A Performance Comparison of Polar Codes with Convolutional Turbo Codes

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Üstün Özgür

November 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erdal Arıkan(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Defne Aktaş

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ali Özgür Yılmaz

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

A Performance Comparison of Polar Codes with Convolutional Turbo Codes

Üstün Özgür
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Erdal Arıkan
November 2009

Polar codes introduced recently by Arıkan are the first low-complexity codes achieving symmetric capacity for arbitrary binary-input discrete memoryless channels (B-DMCs). Although being theoretically significant, their practical significance is an issue that has not yet been fully explored. Previous studies have compared polar codes with Reed-Muller codes, where it was found that polar codes can outperform them. In this thesis, to investigate how polar codes perform against state-of-the-art forward error correction (FEC) codes used in practice, we implement a IEEE 802.16 based link-level Worldwide Interoperability for Microwave Access (WiMAX) simulator which incorporates several WiMAX FEC options, and polar codes. IEEE 802.16 standards family define standards for current and next generation broadband wireless access, which will make high data rate multimedia applications in mobile environments a reality. Next generation broadband access standard, pursued by the IEEE 802.16 Task Group m is a work in progress, and requires even more sophisticated error correction schemes so that higher throughput, better QOS, higher mobilities, wider ranges and lower latencies are supported. We perform performance comparison simulations with

the convolutional turbo codes (CTC) configurations defined in IEEE 802.16e to see how much of a performance gap exists between polar codes and CTCs. The main findings of the thesis are that, although the polar codes achieve capacity for specific conditions, as expected, for the code lengths and channel conditions we have simulated, the performance of them cannot compete with that of the CTCs with equivalent rates and lengths. It remains a task to see whether polar codes can achieve similar performances with CTCs when used as component codes in other configurations and aid in the advancement of new communication technologies.

Keywords: WirelessMAN, WirelessMAN-OFDMA, IEEE 802.16e, IEEE 802.16m, physical layer technologies, polar codes, convolutional turbo codes, WiMAX, WirelessMAN-OFDMA Simulator, MIMO, Reed-Muller Codes, Performance Comparison

ÖZET

KUTUPLAŞMA KODLARININ EVRİŞİMLİ TURBO KODLAR İLE BAŞARIM KARŞILAŞTIRMASI

Üstün Özgür
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Erdal Arıkan
Kasım 2009

Yakın bir tarihte Arıkan tarafından tanıtılan, ikili ayrık hafızasız kanallar için simetrik kapasiteye ulaştığı kanıtlanan ilk ileri hata düzeltme yöntemi olan kutuplaşma kodlarının teorik önemi gösterilmiş olsa da, pratikte bu kodların önemi daha tam olarak araştırılmış durumda bulunmamaktadır. Önceki çalışmalar kutuplaşma kodlarının Reed-Muller kodlarından daha yüksek performans ortaya koyduğunu göstermektedir. Bu tezde, kutuplaşma kodlarının en gelişmiş ileri hata düzeltme kodları karşısında nasıl bir başarım gösterdiğini incelemek amacıyla IEEE 802.16e PHY (fiziksel) seviye belirlemesi (spesifikasyonu) tabanlı ve WiMAX onaylamasıyla (sertifikasyonuyla) uyumlu bir bağlantı seviyesi benzetici (simulatör) gerçekleştirilmiştir ve bu benzetici içerisine WiMAX ileri hata düzeltme kodları ve kutuplaşma kodları entegre edilmiştir. IEEE 802.16 standartları ailesi güncel ve sonraki nesil genişbant kablosuz erişim için standartlar tanımlayarak, yüksek veri hızı gerektiren çokluortam uygulamalarının mobil ortamlarda kullanımını olanaklı kılmayı amaçlamaktadır. IEEE 802.16 Görev Grubu m tarafından geliştirilmekte olan sonraki nesil genişbant erişim standardı daha yüksek aktarım hızı, daha iyi servis kalitesi (QoS), daha yüksek mobilite,

data geniş alan ve daha düşük gecikme desteği amaçladığı için hata düzeltme yöntemlerinde önceki standartlara göre daha gelişmiş tasarımlar gerektirmektedir. Tezde kutuplaşma kodlarının IEEE 802.16e standardı dahilinde tanımlanan CTC düzenleşimleriyle başarım karşılaştırması yapıldı. Tezin genel sonuçları şu şekilde özetlenebilir: Kutuplaşma kodları her ne kadar bahsedilen koşullarda kapasiteye ulaşsa da bu kodların başarımı, benzetimi yapılan kod uzunluk ve oranlarında ve benzetim ortamlarında, CTC kodlarının başarımıyla yarışamadı. Kutuplaşma kodlarının öğe kodlar olarak yeni kod düzenleşimlerinde kullanılması durumunda nasıl bir başarım göstereceği ve yeni iletişim teknolojilerine nasıl katkı sağlayabileceği gelecek çalışmaların konusunu oluşturacaktır.

Anahtar Kelimeler: WirelessMAN, WirelessMAN-OFDMA, IEEE 802.16e, IEEE 802.16m, fiziksel seviye teknolojileri, kutuplaşma kodları, evrişimli turbo kodlar,WiMAX, WirelessMAN-OFDMA Benzeticisi, MIMO, Reed-Muller Kodları, Başarım Karşılaştırması

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Erdal Arıkan for his constant guidance and support during my studies and graduate education. Definitely the most exceptional person I have known, he is a man unmatched in intelligence and knowledge, yet so humble, kind and helpful all the time. More than an advisor, he has at times been like a father to me, always tolerating and forgiving. I regret that I have not lived upto his exceptations and possibly caused loss of time of this great man, which could otherwise be spent more efficiently without doubt.

I would like to thank my family, my mother Reyhan and my grandparents Şükran and İbrahim for always believing in me no matter what. They have been my constant supporters and without their support, I would not have endured this process.

I would like to thank all my friends, especially Damla Ateş, Can Bal and Ozan Doğu Tuna for their endless support and belief in me. Even though I believe they base their beliefs on history rather than present, I too hope that future will resemble history much more than present. I would like to thank all my past teachers, and all other people who have passed through my life.

I would also like to thank Assist. Prof. Defne Aktaş and Assoc. Prof. Ali Özgür Yılmaz for serving as members of my thesis committee, accepting my invitation without hesitation.

I would like to thank Iterative Solutions for providing the Iterative Solutions Coded Modulation Library (CML)[1] which we use to integrate the WiMAX CTC codes in our simulator. I would like to thank Prof. Arıkan for providing the source code used for polar code support in our simulator.

Last, but not the least, I would like to thank TÜBİTAK for providing financial assistance for the polar coding project involving polar codes and throughout my graduate studies.

[1] http://www.iterativesolutions.com/Matlab.htm

Contents

List of Figures

List of Tables

To my grandparents Şükran and İbrahim Üstün...

Anneannem ve dedeme...

PRELIMINARIES

Abbreviations and Acronyms

- AMC Adaptive Modulation and Coding
- AMS Adaptive MIMO Switching
- AWGN Additive white Gaussian noise
- BER Bit Error Rate
- BS Base Station
- CQI Channel Quality Indicator
- CSI Channel State Information
- EMD 802.16m Evaluation Methodology Document
- ESM Effective SINR Mapping
- IDFT Inverse Discrete Fourier Transform
- LDPC Low Density Parity Check (Code)
- MAP Maximum A Posteriori
- MIMO Multiple-Input Multiple-Output
- MMIB Mean Mutual Information per Bit
- OFDM Orthogonal Frequency Division Multiplexing
- OFDMA Orthogonal Frequency Division Multiple Access
- PER Packet error rate
- PHY Physical (layer)

- PUSC Partial Usage of Subcarriers
- RBIR Received Bit Mutual Information Rate
- SISO Single-Input Single-Output
- SNR Signal to Noise Ratio
- SOFDMA Scalable OFDMA
- STBC Space–time block code
- WiMAGIC Worldwide Interoperability Microwave Broadband Access System for Next Generation Wireless Communications
- WiMAX Worldwide Interoperability for Microwave Access
- MCS: Modulation Coding Scheme

Chapter 1

INTRODUCTION

Polar codes, recently introduced by Arıkan in [2] are the first low-complexity codes that theoretically achieve symmetric capacity of binary-input discrete memoryless channels (B-DMCs), however its practical value has not been investigated yet. It is known that under certain conditions and decoding schemes, it outperforms Reed-Muller codes[3], however its performance advantage suggested that in its current state, it would not compete with state-of-the-art capacity achieving forward error correction (FEC) codes like convolutional turbo codes (CTCs) since CTCs outperform Reed-Muller codes by a wide margin, while the performance difference between the polar codes and the Reed-Muller codes are not that high. In this thesis, we mainly investigate how apart are the performance curves for polar codes so that we get an idea on how much polar codes should be improved from their current state to be useful in practical systems. To test the codes in a realistic environment, we decided to test the codes in a Worldwide Interoperability for Microwave Access (WiMAX) simulator, since it is an important standard for next generation broadband internet access.

Broadband internet access has become an indispensable part of our lives. With its mass adoption, the Internet has become the most important tool in

managing our lives: it has changed the way we work, the way we socialize, the way we communicate. Another important technology that has changed the way we live is that the mobile devices, cellular phones and laptop computers are now ubiquitous. The next paradigm shift in human evolution will happen when these two technologies fully merge; just like the biological evolution has provided us the basic communication tools, our sociological evolution has brought us more and more communication skills and it paves the way for a global human society.

Although there are alternatives to broadband wireless access (BWA) at the moment, the mission of obtaining true BWA has still not been met. By true BWA, we mean that users will have access to data rates that can meet the demands of multimedia applications. The need for high data-rate multimedia applications is very important, since without those applications, the devices through which we access the Internet will still be perceived as agents between us and the Internet. It will be when the devices provide such a realistic environment that its very presence will be forgotten that we will be seamlessly interacting online, every part of lives will be accompanied by a constant connection to the Internet superhighway; a future where every device we use is constantly online.

Of the current options, WiFi simply does not have the coverage to provide such a seamless connection everywhere, and pre-3G cellular technologies fail to provide any useful data rates. With the now becoming popular 3G technologies everywhere, and the emerging 4G technologies, broadband wireless access will be taken to a new level.

IEEE 802.16 standards enter the scene of networking as the perfect solution for providing BWA solutions. For 3G networks, Mobile WiMAX standard based on this family's 2005e standard has been selected as an option, and its next-generation incarnation, IEEE 802.16m aims to satisfy the requirements of the next wave of high speed multimedia supporting wireless standards, the IMT-Advanced, i.e. 4G.

IEEE 802.16 and WiMAX standards do not only provide another option for BWA; but they even serve to a better cause. In under-developed and developing countries where even cellular and PSTN technologies have not been deployed, they will serve those countries as the sole Internet connection, and this will be a huge step forward in bringing the whole humanity the same set of tools and opportunities. Although the adoption seems at danger in developed countries due to the dominance of cellular operators which see LTE as the natural evolution path; WiMAX, with its open nature, will and should be the choice for other markets.

In this thesis, we implement a IEEE 802.16e simulator in MATLAB to perform simulations in which we test the proposed options under different conditions. This standard has been finalized and its latest revisions have been merged into one document in IEEE 802.16-2009. The next step the IEEE 802.16 Technical Working Group will take is IEEE 802.16m, in which even higher data rate applications with better quality of service (QoS) support will be provided.

Our main aim in this thesis was to provide new contributions to the ongoing standardization effort by seeing whether a newly proposed forward error correcting (FEC) technique introduced by Arıkan, polar codes, can be a usable FEC option as it is the first FEC scheme that achieves the symmetric capacity under conditions that will be explained in the relevant sections.

Chapter 2 presents the IEEE 802.16 family of standards and how the WiMAX certification was based upon them. It gives an overview of the history and future of these standards and introduces IEEE 802.16d, 802.16e and 802.16m.

In Chapter 3, we give a detailed description of the IEEE 802.16e simulator we have implemented. Since the options presented in IEEE 802.16 are various, we give detailed information regarding the choices, simplifications and assumptions we have used.

Chapter 4 gives information on polar codes; their construction and integration into the IEEE 802.16 simulation chain. Additionally, we present how we form polar code configurations that match the Convolutional Turbo Code (CTC) configurations defined in the standard.

In Chapter 5, we give our results on comparing the polar codes and CTC schemes under different channel conditions and antenna settings.

Chapter 6 concludes, summarizing our results; and in Appendix, we give extra information regarding the supported modulation and coding schemes, a user guide on the simulator, and our results on optimization of the simulator using C files integrated into MATLAB.

Chapter 2

IEEE 802.16 FAMILY OF STANDARDS AND WiMAX

IEEE 802 family of standards are a standards family of IEEE that focuses on various networking technologies such as personal, local and metropolitan area networks, namely PANS, LANs and MANs. These standards are maintained by several working groups each focusing on a different selection of standards under the direction of IEEE 802 LAN/MAN Standards Committee.

Of these family of standards, the most well-known ones are the 802.3 standard, which standardizes the Ethernet technology, and standard families 802.11 on Wireless LANs and 802.15 on Wireless PANs, upon which respectively the WiFi certification and Bluetooth protocols are based.

IEEE 802.16, another subset of the IEEE 802 family, is itself a family of standards developed by the 802.16 Working Group. Its main aim is to develop wireless broadband standards for metropolitan area sized networks; hence its alternative name, "WirelessMAN".

Within the working group for 802.16, there are several task groups each developing a different 802.16 standard on a different topic. Some of these have completed their tasks, and some are still active. As of August 2009, the WirelessMAN website lists four active task groups, namely "License-Exempt Task Group", " Relay Task Group", "Task Group m (TGm)" and the "Maintenance Task Group". These groups are working on standards 802.16h, 802.16j, 802.16m and 802.16-2004 respectively, either working on drafts or amendments.

2.1 History and Future of IEEE 802.16 Standards and WiMAX

2.1.1 Notable IEEE 802.16 Standards

The IEEE 802.16 Working Group was founded in 1999, and has since created several different standards for wireless MANs. The standards were aimed at fixed deployment scenarios initially, and the major product of that period was the standard 802.16-2004 (16d) published in 2004, a culmination of the previous standards that far, supporting various frequency profiles (2-11 GHz and 10-66 GHz). [4]

A year later, this standard was amended by the work of 'Task Group e' in order to support mobile scenarios, and the new standard, accepted in 2005, which included corrections to, omissions from and amendments to 802.16-2004[5] was published as 802.16e-2005[6], in early 2006. This amendment is also known as "Mobile WirelessMAN", although one should note that fixed communication support still exists when the standards 2004 and 2005 are considered in conjunction [7].

Further amendments published as 802.16f and 802.16g, along with 802.16e-2005 and 802.16d-2004, and fixes have been finally merged into one document and recently published as 802.16-Rev2-2009, which supersedes previous standards.

2.1.2 WiMAX Certification

The IEEE 802.16 standards are responsible for specifying only the PHY and MAC layers of the air interface between a subscriber station and a base station; and as such, other issues for the system to be viable as a practical alternative, such as the end-to-end specification which determines the network architecture are not included.

Furthermore, probably for sake of completeness or political reasons (to appease the proponents of various techniques) , the IEEE 802.16 standards include several optional choices for different building blocks, but the implementation of all of these is cumbersome for vendors, and a common subset of choices has to be selected.

Therefore, akin to the relationship between the 802.11 family of standards and the WiFi certification which is based on that family, the need for certification of products based on 802.16 standards has resulted in the emergence of WiMAX certification, which is controlled by the industry led, non-profit "WiMAX Forum". WiMAX Forum was established in 2003 with the mission to promote WiMAX as a technology based on air interface specifications as a subset of the IEEE 802.16 family of standards, complemented with a set of network specifications. Tasks within WiMAX are carried by its own working groups.

WiMAX stands for "Worldwide Interoperability for Microwave Access" and as its name implies, its main focus is on maintaining the conformance and interoperability of products so that vendor lock-in is prohibited and products of different vendors operate in harmony within a WiMAX ecosystem. The technical

working group states on the WiMAX Forum website that its main goal "is to develop technical product specifications and certification test suites for the air interface based on the OFDMA PHY, complementary to the IEEE 802.16 standards, primarily for the purpose of interoperability and certification of Mobile Stations, Subscriber Stations and Base Stations conforming to the IEEE 802.16 standards.".

Since there is more than one 802.16 standard, correspondingly, there are several WiMAX certifications. Fixed WiMAX is the one based on 802.16d-2004; and mobile WiMAX is the one based on 802.16e-2005. More precisely, the certification based on 802.16e-2005, complemented with Network Profile 1 is called the Mobile WiMAX release 1 certification; and similarly, the certification based on 802.16Rev2-2009, is called the Mobile WiMAX release 1.5 certification.

As of October 2007, The ITU Radiocommunication Assembly recognizes Mobile WiMAX to be in the set of radio interface options for the IMT-2000 standards. These standards are also called 3G standards, and aim to determine the specifications for mobile telecommunication systems requiring high-speed (broadband) data rates.

2.1.3 Roadmap of IEEE 802.16 and WiMAX

The next generation of IEEE 802.16, 802.16m is currently being developed by the Task Group m. This standard is being prepared as an amendment to the previous standards, and therefore the previous standards will remain fixed and new contributions will be appended as additional chapters.

In parallel to the efforts of the Task Group m, the Mobile WiMAX release 2.0 certification is being prepared based on this new standard. As WiMAX has been recognized as a 3G technology, the aim of the WiMAX forum is to make the next

generation of WiMAX (also called WiMAX2) to be a part of the IMT-Advanced standards, better known in public as 4G.[8]

The key enhancements targeted in IEEE 802.16m are given in [9] as follows:

- Doubling relative throughput of a data only system compared to WirelessMAN-OFDMA, support of data rates of several hundreds of Mbit/s
- Doubling relative sector throughput and increasing VoIP capacity by a factor of 1.5
- Mobility support at speeds as high as 350 km/h
- Improved cell coverage, upto a radius of 100 km
- Increased spectral efficiency, decreased latencies

2.2 PHY LAYER OF WirelessMAN-OFDMA

IEEE 802.16-2009 standard specifies the air interface for broadband wireless access systems via three different physical layer (PHY) specifications aiming different operational conditions; namely WirelessMAN-Single Carrier(SC) PHY specification, WirelessMAN-OFDM PHY and finally WirelessMAN-OFDMA PHY. Of these, the single carrier case is focused on higher frequencies of range 10-66 GHz; while the latter two focus on the frequency band below 11 GHz, aiming communication in non line-of-sight situations.

In this thesis, we are concerned with the WirelessMAN-OFDMA PHY specification upon which WiMAX Forum has based the Mobile WiMAX certification. This PHY mode was first introduced by the IEEE 802.16e-2005 amendment to the preliminary IEEE 802.16-2004 standard; and the final state has been reached in the IEEE 802.16-2009 standard which supersedes both.

Major technologies involved in the operation of the WirelessMAN-OFDMA mode are the OFDMA technology and relevant subchannelization schemes, Space-Time Coding supporting various MIMO configurations, and several channel encoding schemes which make use of different Forward Error Correcting (FEC) codes.

Via OFDMA support, the mode supports multiple users whose data is spread across the time and frequency dimensions. This provides frequency diversity in addition to multi-user support. The capacity of the system is increased through the usage of different antenna configurations which are defined by the STC schemes; and based on the operational modes, adaptive antenna selection might be employed. Finally, as the system supports multiple modulation and coding schemes (MCs), more suitable schemes which increase reliability and/or rate might be chosen via link-adaptation, also called adaptive modulation and coding (AMC).

2.2.1 OFDMA Support in WirelessMAN-OFDMA

To understand OFDMA as introduced [10], one should first consider the OFDM modulation. As transmission rate increases in a wireless communication system, two fundamental problems arise: intersymbol interference and multipath fading.

Intersymbol interference occurs as subsequent symbols are affected by each other, causing 'temporal spreading and consequent overlap of individual pulses to the degree that the receiver cannot reliably distinguish between changes of state, i.e. , between individual signal elements' as explained in the definition of the term in Federal Standard 1037C.

Multipath fading occurs when transmission occurs via two or more different paths which arrive at the receiver at distinct time instants. In such a case, the

part of a symbol carried on a path with a significant delay might interfere with symbols at other time instants.

To mitigate this problem, the data rate should be decreased, however given the current demands, this data rate decrease should be done in such as a way that the total data rate is preserved. One such approach to solving this problem is separating the signal into multiple parallel parts which are transmitted simultaneously at a slow rate. In order to be able to distinctly send and receive these signals, they are allocated orthogonal parts of the frequency spectrum, and each of these parts, the smallest unit on which allocation occurs is called a *subcarrier*.

Under perfect synchronization, and given that the coherence time is high enough, each of these subcarriers now experience slow-fading, and the channels experienced can be modeled as single path (single tap) channels.

Each FFT component in an OFDM symbol is mapped to a subcarrier. By a coarse definition, a group of subcarriers is called a *subchannel* and burst formation is accomplished through packing these subchannels in a *burst*, following a predefined algorithm. This procedure basically assures that the subcarriers allocated to a burst are non-contiguous, and therefore a higher frequency diversity is achieved.

WirelessMAN-OFDMA PHY supports various FFT sizes and consequently various OFDM configurations. Specifically, FFT sizes of 2048, 1024, 512 and 128 are supported so that the system operates under different channel bandwidths.

The multi-user access extension was first introduced in [10]. In this mode, after the subcarriers are segregated via OFDM, different users access the system through sets of subcarriers.

There are three types of subcarriers; data subcarriers carry data, null subcarriers are used in guard bands and the DC component; and the pilot subcarriers are used to estimate channel conditions and synchronization purposes.

Once each data subcarrier is assigned (along with its sister subcarriers in the same subchannel) to owners and filled with data from its owner, the resultant OFDMA waveform is converted to time-domain via an IFFT. The resulting time waveform is prepended by a cyclic prefix which comprises of a ratio of the waveform from the end. The main contribution of this operation is immunity against multipath interference and synchronization problems.

2.2.1.1 Subchannelization

How subcarriers are packed into subchannels, and how these subchannels are used to form burst determines the *subchannelization strategy*. There are several different subchannelization schemes in WirelessMAN-OFDMA, namely PUSC, FUSC, TUSC and AMC.

Similar to other system parameters, most of these choices are optional and only PUSC is mandatory. As such, our simulator only implemented the PUSC scheme, however for completeness, other schemes will be introduced below as well. The introduction will be followed by a detailed explanation of the PUSC scheme we have implemented and some visual aids are given to gain further insight as to how the subchannels are formed.

Partial Usage of Subcarriers (PUSC)

PUSC subchannelization scheme stands for Partial Usage of Subcarriers. In this scheme, the cell is divided into different sectors, and the available subchannels are divided into segments. As a result, only a part of the available subcarriers can be used, hence the name partial. How segments are formed is actually based on the frequency reuse scheme, so if a segment

covers the whole frequency range, PUSC might cover the whole frequency range, albeit the subchannelization still occurs according to the partial pattern.

We divide the subcarriers into clusters of 14 adjacent subcarriers. Then, we renumber the physical clusters to logical clusters. Next, we map the pilots and take them out. Then, we create six major groups consisting of 24 or 16 clusters (for an FFT size of 2048). Then, from each major group, we select the subcarriers for each subchannel. See Figure 2.1 for some sample subchannels.

Full Usage of Subcarriers (FUSC)

FUSC subchannelization scheme stands for Full Usage of Subcarriers. In this scheme, all the subcarriers are available for usage. We group the subcarriers so that there are N groups where N is the number of subcarriers we want in a subchannel, i.e. 48. Then, from each group, we pick a subcarrier for each subchannel. See Figure 2.2 for some sample subchannels.

Tile Usage of Subcarriers (TUSC)

Tiles are used in TUSC instead of the clusters in PUSC. More information is given in the standard.

AMC

Subcarriers are distributed to subchannels in adjacent sets.

More information regarding the subcarrier permutation types, with examples, are given in [11].

2.2.1.2 Burst Construction and Burst Profile Selection

Now that we have covered how subchannels are formed from subcarriers, we can go on to explain how these subchannels are allocated to users. The allocation of

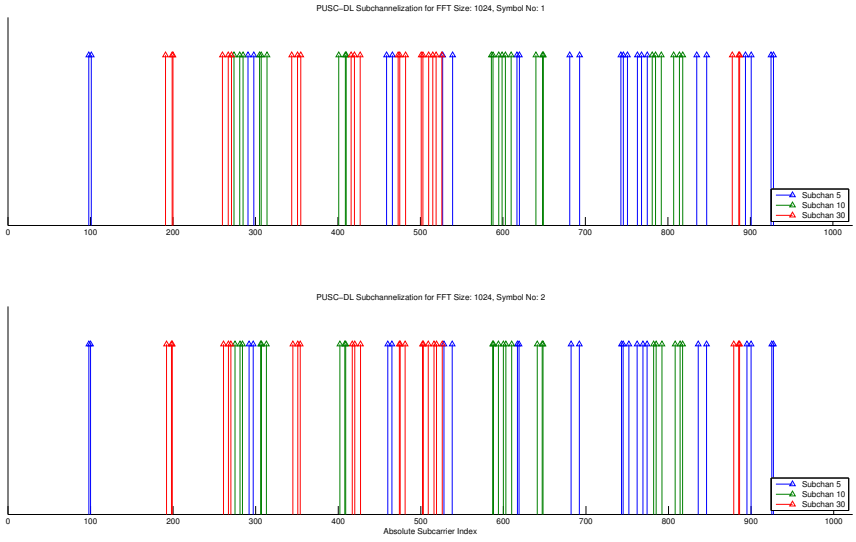

Figure 2.1: Sample PUSC-DL Subchannel Formations for $N_{FFT} = 1024$

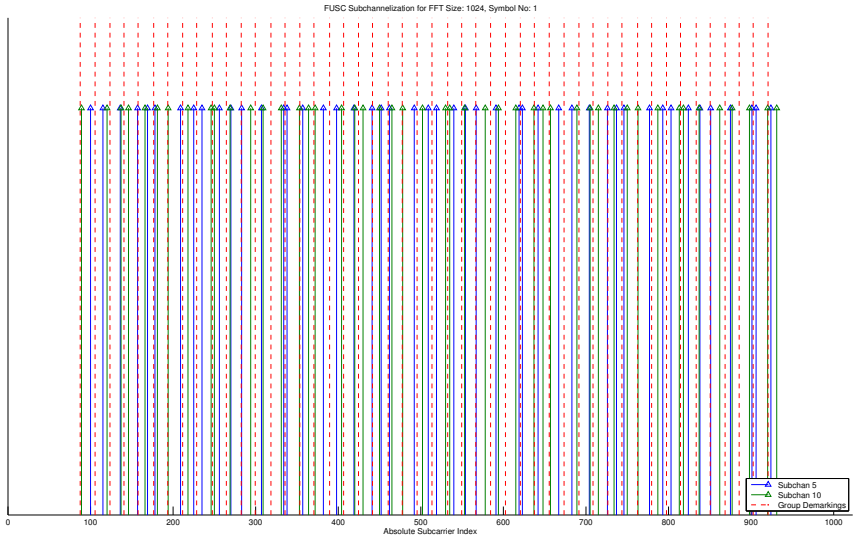

Figure 2.2: Sample FUSC Subchannel Formations for $N_{FFT} = 1024$ – Red dashed vertical lines show the group demarkings

subchannels occurs via supersets of subchannels called bursts. In WirelessMAN-OFDMA, a burst is defined as the region of the total OFDMA frame reserved for a particular user.

For example, if a user is allocated subchannels 10-20 over the OFDM symbols 5-15, that rectangle constitutes the burst allocated for that user. The allocation is not necessarily a single rectangle, but might be a combination of several rectangles.

A user might transmit more than one burst in a frame, however only one user can transmit at a given burst region. A data region, or a partition are equivalent terms for defining a burst.

The coding and modulation scheme in the burst is constant throughout the burst, and this selection is called the 'burst profile'. As such, a burst is composed of an integer multiple of FEC blocks each coded by the same coder and constellation mapped by the same mapper.

A burst profile in WirelessMAN-OFDMA is determined by two parameters: The choice of Forward Error Correcting (FEC) code, and the choice of constellation mapping.

FEC Code Choices Choosing a FEC code requires deciding on three FEC parameters: The type of FEC, the rate of FEC, and the length of the data vector to be encoded.

Types of FEC supported in WirelessMAN-OFDMA compromise convolutional coding, which is mandatory, and three other optional FEC types, namely Convolutional Turbo Codes (CTC), Low-density Parity Check Codes (LDPC), and Block Turbo Codes (BTC).

Each of these support various rates and payload lengths, which are given in tables in Appendix B.

Constellation Mapping Choices The choices for constellation mapping are restricted to three, QPSK, 16-QAM and 64-QAM. Of these, QPSK is the mandatory one, and the support for others is optional. All mappings obey the Gray mapping criteria, and the constellation for each are given in Figure 2.2.2.4.

2.2.1.3 Burst Zone

Multiple bursts form a burst zone via stacking the downlink (DL) bursts together and the UL bursts together with Transmit Transition Gap (TTG) and Receive Transition Gap (RTG) in between the stacks. Bursts are combined with the preamble, Downlink MAP (DL-MAP) , frame control header (FCH) and Uplink Map (UL-MAP), the ranging subchannel to make up a burst zone. The subchannelization is fixed within a burst zone, similar to the fixation of burst profiles within bursts; and there is a mandatory burst zone using PUSC.

The discussion above applies to TDD frames which we are concerned with in this thesis; also supported in WiMAX 1.0. As the latest 802.16 standard supports FDD operation as well, WiMAX is poised to support FDD systems in the future; however that issue is beyond the scope of this thesis.

2.2.1.4 Data Mapping Through Slots

Once a user is allocated a burst, we already know the subchannels in that burst. Through subchannelization, we also know the subcarriers within each subchannel; therefore the set of subcarriers a user is assigned is known. Assume for the moment that the user has full-buffer data, and her data has already been encoded and modulated into (baseband) symbols.

Data mapping at those points occurs through another allocation unit called slots. A slot is defined as a set of subchannels based on the subchannelization scheme. For the mandatory scheme of DL-PUSC, a slot is defined as a subchannel over two OFDM symbols.

Since the slot is defined as the smallest data allocation unit, the data mapping follows this simple rule: Fill the slots one by one, selecting them first by increasing the subchannel index and then by increasing the OFDM symbol index. The catch here is that since a slot itself spans more than one OFDM symbol in the DL-PUSC case, allocation is performed as below:

1. Determine the slot to be filled. It is the left-most slot, i.e. it should have the lowest subchannel index and if there is a tie there, it should have the lowest OFDM index.

2. Fill the first subchannel inside the first OFDM symbol within that slot. Then, continue by filling the subchannel in the next OFDM symbol.

3. Once the slot is filled, proceed to the next slot, starting from step 1.

One final remark that should be made about slots is that since the encoded and modulated blocks will be mapped to the time-frequency matrix slot by slot, a single slot results in low-length blocks. To overcome this, which degrades performance, slot concatenation is defined that merges a number of slots according to the MCs selected inside the burst.

For illustration purposes, assume that a user is allocated the first 8 subchannels across 22 OFDMA symbols through the DL-PUSC scheme for an FFT size of 1024. A slot is defined as one subchannel over two OFDMA symbols in PUSC. Therefore, the user is assigned a total of $22 * 8/2 = 88$ slots. Assume that the MCS chosen is rate-1/2 CTC with QPSK modulation.

The largest encoded and modulated payload length is 432 symbols for this burst profile; so we need at least 432 subcarriers for each FEC block. The maximum allowed concatenation number is 10, therefore, initially, the slot concatenation tries to create as many code mappings as possible through concatenations of 10 slots. Once it creates 7 of such blocks consisting of 10 slots, there are 18 slots remaining, which it uses as 2 blocks of 9 slots. So, the major logic behind the algorithm is create as much as longest possible slots, while at the same time trying to create longer slot configurations from the remaining slots. The exact formulation is given in IEEE 802.16-Rev2[1].

2.2.2 Channel Encoding

Channel encoding comprises the following steps in sequential order: Randomizer, FEC encoding, bit-interleaving and modulation. The input is the raw data from the user, and the output of the procedure is the data to be mapped via the OFDMA allocation.

2.2.2.1 Randomization

This step is restarted for each FEC block. The data is serialized, then XOR'ed with a pseudorandom bit sequence. The pseudorandom bit stream is generated using a seed of $[0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1]$ and effect of randomization on a sequence of raw data is shown in Figure 2.3.

2.2.2.2 FEC Coding

Four FEC choices are provided, and each choice comes with a set of supported configurations for constellation size and FEC rate. A selection of constellation size and FEC rate constitutes a burst profile selection. Besides the burst profile

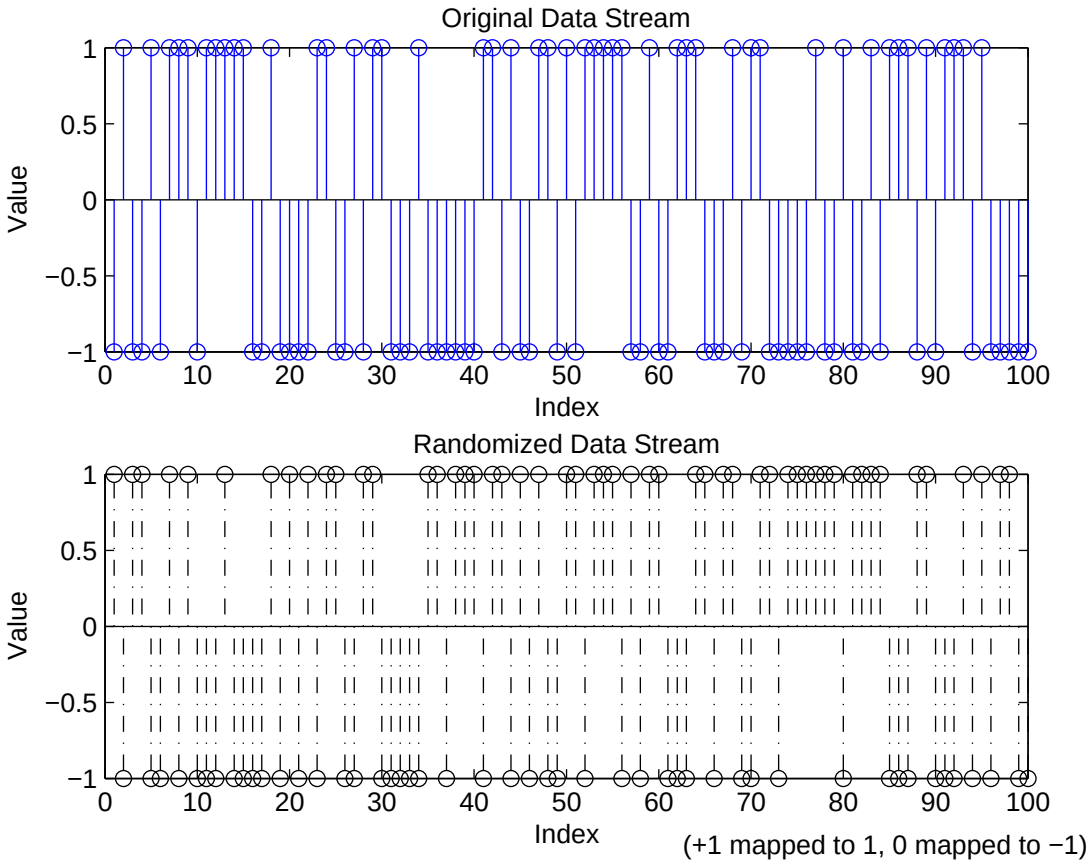

Figure 2.3: Data Stream and Its Randomized Version

selection, another degree of freedom for code construction is the code length. For a given rate and constellation choice, each code provides different code lengths that fit into the concatenated slots explained in previous sections. So, while a burst has a fixed burst profile, even though we know that a code $\mathcal{C}$ with a FEC rate $\mathcal{R}$ is chosen there along with either QPSK, 16-QAM or 64-QAM, the code length depends on the allocated data region size. Longer code lengths are favored for better performance.

We omit details regarding the encoding and decoding of these well-known codes since they are discussed extensively in the literature. For a brief overview of all the FEC schemes in IEEE 802.16e, [12] presents a concise introduction and a performance comparison of these codes.

In the next subsections, we give information regarding the rates of these FEC options.

2.2.2.2.1 Convolutional Encoding Convolutional encoding is the only mandatory FEC coding scheme. Set of supported rates is $\{1/2, 2/3, 3/4\}$ although the latter two are obtained via puncturing of codes with rate $1/2$. Generator polynomials for outputs of coding, X and Y are given as $G_1 = 171_{\mathrm{OCT}}$ and $G_2 = 133_{\mathrm{OCT}}$, of whose graphical interpretation is given in Figure 2.4. A complete list of supported convolutional code configurations is given in Appendix B.

Convolutional codes are mandatory in WiMAX as well.

2.2.2.2.2 Block Turbo Codes These codes are optional codes and not used in WiMAX. BTC is a two dimensional code making use as component codes binary extended Hamming codes.

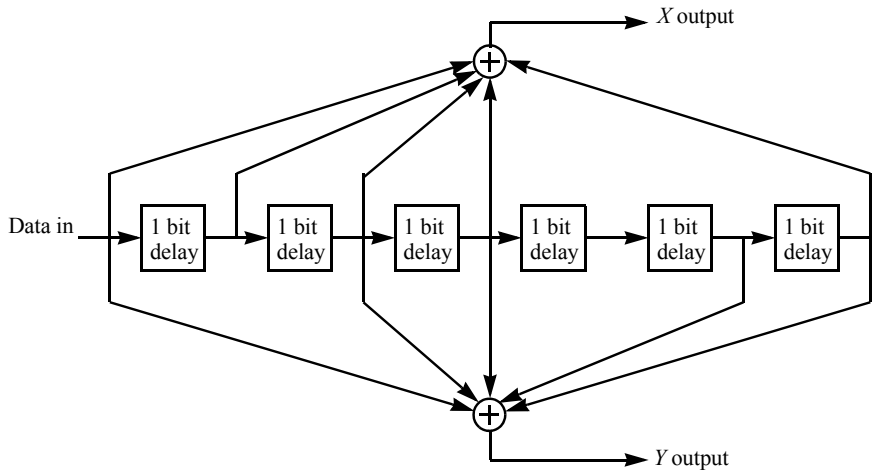

Figure 2.4: Block diagram depicting the encoding process for Convolutional Encoder (courtesy of [1])

2.2.2.2.3 CTCs Convolutional turbo codes are high-performance capacity achieving codes. These codes are optional in IEEE 802.16-2009, but are mandatory in WiMAX. These make use of a slot concatenation rule different from that of convolutional codes, and 8 different burst profiles are supported, namely QPSK with rates $\{1/2, 3/4\}$, 16-QAM with rates $\{1/2, 3/4\}$ and 64-QAM with rates $\{1/2, 2/3, 3/4, 5/6\}$. Along with different number of code lengths for each profile, the total number of possible CTC code configuration is 32 and the list is given in the Appendix B.

CTC encoder, with its constituent encoders are given in Figure 2.5. As seen from the figure, there are three paths: The top two paths are for the systematic transaction, i.e. the input is fed to the output intact. In the third path, however, the input first passes through a constituent encoder. The input to this block is determined as a combination of the original channel input paths A and B, through interleaving and switching. Inside this encoder, one can see that the bits fed first are the natural ordered input bits. For the second case of inputs, interleaved bits are used. The outputs are collected to generate subpackets in the

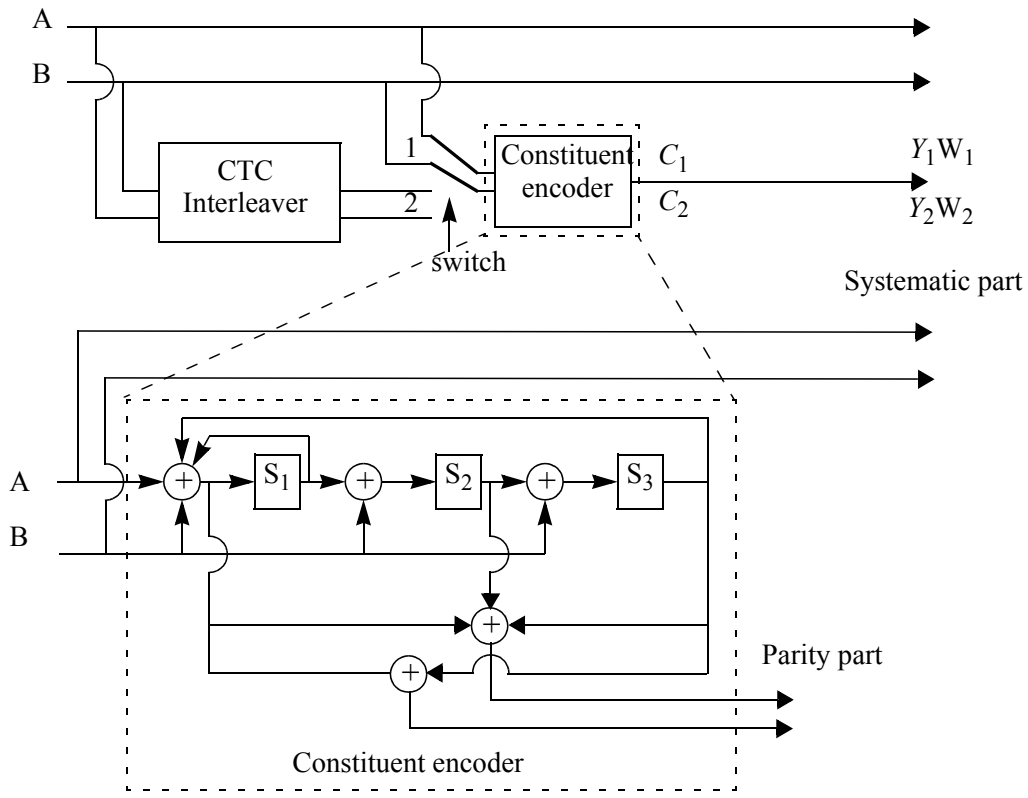

Figure 2.5: CTC Encoder Diagram(courtesy of [1])

following order: A, B, Y_1, Y_2, W_1, W_2. Interested reader might find detailed information about this process in the final version of the IEEE 802.16 standard [1].

2.2.2.2.4 LDPC Similar to CTCs, LDPC codes also achieve capacity practically.

Rates $\{1/2, 2/3, 3/4, 5/6\}$ are supported for each constellation choice, so there are 12 different LDPC burst profiles.

Numerous block lengths are supported for each profile, the whole list is given in the Appendix B.

2.2.2.3 Interleaving

After the data is encoded via the FEC, a block is interleaved via a two-stage block permutation. Neighbor subcarriers are separated via the first operation and is independent of modulation.

The second permutation is modulation dependent, and serves in scattering neighboring bits to random parts of the constellation.

At the receiver side, reverse permutations are applied in reverse, in order to retrieve the original sequence.

Since the CTC encoder already has an integral interleaver, interleaver block is not used for configurations using CTC as FEC choices.

For illustration purposes, observe the transformation of the following encoded block sequence indices k in second column in Table 2.1. m_k refers to the index of that bit after the first permutation and is defined by the operation:

$$m_k = (N_{cbps}/d)k \mod (d) + floor(k/d)k = 0, 1, \ldots, N_{cbps} - 1 \quad d = 16 \qquad (2.1)$$

where N_{cpc} is the compression gain obtained by modulation mapping and equal to 2, 4, 6 for QPSK, 16-QAM and 64-QAM respectively and N_{cbps} is the code block length input to interleaving process.

Similarly, observe how the second permutation defined by the equation below, interleaves those bits in Table 2.1, where $s = N_{cpc}/2$:

$$j_k = s \cdot floor(m_k/s) + (m_k + N_{cbps} - floor(d \cdot m_k/N_{cbps})) \mod (s) \qquad (2.2)$$

Table 2.1: The Effect of Interleaving Steps on the Index Number for a Code Length of 192 in a 64-QAM Setting

k	m_k	j_k
0	0	0
1	12	14
2	24	25
3	36	36
4	48	50
5	60	61
6	72	72
7	84	86
8	96	97
9	108	108
10	120	122
11	132	133
12	144	144
13	156	158
14	168	169
15	180	180
16	1	1
17	13	12
18	25	26
19	37	37
20	49	48
21	61	62
22	73	73
23	85	84

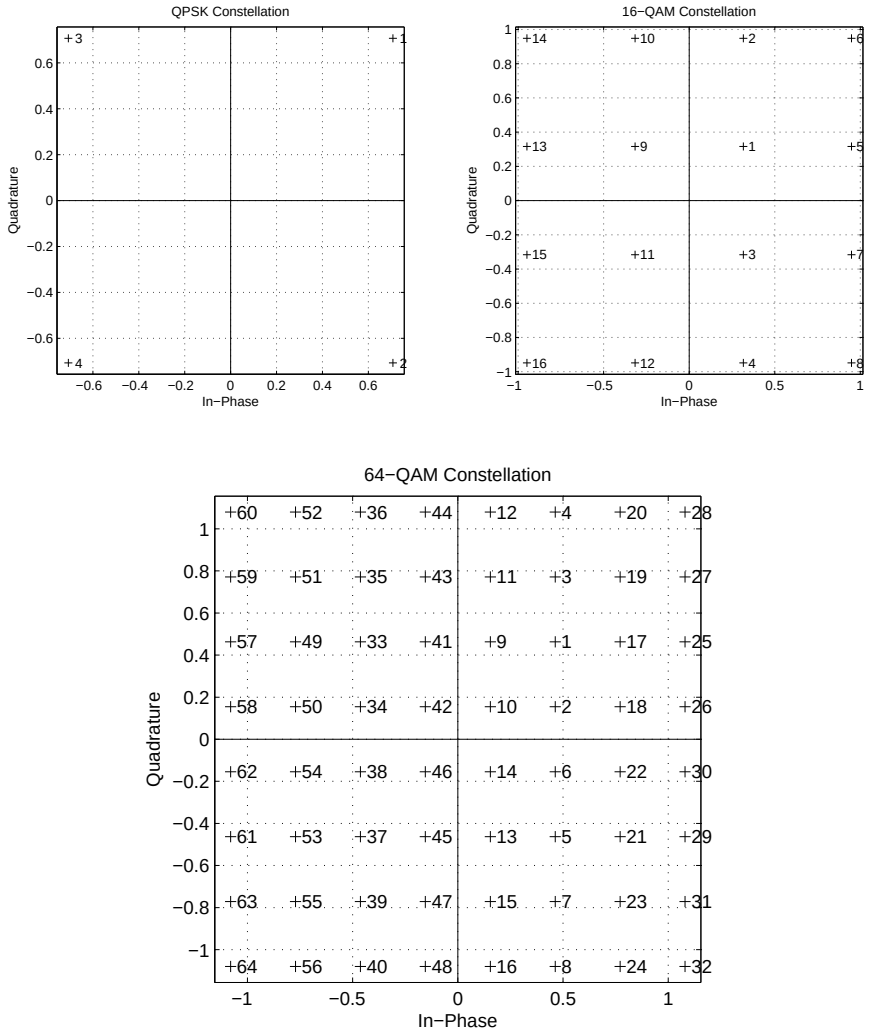

Figure 2.6: Constellations for Different Mappings (Numbers signify the decimal notation of bit sequences, added by 1)

2.2.2.4 Modulation

Constellation mapping using Gray-mapped QPSK, 16-QAM are mandatory while 64-QAM support is optional. The average symbol power is normalized by multiplying the constellation by an appropriate scaling factor.

The figures below show the constellations for QPSK, 16-QAM and 64-QAM.

Once the data is constellation mapped, it is further multiplied by a factor of $2 \cdot (1/2 - w_k)$ where w_k is a pseudo-random bit sequence defined according to [1,

Section 8.4.9.1.4.1]. Pilot carriers are modulated in a different fashion, boosted 2.5dB compared to the data subcarriers. Interested reader may find further detail in [1, Section 8.4.9.1.4.1].

2.2.3 Multiple Input Multiple Output : MIMO

As channel capacity using single antennas were approached with capacity achieving codes such as LDPC and CTC, improvements in data rate increase needed to be done in a different fashion. The single input single output (SISO) scheme was no longer sufficient. It was shown that the exploitation of multiple antennas at the transmitter and/or receiver side resulted in linear capacity increases with the minimum of the number of transmit and receive antennas [13]. As a result, much of the telecommunications research in the last decade concentrated on Multiple Input Multiple Output (MIMO) concepts, and this technology was immediately applied to real life problems.

WirelessMAN-OFDMA is also one of the newest telecommunications standards of almost all making use newer MIMO technologies. It supports various MIMO schemes, all of them optional. Of particular note is the 2x1 Alamouti scheme [14]. IEEE 802.16m also defines a MIMO 2x2 option which we have implemented in our simulator as well.

For these MIMO schemes, we use the method first proposed by Alamouti in [14]. Two different schemes have been used: 2 by 1, for which there are two transmit antennas and 1 receive antenna; and 2 by 2 for which there are 2 transmit and 2 receive antennas.

Although we did not do simulations with 2x1 MIMO scheme, but only 2x2, it is more appropriate to first introduce this basic case.

For the 2x1 MIMO scheme based on Alamouti's paper, assume the fading coefficients between the first transmit antenna and the receive antenna are denoted by h_0 where $h_0 = \alpha_0 e^{j\theta_0}$ and similarly the fading coefficients between the second transmit antenna and the receive antenna are denoted by h_1 where $h_1 = \alpha_0 e^{j\theta_0}$.

We assume the fading coefficients are constant over two OFDMA symbols, therefore we make the corresponding changes in our channel models.

The symbol transmitted from the first transmit antenna is denoted by s_0 at the first OFDMA symbol period (for each subcarrier) and by s_1 for the second transmit antenna. In the next OFDMA symbol, $-s_1^*$ is transmitted from the first antenna, while s_0^* is transmitted from the second antenna.

After noise in the form of AWGN is added to the received signals at the first and second time instances, we get the following equations:

$$r_0 = h_0 s_o + h_1 s_1 + n_0 \tag{2.3}$$

$$r_1 = -h_0 s_1^* + h_1 s_0^* + n_1 \tag{2.4}$$

Next, we perform the Alamouti combining scheme to decouple the two symbol estimates, such that

$$\tilde{s_0} = (\alpha_0^2 + \alpha_1^2) s_0 + h_0^* n_0 + h_1 n_1^* \tag{2.5}$$

$$\tilde{s_1} = (\alpha_0^2 + \alpha_1^2) s_1 - h_0^* n_1 * + h_1^* n_0 \tag{2.6}$$

Now that we have separated the two estimates, ordinary maximum likelihood detection is performed, with proper normalization, which will be explained after introducing the 2x2 case.

For the two transmit-two receive antenna case, the methodology is similar, and done as outlined again in [14].

Instead of h_0 and h_1, this time, we have h_0 for the coefficients between transmit antenna 1 and receive antenna 1, h_1 for the coefficients between transmit

antenna 2 and receive antenna 1, h_2 for the coefficients between transmit antenna 1 and receive antenna 2 and finally h_3 for the coefficients between the transmit antenna 2 and receive antenna 2.

The received symbols in that case are the following:

$$r_0 = h_0 s_0 + h_1 s_1 + n_0 \tag{2.7}$$

$$r_1 = -h_0 s_1^* + h_1 s_0^* + n_1 \tag{2.8}$$

$$r_2 = h_2 s_0 + h_3 s_1 + n_2 \tag{2.9}$$

$$r_3 = -h_2 s_1^* + h_3 s_0^* + n_3 \tag{2.10}$$

The combiner in this scheme results in:

$$\tilde{s_0} = h_0^* r_0 + h_1 r_1^* + h_2^* r_2 + h_3 r_3^* \tag{2.11}$$

$$\tilde{s_1} = h_1^* r_0 - h_0 r_1^* + h_3^* r_2 - h_2 r_3^* \tag{2.12}$$

This time, the final decoupled estimates are given as:

$$\tilde{s_0} = (\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2) s_0 + h_0^* n_0 + h_1 n_1^* + h_2^* n_2 + h_3 n_3^* \tag{2.13}$$

$$\tilde{s_1} = (\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2) s_1 - h_0 n_1^* + h_1^* n_0 - h_2 n_3^* + h_3^* n_2 \tag{2.14}$$

Again, once these estimates are gained, we feed them into the maximum likelihood decoder as usual.

The caveat in both these cases is that we first normalize the two estimates by the square root of the term before the sent signal, for example for the 2x1 case, the sum of magnitude squared of h_0 and magnitude squared of h_1. This is crucial in order to keep the SNR reported to the decoder the same as before. Accordingly, however the fading coefficients are now not equal to the first term ahead of s_0 and s_1, but divided by the same number.

One other important issue is that, since the total energy transmitted should be kept constant, we send signals with half the total energy allocated from both

antennas. This, in turn, can be equivalently represented by a division of the fading coefficients by a factor of $\sqrt{2}$ (Actually, the SNR value perceived is multiplied by a factor of the same amount).

The case is similar for the two by two case, in which we first normalize the decoupled estimates by the square root of the term in front of the original signals, and change the fading coefficients to reflect this change, finally dividing the fading coefficients by a factor of $\sqrt{2}$ in order to keep the SNR levels accurate.

Chapter 3

POLAR CODES

Polar coding is a recently introduced coding scheme by Arıkan that constructs the first known codes that theoretically achieve channel capacity with low-complexity under certain constraints. Specifically, the method achieves the symmetric capacity of binary-input discrete memoryless channels (B-DMC), and given the channel is symmetric, the achieved symmetric capacity is equal to the channel capacity.

3.1 Preliminaries

The notation in this chapter follows strictly the notation presented in the Preliminaries section of [2]. To summarize the notation there, a B-DMC is represented by $W : \mathcal{X} \to \mathcal{Y}$ with $W(y|x)$ denoting the probability of receiving $y \in \mathcal{Y}$, given that $x \in \mathcal{X}$ is sent. If the channel is used subsequently N times, W^N denotes this extension.

The symmetric capacity is denoted by $I(W)$ as an indicator on the upperbound for rate, defined as equal to $\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0)+\frac{1}{2}W(y|1)}$ and the Bhattacharyya parameter is denoted by $Z(W)$ and defined as $Z(W) =$

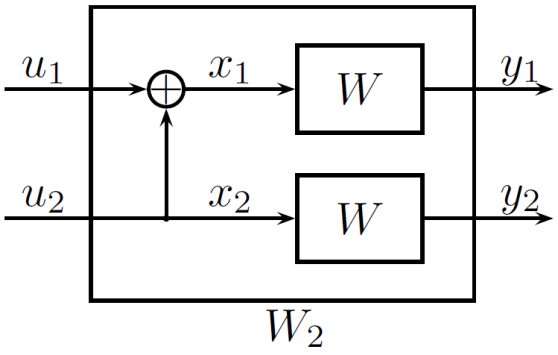

Figure 3.1: Raw Channel and Initial Transformation to Form Size-2 Superchannel (from [2])

$\sum_{y\in\mathcal{Y}} \sqrt{W(y|0)W(y|1)}$, useful as an upperbound on the MAP decision error of a generic single transmission, single detection scenario.

The symmetric capacity becomes equivalent to the channel capacity defined by Shannon if the channel under consideration is a symmetric channel. A symmetric channel satisfies the following properties: The output alphabet should have a reversible permutation π such that $\pi^{-1} = \pi$ and $W(y|1) = W(\pi(y)|0), \forall y \in \mathcal{Y}$.

3.2 Overview

The scheme makes use of the fact that given N independent channels, it is possible to construct N different (perceived) channels which exhibit a polarization effect such that their symmetric capacities approach the poles of capacity limits, i.e. either go to 0 or 1. In those cases, the channel becomes completely unreliable (pure noise) or very reliable (pure information). Arıkan names this phenomenon "channel polarization".

Once we understand this polarization behavior, the problem of coding making use of it boils down to two main issues. The first issue is whether we can find a code encoding scheme that produces such polarized channels. Once we find

such channels, the second issue, encoding at a given rate, becomes simple. We basically make use of the channels whose symmetric capacity approaches 1, and send fixed data through the noisy channels.

The first issue, producing polarized channels is a two-step process. Given N binary-input (possibly similar) channels, we first synthesize via a method called "channel combining" one superchannel whose input is a vector of length N . The output of this channel, which acts like an envelope is the same length as its input, a vector of length N.

Consequently, we analyze, i.e. decompose the superchannel into N distinct and dissimilar binary-input channels through the "channel splitting" process. These output channels are in fact imaginary channels that are perceived by their input bits, and are not binary channels in the ordinary sense that a given input produces an output of same size (one bit). Therefore they are unlike the binary-input, same-length output as input channels we had used as ingredients.

The output of a single polarized channel is the Cartesian product of the whole output set of the superchannel, and the input set of the other single polarized channels that consists of channels with indices smaller than the current channel, i.e. the inputs to the upper single channels. So, the output of a single channel consists of ordered pairs that connect outputs to previous inputs.

After we polarize the raw channels through these two methods, the second issue emerges: Which channels should we use at a certain rate? This step is called "code construction" since it is at this point we determine which channels will be used to transmit information and which channels will serve for redundancy. The intuitive approach to this question is selecting the channels according to either their capacities or their error probabilities. One should simply select the channels with the highest capacity or the channels with the lowest error probabilities.

More precisely, we are concerned with the symmetric capacities and the Bhattacharyya parameters of the channels, which are denoted by $I(W)$ and $Z(W)$ respectively. It turns out that, as expected, using these two are completely equivalent since the highest "rate" sustained by a channel is correlated by its "reliability". It is shown in [2] that the sum of the Bhattacharyya parameters of the channels on which we send information gives a bound on the probability of block error under successive cancellation decoding, and hence their usage is favored for channel sorting.

3.3 Channel Transformation

As mentioned above, we need a transform to produce N polar channels out of N raw channels. The first step, channel combining is a recursive process in which one begins with a single channel, and creates superchannels of increasing size, at each step combining the two most recently generated superchannels. Therefore, two independent copies of the initial single channel W give rise to the first superchannel W_2. Two such superchannels are combined to form the second level superchannel W_4. This recursion is generalized such that at level $n-1$, two superchannels of size $N/2$ such that $N = 2^n$, are combined to obtain the superchannel of size N.

The initial combination first requires that inputs are paired adjacently. The geometrically lower input (with the higher index) is mod-2 added to the upper input to create the input to the raw channel geometrically higher. The lower input item in the pair is transferred as the input of the second, lower channel. This is equivalent to multiplying the input pair (as a row vector) with the matrix $F_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ operating over the binary field $GF(2)$.See Figure 3.1.

This operation has to be repeated at each channel combining step. One other step is required for other combination instances: a reverse shuffle operation[1] so that the odd indexed terms of the latest operation are stacked at the top in ascending index order, and the even ordered terms are stacked below those, again in ascending index order. This operation actually exists in the initial case as well, however in that case, we already have the odd numbered unit x_1 above the even numbered one, x_2 in that case.

Therefore, the algorithm to generate W_N , a superchannel of size N is generalized as below:

1. Pair the input bits and apply F_2 to each pair.
2. Perform a permutation operation on the N-length output such that the odd-numbered latest inputs are above even-numbered ones (indexing starts from 1).
3. The first half is used to create a smaller superchannel of size $N/2$. Similarly, the second half is used to create another smaller superchannel of size $N/2$.
4. To create these channels, return to step 1, and repeat until we reach a target superchannel of size 1.

Since there are $\log(N)$ recursion steps above in Figure 3.2, and N operations at each step as we have N units, the complexity of the encoding process is $O(N \log(N))$ and therefore the encoding operation is of low-complexity. A rigorous proof of this is given in [2].

Channel combining alone is sufficient to encode a given input sequence of length N, but does not give any information as to what kind of a channel our

[1] Actually, this step is not required during encoding given an equivalent process is performed during decoding.

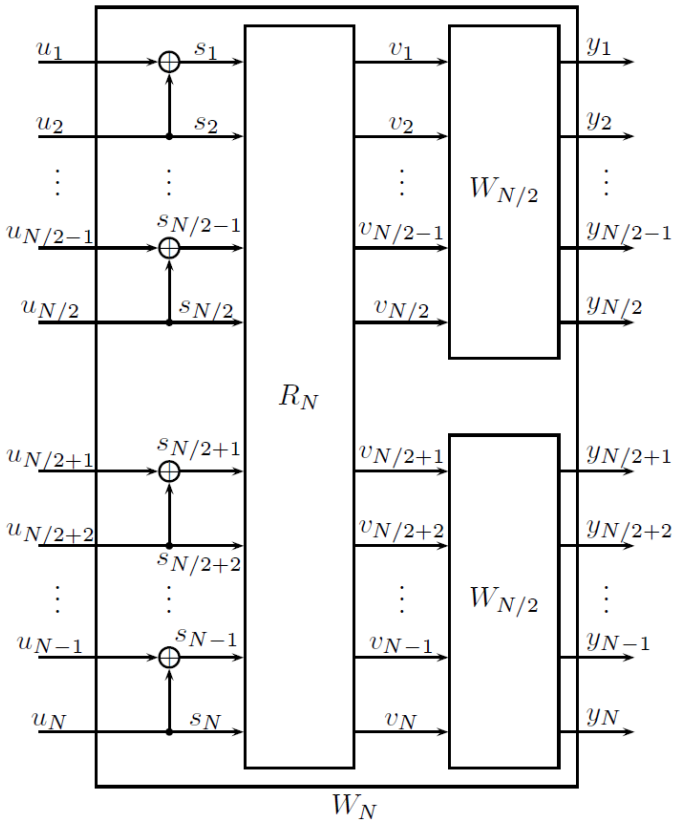

Figure 3.2: Recursive Construction of the Combined Channel of Size N (from[2])

inputs experience. We need this information in order to determine how to form the input sequence itself and to see how the polarization effect takes place.

Through channel splitting, we define the new, abstract channels an individual input bit experiences. It is given formally as:

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) = \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N)$$

3.4 Properties of Polarized Channels

Now that we have defined the precise abstract channel experienced, we can treat these as ordinary channels, and calculate the symmetric capacity and the Bhattacharyya parameters. Once we calculate and sort the Bhattacharyya parameters, given a code length of N, we select K of the channels through which to send the information bits such that the ratio K/N matches our target code rate.

Figure 3.3 shows the symmetric capacities of channels as n varies from 0 to 6 where $N = 2^n$ determines the encoded block length. The numbers next to points signify their index in the preencoding stage, and the numbers below them in italics show their position when the symmetric capacities are sorted in ascending order.

The difficulty here is that if we use arbitrary raw channels, there is no simple way to calculate the Bhattacharyya parameters. This constitutes the actual complexity of polar codes, since the encoding process is not complex, and the decoding process, dual of encoding is on the complexity order of encoding.

However, if we select a BEC with an erasure probability $\epsilon = 0.5$ as raw channels, the computation of the symmetric capacity and the Bhattacharyya parameters are straightforward and recursive. In fact, this channel type is ideal to show the effects of polarization.

Below, we present the Bhattacharyya parameters of codes of length 2^n for $n \in \{0, 1, \ldots, 10\}$. We note that the general trend is that the reliabilities of individual channels increase as their indices increase, however this behaviour is not observed all the time. Especially in the midrange, the order of reliability turns out to be rather unpredictable without exact computation.

To gain further insight to the polarization process, consider the following histogram where the Bhattacharyya parameters are separated into 40 distinct

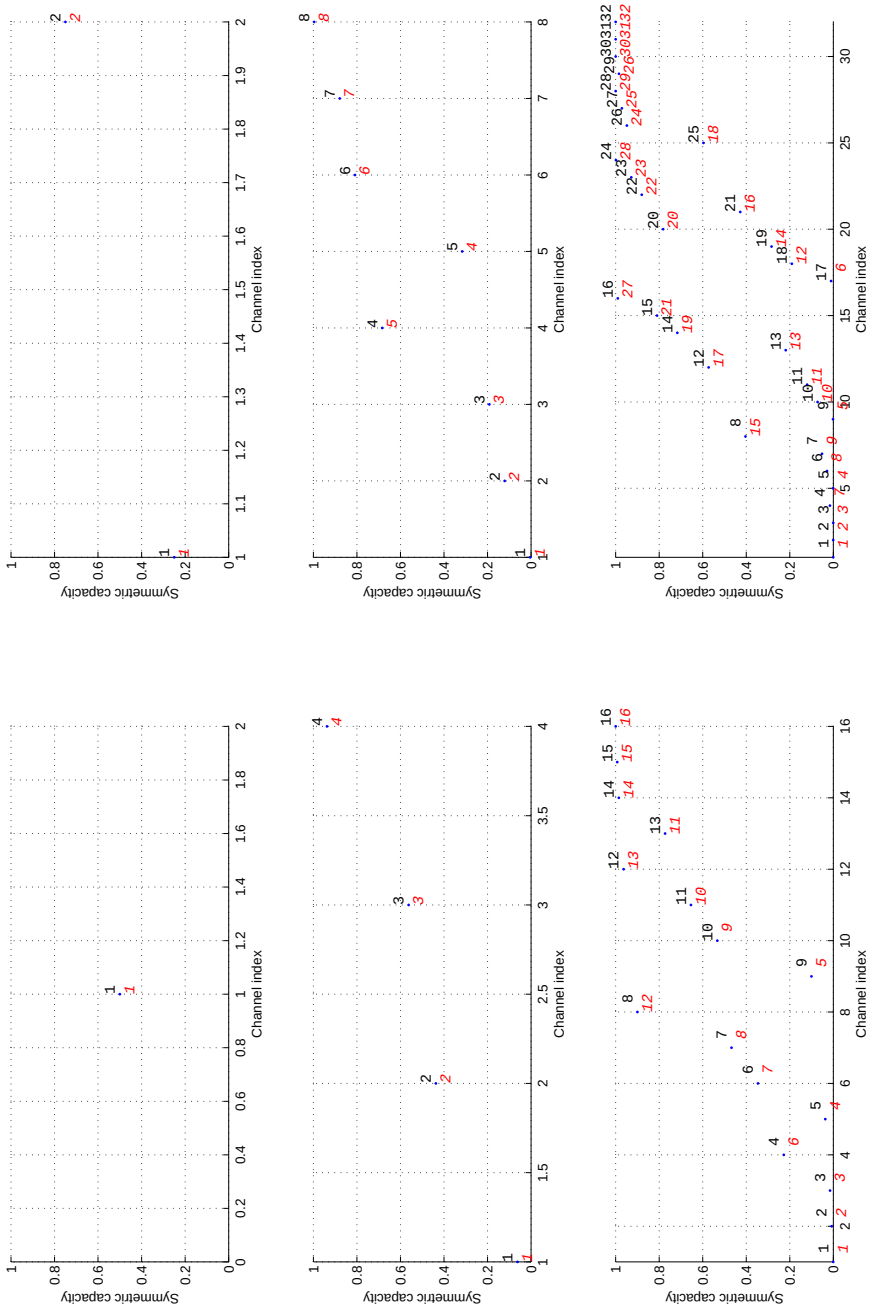

Figure 3.3: Polarization of Channels as $n : 0 \rightarrow 6$ with Slanted Numbers Signifying the Position with Reverse Sorted Symmetric Capacity – the higher the more reliable channel

bins in the interval $[0, 1]$. Notice how the parameters are accumulated at the first and last bins as the code length increases. The interpretation here is that we start with an unreliable single channel for $n = 0$. For $n = 1$, we now have two channels slightly apart. The separation increases as n increases, and we start to have two sets of more reliable channels or less reliable channels. In fact, these graphs show that the polarization effect does not prevail for small code lengths as evidenced by the significance of intermediate bins. The speed at which those intermediate bins are emptied constitutes the issue of rate of channel polarization.[2]

3.5 Polar Coding Revisited: Comparison with Reed-Muller Codes

We had previously defined the encoding of polar codes as following: Start with an F_2 kernel transformation on pairs proceeded with a permutation, and perform a recursion, following the same procedure to generate half-length superchannels. The mapping until the recursion is linear, and it was shown in [2] that the whole process is therefore linear. So, if the input to the combined channel is the vector x_1^N and the inputs to the raw channels are members of the vector u_1^N, the transformation can be summarized as $x_1^N = u_1^N G_N$.

This approach forms the equivalent view on the encoding algorithm for polar codes in [2]: The overall mapping can be summarized by a generator matrix G_N, which is defined as a matrix of size N and calculated as $B_N F^{\otimes n}$ where B_N refers to a bit-reversal process and $F^{\otimes n}$ is the n^{th} Kronecker power of the kernel matrix $F = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$.

[2]Note that for the last plot, we focus on the 1% region to see how the non-polarized channels have decreased in size, they are on the order of 0.1 percent

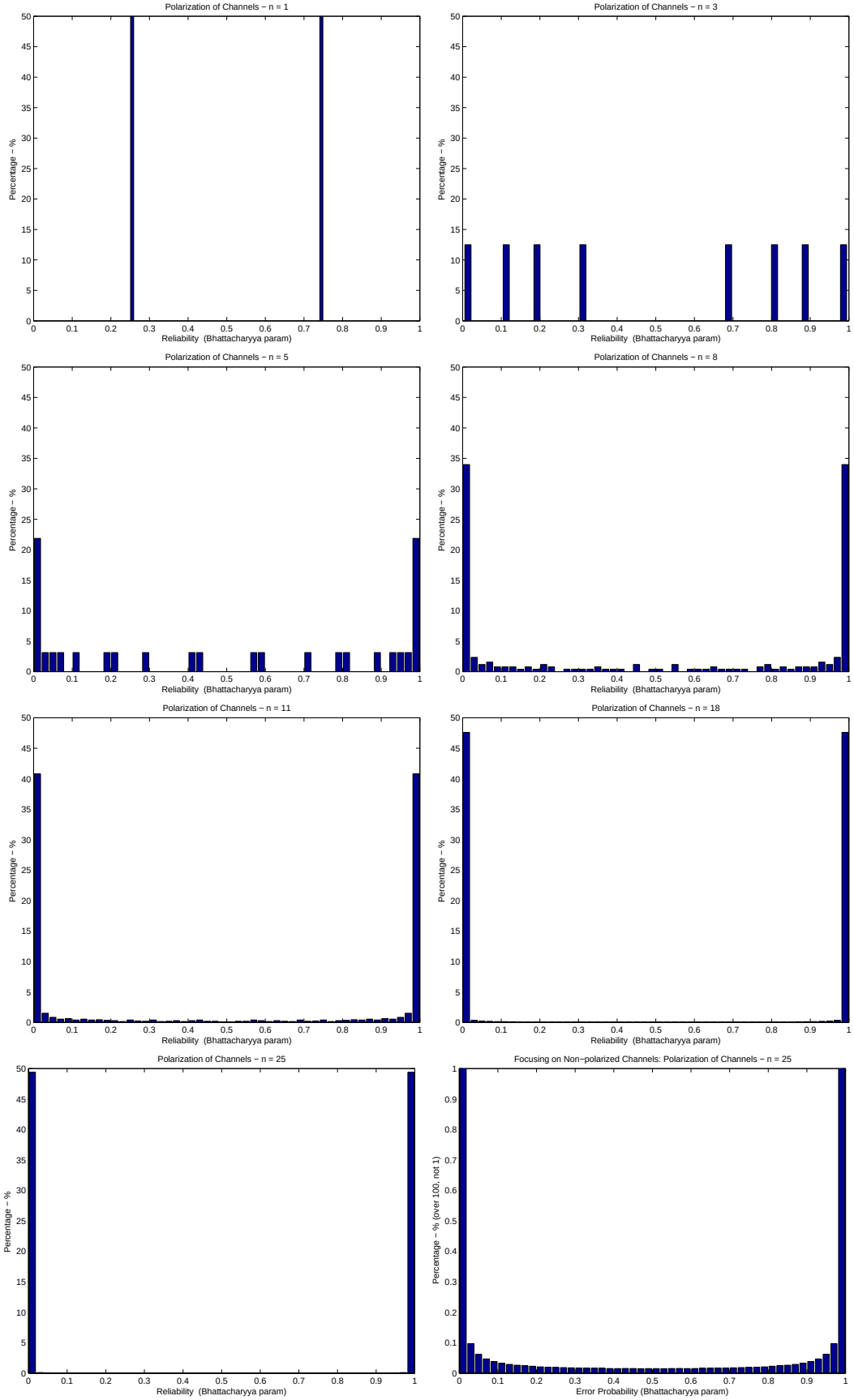

Figure 3.4: Polarization of Channels as n increases

At this point, we digress from discussion of polar codes to note that Reed-Muller codes are actually encoded similarly, using a generator matrix of same number of columns, but reduced row length [3]. While polar coding sets the unreliable input bits to a fixed value, preserving the full size of the generator matrix, RM codes remove rows corresponding to the unreliable input bits, not using them at all. Other than this, the major difference between the two type of codes lies however in the selection of rows, that is, the encoding operations are the same, a matrix multiplication; but the code constructions differ.

The motivation behind Reed-Muller code construction is the following: Instead of using Bhattacharyya parameters, the selection is done using the Hamming weight of rows. The rows with the heaviest weights possible are chosen, such as for a (N, K) code, we select rows such that the rows not selected are at most as heavy as the lightest row selected. It is clear that this selection is not unique, unlike polar codes where each channel has a distinct metric.

We observe that for some code configurations, there are matching instances between the two, that is the same set of rows are selected so that their matching locations carry the information bits. However, we see that as the code length increases, dissimilarities take hold.

The intuition behind the heavy-weight row-selection of Reed-Muller codes can be explained through a rather coarse pipeline analogy. Think of the paths emerging from the transformed channel inputs to the raw channel inputs as drains emerging from households to the common drains. The nodes at which mod-2 summation occurs can be likened to junctions where two pipes are combined.

First observe the last transformed channel for the Reed-Muller case. This pipe disseminates its contents through not only its horizontal path, but upwards to other paths, via diagonal and horizontal means. In fact, once its content is mixed with another pipe above, if that other pipe itself disseminates its new

contents to other pipes (which contains remnants of the initial pipe), we observe that the contents of the first pipe is disseminated even further.

Actually, that pipe simply makes use of the best redundancy possible and in the N-channel case, its information is spread onto all the upstream paths, and in turn to all the raw channels. Now, had there been no other contribution from other pipes, this would actually be equivalent to using the channel N times to transmit one bit. Even though we know there *are* contributions, we still get a reliability in between the original reliability and that of extension N channel. The intuition that this channel is more reliable – hence polarized towards a symmetric capacity of 1 – is therefore obvious.

Also observe via this example that the "interference" caused by an input on other paths *only* occurs in the upstream direction. An information sent travels not only via its horizontal path, but it gets mixed up by other input bits which are nearer to the top; and the output generated via sending it is on not only the horizontal path, but all the output bins. This gives more insight to the input-output characteristics of the new channels viewed as a black box.

Next, observe the first transformed channel. The pipe corresponding to this is polluted with the contents of pipes below, and it still is using only one path. If the contribution from other pipes is none, we end up with is a channel that is the same as the old raw channel. Clearly, there are contributions from other pipes and the channel is at best as reliable as it once was: this pollution makes the channel even less reliable, hence the channel polarizes towards a symmetric capacity of 0.

So, what about other locations such that we can obtain a code rate of K/N? The approach Reed-Muller coding uses here is quite simple: It says that the number of contributions to a path make it dirtier in a linear fashion, i.e. it simply calculates how many other contributions are on a horizontal path. If there are

too many contributions from other pipes, there is too much pollution and this makes the channel less reliable. Hence, it selects the pipes on which there are as few nodes as possible. It does not matter how much a contribution pollutes the channel, it only matters that it pollutes it. In fact, transforming the Trellis structure such that nodes without addition are mapped to 1s and nodes with addition are mapped to 0, we select the heaviest rows, that is, the rows with as less additions as possible, number of zeroes as small as possible.

The approach polar coding uses is more sophisticated. As a general trend, similarly, it accepts that the more contributions to a path, the dirtier it will be. However, this is not the actual rule. In a way, one may visualize that polar coding also calculates not only if a pipe contributes its content to other pipes, but how it contributes. It therefore takes into account not only the number of contributions, but also the quality or significance of those contributions. If a path has more contributions, but the path is resilient against those than paths with less contributions, polar coding will select that path.

How polar coding determines the significance of contribution we talked about in the analogy is dependent on the channel used, so polar coding is channel-aware. In fact, it is this awareness of channel itself in polar coding that should in theory make this method more suitable for adapting the code according to channel conditions, a technique known as Adaptive Modulation Coding. Although we have not yet made any such experiments, the flexibility of polar codes to different channel conditions combined with its pliant rate adaptation makes it suitable as an ingredient for future AMC schemes.

3.6 Decoding

There are different decoding mechanisms that might be employed to decode polar codes. The initial decoding algorithm given in [2] was successive-cancellation

(SC). This method is vital for theoretical proofs, but fails to provide good performance at practical code lengths. Another choice is the Belief-Propagation (BP) algorithm. This method works via iterations, and gives better performance than the SC algorithm. One final choice is ML decoding, which is optimal, but has higher complexity which makes decoding impossible above a certain block length. As such, polar codes can be decoded through three different mechanisms suitable for different purposes and situations.

The decoding strategy we use for polar codes is belief propagation since it has been shown that it performs better than the successive cancellation decoding scheme, and achieves similar performance to that of ML decoding scheme whose complexity prohibits its usage in performance demanding applications and at long FEC lengths. We are using the MATLAB implementation of the decoder Arıkan has used in his paper [3].

Chapter 4

PHY LAYER SIMULATOR

We have implemented a WirelessMAN-OFDMA PHY layer link-level simulator supporting multiple subscribers across a frame using MATLAB. The simulator is available both via a GUI for ease of use; and script files to automate the simulation of several configurations in batch. The MATLAB version used in our development and simulations is 2008b. The GUI requires at least MATLAB R2008a although the command line interface should support even older versions. We have not made use of fairly new MATLAB technologies like object oriented programming although the structure is quite modular with functions neatly separated for each individual operation. Simulating a frame full of DL and UL bursts depends on the code choice, however usually lasts around 3 seconds for our configuration of a Core 2 Q6600 machine with 8 GB RAM, running Debian-AMD64.

4.1 Simulation Chain

A block diagram is given below in Figure 4.2. The channel coding step, which comprises randomization, FEC encoding, interleaving and constellation mapping

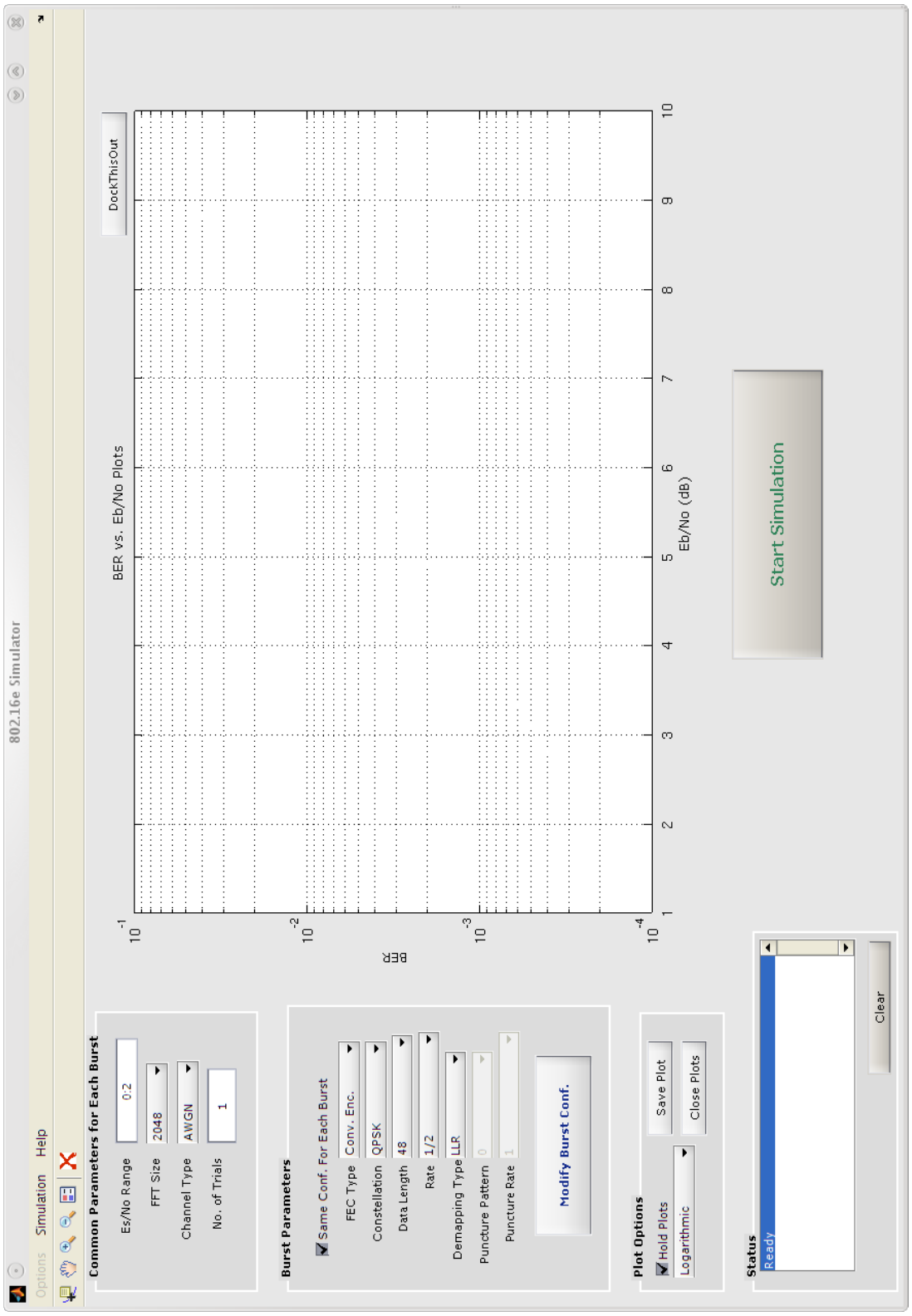

Figure 4.1: Screenshot of the IEEE 802.16-2009 Simulator GUI

is followed by subcarrier allocation with a PUSC permutation scheme. Inverse fast Fourier transform (IFFT) is applied on the resulting signals, and after adding the cyclic prefix, transmission occurs at baseline.

At the receiver, first the cyclic prefix is removed, then fast Fourier transform (FFT) is applied; and the resulting signal is fed to the demapper which produces the log likelihood ratio (LLR) values. Subcarrier deallocation occurs, and each burst is regenerated. Each FEC block in the burst is decoded with the steps of the encoding process in reverse order.

4.1.1 Simulator Blocks

Encoder These blocks can be separated into three categories:

- Allocation blocks : These blocks are for the burst allocation within the frame, and subcarrier allocation for each burst. Each subscriber will be assigned a burst whose subcarrier indexes over OFDM symbols are determined via these blocks.
- Channel encoding blocks: These compromise the randomization, FEC encoding, interleaving and mapping blocks.
- Channel and antenna blocks: Channel model blocks and support for MIMO configurations are given in this block.

Decoder The blocks in the encoder will be presented in reverse order. LLR demapping will be followed by the explanation of deinterleaving, FEC decoding and derandomization blocks.

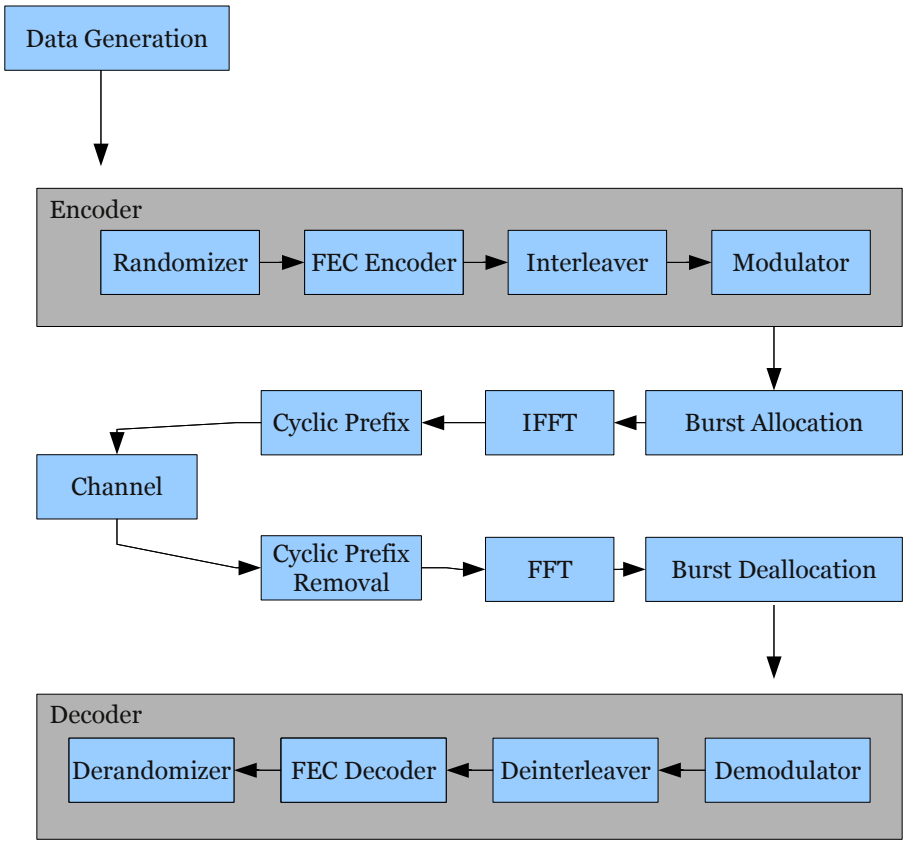

Figure 4.2: Block Diagram of the Simulator

4.1.1.1 FEC

The simulator uses CML Encoded Library[16] as the FEC backend which features MEX-files written to speed up simulation; and as such supports the burst configurations there, i.e. most of the configurations in the standard. Currently only CTC and LDPC codes in the CML library are made use of, convolutional encoding is implemented in MATLAB; and BTC is not supported (though support for BTC can easily be added).

Besides standard WiMAX codes, polar codes are added as a FEC option. Key functions within polar code encoding and decoding are written as MEX files within MATLAB which contribute a speed increase of on average 18 times, see Table 4.1 to compare the performance improvement using different compilers. Further information regarding our optimization efforts are given in Appendix A.

Table 4.1: MEX File Performance Improvements Using Different Compilers

Compiler	Average Gain (x)
MS	17.7779
LCC	10.7501
Mingw	18.3047
Cygwin-mingw	18.0044

Another issue to consider in integrating polar coding is the code configurations. Due to their recursive nature, polar codes natively support code lengths of 2^n. The rate is determined by selecting a portion of this code length as information positions, hence rate matching is fairly easy for polar codes if the denominator of the rate fraction is a power of two. So, rates like 1/2, 3/4 with code lengths 128, 256 are easy to implement.

We compare polar codes with CTC's with code length given in Appendix B. Those code length however are not multiples of 2, therefore in order to compare the performance of two codes, we have selected the following route: For an (N, K) CTC code, the equivalent polar code is the given by (N', K') where N' is the

nearest power of 2 to N, and K' is chosen such that the code rate is preserved. For a list of equivalent polar codes to those defined in the standard, refer to Appendix B.

Another optimization we have done in the whole simulation chain has been to make use of parallel for (parfor) blocks in MATLAB to speed up the embarrassingly parallel parts of the chain, i.e. parts which can be carried out independently, like encoding different FEC blocks in parallel. On the Core 2 Q6600 machine sporting four cores, this parallelization has resulted in a speed increase of 2. We suspect that the reason speed increase is not proportional to the increase in the number of CPUs to be a bottleneck caused by the communication overhead since the parallelized portions are too small. In the pursuit of these parallelization efforts, we have made some experiments with using a cluster of computers for simulations; however found that for our current simulation needs, this is not necessary at this point.

4.1.1.2 Subcarrier Allocations

PUSC is used for subcarrier allocations, and currently only the mandatory PUSC zone is used. The zone is divided into several rectangular bursts, both in DL and UL. Unless stated otherwise, we use 22 DL OFDM symbols, and 15 UL OFDM symbols. There are 6 partitions for DL and UL respectively, and each subscriber is assigned the whole horizontal space, and 1/6th of the subchannels available, e.g. 5 subchannels for $N_{FFT} = 1024$. This burst configuration is given below in Figure 4.3.

Although we have used fixed data regions for subscribers in our simulations, the geometric allocation of each burst (the OFDMA symbols and logical subchannels each burst spans) and burst profiles for each burst (modulation type,

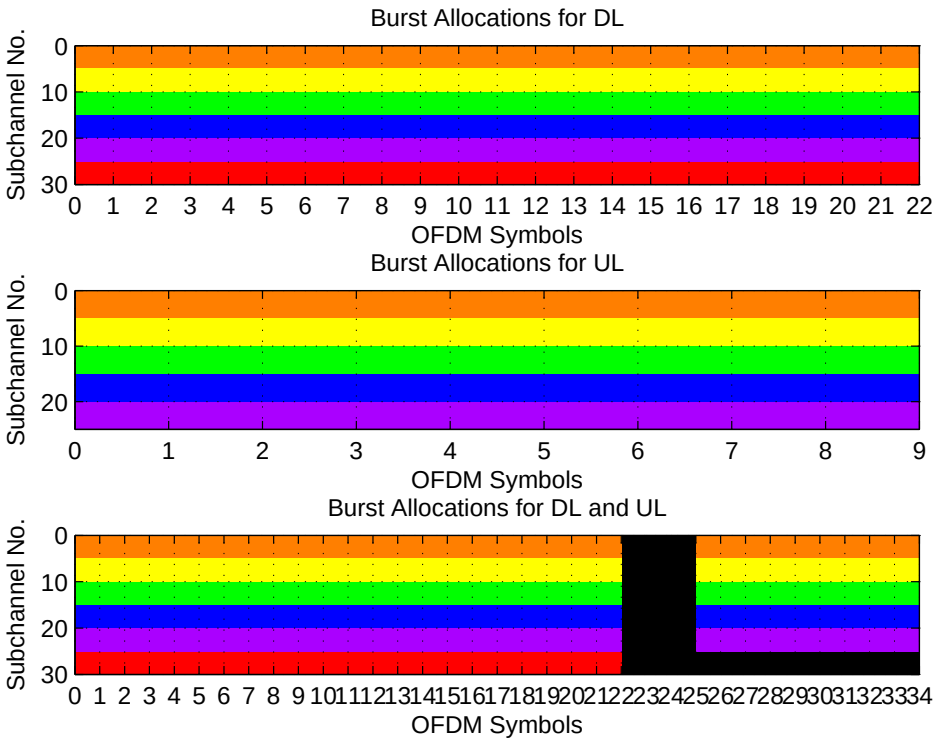

Figure 4.3: Default Allocation of Bursts for $N_{FFT} = 1024$

FEC type, rate) can be defined using a MATLAB GUI. Figure 4.4 shows a sample burst configuration and resulting plot generated by the GUI. This allocation step is currently manual.

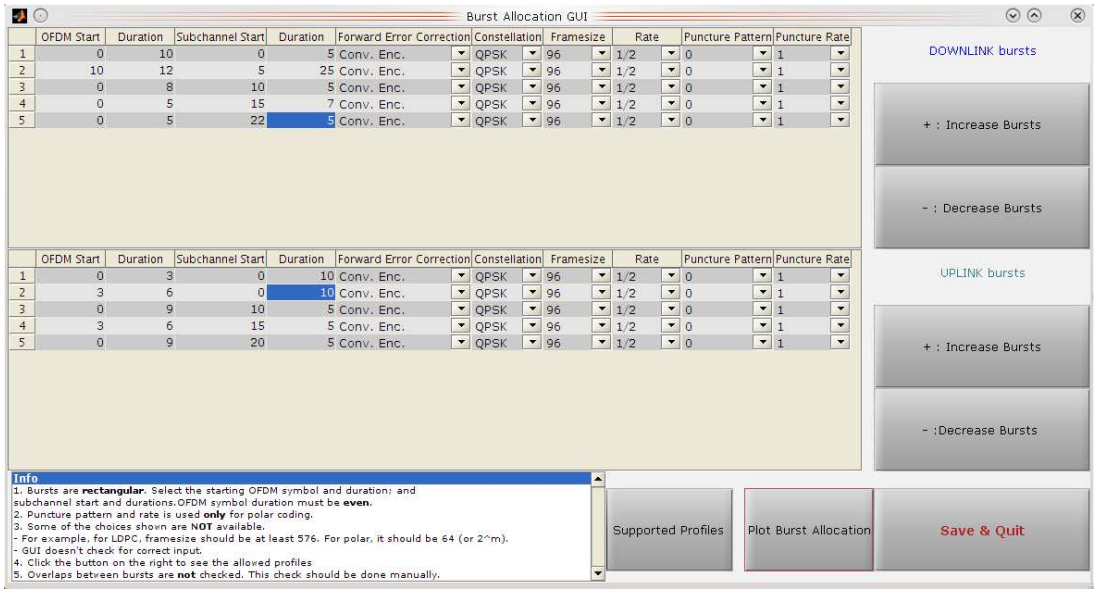

Figure 4.4: Screenshot of the Burst Allocation GUI

4.1.1.3 Channel Model

Several channel models have been integrated. AWGN Channel, Rayleigh fading channel and ITU Baseline channels with Pedestrian B, Modified Pedestrian B, Vehicular A and Modified Vehicular B scenarios are supported. For the Rayleigh channel, we assume that each OFDM subcarrier is multiplied by an independent channel coefficient.

The ITU-Baseline channel has been implemented by IASA, and integrated into our test suite by us. The non-modified channels use 6 taps while the modified versions use 24 taps.

Tables 4.2 and 4.3 give the power delay profiles and delays of taps associated with the modified channel versions. These values are used to sample the channel

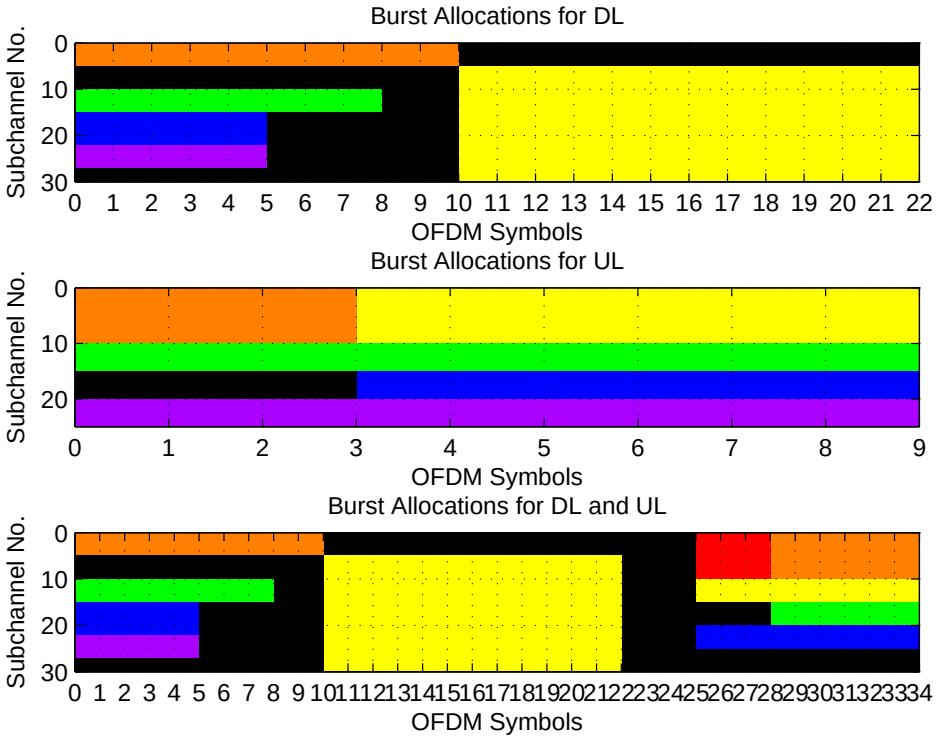

Figure 4.5: Manually Configured Burst Allocation

at the system sampling rate, and the output is normalized in order to generate the modified Pedestrian B and modified Vehicular A channels we are using.

We assume that we already know the SNR value of the AWGN channel so that demapper works correctly. Similarly, we assume that we have perfect knowledge of the channel if it is fading, so a zero-forcing equalizer is used in all such cases as defined in [12], i.e. for a complex channel factor H_i and received symbol R_i, the symbol input to the demapper is $\tilde{R}_i$ such that:

$$\tilde{R}_i = \frac{R_i}{H_i} = X_i + \frac{N_i}{H_i}$$

This is perhaps suboptimal in the sense that if the channel response is too low, it will amplify the additive noise, however this recovery fits our purposes. For a detailed discussion on channel estimation problems, the interested reader is encouraged to read [17].

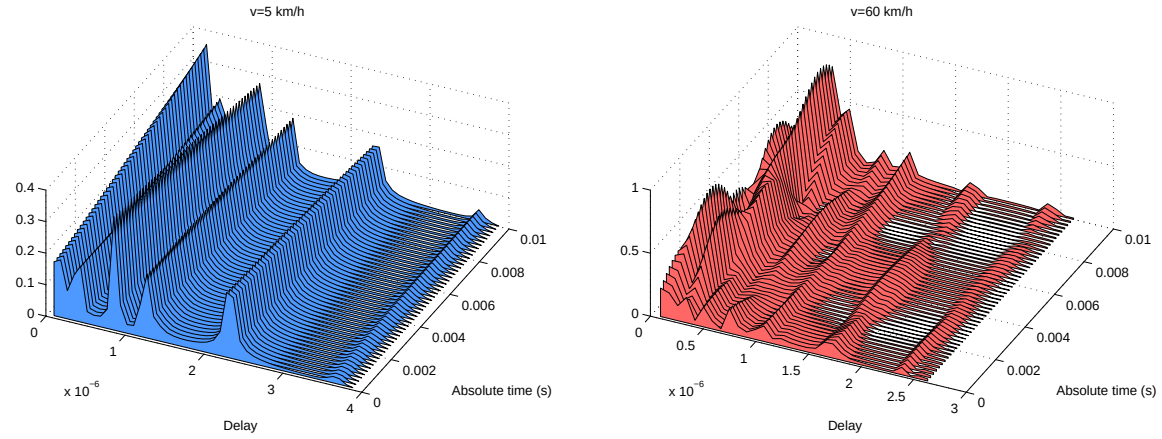

Figure 4.6: Channel instances for the Modified Pedestrian B Model (left) at $v = 5$ km/h and Modified Vehicular A Model (right) at $v = 60$ km/h

4.2 Simulation Environment

This section will give information on the simulator environment and the chosen simulation parameters.

4.2.1 Parameters of the Simulator

The following parameters are common for different burst profiles:

The parameters in Table 4.5, and their options are different for each burst profile:

Note that the FEC information bit length is actually not part of a burst profile, it is chosen according to the space allocated, however in our implementation, we have neglected the slot concatenation rule, and assumed that the FEC length is chosen for the burst profile, and the burst is filled such that as much FEC blocks with the selected code configuration are mapped. Remaining portions of bursts are filled with zeros.

Table 4.2: Power Delay Profile for the ITU Modified Pedestrian B Channel

Relative Path Power (db)	Delay (s)
-1.1750	0.0000
0	40e-009
-0.1729	70e-009
-0.2113	120e-009
-0.2661	210e-009
-0.3963	250e-009
-4.3200	290e-009
-1.1608	350e-009
-10.4232	780e-009
-5.7138	830e-009
-3.4798	880e-009
-4.1745	920e-009
-10.1101	1.20e-006
-5.6460	1.25e-006
-10.0817	1.31e-006
-9.4109	1.35e-006
-13.9434	2.29e-006
-9.1845	2.35e-006
-5.5766	2.38e-006
-7.6455	2.40e-006
-38.1923	3.70e-006
-22.3097	3.73e-006
-26.0472	3.76e-006
-21.6155	3.87e-006

To make the comparisons easier, we assign different configurations for different subscribers, but the channel characteristics (channel type, scenario chosen, and channel parameters) of the subscribers are actually the same.

4.2.2 Simulation Parameters

In this section, we give information regarding the parameter set chosen to implement the simulator. As mentioned before, IEEE 802.16 standards are full of choices, making the implementation hard; whereas the WiMAX certification chooses a subset of the available choices. Our choice selection has been mainly in the direction of WiMAX Forum's choices, however in some cases, we have

Table 4.3: Power Delay Profile for the ITU Modified Vehicular A Channel

Relative Path Power (db)	Delay (s)
-3.1031	0.0000
-416.60e-003	50e-009
0.0000	90e-009
-1.0065	130e-009
-1.4083	270e-009
-1.4436	3e-009
-1.5443	390e-009
-4.0437	420e-009
-16.6369	670e-009
-14.3955	750e-009
-4.9259	770e-009
-16.5160	8e-009
-9.2222	1.04e-006
-11.9058	1.06e-006
-10.1378	1.07e-006
-14.1861	1.19e-006
-16.9901	1.67e-006
-13.2515	1.71e-006
-14.8881	1.82e-006
-30.3480	1.84e-006
-19.5257	2.48e-006
-19.0286	2.50e-006
-38.1504	2.54e-006
-20.7436	2.62e-006

Table 4.4: Common parameters for Bursts

N_{FFT}	2048, 1024, 512
Channel Model	AWGN, Rayleigh, Modified Pedestrian B or Vehicular A
Cyclic Prefix	1/32, 1/16, 1/8, 1/4

Table 4.5: Individual Parameters for Each Burst

Constellation Mapping	QPSK, 16-QAM, 64-QAM
FEC Encoding	Convolutional Encoding, CTC, LDPC, Polar codes
FEC Information Bit Length	Varies for each FEC type
FEC Rate	Varies for each FEC type
Puncturing Pattern/Rate	Only available for polar code yet
OFDM symbol start/end	Rectangular bursts are supported
Logical subchannel start/end	Manually defined

implemented IEEE 802.16 only options such as LDPC encoding support since we already had the support via CML.

Some sections of the standard have been omitted for ease of implementation, and for their being unnecessary for our purposes. Namely, we have done the following simplifications:

- Frames are composed of only the subscriber bursts, namely we have not included the preamble, FCH, TTG or RTG. These are mainly used for synchronization and control purposes such as informing the subscriber stations of the chosen MCs. We already assume perfect knowledge on these.
- Similarly, although the locations for pilot carriers are reserved, we do not do any operation on the pilot carriers since we already assume that we have perfect synchronization and perfect channel knowledge.
- Although we are using PUSC, we have given the option of using the full spectrum range to the subscribers, i.e. we have not dealt with segmentation since it is more of a system-level topic.
- Although we have implemented FUSC too, since only PUSC is used in real scenarios, we have omitted integrating it in our simulator.
- Burst profiles are not chosen according to an AMC algorithm, but rather determined by the user. This is because in a link-level simulator with a simulation length of 1 frame, this is not meaningful. Our aim rather has been to see the link-layer performance through the iteration of various channel conditions of fixed burst profiles.
- Repetition coding is not supported.
- The standard specifies a final randomization step after the constellation mapping, namely a multiplication via a PRBS as we defined in Section 2.2.2.4. We have not implemented that part since the explanation

in the standard was not too detailed, and the effect of this is negligible for our purposes; as we already implemented equivalent mechanisms for randomization and interleaving of both the data bits and subcarriers.

- Of the MIMO schemes, we have implemented 2x1 and 2x2 Alamouti schemes for DL. For UL, support is optional, and as such, only SISO is used for simulations even though MIMO option is selected.

- The GUI gives burst profile choices via three different inputs: Namely the FEC choice, constellation size and FEC length. The command line interface however eases simulations via a predefined burst configuration selection which makes comparisons easier. For example, a burst profile of A1 in the command interface maps to choosing CC as a FEC option, 48 bits as the FEC length and ratio 1/2. The order of these burst configations are given in the Appendix B, and simply defining as burst parameters A1,B1,C1,D1,E1 for example gives the user the opportunity to compare different profiles easily. Since this approach is easier, we have only integrated the polar code support in the command line interface.

- Cyclic prefix support exists, but is disabled since it does not change performance. Similarly, conversion of the OFDMA waveform to time domain is not performed, instead the channel parameters are transformed to frequency domain.

- HARQ support does not exist since simulations are run for a single frame. It is more of a system level issue and we assume full buffer.

4.2.2.1 Primitive and Derived Parameters

Primitive parameters specified in the 802.16e standard and 802.16m EMD documentation will be introduced in this subsection.

Table 4.6: Sampling factors for Different Bandwidths

n	Bandwidth (Multiple of)
8/7	1.75 MHz
28/25	1.25, 1.5, 2, 2.75 MHz
8/7	None of the above

Previously named 802.16e, WirelessMAN-OFDMA PHY is developed for non-line of sight (NLOS) operation, unlike its predecessor 802.16d. The specification supports multiple FFT sizes, starting from 128 point FFT, up to powers of 2 till 2048. The FFT size is modified in order to keep the subcarrier spacing fixed, which is 10.94 kHz. As a result, the OFDMA in WirelessMAN-OFDMA is SOFDMA, which stands for Scalable OFDMA.

In WiMAX, FFT size is configured according to the bandwidth allocated. For a bandwidth of 20 MHz, an FFT size of 2048 is chosen. Similarly, for a bandwidth of 10 MHz, the FFT size is halved to 1024 points, and for a bandwidth of 5 MHz, the FFT size is 512. Although WiMAX supports bandwidths multiples of 1.25, 2 or 2.75 MHz, other compatible systems such as the WiBro in South Korea support bandwidths multiple of 1.75 MHz. In all cases however, the sampling factor is modified according to Table 4.6 to keep the subcarrier spacing fixed and the corresponding FFT sizes are chosen in each case.

Once the OFDMA frame at a scheduling instant is filled, it is converted to time domain via inverse Fourier transform (IFFT). A cyclic prefix is added to mitigate intersymbol interference, and make circular convolution possible.

The primitive parameters used in the simulations are listed below as:

- Bandwidth (BW): Nominal bandwidth. We assume a bandwidth of 10 MHz in our simulations unless stated otherwise.

- Number of used subcarriers: 1680 for 2048-FFT, 840 for 1024-FFT. For a bandwidth of 10 MHz, we use 1024-FFT.
- Sampling factor : This parameter is modified in order to make the subcarrier spacing as specified. See Table 4.6.
- Cyclic prefix ratio: This determines the length of the cyclic prefix added from the end of the signal to the beginning. Supported values for WirelessMAN-OFDMA are 1/32, 1/16, 1/8 and 1/4.

Based on primitive parameters, various other parameters might be derived. These include the following:

- FFT Size: This is the smallest power of 2 greater than used subcarriers. It is an exponent of 2 between 128 and 2048. The FFT size is dependent on the bandwidth, for example, for 10 Mhz, we use 1024 FFT.
- Sampling frequency (f_{sampling})
- Subcarrier spacing : This is kept constant for different FFT sizes.
- Cyclic prefix time (T_g)
- OFDMA symbol time (T_{OFDMA})
- Sampling time

The formulation for the items above are given in [1, Section 8.4.2.4].

Important parameters derived for different FFT sizes are given in Table 4.7. T_{useful} determines the length of OFDM symbol with no cyclic prefix. G is the cyclic prefix ratio, 1/8 in all cases; T_g is the length of the cyclic prefix interval and T_{OFDMA} is the length of the OFDMA symbol with cyclic prefix. Overall maximum data rates are 3.168e+06 bps, 1.584e+07 bps, 3.168e+07 bps and 6.336e+07 bps respectively for $N_{\text{FFT}} = 128$, 512, 1024 and 2048.

Table 4.7: OFDMA Parameters for Different FFT Sizes

N_{FFT}	f_{sampling} (Hz)	Subcarrier Spacing (Hz)	T_{useful} (s)	G	T_g (s)	T_{OFDMA} (s)	Symbols per frame
128	1.400e+06	1.094e+04	9.143e-05	1.250e-01	1.143e-05	1.029e-04	48
512	5.600e+06	1.094e+04	9.143e-05	1.250e-01	1.143e-05	1.029e-04	48
1024	1.120e+07	1.094e+04	9.143e-05	1.250e-01	1.143e-05	1.029e-04	48
2048	2.240e+07	1.094e+04	9.143e-05	1.250e-01	1.143e-05	1.029e-04	48

4.2.3 Simulation Assumptions

In SISO case, the channel is assumed to be fixed for a OFDM symbol. Therefore, we neglect intercarrier interference and assume the subscribers experience flat fading at each subcarrier.

For MIMO, we have implemented both the 2 transmit - 2 receive antenna, and 2 transmit - 1 receive antenna schemes. WiMAX does not support the 2 transmit - 2 receive scheme, however this functionality is added to compare the results with those for IEEE 802.16 m and the WiMAGIC project, which uses 2x2 scheme instead of 2x1 scheme.

Although we have implemented MIMO 2x2, in our results section, we only analyze the 2x2 scheme since it brings a much better performance improvement.

As stated before, the Alamouti scheme requires that the channel is fixed for two subsequent symbols. Since our simulations involve OFDM symbols, this requirement translates as the channel being fixed for two subsequent OFDM symbols.

4.2.4 Simulation Performance Metrics

In our simulations, our primary performance metric will be BER vs Signal-to-Noise(SNR) and FER vs SNR performance curves. For comparisons, we perform inital experiments for each coding scheme with SISO in AWGN channel.

Recall that, as Shannon proved, in an AWGN channel, the rate of reliable data transmission (also known as the capacity) is upperbounded as in the following inequality[15]:

$$R_T \leq W \log_2 \left(1 + \frac{P}{N_o W}\right)$$
$$\eta \leq \log_2 \left(1 + \frac{E_b}{N_o}\eta\right)$$

where $\eta \triangleq \frac{R_T}{W}$ and called the *spectral efficiency* and we make use of the following of primary variables:

- $N_o/2$ is the (two-sided) power spectral density of noise per degree of freedom in Watts/Hz.
- W is the channel bandwidth in Hz.
- P is the average transmitter power (over all degrees of freedom) in Watts.
- R_T is the rate of transmission in bits/sec.

and E_b denotes the energy per bit, defined as:

$$E_b \triangleq P/R_T$$

Rearranging the final inequality, with other variable replacements yields:

$$\frac{E_b}{N_o} \geq \frac{2^\eta - 1}{\eta}$$

This constitutes the Shannon limit on $\frac{E_b}{N_o}$ for a given spectral efficiency η, similar to the Shannon limit on rate, commonly known as *capacity*. The question then becomes one of getting the value of η for a given code configuration candidate, i.e. a given rate and modulation.

Note that for a modulation and coding scheme (MCS) with coding rate R (k/n –dimensionless) and 2^m-ary signal constellation:

$$R_T = mRD$$

where D is the number of modulation symbols per second. For 1-dimensional modulation schemes, such as Pulse Amplitude Modulation (PAM) and Binary Phase Shift Keying (BPSK), $D = 2W$. For 2-dimensional modulation such as Quadrature Phase Shift Keying (QPSK) and Quadrature Amplitude Modulation (QAM), $D = W$.

Thus,

$$\eta = \frac{R_T}{W} = \frac{2mR}{d}$$

where d is the dimension of the modulation scheme ($d = 1$ for PAM, $d = 2$ for QAM).

Hence, for a given code rate R and a chosen modulation order m, one can calculate η and then calculate the Shannon limit on E_b/N_o.

In our simulations, we have chosen rates of 1/2 and 3/4 and constellation sizes of 4 and 16, as a subset of the WiMAX standard. The Shannon limits on E_b/N_o for these cases in an AWGN channel are given below:

Rate	Modulation	η	Shannon Limit on E_b/N_o
1/2	QPSK	1	1.0 (0 dB)
3/4	QPSK	3/2	2.16 (3.35 dB)
1/2	16-QAM	2	3.5 (5.44 dB)
3/4	16-QAM	3	7.67 (8.85 db)

Incidentally, the Shannon limit on E_b/N_o is the optimal limit a given code can approach; and is the place at which the code should have a waterfall behaviour.

So, the expected behaviour of a perfect code would be to be erroneous until this limit is achieved, and then once this limit is surpasses, immediately correct all the errors observed.

This is obviously not easily achievable in practice, however one important observation that can be made about this limit is that, it is actually around this limit point that codes tend to start performing better, and their downward slope increases.

Chapter 5

SIMULATION RESULTS AND ANALYSIS

In this chapter, we present our simulation results. Our implementation task has been two fold, namely first to implement a complete, end-to-end IEEE 802.16e simulator chain; and second to integrate polar codes as a FEC option to see how this new coding option fares with well-known, state-of-the art FEC options. Since WiMAX specification has selected only CTC and CC as FEC options, and since CTC is known to perform better than CC, we have selected to perform comparison simulations comparing mainly CTC and polar codes. Some simulation results, however, compare the coded simulation results with uncoded simulation results in order to get a reference point. We also investigate how changing the MIMO scheme changes performance, and the effect of PUSC for modified Pedestrian B and Vehicular A channels.

The sections simulations are presented, are formed as follows:

1. We start with the SISO cases. For the SISO case, we first consider the cases for which the modulation is QPSK.

2. For the SISO case with a modulation of QPSK, we first analyze the cases for which the rate is 1/2.

3. For those cases, we do simulations for four different channel types: AWGN, flat-fading Rayleigh, modified Pedestrian B and modified Vehicular A channels.

4. For each case, we present the Bit Error Rate (BER) vs E_b/N_o, and Frame (Code Block) Error Rate (FER) vs E_b/N_o performance curves.

5. We repeat Step 4 for a code rate of 3/4.

6. We repeat Steps 3-4 for a modulation of 16 QAM.

7. We repeat Steps 2-4 for the MIMO choice of 2x2.

Throughout the simulations, for a given modulation choice and code rate, we have compared four different configurations: Two CTC code configurations with different codelengths, and two other polar code configurations with different codelengths. The code configurations and code lengths therefore only differ for different rates.

The equivalent polar code lengths for their CTC counterparts are found by finding the power of two nearest to the CTC code length, since no puncturing is done and polar codes allow codelengths of powers of two. Therefore, the code lengths are not identical, but very similar and the code rates are preserved. In each case, we select the CTC configuration with the longest codelength and another with a shorter codelength in order to identify the effect of codelength on the performance.

The code choices employed for QPSK Rate 1/2 and Rate 3/4 cases are given in Tables 5.1 and 5.2 respectively:

Table 5.1: Code Configurations for QPSK Modulation and Rate 1/2

Code Type	Code Rate	Information Length (bits)	Coded Block Length (bits)	Spectral Efficiency (η)	Shannon Limit on E_b/N_o (dB))
CTC	1/2	240	480	1	0
CTC	1/2	480	960	1	0
Polar	1/2	256	512	1	0
Polar	1/2	512	1024	1	0

Table 5.2: Code Configurations for QPSK Modulation and Rate 3/4

Code Type	Code Rate	Information Length (bits)	Coded Block Length (bits)	Spectral Efficiency (η)	Shannon Limit on E_b/N_o (dB)
CTC	3/4	216	288	3/2	0.8599
CTC	3/4	432	576	3/2	0.8599
Polar	3/4	192	256	3/2	0.8599
Polar	3/4	384	512	3/2	0.8599

For the 16 QAM cases, same code configurations as in QPSK cases are used. The only differences are the spectral efficiency η and Shannon limit on E_b/N_o for these cases. These configurations are presented in Tables 5.3-5.4.

Table 5.3: Code Configurations for 16 QAM Modulation and Rate 1/2

Code Type	Code Rate	Information Length (bits)	Coded Block Length (bits)	Spectral Efficiency (η)	Shannon Limit on E_b/N_o (dB)
CTC	1/2	240	480	2	1.7609
CTC	1/2	480	960	2	1.7609
Polar	1/2	256	512	2	1.7609
Polar	1/2	512	1024	2	1.7609

5.1 SISO Results

We first give the simulation results in a SISO setting under various channel conditions and various code configurations as specified in the WiMAX standard.

Table 5.4: Code Configurations for 16 QAM Modulation and Rate 3/4

Code Type	Code Rate	Information Length (bits)	Coded Block Length (bits)	Spectral Efficiency (η)	Shannon Limit on E_b/N_o (dB)
CTC	3/4	216	288	3	3.6798
CTC	3/4	432	576	3	3.6798
Polar	3/4	192	256	3	3.6798
Polar	3/4	384	512	3	3.6798

We first give the results with QPSK modulation, and then give the results for 16-QAM modulation.

5.1.1 Results with QPSK Modulation in a SISO Setting

In this subsection, we present the results of the simulations in which QPSK modulation is used.

We first give the performance results for codes with rates equal to 1/2 in Figures 5.2–5.9. Then the simulation results with rates equal to 3/4 are plotted in Figures 5.10–5.19.

For QPSK modulation, for a rate of 1/2, we observe that the performance gap between CTC and equivalent polar codes is around 2 dB at a FER error rate of 10^{-3} in an AWGN channel.

The performance gap is around 3 dB for the same FER error rate in Rayleigh channel.

For the modified Pedestrian B channel with a $v = 5km/h$, the performance gap is similarly around 2-3 dBs.

For the modified Vehicular A channel with a $v = 60km/h$, the performance gap is similarly around 3-4 dBs.

We also observe that since we assume the channel coefficients are constant in two OFDM symbols for both Pedestrian B and Vehicular A channels, the performance of codes in these two channels are very similar.

5.1.1.1 Comparison of Codes with Coding Rate = 1/2

In this part, we present the results of our simulations of the same code configurations, under different channel conditions, with code rates fixed to 1/2, the modulation fixed to QPSK and the antenna setting is SISO.

However, first, for reference purposes we start by presenting the results with uncoded simulation results overlayed on top of coded configuration simulations in Figure 5.1.

Figures 5.2 through 5.9 present the BER vs E_b/N_o and FER vs E_b/N_o curves for each of the four channels, namely the AWGN channel, Rayleigh channel, Modified Pedestrian B Channel and finally Modified Vehicular A channel.

Similarly, before presenting the Rayleigh channel results by themselves, we present the uncoded and coded simulation results under Rayleigh channel in Figure 5.4 on page 70.

We observe that in all cases in this section, the CTC codes outperform polar codes by more than a few dBs, and usually the higher the code length, the better the performance.

For clarity, the coded configuration results of the same simulation are presented again without the uncoded simulation results, and with the Shannon limit on E_b/N_o as a reference point in Figure 5.2.

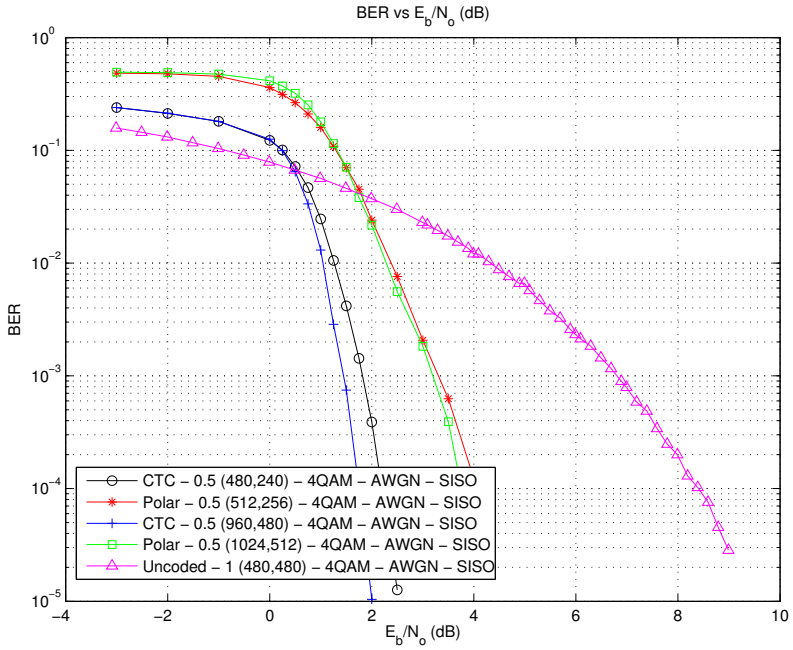

Figure 5.1: Comparison of Coded (Rate = 1/2) and Uncoded Schemes Under AWGN Channel with QPSK Modulation

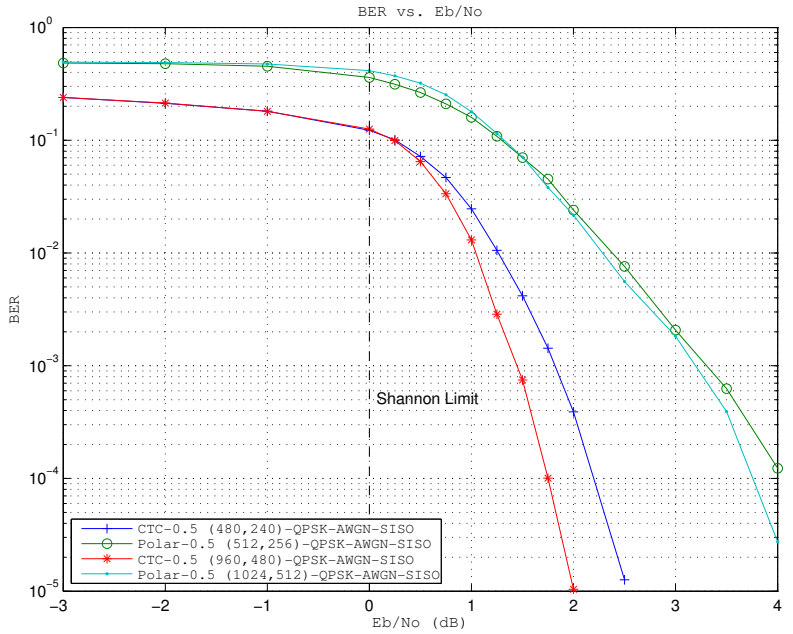

Figure 5.2: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with QPSK Modulation

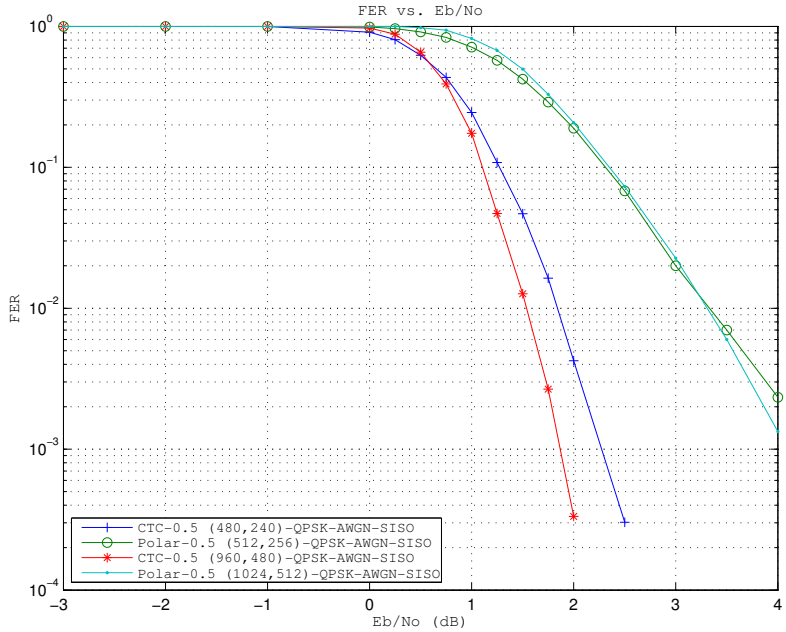

Figure 5.3: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with QPSK Modulation

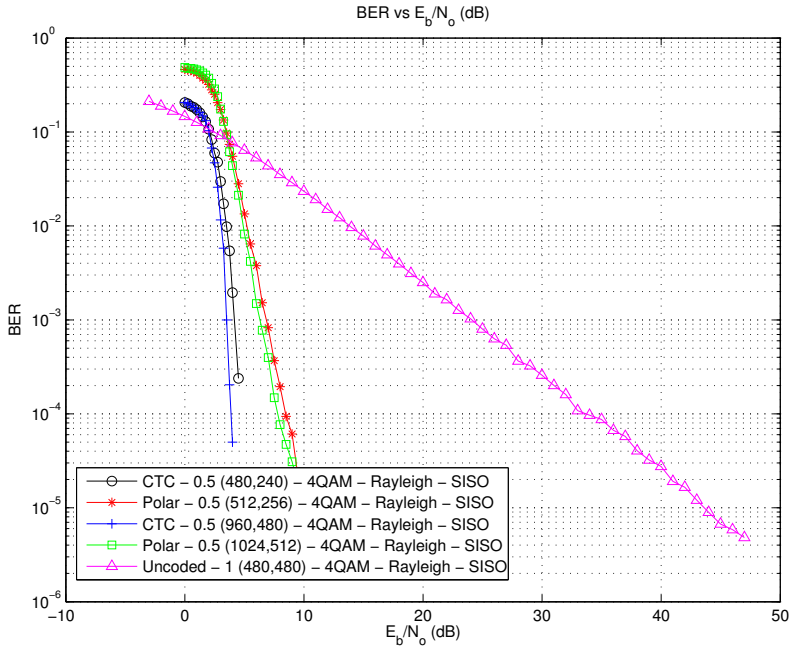

Figure 5.4: Comparison of Coded (Rate = 1/2) and Uncoded Schemes Under Rayleigh Channel with QPSK Modulation

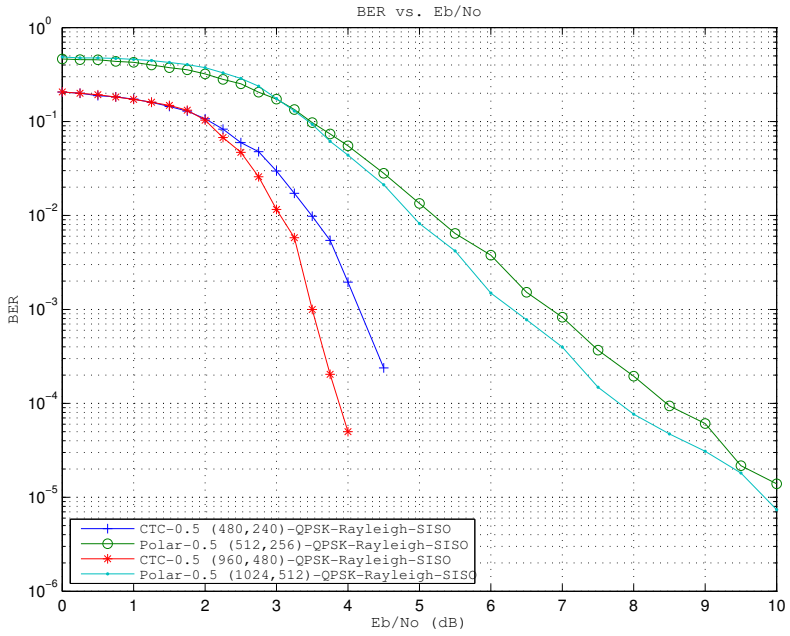

Figure 5.5: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with QPSK Modulation

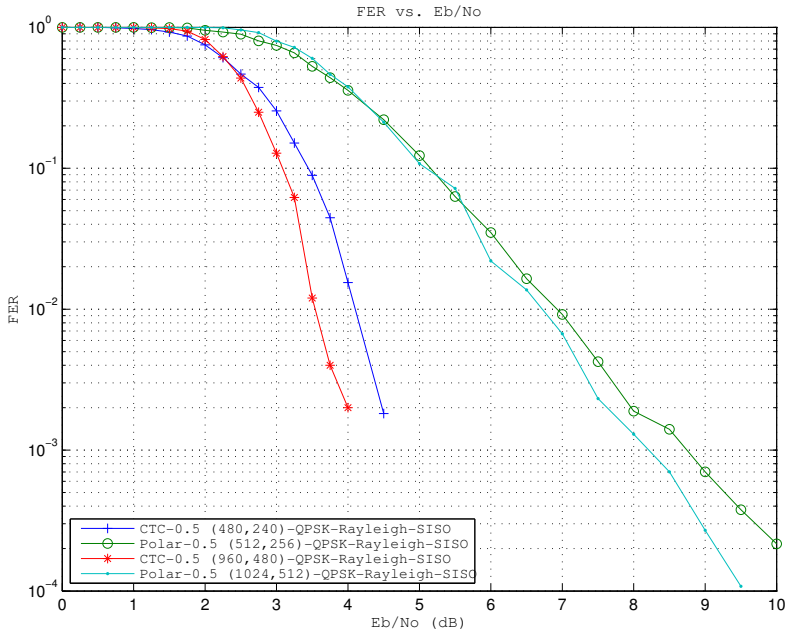

Figure 5.6: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with QPSK Modulation

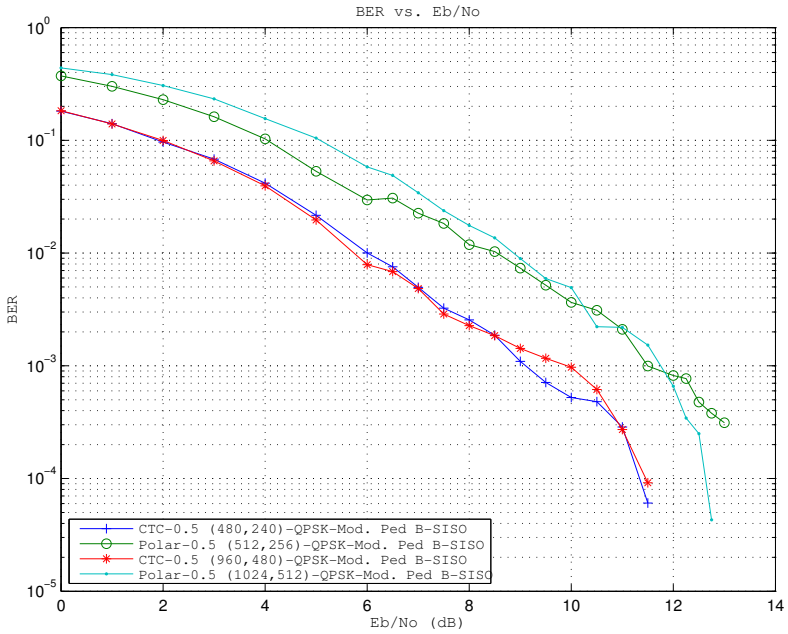

Figure 5.7: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with QPSK Modulation

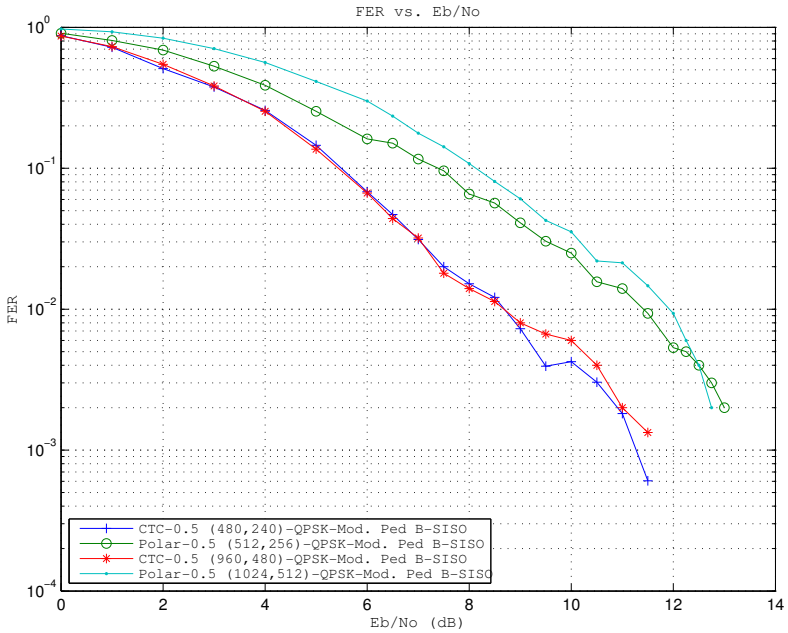

Figure 5.8: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with QPSK Modulation

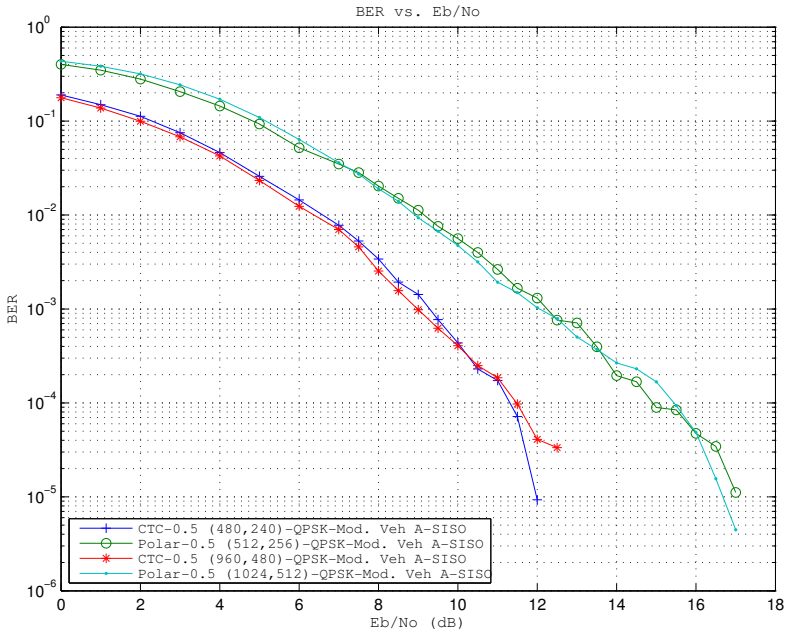

Figure 5.9: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with QPSK Modulation

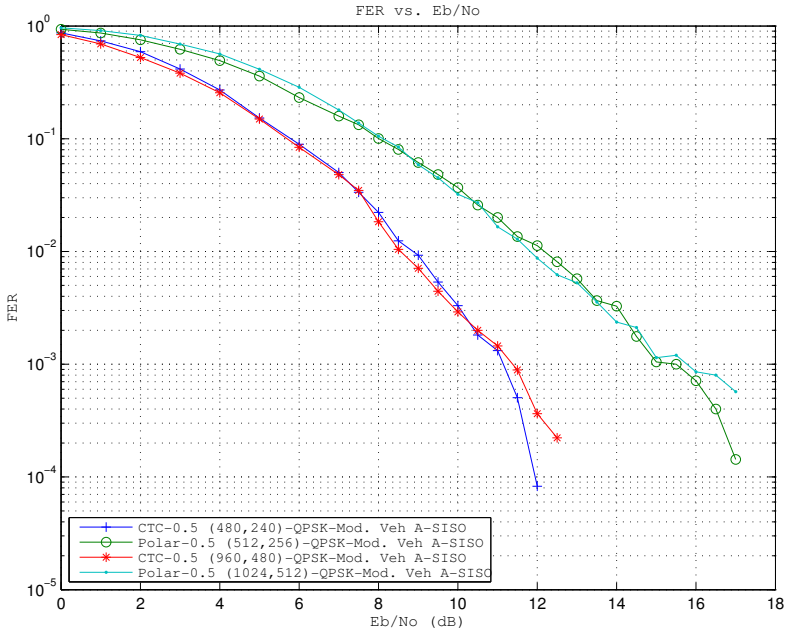

Figure 5.10: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with QPSK Modulation

5.1.1.2 Comparison of Codes with Coding Rate = 3/4

Similar to previous part, we present the results of our simulations of the same code configurations, under different channel conditions, with code rates fixed to 3/4, the modulation fixed to QPSK and the antenna setting is SISO.

However, first, for reference purposes we start by presenting the results with uncoded simulation results overlayed on top of coded configuration simulations in Figure 5.11.

Figures 5.12 through 5.19 present the BER vs E_b/N_o and FER vs E_b/N_o curves for each of the four channels, namely the AWGN channel, Rayleigh channel, Modified Pedestrian B Channel and finally Modified Vehicular A channel.

The performance results are very similar to that for the rate 1/2 cases. CTCs outperform polar codes by more than 2-3 dBs for each of the channels.

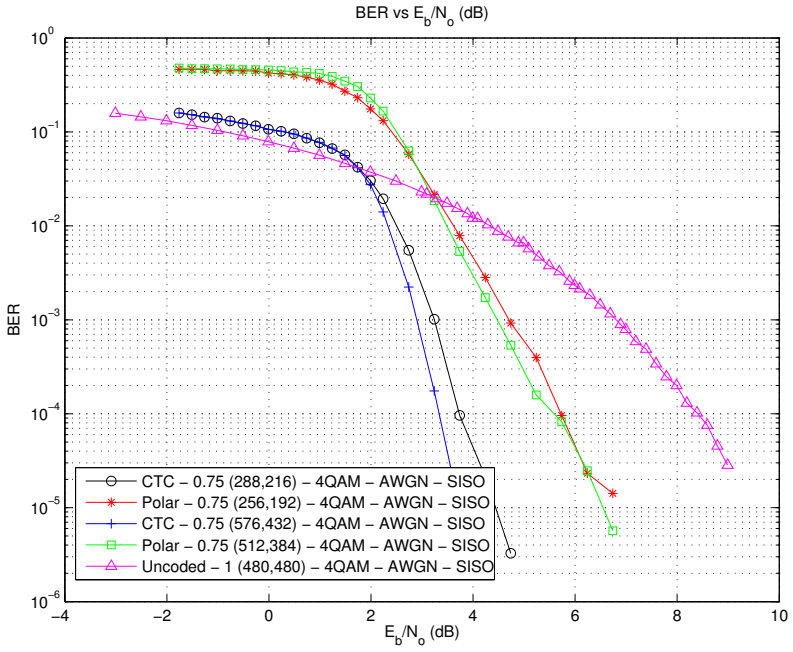

Figure 5.11: Comparison of Coded (Rate = 3/4) and Uncoded Schemes Under AWGN Channel with QPSK Modulation

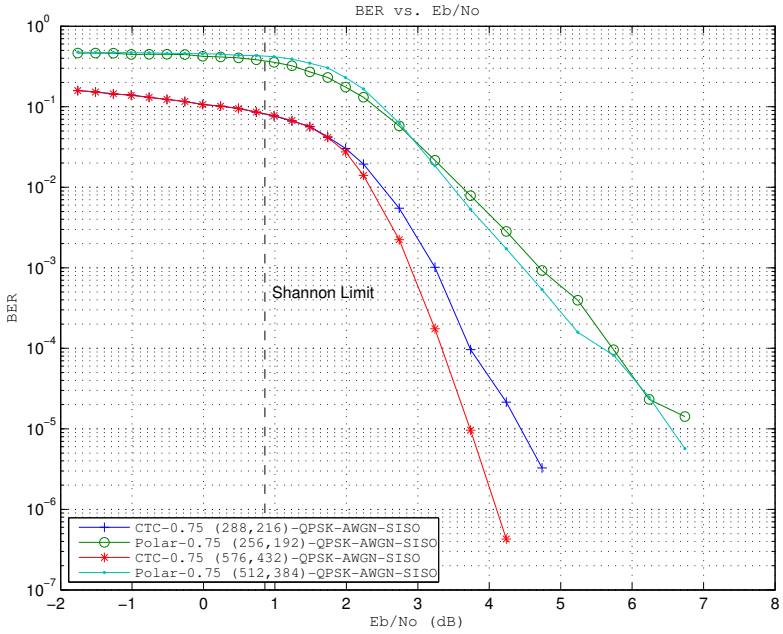

Figure 5.12: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with QPSK Modulation

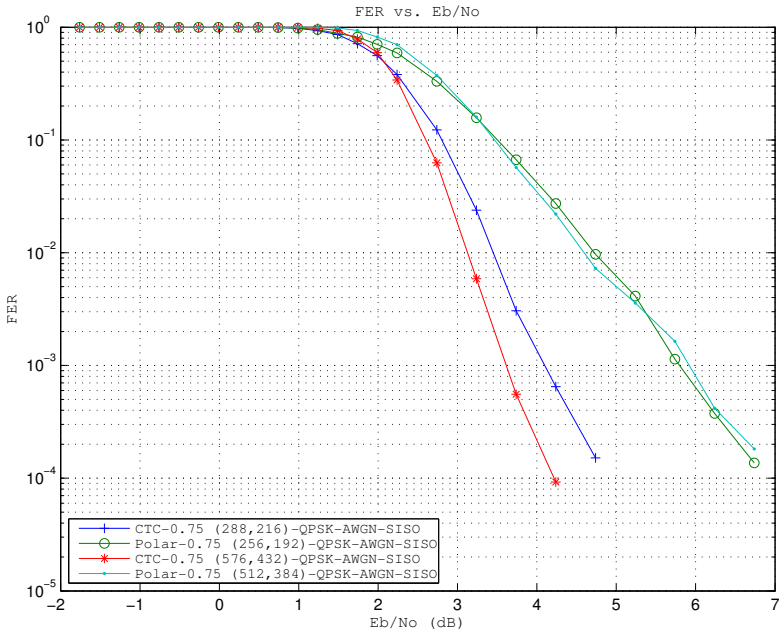

Figure 5.13: Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with QPSK Modulation

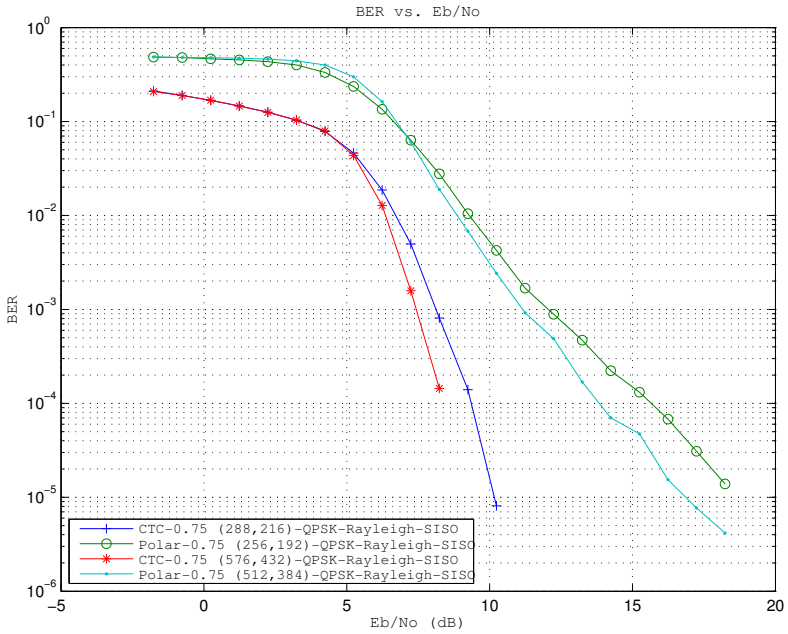

Figure 5.14: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with QPSK Modulation

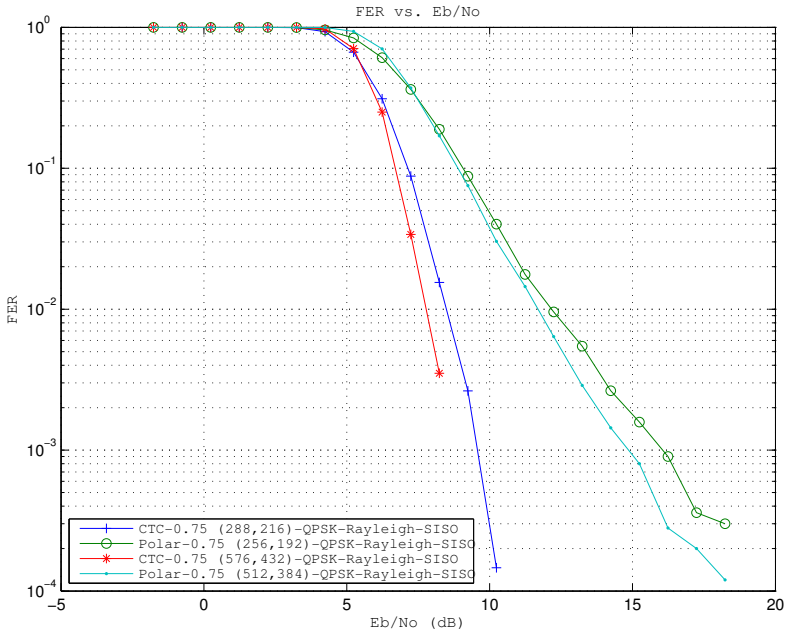

Figure 5.15: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with QPSK Modulation

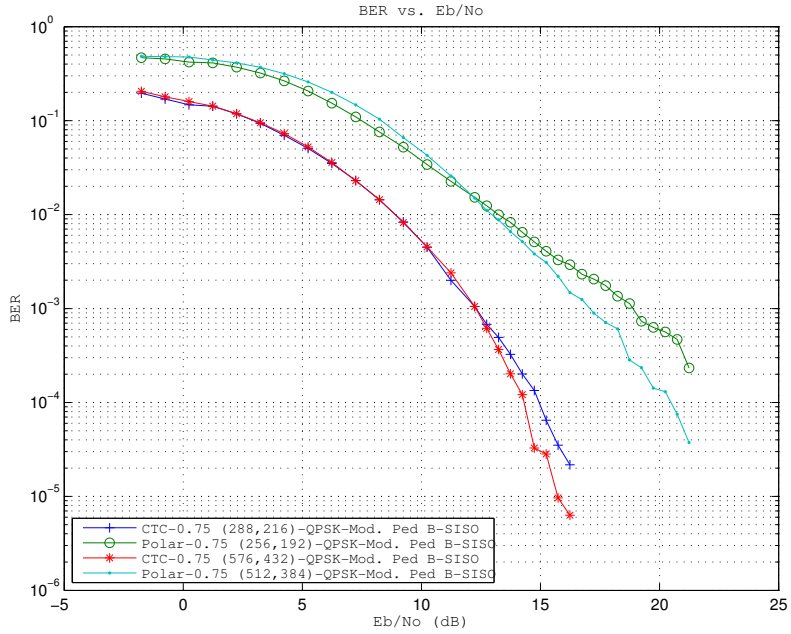

Figure 5.16: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with QPSK Modulation

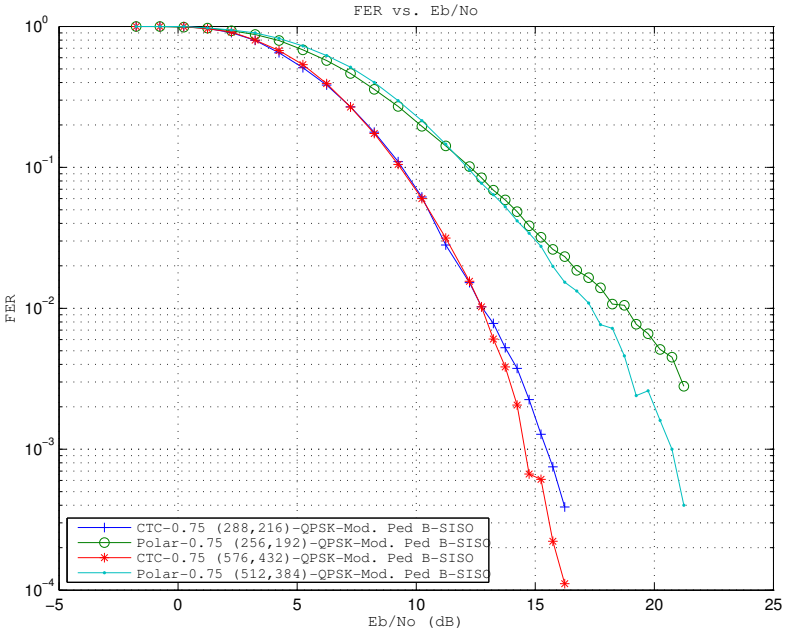

Figure 5.17: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with QPSK Modulation

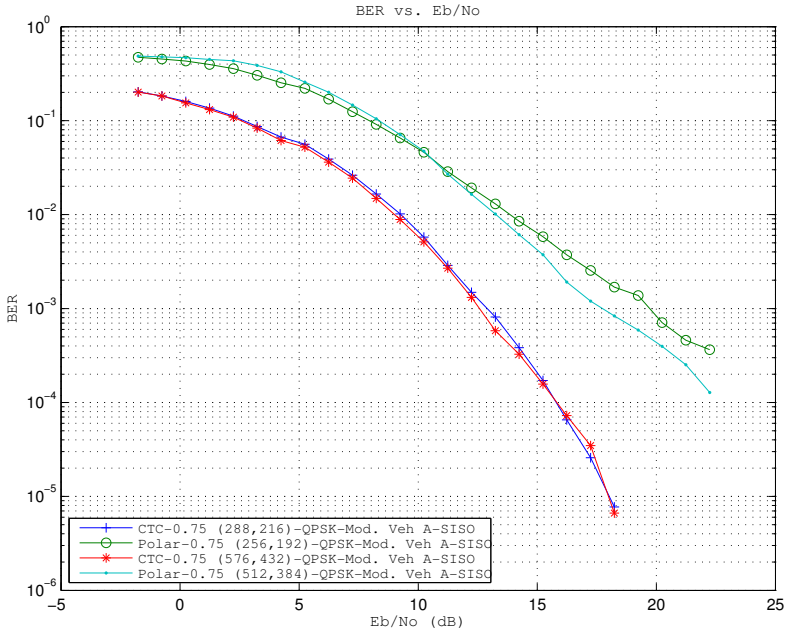

Figure 5.18: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with QPSK Modulation

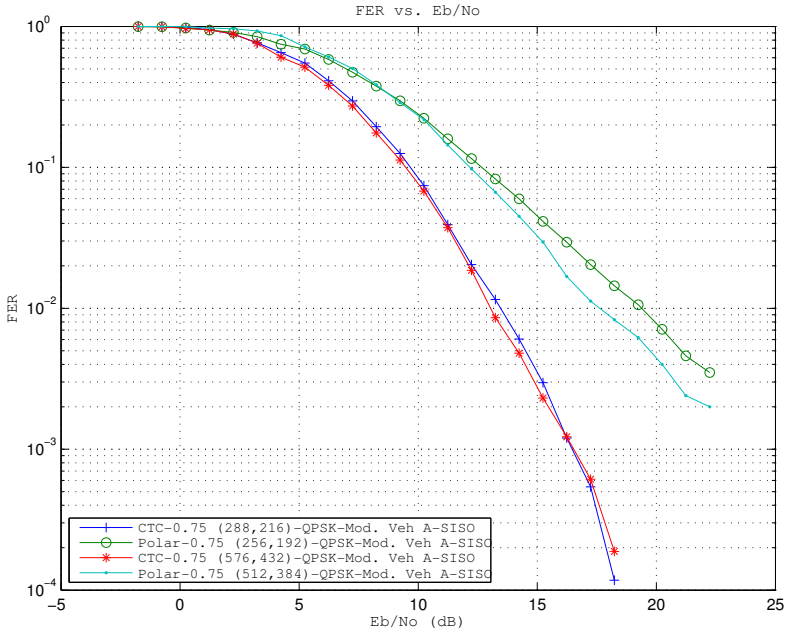

Figure 5.19: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with QPSK Modulation

5.1.2 Results with 16-QAM Modulation in a SISO Setting

In this subsection, we present the results of the simulations in which 16-QAM modulation is used.

For 16-QAM modulation, for a rate of 1/2 for a FER of 10^{-2}, the performance gap is 2 dB in an AWGN channel.

For 16-QAM modulation, for a rate of 1/2 for a FER of 10^{-2}, the performance gap is 3 dB in other (fading) channels.

5.1.2.1 Comparison of Codes with Coding Rate = 1/2

In this subsection, we present the results of our simulations for a SISO antenna configuration, a modulation choice of 16 QAM and FEC code configurations of rate 1/2.

We observe that the performance curves are shifted to the right with respect to the QPSK cases, in an amount denoting the difference between the Shannon limits on E_b/N_0 cases. Other than that, the performance curves feature resemblance to their QPSK counterparts, with the CTC code configurations outperforming their polar equivalents by a few dBs for both BER and FER performances.

5.1.2.2 Comparison of Codes with Coding Rate = 3/4

In this subsection, we present the results of our simulations for a SISO antenna configuration, a modulation choice of 16 QAM and FEC code configurations of rate 3/4. Since the spectral efficiency η is highest in these configurations, the Shannon limit on E_b/N_o reaches the highest point for our simulations. This is observed by a shift to the right for all cases. Other than this shift, the results

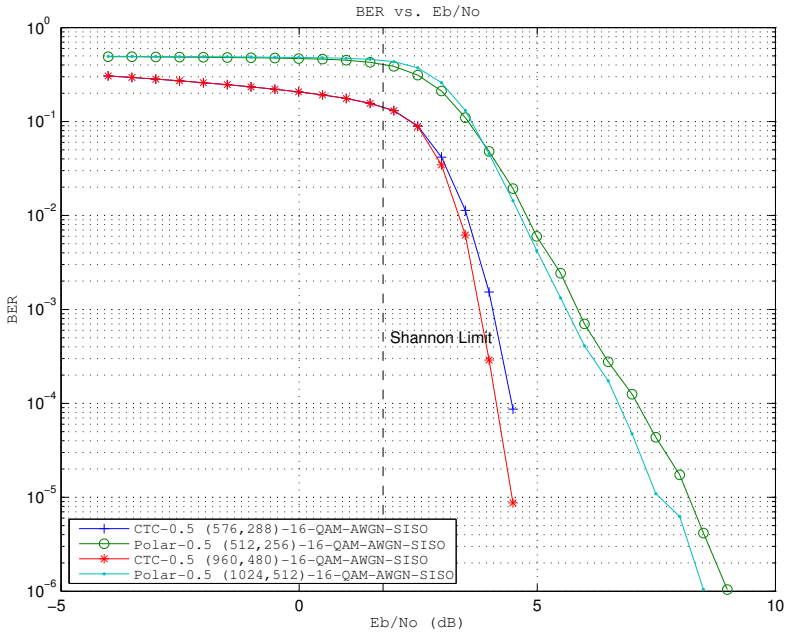

Figure 5.20: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with 16-QAM Modulation

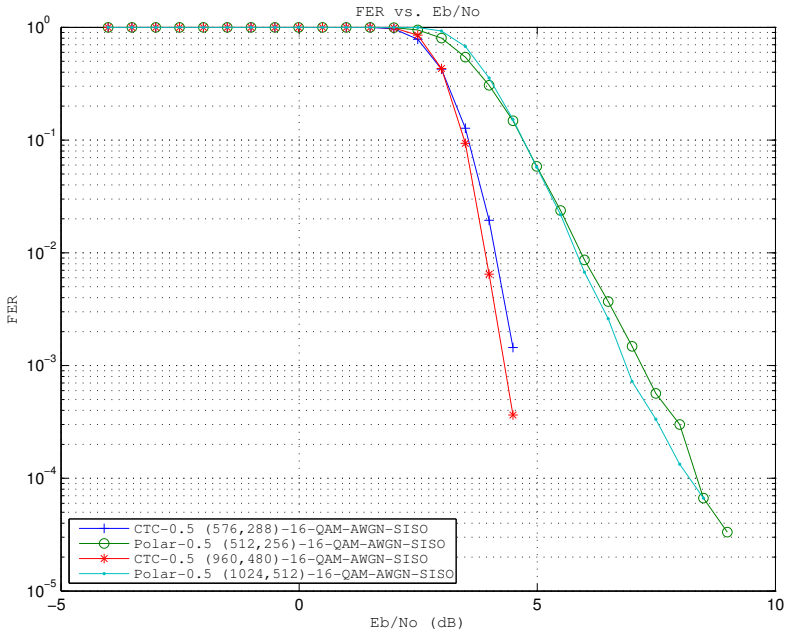

Figure 5.21: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with 16-QAM Modulation

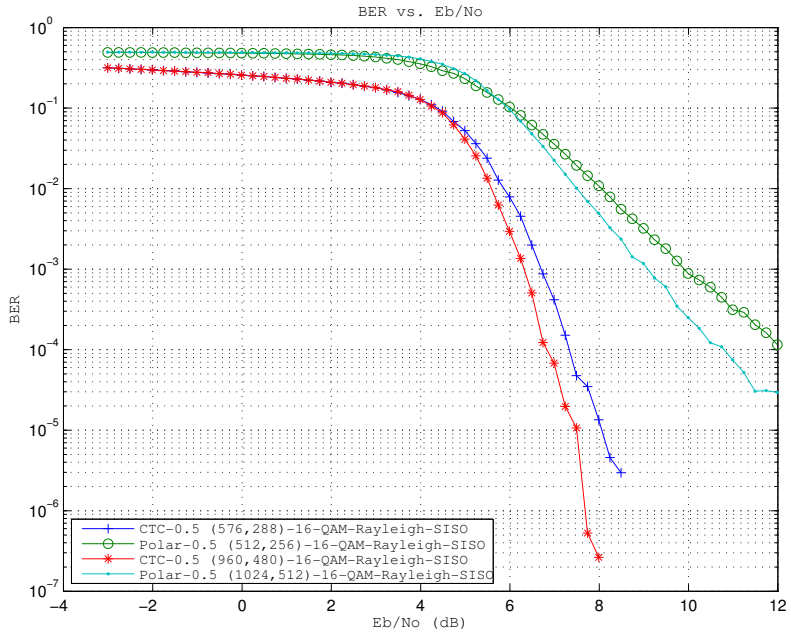

Figure 5.22: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with 16-QAM Modulation

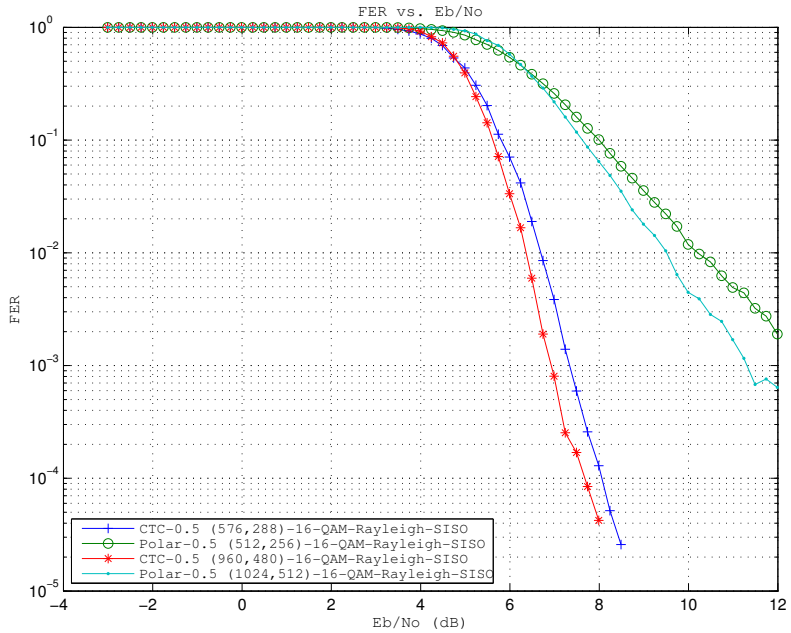

Figure 5.23: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with 16-QAM Modulation

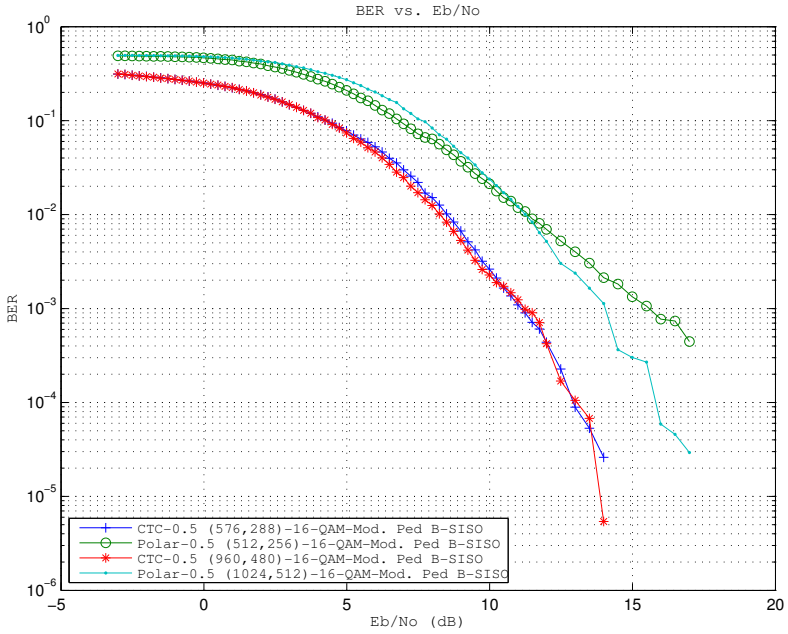

Figure 5.24: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with 16-QAM Modulation

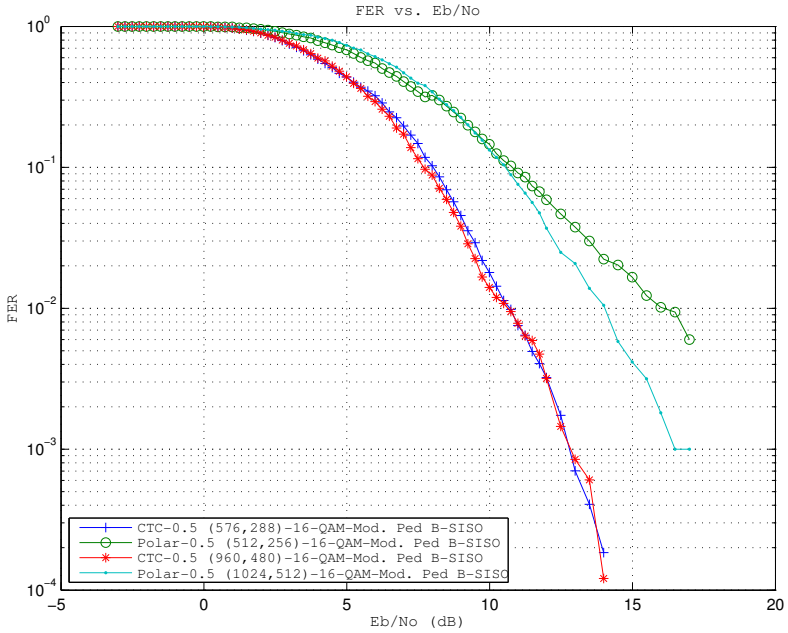

Figure 5.25: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with 16-QAM Modulation

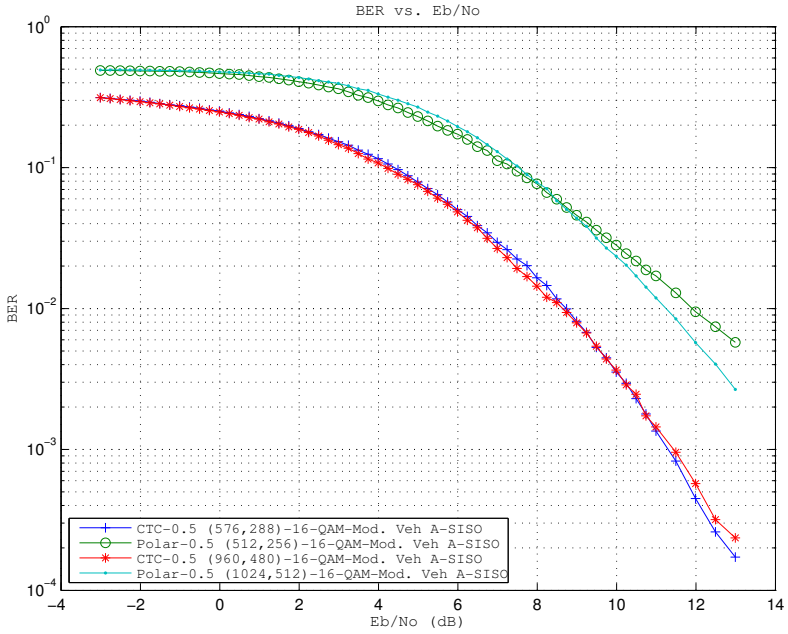

Figure 5.26: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with 16-QAM Modulation

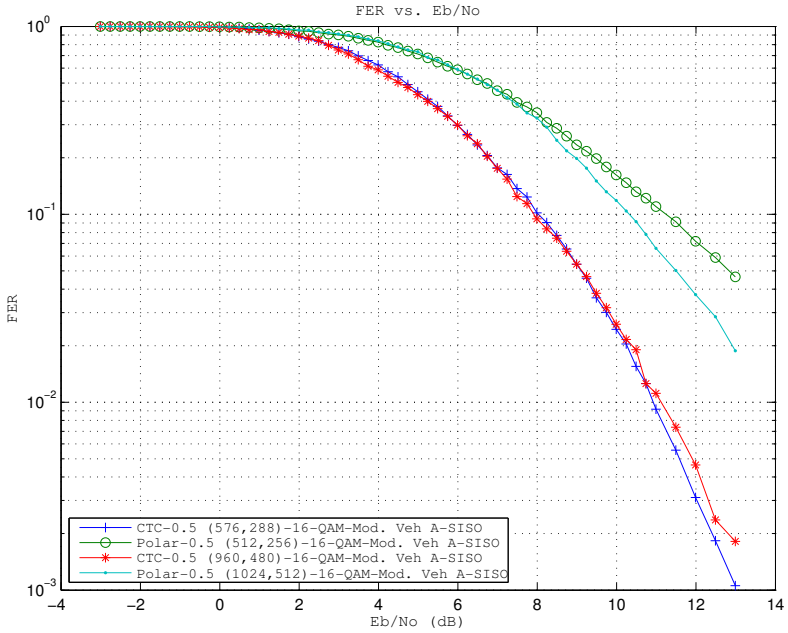

Figure 5.27: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with 16-QAM Modulation

are pretty much similar to the previous sections, with CTC outperforming the polar codes in all cases.

Our particular interesting observation might be that, for CTC codes, the effect of codelength in performance is as expected in almost all cases: The longer the codelength, the better. For polar codes, however, this is not always the case: For the AWGN channel, we observe a better performance from the longer code, as shown in Figure 5.28.

For the Rayleigh channel, this observation is no longer that strong, as shown in Figure 5.30; where we see that increasing the codelength slightly increases the BER performance, however yet still slightly decreases the FER performance. However, overall, we might say that the performances are almost equal.

Finally, for the modified Pedestrian B and Vehicular A channels however, we see in Figures 5.32 - 5.35 that increasing the codelength with polar codes causes an observable decrease in code performance. This might be explained by the wrong selection of frozen bit positions since the selection is done with a BEC channel assumption, and since the decoding is very sensitive to wrong decisions on that set, the error propagates from the wrongly selected significant bits to lower bits, resulting in an overall worse performance for fading channels.

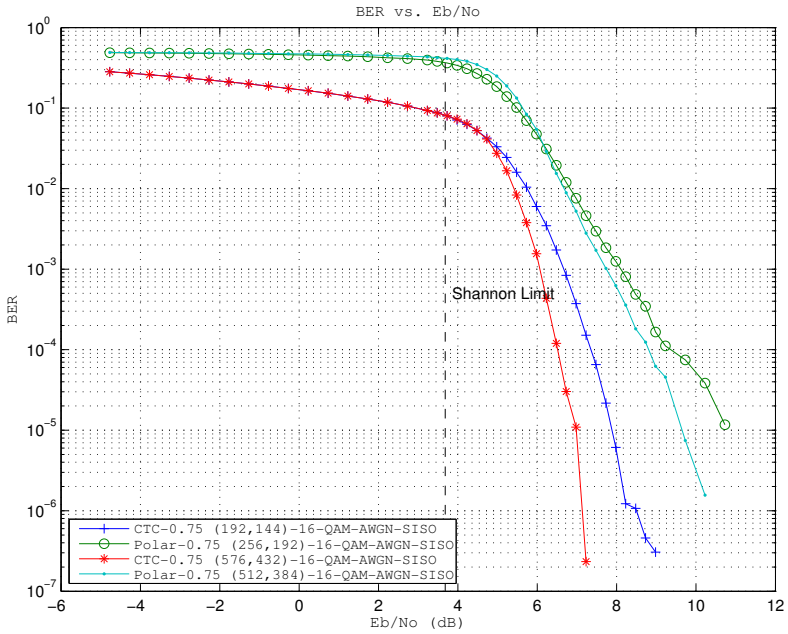

Figure 5.28: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with 16-QAM Modulation

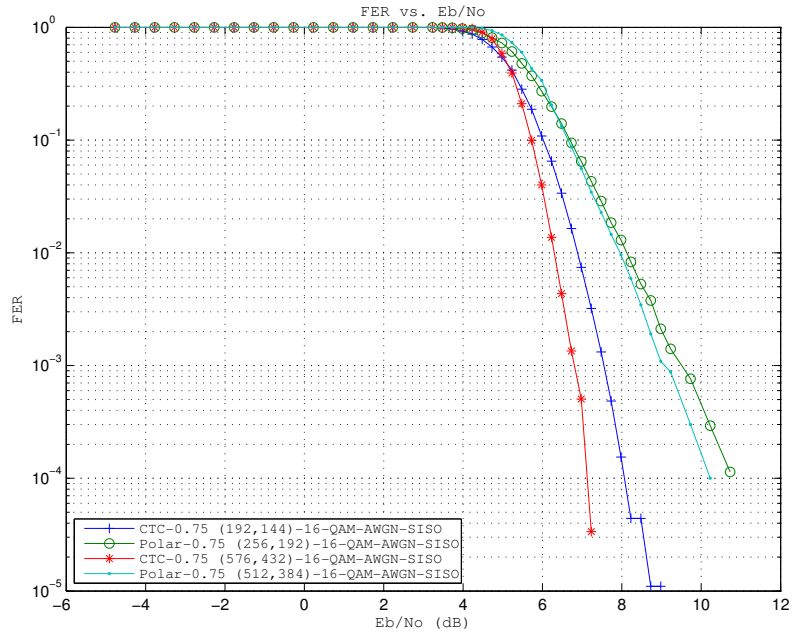

Figure 5.29: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with 16-QAM Modulation

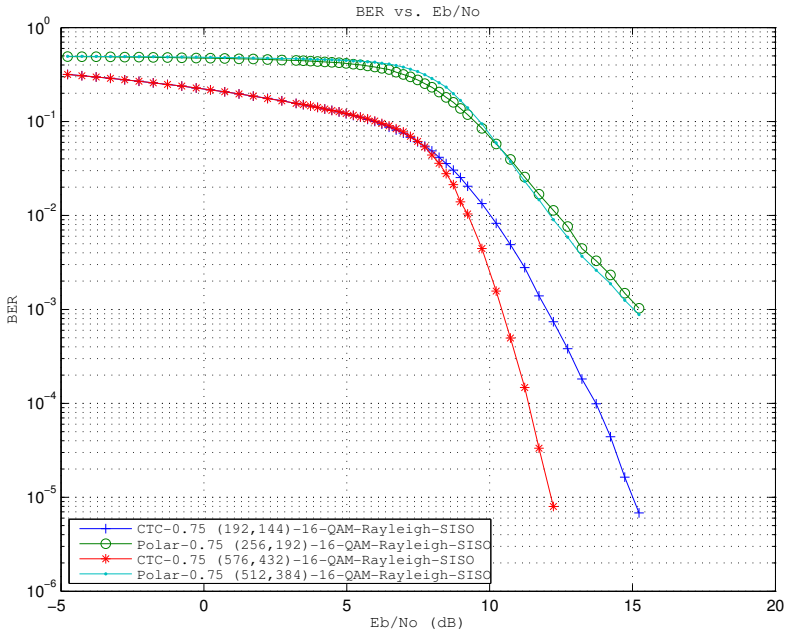

Figure 5.30: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in an AWGN Channel in a SISO Setting, with 16-QAM Modulation

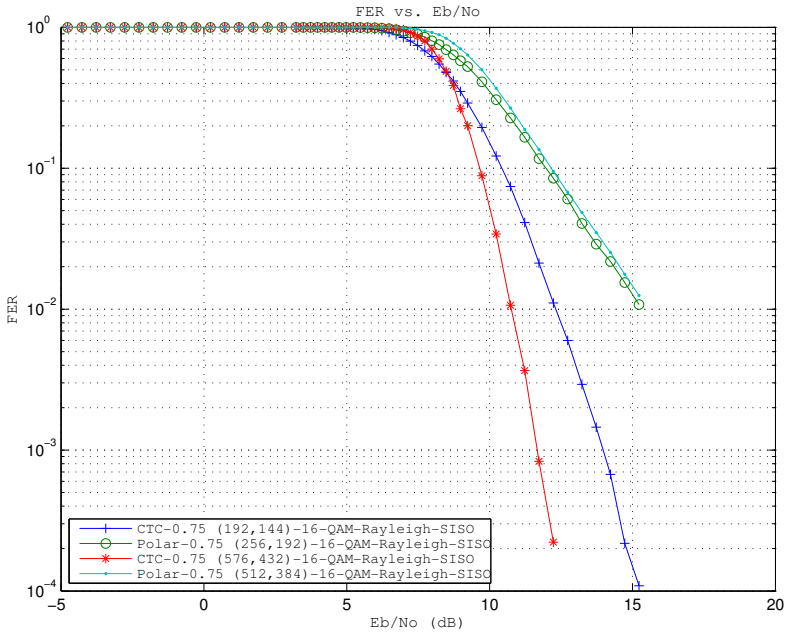

Figure 5.31: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a SISO Setting, with 16-QAM Modulation

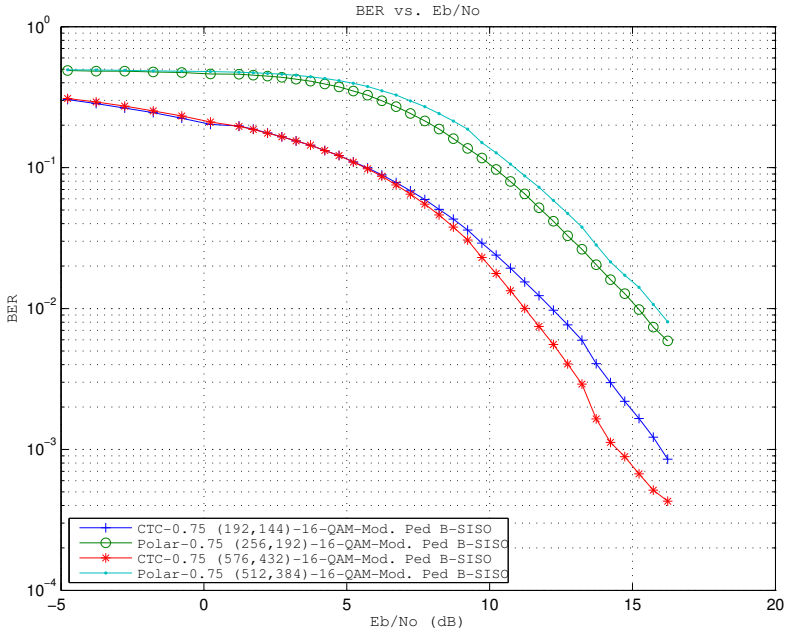

Figure 5.32: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with 16-QAM Modulation

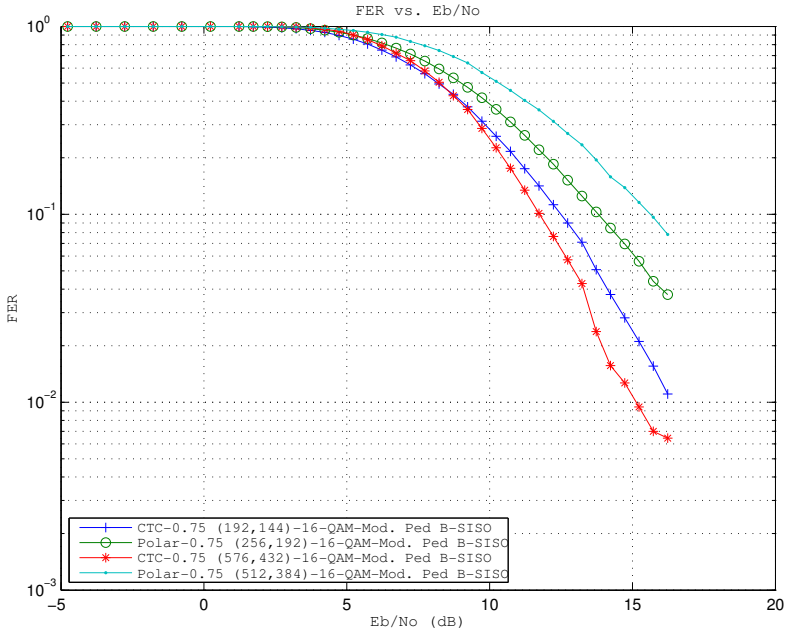

Figure 5.33: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Pedestrian B Channel ($v = 5km/h$) in a SISO Setting, with 16-QAM Modulation

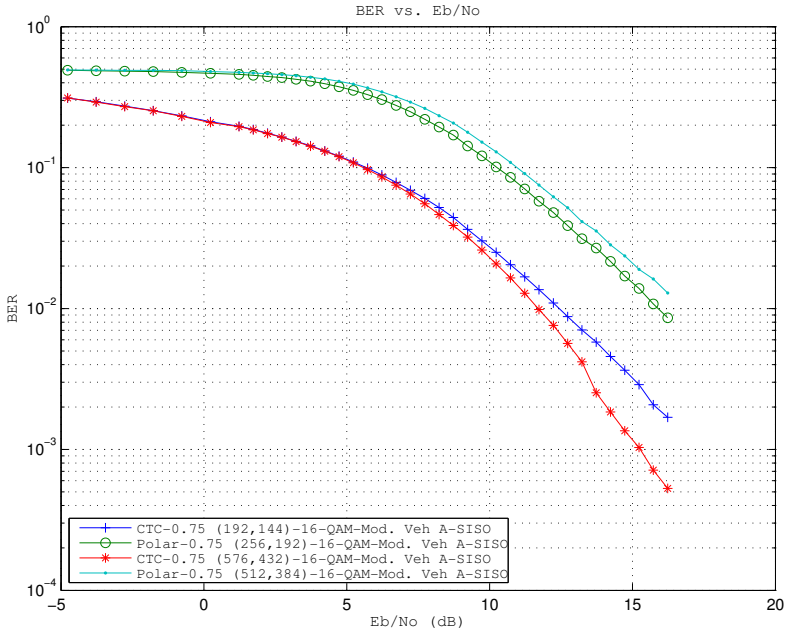

Figure 5.34: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with 16-QAM Modulation

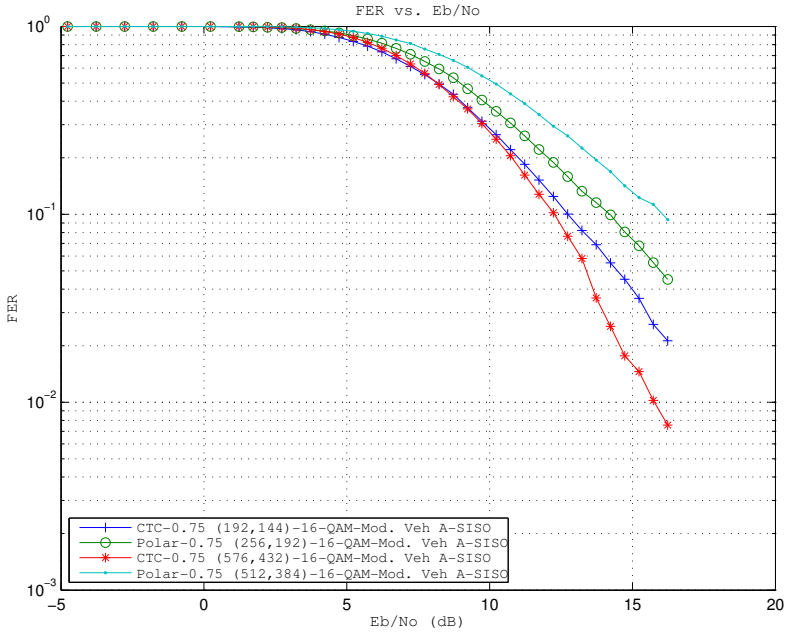

Figure 5.35: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Modified Vehicular A Channel ($v = 60km/h$) in a SISO Setting, with 16-QAM Modulation

5.2 MIMO 2x2 Results

In this section, the same simulations as in previous sections are done, except that we change the antenna configuration to MIMO 2x2. The performance increase is evident in all cases; and the observations stated in the previous sections hold true in all the sub-cases we are considering. Therefore, it suffices to say that, under all scenarios we have simulated with MIMO 2x2 configuration, we have observed that CTCs outperform polar codes by a few dB margins, and we will not restate those observations for each individual case.

The general trend that CTC codes perform better than polar codes continues for this set of simulations. We observe that with both QPSK and 16-QAM modulations, and with rates 1/2 and 3/4, the performance gap between CTC codes and polar codes is in the order of a few dBs.

The effect of increase in performance with increasing codelength is more observable for the MIMO case, possibly due to the increase in the inherent reliability of the frozen positions for polar codes due to using two channels instead of one.

5.2.1 Results with QPSK Modulation in a MIMO 2x2 Setting

5.2.1.1 Comparison of Codes with Coding Rate = 1/2

In this subsection, we present the results of our simulations for a MIMO 2x2 antenna configuration, a modulation choice of QPSK and FEC code configurations of rate 1/2. Figures 5.36 and 5.37 give the results under Rayleigh channel, while Figures 5.38- 5.41 show the results of simulations under modified Pedestrian B and Vehicular A channels.

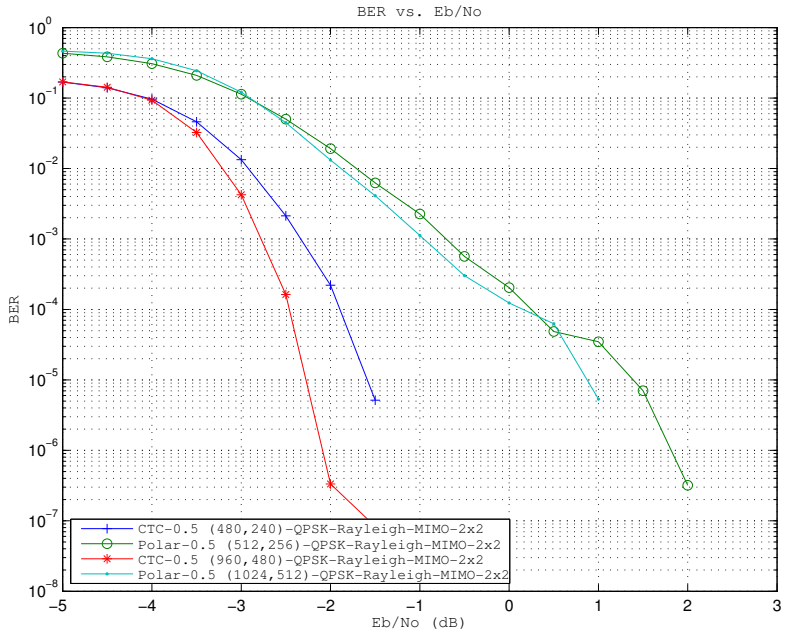

Figure 5.36: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with QPSK Modulation

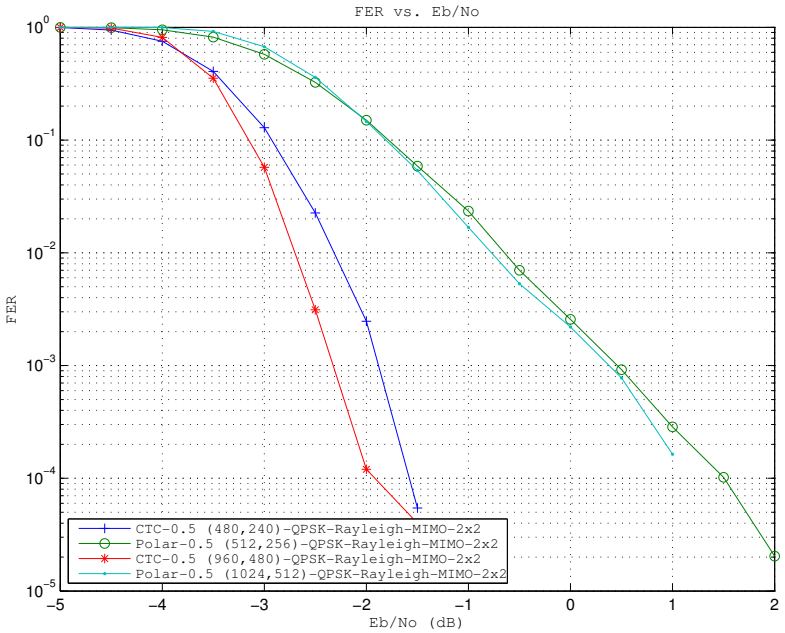

Figure 5.37: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with QPSK Modulation

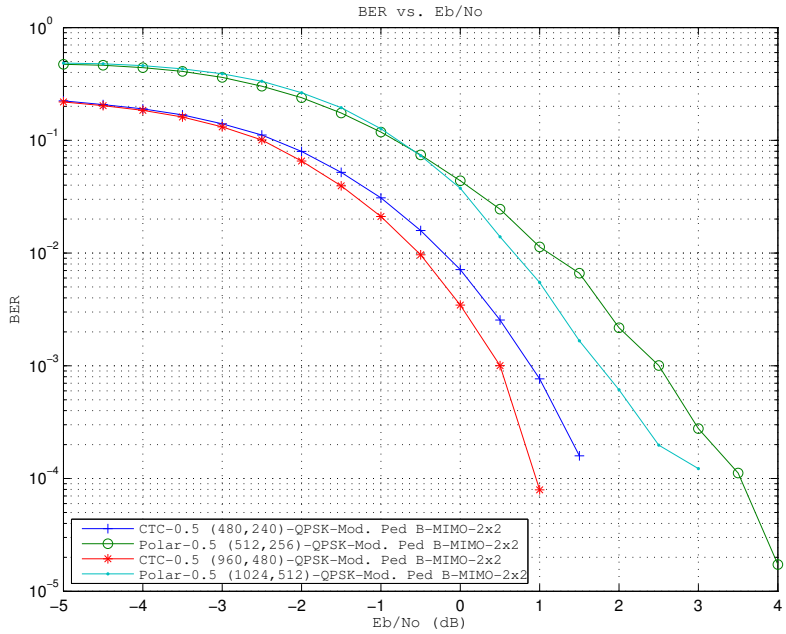

Figure 5.38: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with QPSK Modulation

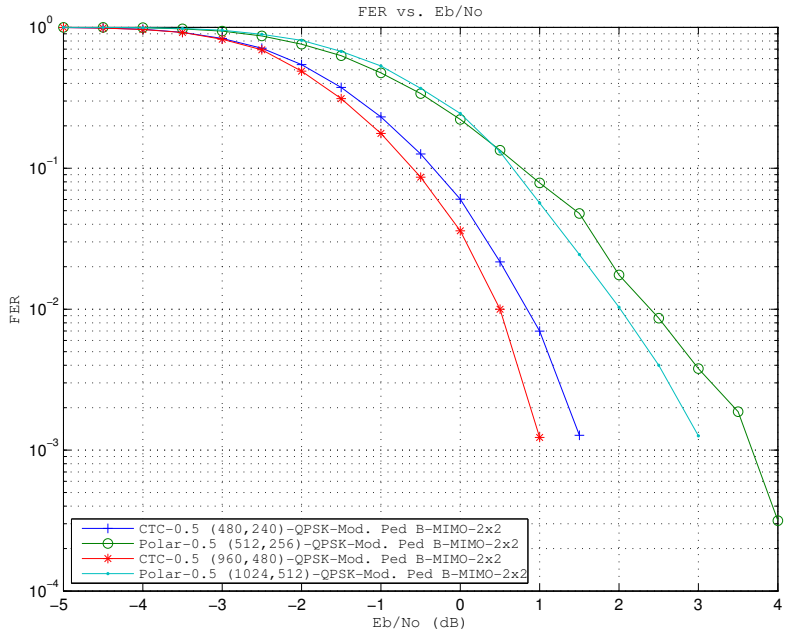

Figure 5.39: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with QPSK Modulation

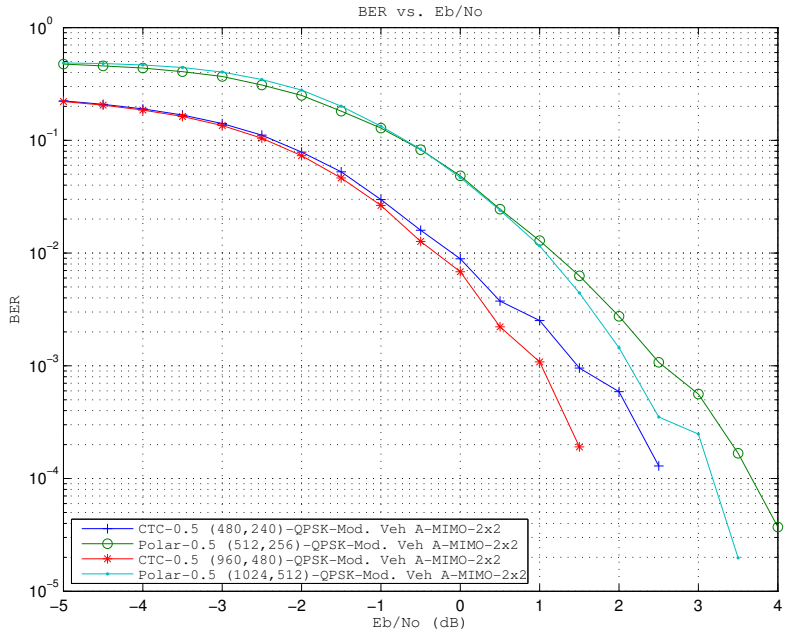

Figure 5.40: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with QPSK Modulation

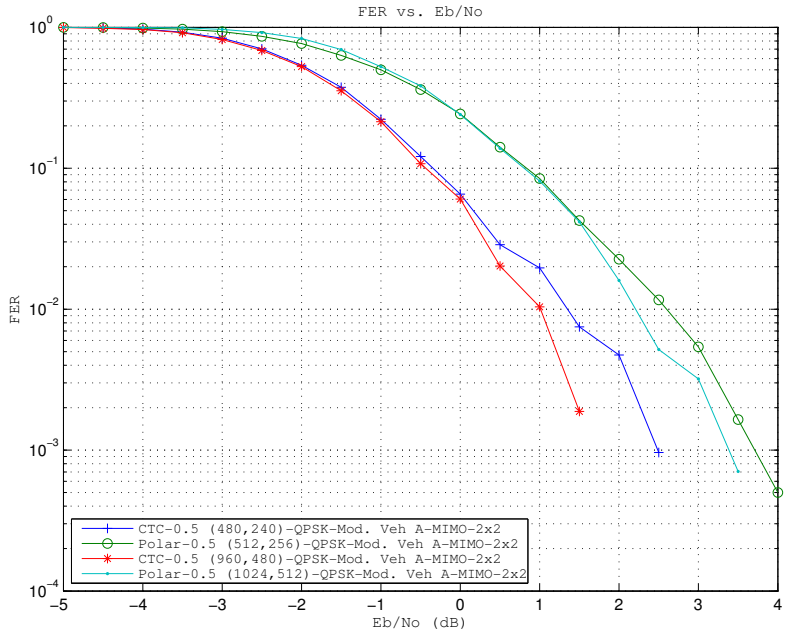

Figure 5.41: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with QPSK Modulation

5.2.1.2 Comparison of Codes with Coding Rate = 3/4

In this subsection, we present the results of our simulations for a MIMO 2x2 antenna configuration, a modulation choice of QPSK and FEC code configurations of rate 3/4. Figures 5.42 and 5.43 give the results under Rayleigh channel, while Figures 5.44- 5.47 show the results of simulations under modified Pedestrian B and Vehicular A channels.

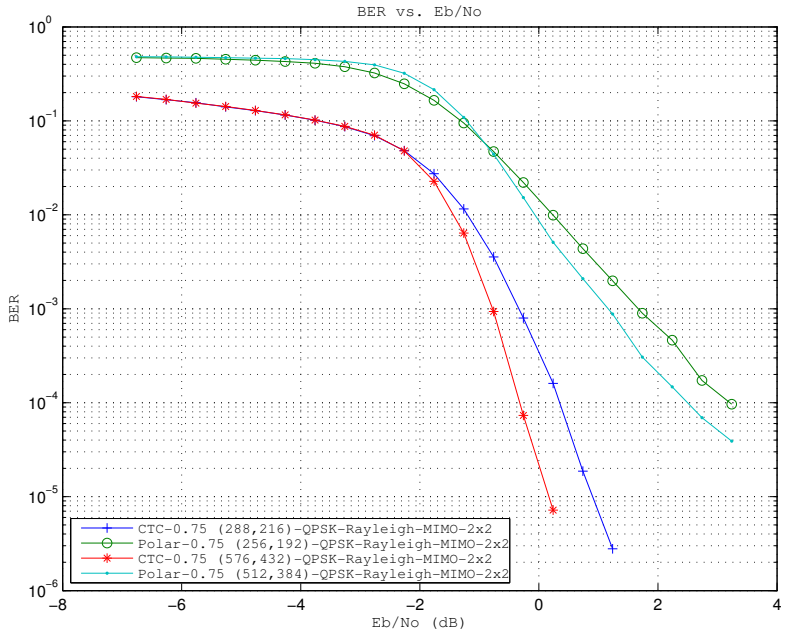

Figure 5.42: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with QPSK Modulation

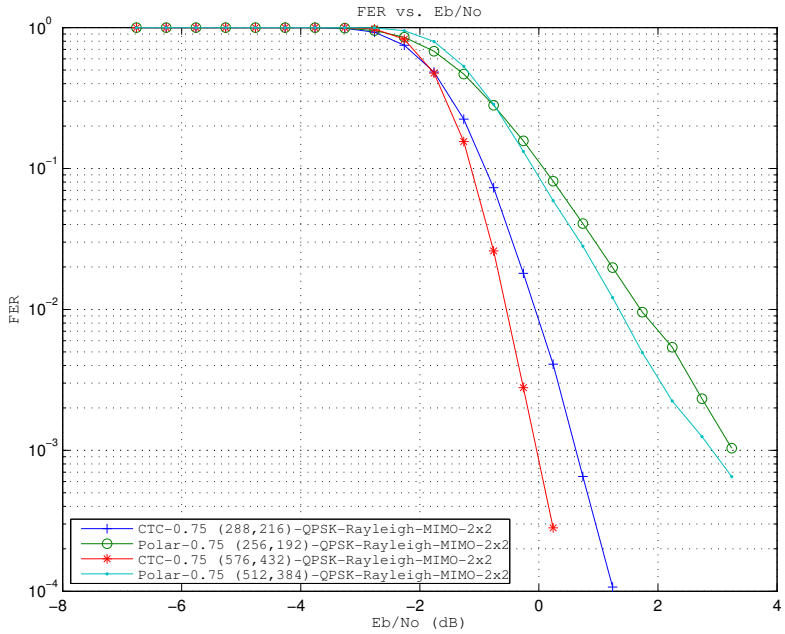

Figure 5.43: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with QPSK Modulation

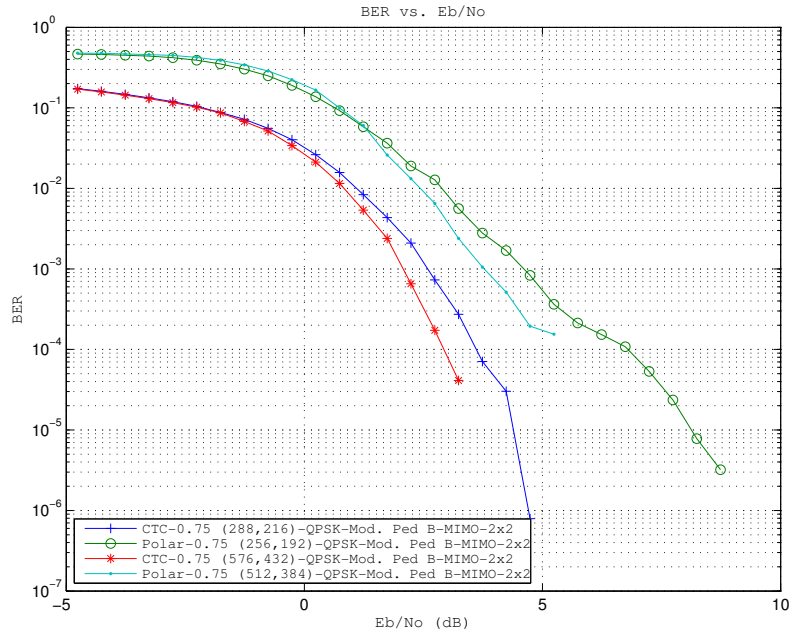

Figure 5.44: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with QPSK Modulation

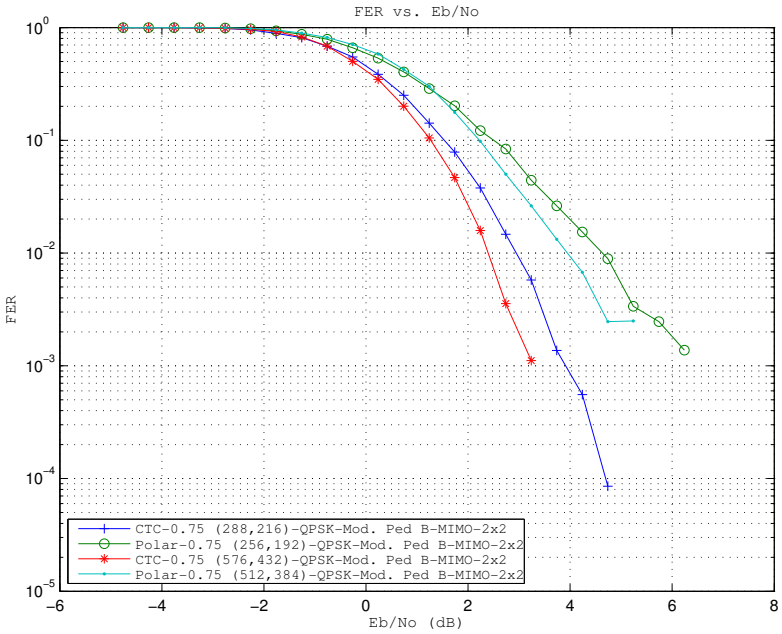

Figure 5.45: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with QPSK Modulation

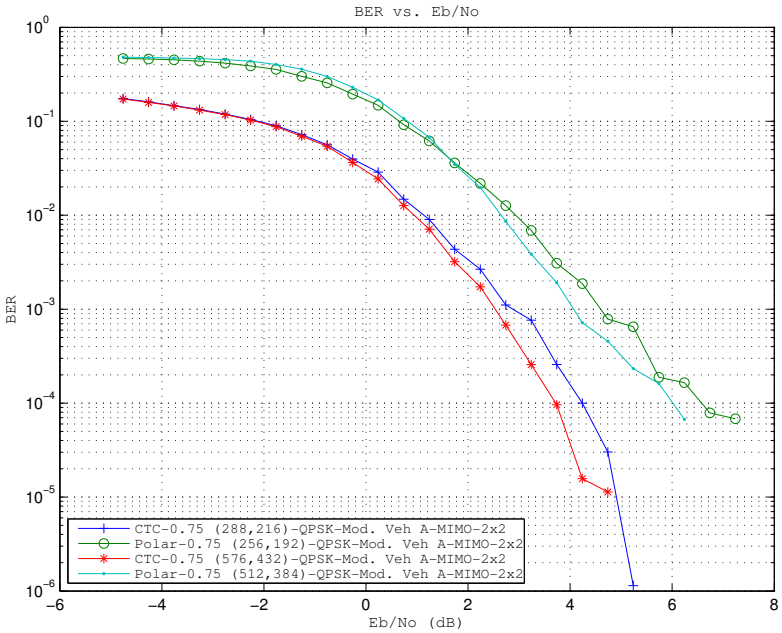

Figure 5.46: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with QPSK Modulation

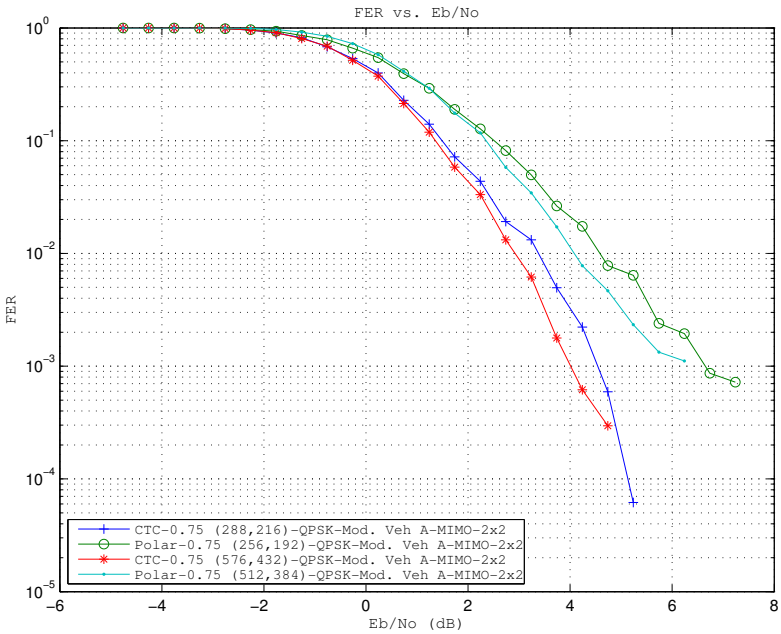

Figure 5.47: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with QPSK Modulation

5.2.2 Results with 16-QAM Modulation in a MIMO 2x2 Setting

5.2.2.1 Comparison of Codes with Coding Rate = 1/2

In this subsection, we present the results of our simulations for a MIMO 2x2 antenna configuration, a modulation choice of 16 QAM and FEC code configurations of rate 1/2. Figures 5.48 and 5.49 give the results under Rayleigh channel, while Figures 5.50- 5.53 show the results of simulations under modified Pedestrian B and Vehicular A channels.

The performance gap in these scenarios remains within a few dBs; mostly 1-2 dB's.

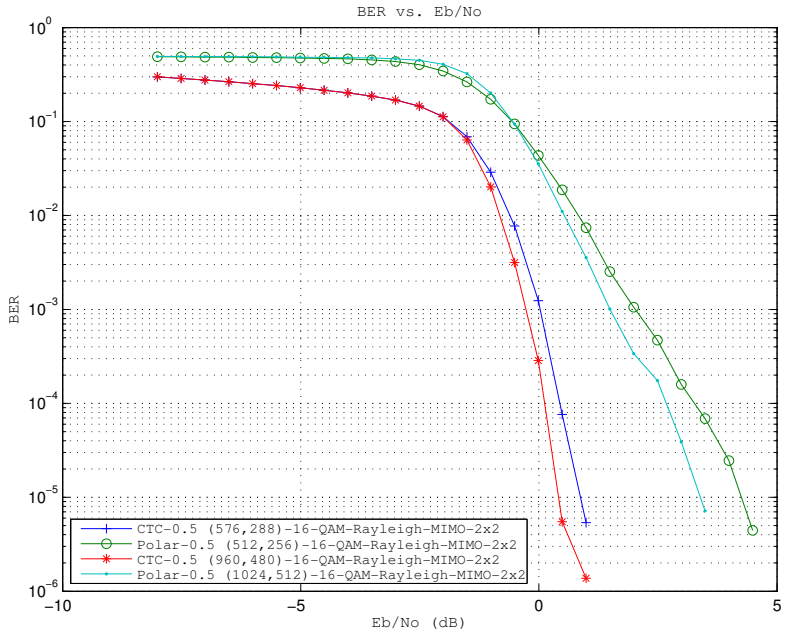

Figure 5.48: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

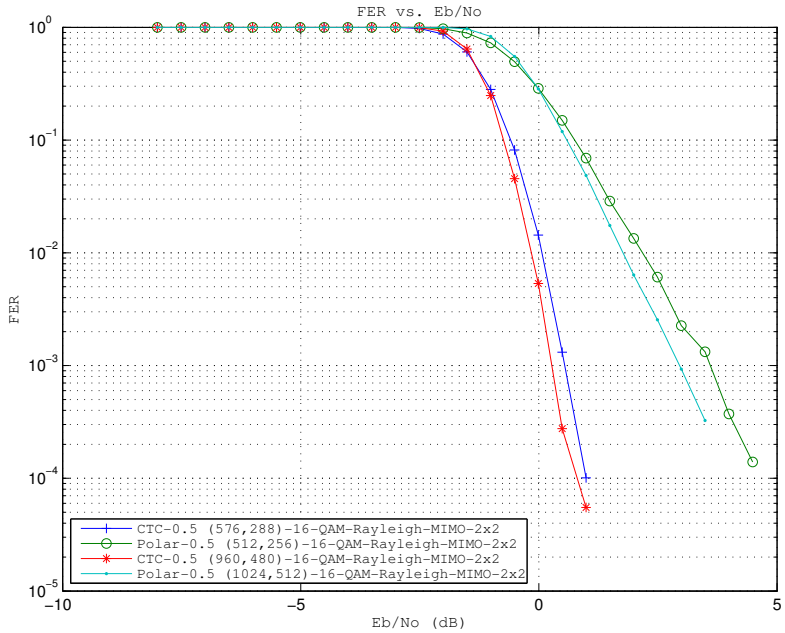

Figure 5.49: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

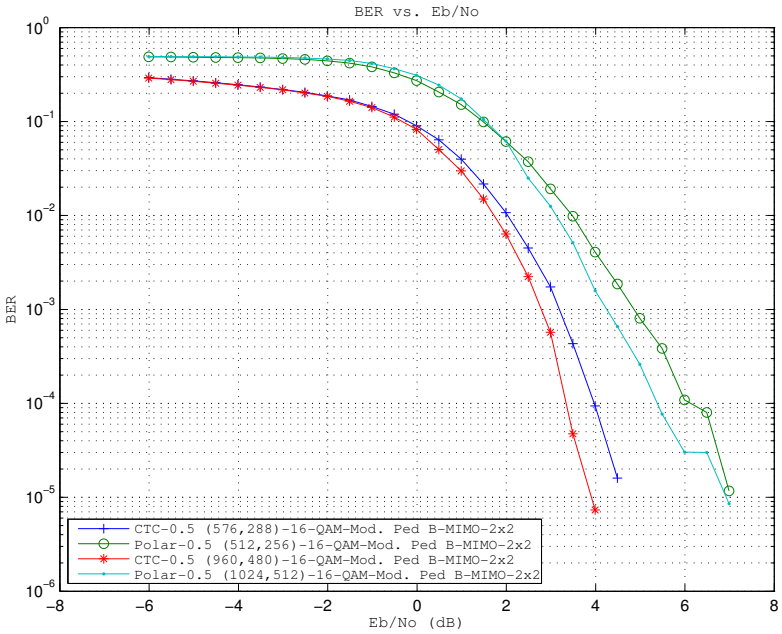

Figure 5.50: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

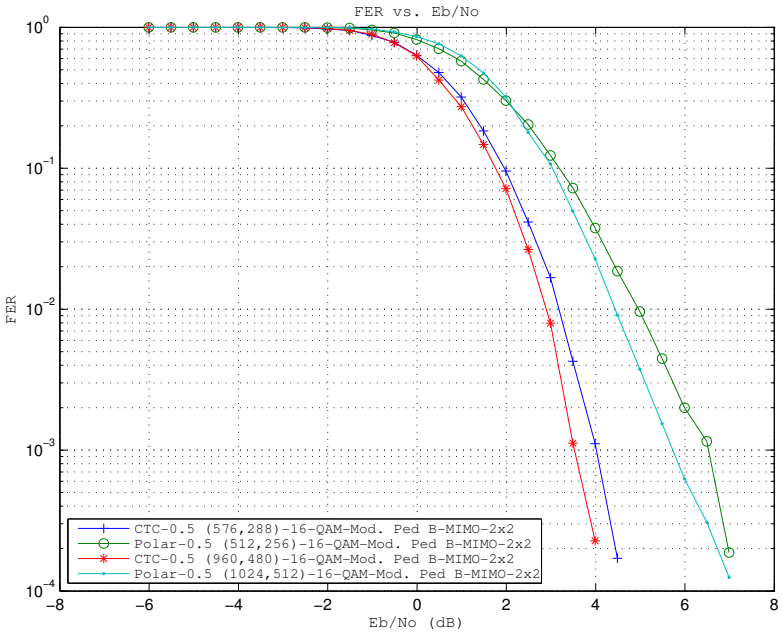

Figure 5.51: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

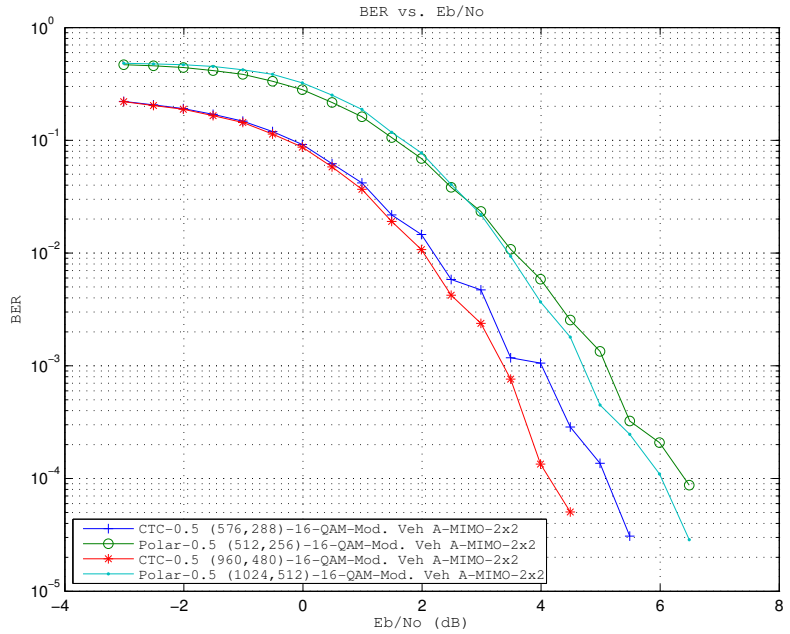

Figure 5.52: BER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

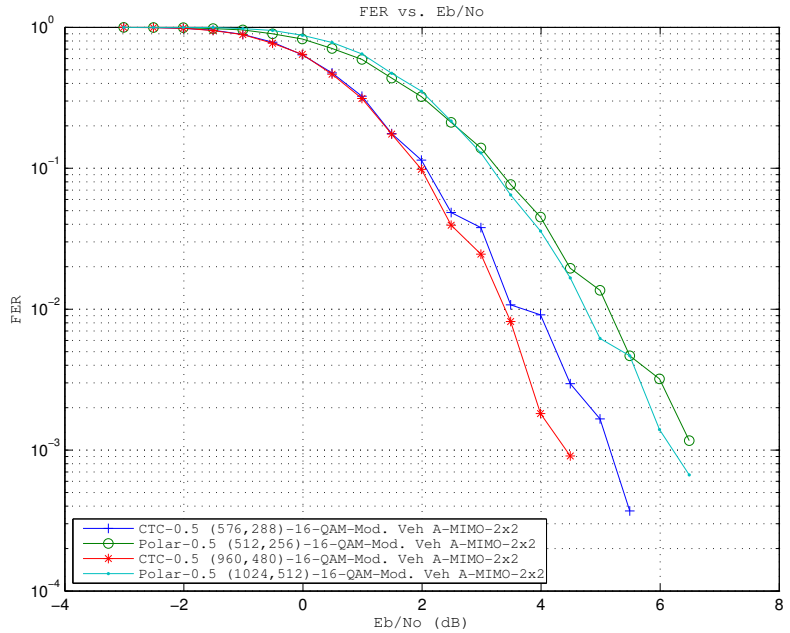

Figure 5.53: FER vs E_b/N_o Performance Curve for Rate 1/2 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

5.2.2.2 Comparison of Codes with Coding Rate = 3/4

In this subsection, we present the results of our simulations for a MIMO 2x2 antenna configuration, a modulation choice of 16 QAM and FEC code configurations of rate 3/4. Figures 5.54 and 5.55 give the results under Rayleigh channel, while Figures 5.56- 5.59 show the results of simulations under modified Pedestrian B and Vehicular A channels.

Again, the performance gap in these scenarios remains within a few dBs; mostly 1-2 dB's; possibly making these scenarios the ones in which the performance of polar codes is nearest to their CTC counterparts.

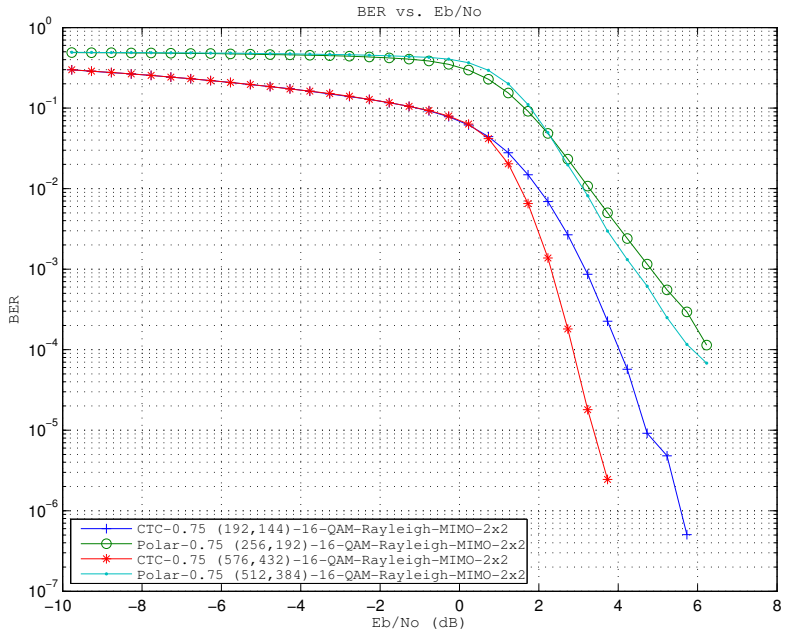

Figure 5.54: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

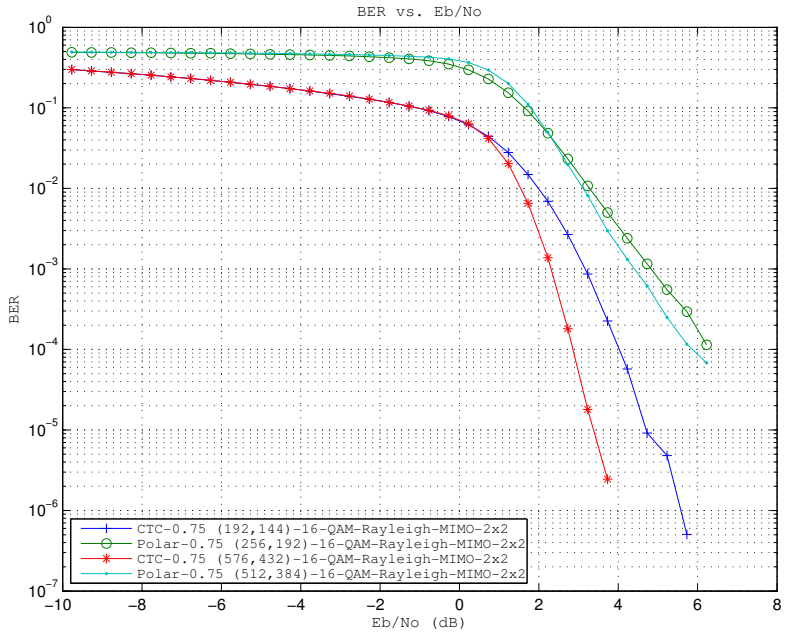

Figure 5.55: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Rayleigh Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

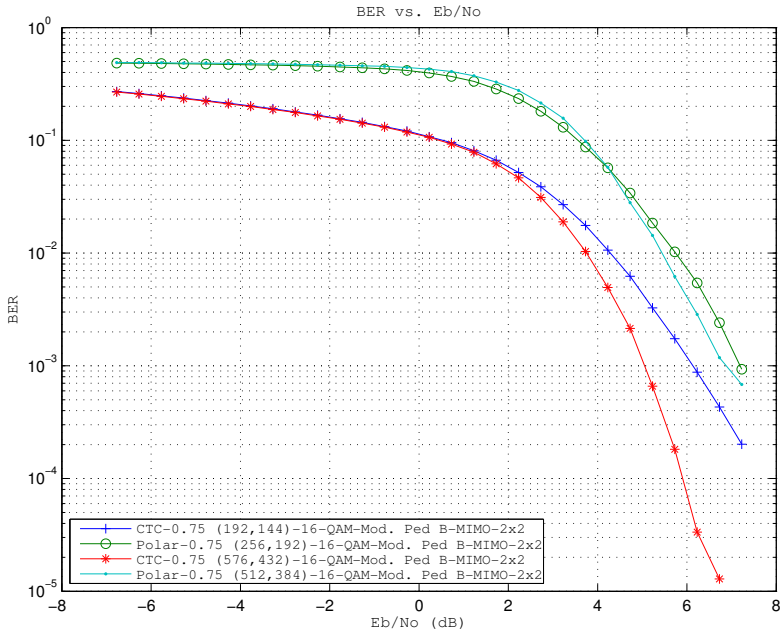

Figure 5.56: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

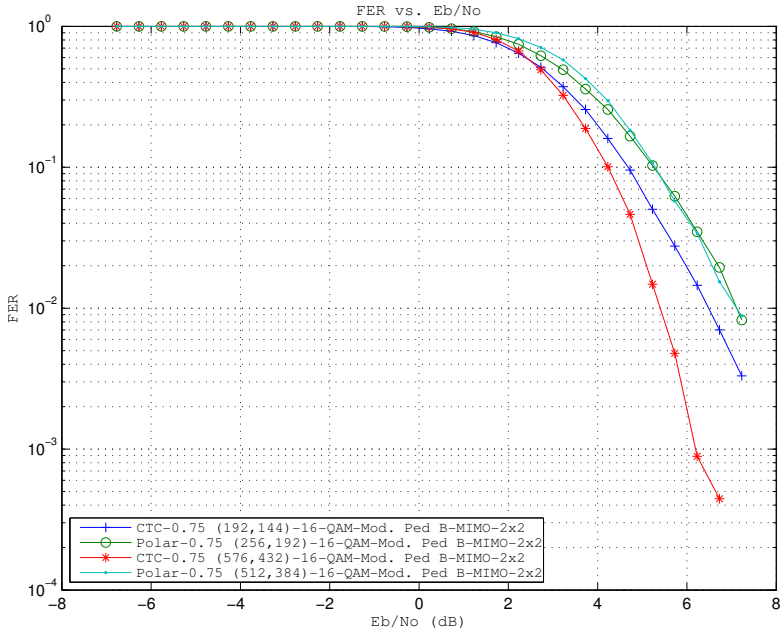

Figure 5.57: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Pedestrian B Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

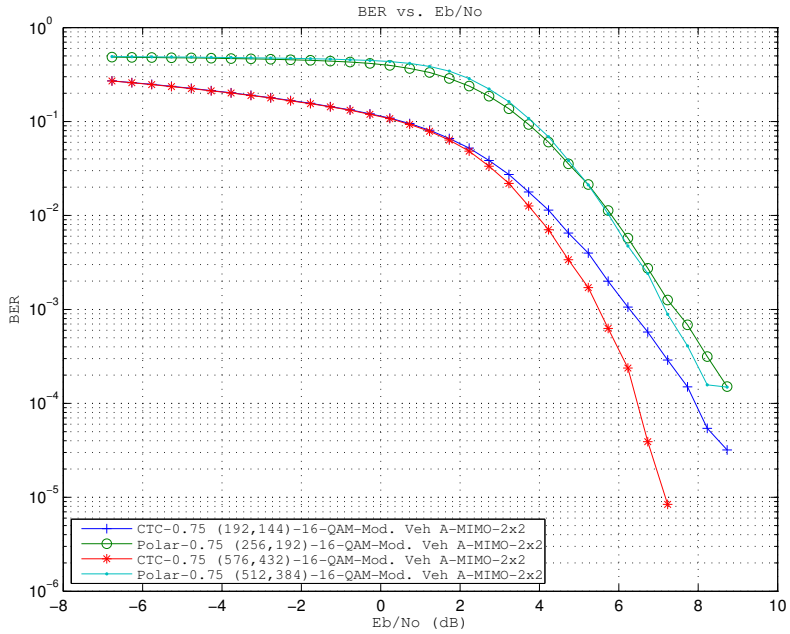

Figure 5.58: BER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

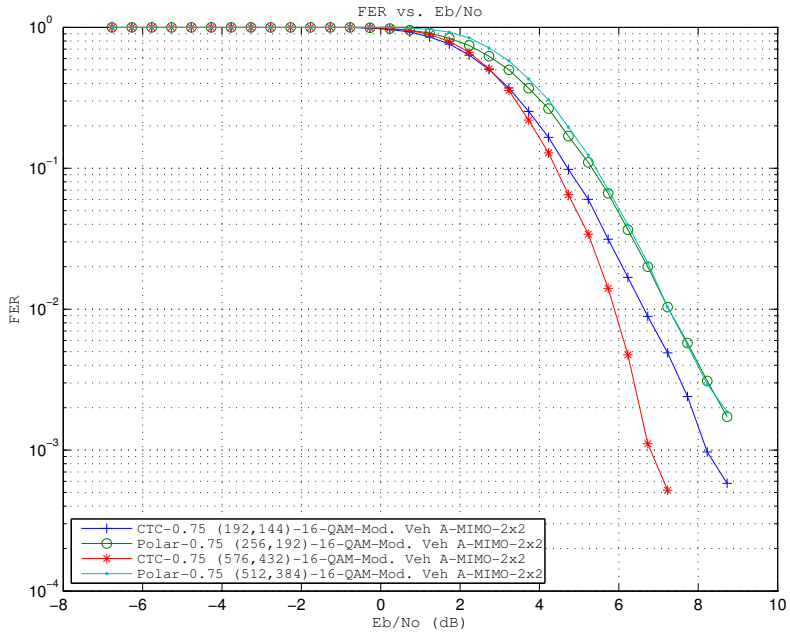

Figure 5.59: FER vs E_b/N_o Performance Curve for Rate 3/4 Polar and CTC Codes at Two Different Code Lengths in Mod. Vehicular A Channel in a MIMO 2x2 Setting, with 16 QAM Modulation

5.3 Analysis of Performance Gap Between the Polar Codes and CTCs

Clearly, at their current implementation, polar codes cannot compete with CTC codes. There are several issues to be considered:

The reason there is a performance gap at low SNR values (the vertical gap between the CTC and polar plots) can be explained as follows:

First of all, CTC is a systematic code while polar is not. The actual data is used intact in the CTC code, while the encoder of the polar codes modify the source values for every value.

For polar codes, since the selection of frozen bits is very important, and the system is very sensitive to wrong decisions on the choice of that set, deciding on the wrong set results in an iteration of wrong knowledge from the most reliable bit to less reliable bits. Therefore, since the estimate for the bit that is assumed the most reliable is not actually yielding a correct result, the error propagates and almost all the remaining values at other positions are decoded wrongly.

The selection of frozen bits for polar codes is, as mentioned before, a channel dependent process. However, the process is only explicitly formulated so far for B-DMC channels. In our simulations, no matter what the channel we are operating under, we have constructed polar codes as if the channel was a B-DMC. This choice obviously made the set of frozen bits unreliable for the channel at hand, although the formation of the set is straightforward.

On the other hand, CTC does not suffer from this error propagation phenomena as heavily as polar codes, due its systematic nature. As a consequence, CTC codes usually have two different operating regions: One in which the slope of the performance curve is low, however still downward (unlike polar codes for

which the error rate remains steady for an interval), and another waterfall region in which the slope increases to large negative values (similar to the behavior of polar codes – although the performance curves are usually more steep for CTC codes.)

The selection of the frozen bit positions might be improved as mentioned in [2] through the use of Monte-Carlo simulations. Our scope in this thesis has essentially been polar codes for which the construction depends on the channel being assumed B-DMC, however we have made a sample simulation in which the system is trained through Monte-Carlo simulations to be optimized against an AWGN channel with an SNR value of 8 dB. We provide sample results for the simulations with the trained polar codes in Section 5.3.1, where the expected improvement is visible.

5.3.1 Improvement Through Selection of Frozen Positions

Although our study in this thesis has involved polar codes as described in [2] with an assumption of BEC for choosing the frozen bit locations, it turns out that this approach makes the performance of polar codes significantly worse. As a sample case showing what achievements can be done if a more appropriate channel model is used, we present some sample results in which the performance of polar codes is improved. We perform Monte-Carlo simulation as explained in [2] in order to determine a new set of frozen positions.

If the frozen position set is chosen according to the channel, polar code shows improvements as expected (Figure 5.60). For example, under AWGN channel, if the frozen position set is chosen assuming an AWGN channel with 8 dB SNR for 16-QAM modulation and rate 1/2 with a codelength of 256, we observe that the

gap between CTC and the original polar code which was optimized for the BEC channel is halved.

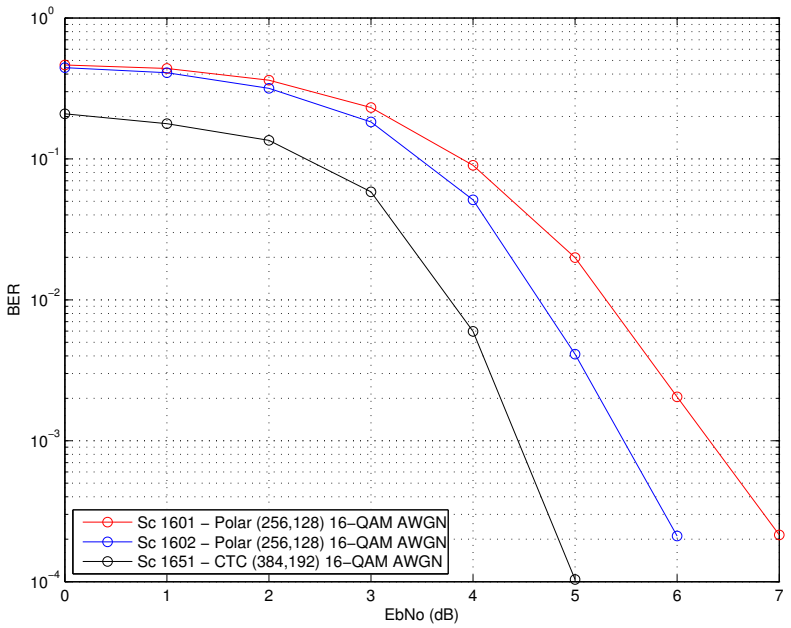

Figure 5.60: BER vs E_b/N_o: Improvement of Polar Code under AWGN Channel Through More Appropriate Frozen Bit Position Selection (Sc 1601 Refers to the Original Polar Code, Sc 1602 Refers to the Improved Polar Code (with Frozen Positions Changed to that of an AWGN Channel with 8 dB SNR), Sc 1605 refers to the reference CTC Code)

Similarly, under Rayleigh channel, there is an improvement even if the channel is still assumed to be an AWGN channel with an SNR of 8 dB. (Figure 5.62)

These results show that the performance of polar codes could be improved further, what we have presented in this subsection only scratches the surface, and presents our early attempts at improving the codes according to channel conditions. As mentioned before, these are early attempts at making polar codes more suitable for the channel at hand, and as such should only be considered as a sample demonstration that improvement is possible. This does not constitute a direct part in the general flow of the thesis, but is rather given as an insight for our future work.

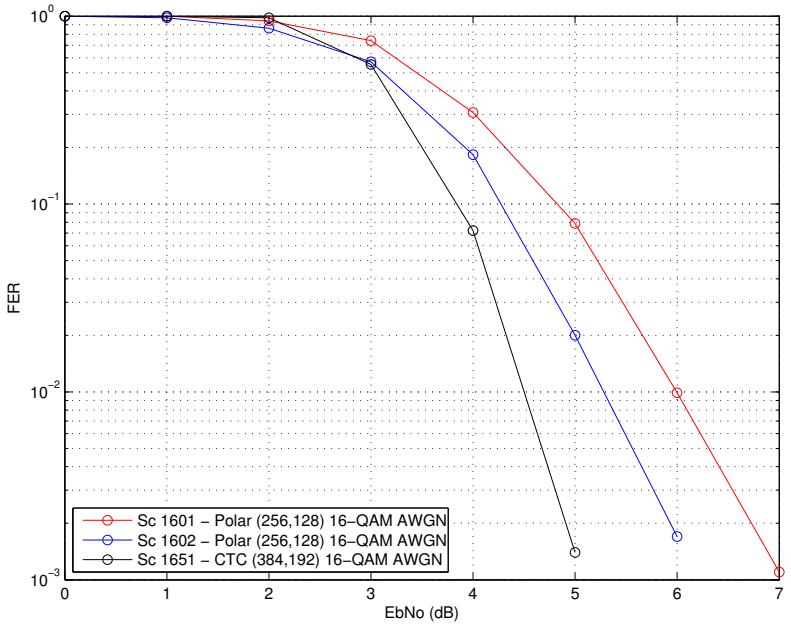

Figure 5.61: FER vs E_b/N_o: Improvement of Polar Code under AWGN Channel Through More Appropriate Frozen Bit Position Selection (Sc 1601 Refers to the Original Polar Code, Sc 1602 Refers to the Improved Polar Cod (with Frozen Positions Changed to that of an AWGN Channel with 8 dB SNR), Sc 1605 refers to the reference CTC Code)

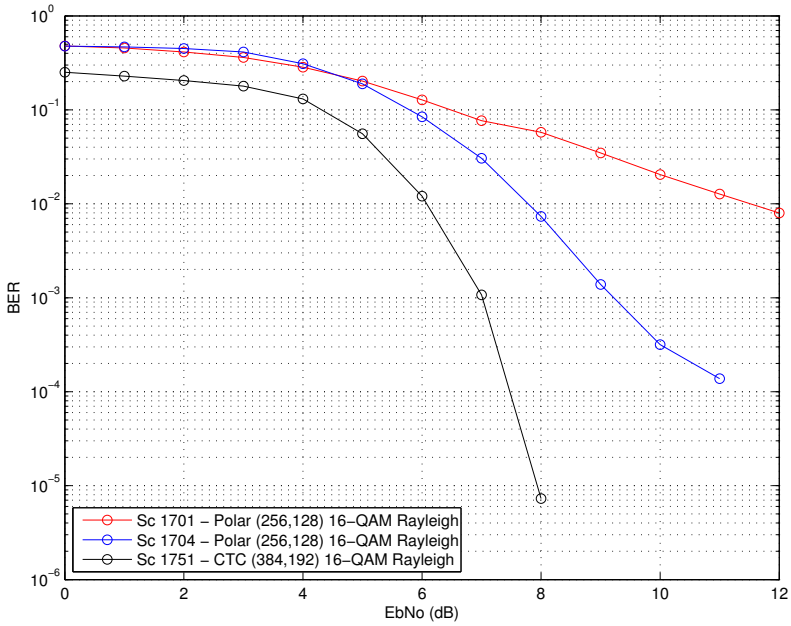

Figure 5.62: BER vs E_b/N_o: Improvement of Polar Code under Rayleigh Channel Through More Appropriate Frozen Bit Position Selection (Sc 1701 Refers to the Original Polar Code, Sc 1704 Refers to the Improved Polar Code (with Frozen Positions Changed to that of an AWGN Channel with 8 dB SNR), Sc 1605 refers to the reference CTC Code)

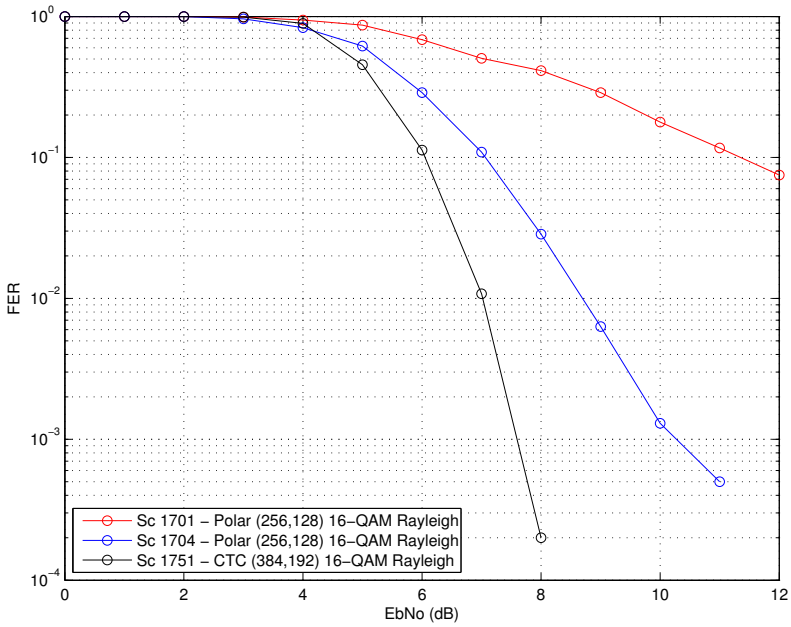

Figure 5.63: FER vs E_b/N_o: Improvement of Polar Code under Rayleigh Channel More Appropriate Frozen Bit Position Selection (Sc 1701 Refers to the Original Polar Code, Sc 1704 Refers to the Improved Polar Code (with Frozen Positions Changed to that of an AWGN Channel with 8 dB SNR), Sc 1605 refers to the reference CTC Code)

5.4 Comparison of Similar Scenarios under Different MIMO Schemes and Channel Conditions

In this section, we perform a comparative analysis of similar code configurations under various conditions in order to see how changing the MIMO scheme or channel condition affects the performance.

In the first subsection, we analyze the effect of channel while keeping the code configurations and MIMO scheme fixed, and compare the result of separate scenarios. (Figures 5.64-5.69)

Conversely, in the second subsection, we analyze the effect of MIMO scheme while keeping the code configurations and the channel fixed, and plot all the results on one figure to see the effect. (Figures 5.71- 5.74)

5.4.1 Effect of Channel in Similar Code Configurations and MIMO Settings

In this subsection, we use the results in previous subsection to see how changing the channel changes performance. Effectively, we are replotting the results of selected previous individual simulations on top of each other in order to be able to compare them within the same context.

In order to investigate the effect of channel, given the same modulation choice and antenna setting, we change the channel for each configuration in the first part.

For the SISO configuration (Figures 5.64- 5.67), as expected, AWGN channel yields the best performance since there is no fading. The flat fading causes by

the Rayleigh channel results in a further 4-5 dB drop in performance, while the modified Pedestrian B and Vehicular A channels perform similarly (due to the assumption that the channels are static over two OFDM symbols), and results in a further 5 dB decrease in performance. These observations are valid for scenarios with both QPSK and 16-QAM modulations.

For MIMO 2x2 configuration (Figures 5.71-5.74), we have not simulated the AWGN channel cases; therefore only comparisons between the Rayleigh fading and modified Pedestrian B channel (or equivalently the modified Vehicular A channel) might be done. Here, again, the performance gap is around 5-6 dB's.

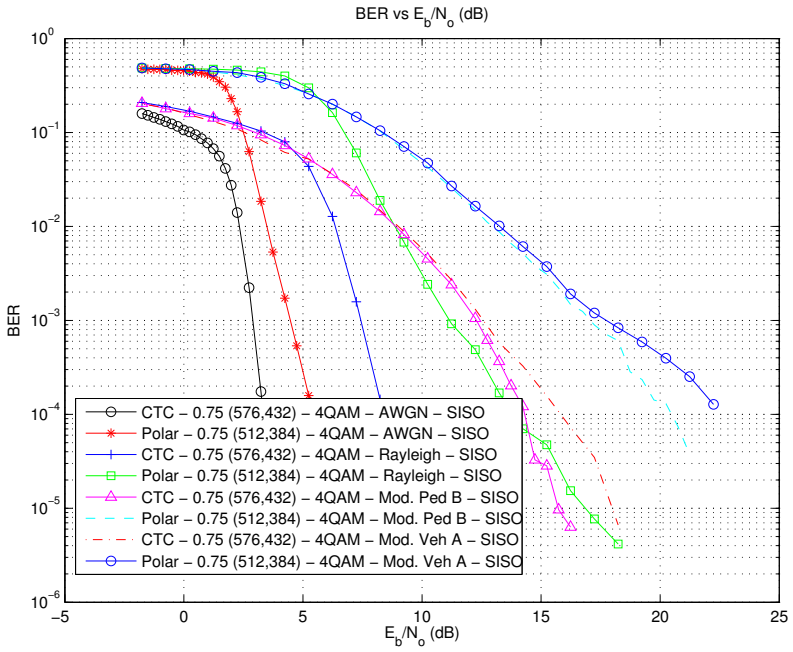

Figure 5.64: Effect of Channel in Similar Code Configurations in a SISO Setting with QPSK Modulation: BER vs E_b/N_o Plot

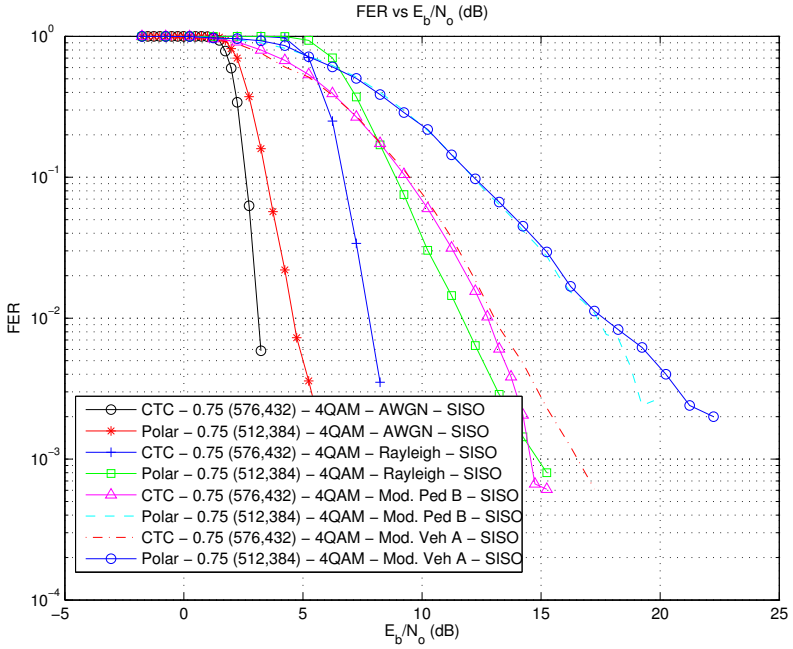

Figure 5.65: Effect of Channel in Similar Code Configurations in a SISO Setting with QPSK Modulation: FER vs E_b/N_o Plot

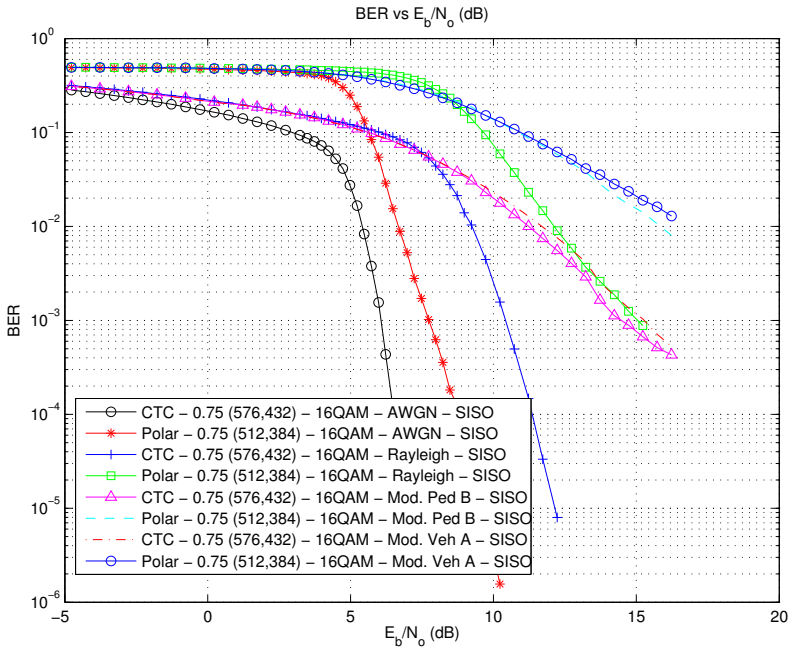

Figure 5.66: Effect of Channel in Similar Code Configurations in a SISO Setting with 16 QAM Modulation: BER vs E_b/N_o Plot

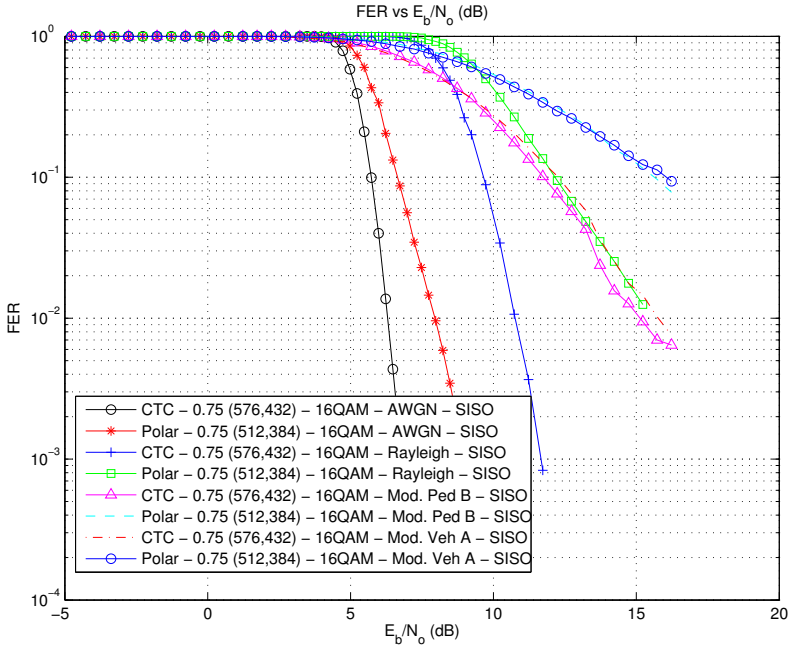

Figure 5.67: Effect of Channel in Similar Code Configurations in a SISO Setting with 16 QAM Modulation: FER vs E_b/N_o Plot

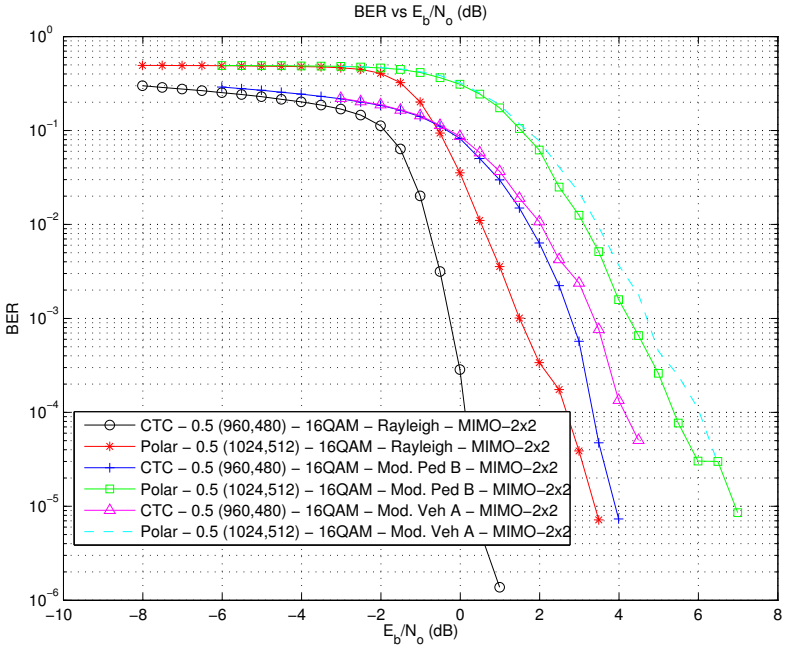

Figure 5.68: Effect of Channel in Similar Code Configurations in a MIMO 2x2 Setting with 16 QAM Modulation: BER vs E_b/N_o Plot

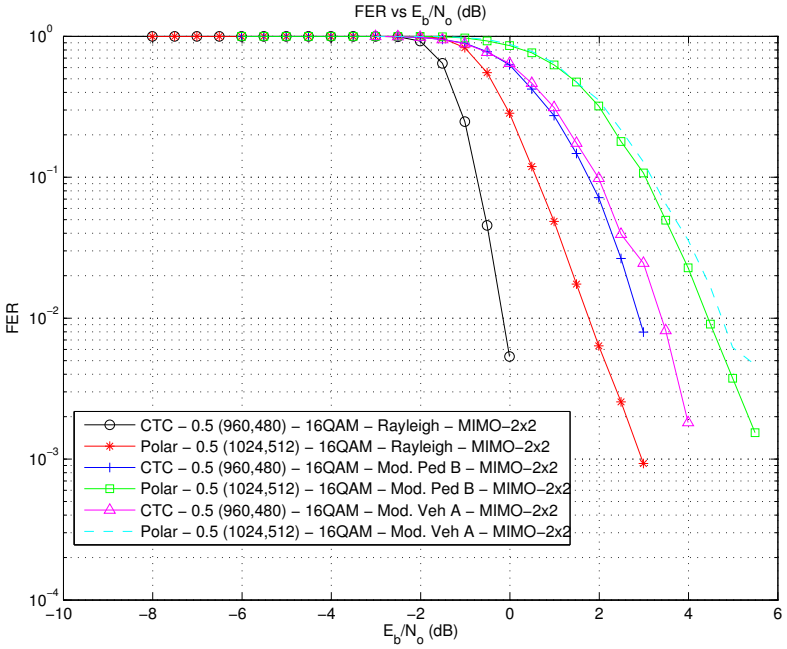

Figure 5.69: Effect of Channel in Similar Code Configurations in a MIMO 2x2 Setting with 16 QAM Modulation: FER vs E_b/N_o Plot

5.4.2 Effect of MIMO Scheme in Similar Code Configurations and Channels

Similar to previous subsection, our goal is this subsection is to visualize the effect of different antenna configurations on performance for a fixed channel (Rayleigh or Modified Vehicular A). Figures 5.71 and 5.72 show the performance curves for the Rayleigh channel while Figures 5.73 and 5.74 are for the modified Vehicular A channel. The results for modified Pedestrian B channel are not included since due to our static channel assumption for two subsequent OFDM symbols, results for the modified Pedestrian B channel are fairly equivalent to results for the modified Vehicular A channel.

First, however, in order to form a reference point, we first present the results of uncoded modulation with different antenna configurations in Figure 5.70.

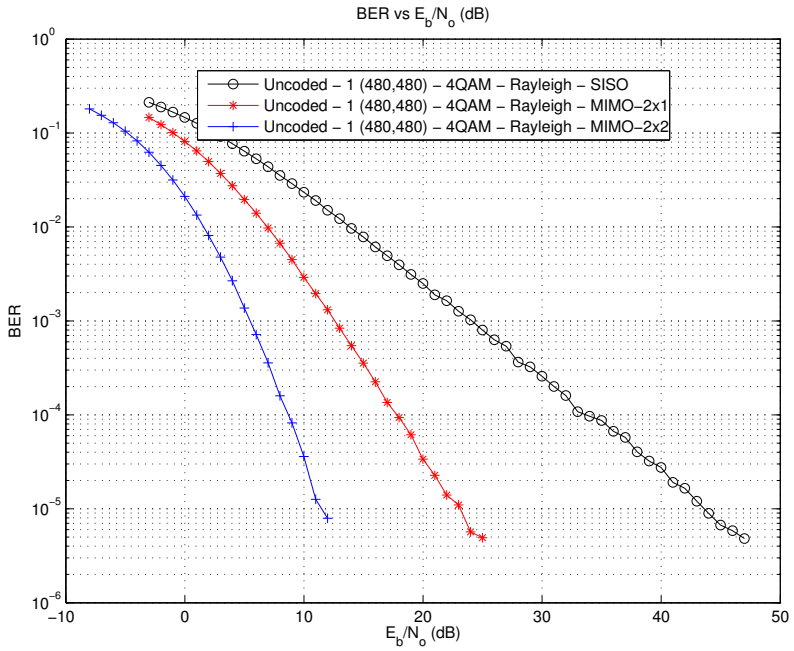

Figure 5.70: Performance of Uncoded Modulation Under Rayleigh Channel with QPSK Modulation for SISO, MIMO 2x1 and MIMO 2x2 Antenna Schemes

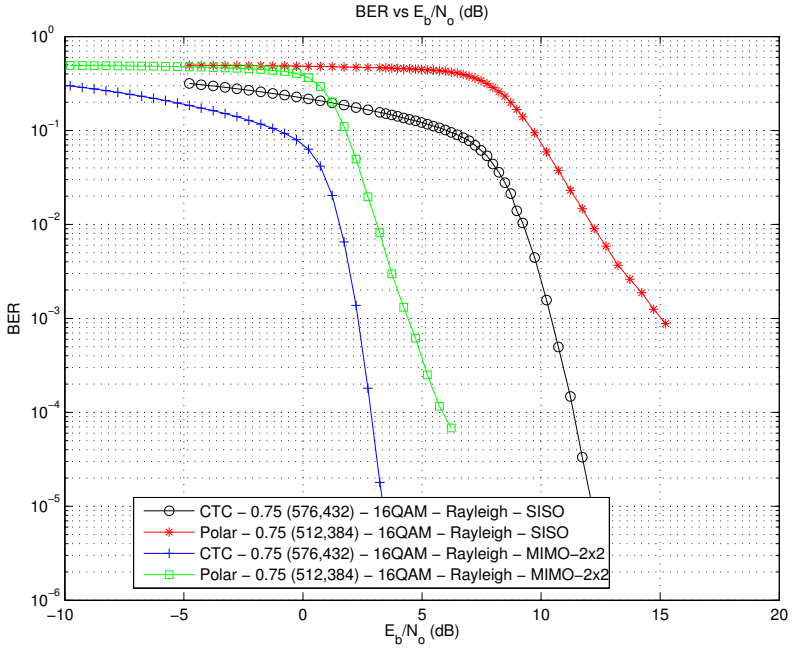

Figure 5.71: Effect of MIMO 2x2 Scheme for Similar Code Configurations in Rayleigh Channel: BER vs E_b/N_o Plot

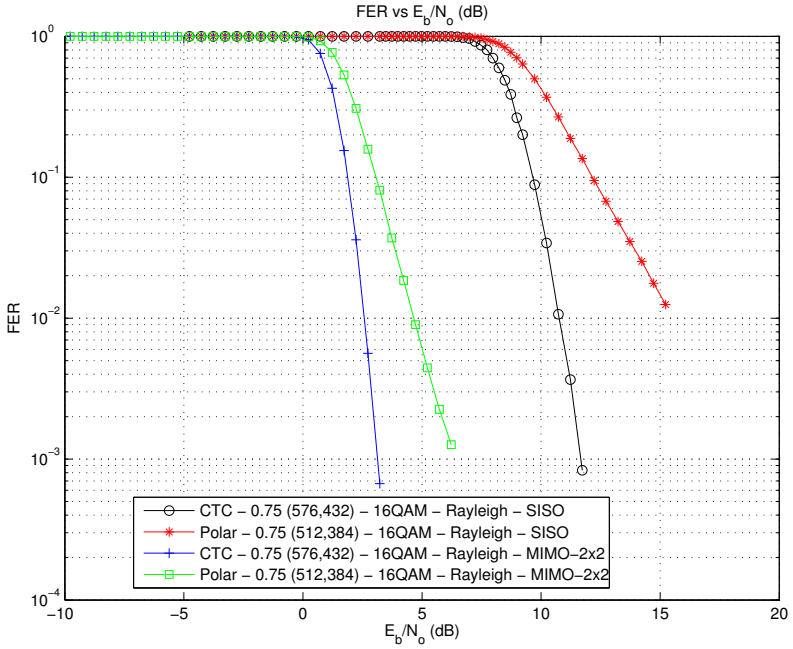

Figure 5.72: Effect of MIMO 2x2 Scheme for Similar Code Configurations in Rayleigh Channel: FER vs E_b/N_o Plot

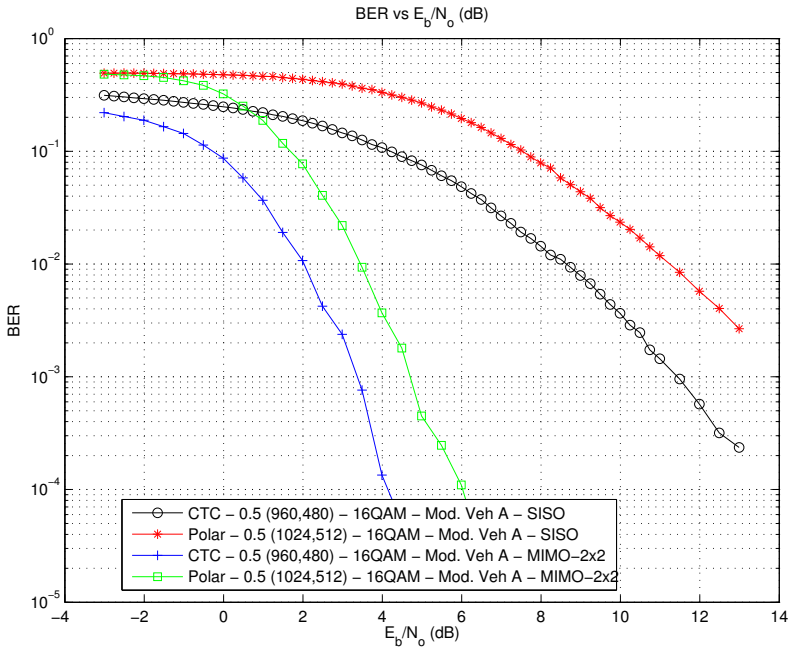

Figure 5.73: Effect of MIMO 2x2 Scheme for Similar Code Configurations in Modified Vehicular A Channel: BER vs E_b/N_o Plot

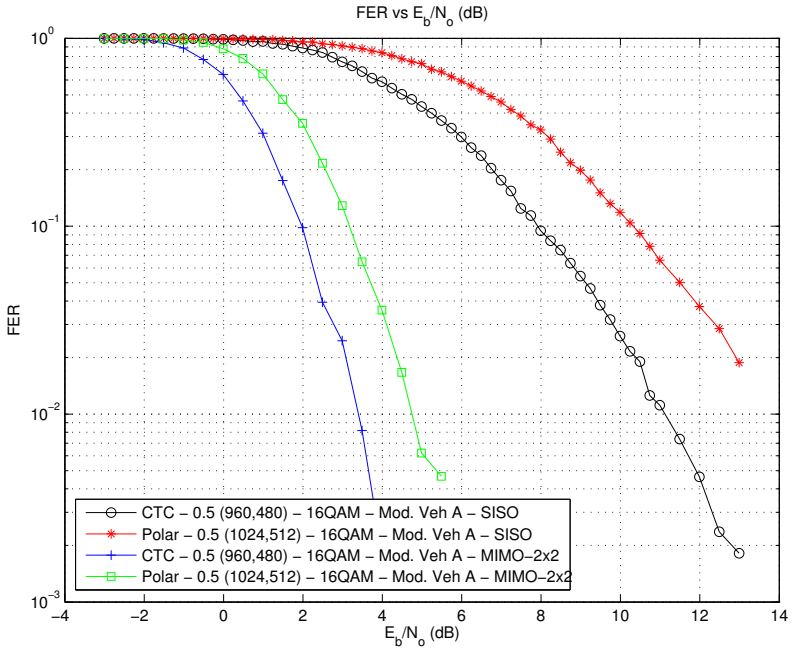

Figure 5.74: Effect of MIMO 2x2 Scheme for Similar Code Configurations in Modified Vehicular A Channel: FER vs E_b/N_o Plot

5.5 Effect of PUSC on Performance

In this section, we investigate the effect of PUSC on performance. Obviously, PUSC does not bring any performance improvements for the AWGN channel and Rayleigh channel. Therefore, we have investigated the effect of PUSC in Modified Pedestrian B channel, since our assumption that the channel is static over two OFDM symbols causes the performance of the system under Modified Pedestrian B channel to be very similar to that under Modified Vehicular A channel.

In all cases, we have fixed the code rate to 1/2, and investigated the effect of PUSC under various modulations and antenna configurations.

Clearly, PUSC results in a performance increase for error rates below 10^{-1}. This performance increase tends to be on the order of 1-2 dBs in almost all cases.

Figure 5.75 shows the effect of PUSC under QPSK modulation in a SISO setting. Similarly, in Figure 5.77, we see the effect of PUSC under 16 QAM modulation for the same antenna configuration.

Figure 5.79 shows the effect of PUSC under QPSK modulation in a MIMO 2x2 setting. Similarly, in Figure 5.81, we see the effect of PUSC under 16 QAM modulation for the same antenna configuration.

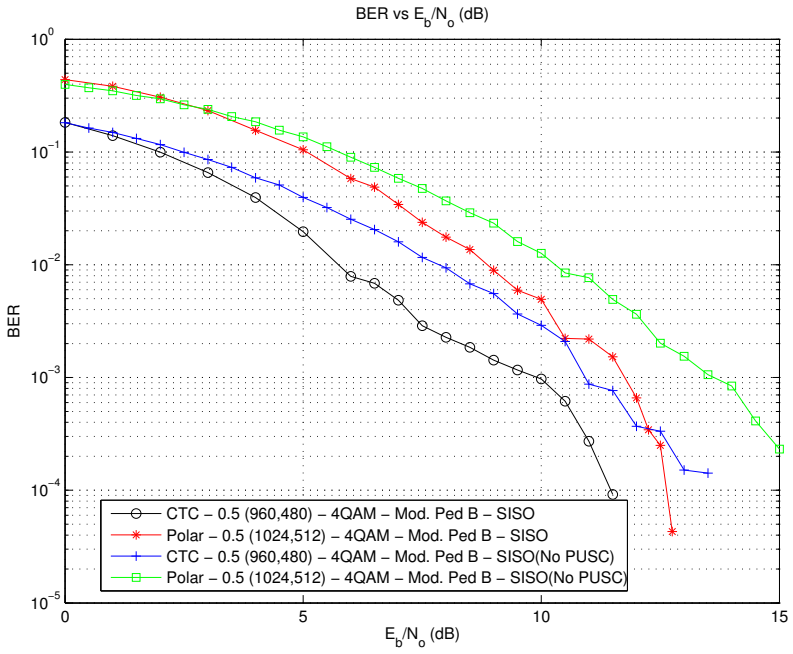

Figure 5.75: BER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for SISO Scheme with QPSK Modulation

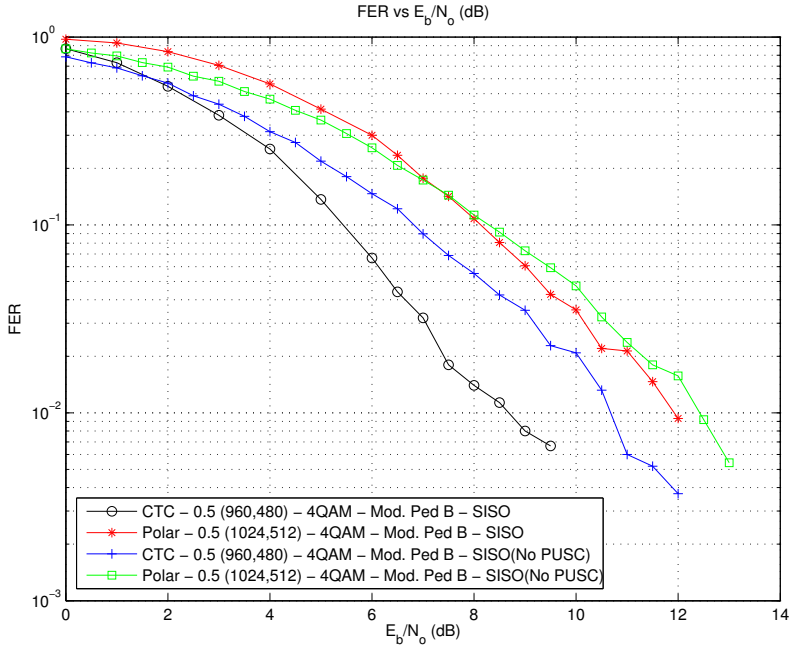

Figure 5.76: FER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for SISO Scheme with QPSK Modulation

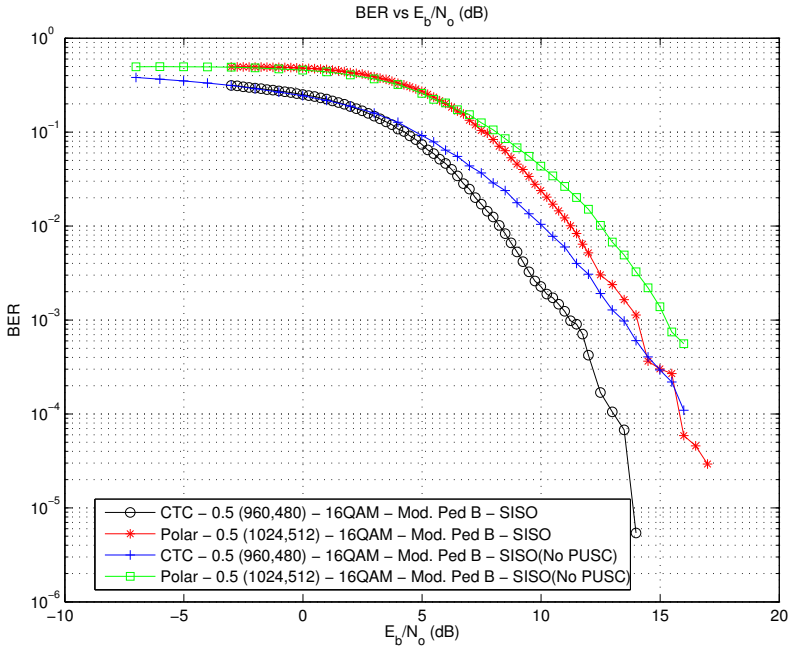

Figure 5.77: BER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for SISO Scheme with 16 QAM Modulation

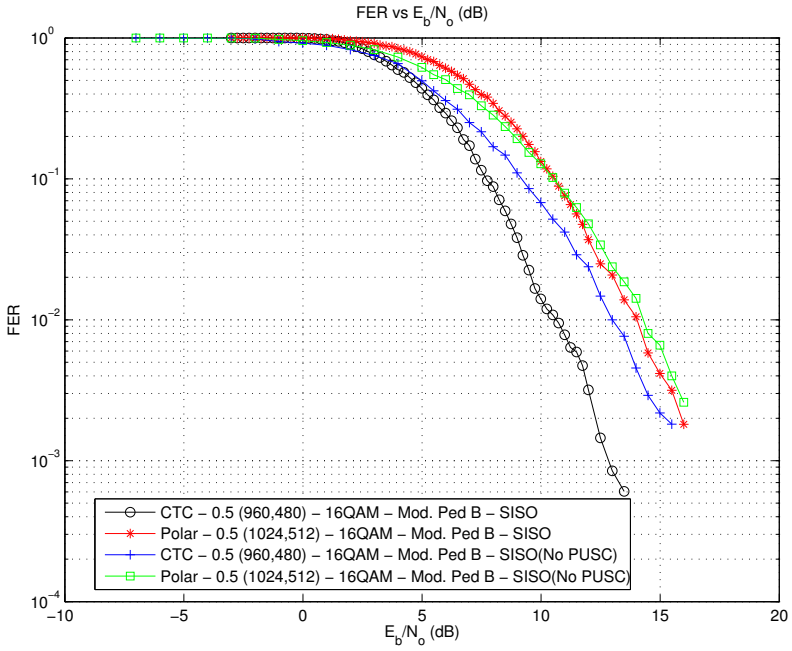

Figure 5.78: FER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for SISO Scheme with 16 QAM Modulation

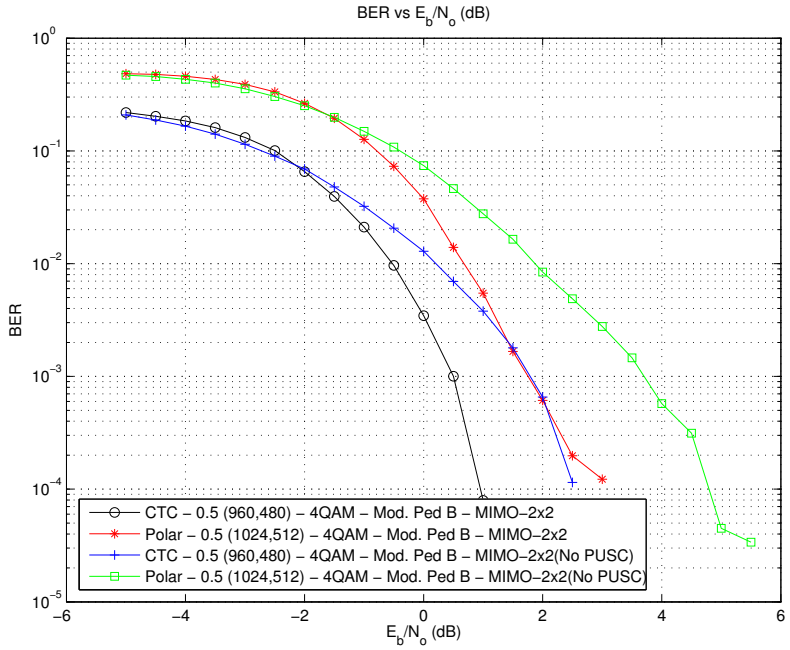

Figure 5.79: BER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for MIMO 2x2 Scheme with QPSK Modulation

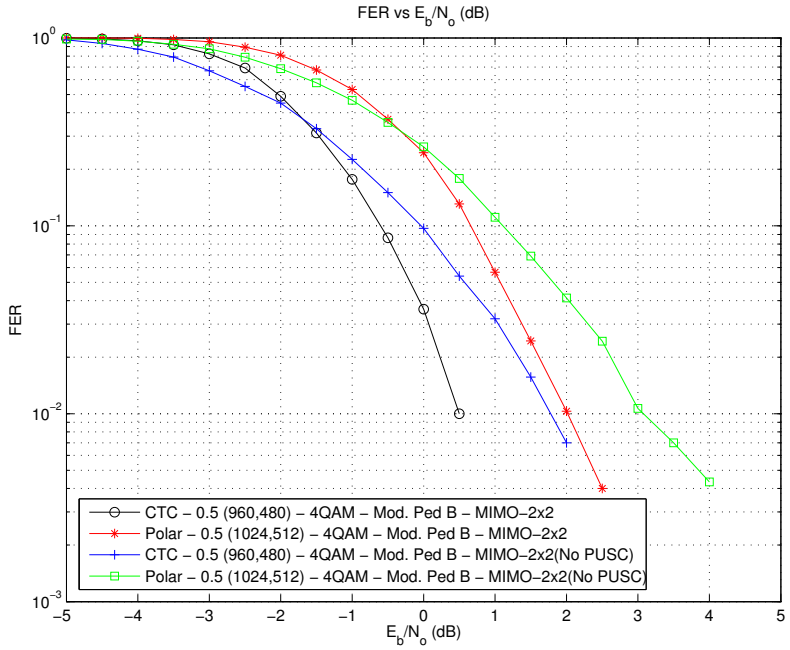

Figure 5.80: FER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for MIMO 2x2 Scheme with QPSK Modulation

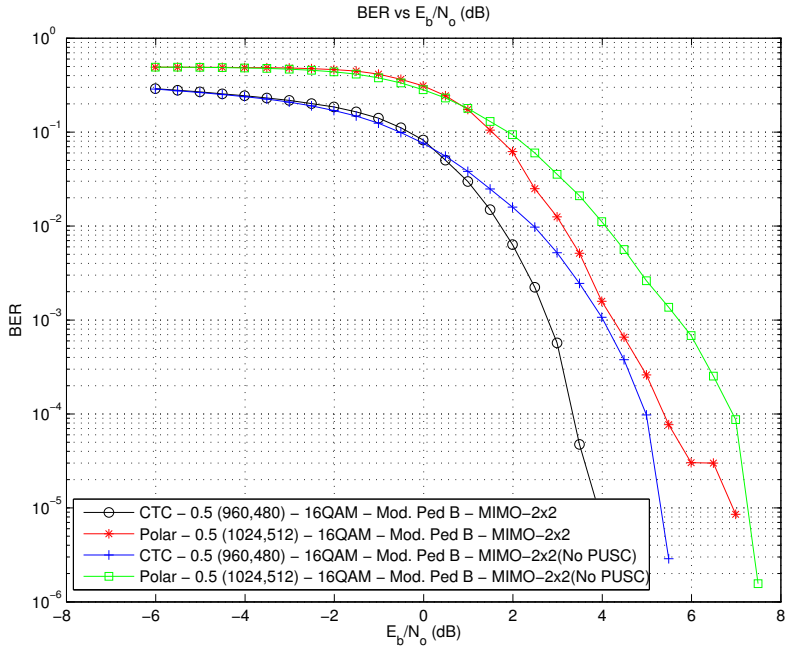

Figure 5.81: BER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for MIMO 2x2 Scheme with 16 QAM Modulation

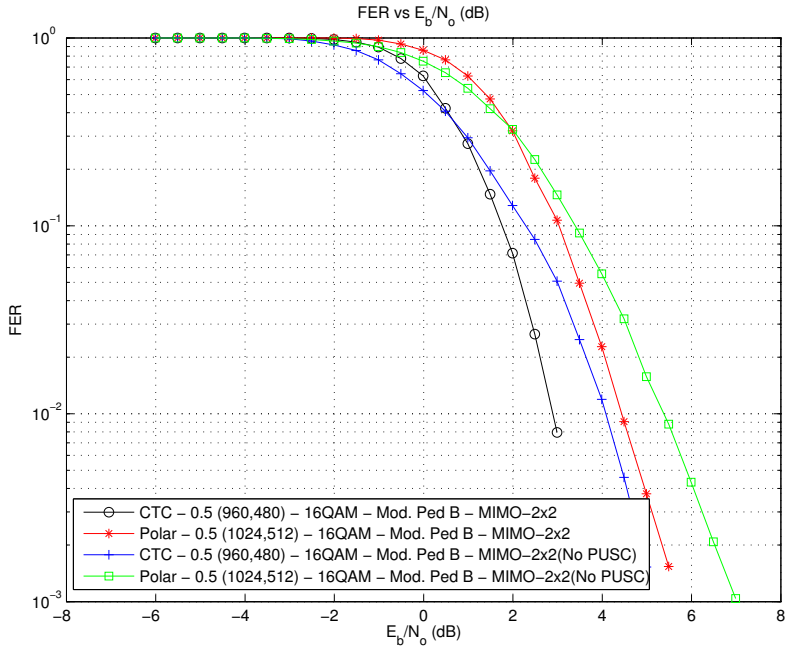

Figure 5.82: FER vs EbNo Performance of Two Code Configurations with PUSC Enabled and Disabled; under Mod. Ped B Channel for MIMO 2x2 Scheme with 16 QAM Modulation

Chapter 6

CONCLUSION

It has been theoretically shown that polar codes achieve symmetric capacity for B-DMC's, however it was a task to see how these codes perform under AWGN and baseline channels in a WiMAX simulator.

It has been shown that for small blocklengths as specified under IEEE 802.16 standards, the performance of polar codes is worse compared with CTC codes with same configurations. The low-complexity encoding and decoding of polar codes do not help in these cases, since the polarization effect does not show itself.

Our tests have involved testing under SISO and and MIMO 2x2 conditions with channel models ranging from simple AWGN channel to modified pedestrian B and vehicular A channels of the ITU. Under no setting we have found a polar code that matches the performance of CTCs. Our instantaneous and perfect knowledge of channel information made it difficult to observe difference between the pedestrian and vehicular channels. MIMO 2x2, as expected, outperforms the SISO antenna scheme by more than a few dBs.

We note that we have assumed that the channels we have combined to generate polar codes are B-DMCs when in reality, they are channels based either

on AWGN or baseline channels. Generation of reliability parameters for these channel conditions should aid code performance.

One other issue to mention about polar codes is that, since the encoding assumes that bits go through the channel immediately after encoding, polar codes, used as a combined modulation and encoding block might improve performance. In fact, polar coding theory has recently been generalized to q-ary channels [18] and work is in progress to extend it to real numbers, under which this vision might be fulfilled. It might even be suggested that since polar coding operates on channels, alternatives to OFDMA might be rethought with polar coding in mind.

Tasks to improve the mentioned issues will be part of our ongoing efforts, and we believe that such efforts will give us a better understanding in solving adaptive modulation and coding methods. The awareness of polar codes to channel conditions, and its full control on the whole chain if it replaces the modulation steps, and its support of granular rate changes might lead to a more flexible adaptation scheme.

APPENDIX A

PERFORMANCE IMPROVEMENTS USING MEX FILES

A.1 Summary

On average, the performance gain is about 18 fold. There does not seem to be much difference between the compilers, though LCC performed significantly worse w.r.t. others (half the performance.)

Changing N or the number of trials does not have a perceivable effect on the performance. It should be noted, however that, for small number of trials, since the compiled C program is loaded into memory for the first time, the performance is significantly lower wrt those with more samples. This issue, however is not significant, since the average performance is satisfactory.

A.2 Results

Below, we provide tables and figures showing averages, holding a variable (compiler, N or the number of trials) constant.

Table A.1: Average Gains for Different Compilers

Compiler	Average Gain
MS	17.7779
LCC	10.7501
Mingw	18.3047
Cygwin-mingw	18.0044

Table A.2: Average Gain as Polar Code Length N Increases

N	Average Gain
256	14.0251
512	16.4667
1024	16.5585
2048	16.749
4096	16.6891
8192	16.3234
16384	16.6529

Table A.3: Average Gain as Trial Number Increases

Trial Number	Average Gain
10	14.012
2008	16.6897
4006	16.616
6004	16.63
8002	16.672
10000	16.6358

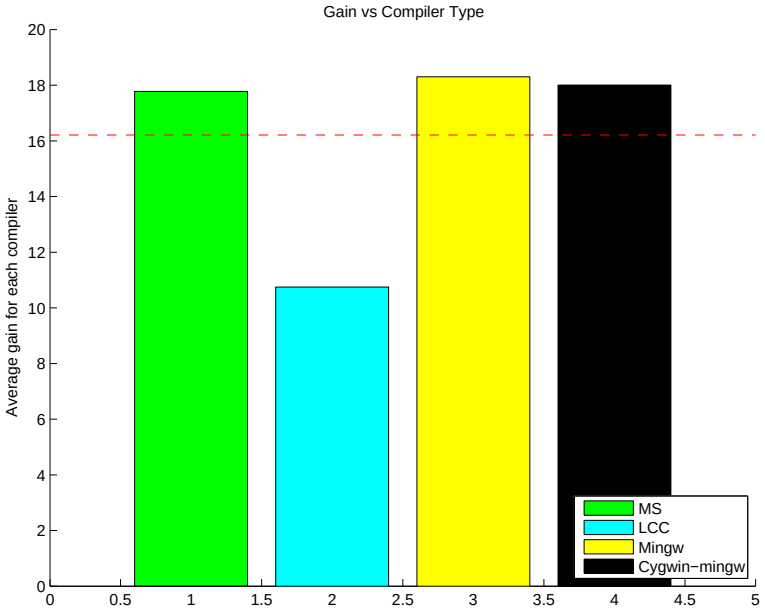

Figure A.1: Gain Averages for Different Compilers

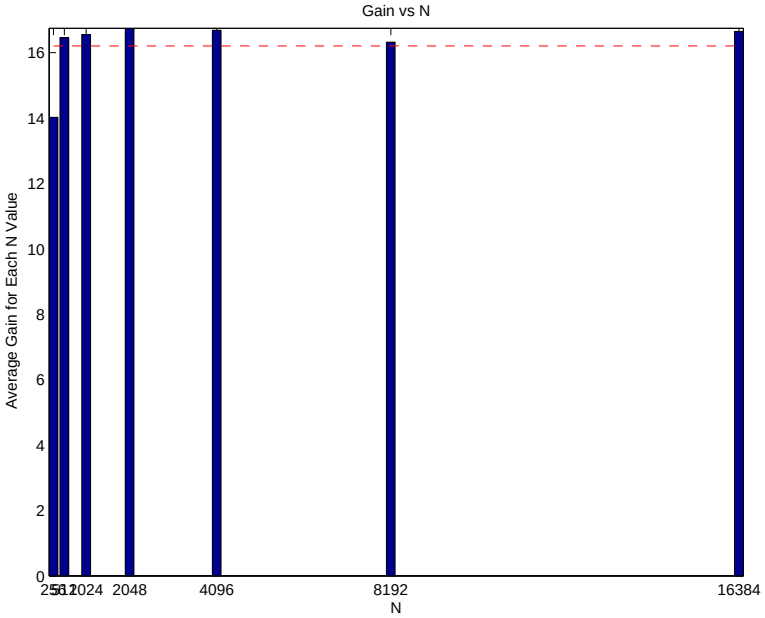

Figure A.2: Gain Averages for Different N Values

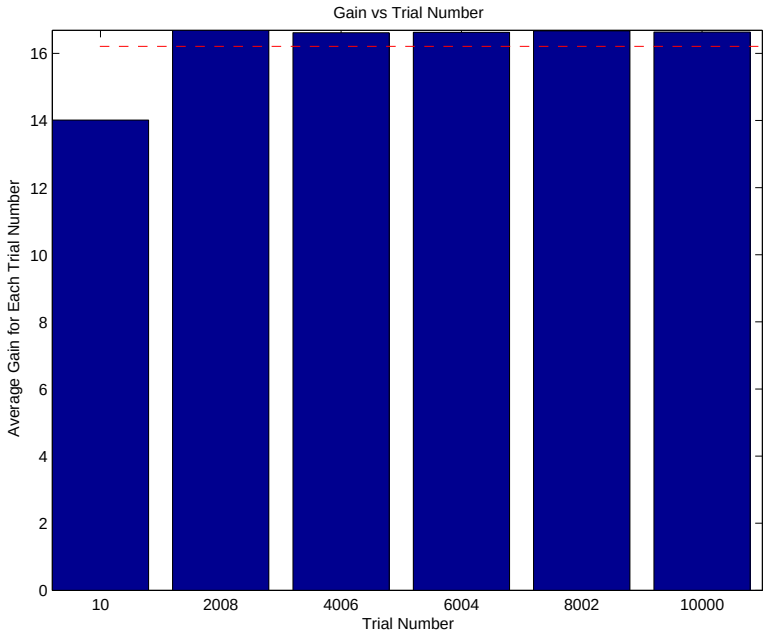

Figure A.3: Gain Averages for Different Number of Trials

APPENDIX B

ALLOWED BURST PROFILE CONFIGURATIONS

Tables B.1, B.2 and B.3 display the code configurations for Convolutional Coding, LDPCs and CTCs respectively. Note that a burst profile below might have more than one block length choice, so we list them as separate profiles. Columns where status is denoted by N means corresponding profiles are not supported yet.

For non-shortened polar codes, we choose as block lengths the nearest power of 2 to the block lengths for the CTC configurations above. Rate is chosen accordingly, in an approximate manner. For example, if 2/3 is desired, 21/32 is chosen since the denominator has to be a power of 2.

For shortened polar codes, we select the same configurations as their CTC counterparts.

Table B.1: Convolutional Coding

Profile No.	Modulation	Payload (bits)	Payload (bytes)	Rate	Status[1]
1	QPSK	6	48	1/2	
2	QPSK	9	72	3/4	N
3	QPSK	12	96	1/2	
4	QPSK	18	144	1/2	
5	QPSK	18	144	3/4	N
6	QPSK	24	192	1/2	
7	QPSK	27	216	3/4	N
8	QPSK	30	240	1/2	
9	QPSK	36	288	1/2	
10	QPSK	36	288	3/4	N
11	16-QAM	12	96	1/2	
12	16-QAM	18	144	3/4	N
13	16-QAM	24	192	1/2	
14	16-QAM	36	288	1/2	
15	16-QAM	36	288	3/4	N
16	64-QAM	18	144	1/2	
17	64-QAM	24	192	2/3	N
18	64-QAM	27	216	3/4	N
19	64-QAM	36	288	1/2	

Table B.2: LDPC Block Sizes (n denotes the coded block length)

Profile No	Modulation	n (bit)	n (bytes)	Rates
1	QPSK	576	72	1/2, 2/3, 3/4, 5/6
2	QPSK	672	84	1/2, 2/3, 3/4, 5/6
3	QPSK	768	96	1/2, 2/3, 3/4, 5/6
4	QPSK	864	108	1/2, 2/3, 3/4, 5/6
5	QPSK	960	120	1/2, 2/3, 3/4, 5/6
6	QPSK	1056	132	1/2, 2/3, 3/4, 5/6
7	QPSK	1152	144	1/2, 2/3, 3/4, 5/6
8	QPSK	1248	156	1/2, 2/3, 3/4, 5/6
9	QPSK	1344	168	1/2, 2/3, 3/4, 5/6
10	QPSK	1440	180	1/2, 2/3, 3/4, 5/6
11	QPSK	1536	192	1/2, 2/3, 3/4, 5/6
12	QPSK	1632	204	1/2, 2/3, 3/4, 5/6
13	QPSK	1728	216	1/2, 2/3, 3/4, 5/6
14	QPSK	1824	228	1/2, 2/3, 3/4, 5/6
15	QPSK	1920	240	1/2, 2/3, 3/4, 5/6
16	QPSK	2016	252	1/2, 2/3, 3/4, 5/6
17	QPSK	2112	264	1/2, 2/3, 3/4, 5/6
18	QPSK	2208	276	1/2, 2/3, 3/4, 5/6
19	QPSK	2304	288	1/2, 2/3, 3/4, 5/6
20	16-QAM	576	72	1/2, 2/3, 3/4, 5/6
21	16-QAM	768	96	1/2, 2/3, 3/4, 5/6
22	16-QAM	960	120	1/2, 2/3, 3/4, 5/6
23	16-QAM	1248	156	1/2, 2/3, 3/4, 5/6
24	16-QAM	1344	168	1/2, 2/3, 3/4, 5/6
25	16-QAM	1536	192	1/2, 2/3, 3/4, 5/6
26	16-QAM	1728	216	1/2, 2/3, 3/4, 5/6
27	16-QAM	1920	240	1/2, 2/3, 3/4, 5/6
28	16-QAM	2112	264	1/2, 2/3, 3/4, 5/6
29	16-QAM	2304	288	1/2, 2/3, 3/4, 5/6
30	64-QAM	576	72	1/2, 2/3, 3/4, 5/6
31	64-QAM	864	108	1/2, 2/3, 3/4, 5/6
32	64-QAM	1152	144	1/2, 2/3, 3/4, 5/6
33	64-QAM	1440	180	1/2, 2/3, 3/4, 5/6
34	64-QAM	1728	216	1/2, 2/3, 3/4, 5/6
35	64-QAM	2016	252	1/2, 2/3, 3/4, 5/6
36	64-QAM	2304	288	1/2, 2/3, 3/4, 5/6

Table B.3: CTC channel coding per modulation

Profile No	Modulation	Data block size (bytes)	Code Rate	Data block size (bits)	Encoded data block size (bytes)
1	QPSK	6	1/2	48	12
2	QPSK	12	1/2	96	24
3	QPSK	18	1/2	144	36
4	QPSK	24	1/2	192	48
5	QPSK	30	1/2	240	60
6	QPSK	36	1/2	288	72
7	QPSK	48	1/2	384	96
8	QPSK	54	1/2	432	108
9	QPSK	60	3/4	480	80
10	QPSK	9	3/4	72	12
11	QPSK	18	3/4	144	24
12	QPSK	27	3/4	216	36
13	QPSK	36	3/4	288	48
14	QPSK	45	3/4	360	60
15	QPSK	54	3/4	432	72
16	16-QAM	12	1/2	96	24
17	16-QAM	24	1/2	192	48
18	16-QAM	36	1/2	288	72
19	16-QAM	48	1/2	384	96
20	16-QAM	60	1/2	480	120
21	16-QAM	18	3/4	144	24
22	16-QAM	36	3/4	288	48
23	16-QAM	54	3/4	432	72
24	64-QAM	18	1/2	144	36
25	64-QAM	36	1/2	288	72
26	64-QAM	54	1/2	432	108
27	64-QAM	24	2/3	192	36
28	64-QAM	48	2/3	384	72
29	64-QAM	27	3/4	216	36
30	64-QAM	54	3/4	432	72
31	64-QAM	30	5/6	240	36
32	64-QAM	60	5/6	480	72

Table B.4: Equivalent Polar Code Configurations for the CTC Code Configurations Above

Profile No	Modulation	Code Rate	Data block size (bits)	Encoded data block size (bits)
1	QPSK	1/2	64	128
2	QPSK	1/2	128	256
3	QPSK	1/2	128	256
4	QPSK	1/2	256	512
5	QPSK	1/2	256	512
6	QPSK	1/2	256	512
7	QPSK	1/2	512	1024
8	QPSK	1/2	512	1024
9	QPSK	1/2	512	1024
10	QPSK	3/4	96	128
11	QPSK	3/4	192	256
12	QPSK	3/4	192	256
13	QPSK	3/4	384	512
14	QPSK	3/4	384	512
15	QPSK	3/4	384	512
16	16-QAM	1/2	128	256
17	16-QAM	1/2	256	512
18	16-QAM	1/2	256	512
19	16-QAM	1/2	512	1024
20	16-QAM	1/2	512	1024
21	16-QAM	3/4	192	256
22	16-QAM	3/4	384	512
23	16-QAM	3/4	384	512
24	64-QAM	1/2	128	256
25	64-QAM	1/2	256	512
26	64-QAM	1/2	512	1024
27	64-QAM	2/3	171	256
28	64-QAM	2/3	341	512
29	64-QAM	3/4	192	256
30	64-QAM	3/4	384	512
31	64-QAM	5/6	213	256
32	64-QAM	5/6	427	512

Bibliography

[1] IEEE 802.16-2009, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems ," 2009.

[2] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[3] ——, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, June 2008.

[4] K. Etemad, "Overview of Mobile WiMAX technology and evolution," *IEEE Communications Magazine*, vol. 46, no. 10, pp. 31–40, October 2008.

[5] IEEE Std 802.16-2004, "Standard for Local and Metropolitan area networks, Part 16: Air Interface for Fixed broadband wireless access systems," 2004.

[6] IEEE Std 802.16e-2005, "Standard for Local and Metropolitan Area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," 2006.

[7] S. Ahmadi, "An overview of next-generation Mobile WiMAX technology," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 84–98, June 2009.

[8] W. Kim, "Mobile WiMAX, the leader of the mobile internet era - [Wimax report]," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 11–12, June 2009.

[9] IEEE 802.16 Broadband Wireless Access Working Group, "IEEE 802.16m System Requirements."

[10] H. Sari and G. Karam, "Orthogonal frequency-division multiple access and its application to CATV networks," *European Transactions on Telecommunications*, vol. 9, no. 6, pp. 507–516, 1998.

[11] M. Maqbool, M. Coupechoux, and P. Godlewski, "Subcarrier permutation types in IEEE 802.16e," Tech. Rep., 2008. [Online]. Available: http://www.telecom-paristech.fr/data/files/docs/id7921208254315271.pdf

[12] B. Baumgartner, M. Reinhardt, G. Richter, and M. Bossert, "Performance of forward error correction for IEEE 802.16e," in *Proc. of International OFDM Workshop*. Citeseer, 2005.

[13] I. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[14] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[15] G. Forney and D. Costello, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[16] Iterative Solutions, "Coded Modulation Library." [Online]. Available: http://www.iterativesolutions.com/Matlab.htm

[17] B. Baumgartner, M. Mayrock, M. Reinhardt, and M. Bossert, "Performance of channel estimation for IEEE 802.16e," in *Proceedings 10th International OFDM Workshop*. Citeseer, 2005.

[18] E. Sasoglu, E. Telatar, and E. Arıkan, "Polarization for arbitrary discrete memoryless channels," 2009. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:0908.0302