic modulation	$\hat{\Phi}(\mu, \boldsymbol{\chi})$
128-QAM	3.2272
256-QAM	3.2289
512-QAM	3.2833
1024-QAM	3.2995

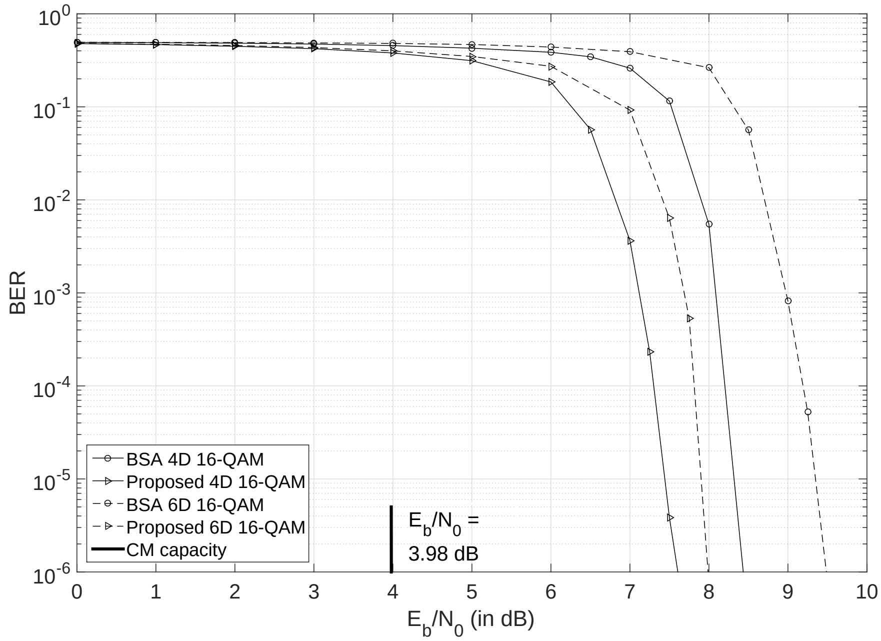

Figure 4.3: BER performance of BICM-ID with 4-D and 6-D 16-QAM over Rayleigh fading channels.

Fig. 4.3 plots the BER performance of BICM-ID employing different $2N$-D mappings of 16-QAM for $N = 2, 3$. From Fig. 4.3, it can be observed that our 4-D and 6-D 16-QAM mappings respectively achieve 0.85 dB and 1.5 dB gain over their BSA counterparts. The BER performance of BICM-ID with 4-D 32-QAM and 4-D 64-QAM is plotted in Fig. 4.4. It can be seen from this figure that compared to the BSA mappings, the proposed mappings significantly improve the BER performance of BICM-ID using 4-D 32-QAM and 4-D 64-QAM. In particular, our resulting mapping for 4-D 32-QAM offers a gain of 1.4dB while for 4-D 64-QAM it offers a gain of 2.6 dB compared with the BSA mappings.

In Fig. 4.5, we compare the error bounds for BICM-ID presented in [15] using MD 16-QAM mappings. This figure shows that our proposed mappings provide a gain of 0.9 dB and 0.73 dB respectively over the BSA results for 4-D and 6-D 16-QAM. Fig. 4.6 plots the error bounds for BICM-ID with 4-D 32-QAM and 4-D 64-QAM. This figure shows that in comparison with the BSA mappings, our proposed mappings improve the error bound respectively by 0.5 dB and 0.53 dB.

It is important to note that the BSA could not be applied to find a mapping for 6-D 32-QAM due to the computational time constraint. However, our proposed method easily finds suitable mappings for an unlimited dimension of modulations larger than 32-QAM. Moreover, for smaller MD modulations, our proposed mappings outperform the BSA mappings. This indicates the efficiency of our proposed method compared to the BSA.

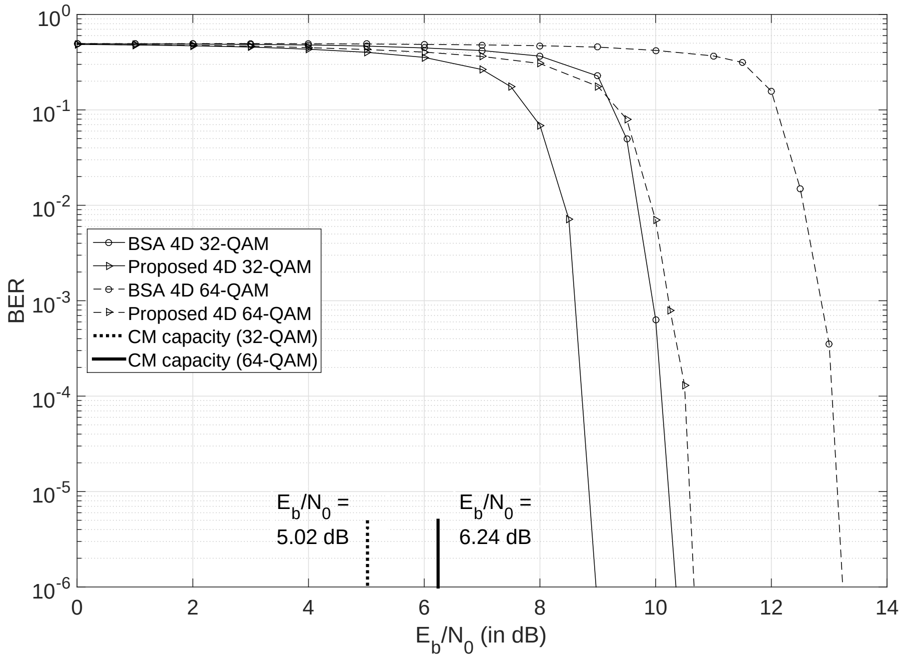

Figure 4.4: BER performance of BICM-ID with 4-D 32- and 64-QAM over Rayleigh fading channels.

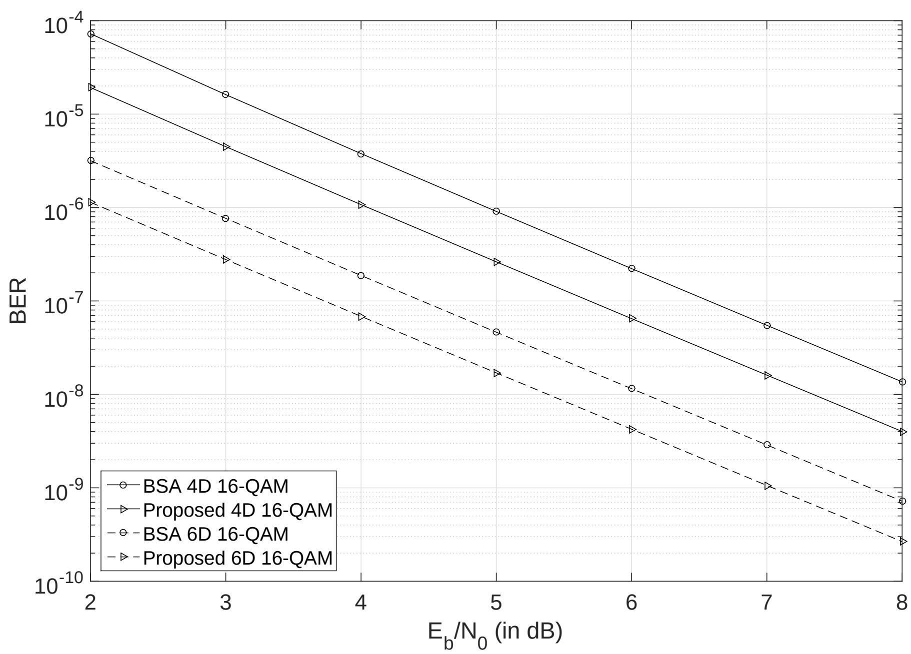

Figure 4.5: Error-floor bounds of BER for BICM-ID with 4-D and 6-D 16-QAM over Rayleigh fading channels.

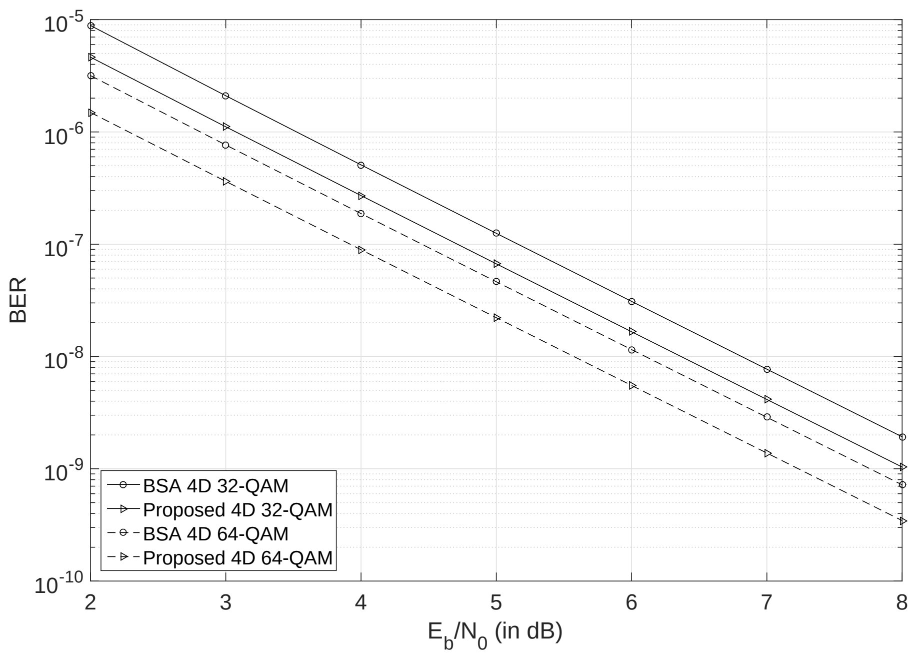

Figure 4.6: Error-floor bounds of BER for BICM-ID with 4-D 32- and 64-QAM over Rayleigh fading channels.

Chapter 5

Multi-dimensional Mapping of M-QAM Constellations for BICM-ID over AWGN Channels

The AWGN channel is an effective model for many communication links such as satellite communication link. To improve the error performance and the data rate of BICM-ID over AWGN channels, efficient mappings of large constellations are required. However, as mentioned earlier, developing optimum mappings of large constellations for BICM-ID is complicated. This is because of the significantly large number of possible mappings for large constellations. In fact, for a constellation with order 2^m, there are $2^m!$ possible mappings, which quickly approaches infinity by increasing m. In this chapter, we first propose an optimum MD mapping for 16-QAM. Then, employing an innovative transfer system, we construct highly efficient MD mappings for any order/dimension of QAM constellations using the proposed MD 16-QAM mapping. Throughout this chapter, we assume that in a 2^m-ary QAM constellation: (i) the MSED between symbols is equal to 1 and (ii) the symbol index in 2^m-QAM starts from the top left corner of the constellation and increases from top to bottom and from left to right (see for example Fig. 5.1 for 16-QAM, where the symbol with index i is indicated by S_i).

5.1 Optimum MD 16-QAM mapping

In a $2N$-D 16-QAM mapping, a sequence of n binary bits is mapped to a vector of N 16-QAM symbols, where $n = 4N$. The proposed method uses the precoding process given in 5.1, followed by an intermediate mapping. Let $\boldsymbol{l} = [l^1, \cdots, l^n]$ be an n-bit label and $\hat{\boldsymbol{l}} = [\hat{l}^1, \cdots, \hat{l}^n]$ be the precoded version of $\boldsymbol{l}$. Each element of $\hat{\boldsymbol{l}}$ is obtained according to the following precoding process:

$$\hat{l}^i = \begin{cases} W(\boldsymbol{l}) & \text{if } i = \text{chosen-index}, \\ l^i \oplus W(\boldsymbol{l}) & \text{otherwise}, \end{cases} \tag{5.1}$$

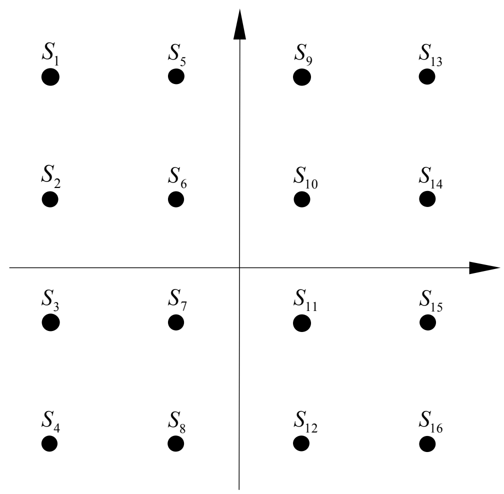

Figure 5.1: Symbol arrangement in a 16-QAM constellation.

where $W(\boldsymbol{x})$ denotes an indicator function that takes the value one if $\boldsymbol{x}$ has an odd Hamming weight, otherwise it is equal to zero. The chosen-index can also take a value from $\{1, 2, \cdots, n\}$, and $\oplus$ is the modulo-2 addition. The proposed MD 16-QAM mapping method assigns the label $\boldsymbol{l}$ to the symbol-vector $\boldsymbol{s} = [s^1, \cdots, s^N]$ as follows

$$s^j = \lambda(\hat{\boldsymbol{l}}^j), \tag{5.2}$$

where

$$\hat{\boldsymbol{l}}^j = [\hat{l}^{4(j-1)+1}, \cdots, \hat{l}^{4j}], \tag{5.3}$$

and j takes a value from $\{1, \cdots N\}$, and λ is the intermediate 16-QAM mapping function (which will be discussed later in this section).

Let d_i^2 indicate the i^{th} unique squared Euclidean distance (SED) between the symbols in a 16-QAM constellation, where $d_i^2 < d_{i+1}^2$. As mentioned in 1.6, an optimum mapping offers the maximum possible value of $\hat{d}_{min}^2$.

Proposition 5.1. *For a 2N-D 16-QAM mapping, the maximum $\hat{d}_{min}^2$ is $(N-1)d_5^2 + d_4^2$.*

Proof. If s is a central-symbol in a 16-QAM constellation, i. e., $s \in \Upsilon = \{S_6, S_7, S_{10}, S_{11}\}$, there is only one 16-QAM symbol at the SED of d_5^2 from s, and each of the remaining symbols has a smaller SED from s. Assume that $\boldsymbol{l} = [l^1, \cdots, l^n]$ is the assigned label to the symbol-vector $\boldsymbol{s} = [s^1, \cdots, s^N]$, where $\forall i; s^i \in \Upsilon$. There are n distinct labels that are different from $\boldsymbol{l}$ in only one bit position. Moreover, regarding the above discussion, there is only one symbol-vector with SED of Nd_5^2 from $\boldsymbol{s}$. Consequently, the MSED between $\boldsymbol{s}$ and the symbol-vectors with a Hamming distance of one bit from $\boldsymbol{s}$ cannot be larger than $(N-1)d_5^2 + d_4^2$. As a result, the maximum possible $\hat{d}^2_{min}$ for a $2N$-D 16-QAM mapping is $(N-1)d_5^2 + d_4^2$. □

Proposition 5.2. *Let us assume that $\boldsymbol{l} = [l^1, \cdots, l^n]$ is a particular n-bit sequence and $\boldsymbol{l}_k = [l_k^1, \cdots, l_k^n]$ is different with $\boldsymbol{l}$ only in the k^{th} bit position, where $k \in \{1, \cdots, n\}$. Also, assume that $\hat{\boldsymbol{l}} = [\hat{l}^1, \cdots, \hat{l}^n]$ and $\hat{\boldsymbol{l}}_k = [\hat{l}_k^1, \cdots, \hat{l}_k^n]$ are the precoded versions of $\boldsymbol{l}$ and $\boldsymbol{l}_k$, respectively. Let us define $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ as*

$$\begin{aligned} \hat{\boldsymbol{l}}^j &= [\hat{l}^{4(j-1)+1}, \cdots, \hat{l}^{4j}], \\ \hat{\boldsymbol{l}}_k^j &= [\hat{l}_k^{4(j-1)+1}, \cdots, \hat{l}_k^{4j}], \end{aligned} \tag{5.4}$$

where $j \in \{1, \cdots, N\}$. Then, for all values of j, the Hamming distance between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ is either 3 or 4 bits.

Proof. Suppose that the chosen-index in (5.1) is equal to p. Regarding Proposition 2.1, if p is equal to k, then $\hat{\boldsymbol{l}}$ and $\hat{\boldsymbol{l}}_k$ are different in all n bits; otherwise, they are the same in the k^{th} bit position and are different in the remaining $(n-1)$ bits. As a result, we can write

$$d_H(\hat{\boldsymbol{l}}^j, \hat{\boldsymbol{l}}_k^j) = \begin{cases} 4 & \text{if } p = k \ , \ \forall j, \\ 4 & \text{if } p \neq k \ , \ \ j = q, \\ 3 & \text{if } p \neq k \ , \ \ j \neq q, \end{cases} \tag{5.5}$$

where $d_H(a, b)$ is the Hamming distance between a and b and q is given by

$$q = \lfloor \frac{k-1}{4} \rfloor + 1, \tag{5.6}$$

where $\lfloor . \rfloor$ represents the floor function.

□

Proposition 5.3. *In (5.5), for all values of j and k, the Hamming distances of 3 and 4 bits between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ occur $n(N-1)+1$ and $(n-1)$ times, respectively.*

Proof. In Proposition 5.2, suppose that the chosen-index is equal to p. Then, regarding Proposition 2.1, there are two possible cases as follows.

Case 1: $k = p$. $\hat{\boldsymbol{l}}$ and $\hat{\boldsymbol{l}}_k$ are different in all n bits. Then, for all values of j and k, there is a 4-bit difference between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$.

Case 2: $k \neq p$. $\hat{\boldsymbol{l}}_i$ and $\hat{\boldsymbol{l}}_k$ are the same in k^{th} bit and they are different in $(n-1)$ remaining bits. Then, using (5.5), the Hamming distance between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ is given by

$$d_H(\hat{\boldsymbol{l}}^j, \hat{\boldsymbol{l}}_k^j) = \begin{cases} 3 & \text{if } j = q, \\ 4 & \text{otherwise,} \end{cases} \tag{5.7}$$

where q is defined in (5.6).

For a given value of p and for all values of k, *Case 1* occurs only one time. In this case, for $j = 1, \cdots, N$, the Hamming distance of 4 bits between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ occurs N times overall. Similarly, since for a given p, there are $(n-1)$ possible values of k such that $k \neq p$, then *Case 2* occurs $(n-1)$ times. Each occurrence results in $(N-1)$ Hamming distance of 4 bits and one Hamming distance of 3 bits between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$, where $j = 1, \cdots, N$. Therefore, for a particular $\boldsymbol{l}$ and for all values of j and k, between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$, a 4-bit distance occurs $n(N-1)+1$ times and a 3-bit distance occurs $(n-1)$ times. $\square$

Suppose that in the proposed MD mapping, $\boldsymbol{l}$ is mapped to the symbol-vector $\boldsymbol{s} = [s^1, \cdots, s^N]$ and $\boldsymbol{l}_k$ is mapped to $\boldsymbol{s}_k = [s_k^1, \cdots, s_k^N]$, where $s^j = \lambda(\hat{\boldsymbol{l}}^j)$ and $s_k^j = \lambda(\hat{\boldsymbol{l}}_k^j)$ are two symbols in the 16-QAM intermediate mapping and they have a Hamming distance of either 3 or 4 bits from each other. As a result, to increase $\hat{d}_{min}^2$, i. e., the MSED between $\boldsymbol{s}$ and $\boldsymbol{s}_k$, our approach is to increase the MSED between the symbols with a Hamming distance of 3 or 4 bits in the intermediate 16-QAM mapping. Moreover, as Proposition 5.3 declares, the 4-bit distance between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ is more often. Therefore, it is preferred that the MSED between the symbols with 4-bit distance in the intermediate mapping to be as large as possible.

5.1.1 16-QAM intermediate mapping

Each symbol in a 16-QAM mapping is mapped by a 4-bit binary label. In the set of all possible 4-bit labels, there are 5 labels with a Hamming distance of 3 or 4 bits from a given label, $\boldsymbol{l}$. We refer to these 5 labels as the forbidden-labels of label $\boldsymbol{l}$. Let d_{min}^2 be the desired MSED between a particular symbol, S, and the symbols that have a Hamming distance of 3 or 4 bits from S. We refer to symbols whose SED from S is less than d_{min}^2 as the forbidden-symbols of symbol S and to the remaining symbols as the authorized-symbols of symbol S. As the main principle, if a label is mapped to the symbol S, none of its forbidden-labels can be mapped to the forbidden-symbols of symbol S. We start the mapping process by assigning labels to the central-symbols because central-symbols have the smallest number of authorized-symbols; therefore, it is easier to map them at first. The maximum possible d^2_{min} for a 16-QAM constellation is d^2_4. In fact, d^2_4 is the maximum MSED between a central-symbol and any set of 5 symbols in a 16-QAM constellation.

Let in Fig. 5.1, A_i indicate the set of authorized-symbols for symbol S_i. Then, for the central-symbols, i.e., S_6, S_7, S_{10}, and S_{11}, we have $A_6 = \{S_4, S_{12}, S_{13}, S_{15}, S_{16}\}$, $A_7 = \{S_1, S_9, S_{13}, S_{14}, S_{16}\}$, $A_{10} = \{S_1, S_3, S_4, S_8, S_{16}\}$, and $A_{11} = \{S_1, S_2, S_4, S_5, S_{13}\}$. The label of each central-symbol has 5 forbidden-labels. Moreover, each of the forbidden-labels is allowed to be mapped only to one of the corresponding authorized-symbols. As a result, if S_i and S_j are two central-symbols with α common members in their set of authorized-symbols, then they must have α common members in their set of forbidden-labels as well. For a 16-QAM constellation, $\alpha = 2$. Therefore, the labels of two particular central-symbols must have two common members in their set of forbidden-labels. Consequently, regarding the definition of forbidden-labels, the Hamming distance between two central-symbols should be either one or two bits.

The maximum Euclidean distance between a central-symbol and its authorized-symbols is d_5. Moreover, each central-symbol has only one authorized-symbol at the Euclidean distance of d_5. In the proposed intermediate 16-QAM mapping, we constrain the Hamming distance between a central-symbol and its only authorized-symbol at the Euclidean distance of d_5 to be 4 bits. For example, S_{16} is the only symbol from A_6 at the Euclidean distance d_5 from the central-symbol S_6. Therefore, the labels mapped to S_6 and S_{16} should be different in all 4 bits. The symbol S_{16} is also a common member in both A_7 and A_{10}, which constrains both symbols S_7 and S_{10} to have a 3-bit Hamming distance from S_{16}. This causes the symbol S_6 to have a one bit Hamming distance from both S_7 and S_{10}, which are at the Euclidean distance d_1 from S_6. There are the same conditions for other central-symbols whose Euclidean distance from each other is d_1.

In summary, the intermediate mapping is developed in the three following steps:

1- The central-symbols are mapped such that there is a one bit Hamming distance between two central-symbols with the Euclidean distance d_1.

2- Let S be a central-symbol that is mapped by the label $\boldsymbol{l}$. The symbol at the Euclidean distance d_5 from S should be mapped by the label $\bar{\boldsymbol{l}}$.

3- Regarding the main principle, each of the eight remaining labels has two permitted symbols to map. However, some of these labels are in the forbidden-labels of each other. As a result, mapping a label to one of the remaining symbols affects the rest of the mapping process. This is discussed further in the following.

Assume that the chosen-index in the precoding function in (5.1) is equal to p and the two n-bit labels, i.e., $\boldsymbol{l}$ and $\boldsymbol{l}_k$, are different only in the k^{th} bit position, where $p \neq k$. Then, the precoded versions of $\boldsymbol{l}$ and $\boldsymbol{l}_k$, i. e., $\hat{\boldsymbol{l}} = [\hat{l}^1, \cdots, \hat{l}^n]$ and $\hat{\boldsymbol{l}}_k = [\hat{l}^1_k, \cdots, \hat{l}^n_k]$, are different in all bits except for in the k^{th} bit position [23]. Suppose that $\boldsymbol{l}$ and $\boldsymbol{l}_k$ are mapped to $\boldsymbol{s} = [s^1, \cdots, s^N]$ and $\boldsymbol{s}_k = [s_k^1, \cdots, s_k^N]$, respectively, where $s^j = \lambda(\hat{\boldsymbol{l}}^j)$, $s_k^j = \lambda(\hat{\boldsymbol{l}}_k^j)$, and $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ are defined in (5.4). Let $\hat{\boldsymbol{l}}^q$ include both p^{th} and k^{th} bits of $\hat{\boldsymbol{l}}$, i.e.,

$$\begin{aligned} q &= \lfloor \frac{p-1}{4} \rfloor + 1 \\ &= \lfloor \frac{k-1}{4} \rfloor + 1. \end{aligned} \tag{5.8}$$

Also assume that $w = Q(k)$ and $z = Q(p)$, where $Q(x)$ is defined as

$$Q(x) = x - 4 \times (q-1). \tag{5.9}$$

Then, $\hat{\boldsymbol{l}}^q = [\hat{l}^{q,1}, \cdots, \hat{l}^{q,4}]$ and $\hat{\boldsymbol{l}}_k^q = [\hat{l}_k^{q,1}, \cdots, \hat{l}_k^{q,4}]$ are the same in the w^{th} bit position, i.e., $\hat{l}^{q,w} = \hat{l}_k^{q,w}$, and they are different in the remaining 3 bits including the z^{th} bit. As a result, in the proposed method, for $\hat{\boldsymbol{l}}^q$ there is no $\hat{\boldsymbol{l}}_k^q$ that is different from $\hat{\boldsymbol{l}}^q$ in all bits except for in the z^{th} bit. Therefore, in the third step of designing the intermediate mapping, the labels that are different in all bits except for in the z^{th} bit are mapped to the symbol pairs with the Euclidean distance d_4 from each other. Then, these symbol pairs will never be used as s^q and s_k^q in the proposed MD mapping. Hence, some symbol pairs with the Hamming distance of 3 bits and the Euclidean distance d_4 in the intermediate mapping will be unused in the proposed MD mapping. As a consequence, in the resulting MD mapping, the MSED of $(N-1)d_5 + d_4$ between two labels with a Hamming distances of one bit will occur fewer times. This diminishes $\hat{N}_{min}$ (defined in 1.9) and makes the proposed mapping much closer to the optimum mapping.

Proposition 5.4. *Suppose that in the third step of designing the intermediate mapping, $\boldsymbol{b}_i = [b_i^1, \cdots, b_i^4]$ and $\boldsymbol{b}_j = [b_j^1, \cdots, b_j^4]$ are two binary labels with the Hamming distance of 3 bits. Let $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ be mapped to the symbols S_i and S_j, which are at the Euclidean distance d_4 from each other. Also, assume that $\boldsymbol{b}_x = [b_x^1, \cdots, b_x^4]$ and $\boldsymbol{b}_y = [b_y^1, \cdots, b_y^4]$ are two binary labels, which are different in only one bit position. Let $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ be mapped to two central-symbols, i.e., S_x and S_y, which are at the Euclidean distance d_1 from each other. Moreover, suppose that $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ have a 3-bit Hamming distance from $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, respectively. Then, in order for $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ to be the same in the z^{th} bit, they need $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ to be different in the z^{th} bit position.*

Proof. Suppose that S_r and S_t are two central-symbols, where S_r is the neighbor of both S_x and S_t, and S_t is the neighbor of S_y. Let S_r and S_t be mapped by the 4-bit binary labels $\boldsymbol{b}_r$ and $\boldsymbol{b}_t$, respectively. Without loss of generality, assume that $z = 1$, $\boldsymbol{b}_i = [0, 0, 0, 0]$

and $\boldsymbol{b}_j = [0,1,1,1]$. Because $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ have not been mapped in the second step then $\bar{\boldsymbol{b}}_i = [1,1,1,1]$ and $\bar{\boldsymbol{b}}_j = [1,0,0,0]$ cannot be used for the central-symbols in the first step. The labels $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$, which have a 3-bits Hamming distance from $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, respectively, will take a label from $\{[1,0,1,1],[1,1,0,1],[1,1,1,0]\}$ and $\{[1,1,0,0],[1,0,1,0],[1,0,0,1]\}$, respectively. Without loss of generality, let $\boldsymbol{b}_x = [1,0,1,1]$, then $\boldsymbol{b}_y$ can take a label from $\{[1,0,1,0],[1,0,0,1]\}$. Let $\boldsymbol{b}_y = [1,0,1,0]$, then $\boldsymbol{b}_t$, which should differ from $\boldsymbol{b}_y$ at one bit position but not in the z^{th} bit, will take a label from $\{[1,1,1,0],[1,0,0,0]\}$. But, $[1,0,0,0]$ is different from $\boldsymbol{b}_j$ in all bits, and thus, it is unusable for central-symbols. As a result, S_t is mapped by $[1,1,1,0]$. Now, the only choices for $\boldsymbol{b}_r$, which should be different from $\boldsymbol{l}_x$ and $\boldsymbol{l}_t$ at only one bit position but not in the z^{th} bit, are $[1,0,1,0]$ and $[1,1,1,1]$. However, $[1,0,1,0]$ cannot be used for $\boldsymbol{b}_r$ because it has already been mapped to $\boldsymbol{b}_y$; moreover, $[1,1,1,1]$ is different from $\boldsymbol{b}_i$ in all bits, and thus, it is not authorized to be mapped to a central-symbol. Therefore, there is not an authorized label for S_r. The same conditions apply for the other scenarios in which S_x or S_y must select another label from their set of authorized-labels. Consequently, $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ have to be different in the z^{th} bit. $\square$

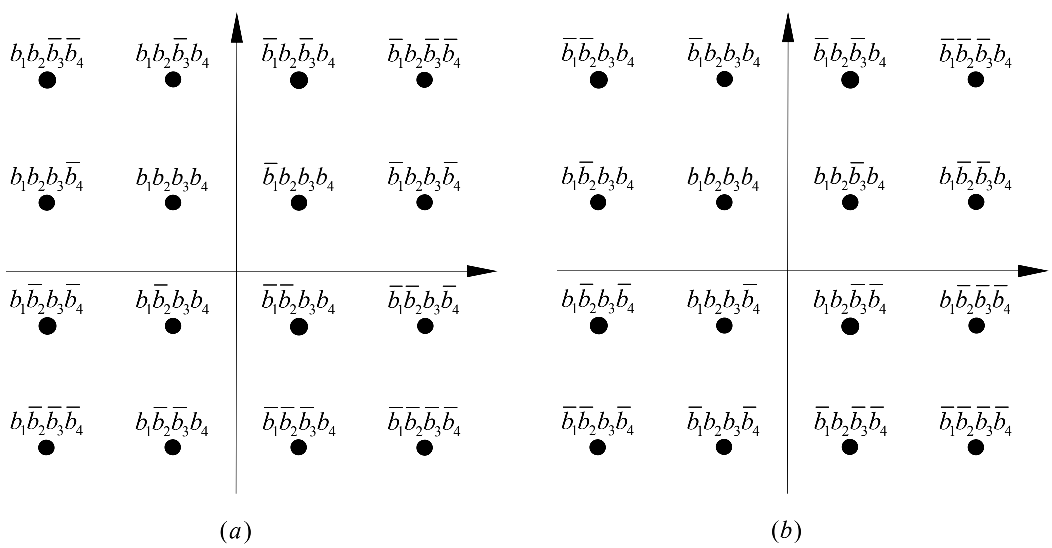

Figure 5.2: Proposed intermediate 16-QAM mapping for (a). $z \in \{1,2\}$, and (b). $z \in \{3,4\}$.

Fig. 5.2. a and Fig. 5.2. b indicate our proposed intermediate mappings of 16-QAM with $z \in \{1,2\}$ and $z \in \{1,2\}$, respectively, where $b_i \in \{0,1\}$ for $i = 1,\cdots,4$. In what follows, we explain how the intermediate mapping of 16-QAM shown in Fig. 5.2. a is developed.

Step 1: As aforementioned, in the first step, the central-symbols are mapped by the binary labels with a Hamming distance of one bit. We first map the label $[b_1, b_2, b_3, b_4]$ to the symbol S_6. Then, S_7 and S_{10}, whose Euclidean distances from S_6 are d_1, are mapped by the labels with the Hamming distance one from S_6, i. e, by $[b_1, \bar{b}_2, b_3, b_4]$ and $[\bar{b}_1, b_2, b_3, b_4]$, respectively. The symbol S_{11} is at the Euclidean distance d_1 from both S_7 and S_{10} thus its Hamming distance from each of S_7 and S_{10} should be one bit. Therefore, it is mapped by $[\bar{b}_1, \bar{b}_2, b_3, b_4]$.

Step 2: The symbols S_1, S_4, S_{13} and S_{16} are at the Euclidean distance d_5 from the central-symbols S_{11}, S_{10}, S_7 and S_6, respectively. As a result, each of them takes a label with a Hamming distance of 4 bits from the label of the corresponding central-symbol at the Euclidean distance d_5.

Step 3: Each of the eight remaining labels for step 3 are forbidden by only one of the central-symbols, and each have two authorized-symbols to map. The first label is chosen randomly and is arbitrarily mapped to one of its authorized-symbols. However, the rest of the labels each one will have only one authorized-symbol to map. The only rule that should be considered is the main principle of the mapping method. For example, the label $[b_1, b_2, b_3, \bar{b}_4]$ has two authorized-symbols to map, i.e., S_2 and S_5. In this example, we map it to S_2. The label $[b_1, \bar{b}_2, \bar{b}_3, b_4]$ is forbidden by S_{10} and it was authorized to be mapped to one of S_3 and S_8. However, it cannot be mapped to S_3 anymore. This is because $[b_1, \bar{b}_2, \bar{b}_3, b_4]$ has a Hamming distance of 3 bits from the label of S_2, and the Euclidean distance of S_3 from S_2 is smaller than d_4. As a result, $[b_1, \bar{b}_2, \bar{b}_3, b_4]$ is mapped to its other authorized-symbol, i.e., S_8. In a similar way, the rest of the labels are mapped to the remaining symbols, as shown in Fig. 5.2. a.

Proposition 5.5. *In the proposed $2N$-D 16-QAM mapping, the MSED between symbol-vectors with a Hamming distance of one bit is equal to $(N-1)d_5^2 + d_4^2$.*

Proof. Let $\boldsymbol{l}$ and $\boldsymbol{l}_k$ be two $4N$-bit labels that are different only in the k^{th} bit position and they are mapped to the symbol-vectors $\boldsymbol{s} = [s^1, \cdots, s^N]$ and $\boldsymbol{s}_k = [s_k^1, \cdots, s_k^N]$, respectively. The precoded versions of $\boldsymbol{l}$ and $\boldsymbol{l}_k$ are $\hat{\boldsymbol{l}} = [\hat{\boldsymbol{l}}^1, \cdots, \hat{\boldsymbol{l}}^N]$ and $\hat{\boldsymbol{l}}_k = [\hat{\boldsymbol{l}}_k^1, \cdots, \hat{\boldsymbol{l}}_k^N]$, respectively, where $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ are defined in (5.4). Using (5.2), for all values of j, $s^j = \lambda(\hat{\boldsymbol{l}}^j)$ and $s_k^j = \lambda(\hat{\boldsymbol{l}}_k^j)$, where λ is the proposed intermediate mapping function. Regarding the specifications of the precoding function, $\hat{\boldsymbol{l}}$ and $\hat{\boldsymbol{l}}_k$ are different either in $4N$ bits or $(4N-1)$ bits. If this distance is $4N$ bits, then for all values of j the Hamming distance between $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ is 4 bits. Otherwise, $\hat{\boldsymbol{l}}^j$ and $\hat{\boldsymbol{l}}_k^j$ have a Hamming distance of 4 bits for $(N-1)$ values of j, and they have a Hamming distance of 3 bits for the one remaining value of j. Moreover, the intermediate mappings indicated in Fig. (5.2) have two characteristics: (i) the Euclidean distance between two symbols with a Hamming distance of 4 bits is d_5 and (ii) the Euclidean distance between two symbols with a Hamming distance of 3 is either d_4 or d_8. As a result, if $\hat{\boldsymbol{l}}$ and $\hat{\boldsymbol{l}}_k$ have a Hamming distance of $4N$ and $(4N-1)$ bits, the MSED between $\boldsymbol{s}$ and $\boldsymbol{s}_k$ is equal to Nd_5^2 and $(N-1)d_5^2+d_4^2$, respectively. Consequently, overall in the proposed $2N$-D 16-QAM mapping, the MSED between $\boldsymbol{s}$ and $\boldsymbol{s}_k$ is equal to $(N-1)d_5^2+d_4^2$. □

5.1.2 Proposed 4-D 16-QAM mapping

Example 5.1. Let in the proposed MD 16-QAM mapping, $N=2$, the chosen-index be equal to one ($p=1$), $\boldsymbol{l}=[0,1,1,1,0,0,1,1]$, and in Fig. 5.2, $b_i=0$ for $i=1,...,4$. Using (5.9) we have

$$z = Q(p) = 1. \tag{5.10}$$

As a result, the mapping in Fig. 5.2(a) is used as the intermediate mapping. The precoded version of $\boldsymbol{l}$, i.e., $\hat{\boldsymbol{l}}=[1,0,0,0,1,1,0,0]$, can be rewritten as $\hat{\boldsymbol{l}}=[\hat{\boldsymbol{l}}^1,\hat{\boldsymbol{l}}^2]$, where $\hat{\boldsymbol{l}}^1=[1,0,0,0]$ and $\hat{\boldsymbol{l}}^2=[1,1,0,0]$. Let in the proposed mapping method, $\boldsymbol{l}$ be mapped to $\boldsymbol{s}=[s^1,s^2]$; then, we can write

$$\begin{aligned} s^1 &= \lambda(\hat{\boldsymbol{l}}_1) = \lambda([1,0,0,0]) = S_{10}, \\ s^2 &= \lambda(\hat{\boldsymbol{l}}_2) = \lambda([1,1,0,0]) = S_{11}. \end{aligned} \tag{5.11}$$

As a consequence, in the proposed 4-D 16-QAM mapping, the label $\boldsymbol{l}=[0,1,1,1,0,0,1,1]$ is mapped to the symbol-vector $\boldsymbol{s}=[S_{10},S_{11}]$.

Table 5.1 shows the proposed 4-D 16-QAM mapping. To achieve this mapping, it is

Table 5.1: The proposed 4D mapping of 16-QAM for chosen-index equal to 1.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
S_1	51	177	53	183	178	48	180	54	58	184	60	190	187	57	189	63
S_2	147	17	149	23	18	144	20	150	154	24	156	30	27	153	29	159
S_3	83	209	85	215	210	80	212	86	90	216	92	222	219	89	221	95
S_4	243	113	245	119	114	240	116	246	250	120	252	126	123	249	125	255
S_5	163	33	165	39	34	160	36	166	170	40	172	46	43	169	45	175
S_6	3	129	5	135	130	0	132	6	10	136	12	142	139	9	141	15
S_7	195	65	197	71	66	192	68	198	202	72	204	78	75	201	77	207
S_8	99	225	101	231	226	96	228	102	106	232	108	238	235	105	237	111
S_9	220	94	218	88	93	223	91	217	213	87	211	81	84	214	82	208
S_{10}	124	254	122	248	253	127	251	121	117	247	115	241	244	118	242	112
S_{11}	188	62	186	56	61	191	59	185	181	55	179	49	52	182	50	176
S_{12}	28	158	26	152	157	31	155	25	21	151	19	145	148	22	146	16
S_{13}	76	206	74	200	205	79	203	73	69	199	67	193	196	70	194	64
S_{14}	236	110	234	104	109	239	107	233	229	103	227	97	100	230	98	224
S_{15}	44	174	42	168	173	47	171	41	37	167	35	161	164	38	162	32
S_{16}	140	14	138	8	13	143	11	137	133	7	131	1	4	134	2	128

assumed that the chosen-index in (5.1) is equal to one, and b_i in Fig. 5.2 is equal to 0 for $i = 1, ..., 4$. In this table, the label in the $(j+1; k+1)^{th}$ entry is mapped to the symbol-vector $\boldsymbol{x} = (S_j, S_k)$. For example, the label 115, corresponding to the binary label $\boldsymbol{l} = [0,1,1,1,0,0,1,1]$, is the $(11,12)^{th}$ entry of Table 5.1. This means that 115 is mapped to the symbol-vector $\boldsymbol{x} = [S_{10}, S_{11}]$.

5.2 MD mapping of 2^m-QAM

We propose a transfer system that takes the proposed MD 16-QAM as the input and constructs a desired MD mapping of rectangular 2^m-QAM through $m-3$ steps ($m > 4$). In this approach, 16 symbols are first chosen from the 2^m-QAM constellation. Then, an MD mapping is developed employing these selected symbols and using the proposed MD 16-QAM mapping. It is important to note that the proposed method is specifically designed for the rectangular 2^m-QAM constellations. Such constellations are composed of $2^{\lfloor \frac{m}{2} \rfloor}$ rows and $2^{m-\lfloor \frac{m}{2} \rfloor}$ columns of signal points (symbols). In what follows, we describe the proposed method in two main sections.

5.2.1 MD mapping using 16 symbols from 2^m-QAM

The 16 chosen symbols from 2^m-QAM in the first step of the proposed method is shown in Table 5.2. In this Table, Λ_j represents the 2^m-QAM symbol with index j. These 16 symbols are selected such that their structure in the 2^m-QAM constellation is equivalent to a 16-QAM constellation (see Fig. 5.3 as an example for $m = 6$).

Let in the proposed method, $\boldsymbol{l} = [l^{(1)}, \cdots, l^{(mN)}]$ be an mN-bit sequence that is mapped to a vector of N symbols from 2^m-QAM. In step i, $\boldsymbol{l}_i = [l_i^{(1)}, \cdots, l_i^{((i+3)N)}]$ denotes the $(i+3)N$ least significant bits of $\boldsymbol{l}$, where l_i^j is given by

$$l_i^j = l^{(mN-(i+3)N+j)}. \tag{5.12}$$

Table 5.2: The 16 chosen symbols from 2^m-QAM in the first step of the proposed method.

1	Λ_1	9	$\Lambda_{1+2^{m-1}}$
2	$\Lambda_{1+2^{\lfloor \frac{m}{2} \rfloor-2}}$	10	$\Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor-2}}$
3	$\Lambda_{1+2^{\lfloor \frac{m}{2} \rfloor-1}}$	11	$\Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor-1}}$
4	$\Lambda_{1+3\times2^{\lfloor \frac{m}{2} \rfloor-2}}$	12	$\Lambda_{1+2^{m-1}+3\times2^{\lfloor \frac{m}{2} \rfloor-2}}$
5	$\Lambda_{1+2^{m-2}}$	13	$\Lambda_{1+3\times2^{m-2}}$
6	$\Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor-2}}$	14	$\Lambda_{1+3\times2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor-2}}$
7	$\Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor-1}}$	15	$\Lambda_{1+3\times2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor-1}}$
8	$\Lambda_{1+2^{m-2}+3\times2^{\lfloor \frac{m}{2} \rfloor-2}}$	16	$\Lambda_{1+3\times2^{m-2}+3\times2^{\lfloor \frac{m}{2} \rfloor-2}}$

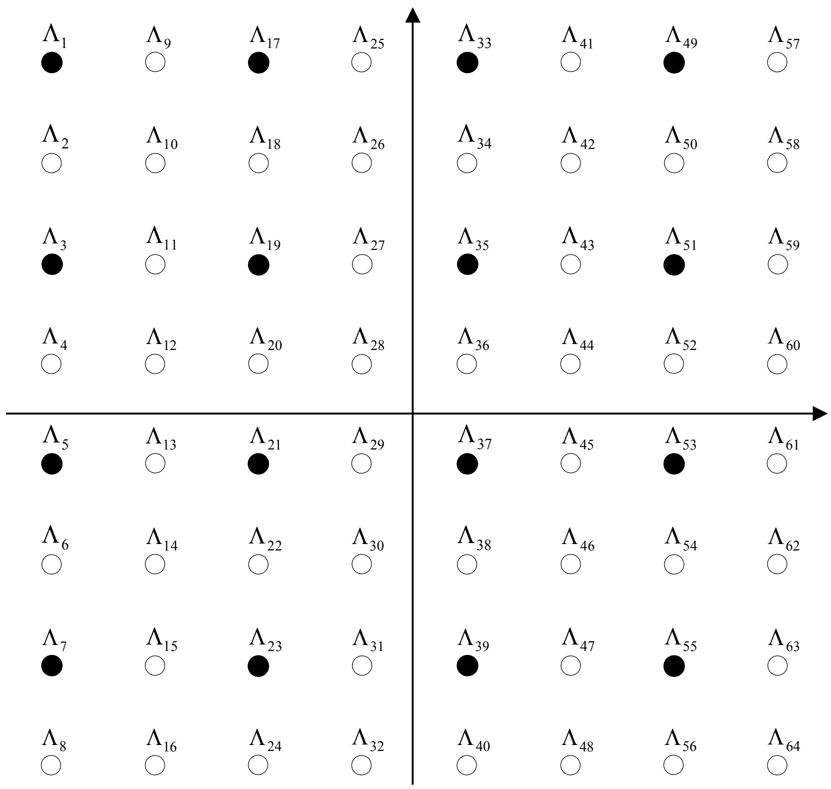

Figure 5.3: 64-QAM constellation. The black symbols represent the 16 selected symbols in the first step of the proposed mapping method.

In step $i = 1$, the proposed MD 16-QAM mapping method is applied to map $\boldsymbol{l}_1$ to an N-tuple vector of 16-QAM symbols. Next, each 16-QAM symbol in the resulting mapping is replaced by one of the 16 chosen symbols from 2^m-QAM constellation, according to Table 5.3. In this table, for example, $S_i \leftarrow \Lambda_j$ means that in a given symbol-vector, the 16-QAM symbol S_i should be replaced by the 2^m-QAM symbol Λ_j.

Table 5.3: The chart of substitution of 2^m-QAM symbols for the 16-QAM symbols.

$S_1 \leftarrow \Lambda_1$	$S_9 \leftarrow \Lambda_{1+2^{m-1}}$
$S_2 \leftarrow \Lambda_{1+2^{\lfloor \frac{m}{2} \rfloor -2}}$	$S_{10} \leftarrow \Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor -2}}$
$S_3 \leftarrow \Lambda_{1+2^{\lfloor \frac{m}{2} \rfloor -1}}$	$S_{11} \leftarrow \Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor -1}}$
$S_4 \leftarrow \Lambda_{1+3\times 2^{\lfloor \frac{m}{2} \rfloor -2}}$	$S_{12} \leftarrow \Lambda_{1+2^{m-1}+3\times 2^{\lfloor \frac{m}{2} \rfloor -2}}$
$S_5 \leftarrow \Lambda_{1+2^{m-2}}$	$S_{13} \leftarrow \Lambda_{1+3\times 2^{m-2}}$
$S_6 \leftarrow \Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -2}}$	$S_{14} \leftarrow \Lambda_{1+3\times 2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -2}}$
$S_7 \leftarrow \Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -1}}$	$S_{15} \leftarrow \Lambda_{1+3\times 2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -1}}$
$S_8 \leftarrow \Lambda_{1+2^{m-2}+3\times 2^{\lfloor \frac{m}{2} \rfloor -2}}$	$S_{16} \leftarrow \Lambda_{1+3\times 2^{m-2}+3\times 2^{\lfloor \frac{m}{2} \rfloor -2}}$

According to proposition 5.5, in the proposed $2N$-D 16-QAM mapping, $\hat{d}^2_{min} = (N-1)d_5^2 + d_4^2$, where d_4 and d_5 are equal to the Euclidean distance of S_1 from S_7 and S_{11}, respectively. Furthermore, in Table 5.3, the 16-QAM symbols S_1, S_7 and S_{11} are replaced by 2^m-QAM symbols Λ_1, $\Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor -1}}$ and $\Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -1}}$, respectively. As a consequence, $\hat{d}^2_{min}$ for the obtained mapping using 16 symbols from 2^m-QAM is equal to $(N-1)\hat{d}_5^2+\hat{d}_4^2$, where $\hat{d}_4$ and $\hat{d}_5$ represent the Euclidean distance of Λ_1 from $\Lambda_{1+2^{m-1}+2^{\lfloor \frac{m}{2} \rfloor -1}}$ and $\Lambda_{1+2^{m-2}+2^{\lfloor \frac{m}{2} \rfloor -1}}$, respectively. In other words, we can write

$$\begin{aligned} \hat{d}_4^2 &= 2^{2m-2\lfloor \frac{m}{2} \rfloor -4} + 2^{2\lfloor \frac{m}{2} \rfloor -2}, \\ \hat{d}_5^2 &= 2^{2m-2\lfloor \frac{m}{2} \rfloor -2} + 2^{2\lfloor \frac{m}{2} \rfloor -2}. \end{aligned} \tag{5.13}$$

5.2.2 Transferring from 16-QAM to 2^m-QAM

In step $i-1$ ($i > 1$), let $\boldsymbol{l}_{i-1}$ be mapped to $\boldsymbol{s}_{i-1} = [s^1_{i-1}, \cdots, s^N_{i-1}]$, where $s^j_{i-1} = x^j_{i-1} + I y^j_{i-1}$ and $I^2 = -1$. In step i, $\boldsymbol{l}_i$ is mapped to $\boldsymbol{s}_i = [s^1_i, \cdots, s^N_i]$, where $s^j_i = x^j_i + I y^j_i$ and x^j_i and y^j_i are calculated using Table 5.4. In this table, $\boldsymbol{e}_i = [e^1_i, \cdots, e^N_i]$ represents the N most significant bits of $\boldsymbol{l}_i$, i.e., $e^j_i = l^j_i$, w_i denotes the Hamming weight of $\boldsymbol{e}_i$, $\mathbb{E}$ and $\mathbb{O}$ represent the set of even and odd integers, respectively, and $sgn(.)$ is the sign function, which is defined as

$$sgn(X) = \begin{cases} 1 & \text{if } X > 0, \\ 0 & \text{if } X = 0, \\ -1 & \text{if } X < 0. \end{cases} \tag{5.14}$$

For example, in step i, if $w_i \in \mathbb{E}$, $e^j_i = 1$, and $i \in \mathbb{E}$, in order to calculate the real part of s^j_i, i.e., x^j_i, one needs to use $x^j_i = x^j_{i-1} + 2^{m-\lfloor \frac{m}{2} \rfloor - \lfloor \frac{i}{2} \rfloor -2}$ from Table 5.4.

Table 5.4: Transfer system to generate symbol coordinates for MD 2^m-QAM mappings.

		$w_i \in \mathbb{E}$	$w_i \in \mathbb{O}$
$e^j_i = 0$		$x^j_i = x^j_{i-1}$ $y^j_i = y^j_{i-1}$	$x^j_i = x^j_{i-1} - sgn(x^j_{i-1})2^{m-\lfloor \frac{m}{2} \rfloor -1}$ $y^j_i = y^j_{i-1} - sgn(y^j_{i-1})2^{\lfloor \frac{m}{2} \rfloor -1}$
$e^j_i = 1$	$i \in \mathbb{E}$	$x^j_i = x^j_{i-1} + 2^{m-\lfloor \frac{m}{2} \rfloor - \lfloor \frac{i}{2} \rfloor -2}$ $y^j_i = y^j_{i-1}$	$x^j_i = x^j_{i-1} + (2^{-\lfloor \frac{i}{2} \rfloor -1} - sgn(x^j_{i-1}))2^{m-\lfloor \frac{m}{2} \rfloor -1}$ $y^j_i = y^j_{i-1} - sgn(y^j_{i-1})2^{\lfloor \frac{m}{2} \rfloor -1}$
	$i \in \mathbb{O}$	$x^j_i = x^j_{i-1}$ $y^j_i = y^j_{i-1} - 2^{\lfloor \frac{m}{2} \rfloor - \lfloor \frac{i-1}{2} \rfloor -2}$	$x^j_i = x^j_{i-1} - sgn(x^j_{i-1})2^{m-\lfloor \frac{m}{2} \rfloor -1}$ $y^j_i = y^j_{i-1} - (2^{-\lfloor \frac{i-1}{2} \rfloor -1} + sgn(y^j_{i-1}))2^{\lfloor \frac{m}{2} \rfloor -1}$

Proposition 5.6. *Let* $\boldsymbol{l}_i = [l_i^1, \cdots, l_i^{(i+3)N}]$ *and* $\boldsymbol{l}_{i,k} = [l_{i,k}^1, \cdots, l_{i,k}^{(i+3)N}]$ *be two labels in step* i *that are different only in the* k^{th} *bit position and are mapped to symbol-vectors* $\boldsymbol{s}_i = [s_i^1, \cdots, s_i^N]$ *and* $\boldsymbol{s}_{i,k} = [s_{i,k}^1, \cdots, s_{i,k}^N]$*, respectively. For all* i*, the MSED between* $\boldsymbol{s}_i$ *and* $\boldsymbol{s}_{i,k}$*, i.e.,* $\hat{d}^2_{min,i}$*, is greater than or equal to* $(N-1)\hat{d}_5^2 + \hat{d}_4^2$.

Proof. See Appendix B. □

Example 5.2. In the proposed MD mapping method, let us set $m = 6$ (64-QAM), $N = 2$, and $\boldsymbol{l} = [1,1,1,0,0,1,1,1,0,0,1,1]$. In the following three steps, $\boldsymbol{l}$ is mapped to a vector of 64-QAM symbols, i.e., $\boldsymbol{s} = [s^1, s^2]$.

Step 1: Let $\boldsymbol{l}_1 = [0,1,1,1,0,0,1,1]$ be the eight least significant bits of $\boldsymbol{l}$ and the chosen-index be equal to one ($p = 1$). The sequence $\boldsymbol{l}_1$ is mapped to the symbol-vector $\boldsymbol{s}_1 = [s_1^1, s_1^2]$, where $s_1^1 = x_1^1 + Iy_1^1$ and $s_1^2 = x_1^2 + Iy_1^2$ belong to the 64-QAM signal set. In the proposed 4-D 16-QAM mapping when $p = 1$, the sequence $[0,1,1,1,0,0,1,1]$, i.e., $\boldsymbol{l}_1$, is mapped to the symbol-vector $[S_{10}, S_{11}]$ (c.f. Example 5.1, Section 5.1.2). From Table 5.3, the equivalent 64-QAM symbols for S_{10} and S_{11} are Λ_{35} and Λ_{37}, respectively. As a result, $s_1^1 = \Lambda_{35}$ and $s_1^2 = \Lambda_{37}$.

Step 2: Let $\boldsymbol{l}_2 = [1,0,0,1,1,1,0,0,1,1]$ be the ten least significant bits of $\boldsymbol{l}$ and be mapped to $\boldsymbol{s}_2 = [s_2^1, s_2^2]$. Considering $\boldsymbol{l}_1$ from step 1, $\boldsymbol{l}_2$ can be rewritten as $\boldsymbol{l}_2 = [\boldsymbol{e}_2, \boldsymbol{l}_1]$, where $\boldsymbol{e}_2 = [1,0]$. Table 5.4 is used to obtain the symbols s_2^1 and s_2^2 using the symbols s_1^1 and s_1^2 from step 1, respectively. Since $i \in \mathbb{E}$ ($i = 2$), the Hamming weight of $\boldsymbol{e}_2$ is odd ($w_i \in \mathbb{O}$), and $e_2^1 = 1$, then the following equations from Table 5.4 are used to calculate the real and imaginary parts of s_2^1.

$$\begin{aligned} x_i^j &= x_{i-1}^j + (2^{-\lfloor \frac{i}{2} \rfloor - 1} - sgn(x_{i-1}^j))2^{m - \lfloor \frac{m}{2} \rfloor - 1} \\ y_i^j &= y_{i-1}^j - sgn(y_{i-1}^j)2^{\lfloor \frac{m}{2} \rfloor - 1}. \end{aligned} \tag{5.15}$$

Let us set $x_1^1 = 0.5$ and $y_1^1 = 1.5$ ($s_1^1 = \Lambda_{35} = 0.5 + I1.5$) in the above equations, which results in $x_2^1 = -2.5$ and $y_2^1 = -2.5$. Similarly, since $e_2^2 = 0$, then the following equations from Table 5.4 are used to calculate the real and imaginary parts of s_2^2.

$$\begin{aligned} x_i^j &= x_{i-1}^j - sgn(x_{i-1}^j)2^{m - \lfloor \frac{m}{2} \rfloor - 1} \\ y_i^j &= y_{i-1}^j - sgn(y_{i-1}^j)2^{\lfloor \frac{m}{2} \rfloor - 1}. \end{aligned} \tag{5.16}$$

From step 1, we set $x_1^2 = 0.5$ and $y_1^2 = -0.5$ ($s_1^2 = \Lambda_{37} = 0.5 - I0.5$) in the above equations, which results in $x_2^2 = -3.5$ and $y_2^2 = 3.5$.

Step 3: In this step, $\boldsymbol{l}_3 = \boldsymbol{l}$ and it can be rewritten as $\boldsymbol{l}_3 = [\boldsymbol{e}_3, \boldsymbol{l}_2]$, where $\boldsymbol{e}_3 = [1, 1]$. Let $\boldsymbol{l}_3$ be mapped to $\boldsymbol{s}_3 = [s_3^1, s_3^2]$. The symbols s_3^1 and s_3^2 are obtained as follows. Since $i \in \mathbb{O}$ $(i = 3)$, the Hamming weight of $\boldsymbol{e}_3$ is even $(w_i \in \mathbb{E})$, and $e_3^1 = e_3^2 = 1$, then the following equations from Table 5.4 are used to calculate the real and imaginary parts of s_3^1 and s_3^2.

$$\begin{aligned} x_i^j &= x_{i-1}^j \\ y_i^j &= y_{i-1}^j - 2^{\lfloor \frac{m}{2} \rfloor - \lfloor \frac{i-1}{2} \rfloor - 2}. \end{aligned} \tag{5.17}$$

Substituting x_2^1, x_2^2, y_2^1, and y_2^2 in the above equations gives $s_3^1 = -2.5 - I3.5$ $(= \Lambda_{16})$ and $s_3^2 = -3.5 + I2.5$ $(= \Lambda_2)$. Consequently, $\boldsymbol{l}$ is mapped to $\boldsymbol{s} = (\Lambda_{16}, \Lambda_2)$.

5.3 Numerical results

This section provides a selection of numerical examples to illustrate the performance and advantage of our proposed MD mapping for BICM-ID systems over the AWGN channel. We compare our resulting mappings with the mappings that are optimized for the AWGN channel employing the well-known BSA [9]. From many aspects, the BSA is the best known computer search technique to find desired mappings for BICM-ID. However, the BSA becomes intractable to achieve suitable mappings for larger MD constellations, e.g., 6-D 64-QAM, due to computational time constraints. We consider a rate-1/2 convolutional code with the generator polynomial of $(13, 15)_8$. An interleaver length of about 10000 bits is used. All gains reported in this section are measured at a BER of 10^{-6}.

Table 5.5 compares the values of $\hat{d}^2_{min}$ and $\hat{N}_{min}$ for our proposed mappings and the BSA mappings of MD 2^m-QAM $(m = 4, 5, 6)$. In the case of MD mappings of higher order

Table 5.5: $\hat{d}^2_{min}$ and $\tilde{N}_{min}$ for different mappings.

Modulations	$\hat{d}^2_{min}$	$\tilde{N}_{min}$
BSA 4-D 16-QAM	1.2	2
Proposed 4-D 16-QAM	2.60	6.40×10^2
BSA 6-D 16-QAM	1.33	1.9×10^2
Proposed 6-D 16-QAM	2.8	1.6×10^4
BSA 4-D 32-QAM	0.8	1
Proposed 4-D 32-QAM	2.15	1.02×10^3
BSA 4-D 64-QAM	1.2	2
Proposed 4-D 64-QAM	2.48	1.02×10^4

Table 5.6: $\hat{d}^2_{min}$ and $\tilde{N}_{min}$ for MD mapping of higher order modulations.

Modulations	$\hat{d}^2_{min}$	$\tilde{N}_{min}$
Proposed 4-D 128-QAM	2.11	1.6×10^4
Proposed 4-D 256-QAM	2.45	1.6×10^5
Proposed 4-D 512-QAM	2.10	2.6×10^5
Proposed 4-D 1024-QAM	2.44	2.6×10^6

QAMs, the BSA results could not be obtained due to the computational time constraints. Therefore, in Table 5.6, we report the values of $\hat{d}^2_{min}$ and $\hat{N}_{min}$ only for the proposed MD mapping of larger constellations. Table 5.5 shows that in comparison with the BSA mappings, the proposed mappings offer greater values of $\hat{d}^2_{min}$. As a result, it is expected that the proposed mappings improve the error performance of BICM-ID over the AWGN channel. This is confirmed by the simulation results plotted in Fig. 5.4 and Fig. 5.5. It can be observed from Fig. 5.4 that the proposed 4-D 16-QAM and 6-D 16-QAM mappings outperform their BSA counterparts by 1.4 dB and 2.5 dB, respectively. Fig. 5.5 indicates that the proposed 4-D 32-QAM and 4-D 64-QAM mappings offer a gain of 2.55 dB and 3.6 dB, respectively, over the corresponding BSA mappings. The error bounds for different mappings are plotted in Fig. 5.6 and Fig. 5.7. These figures show that the proposed mappings offer lower error floors in comparison with the BSA mappings.

Please note that the BSA becomes intractable when finding a MD mapping of a large modulation. But, our proposed method is a heuristic method and generates good MD mappings of large modulations instantaneously. Furthermore, for smaller MD modulations, our proposed mappings improve the system performance compared to the BSA mappings. This shows the efficiency of our proposed method compared to the BSA.

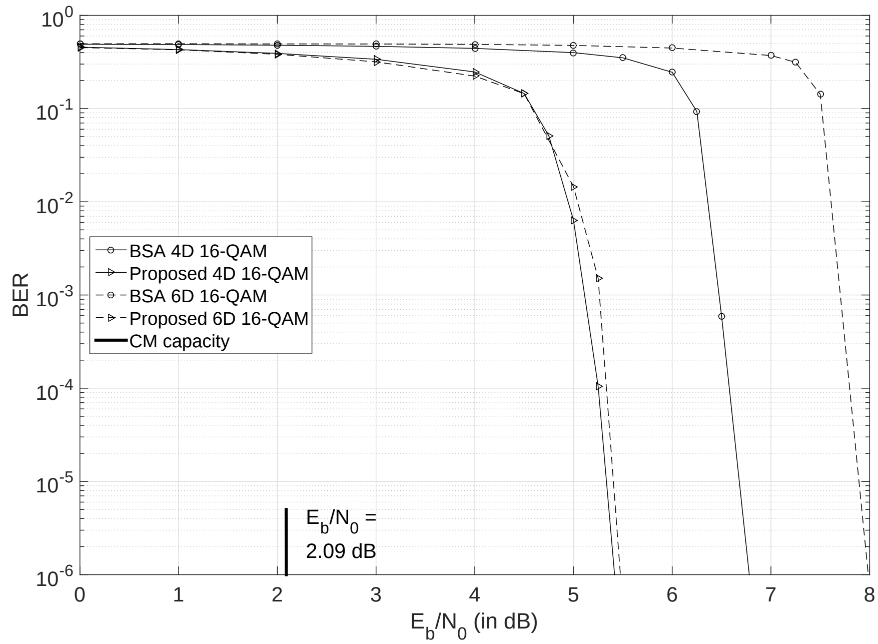

Figure 5.4: BER performance of BICM-ID over the AWGN channel.

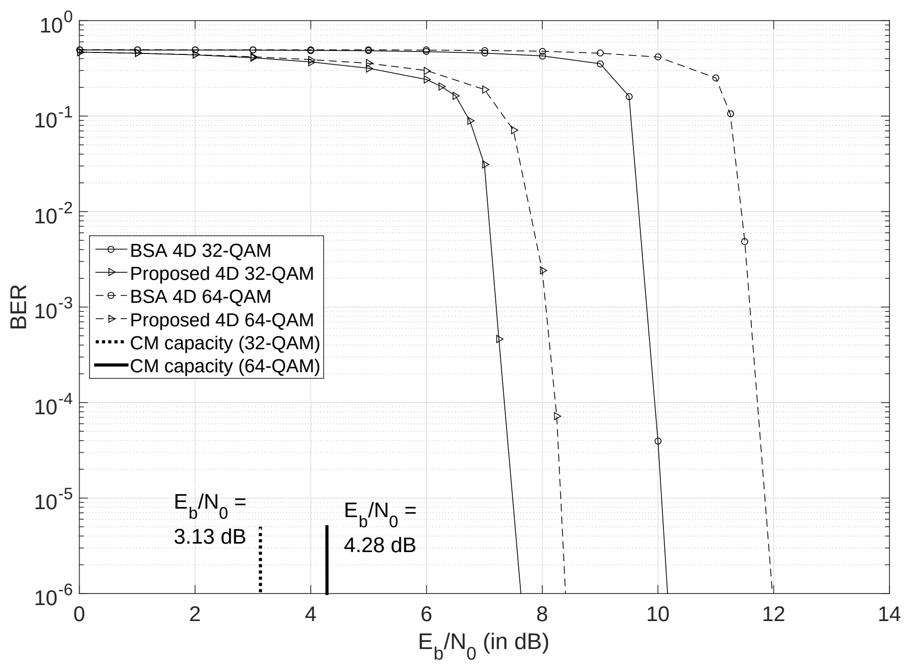

Figure 5.5: BER performance of BICM-ID over the AWGN channel.

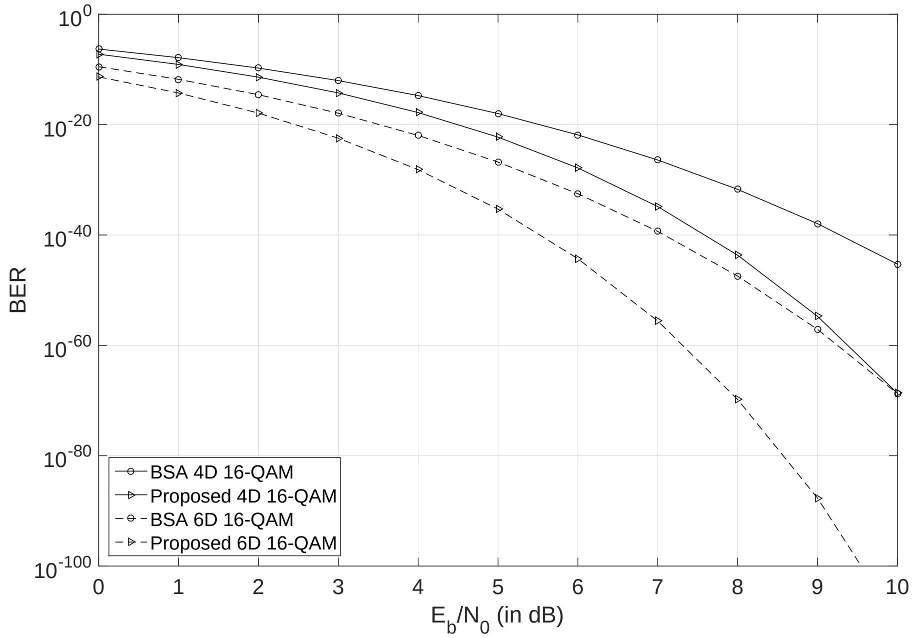

Figure 5.6: Error-floor bounds over the AWGN channel.

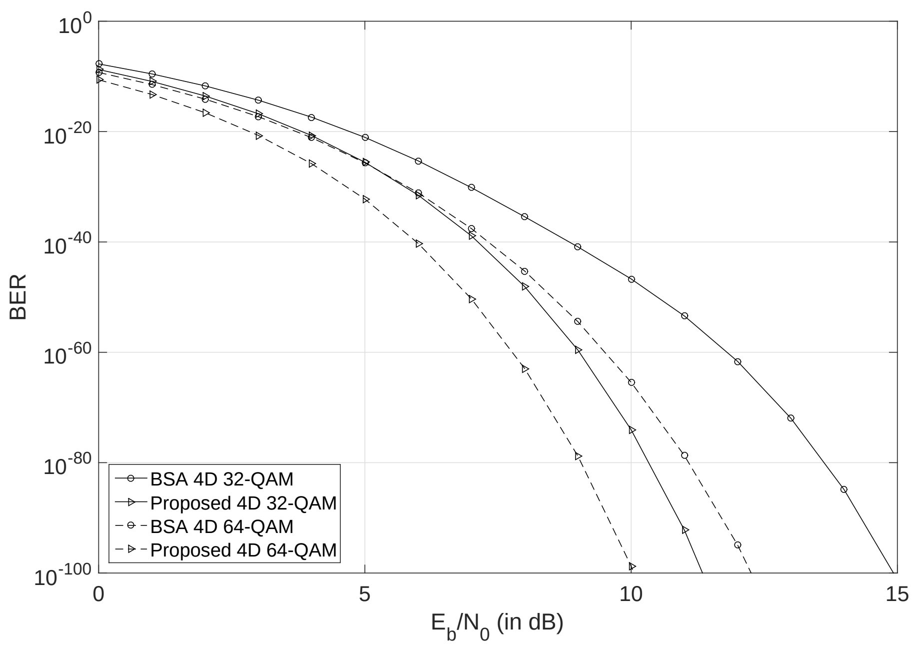

Figure 5.7: Error-floor bounds over the AWGN channel.

Chapter 6

Conclusions

6.1 Work accomplished in this thesis

In this thesis, we have studied the mapping problem of BICM-ID for a wide range of modulations. In particular, we have focused on the 2-D and MD mapping of 2^m-ary (unlimited m) modulations. We have proposed heuristic methods as well as computer search techniques to achieve efficient mappings for BICM-ID. Presented analytic and simulation results show that in comparison with the-state-of-the-art mappings such as the BSA mappings, our proposed mappings significantly improve the system's error performance over the AWGN and Rayleigh fading channels.

In Chapter 2, we have proposed a novel mapping method to find efficient 2-D mappings of higher order QAM and PSK modulations for BICM-ID. Our method generates mappings through a systematic approach. Two main qualities of our proposed method are as follows: (i) it is a very simple method, and (ii) it generates efficient mappings for 2^m-ary QAM and PSK with an unlimited value of m. Simulation results show that compared to the best previously known mappings at a target BER rate of 10^-6, our proposed mappings can save up to 5.7 dB and 4.4 dB of the transmit signal over the AWGN and Rayleigh fading channels, respectively.

In Chapter 3, we have proposed a heuristic method to design MD mappings for BICM-ID systems using 16- and 64-QAM. The innovation of the proposed method in this chapter is that it can efficiently generate MD mappings using 16- and 64-QAM. Presented numerical results show that in comparison with the well-known BSA mappings and random mappings, our mappings outperform significantly over AWGN and Rayleigh fading channels. Compared to the BSA mappings for a target BER of 10^{-6}, our mappings can save up to 3.5 dB and 3 dB transmit signal energy over AWGN and Rayleigh fading channels, respectively. The corresponding performance gains are larger compared to random mappings. The proposed mappings also have improved the error-floor performance compared to random mappings and the mappings obtained by the BSA.

In Chapter 4, we have introduced a novel mapping method to construct efficient MD mappings to improve the error performance of BICM-ID over Rayleigh fading channels. We have broken the MD mapping design problem into four distinct 2-D mapping functions.

Then, we have developed cost functions, which are optimized to minimize the error-floor of MD mappings. Due to the lower complexity of 2-D space, the optimization approach is very simple and results in excellent MD mappings of higher order constellations such as 2^m-QAM for $m = 7, 8, 9, 10$. However, the well-known BSA becomes intractable in finding suitable MD mappings of these large constellations. In the case of MD mapping of 2^m-QAM for $m = 4, 5, 6$, our proposed mappings outperform the BSA mappings at the BER of practical interest, i.e., 10^{-6}, by up to 2.6 dB. In addition, the proposed mappings offer lower error-floors compared to their BSA counterparts. Consequently, our proposed mappings outperform the best previously known mappings, i.e., the BSA mappings, in both low and high SNR regions.

In chapter 5, we have proposed an optimum MD mapping of 16-QAM for BICM-ID performance over the AWGN channel. Then, a transferring system is developed to construct MD mappings of higher order QAMs using the proposed MD mapping of 16-QAM. It is proven that the proposed transferring system guarantees efficient MD mappings of 2^m-QAM (unlimited value of m). Simulation results show that our proposed mappings save up to 3.6 dB of the transmit power for a target BER of 10^{-6} compared to the well-known BSA mappings. Moreover, our proposed mappings improve the error-floor of BICM-ID. Consequently, for MD mapping of medium constellations such as 64-QAM, our proposed mappings outperforms the BSA mappings, in both low and high SNR region over the AWGN channel. Moreover, our proposed method constructs efficient unlimited dimension mappings of unlimited order QAMs for AWGN channels.

6.2 Future work

For all the results and methods presented in this thesis, it is assumed that the CSI is perfectly known at the receiver side. However, for many practical applications, the CSI is partially known at the receiver. In this case, all mapping design guidelines might be subject to change. As a result, new mapping methods will be required for different modulations based on the new mapping guidelines.

This thesis addresses the mapping problem for the constellations with fixed signal points. In particular, the proposed mapping methods in this thesis mostly consider the QAM and PSK constellations. However, changing the signal points' positions in the constellations can improve the system performance significantly. Therefore, optimizing signal constellations for BICM-ID as well as finding suitable mappings for the optimized constellations are interesting problems to address in future work.

Bibliography

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

[2] A. Alvarado "Towards fully optimized BICM transmissions," Ph.D. dissertation, Dept. Sign. and syst., Chalmers Univ. of Tech., Goteborg, Sweden 2010.

[3] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, pp. 169-171, Nov. 1997.

[4] S. T. Brink, J. Speidel, and R. H. Han, "Iterative demapping for QPSK modulation," *Electron. Lett.*, vol. 34, pp. 1459-1460, Jul. 1998.

[5] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *Eur. Trans. Telecommun.*, vol. 9, pp. 155-172, Mar. 1998.

[6] X. Li, A. Chindapol, and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding and 8PSK signaling," *IEEE Trans. Commun.*, vol. 50, pp. 1250-1257, Aug. 2002.

[7] N. H. Tran and H. H. Nguyen "A novel multi-dimensional mapping of 8-PSK for BICM-ID," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 1133-1142, Mar. 2007.

[8] N. H. Tran and H. H. Nguyen, "Signal mappings of 8-ary constellations for bit interleaved coded modulation with iterative decoding," *IEEE Trans. Broadcasting*, vol. 52, pp. 92-99, Mar. 2006.

[9] F. Schreckenbach, N. Gortz, J. Hagenauer, and G. Bauch,"Optimized symbol mappings for bit-interleaved coded modulation with iterative decoding," *IEEE Global Telecommunications Conference*, vol. 6, pp. 3316-3320, Dec. 2003.

[10] N. H. Tran and H. H. Nguyen, "Improving the performance of QPSK BICM-ID by mapping on the hypercube," *IEEE Veh. Technol. Conf.*, pp. 1299-1303, Sept. 2004.

[11] J. Tan and G. L. Stüber, "Analysis and design of symbol mappers for iteratively decoded BICM," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 662-672, Mar. 2005.

[12] F. Simoens, H. Wymeersch, H. Bruneel, and M. Moeneclaey, "Multi-dimensional mapping for bit-interleaved coded modulation with BPSK/QPSK signaling," *IEEE Commun. Lett.*, vol. 9, pp. 453-455, May 2005.

[13] Y. Huang and J. A. Ritcey, "Optimal constellation labeling for iteratively decoded bit-interleaved space-time coded modulation," *IEEE Trans. Inform. Theory*, vol. 51, pp. 1865-1871, May 2005.

[14] D. Torrieri and M. C. Valenti,"Constellation labeling maps for low error floors," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 5401-5407, Dec. 2008.

[15] N. H. Tran and H. H. Nguyen "Design and performance of BICM-ID systems with hypercube constellations," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 1169-1179, May 2006.

[16] N. Gresset, J. J. Boutros, and L. Brunel, "Multidimensional mappings for iteratively decoded BICM on multiple-antenna channels," *IEEE Trans. Inform. Theory*, vol. 51, pp. 3337-3346, Sept. 2005.

[17] M. C. Valenti, R. Doppalapudi, and D. Torrieri, "A genetic algorithm for designing constellations with low error floors," *42nd Annual Conference on Information Sciences and Systems*, pp. 1155-1160, Mar. 2008.

[18] M. Samuel, M. Barsoum, and M. P. Fitz, "On the suitability of gray bit mappings to outer channel codes in iteratively decoded BICM," *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, pp. 982-985, 2009.

[19] P. K. Vitthaladevuni, M. S. Alouini, and J. C. Kieffer, "Exact BER computation for cross QAM constellations," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 3039-3050, Nov. 2005.

[20] S. P. Herath, N. H. Tran, and T. Le-Ngoc, "Rotated multi-D constellations in Rayleigh fading: mutual information improvement and pragmatic approach for near-capacity performance in high-rate regions," *IEEE Trans. Commun.*, vol. 60, pp. 3694-3704, Dec. 2012.

[21] H. M. Navazi and Md. J. Hossain, "Efficient multi-dimensional mapping using QAM constellations for BICM-ID," *IEEE Trans. Wireless Commun.*, accepted, Sept. 2017.

[22] H. M. Navazi and H. H. Nguyen, "A novel and efficient mapping of 32-QAM constellation for BICM-ID systems," *Springer, Wireless Personal Communications*, Vol. 79, pp. 197-210, Jun. 2014.

[23] H. M. Navazi and Md. J. Hossain, "A novel symbol mapping method for BICM-ID systems for higher order signal constellations," *IEEE Commun. Lett.*, Vol. 18, pp. 1323-1326, Aug. 2014.

[24] T. Koike-Akino and V. Tarokh, "Sphere packing optimization and EXIT chart analysis for multi-dimensional QAM signaling," *IEEE International Conference on Communications, Dresden, 2009*, pp. 1-5.

[25] M. Lamarca, H. Lou, and J. Garcia-Frias, "Random labeling: A new approach to achieve capacity in MIMO quasi-static fading channels," *IEEE International Symposium on Information Theory, Seattle, WA, 2006*, pp. 1959-1963.

[26] D. Divsalar and M. K. Simon, "The design of trellis coded modulation for MPSK for fading channels: performance criteria," *IEEE Trans. Commun.*, vol. 36, pp. 1004-1012, Sept. 1988.

[27] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," *in Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, pp.1064-1070, May 1993.

[28] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927-946, May 1998.

[29] P. K. Vitthaladevuni and M.-S. Alouini, "A closed-form expression for the exact BER of generalized PAM and QAM constellations," *IEEE Trans. Commun.*, vol. 52, pp. 698-700, May 2004.

[30] P. K. Vitthaladevuni and M.-S. Alouini, "Exact BER computation of generalized hierarchical PSK constellations," *IEEE Trans. Commun.*, vol. 51, pp. 2030-2037, Dec. 2003.

[31] J. G. Smith, "Odd-bit quadrature amplitude shift keying," *IEEE Trans. Commun.*, vol. COMM-23, no. 3, pp. 385-389, Mar. 1975.

[32] H. M. Navazi and Md. J. Hossain, "A novel symbol mapping method for BICM-ID systems for higher order signal constellations," [Online]. Available: https://www.dropbox.com/s/8xl1al1nmaxmleo/Report.pdf.

[33] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, pp. 56-67, Jan. 1982.

[34] A. Chindapol and J. A. Ritcey, "Design, analysis and performance evaluation for BICMID with square QAM constellations in Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 944-957, May 2001.

Appendices

Appendix A

Proof for Proposition 2.3

Proof. We prove Proposition 2.3 for two groups of modulations as follows: (i) PSK and square QAM and (ii) cross QAM.

PSK and square QAM: Perfect Gray mappings can be obtained from the natural binary labeling for PSK and square QAM constellations using the simple procedure described in [29] and [30]. It is well-known that a given symbol in a Gray mapping has the Hamming distance of one bit from each of its adjacent symbols.

If $\boldsymbol{l}_k = [l_k^1, l_k^2, \cdots, l_k^m]$ and $\boldsymbol{l}_n = [l_n^1, l_n^2, \cdots, l_n^m]$ are two m-bit labels, the proposed mapping method maps $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ to the symbols whose labels in the corresponding Gray mapping are $\hat{\boldsymbol{l}}_k = [\hat{l}_k^1, \hat{l}_k^2, \cdots, \hat{l}_k^m]$ and $\hat{\boldsymbol{l}}_n = [\hat{l}_l^n, \hat{l}_n^2, \cdots, \hat{l}_n^m]$, respectively. Thus, if $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ are adjacent symbols in the resulting mapping, $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are adjacent symbols in the Gray mapping and have the Hamming distance of one bit from each other. According to Proposition 2.2 (c.f., eq. (2.12)), the original label $\boldsymbol{l}_t$ can be obtained uniquely from its precoded version $\hat{\boldsymbol{l}}_t$ using the reverse process, i.e., $\boldsymbol{l}_t = \Psi^{-1}(\hat{\boldsymbol{l}}_t)$. Hence, in order to prove that a given symbol in the resulting mapping, e.g., $\boldsymbol{l}_k$, has the Hamming distance of either two, $(m-1)$ or m bits from its neighbour, e.g., $\boldsymbol{l}_n$, we need to prove the following. If $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have the Hamming distance of one bit from each other, $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ should have the Hamming distance of either two, $(m-1)$ or m bits from each other. Using eq. (2.12), we can rewrite $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ as follows:

$$l_k^i = \begin{cases} \hat{l}_k^i \oplus \hat{l}_k^q & \text{if } i \neq q, \\ W(\hat{\boldsymbol{l}}_k) \oplus A(m) \times \hat{l}_k^q & \text{otherwise,} \end{cases} \tag{A.1}$$

$$l_n^i = \begin{cases} \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, \\ W(\hat{\boldsymbol{l}}_n) \oplus A(m) \times \hat{l}_n^q & \text{otherwise.} \end{cases} \tag{A.2}$$

Since both of $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have the same length, $A(m)$ takes the same value in eq. (A.1) and eq. (A.2). Assume that $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ only at the j^{th} bit position and chosen-index is equal to q. There are two cases as follows:

Case 1: $j = q$. Since $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ differ in the j^{th} bit position and when $j = q$, we can write

$$\begin{cases} \hat{l}_k^i = \bar{\hat{l}}_n^i & \text{if } i = q, \\ \hat{l}_k^i = \hat{l}_n^i & \text{otherwise,} \end{cases} \tag{A.3}$$

then if $\hat{l}_k^q = B$, using eq. (A.3) we have $\hat{l}_n^q = \bar{\hat{l}}_k^q = \bar{B}$. Since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ in one bit position, one of them has an odd Hamming wight and the other has an even Hamming wight. As a result, if $W(\hat{\boldsymbol{l}}_k) = C$, $W(\hat{\boldsymbol{l}}_n) = \bar{C}$. Hence, when $i \neq q$, by replacing $\hat{l}_k^i$ by $\hat{l}_n^i$, $\hat{l}_k^q$ by B, $W(\hat{\boldsymbol{l}}_k)$ by C, $\hat{l}_n^q$ by $\bar{B}$, and $W(\hat{\boldsymbol{l}}_n)$ by $\bar{C}$ we can rewrite eq. (A.1) and eq. (A.2) as follows:

$$l_k^i = \begin{cases} \hat{l}_n^i \oplus B & \text{if } i \neq q, \\ C \oplus A(m) \times B & \text{otherwise,} \end{cases} \tag{A.4}$$

$$l_n^i = \begin{cases} \hat{l}_n^i \oplus \bar{B} & \text{if } i \neq q, \\ \bar{C} \oplus A(m) \times \bar{B} & \text{otherwise.} \end{cases} \tag{A.5}$$

Let us assume that $\boldsymbol{l}_x = [l_x^1, l_x^2, \cdots, l_x^m]$ and $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$, i.e., $l_x^i = l_k^i \oplus l_n^i$ for all i. Using eq. (A.4) and eq. (A.5), we can write

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} \hat{l}_n^i \oplus B \oplus \hat{l}_n^i \oplus \bar{B} & \text{if } i \neq q, \\ C \oplus (A(m) \times B) \oplus \bar{C} \oplus (A(m) \times \bar{B}) & \text{otherwise.} \end{cases} \tag{A.6}$$

Using the fact $\hat{l}_n^i \oplus B \oplus \hat{l}_n^i \oplus \bar{B} = (\hat{l}_n^i \oplus \hat{l}_n^i) \oplus (B \oplus \bar{B}) = 0 \oplus 1 = 1$ and $C \oplus (A(m) \times B) \oplus \bar{C} \oplus (A(m) \times \bar{B}) = (C \oplus \bar{C}) \oplus A(m) \times (B \oplus \bar{B}) = (1 \oplus A(m)) = \bar{A}(m)$, we can rewrite eq. (A.6) as

$$l_x^i = \begin{cases} 1 & \text{if } i \neq q, \\ \bar{A}(m) & \text{otherwise.} \end{cases} \tag{A.7}$$

If m is odd, $\bar{A}(m) = 1$ (c.f., Proposition 2.2) and $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ has a Hamming weight of m, which implies that the labels of two adjacent symbols, $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$, in the resulting mapping are different in all m bit positions. On the other hand, if m is even, $\bar{A}(m) = 0$ (c.f., Proposition 2.2) and $\boldsymbol{l}_x$ has the Hamming weight of $(m-1)$. This implies that $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ are different in $(m-1)$ bits.

Case 2: $j \neq q$. In this case using eq. (2.1), we can rewrite eq. (A.1) and eq. (A.2) as follows:

$$l_k^i = \begin{cases} \hat{l}_k^j \oplus \hat{l}_k^q & \text{if } i \neq q, i = j, \\ \hat{l}_k^i \oplus \hat{l}_k^q & \text{if } i \neq q, i \neq j, \\ W(\hat{\boldsymbol{l}_k}) \oplus A(m) \times \hat{l}_k^q & \text{if } i = q, \end{cases} \tag{A.8}$$

$$l_n^i = \begin{cases} \hat{l}_n^j \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq j \\ W(\hat{\boldsymbol{l}_n}) \oplus A(m) \times \hat{l}_n^q & \text{if } i = q. \end{cases} \tag{A.9}$$

Since $\hat{\boldsymbol{l}_k}$ and $\hat{\boldsymbol{l}_n}$ differ only in the j^{th} bit position, if $\hat{l}_k^j = B$, we can write $\hat{l}_l^j = \bar{\hat{l}}_k^j = \bar{B}$. When $i \neq j$, we have $\hat{l}_k^i = \hat{l}_n^i$, which yields $\hat{l}_k^q = \hat{l}_n^q$ as well. Moreover, Since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ in one bit position, one of them has an odd Hamming weight and the other one has an even Hamming weight. As a result, if $W(\hat{\boldsymbol{l}}_k) = C$, we can write $W(\hat{\boldsymbol{l}}_n) = \bar{C}$. Thus, by replacing $\hat{l}_k^j$ by B, $\hat{l}_n^j$ by $\bar{B}$, $\hat{l}_k^i$ by $\hat{l}_n^i$, $\hat{l}_k^q$ by $\hat{l}_n^q$, $W(\hat{\boldsymbol{l}}_k)$ by C and $W(\hat{\boldsymbol{l}}_n)$ by $\bar{C}$, we can rewrite eq. (A.8) and eq. (A.9) as follows:

$$l_k^i = \begin{cases} B \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq j, \\ C \oplus A(m) \times \hat{l}_n^q & \text{if } i = q, \end{cases} \tag{A.10}$$

$$l_n^i = \begin{cases} \bar{B} \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq j, \\ \bar{C} \oplus A(m) \times \hat{l}_n^q & \text{if } i = q. \end{cases} \tag{A.11}$$

Let us assume that $\boldsymbol{l}_x = [l_x^1, l_x^2, \cdots, l_x^m]$ and $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$, i.e., $l_x^i = l_k^i \oplus l_n^i$ for all i. As a result, using eq. (A.10) and eq. (A.11) we can write

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} B \oplus \hat{l}_n^q \oplus \bar{B} \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q \oplus \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq j, \\ C \oplus (A(m) \times \hat{l}_n^q) \oplus \bar{C} \oplus (A(m) \times \hat{l}_n^q) & \text{if } i = q. \end{cases} \tag{A.12}$$

Using the facts $B \oplus \hat{l}_n^q \oplus \bar{B} \oplus \hat{l}_n^q = (B \oplus \bar{B}) \oplus (\hat{l}_n^q \oplus \hat{l}_n^q) = 1 \oplus 0 = 1$, $\hat{l}_n^i \oplus \hat{l}_n^q \oplus \hat{l}_n^i \oplus \hat{l}_n^q = (\hat{l}_n^i \oplus \hat{l}_n^i) \oplus (\hat{l}_n^q \oplus \hat{l}_n^q) = 0 \oplus 0 = 0$ and $C \oplus (A(m) \times \hat{l}_n^q) \oplus \bar{C} \oplus (A(m) \times \hat{l}_n^q) = (C \oplus \bar{C}) \oplus A(m) \times (\hat{l}_n^q \oplus \hat{l}_n^q) = 1 \oplus (A(m) \times 0) = 1$, we rewrite eq. (A.12) as

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} 1 & \text{if } i \neq q, i = j, \\ 0 & \text{if } i \neq q, i \neq j, \\ 1 & \text{if } i = q. \end{cases} \tag{A.13}$$

Thus, the Hamming weight of $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ is two, which implies that the labels of two adjacent symbols in the resulting mapping, i.e., $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$, are different in two bit positions.

In summary, if $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ are the labels of two adjacent symbols in the resulting square QAM or PSK constellations with our proposed mapping and $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are different in one bit position, say the j^{th} bit position, the Hamming distance between $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ denoted by $d_{\mathrm{H}}(\boldsymbol{l}_k, \boldsymbol{l}_n)$ is given by (from the above discussions):

$$d_{\mathrm{H}}(\boldsymbol{l}_k, \boldsymbol{l}_n) = \begin{cases} m & \text{if } j = q, \text{ and } m \text{ is odd,} \\ m-1 & \text{if } j = q, \text{ and } m \text{ is even,} \\ 2 & \text{if } j \neq q, \end{cases} \tag{A.14}$$

where q is the chosen-index, and $j = 1, 2, \cdots, m$, is the bit position in which the labels of two adjacent symbols in the Gray mapping are different from each other. Let us assume that the chosen-index is equal to q. Then, if $j = q$, the Hamming distance between two adjacent symbols in our resulting PSK and square QAM mappings is $d_{\mathrm{H}}(\boldsymbol{l}_k, \boldsymbol{l}_n) = m$ or $(m-1)$ (c.f., eq. (A.14)). However, for a given value of q, the fraction of the time that we get $j = q$ for two adjacent symbols is less than $\frac{1}{m}$. Therefore, the fraction of adjacent symbols that will have the Hamming distance of m or $(m-1)$ bits is less than $\frac{1}{m}$, which tends to be smaller for higher order constellations. Also, the fraction of adjacent symbols that will have the Hamming distance of two bits from each other is larger than $(\frac{m-1}{m})$, which tends to be larger for higher order constellations. Therefore, our proposed mapping method yields a smaller average Hamming distance between adjacent symbols in the resulting PSK and square QAM mappings especially for larger constellations as shown in Table A.1.

Table A.1: Hamming distance between adjacent symbols for proposed mappings.

Modulation	m	Total number of adjacent symbols	Percentage of adjacent symbols with Hamming distance of 2 bits	Percentage of adjacent symbols with Hamming distance of $(m-1)$ bits	Percentage of adjacent symbols with Hamming distance of m bits	Average Hamming distance between adjacent symbols
16-QAM	4	24	83.33%	16.67%	0%	2.17
32-QAM	5	52	88.46%	0%	11.54%	2.35
64-QAM	6	112	92.86%	7.14%	0%	2.21
128-QAM	7	232	94.83%	0%	5.17%	2.26
256-QAM	8	480	96.67%	3.33%	0%	2.17
512-QAM	9	976	97.54%	0%	2.46%	2.17
1024-QAM	10	1984	98.39%	1.61%	0%	2.11
16-PSK	4	16	87.50%	12.5%	0%	2.13
32-PSK	5	32	93.75%	0%	6.25%	2.19
64-PSK	6	64	96.87%	3.13%	0%	2.09
128-PSK	7	128	98.44%	0%	1.56%	2.08
256-PSK	8	256	99.22%	0.78%	0%	2.04
512-PSK	9	512	99.61%	0%	0.39%	2.03
1024-PSK	10	1024	99.80%	0.20%	0%	2.01

Cross QAM: Perfect Gray mappings cannot be defined for cross QAM constellations. However, pseudo-Gray mappings can be obtained for such constellations using the procedure described in [29] and [31]. It is worth noting that the Hamming distance between two adjacent symbols in a pseudo-Gray mapping of a cross QAM constellation is at most 2 bits.

Let $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ be the labels of two adjacent symbols in our proposed mapping for a cross QAM. Thus, $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are the labels of the adjacent symbols in the corresponding pseudo-Gray mapping of the constellation and have the Hamming distance of either one or two bits from each other. In order to prove that the label of a symbol in the proposed mapping e.g., $\boldsymbol{l}_k$, has the Hamming distance of either two, $(m-1)$ or m bits from the label of its neighbouring symbols, e.g., $\boldsymbol{l}_n$, it suffices to prove the following. If $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have the Hamming distance of one or two bits from each other, then the Hamming distance between $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ is either two, $(m-1)$, or m bits. It is already proven that if $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have a Hamming distance of one bit, $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ have a Hamming distance of either two, $(m-1)$, or m from each other. As a result, we need to prove that, if $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ have the Hamming distance of two bits from each other, then $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ have the Hamming distance of either two, $(m-1)$, or m bits from each other. This proof is given below.

Assume that $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are different in two bit positions, i.e., the h^{th} and the j^{th} bit positions, and the chosen-index is equal to q. Two cases can be discussed as follows.
Case 1: $h \neq q$ and $j \neq q$. In this case, we can rewrite eq. (A.1) and eq. (A.2) as follows:

$$l_k^i = \begin{cases} \hat{l}_k^h \oplus \hat{l}_k^q & \text{if } i \neq q, i = h, \\ \hat{l}_k^j \oplus \hat{l}_k^q & \text{if } i \neq q, i = j, \\ \hat{l}_k^i \oplus \hat{l}_k^q & \text{if } i \neq q, i \neq h, i \neq j, \\ W(\hat{\boldsymbol{l}_k}) \oplus A(m) \times \hat{l}_k^q & \text{if } i = q, \end{cases} \tag{A.15}$$

$$l_l^i = \begin{cases} \hat{l}_n^h \oplus \hat{l}_n^q & \text{if } i \neq q, i = h, \\ \hat{l}_n^j \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq h, i \neq j, \\ W(\hat{\boldsymbol{l}_n}) \oplus A(m) \times \hat{l}_n^q & \text{if } i = q. \end{cases} \tag{A.16}$$

Since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are different in the h^{th} and the j^{th} bit positions, if $\hat{l}_k^h = B$, then $\hat{l}_n^h = \bar{\hat{l}}_k^h = \bar{B}$, and if $\hat{l}_k^j = C$, then $\hat{l}_n^j = \bar{\hat{l}}_k^j = \bar{C}$. Also if $i \neq h$ and $i \neq j$, then $\hat{l}_k^i = \hat{l}_n^i$, which results in $\hat{l}_k^q = \hat{l}_n^q$. Moreover, since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ in two bit positions, both of them has either an odd Hamming weight or an even Hamming weight. As a result, if $W(\hat{\boldsymbol{l}}_k) = D$, then $W(\hat{\boldsymbol{l}}_n) = D$ as well. Thus, in eq. (A.15) and eq. (A.16), by replacing $\hat{l}_k^h$ by B, $\hat{l}_n^h$ by $\bar{B}$, $\hat{l}_k^j$ by C, $\hat{l}_n^j$ by $\bar{C}$, $\hat{l}_k^i$ by $\hat{l}_n^i$, $\hat{l}_k^q$ by $\hat{l}_n^q$, $W(\hat{\boldsymbol{l}}_k)$ by D, and $W(\hat{\boldsymbol{l}}_n)$ by D, we can rewrite eq. (A.15) and eq. (A.16) as follows:

$$l_k^i = \begin{cases} B \oplus \hat{l}_n^q & \text{if } i \neq q, i = h, \\ C \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq h, i \neq j, \\ D \oplus A(m) \times \hat{l}_n^q & \text{if } i = q, \end{cases} \tag{A.17}$$

$$l_n^i = \begin{cases} \bar{B} \oplus \hat{l}_n^q & \text{if } i \neq q, i = h, \\ \bar{C} \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq h, i \neq j, \\ D \oplus A(m) \times \hat{l}_n^q & \text{if } i = q. \end{cases} \tag{A.18}$$

Let us assume that $\boldsymbol{l}_x = [l_x^1, l_x^2, \cdots, l_x^m]$ and $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ so that $\forall i;\ l_x^i = l_k^i \oplus l_n^i$. As a result, using eq. (A.17) and eq. (A.18) we have

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} B \oplus \hat{l}_n^q \oplus \bar{B} \oplus \hat{l}_n^q & \text{if } i \neq q, i = h, \\ C \oplus \hat{l}_n^q \oplus \bar{C} \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q \oplus \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq h, i \neq j, \\ D \oplus (A(m) \times \hat{l}_n^q) \oplus D \oplus (A(m) \times \hat{l}_l^q) & \text{if } i = q. \end{cases} \tag{A.19}$$

In eq. (A.19), $B \oplus \hat{l}_n^q \oplus \bar{B} \oplus \hat{l}_n^q$ results in $(B \oplus \bar{B}) \oplus (\hat{l}_n^q \oplus \hat{l}_n^q)$, which equals to $1 \oplus 0 = 1$; $C \oplus \hat{l}_n^q \oplus \bar{C} \oplus \hat{l}_n^q$ results in $(C \oplus \bar{C}) \oplus (\hat{l}_n^q \oplus \hat{l}_n^q)$, which equals to $1 \oplus 0 = 1$; $\hat{l}_n^i \oplus \hat{l}_n^q \oplus \hat{l}_n^i \oplus \hat{l}_n^q$ results in $(\hat{l}_n^i \oplus \hat{l}_n^i) \oplus (\hat{l}_n^q \oplus \hat{l}_n^q)$, which equals to $0 \oplus 0 = 0$. Moreover, $D \oplus (A(m) \times \hat{l}_n^q) \oplus D \oplus (A(m) \times \hat{l}_n^q)$ results in $(D \oplus D) \oplus A(m) \times (\hat{l}_n^q \oplus \hat{l}_n^q)$, which equals to $0 \oplus A(m) \times 0 = 0$. Consequently, we can rewrite eq. (A.19) as

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} 1 & \text{if } i \neq q, i = h, \\ 1 & \text{if } i \neq q, i = j, \\ 0 & \text{if } i \neq q, i \neq h, i \neq j, \\ 0 & \text{if } i = q. \end{cases} \tag{A.20}$$

Thus, the Hamming weight of $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ is two, which implies that in this case, the labels of two adjacent symbols in the proposed mapping, i.e., $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$, are different only in two bit positions.

Case 2: One of h or j is equal to q. Without losing generality, we assume that $h = q$

and $j \neq q$[4]. In this case, we can rewrite eq. (A.1) and eq. (A.2) as follows:

$$l_k^i = \begin{cases} \hat{l}_k^j \oplus \hat{l}_k^q & \text{if } i \neq q, i = j, \\ \hat{l}_k^i \oplus \hat{l}_k^q & \text{if } i \neq q, i \neq j \\ W(\hat{\boldsymbol{l}}_k) \oplus A(m) \times \hat{l}_k^q & \text{if } i = q, \end{cases} \tag{A.21}$$

$$l_n^i = \begin{cases} \hat{l}_n^j \oplus \hat{l}_n^q & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \hat{l}_n^q & \text{if } i \neq q, i \neq j \\ W(\hat{\boldsymbol{l}_n}) \oplus A(m) \times \hat{l}_n^q & \text{if } i = q. \end{cases} \tag{A.22}$$

Since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ in the q^{th} and the j^{th} bit positions, if $\hat{l}_k^q = B$, then $\hat{l}_n^q = \bar{\hat{l}}_k^q = \bar{B}$, and if $\hat{l}_k^j = C$, then $\hat{l}_n^j = \bar{\hat{l}}_k^j = \bar{C}$. Also, if $i \neq q$ and $i \neq j$, then $\hat{l}_k^i = \hat{l}_n^i$. Moreover, since $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ differ in two bit positions, then both of them has either an odd Hamming weight or an even Hamming weight. As a result, if $W(\hat{\boldsymbol{l}}_k) = D$, then $W(\hat{\boldsymbol{l}}_n) = D$ as well. Hence, by replacing $\hat{l}_k^q$ by B, $\hat{l}_n^q$ by $\bar{B}$, $\hat{l}_k^j$ by C, $\hat{l}_n^j$ by $\bar{C}$, $\hat{l}_k^i$ by $\hat{l}_n^i$, $W(\hat{\boldsymbol{l}}_k)$ by D, and $W(\hat{\boldsymbol{l}}_n)$ by D, we can rewrite eq. (A.21) and eq. (A.22) as follows:

$$l_k^i = \begin{cases} C \oplus B & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus B & \text{if } i \neq q, i \neq j \\ D \oplus A(m) \times B & \text{if } i = q, \end{cases} \tag{A.23}$$

$$l_n^i = \begin{cases} \bar{C} \oplus \bar{B} & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus \bar{B} & \text{if } i \neq q, i \neq j \\ D \oplus A(m) \times \bar{B} & \text{if } i = q. \end{cases} \tag{A.24}$$

Suppose that $\boldsymbol{l}_x = [l_x^1, l_x^2, \cdots, l_x^m]$ and $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ such that $\forall i;\ l_x^i = l_k^i \oplus l_n^i$. As a result, using eq. (A.23) and eq. (A.24) we can write

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} C \oplus B \oplus \bar{C} \oplus \bar{B} & \text{if } i \neq q, i = j, \\ \hat{l}_n^i \oplus B \oplus \hat{l}_n^i \oplus \bar{B} & \text{if } i \neq q, i \neq j, \\ D \oplus (A(m) \times B) \oplus D \oplus (A(m) \times \bar{B}) & \text{if } i = q. \end{cases} \tag{A.25}$$

In eq. (A.25), $C \oplus B \oplus \bar{C} \oplus \bar{B}$ results in $(B \oplus \bar{B}) \oplus (C \oplus \bar{C})$, which equals $1 \oplus 1 = 0$; $\hat{l}_n^i \oplus B \oplus \hat{l}_n^i \oplus \bar{B}$ results in $(\hat{l}_n^i \oplus \hat{k}_n^i) \oplus (B \oplus \bar{B})$, which equals $0 \oplus 1 = 1$; $D \oplus (A(m) \times B) \oplus D \oplus (A(m) \times \bar{B})$ results in $(D \oplus D) \oplus A(m) \times (B \oplus \bar{B})$, which equals to $0 \oplus (A(m) \times 1) = A(m)$. It is worth noting that since m is odd for non-square QAM, then $A(m)$ is equal to 0.

[4] It is important to note that it has no importance that which of h or j is equal to q. Both of them give exactly the same result.

Consequently, we can rewrite eq. (A.25) as

$$l_x^i = l_k^i \oplus l_n^i = \begin{cases} 0 & \text{if } i \neq q, i = j, \\ 1 & \text{if } i \neq q, i \neq j, \\ 0 & \text{if } i = q. \end{cases} \tag{A.26}$$

As a result, the Hamming weight of $\boldsymbol{l}_x = \boldsymbol{l}_k \oplus \boldsymbol{l}_n$ is $m-2$; in this case, the labels of two adjacent symbols in the proposed mapping, i.e., $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$, are different only in $(m-2)$ bit positions. However, in a pseudo-Gray mapping of a cross QAM constellation, if $\mathbb{S}$ is the set of all possible values of j and h, then the cardinality of the set $\mathbb{S}$ is equal to 3. Therefore, by choosing q (chosen-index) such that $q \neq j$ and $q \neq h$, there will not be any Hamming distance of $(m-2)$ bits between two adjacent symbols in the proposed mappings of a cross-QAM constellation. As a result, all two bit Hamming distances between $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ in the pseudo-Gray mapping result in a two bit Hamming distances between $\boldsymbol{l}_k$ and $\boldsymbol{l}_n$ in the proposed mappings. However, using eq (A.14), when the Hamming distance between $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ in the pseudo-Gray mapping is one bit, there will be a Hamming distance of m bits between two adjacent symbols in the proposed mappings. This happens only when $\hat{\boldsymbol{l}}_k$ and $\hat{\boldsymbol{l}}_n$ are different at the q^{th} bit position. As a result, for the proposed cross QAM mappings, the fraction of adjacent symbols with Hamming distance m can be at most $\frac{1}{m}$, and consequently, the fraction of adjacent symbols with Hamming distance two is larger that $(\frac{m-1}{m})$. □

Appendix B

Proof for Proposition 5.6

Let us define $\boldsymbol{l}_i = [\boldsymbol{e}_i, \boldsymbol{l}_{i-1}]$, where $\boldsymbol{e}_i = [e_i^1, \cdots, e_i^N]$, in which $e_i^j = l_i^j$, and $\boldsymbol{l}_{i-1} = [l_{i-1}^1, \cdots, l_{i-1}^{(i+2)N}]$, in which $l_{i-1}^j = l_i^{N+j}$. Similarly, we define $\boldsymbol{l}_{i,k} = [\boldsymbol{e}_{i,k}, \boldsymbol{l}_{i-1,k}]$, where $\boldsymbol{e}_{i,k} = [e_{i,k}^1, \cdots, e_{i,k}^N]$, in which $e_{i,k}^j = l_{i,k}^j$, and $\boldsymbol{l}_{i-1,k} = [l_{i-1,k}^1, \cdots, l_{i-1,k}^{(i+2)N}]$, in which $l_{i-1,k}^j = l_{i,k}^{N+j}$. Suppose that in step $i-1$, $\boldsymbol{l}_{i-1}$ and $\boldsymbol{l}_{i-1,k}$ are mapped to $\boldsymbol{s}_{i-1} = [s_{i-1}^1, \cdots, s_{i-1}^N]$ and $\boldsymbol{s}_{i-1,k} = [s_{i-1,k}^1, \cdots, s_{i-1,k}^N]$, respectively. We group all possible cases for $\boldsymbol{l}_i$ and $\boldsymbol{l}_{i,k}$ in 10 cases (B.1) and investigate them in the following. When $k \leq N$ (*Case 1* to *Case 7*), $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i,k}$ are different only in the k^{th} bit position. Without loss of generality, it is assumed that in *Case 1-7*, the Hamming weight of $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i,k}$ is even and odd, respectively.

$$\begin{cases} k \leqslant N \begin{cases} j \neq k \begin{cases} e_i^j = e_{i,k}^j = 0; \quad Case\ 1 \\ e_i^j = e_{i,k}^j = 1 \begin{cases} i \in \mathbb{E}; \quad Case\ 2 \\ i \in \mathbb{O}; \quad Case\ 3 \end{cases} \end{cases} \\ j = k \begin{cases} e_i^j = \bar{e}_{i,k}^j = 0 \begin{cases} i \in \mathbb{E}; \quad Case\ 4 \\ i \in \mathbb{O}; \quad Case\ 5 \end{cases} \\ e_i^j = \bar{e}_{i,k}^j = 1 \begin{cases} i \in \mathbb{E}; \quad Case\ 6 \\ i \in \mathbb{O}; \quad Case\ 7 \end{cases} \end{cases} \end{cases} \\ k > N \begin{cases} w_i \in \mathbb{E}; \quad Case\ 8 \\ w_i \in \mathbb{O} \begin{cases} sgn(x_{i-1}^j) = sgn(x_{i-1,k}^j); \quad Case\ 9 \\ sgn(x_{i-1}^j) = -sgn(x_{i-1,k}^j); \quad Case\ 10 \end{cases} \end{cases} \end{cases} \tag{B.1}$$

Case 1: $k \leqslant N$, $j \neq k$, $e_i^j = e_{i,k}^j = 0$. Using Table 5.4 we can write

$$\begin{aligned} x_i^j - x_{i,k}^j &= sgn(x_{i-1,k}^j) 2^{m - \lfloor \frac{m}{2} \rfloor - 1} \\ y_i^j - y_{i,k}^j &= sgn(y_{i-1,k}^j) 2^{\lfloor \frac{m}{2} \rfloor - 1}, \end{aligned} \tag{B.2}$$

and as a result,

$$\begin{aligned}|s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &= 2^{2m-2\lfloor\frac{m}{2}\rfloor-2} + 2^{2\lfloor\frac{m}{2}\rfloor-2} \\ &= \hat{d}_5^2.\end{aligned} \tag{B.3}$$

Case 2: $k \leqslant N$, $j \neq k$, $e_i^j = e_{i,k}^j = 1$, $i \in \mathbb{E}$. Using Table 5.4 we can write

$$\begin{aligned}x_i^j - x_{i,k}^j &= sgn(x_{i-1,k}^j)2^{m-\lfloor\frac{m}{2}\rfloor-1} \\ y_i^j - y_{i,k}^j &= sgn(y_{i-1,k}^j)2^{\lfloor\frac{m}{2}\rfloor-1},\end{aligned} \tag{B.4}$$

and as a result,

$$\begin{aligned}|s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &= 2^{2m-2\lfloor\frac{m}{2}\rfloor-2} + 2^{2\lfloor\frac{m}{2}\rfloor-2} \\ &= \hat{d}_5^2.\end{aligned} \tag{B.5}$$

Case 3: $k \leqslant N$, $j \neq k$, $e_i^j = e_{i,k}^j = 1$, $i \in \mathbb{O}$. Using Table 5.4, we can write

$$\begin{aligned}x_i^j - x_{i,k}^j &= sgn(x_{i-1,k}^j)2^{m-\lfloor\frac{m}{2}\rfloor-1} \\ y_i^j - y_{i,k}^j &= sgn(y_{i-1,k}^j)2^{\lfloor\frac{m}{2}\rfloor-1},\end{aligned} \tag{B.6}$$

and as a result,

$$\begin{aligned}|s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &= 2^{2m-2\lfloor\frac{m}{2}\rfloor-2} + 2^{2\lfloor\frac{m}{2}\rfloor-2} \\ &= \hat{d}_5^2.\end{aligned} \tag{B.7}$$

Case 4: $k \leqslant N$, $j = k$, $e_i^j = \bar{e}_{i,k}^j = 0$, $i \in \mathbb{E}$. Using Table 5.4, we can write

$$\begin{aligned}x_i^j - x_{i,k}^j &= (sgn(x_{i-1,k}^j) - 2^{-\lfloor\frac{j}{2}\rfloor-1})2^{m-\lfloor\frac{m}{2}\rfloor-1} \\ y_i^j - y_{i,k}^j &= sgn(y_{i-1}^j)2^{\lfloor\frac{m}{2}\rfloor-1}.\end{aligned} \tag{B.8}$$

Since $|sgn(x_{i-1,k}^j) - 2^{-\lfloor\frac{j}{2}\rfloor-1}| \geq 2^{-1}$, then

$$\begin{aligned} |s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &\geqslant 2^{2m-2\lfloor \frac{m}{2} \rfloor - 4} + 2^{2\lfloor \frac{m}{2} \rfloor - 2} \\ &= \hat{d}_4^2. \end{aligned} \tag{B.9}$$

Case 5: $k \leqslant N$, $j = k$, $e_i^j = \bar{e}_{i,k}^j = 0$, $i \in \mathbb{O}$. Using Table 5.4, we can write

$$\begin{aligned} x_i^j - x_{i,k}^j &= sgn(x_{i-1,k}^j) 2^{m-\lfloor \frac{m}{2} \rfloor - 1} \\ y_i^j - y_{i,k}^j &= 2^{\lfloor \frac{m}{2} \rfloor - 1}(2^{-\lfloor \frac{j-1}{2} \rfloor - 1} + sgn(y_{i-1,k}^j)), \end{aligned} \tag{B.10}$$

Since $|2^{-\lfloor \frac{j-1}{2} \rfloor - 1} + sgn(y_{i-1,k}^j)| \geq 2^{-1}$, then

$$\begin{aligned} |s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &\geqslant 2^{2m-2\lfloor \frac{m}{2} \rfloor - 2} + 2^{2\lfloor \frac{m}{2} \rfloor - 4} \\ &= \hat{d}_4^2. \end{aligned} \tag{B.11}$$

Case 6: $k \leqslant N$, $j = k$, $e_i^j = \bar{e}_{i,k}^j = 1$, $i \in \mathbb{E}$. Using Table 5.4, we can write

$$\begin{aligned} x_i^j - x_{i,k}^j &= 2^{m-\lfloor \frac{m}{2} \rfloor - 1}(2^{-\lfloor \frac{j}{2} \rfloor - 1} + sgn(x_{i-1}^k)) \\ y_i^j - y_{i,k}^j &= sgn(y_{i-1,k}^j) 2^{\lfloor \frac{m}{2} \rfloor - 1}. \end{aligned} \tag{B.12}$$

Since $|2^{-\lfloor \frac{j}{2} \rfloor - 1} + sgn(x_{i-1}^j)| \geq 2^{-1}$, then

$$\begin{aligned} |s_i^j - s_{i,k}^j|^2 &= |x_i^j - x_{i,k}^j|^2 + |y_i^j - y_{i,k}^j|^2 \\ &\geqslant 2^{2m-2\lfloor \frac{m}{2} \rfloor - 4} + 2^{2\lfloor \frac{m}{2} \rfloor - 2} \\ &= \hat{d}_4^2. \end{aligned} \tag{B.13}$$

Case 7: $k \leqslant N$, $j = k$, $e_i^j = \bar{e}_{i,k}^j = 1$, $i \in \mathbb{O}$. Using Table 5.4 we can write

$$\begin{aligned} x_i^j - x_{i,k}^j &= sgn(x_{i-1,k}^j) 2^{m-\lfloor \frac{m}{2} \rfloor - 1} \\ y_i^j - y_{i,k}^j &= 2^{\lfloor \frac{m}{2} \rfloor - 1}(-2^{-\lfloor \frac{j-1}{2} \rfloor - 1} + sgn(y_{i-1,k}^j)). \end{aligned} \tag{B.14}$$

Since $|-2^{-\lfloor\frac{j-1}{2}\rfloor-1}+sgn(y^j_{i-1,k})|\geq 2^{-1}$, then

$$\begin{aligned} |s^j_i - s^j_{i,k}|^2 &= |x^j_i - x^j_{i,k}|^2 + |y^j_i - y^j_{i,k}|^2 \\ &\geqslant 2^{2m-2\lfloor\frac{m}{2}\rfloor-2} + 2^{2\lfloor\frac{m}{2}\rfloor-4} \\ &= \hat{d}^2_4. \end{aligned} \tag{B.15}$$

From *Case 1* to *Case 7*, we conclude that

$$\begin{cases} |s^j_i - s^j_{i,k}|^2 = \hat{d}^2_5 & \text{if } j \neq k, k \leqslant N \\ |s^j_i - s^j_{i,k}|^2 \geqslant \hat{d}^2_4 & \text{if } j = k, k \leqslant N, \end{cases} \tag{B.16}$$

which results in

$$\begin{aligned} \|\boldsymbol{s}_i - \boldsymbol{s}_{i,k}\|^2 &= \sum_{j=1}^{N} \|s^j_i - s^j_{i,k}\|^2 \\ &= (N-1)\hat{d}^2_5 + |s^k_i - s^k_{i,k}|^2 \\ &\geqslant (N-1)\hat{d}^2_5 + \hat{d}^2_4. \end{aligned} \tag{B.17}$$

It is worth noting that in *Case 8-10*, k is greater than N; therefore, $\boldsymbol{e}_i = \boldsymbol{e}_{i,k}$.

Case 8: $k > N$, $w_i \in \mathbb{E}$. Let us define $s^j_i = s^j_{i-1} + \gamma^j_i$ and $s^j_{i,k} = s^j_{i-1,k} + \gamma^j_{i,k}$, where $\gamma^j_i = (x^j_i - x^j_{i-1}) + I(y^j_i - y^j_{i-1})$ and $\gamma^j_{i,k} = (x^j_{i,k} - x^j_{i-1,k}) + I(y^j_{i,k} - y^j_{i-1,k})$. Using Table 5.4, for all values of i we can write

$$\gamma^j_{i,k} = \gamma^j_i = \begin{cases} 0 & \text{if } e^j_i = 0 \\ 2^{m-\lfloor\frac{m}{2}\rfloor-\lfloor\frac{j}{2}\rfloor-2} & \text{if } e^j_i = 1,\ i \in \mathbb{E} \\ -I2^{\lfloor\frac{m}{2}\rfloor-\lfloor\frac{j-1}{2}\rfloor-2} & \text{if } e^j_i = 1,\ i \in \mathbb{O}, \end{cases} \tag{B.18}$$

and as a result,

$$\begin{aligned} |s^j_i - s^j_{i,k}| &= |s^j_{i-1} + \gamma^j_i - s^z_{i-1,k} - \gamma^j_{i,k}| \\ &= |s^j_{i-1} - s^j_{i-1,k}|. \end{aligned} \tag{B.19}$$

When $k > N$ and $w_{\boldsymbol{i}} \in \mathbb{O}$, using Table 5.4 we can write

$$\begin{aligned} x_i^j - x_{i,k}^j = & \quad (x_{i-1}^j - x_{i-1,k}^j) + (sgn(x_{i-1,k}^j) \\ & \quad -sgn(x_{i-1}^j))2^{m-\lfloor \frac{m}{2} \rfloor -1}, \\ y_i^j - y_{i,k}^j = & \quad (y_{i-1}^j - y_{i-1,k}^j) + (sgn(y_{i-1,k}^j) \\ & \quad -sgn(y_{i-1}^j))2^{\lfloor \frac{m}{2} \rfloor -1}. \end{aligned} \tag{B.20}$$

Let us define $\nabla_x = m - \lfloor \frac{m}{2} \rfloor$ and $\Delta_{x,i} = (x_i^j - x_{i,k}^j)$. Without loss of generality, assume that $x_{i-1}^j > x_{i-1,k}^j$ and therefore $\Delta_{x,i-1} > 0$. In *Case 9* and *Case 10*, we prove that

$$\begin{cases} 2^{\nabla_x -2} \leq |\Delta_{x,i}| \leq 3 \times 2^{\nabla_x -2} & \text{if } 2^{\nabla_x -2} \leq |\Delta_{x,i-1}| \leq 3 \times 2^{\nabla_x -2} \\ |\Delta_{x,i}| = 2^{\nabla_x -1} & \text{if } |\Delta_{x,i-1}| = 2^{\nabla_x -1}. \end{cases} \tag{B.21}$$

Case 9: $k > N$, $w_i \in \mathbb{O}$, $sgn(x_{i-1}^j) = sgn(x_{i-1,k}^j)$. Using (B.20) we have $(x_i^j - x_{i,k}^j) = (x_{i-1}^j - x_{i-1,k}^j)$ and therefore $|\Delta_{x,i}| = |\Delta_{x,i-1}|$. As a result, $|\Delta_{x,i}|$ possesses all the features of $|\Delta_{x,i-1}|$. This indeed proves (B.21) for this case.

Case 10: $k > N$, $w_i \in \mathbb{O}$, $sgn(x_{i-1}^j) = -sgn(x_{i-1,k}^j)$. Without loss of generality, assume that $sgn(x_{i-1}^j) = 1$. From (B.20) we can write

$$x_i^j - x_{i,k}^j = x_{i-1}^j - x_{i-1,k}^j - 2^{\nabla_x}, \tag{B.22}$$

which results in

$$|\Delta_{x,i}| = ||\Delta_{x,i-1}| - 2^{\nabla_x}|. \tag{B.23}$$

If $2^{\nabla_x -2} \leqslant |\Delta_{x,i-1}| \leqslant 3 \times 2^{\nabla_x -2}$, we can use (B.23) to write

$$2^{\nabla_x -2} \leqslant |\Delta_{x,i}| \leqslant 3 \times 2^{\nabla_x -2}. \tag{B.24}$$

In particular, if $|\Delta_{x,i-1}| = 2^{\nabla_x -1}$, from (B.23) we have

$$|\Delta_{x,i}| = 2^{\nabla_x -1}. \tag{B.25}$$

Suppose $\nabla_y = \lfloor \frac{m}{2} \rfloor$, $\Delta_{y,i} = (y_i^j - y_{i,k}^j)$, and $y_{i-1}^j > y_{i-1,k}^j$. Following the same approach in the above two cases, we can prove

$$\begin{cases} 2^{\nabla_y -2} \leq |\Delta_{y,i}| \leq 3 \times 2^{\nabla_y -2} & \text{if } 2^{\nabla_y -2} \leq |\Delta_{y,i-1}| \leq 3 \times 2^{\nabla_y -2} \\ |\Delta_{y,i}| = 2^{\nabla_y -1} & \text{if } |\Delta_{y,i-1}| = 2^{\nabla_y -1}. \end{cases} \tag{B.26}$$

Assume that in step $(i-1)$ and for $j \neq j'$ ($j' \in \{1, \cdots, N\}$), (i) $|s_{i-1}^{j} - s_{i-1,k}^{j}| = \hat{d}_5$, i.e., $|\Delta_{x,i-1}| = 2^{\nabla_x - 1}$ and $|\Delta_{y,i-1}| = 2^{\nabla_y - 1}$, and (ii) for $j = j'$ we have either

$$\begin{cases} 2^{\nabla_x - 2} \leq |\Delta_{x,i-1}| \leq 3 \times 2^{\nabla_x - 2} \\ |\Delta_{y,i-1}| = 2^{\nabla_y - 1} \end{cases} \tag{B.27}$$

or

$$\begin{cases} |\Delta_{x,i-1}| = 2^{\nabla_x - 1} \\ 2^{\nabla_y - 2} \leq |\Delta_{y,i-1}| \leq 3 \times 2^{\nabla_y - 2}. \end{cases} \tag{B.28}$$

It is important to note that the above assumption is already true for step 1 (see proposition 5.5) and for any step when $k \leqslant N$ (see *Case 1-7*). Using (B.27) and (B.28), in step i, we have $|s_i^j - s_{i,k}^j| = \hat{d}_5$ for $j \neq j'$; and when $j = j'$, one of the followings will be satisfied:

$$\begin{cases} 2^{\nabla_x - 2} \leq |\Delta_{x,i}| \leq 3 \times 2^{\nabla_x - 2} \\ |\Delta_{y,i}| = 2^{\nabla_y - 1} \end{cases} \tag{B.29}$$

or

$$\begin{cases} |\Delta_{x,i}| = 2^{\nabla_x - 1} \\ 2^{\nabla_y - 2} \leq |\Delta_{y,i}| \leq 3 \times 2^{\nabla_y - 2}. \end{cases} \tag{B.30}$$

From (B.29) and (B.30), we conclude that $|s_i^{j'} - s_{i,k}^{j'}| \geqslant \hat{d}_4$. As a consequence, we can write

$$\begin{cases} |s_i^j - s_{i,k}^j| = \hat{d}_5 & \text{if } j \neq j' \\ |s_i^j - s_{i,k}^j| \geqslant \hat{d}_4 & \text{if } j = j'. \end{cases} \tag{B.31}$$

From (B.31), we have

$$\begin{aligned} \|\boldsymbol{s}_i - \boldsymbol{s}_{i,k}\|^2 &= \sum_{j=1}^{N} \|s_i^j - s_{i,k}^j\|^2 \\ &= (N-1)\hat{d}_5^2 + |s_i^{j'} - s_{i,k}^{j'}|^2 \\ &\geqslant (N-1)\hat{d}_5^2 + \hat{d}_4^2. \end{aligned} \tag{B.32}$$

Finally, using (B.17) and (B.32) it is concluded that in step i, we have

$$\|\boldsymbol{s}_i - \boldsymbol{s}_{i,k}\|^2 \geqslant (N-1)\hat{d}_5^2 + \hat{d}_4^2, \tag{B.33}$$

which implies that $\hat{d}_{min,i}^2 \geq (N-1)\hat{d}_5^2 + \hat{d}_4^2$.