ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

PERFORMANCE OF 5G CODES OVER A NOISY CHANNEL

M.Sc. THESIS

Mohamed SANFAZ

Department of Communication Systems

Satellite Communication and Remote Sensing Programme

JANUARY 2022

ISTANBUL TECHNICAL UNIVERSITY★ GRADUATE SCHOOL

PERFORMANE OF 5G CODES OVER A NOISY CHANNEL

M.Sc. THESIS

Mohamed SANFAZ
(7051818035)

Department of Communication Systems

Satellite Communication and Remote Sensing Programme

Thesis Advisor: Asst. Prof. Dr. Mustafa HELVACI

JANUARY 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GÜRÜLTÜLÜ BİR KANALÜZERİNDEKİ 5G KODLARIN PERFORMANSI

YÜKSEK LİSANS TEZİ

Mohamed SANFAZ
(705181035)

İletişim Sistemleri Anabilim Dalı

Uydu Haberleşmesi ve Uzaktan Algılama Programı

Tez Danışmanı: Dr. Öğr. Üyesi. Mustafa HELVACI

OCAK 2022

.

Mohamed Sanfaz, a M.Sc student of ITU Graduate School student ID 705181035, successfully defended the thesis entitled “PERFORMANCE OF 5G CODES OVER A NOISY CHANNEL”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below

Thesis Advisor :	**Asst. Prof. Dr. Mustafa HELVACI** İstanbul Technical University	....
Jury Members :	**Asst. Prof. Dr. Sebahattin EKER** Istanbul Technical University	....
	Assoc. Prof. Dr. Eylem ERDOĞAN Istanbul Medeniyet University	....

Date of Submission : 23 December 2021
Date of Defense : 28 January 2022

To my family,

FOREWORD

Finally, After all, the hard work is done. I’m delighted to finish my master's thesis. This is a big step for me in the academic field, which will reflect on both my academic and work life.

Completing a thesis in one of the busiest fields in the research area “coding techniques” and “5G” was a great adventure for me. That adventure led me to obtain good results in this area.

I would like to thank my parents ANIS and ZAINAB, without whom I wouldn’t have made it. They gave me an unlimited amount of support throughout this journey of university and every other thing in life. I can never thank them enough for what they have done for me.

Thanks a lot to my thesis supervisor, Dr. MUSTAFA HELVACI. He helped me and supported me through the thesis process. He treated me like I was his son, and he collaborated with me when my family had COVID.

Thanks to my brothers and sister for their support in this master's degree journey. Also, I want to express my gratitude to my previous professors who wrote the recommendation letters for me; without them, I wouldn’t even be at Istanbul Technical University.

January 2022 Mohamed SANFAZ

TABLE OF CONTENTS

ABBREVIATIONS

ASK	: Amplitude shift keying
PSK	: Phase-shift keying
BPSK	: Binary Phase-Shift Keying
CRC	: Cyclic Redundancy Check
eMBB	: enhanced Mobile Broadband
FSK	: Frequency Shift Keying
LDPC	: Low Density Parity Check Matrix
LLR	: Log likelihood Ratio
LTE	: Long Term Evolution
1G	: First Generation
2G	: Second Generation
3G	: Third Generation
4G	: Fourth Generation
5G	: Fifth Generation
QAM	: Quadrate Amplitude Modulation
QOS	: Quality Of Service
SNR	: Signal Noise Ratio
SC	: Successive Cancelation
SCL	: Successive Cancelation List

LIST OF TABLES

LIST OF FIGURES

PERFORMANCE OF 5G CODES OVER A NOISY CHANNEL

SUMMARY

At present, the need for mobile internet keeps increasing every day, especially with the rise of IoT devices, as it's estimated that by the year 2025, there will be more than 5 billion IoT devices connected to the network.

For wireless mobile communication, a huge bandwidth is needed to adapt the different rates for different applications. The 5G network will provide lower latency and also achieve higher speeds than previous networks.

In 5G wireless communication, both turbo codes and tail-biting convolutional codes failed to meet 5G standards even though they proved their efficiency for the LTE standard.

In 5G, a more advanced error correction method is needed for both LDPC codes and polar codes, specifically LDPC codes dealing with data channels and polar codes dealing with control channels.

As error correction and detection are the main requirements for 5G wireless communication, the BER performance against the (Eb/NO) performance is really important as you don't want to lose almost any transmitted block.

One of the methods used to check BER against EB/NO was to check an un-coded signal under various types of modulation, from BPSK up to 256 QAM; the higher the modulation, the worse the BER against EB/NO performance was getting.

With 5G packing more data now, even higher than 256 QAM is possible. A performance test of the codes that are being used in 5G has been simulated here. As is customary, the higher the modulation, the worse the BER against EB/NO. A 5G-NR scenario has been performed using BPSK modulation with an AWGN channel to demonstrate how the codes perform under the best modulation scenario. The 5G standard has been applied to both codes as base graph 1 and base graph 2 have been used for LDPC at different code rates. The same goes for polar as channels are in sequential order from worst to best as specified in the standard. The hardware performance for 5G is very challenging, so a single decoder has been used in both codes, with quantization implemented in both of them.

As a result of simulations of BER at both codes, different plots have been shown.

For LDPC codes, performance iterations had a noticeable improvement in BER levels starting at 10 iterations to 20 iterations and from 20 to 30 iterations. Not a huge BER improvement was seen, so 20 iterations have been implemented as the main iteration number for most of the graphs.

For LDPC codes, both base graphs were used. For rate half, with midsize block BG1, had a better performance; for rates 2/3 and 5/6, rate 2/3 had an overall better performance compared to rate 5/6, with 4096 block size providing the best results in both rates.

As for polar codes, successive cancelation was implemented for 256 and 512 block sizes with different rates. The lower the block size, the better the results were obtained for polar codes.

Successive cancelation list with sizes of 2, 4, and 8 was applied to the 256 and 512 block sizes. a big improvement can be noticed only from using a list of size 2.

List 4 and 8 had better results, but as the higher the list goes, the same scenario of higher levels of iterations occurs, not that much improvement in the BER levels and the time consumed in the mathematical operations becomes very large, so list size 4 was implemented for successive cancelation with CRC at different rates.

Polar codes had better BER performance at short block sizes, while LDPC codes had better BER performance at larger block sizes, as shown in the results figures.

GÜRÜLTÜLÜ BİR KANAL ÜZERİNDEKİ 5G KODLARIN PERFORMANSI

ÖZET

Kodlama, iletişim sistemlerinin ilk günlerinden beri içerisinde yer almaktadır. Günlük olarak kullandığımız tüm iletişim sistemleri, belirli bir zamanda kodlama ve kod çözme sürecinden geçmiştir.

Modüle edilmiş sinyalin kodlanmamış bir versiyonunu doğrudan göndermek yerine kodlama ihtiyacı basit şekilde,

Aynı BER'yi daha düşük bir sinyal-gürültü oranında elde etmemizi sağlaması ile açıklanabilir.

Claude Shannon, bir iletişim kanalının belirli bir gürültüsünde, dijital bilginin kanal üzerinden neredeyse hatasız, hesaplanabilir bir maksimum hızda iletişim kurmasının muhtemel olduğunu gösterdi.

Teori, gürültü ve bozuk veri seviyelerine karşın hata düzeltme yöntemleri ile elde edilebilecek maksimum verimlilik seviyesini gösterir.

Kapasitesi C olan bir kanal ve bir kanaldan R hızında gönderilen birkaç bit verildiğinde, eğer C kanalının kapasitesi R hızından büyükse, bazı kodlar BER'nin çok küçük olmasına izin verir.

Bu, teorik olarak, hızın sınırlayıcı kapasite oranının altında olması koşuluyla, herhangi bir oranda neredeyse hatasız bilgi göndermenin muhtemel olduğu anlamına gelir.

Oranın kapasiteden yüksek olduğu durumlarda küçük bir BER'in elde edilebilmesinin mümkün olmadığı görüşü ihmal edilemez.

Tüm kodların belirli bir düzeyde bir BER'i vardır, bu oranın kapasitenin üzerinde olduğu durumlarda bilgiyi kanal genelinde yüksek düzeyde güvenle iletmesi garanti edilmez.

Teori kapasite ile oranın eşit olduğu istisnai durumu içermemektedir.

İletişim sistemleri alanındaki bilim adamları ve uzmanlar, kullandıkları sistemlerin BER'lerini azaltmak için kodlar geliştirmek için çalışmaktadırlar. Bu alanda yapılan eklemeler ihmal edilemez.

Günümüzde özellikle IOT cihazlarının önem kazanması ile beraber, mobil internete olan ihtiyaç her geçen gün artmaktadır.

Hesaplamalara göre 2025 yılında 5 milyardan fazla cihaz ağa bağlanacaktır.

5G daha önceki nesillere kıyasla daha yüksek hızlara ulaşmasının yanında daha düşük gecikme suresine sahip olacaktır.

Farklı uygulamaların farklı oranlarına uyum sağlayabilmesi için kablosuz mobil iletisimde çok yüksek bant genişliğine ihtiyaç duyulmaktadır.

5G kablosuz haberleşmede (turbo codes) ve (tail biting convolutional codes) 5G standartını sağlayamamalarına rağmen, LTE standardında verimliliklerini kanıtlamışlardır.

5G'de, hem LDPC kodları hem de polar kodlar için, özellikle veri kanallarıyla ilgilenen LDPC Kodları ve kontrol kanallarıyla ilgilenen polar kodlar için daha gelişmiş bir hata düzeltme yöntemine ihtiyaç vardır.

Hata düzeltme ve algılama, 5G kablosuz iletişim için temel gereksinim olduğundan, (Eb/NO) performansına karşı BER gerçekten önemlidir, çünkü iletilen hiçbir blogu kaybetmek istemezsiniz.

BER'i EB/NO'ya karşı kontrol etmek için kullanılan yöntemlerden biri, BPSK'dan 256 QAM'ye kadar çeşitli modülasyon türleri altında kodlanmamış bir sinyali kontrol etmekti.

Daha yüksek modülasyon uygulandığında EB/NO performansına karşı BER kötüleşiyordu.

5G'nin artık daha fazla veri paketlemesiyle, 256 QAM'den daha yüksek değerler bile mümkündür.

Burada 5G'de kullanılan kodların bir performans testi simüle edilmiştir.

Alışıldığı gibi, modülasyon ne kadar yüksek olursa, BER EB/NO'ya karşı o kadar kötü olur.

Kodların en iyi modülasyon senaryosu altında nasıl performans gösterdiğini göstermek için bir AWGN kanalıyla BPSK modülasyonu kullanılarak bir 5G-NR senaryosu gerçekleştirilmiştir.

AWGN kanalı uygulandı. BER ile dijital bir iletişim sisteminin performansına karar vermek için, termal gürültü, gürültünün ana kaynağı olarak kabul edildiğinden, iletişim sisteminin performansının belirlenmesinde ana rol oynar.

Termal gürültü uygulanan kablolardaki atomların titreşiminden gelen rastgele istenmeyen sinyallerin birleşiminden gelir.

Bir BPSK altında gürültü olarak bir AWGN ile işlemlerin gerçekleştirilmesi simülasyon için ideal bir senaryo olarak kabul edilir

5G standardı her iki koda da uygulandı, çünkü temel grafik 1 ve temel grafik 2, LDPC için farklı kod oranlarında kullanıldı.

Aynısı polar için de geçerlidir, çünkü kanallar standartta belirtildiği gibi en kötüden en iyiye doğru sıralıdır.

5G için donanım performansı çok zordur, bu nedenle her iki kodda da (quantization) uygulanmış tek bir kod çözücü kullanılmıştır.

BER simülasyonları sonucunda her iki kodda da farklı grafikler gösterilmiştir.

LDPC kodları için, yinelemelerin performansı, 10 yinelemeden 20 yinelemeye kadar BER seviyelerinde gözle görülür bir iyileşme sağladı.

20 yineleme ile 30 yineleme arasında büyük bir BER iyileştirmesi görülmedi, bu nedenle çoğu grafik için 20 yineleme uygulandı.

LDPC kodları için her iki temel grafik de kullanılmıştır. Buyuk boyutlu blok BG1 ile yarı oran için daha iyi bir performans gösterdi.

Oran 2/3 ve 5/6 için oran 2/3, her iki oranda da en iyi sonuçları sağlayan 4096 blok boyutuyla 5/6'ya kıyasla genel olarak daha iyi bir performans sergiledi.

Kutupsal kodlara gelince, 256 512 blok boyutu için (successive cancellation)uygulandı ve farklı oranlarda blok boyutu ne kadar düşükse polar kodlar için o kadar iyi sonuçlar elde edildi.

256 512 blok boyutu için 2, 4 ve 8 boyutunda (successive cancellation list) uygulandı, ancak büyük bir gelişme yalnızca 2 boyutundaki bir liste kullanılarak fark edilebilir.

Liste 4 ve 8 sırasıyla daha iyi sonuçlara sahipti, ancak liste yükseldikçe daha yüksek yineleme seviyeleri senaryosu BER seviyelerinde o kadar büyük bir gelişme olmadı ve matematiksel işlemlerde harcanan zaman çok büyüdü; bu nedenle, CRC ile farklı oranlarda ardışık iptal için liste boyutu 4 uygulandı.

Burada ayrıca blok boyutu ne kadar düşük olursa, genel olarak daha iyi sonuçlar elde edilir.

Sonuç figürlerine gore polar kodları düşük blok boyutunda daha iyi bir performans gösterirken.

LDPC kodları daha büyük blok boyutlarında daha iyi BER performans göstermiştir.

Bu tezde, 5G kodları için BER performansını kontrol etmek için doğrudan düşük kıyı yaklaşımı kullanılmıştır. Bu model diğer kodlara genişletilebilir.

Önerilen yaklaşım, kanal için BPSK modülasyonunu ve AWGN'yi kullanmaktadır.

Bu kombinasyon, altında simüle edilecek kod için ideal senaryoyu sağlar.

Bu yaklaşım, simülasyon cihazının donanım yeteneklerinin bu tür işlemlere izin vermesi, polar kodlar için daha yüksek listelerin oluşması.

LDPC kodları için diğer oranlar ve bloklar bakımından aynı konu veya gelecekte önerilecek veya geliştirilecek kodlar için dünya üzerinde ki tüm araştırmacılar tarafından kullanılabilir.

Bu yaklaşım, kodun bir iletişim sistemi için en iyi koşullar altında nasıl performans gösterdiğini açıklamadır ve kod performansının erken aşamalardan itibaren kötü olup olmadığını belirli bir oranda mı yoksa bir blok boyutu aralığında mı görebilmenizi sağlamaktadır.

Bu sayede karmaşık yüksek QAM modülasyonlarını uygulamaya gerek kalmadan sorunları teşhis edip düzeltme imkanı elde edilmektedir.

Tezin amacı bir açıklama sağlamaktır. Bu çalışma, araştırma alanında bir yıldan fazla çalışmayı temsil etmektedir ve diğer yayınlar için referans olarak kullanılabilir.

1. INTRODUCTION

Coding is something that has been attached to communication systems since the early days of them. All of the communication systems that we use on a daily basis have gone through a process of encoding and decoding at a certain time.

The idea of going through the trouble of coding instead of directly sending an un-coded version of the modulated signal is very simple; coding enables us to obtain the same BER at a lower signal-to-noise ratio.

In 1948, Claude Shannon showed that at a certain noise level of a communication channel, it's likely for digital information to communicate almost with no error through the channel, to a calculable maximum rate.

The theory shows the maximum achievable level of efficiency of error-correcting methods in comparison to the levels of interference noise and corrupted data.

Given a channel with a capacity C and several bits sent through a channel at a rate R, if the capacity of channel C is bigger than the rate R, some codes allow the BER to be very small. That means, in theory, it's likely to send information almost error-free at any rate, with the condition that the rate is below the limiting rate of capacity.

The debate can't be neglected as if the rate is higher than capacity, a small BER can't be obtained. All codes have a BER at a certain level, and as the rate gets bigger, it's not guaranteed to transmit the information with a high level of confidence across the channel when rates are beyond the capacity, although theory doesn't specify the exceptional situation where the rate is equal to the value of capacity.

The simple scenarios, like sending the bit three times and using the best two out of three voting methods, aren't reliable error-correcting methods; they aren't likely to grant that the transmitted block can be sent error-free. Better performance lately has come from more developed methods such as Reed-Solomon and turbo codes. The LDPC codes, alongside the turbo codes, came close to reaching the theoretical level of Shannon theory, but this upgrade came at the cost of a higher level of computational complexity.

As the performance of FEC was more efficient than ARQ in the real-time application, the high data rate FEC mechanism became in demand.

With the combination of today's processors' computing power and these high-efficiency codes, it's likely to obtain a close range to the Shannon limit. LDPC codes were shown to reach 0.0045db of the Shannon limit with a mixture of extremely tall block lengths and a binary AWGN [1].

Scientists and experts in the communication systems field have been working on developing codes to reduce the BER of the systems they use. Every addition to that field can't be totally neglected. For example, LDPC codes were invented back in 1962 by Robert Gallager; they were abandoned at that time because of their computational difficulties. 35 years later, in 1997, as FEC mechanisms were popular with high data rate applications, exact codes were brought back again by Mackay and Neal [2].

With the huge development in technology, they have now been selected as one of the 5G coding methods.

In today's world where smartphones are used daily by millions of people, when a lower signal-to-noise ratio is achieved, that means less energy usage, which gives us more time before the battery drains. That time could be spent on other activities such as listening to music or playing video games.

As we go deeper into coding and to understand the needs of the standards, it's not only about obtaining the same BER at a lower signal to noise ratio, the demands also increase with the growth of technology .

Now it's not only about doing encoding and decoding to get a lower BER, but also about the needs of the application that you are dealing with, as some of these codes are highly efficient with a large block size, some perform better under lower block sizes, and also the divergent is seen between different codes under varying rates [3].

Because 5G-NR is similar to the previous technology, LTE, in terms of providing a better user experience, better channel coding combined with a better system structure is required. With the high data rate in 5G-NR to improve buffer capacity, the encoding processing and decoding processing throughput is split between both of them.

As the needs aren’t only obtaining a lower BER, one of the solutions that has been selected to serve the needs of the 5G standard is 5G NR LDPC codes. When it comes to low power dissipation and high coding gain, they are one of the most well-known solutions.

5G systems also need an upgrade from their previous on-channel capacity, with polar codes being the first codes that almost reach the Shannon capacity limits. They caught the attention of researchers for 5G systems, and later they were chosen as one of the chosen techniques for that system.

Based on the previously mentioned techniques, we can summarize that the main techniques used for coding in 5G NR are 5GNR LDPC codes and polar codes.

LDPC codes are triumphant 3G FEC codes with a parity check matrix. In 5G NR LDPC, these are done by base graphs and are made up of blocks.

With the invention of polar codes by Erdal Erkan, we made huge steps forward when he was able to prove that we could withdraw codes with the property of obtaining capacity under low coding and decoding difficulty with the use of channel polarization [4].

Based on the introduction of the LDPC and Polar codes, the thesis aim is to use the codes to introduce an effective contribution to check BER performance through a noisy channel with low modulation (BPSK) under the 5G-NR standards.

1.1 Purpose of Thesis

1 gbps was the highest in 4G communication. Due to the fast development of technology lately, internet speed has also changed. 5G will support a peak data rate of 20 gbps.

As the data rate increases to provide contentment to the user applications that run in real-time, the end-to-end latency was improved to 0.5 milliseconds, which is way better than the 4G standard. This actively demonstrates that out of 100,000 transmitted blocks, they suffer a loss of 1 block. This will decrease latency. Also, as many different other applications are used for 5G, the codes have to be adaptable in both code word and rate [3].

To meet these requirements, 5G NR employs both LDPC and polar codes for error correction, with LDPC used for data channels and polar codes used for control channels.

This thesis is a high-level, extensive study on the performance of LDPC and polar codes. For 5G-NR systems, they will use the 5G NR standard as Base Graph 1 and Base Graph 2 for LDPC codes and channel polarization for polar codes, so any where in the world, if they have high hardware ability, they can use the standard with a high modulation order and watch how these codes perform.

1.2 Unique Aspect

The approach used in this thesis can be applied to these codes under other block sizes, rates, or other types of noise scenarios. With the higher hardware ability, higher iterations, lists and also higher precision can be obtained as the more blocks are sent, the more accurate results are achieved.

Here, the performance of LDPC and Polar codes was simulated with Matlab with different message sizes using the 5G NR standard as base graph 1, base graph 2, and polar reliability sequence for the polar codes. So the performance of LDPC and Polar codes is simulated with Matlab with different message sizes using the 5G NR standard, performing all of these operations with the AWGN channel.

The AWGN channel is implemented. It plays a main role in deciding the performance of the communication system since, to decide the performance of a digital communication system with BER, thermal noise is considered the main source of the noise. It comes from the combination of random undesired signals from the vibration of atoms in the implemented cables.

Operating with an AWGN as the noise under a BPSK is regarded as an ideal simulation scenario.

2. LITERATURE REVIEW

This chapter demonstrates a complete summary of ideas, approaches, and chosen studies in BER performance. The particular reason for this circumstance is to provide accurate and dependable information as a starting point and develop a powerful understanding that helps us achieve the work's ultimate goal.

2.1 Shannon's Theory

Shannon, through his work in digital communication, found and also proved that we can receive the sent bits correctly under a line in which the bits are reliable to be sent.

2.1.1 The revolution of the mathematical theory of information

In July and October of 1948, Claude Shannon released "A mathematical theory of communication" [5]. This was the spark that lit the light of the information theory era. a hypothesis that unifies and intersects in significant ways with computer science, probability, and statics, as well as many other fields. The field of information theory continues to pave the path for the advancement of communications and additional types of information technology.

Before 1948, these were some of the main types of communication systems: telegraph by Morse in 1830, AM radio in the early years of the 1900s, television, which was between 1925 and 1927, up to frequency modulation by Armstrong in 1936, and pulse code modulation by Reeves between 1937 and 1939.

From those discoveries, we can see some crucial factors that would be crucial for the development of information theory.

The frequency modulation and the pulse code modulation exhibited that in the quest for more dependable communication, sent bandwidth is just another degree of flexibility given to the engineer. Pulse code modulation was also used to send analog continuous-time signals as it was the first digital communication system to account for frequency symbols to be encoded; the Morse code provided an effective method of encoding.

Before Shannon's theories,In 1924, an argument by H. Nyquist stated that the rate of transmission is related to the amount of signal level logarithm at a given time. [6].

He is the one who wondered how much faster telegraphy transmission rates could be attained by substituting Morse code with "optimum code" [7].

Prior to Shannon, another paper, this time by R. Hartley, used words like "rate of communication" and "capacity of a system to transmit information" [1].

R. Hartley achievement was that the evolved perspective is beneficial in that it gives a quick and easy way to check regardless of whether or not assertions made for sending probabilities of a sophisticated system exist in the realm of scope.

His fundamental findings with the RLC circuit Hartley observed that capacity is directly related to the bandwidth of the channel, but before talking about the subject of capacity, he introduced a quantitative measure of information [8].

$$H=n \log s \tag{2.1}$$

where H equals the amount of information at n selections and s denotes the number of symbols available in each selection.

Shannon established the revolutionary conceptualization of the communication process where a mean-square fidelity criterion is employed in the communication process. After finishing his PhD studies, he got back to the field. In the year 1948, when he published his capacity theory, other scientists had already set the essential compromises for the theory of communication, such as the bandwidth, rate of transmission, and SNR [7].

2.1.2 Compression without loss

The technical challenge is unaffected by semantic elements of communication; the fact that the main message was chosen from a list of different messages is the key feature. The important insight made by Shannon is that the source of information should be represented by a random process [7].

Shannon noticed that to use the abundance of sources besides the frequency of the symbols, the memory needs to be taken into consideration; but before starting to solve this dilemma, he contemplates only a random variable with values of n with a probability that goes from p1 to pn with H encoded bits, and he defines the lack of predictability as:

$$H = -\sum_{i=1}^{n} pi\, log\ pi \tag{2.2}$$

With Shannon theory 3, we are ready to accept a non-zero chance of not being able to retrieve the main sequence. We may reduce that likelihood by raising the block length, and hence the latency and difficulty of encoding and decoding operations, but because the theory only addresses the scenario of a code that is not optimum, that is not the best choice for the process.

In his fourth theory, Shannon figured out the solution as we need a probability of error that is below 1. Asymptotically, we are not able to encode with a coding rate that is below the entropy. This definition is widely known as the "powerful debate source coding theory."

The previous conversation is restricted to only fixed-length codes, but Shannon saw that by enabling an encoded sequence of variable length, it’s feasible to attain a probability of zero errors, and this comes without the need to increment the encoding rate.

Due to its importance in the field of data compression, Shannon Theory 3 was tested to see if it is compatible with a memory-based source. To be able to do this process, the entropy rate takes the place of the entropy random variable. Shannon demonstrates that the stationary process entropy rate is the same as the entropy of a sole source symbol when the previous symbols are given; To explain the characteristic of similarity to Shannon's 3 theory, McMillan used the term "AEP," or "asymptotic equipartition property." He exhibited the broad generality of the Shannon 3 theory that the AEP is contented by all ergodic stationary procedures with the use of a limited alphabet [9] [10].

With his studies related to the techniques used in telegraphy, Shannon faced two crucial problems in the area of fixed-to-variable coding: the first is the texture of a minimum average range code, and the second is the changeable extent of the source coding notion. Although L.Kraft was able to fix the structure of the first problem in his thesis [11].

As of now, the chapter only specifies the term "data compression" when the fixed block size is encoded, whether to a fixed or variable-length string, but when it comes to encoded data block designing as a requirement, the variable to fixed encoding method is convenient.

The main idea is to implement commas into the source sequence, so in the variable to fixed source coding, these terms are in the ownership of a dictionary with a set size. When the size of the dictionary is given, the "Tunstall algorithm" will choose the next term based on two factors: that no term is another prefix and that a prefix exists in the dictionary for every source sequence. This method has a slight performance advantage compared with fixed to variable codes [12].

For data compression methods that aren’t aware of source distribution, the term "universal" has been attached to them. The use of universal source code has a lot of practical applications.

Furthermore, because it watches the source output, the encoder may use the utility of it. Thus, it is able to discover how the source is distributed and adjust to it. This applies to the decoder, as its output is a rebuild of the source sequence with no loss.

Assume that we take a description using variables of source ambiguity. In practice, it’s helpful to take uncertainty branches that have distributors shown by a various number of factors, as an example, "Markov chains of different orders" [7].

We could conjure up an image of a universal compression process with two steps. The first step is to evaluate the unknown parameter with the use of the source sequence and explain it to the decoder. The second step is to use a code implemented in the source distribution to compress the incoming data from the source sequence, but there is a trade-off as the higher the complexity of the method, the less efficient it will be to compress the source. It will also require more time to clarify the parameters for the decoder.

Rissanen demonstrated that there are primary reasons to take the MDL criterion when selecting the model. In 1976–1978, A. Lempel and another scientist named J. Ziv introduced the universal most commonly used source coding procedure, with little variation between them [7].

Compared to other ways in this sub segment, this algorithm does not rely on the estimation or approximation of source distribution; it relies on parsing the sequence that came from the source, regardless of how good and simple it is. This theory is not the bottom line of universal source coding.

2.1.3 Dependable communications

Shannon's glibber of communication's core problem is shown in figure 2.1. Shannon completed the graph he made by showing a developed concept called the "channel." It compensates for any predictable or random transformation that may cause the signal to be corrupted during transmission. The job of a transmitter is to add what is called redundancy.

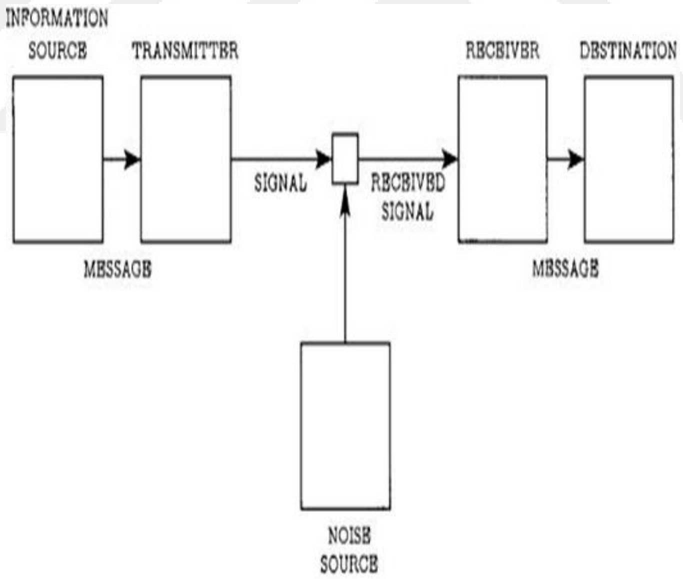


Figure 2.1 : Shannon's observation of the main dilemma of communication [7].

To achieve the perfect encoding, redundancy must be included to battle the specific noise structure involved. Usually, a delay is necessary to get close to ideal encoding. Now, it has the capability of enabling a wide sample of noise to influence the signal before any decisions about the signal are being made on the receiving end.

Shannon's conclusions on communication that is reliable are initially developed in the setting of memory channels that are discreet.

$$C \ = MAX(\ H(x) - Hy(x)) \qquad (2.3)$$

As maximum is measured in terms of all probable information sources, which are consumed as, channel input, it says that at any rate below C the information can be sent across the channel with only a small amount of error. Later Shannon later provided a more powerful statement related to channel capacity.

$$log\frac{N(T,q)}{T} = C \qquad (2.4)$$

where T is the duration and q is the probability of error, N (T, q) stands for maximum block size. Shannon explains that his achievement in this part is introducing the technique of random ending, where the probability of error is averaged according to a codebook option and shown to disappear asymptotically with T when C is below the transmission rate [7].

Without any doubt, the main channel that landed the biggest achievement in the history of information theory is the Gaussian channel.

Shannon released the continuous-time optimal white Gaussian channel with a tight band restriction, and he used the sampling theorem to display the identical channel as a discrete-time channel when it was sampled at double bandwidth.

Shannon continues to get his known formula related to the capacity of white Gaussian channels that are power constrained.

$$C = W \log\frac{(P+N)}{N} \qquad (2.5)$$

where W is the channel bandwidth, P is the transmission power and N is the noise power.

Besides that, Shannon also researched white Gaussian channels related to amplitude bonds instead of power bonds. He noticed that at low SNR, the capacity is given by equation 2.5.

One of the main outcomes that was left unproven in [5] was the converse channel. The authenticity of a powerful converse that is sketched in Shannon theorem 12 for a binary memory less channel was done by J. Wolfowitz in 1957 [13].

2.1.4 Channel capacity

An information theorist can make the possibility of error go all the way to zero by giving a typist text that is redundant, generated from the original one using an algorithm that takes into account the typist's error mistakes[7].

In the years after Shannon’s historical paper in 1948, for the next half-century, the goal was to find a practical code that reached the capacity. In recent years, with the progress of codes like turbo and the comeback of the newly developed version of LDPC, it has been obtained [14].

As AWGN is used in this thesis for this type of channel, the characterization of the code can be summarized into two main factors: SNR and spectral efficiency (ɲ).

Instead of SNR, another close parameter is mainly utilized in power-limited systems, which is Eb/No [14]. where Eb is signal energy per bit, and No is noise spectral density.

The algebraic coding paradigm dominated the first decades of channel coding; the theory was primarily focused on linear blocks over a binary field; the optimal modulation to detect BER is with this equation.

$$Pb\ (E) = Q(\sqrt{SNR}) = Q(\sqrt{(2Eb/No)}) \tag{2.6}$$

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{y^{-2}/_2}\ dy \tag{2.7}$$

2.2 Coding Techniques

In this chapter, we will discuss a variety of coding techniques that were used in the previous technology.

As another form of communication progresses, mobile technology has developed over time with five different generations of upgrades.

The 1G, or first generation, was only able to transmit voice as the transmission was analog. Back in 1980, when 1G was launched for the first time in communication history, it was the first time people around the world were able to use mobile phones.

When 2G was launched, the primary service remained voice transmission. Other services were included, like the ability to send a text message and call holds [15].

Back to where it all started, one of the biggest jumps in communication was 3G technology. It was a time when the availability of data for specific information was at an all-time high, and one of the major issues in this type of communication was transmission power. But with coding, we achieved a better quality of service for certain transmission powers [16].

The progress of coding continues with the start of Turbo codes in 1993, which came from Convolutional codes. They are used in data channels for LTE and other fields in 4G communication, such as digital video broadcasting [15].

Finally, around the year 2020, when 5G was launched, LDPC and polar codes were the main techniques used for 5G technology. Both of these techniques will be discussed in Chapter 3.

2.2.1 Block codes

Due to the noise that is happening inside the channel, the output may be different from the input. To provide data transmission that is considered reliable, the message should be encoded so that it is implemented in a sequence. The sequence is transmitted through the channel and is called a code word. If two code words have the same number of symbols, the code is called a block code with length n. As one of the most important factors in decoding is MD, this identifies the ability of the code to correct an error that happened during the transmission. The code rate also plays an important role because it slows down the transmission of the information due to the redundancy, and according to Shannon, the probability of error moves exponentially to zero the longer the code word gets. Because the encoding implementation is not as complex as the decoding procedure, but because the mapping needs to be stored in memories, sometimes even when the message size is bigger than 50, the table gets large. That's why there is some kind of algebraic structure for these types of codes that assists in facilitating the construction of these codes so that they become easier to realize [17].

Linear block codes are one of the most important in this section of codes. A code that has a generator matrix is known as a systematic code, but not all linear codes are systematic because the generator matrix might not appear as a singular matrix.

Cyclic codes, which are a subclass of linear codes, have a simple encoding and decoding mechanism. The code is called cyclic if every shift from the code is also another code.

Moving to Hamming codes, they have the ability to correct a single error, assuming a message consists of bits from d1 to d4. Hamming adds an extra 3 bits to the message and those added bits are obtained by adding the bits of the message, so the coded message will appear as d1.. d4p1p2p3 It will be able to detect a single error and have the ability to correct that error that happened during the transmission. A double error will cause a failure in the transmission, but it can be detected by adding an extra bit [15].

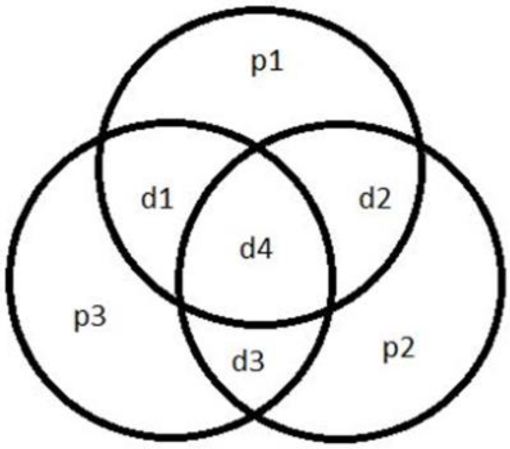


Figure 2.2 : hamming coding process [15].

Another important type of cyclic block code that was invented in 1960 was Reed-Solomon. Even though they are old, they are still one of the most used codes in their category, mainly because of their simplicity. They don’t have many roots as they are considered low-degree polynomials and they achieve singleton bound [17].

$$Mq(n,d) \leq q^{n-d-1} \quad (2.8)$$

$Mq(n, d)$ is the maximum number of possibilities for a codeword in a q-ary block code that has a minimum distance d and a length n.

Reed-Solomon codes have many applications. They were the key factors in the compact disc's success, and they were one of the first uses of error correction in a mass product that is used by many consumers, like the DVD with cross-interleaved It adds an extra redundant byte to every 3 bytes. This method can correct up to 4000 bits [18].

The main type of block codes that are used in this thesis; which is LDPC codes will be discussed in chapter 3 of this thesis.

2.2.2 Turbo and convolutional codes

After being shown to the public by Elias in the year 1955, convolutional codes took place from 2G to 4G networks. The encoder is composed of an XOR process with memory elements in it. The input stream goes through the memory register as each one of these registers contains a single bit, then it’s given to an XOR gate[15].

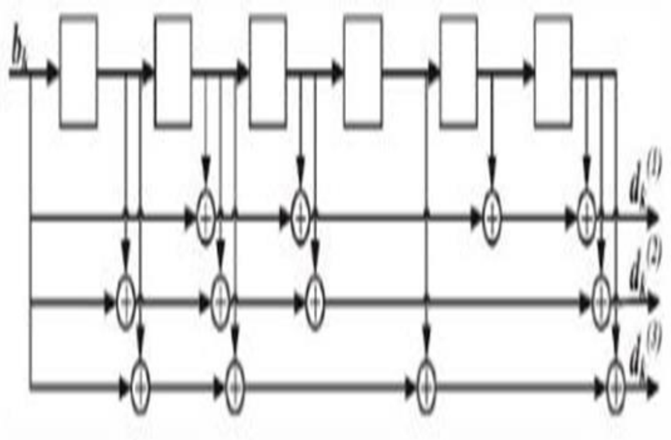


Figure 2.3 : Convolutional encoder [15].

The convolutional codes depend on three main factors: the first is the number of bits; the second is the available number of memory registers; and finally, the number of bits in the output [15].

Since the output depends on the input symbol and the meanings stored in shift registers, It looks like a block code, but a block code with a length of n depends only on the current k symbols, so each code block is formed independently from the others. A convolutional code that has a length n depends both on current and previous data of block length k [17].

As for the decoding process, there are different techniques for convolutional codes. Viterbi decoding, which is considered the best algorithm for convolutional codes, This method was developed by A. Viterbi in the year 1967 [17].

$$r = v + e \tag{2.9}$$

where r is the received sequence, v is the code sequence, and e is the error sequence, the main disadvantage of this Viterbi decoding is that BER increases as the input increases [15].

On the other hand, the advantage of Viterbi as a decoding algorithm is that the complexity of decoding doesn’t depend on the number of symbols in the sequence in an exponential way; it can be done in two different ways: hard decision and a soft decision technique. Better performance is obtained when the soft decision is applied; it makes use of the new data given by the soft decision demodulator. Generally, it works the same as the hard decision algorithm. The slight difference between both techniques is that in soft decision hammer distance isn’t employed as a metric. Another technique is called "sequential decoding." This was even released before Viterbi decoding, but due to the growth of usage of Viterbi decoding in many applications, this method is still restricted. This type of decoding is known to go directly to the depth of the tree, and if the metric indicates that this is not the choice, it goes back to choosing another path. Due to this process, sequential decoding isn’t the best option and hasn’t grown in usage because this process may miss the best path. List decoding is also another method that is used for convolutional codes. Here a distinct path is taken at each phase of decoding. Here the decoder only takes into consideration the branches that are from the best paths compared to Viterbi. It’s a less complex one, but on the other hand, it’s not the most favorable one. The reason is that some paths aren’t even taken into consideration [17].

The other type in this subsection is Turbo codes. They were developed and released in the 1990s. They originated from convolutional codes as the encoder part is made from two convolutional encoders and the encoders, are separated by an interleaver [15].

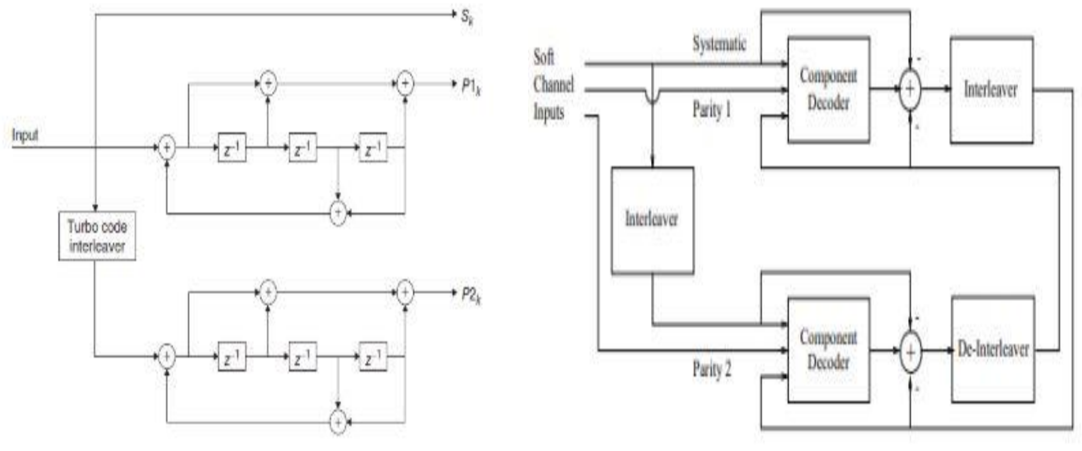


Figure 2.4 : a turbo encoder and decoder [15].

Turbo encoders are the result of two systematic convolutional encoders, and they perform iterative decoding. Turbo codes are commonly used for the transmission of finite length code words [17].

At a rate of 1/3, the encoder produces three outputs: two parity bits, steam, and a systematic bit.Because the result of the rate is slightly less than this value, besides the main two encoders, there is an extra interleaver. On the decoder, a reversed operation occurs; the parity 1 stream is interleaved with systematic steam, and parity 2 steam goes to the second decoder as it will perform the interleaver for the first output, which creates output for it. This is known as iteration [15].

In turbo codes, as the iteration method is employed, the higher the iteration number goes, the better the BER is obtained [17].

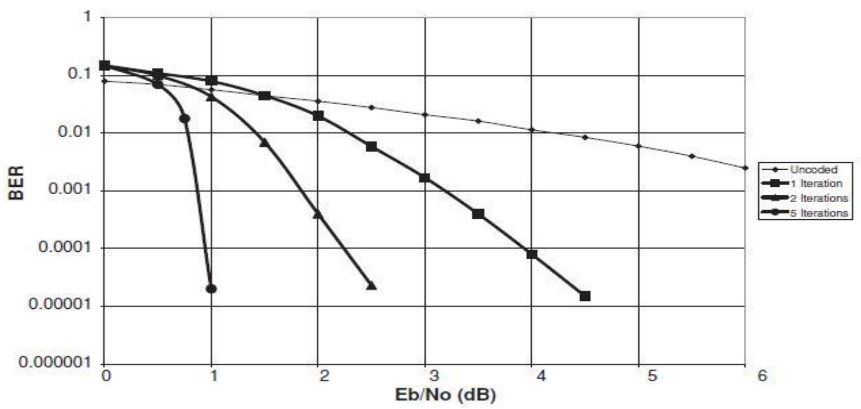


Figure 2.5 : BER vs. Eb/No for various iteration of turbo codes with BPSK, AWGN channel [17].

2.3 Modulation Effect

As time goes by, with the development of wireless communication, the transmission and reception of given, system quality are critical. Modulation techniques had a huge impact on the growth, providing better capacity, quality, and speed for the system. As for digital communication, the critical standard for their system is BER performance. The criteria compare how the system works with the noise implemented [19].

In this section and throughout the whole thesis work, the AWGN channel is implemented. The AWGN channel is well-known as a good place to start learning about performance connections [20].

The reason for using modulation is to send data from a given source over a channel. Modulation can be split into a two-phase process; phase one is band pass, which interprets the incoming information, whether they are digital or analog, into small frequency waves. The second phase is band pass that adjusts giant frequency corresponding with the waves taken out of baseband [2].

A given signal can be modulated based on three different factors: the amplitude, frequency, and phase from which we can obtain these modulations: ASK, PSK, and FSK. ASK is referred to as on-off where 0 means the carrier is multiplied with a small amplitude and the opposite operation for 1, FSK frequency shifting in the range between 1 and 0 and PSK Here, modulation occurs with the phase [19].

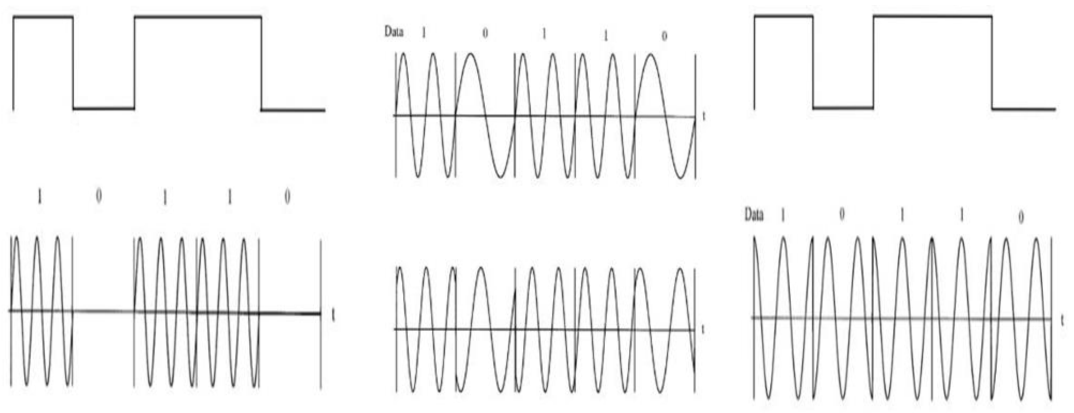


Figure 2.6 : ASK, FSK AND PSK [19].

But why do we go through this long and complicated process instead of directly sending the waves? The reason is directly related to antenna size. Generally, the antenna wavelength is divided by four. So let’s say that the output of a baseband has a frequency of 500 Hz, then according to the relation c=λ f then λ is 600,000, which means an antenna size must be 150 km, which is an impractical thing. On the other hand, if band pass later is applied, 2.5 GHZ needs only an antenna with a 3 cm dimension [17].

2.3.1 PSK and QAM modulations

This type of modulation that phase shifts with the flow of bits that are sent is one of the basic and most famous types of this subsection, BPSK. The constellation here is two, and when AWGN is included, checking the sign is the only requirement to produce a decree. For this type, there is also a quadrate version, which is an extension of BPSK. Here we get a constellation size of four points, and the constellation may be shown as two perpendicular BPSK constellation dots.

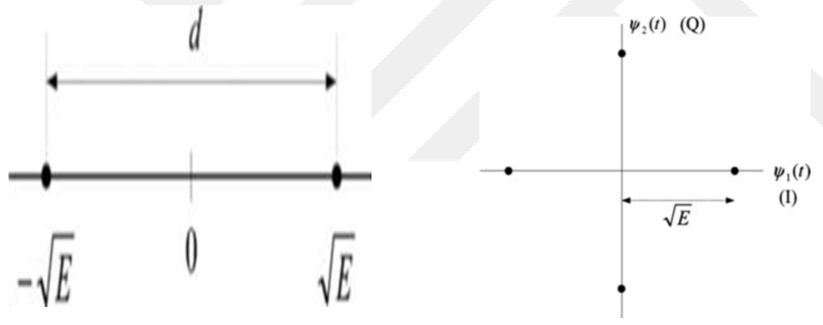


Figure 2.7 : BPSK and QPSK constellation[17].

These two types are the most famous types in the section of PSK. Depending on the steam of the sent bits and other factors, there are other PSK modulation types.

As the attention went to phase modulation more than the amplitude version of it, in 1960, a researcher named C.R. Cahn published a paper mixing both these modulations together. By doing this, it iterates the PSK constellation, which is the known form of the circle.

It is primarily used as an extension to QPSK; QPSK can be represented as a 4-QAM modulation in addition to a bass band, whereas QAM has other M-QAM. Today, even 1024 QAM is available.

2.3.2 BER performance with different modulation

As mentioned previously, BER is the main character when it comes to digital communications performance. In this subsection, we will see how BER performs over various modulation types on the AWGN channel; AWGN has a variety of natural origins.

BER can be identified as the number of bits that are in error divided by the total number of bits that have been sent.

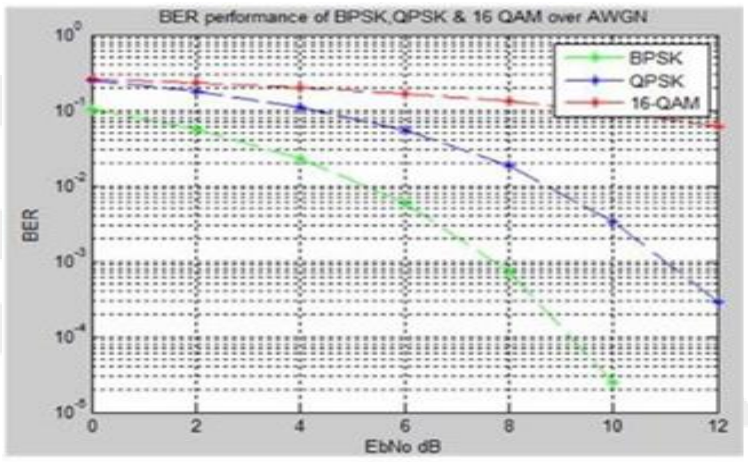


Figure 2.8 : BER performance over AWGN BPSK, QPSK,16QAM[21].

BPSK is considered the "most effective modulation" when it comes to communication systems BER [19].

The results of the modulations shown in figure 2.8 show that BPSK is the best and has a more efficient performance when compared to other modulations[21]

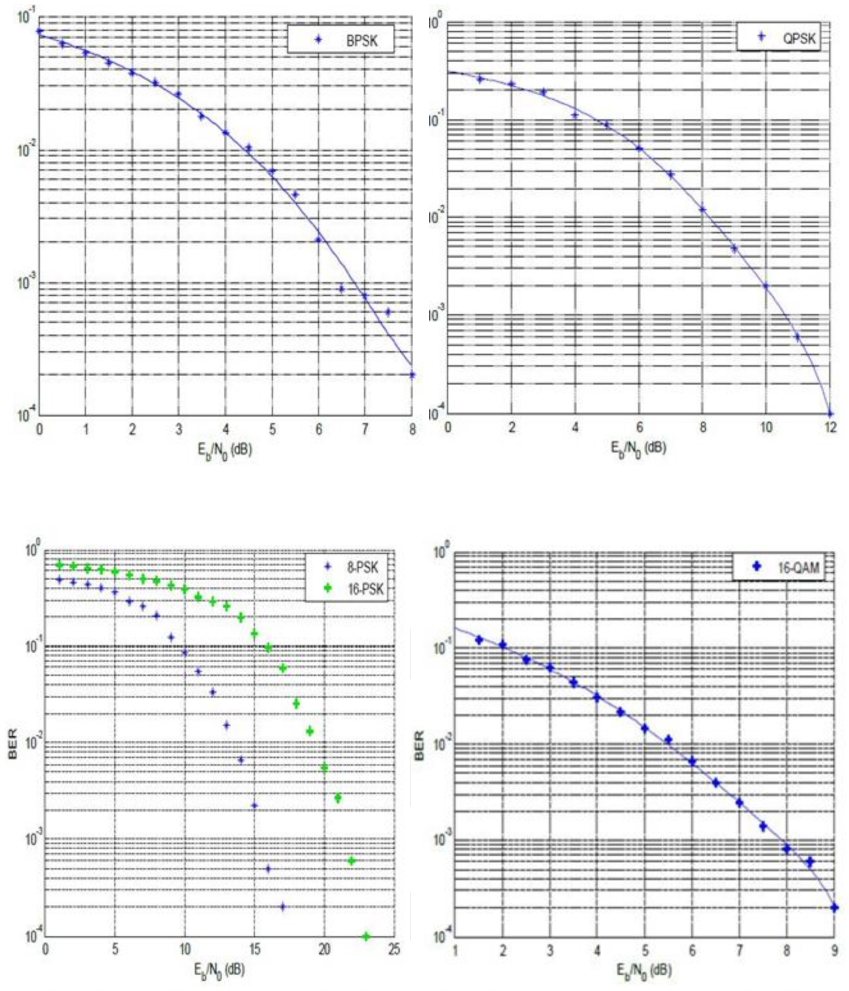


Figure 2.9 : BER performance for various modulations[19].

These studies, and many others by researchers, show that BPSK performs better than other modulations. This gives us an extra bonus to ensure that when the coding technique is done, this type of modulation will show the best form of the results. This was the main reason for applying this type of modulation in this thesis.

3. LDPC AND POLAR CODES

This chapter will go over LDPC and Polar codes, which are both techniques that are used.In the 5G technology coding process, the types of coding and decoding methods will be discussed. The specified types of techniques used in the thesis will be discussed more in the next chapter alongside the results.

3.1 LDPC Codes

Let’s start this section with an overview of these types of codes. They were first introduced by Gallager in the year 1962, and during that time they were directly ignored due to their unserviceable encoding and decoding schemes. Yet, they were brought back to action again in the 1990s [7].

With the development of technology, Turbo and Convolutional codes did the job for the LTE communication systems, but with 5G, both of these codes weren’t sufficient for the communication demands, and one of the codes that were chosen for this type of technology was LDPC codes for their many advantages, like achieving a high coding gain, a large amount of throughput, and the fulfilment of a small latency [21].

When Gallager introduced this technique, he was very confident that at the provided rate, there weren’t any codes that could correct the errors more than his codes; at that time, it was difficult to try that and test the theory, but after Mackay brought them back and showed how their performance was near capacity, like Turbo techniques, the research and work continued [11].

Before getting into how they consist, let’s talk about the advancements in LDPC codes. The simplicity of their decoding scheme means they can use both soft and hard decision decoding; they also provide good error performance as there is always a good hamming distance between the codes. This advantage is the main one over other codes, especially Turbo codes.

Also, they use an iteration system in decoding. This makes it practicable to decode a large LDPC with belief propagation, which is a good thing as it moves towards a less complex decoding also with many other advantages, like the availability to use different code rates [22].

Let’s now talk about how these codes are formed. Here, the code word is obtained by the combination of the information and the block check numeral. The check numeral and the information numeral are specified previously. This can be represented as what is known as the parity check matrix [23].

$$\begin{array}{ccccccc}
\overbrace{x_1 \quad x_2 \quad x_3 \quad x_4}^{INFO} & & & & \overbrace{x_5 \quad x_6 \quad x_7}^{CHECK}
\end{array}$$

$$\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \quad \Leftrightarrow \quad \begin{array}{l} x_5 = x_1 \oplus x_2 \oplus x_3 \\ x_6 = x_1 \oplus x_2 \oplus x_4 \\ x_7 = x_1 \oplus x_3 \oplus x_4 \end{array}$$

Figure 3.1 : Parity-check matrix example.

The LDPC is characterized by two main factors. The first is low density, which means that zeros are higher than ones. This insures a much lower computational difficulty. The second factor is that all codes must satisfy the below equation.

$$CH^T = 0 \tag{3.1}$$

where C here stands for the code word and H^T is the transpose of parity-check matrix.

Instead of the classic matrix representation LDPC codes, there is another way to do that, which is called the Tanner graph. Two types of nodes are represented: the variable or the bit node. Every one of those represents a column of a given parity matrix. The other node is called the check node. Every check node represents a row. So far, in the scenario of a code word n that contains k origin bits, there are going to be n bit nodes, and the result of subtraction between n and k is the check node. When there is a non-zero number in the parity, an edge will be shown in the graph between the bit and the check node. This graph will simplify the decoding procedure as the message will lie between the edges. Another benefit of this type of graph is that

Mackay noticed that short rounds make decoding performance worse, so when building a Tanner graph for good decoding, short rounds should be avoided [24].

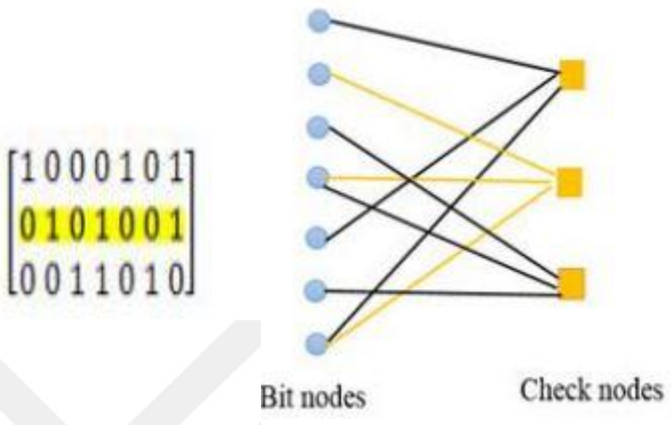


Figure 3.2: Parity matrix with its Tanner graph representation [3].

Tanner graphs were originally used to present LDPC from the standpoint of factor graphs; an example similar to Tanner graphs is factor graphs that come with a cycle graph that consists of a series of linked nodes that begin and terminate at the same node, and other nodes cannot be replicated more than once [23].

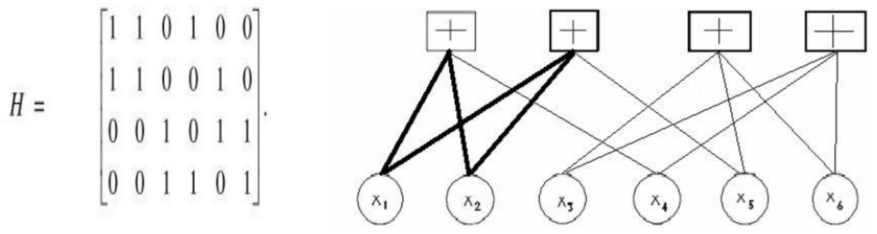


Figure 3.3 : Factor graph for the matrix with a cycle 4 [23].

LDPC codes may be referred with (n,k) the code word length and message bits, also they could be represented as (n,wc,wr) as wc and wr stands for column weight and row weight respectively .

When the LDPC comes in two types, regular and irregular, the weight of a column and the weight of rows are equal in parity. It’s called a regular LDPC code. The weight here represents the number of non-zeros, whether in a row or a column.

When it comes to the irregular type, the weight of columns and rows might change with different rows and columns [24].

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

Figure 3.4 : Parity matrix of size 4x8 for an irregular LDPC Code.

Specifically, for the regular type, the parity has a specified weight for rows and columns, and the amount of one between any two columns has to be 1 or less; for the irregular type, the variation in weight makes the process of decoding more complicated, but it has a better overall BER execution [22].

Before moving into the parity-check matrix into the next subsection of the parity-check matrix, we should mention these LDPC codes: The first is the "Irregular Repeat-Accumulate" LDPC code. For this type, a generator matrix is not needed as it employs shift registering and it works on the basis of a technique that is adopted with the rate The second is spatially coupled LDPC codes. This algorithm uses threshold saturation to provide a space-achieving manner for decoding. It’s also been announced as a nominee for NOMA [3].

Finally, quasi-cyclic LDPC has received attention in the last couple of years. They have had many advancements against other types of LDPC, mainly in hardware use when decoding and encoding processes occur on the circuits. Less difficulty in encoding can be achieved using this type as the parity matrix is sparse (most values are zeros) as they accept multiple sizes of lifting and support different rates. By doing this, it can change between a variety of information lengths [25] [26].

3.1.1 Parity-check matrix

The parity-check matrix is the main component in LDPC. It’s actually part of the name itself. LDPCs are always known by a parity matrix that is sparse, meaning most of them are zeros in their rows and columns. The code has a rate that has its lowest limit defined by equation 3.2.

$$R \geq 1 - \frac{Wc}{Wr} \tag{3.2}$$

when R is the rate of code and Wc, Wr stands for weight of column and weight of row, respectively.

$$\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}$$

Figure 3.5 : Parity-check matrix (20,3,4) size [23].

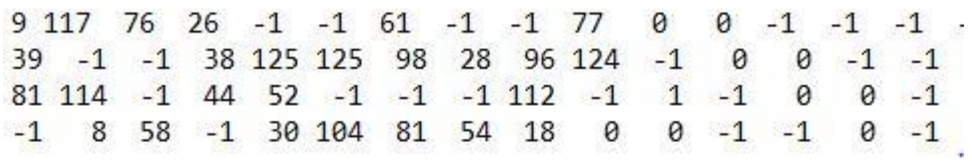
The parity-check matrix size is defined by (n-k)x n where k is message and n is the code word [4].

3.1.2 Encoding of LDPC codes

As the LDPC parity matrix is structured in a double diagonal way, the encoding part of the operation will be achieved with little difficulty; in LDPC, the encoding operation is systematic and some bits are set to be punctured by means of which they do not even enter the buffer [3].

When it comes to complexity in decoding, these codes can be divided into two categories: the randomly created LDPC and the structured type, which has been skilfully sketched; the second type has less complexity in encoding due to the nature of Quaci-cyclic codes.

The experiments on a variety of parities to make sure a less difficult encoding is obtained continued; a researcher named Huang Etal used a compensation matrix procedure combined with double diagonal structure; by doing this step, it’s feasible to minimize coding complexity within a range close to 21%; a small length of code words has been suggested for internet of things programs, and they had some advantages, like no need for iterations, but the main problem was that this type was able only to detect two bits in error and only correct one bit. [27].



```
 9 117  76  26  -1  -1  61  -1  -1  77   0   0  -1  -1  -1
39  -1  -1  38 125 125  98  28  96 124  -1   0   0  -1  -1
81 114  -1  44  52  -1  -1  -1 112  -1   1  -1   0   0  -1
-1   8  58  -1  30 104  81  54  18   0   0  -1  -1   0  -1
```

Figure 3.6 : An extracted part of BG2 with an expansion factor 128.

Encoding can be done in two steps for LDPC. The first is building the parity and, as shown in figure 3.6 in 5G, the parity matrix is specified in a standard. The second is to create a code word from this parity. This process of encoding will be shown in the results section.

3.1.3 Decoding of LDPC codes

When Callager invented LDPC codes for the decoding part, he took into consideration two types of decoding techniques: one is based on hard decision decoding, which is called the "bit flipping algorithm," and the other one is a soft decision method, which is known as the "sum product algorithm." Both of these techniques are under the decoding scheme of message passing. This type uses iterations, as in each iteration, messages are thrown between bit nodes and check nodes. Bit flipping works with a hard decision by means of which, on every received bit, a decision is made whether this bit is going to be zero or one. The decoder, after checking all the parities, will directly change any value that has more than a previously set number. This could be considered as some kind of threshold, because as the value is above that threshold, an action (bit flipping) happens. Each bit node first attaches to a bit received from the channel and sends it to the check nodes to preview the value. The check nodes then do the math to see if the parity is satisfied or not. If everything works properly, the algorithm ends. If not, the bit nodes "flip" the current bit. Otherwise, the value is kept when it reaches the last predetermined iteration number.

Let’s consider this example with this parity in the figure 3.7 with received vector “101011” and code word “001011”.

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

Figure 3.7 : Parity for hard decision decoding [23].

Multiplying the received vector with the parity, we obtain a vector of "1010", which means there is a problem in the first and third bit.

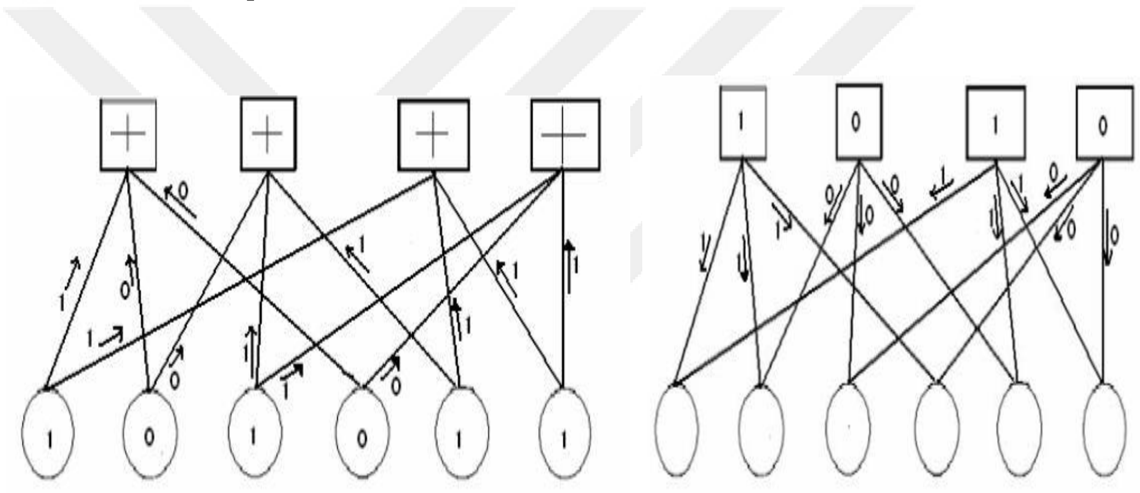


Figure 3.8 : Message passing between bit and check nodes and parity-check qualification[23].

As shown in figure 3.8, the first bit x1 is not satisfied when both of its received values are 1. This means the bit directly flips its value. Multiplying the received vector with the parity, we obtain a vector of "1010", which means there is a problem in the first and third bit.

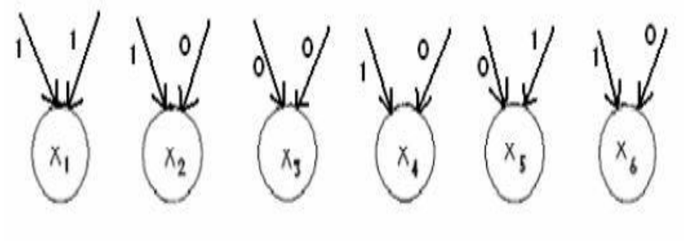


Figure 3.9 : Check node replies to bit nodes.

All the operations will start again until all the bit nodes are qualified, which means every bit will receive zero from both sides, or until the predetermined iteration number is reached. Though this technique seems easy to implement, it has a huge drawback. If the received vector is "101001", when parity is updated, it shows that the first and second bit will not be qualified again, so the more iterations go by, one of them will always be considered wrong.

A more efficient hard decision method is the weighted bit flipping type, specifically RRWPF.

$$V_{i\,j} = \beta \frac{|r_j|}{|r_{i\,max}|} \tag{3.3}$$

Where β is normalised factor, $|r_j|$ is received signal magnitude, and $|r_{i\,max}|$is the highest soft magnitude of all bit nodes.

This method performs better with a higher weight column, as seen in Fig 3.8. The reason for this result is that this method uses calculations depending on the weight of the outcomes, and for this kind of LDPC, when the weight of the column increases, the weight of the rows follows, so you can say that RRWPF obtains better results over an AWGN channel because the higher the weight of the column gets [22].

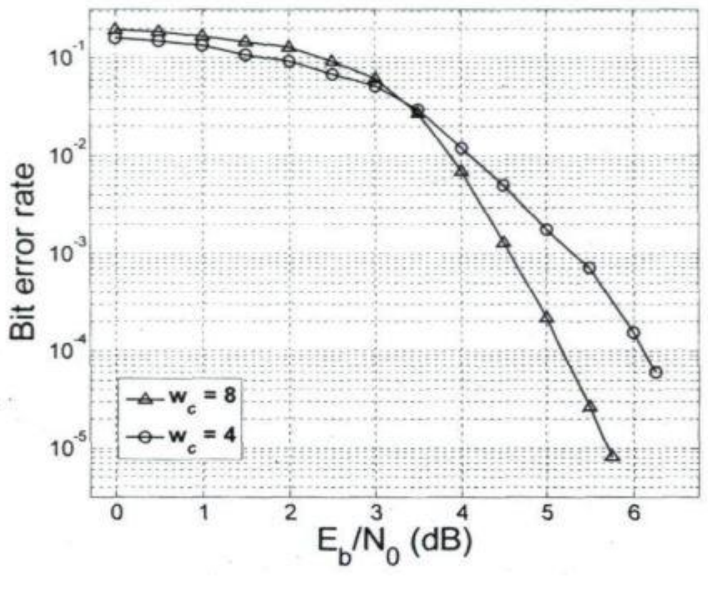


Figure 3.10 : BER performance, two different weight columns, rate ½, code length 400, 50 iterations

Even though the performance is better than the direct bit flipping algorithm, the complexity of this is higher at 50 iterations, which makes the overall decoding more complex, which is why LDPC decoding is more towards soft decision techniques these days.

Soft decision is the most commonly used technique for LDPC codes these days, as the hard decision method is rarely used any more. A better BER performance by far was obtained using soft decision methods. Built around belief propagation, or what can be called a sum product, LLR is used along with min-sum approximation in an iterative way. To understand this concept, an explanation needs to be provided both for repetition codes and for the single parity check codes. The soft decision varies at implementation with different types of channels and modulation schemes, so the flow provided here is related to the actual thesis implementation.

As operations happen in an iterative way between rows and columns, "bit nodes" are doing some kind of repetition code with soft input and output, while "check nodes" are doing single parity check on soft input and soft output operations. These operations have something called intrinsic and extrinsic values.

Let us assume a repetition code with a 1-bit input and three repeated outputs.

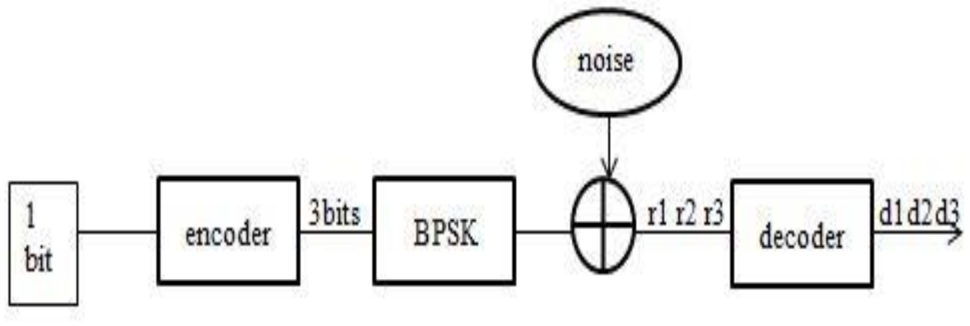


Figure 3.11 : Repetition code example

In this figure 3.11, we will not make the hard decision. We are here interested in some kind of belief. How much do I trust that this bit is 0 or 1 The concept comes from the LLR ratio; with the use of the Bayes rule, we can obtain this equation.

$$\frac{pr(c1 = 0|r1)}{pr(c1 = 1|r1)} = \frac{f(c1 = 0|r1)}{f(c1 = 0|r1)} \tag{3.4}$$

As pr stands for probability f for pdf from Bayes rule as the model used for this thesis is AWGN we can reform this equation as following

$$\frac{\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(r1-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(r1+1)^2}{2\sigma^2}}} = e^{\frac{2r1}{\sigma^2}} \tag{3.5}$$

Taking log of the previous equation, it will result to the following equation:

$$di = \frac{2}{\sigma^2}\, ri \tag{3.6}$$

The value of (i) varies with the given input bit and stands for channel noise variance. This is called the channel LLR or the intrinsic belief. The output of this for the given 3 bits provided in the previous example can be written as follows:

$$Di = (r1 + r2 + r3)\, \frac{2}{\sigma^2} \tag{3.7}$$

The intrinsic and extrinsic values change depending on the chosen value. For example, for d1, r1 is the intrinsic value and r2, r3 are the extrinsic parts.

The reason for mentioning this type of decoder "repetition code" on the LDPC decoder is that it plays an important role in the final decoder.

Moving to the single parity check code, as this part is involved directly with row operations, we have a generator matrix inside of it with a parity part that consists of all ones, so we know all the bits XOR to 0.

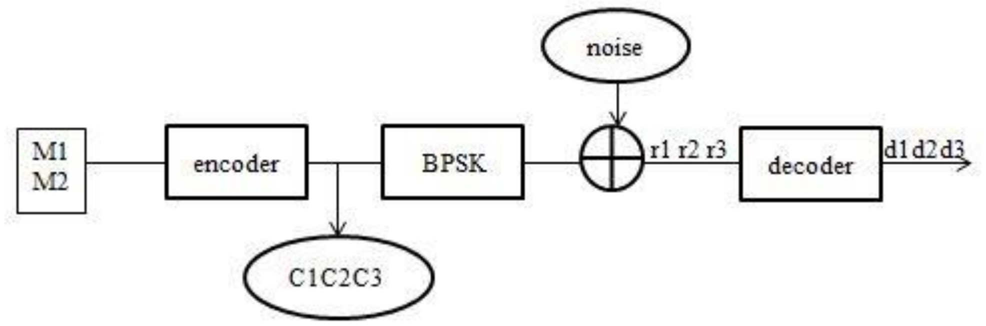


Figure 3.12: (3,2)single parity check code.

So, to obtain the value of d1, we need both intrinsic and extrinsic information. The intrinsic part can be directly taken from r1 as it relates directly to d1, but what about the extrinsic information? Here comes the parity check part. As we know from parity behavior, c1 is equal to c2 XOR c3. Therefore, what we can do here is take intrinsic information about c2 and c3 from r2 and r3, respectively. Use both of them to calculate c1, and from that we get extrinsic information about r1. Adding both intrinsic and extrinsic, we obtain the needed d1.

To simplify the equations we will replace $\frac{pr(c1=0|r1)}{pr(c1=1|r1)}$ with $\frac{p1}{1-p1}$ it's the same is the The probability is only 0 or 1, the same for others. So just the flow of numbers will change p2, p3,... pn, so in order to go from the second and third to the extrinsic information of p1, we need the following

$$\frac{p1-(1-p1)}{p1+(1-p1)} = \frac{p2-(1-p2)}{p2+(1-p2)} + \frac{p3-(1-p3)}{p3+(1-p3)} \tag{3.8}$$

With the help of some algebraic modifications, we can rewrite the previous formula into this new form.

$$\frac{1-\frac{1-p1}{p1}}{1+\frac{1-p1}{p1}} = \frac{1-\frac{1-p2}{p2}}{1+\frac{1-p2}{p2}} + \frac{1-\frac{1-p3}{p3}}{1+\frac{1-p3}{p3}} \tag{3.9}$$

Taking the log of the previous formula and using the tan hyperbolic formula, we end up with the final equation related to the extrinsic part of d1.

$$\tanh\left(\frac{d1extrinsic}{2}\right) = \tanh\left(\frac{d2\ intrinsic}{2}\right) \times \tanh\left(\frac{d3\ intrinsic}{2}\right) \tag{3.10}$$

So the final value of D1 is achieved with the sum of intrinsic and extrinsic values achieved in the previous equation. This tanh can be used and calculated, but as the message size or the block used gets higher, it becomes really difficult to calculate an efficient almost identical to tanh calculation. This is achieved using a min-sum approximation. This approximation is widely used in LDPC codes these days for the simplicity and efficient results it obtains.

To illustrate the idea of the math given for soft decision, let us see an example of how the overall decoder works on a small part of the LDPC matrix.

$$\left[\begin{array}{c|cccccccccc} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{7} & \mathbf{9} & \mathbf{11} & \mathbf{13} & \mathbf{17} & \mathbf{19} & \mathbf{21} \\ \mathbf{1} & 1 & & 1 & & 1 & & 1 & & & \\ \mathbf{6} & 1 & 1 & & & & 1 & & & & 1 \\ \mathbf{10} & 1 & & & 1 & & & & 1 & 1 & \end{array}\right]$$

Figure 3.13: An example based on a part of LDPC matrix.

The darker fonts represent rows and columns that operations occur in; the first estimation can be directly done from the channel LLR by the equation specified in (3.6); then from row 1,6,10 we can get these equations.

$$row1 \quad C1 + C5 + C9 + C13 = 0 \tag{3.11}$$

$$row6 \quad C1 + C3 + C11 + C21 = 0 \tag{3.12}$$

$$row9 \quad C1 + C7 + C17 + C19 = 0 \tag{3.13}$$

The previous equations represent the 2nd, 3rd, and 4th estimations, respectively. We obtained the first estimation from the channel itself. The others are from parity checks. So we can see that every parity in this local structure provides more estimates for the particular bit. It’s calculated using a single parity check decoder. From it, we compute the extrinsic values.

The operations mentioned are only for the first bit, but the same concept applies even for the 500 bits. That bit will have a local structure, and from it, you will know what rows and columns that bit is connected to. This local structure only occupies some of the information as it’s not connected to the rest of the code word. In some cases, the code word may be very long and in order to use other values in an efficient way, message passing is used to improve the results in an iterative way.

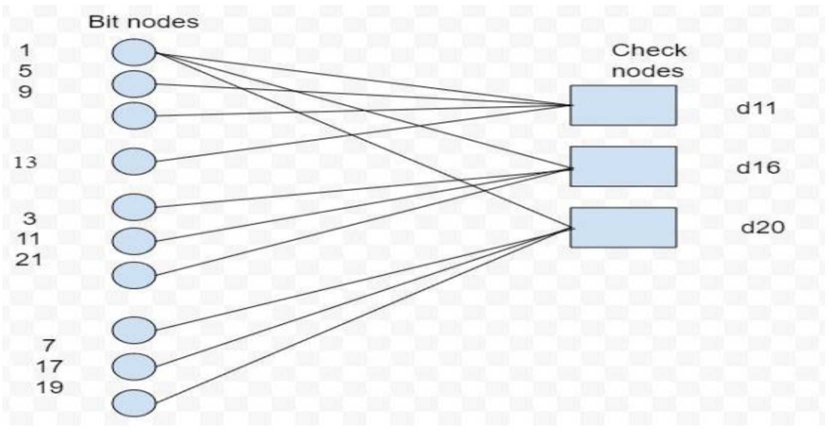


Figure 3.14 : Tanner representation for the matrix.

The local structure can be easily seen in the tanner graph, and the tanner graph captures the message passing through which in that passing the bits exchange the structure without any problems.

First iteration di is sent from the bit node to the check node to all the bits, so now instead of only speaking about the first bit, we speak about all the bit nodes as they will be basing LLR on all the check nodes that they are directly connected with, and all check nodes will operate with a single parity check code.

So after the first iteration, you can say that the channel received values, which come from bit node LLR calculations, ($\frac{2}{\sigma^2}\ ri$) are done. These values are passed to all the check nodes on the tanner graph. After that, the check nodes do single parity check calculations and send the extrinsic on each of the connected edges back to the bit nodes. By doing this, we obtain d11, d16, and d20.

To summarize the situation, row iterations are for check node computations. It computes the LLR for 1 bit, while column iterations are for bit node computations.

This is mainly what happens in a single iteration of the decoder, and this type of decoding is what this thesis LDPC decoder is based on.

3.1.4 LDPC and 5G

After LDPC obtained better capacity performance and a lower decoding complexity, LDPC became the top candidate with the fact that LDPC had already won against turbo codes in other things like URLLC or device-to-device computation; the advancement of the use of flexible block length made them more suitable for 5G. As for the standardization of LDPC codes for 5G, a high data rate is needed, up to 10 GPBS, and with the bits added by encoding, this step will add effects to the speed, so from this situation, the idea of puncturing to make the LDPC code word length less [27].

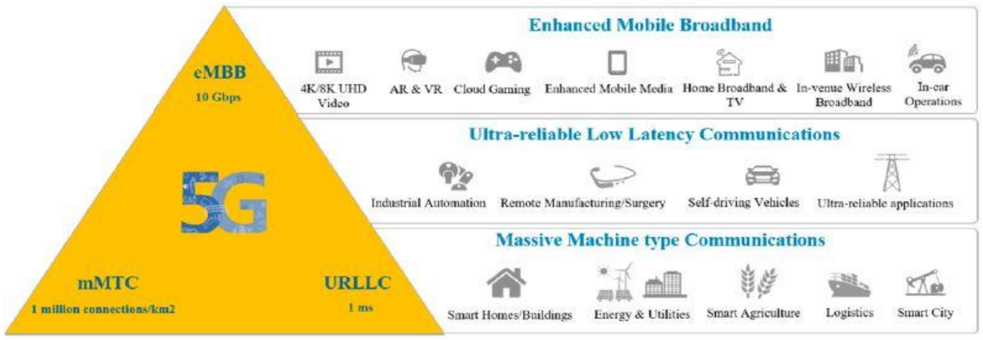


Figure 3.15: Fundamental use range for 5G [10].

Previous standards used some kind of re-transmission procedure to ensure a smaller delay, but this process also requires extra bits, which has a direct impact on 5G technology because the need for a high data rate is the focus.With the help of some techniques, only parts of damaged bits are re-transmitted, but this step adds more to the cost of the process. Scientists have used different parity check matrices to make sure that the process of encoding and decoding has a lower complexity, or we can say less difficulty.

When it comes to practical implementation, LDPC has some creation limits. That's why, in both encoding and decoding, QC-LDPC is used, as these codes are made up of budged identity matrixes [28].

So QC-LDPC has been finalized as the standard for 5G and has been acknowledged as the channel coding outline for the 5G eMBB, which is the scenario proposed for this thesis.

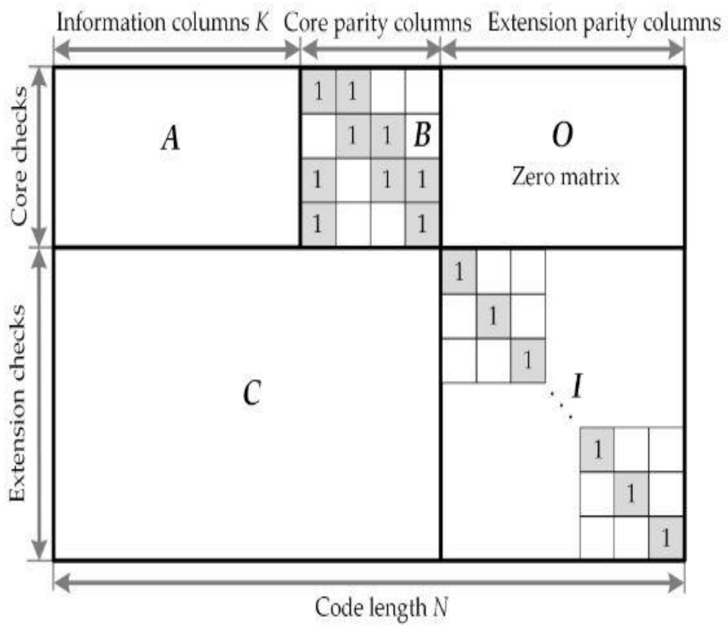


Figure 3.16 : 5G-NR QC-LDPC structure of base matrix [25].

As seen from figure 3.16, from the inside, it’s divided into 5 parts. The I section represents an all identity matrix, O is a matrix full of zeros. This all zeros matrix powerfully backs the "IR-HARQ." The B part is where the double diagonal structure of the matrix occurs, as this structure will help in the process of encoding bits. A corresponds to the systematic bits, and finally, C is for single parity check rows.

The upper left part that gathers the combination of A and B is known as the "kernel," while the other parts are known as the "extension.".

There are two base graphs, "BG1" and "BG2". BG1 is used for great data rates and for extended block lengths. The rates for these graphs are between 1/3 and 8/9 It comes with an original size of 46x68 and comes with 22 message columns.

The highest lifting factor is 384. This means the most information that can be obtained using this graph is 8448 bits. On the other hand, BG2 is used for smaller block lengths and smaller rates; the rates are between 1/5 and 2/3. It comes with a size of 42x52 with 10 message bits. This means that after maximum lifting expansion is applied, the biggest message size is 3840 bits [29].

For the design of parity checks, as for BG1, kb is 22. For BG2, if the block size is bigger than 640, then kb is 10, if the block size is bigger than 560, kb is 9, and if greater than 192, kb is equal to 8. Otherwise, kb is equal to 6. This is used for parity designing purposes, as K is set at 22Zc for BG1 and k is 10Zc for BG2 as Zc is the lifting factor [30].

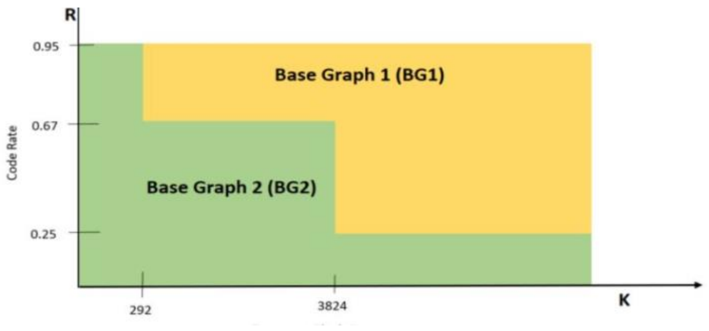


Figure 3.17 : Base graph selection [31].

As for every base graph a different lifting factor the parity for each is obtained from the base graphs according to the following all -1 values are replaced by zeros matrix. the 0 values are replaced with an identity matrix ,and other values are replaced with identity matrix with column shifted to right as an example if value is 3 the identity is shifted to the right 3 times. All of these matrices are size of Zc x Zc by means we replace the numbers with matrices size according to the needed lifting number.

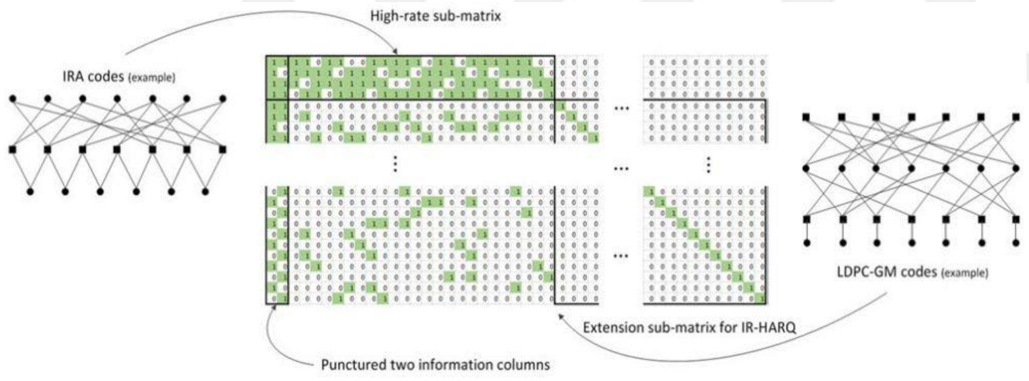

High-rate sub-matrix
IRA codes (example)
LDPC-GM codes (example)
Extension sub-matrix for IR-HARQ
Punctured two information columns

Figure 3.18 : 5G LDPC BG1[32].

The final thing that needs to be mentioned related to LDPC with 5G technology is rate matching. Let us start with base graph 1, as we have a parity of 46x68 with the first 22 containing the message. Therefore, when fixing the expansion factor (z) and the parity matrix, you end up with 22/66. Since in the standard the first two bits are always punctured, this leads to a default rate of 1/3, but what if we need another rate? For example, at a rate of 1/2 In order to achieve this rate, the first 44z bits need to be transmitted, so 24 parity bits are needed. This means, in a decoder, only the 24x46 top-left part of the matrix is used.

Assuming another rate is needed for base graph 1 matrix, let us say a 2/3 rate, the same process is applied. 33z are transmitted, 13z of which is parity, so for the decoder part, the top left part of the parity matrix is used.

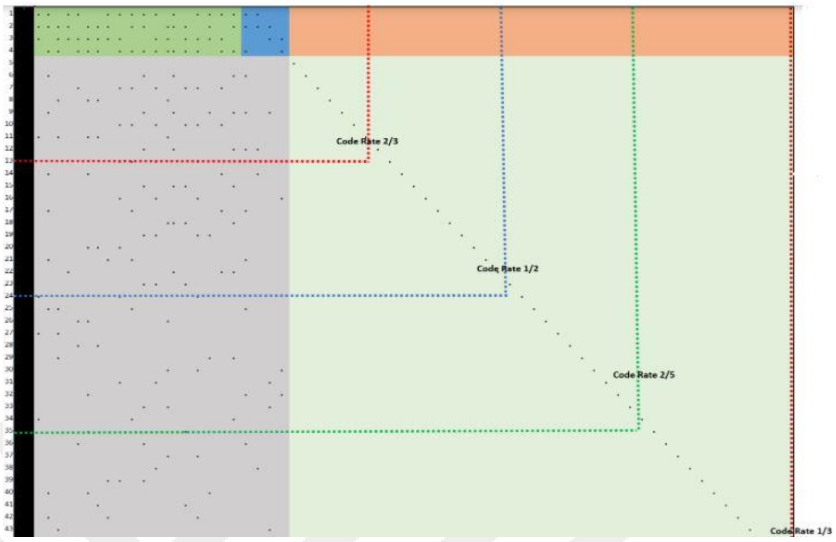


Figure 3.19: Rate matching for base graph 1 at different rates [9].

For base graph 2, the base matrix size is 42x52 with a 10z message size, and with the puncturing occurring, this leads us to end up with a 1/5 rate. So, if a rate of 1/2 is required, 20z must be transmitted, 12 of which are parity, so the base matrix used here will be 12x22 in size.

We can use parts of the base matrix in order to obtain the needed rate, but there are some considerations to take. For example, we can’t get a lower rate than 1/3 for base graph 1 as the whole parity is used to obtain that number. Same for base graph 2, as 1/5 is the lowest rate we can achieve, the whole parity is used to get that number.

3.2 Polar Codes

In 2008 a scientist named “Erdal Arikan” invented the polar codes the codes had an excellent BER performance the codes has some sort of sequential encoding and decoding by means you end up with two type of channels an indeed good or an indeed wicked channel.

With their unique construction and with their list decoding at short block-lengths as BER gain is better than other competitors like turbo codes, with this improvement polar codes were the best nomine for the control channels as the consignment is kind of small [33].

So polar codes have been selected for the 5G standard for the control channels, as in both ways for uplink and downlink channels; when constructing the polar code, it includes how much a specific channel is reliable or trustable for transmission among the other channels, so each bit can be encoded according to that.

As 5G comes with different information and rate sizes, numerous states for the channel, and for those to have several reliability structures for every combination is impractical. The research community did not focus on the aspect of 5G in polar codes; most of their exploration was related to their BER or the scheme of encoder and decoder; so for a joint service providing a manufacturing standard has been agreed upon between the main companies, so that the hardware items will be companionable with each other [34].

To encapsulate the construction step, channels are added together to form a channel vector, and after that, they are divided into computer-generated channels. Here is where the polarization takes place. The bigger the block lengths get, the channels will end up with only two situations: a flawless channel or an impractical channel. This is where the term polarization came from. Like an actual pole, whether you go north or south, there is nothing in the middle [4].

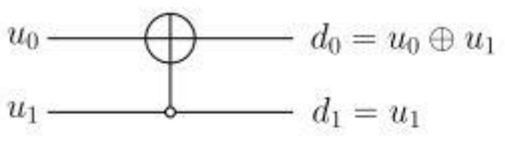


Figure 3.20 : Polarization kernel [34].

So, assuming a polar code of length N at a rate of R,their multiplication is the information bits, but before continuing the encoding process, N-K values will be frozen as they will be the least reliable channels. They are called the "frozen bits." They are considered to be zero. Also, in this thesis scheme, the frozen bits are considered zeros. So a length N encoder output, let’s call it Y, can be written as in equation 3.14.

$$Y = U\ GN \tag{3.14}$$

$$GN = (G2)^{\otimes n} \tag{3.15}$$

$$G2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.16}$$

Where G2 is known as transform kernel, $(G2)^{\otimes n}$ is the kronecker product of the nth value and the value of n is obtained by taking the 2nd log of N.

To illustrate the idea of a kronecker product, here is an example of how it's implemented in polar codes. Let us assume G4, which is a 4bit to 4bit transform, so we have the following:

$$G4 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.17}$$

So what happens here when doing a kronecker product? Each value from the first matrix is replaced by its multiplication with the second matrix, so we end up with the below matrix.

$$G4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.18}$$

As in 5G, we can have up to G1024 as the highest n value in the standard is 10. By taking the 2nd log of 1024, we get the maximum value of n.

When it comes to overall scheme complexity, it is lower compared to LDPC, but SC has a huge delinquent when it comes to correction ability, which is precisely small, so in order to figure out a solution to this dilemma, lists were introduced to polar codes, which is now known as SCL; decoding lists means the higher the list number is, the higher the complexity of decoding gets; in fact, when list = 32, it can be compared in complexity to LDPC codes.[6].

3.2.1 Encoding of polar codes

The encoding of polar codes, like other types of codes, can be done with different methods. The two main types of polar code encoding methods are systematic and non-systematic encoding.

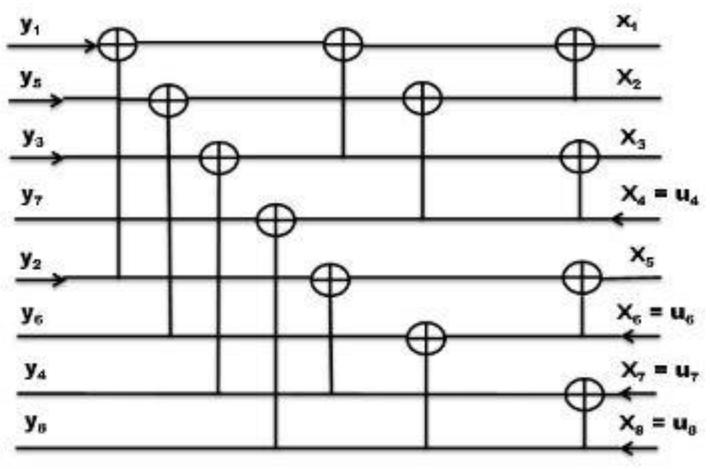


Figure 3.21 : Systematic polar encoder N=8 [35].

Mainly because of the random numbering order of the polarized channels in quality, the non-systematic is mainly applied, as the decoding scheme in 5G also means the encoding process of polar codes is non-systematic.

The process of encoding a non-systematic is through this order: first is the value of N that is used for the process. The value of N is equal to, as in 5G, the value of n varies from 1 to 10, so the maximum N is 1024; then, based on the message size K, the positions of N-K are frozen. This means they are set to zero. The places where the values are frozen depend on the value of N. It's defined by something called the reliability sequence. The 5G has its own version of it based on the value of N and K. The frozen bits are decided directly based on the standard.

The values left after N-K are usually referred to as "u," so the encoding finally happens with this formula.

$$codeword = u.GN \tag{3.19}$$

To illustrate the idea of polar encoding, let's take this example of a small (8, 4) polar code according to the 5G standard [34]. The 1/2/3/5 bits are frozen. A message will be transmitted on the 4/6/7/8 channel.

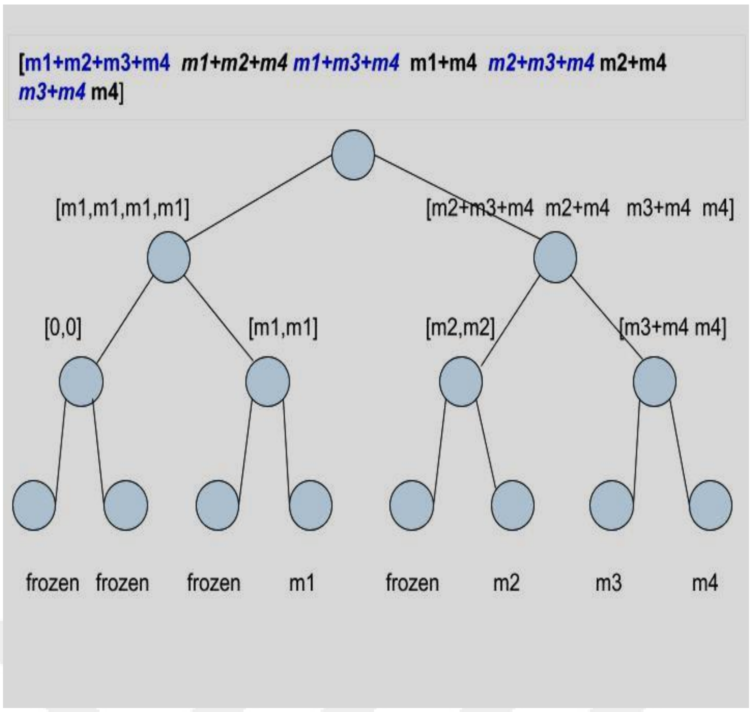


Figure 3.22 : Binary tree representation of (8,4) polar encoding.

As seen from the figure 3.22 the inputs were [m1, m2, m3, m4].

The output in this scenario the code word is [m1+m2+m3+m4 , m1+m2+m4, m1+m3+m4, m1+m4 m2+m3+m4, m2+m4, m3+m4, m4].

The plus sign between message bits is a XOR operation between them, and as seen from the figure, the frozen bits are not in order. That is why it's known as non-systematic encoding.

So for the previous graph, if message bits are [1 1 0 1] the code word for this message is [1 1 0 0 0 0 1 1].

3.2.2 Decoding of polar codes

Unlike the encoding of polar codes, which follows an un-systematic scheme in the 5G scenario, for decoding, there are multiple ways to do that. SC and SCL are the main techniques considered for polar codes.

For SC de-coding, polar codes were particularly created. Channel capacity is achieved by using coding with an endless code length. No matter what other decoding methods there are out there, Because of the weak error correction performance of Belief Propagation and SCAN, the 3GPP decided to put the emphasis on a SC-based decoder on the receiver side of the system. Probabilities, which are numerically unstable and unsuitable for hardware implementation,were used to Soft information characterized the original formulation of SC. At first, LLR was used to lessen the instability, then log-likelihood ratios were used to eradicate it completely. The SC method is simple to implement as in software, but most importantly, it's applicable to hardware [34].

When employing the SC method, the first bit gets estimated and then, using that estimate as well as the received vector, the second bit is calculated in bitwise fashion. The very first two predictions and the received vector are used to conclude an estimate for the following bit in the same way. The procedure is repeated until all the bits on the receiving end have been decoded completely.

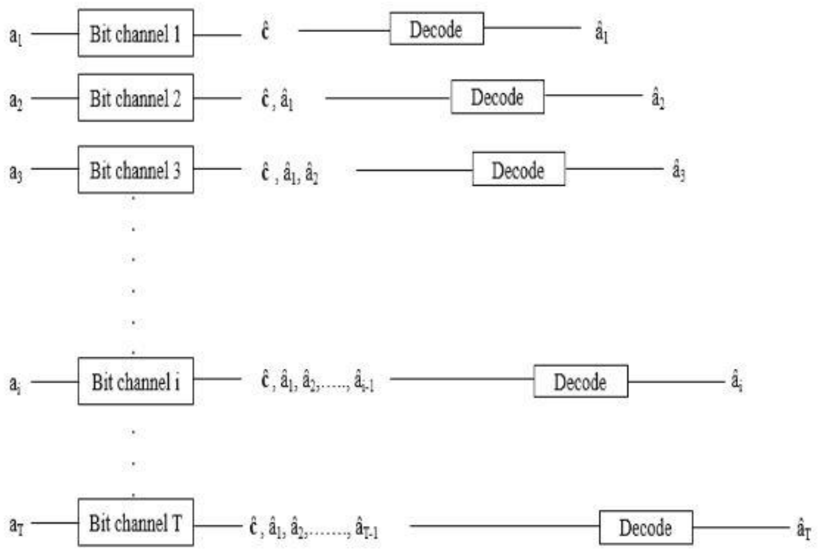


Figure 3.23: Sequential decode process [36].

As seen in figure 3.23, there is a received vector alongside the previous input from each of the prior channels at the receiver. This needs to be decoded through the SC to come up with an estimation for the bit.

Our major goal is to recover the given vector's actual source bits. The obtained bits now serve as the code word's soft information. Since the combined outcomes of code words will take us to our initial source bit, obtaining the merged soft information is our goal.

An SPC is used to estimate the initial bit. As a result of this extra information, the min-sum approach can make more accurate predictions of how many message bits are left. The next estimate can be directly obtained using the repetition code, as if the previous bit is 0, then the "soft information" is added, and if it's 1, the estimate is achieved through the difference of those bits[36] .

As with the LDPC code, instead of using the "tanh", we are directly using min-sum approximation as it is simpler to use and provides almost the same soft outputs as applying the complex "tanh" calculations.

So, according to the previous figure, with a two-bit system, we can represent the SPC or the min-sum part with the following equation:

$$L\,(\hat{a}1) \;=\; sgn\,(\hat{c}1).sgn\,(\hat{c}2).min\,(\,|\hat{c}1\,|,|\hat{c}2\,|) \qquad (3.20)$$

"sgn" stands for the sign of the bit and "min" for the lowest value between the two variables. After obtaining the result, as the modulation scheme is BPSK, where 0 goes to 1 and 1 goes to –1, a hard threshold is set based on this modulation.

$$L(\,a_1\,) \geq\; 0, \hat{a} \;=\; 0 \qquad (3.21)$$

$$L(\,a_1\,) <\; 0, \hat{a} \;=\; 1 \qquad (3.22)$$

The second estimation of the repetition code can be represented with the following equations.

$$if\ a_1 = 0, L(\,a_2\,) = \;(\hat{c}2\;+\;\hat{c}1\,) \qquad (3.23)$$

$$if\ a_1 = 1, L(\,a_2\,) = \;(\hat{c}2 -\;\hat{c}1\,) \qquad (3.24)$$

The equations above are for 2 bits, but this method may also be used for bigger source bits. This approach is unique in that it is simple and straightforward to use.

On the other hand, with a modest or average number of block lengths, SC decoding is unable to provide improved achievement outcomes. Successive Cancellation List Decoding is employed to prevent this issue. In a SC decoder, just one route from the decoding pathways is activated. However, in a SCL decoder, a number of optimal decoding routes is activated. This L can vary; 1 or 2 could be used, but usually a bigger value of 4 or 8 is implemented. The bigger the list size, the more accurate decoding is achieved, but at the cost of higher decoding complexity.

The decoding of a list is done in a different way. Back in the SC decoding, one code word is generated from the received vector.

Because CRCs have excellent error detection features, they are utilized in list decoders.

As a result, at the physical layer, every message will have a CRC attached to it. The higher levels will examine the decoded code word from the physical layer to see whether the CRC was fulfilled. If it is not, the higher layers will detect the problem and request a re-transmission of the message. Decoding lists make use of the CRC. We check for it on a physical layer of generally 4 or 8 lists, and if any of them passes, then it's selected as a valid code word.

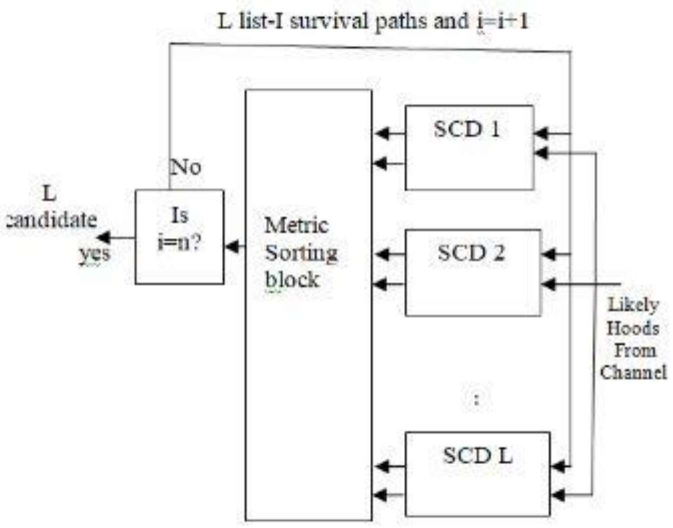


Figure 3.24 : SCL decoder [35].

The value of CRC varies, and it is specified in the standard depending on many different aspects. It will be described in more detail in polar codes and the 5G subsection.

So, how can we use the decoder to generate several code words? It is straightforward in SC polar codes. You are doing sequential decoding. You begin by deciding on the first piece of the code phrase that you get. There is just one thing to do if it is a frozen direct zero. Once a message is received, a choice is made depending on whether belief 0 or 1 is present. Even if the decoder is able to correct the mistake, the final code word will be incorrect. Therefore, sequential decoding, on the other hand, prevents you from going back and fixing mistakes.

Instead of merely having one route available in the decoder, SCL allows the user to make a decision, and then the decoder advances into two pathways after the decision points to either zero or one, providing a decision metric for each option made. The decision can be made in accordance with or in opposition to the belief.

If the decision is the same as the belief, then nothing happens. However, if the decision is contrary to the belief, which is possible in that scenario, for example, if the bit is frozen and you receive a value for it after transmission, then you are making a decision that is contrary to the actual belief. You will have to pay a price for your choice. The decision metric now includes the bit's absolute value added to it.

It is possible to think of route metrics as the total of all the decision metrics along a given path. So it is a confidence in the candidate code word, and if none of the candidates passes the check after verifying the CRC of the message, the route with the lowest path metric is selected. If there are several message candidates, every one of them passes the CRC. Once again, the lowest path metric was picked.

An important thing that needs to be mentioned is that the selected list size is considered fixed. It can not be changed during the process. For example, if the list size is four and after certain operations in the decoder, the split reaches eight, what will happen is that the four lists with the highest path metric are directly discarded.

When compared to SC, the SCL decoder improves the efficiency of polar codes. However, the complexity of its decoding increases with list size L.

3.2.3 Polar codes and 5G

The same as with LDPC codes, 5G polar codes have a specific standardad attached to them. When comparing polar codes to 4G LTE TBCCs, turbo codes, and also to NR LDPC codes, the major benefit of polar codes has been that polar codes with SCL and CRC outer codes commonly generate better results at reasonable data packet sizes on the order of K = 250 b or even less, which is adequate for classic control information. In comparison to the previously mentioned codes, the polar code employed in 5G NR is substantially more complicated [37].

In 5G scenarios, the number of bits of information A, which equals the sum of message bits and CRC at SCL, is fixed, and a code word of length E is constructed to attain the desired rate R = A/E demanded by higher communication layers. The The length of a mother polar code is first created to meet this condition, then the targeted code length E is matched by puncturing, shortening, or repeating of the mother polar code as required. It's important to note that the lower bound is set to 32 by default, whereas for uplink and downlink, the maximum N is set to 1024,512 and 11 respectively [33].

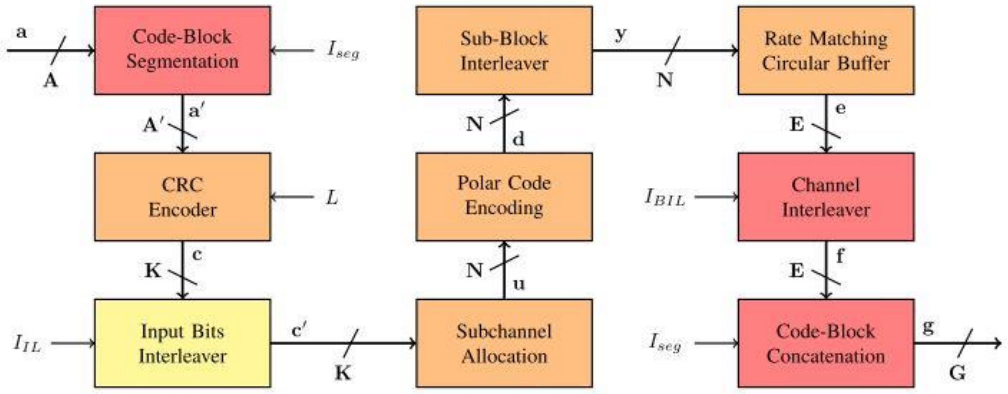


Figure 3.25 : Polar codes encoding sequence for 5G [34].

In the previous figure, yellow stands for the downlink channel, red for the uplink channel, and orange for both.

Message segmentation takes into account a message (A) payload (G). The information may be divided into two blocks and encoded separately. For uplink segmentation, it is essential when message or payload lengths surpass the mother polar code length. In this scenario, shortening or puncturing is used as a rate-matching tool; downlink segmentation is not required for the reason that message sizes are indeed shorter than mother polar code lengths and repetition is used for big payload sizes. As a result of these changes,segmentation will be turned on. "$(A \geq 1013) \vee (A \geq 360 \wedge G \geq 1088)$"[3]. In this instance, the message is broken into 2 pieces based on its length.

Another thing is that rate matching is employed here, not as a big implementation as LDPC codes but also used here; the rate matcher's purpose is to modify the size of the code to tie the available resources. To modify the code length, three different methods are used: puncturing, shorting, and repetition.

Puncturing occurs when the mother polar (1024 for uplink, 512 for downlink) is bigger than the rate-matched output E, also the message plus CRC divided by the selected E is less than or equal to $\frac{7}{16}$ This number acts as a threshold to differentiate between whenever puncturing or shortening is needed. So here, puncturing occurs and the N-E bits are not sent.

$$\frac{k + nCRC}{E} \leq \frac{7}{16} \tag{3.25}$$

In the previous equation stands for number of CRC bits E is the length of the code word.

Shortening case is close to puncturing it happens when the mother polar is bigger than the rate matched output but here the message plus CRC divided by the selected E is bigger than $\frac{7}{16}$, so here shortening occurs .

$$\frac{k + nCRC}{E} > \frac{7}{16} \tag{3.26}$$

It's the same as the concept of puncturing. The major difference between them is that puncturing occurs at the beginning of the encoder. On the other hand, shortening happens at the end [32].

The final thing used for the rate matching is the repetition. The situation happens when the code word or the rate-matched output is greater than the mother polar value, so collecting E consecutive bits from the circular buffer, emerging from the very first location, and wrapping around.

All the previously mentioned rate-matching techniques are executed with an N-size sub-block interleaver in addition to a buffer.

The sub-block interleaver's job is to organize incoming coded bits in the circular buffer such that they are rejected or accepted in the correct sequence. It only permutes the five most important bits of the binary demonstration for polar encoded bits' index numbers in a 32-bit integer sequence, as defined by the interleaver's design [37].

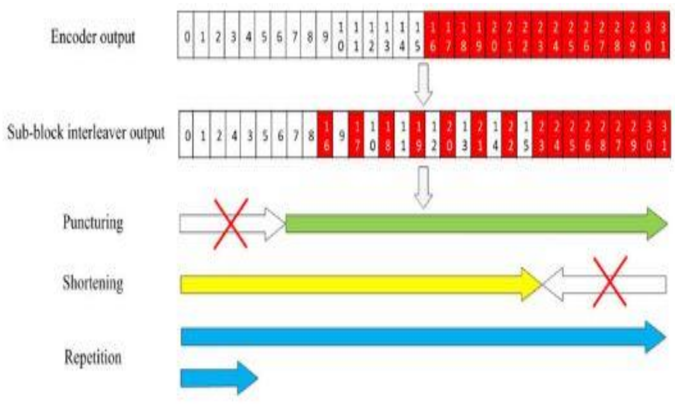


Figure 3.26 : Interleaving besides rate matching process for 5G polar codes [32].

The other thing about 5G polar codes mainly used for SCL, which is the CRC for the 5G standard, is that they are provided by table 3.1.

Table 3.1: CRC codes in 5G[38]

Label	Polynomial
CRC6	$x^6 + x^5 + 1$
CRC11	$x^{11} + x^{10} + x^9 + x^5 + 1$
CRC16	$x^{16} + x^{12} + x^5 + 1$
CRC24A	$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$
CRC24B	$x^{24} + x^{23} + x^6 + x^5 + x + 1$
CRC24C	$x^{24} + x^{23} + x^{21} + x^{20} + x^{17} + x^{15} + x^{13} + x^{12} + x^8 + x^4 + x^2 + x + 1$

So for the uplink control information, different CRC values can be used. For example, when k is between 12 and 19, we can only use a CRC that is equal to 6. On the other hand, when K is greater than 19, a CRC of 11 bits is used. For downlink, a CRC of 24 is used.

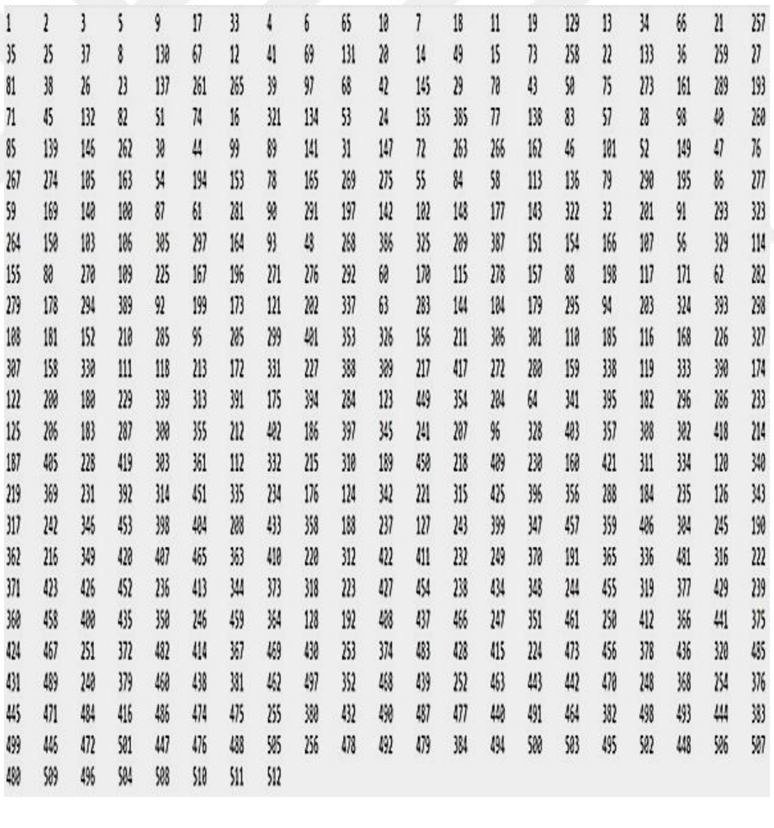
Another thing that needs to be mentioned alongside the CRC is parity bits. Sometimes, when k values are too small, between 12 and 19, Here, three parity check bits are positioned at the entrance of the encoder.

$$k + nCRC + npc \tag{3.27}$$

Therefore, the total message that enters the encoder will be the sum of message bits CRC bits, and parity bits.

For notice, this does not mean the scheme used in this thesis, or generally in 5G polar code standards, is parity-check polar code. These are completely different types of polar codes; they came about because in SCL decoding there is no room for collective growth in CRC bits.

Finally, before finishing this subsection, one of the most important things in 5G polar codes is the reliability sequence.



1	2	3	5	9	17	33	4	6	65	10	7	18	11	19	129	13	34	66	21	257
35	25	37	8	130	67	12	41	69	131	20	14	49	15	73	258	22	133	36	259	27
81	38	26	23	137	261	265	39	97	68	42	145	29	70	43	50	75	273	161	289	193
71	45	132	82	51	74	16	321	134	53	24	135	385	77	138	83	57	28	98	40	260
85	139	146	262	30	44	99	89	141	31	147	72	263	266	162	46	101	52	149	47	76
267	274	105	163	54	194	153	78	165	269	275	55	84	58	113	136	79	290	195	86	277
59	169	140	100	87	61	281	90	291	197	142	102	148	177	143	322	32	201	91	293	323
264	150	103	106	305	297	164	93	48	268	386	325	209	387	151	154	166	107	56	329	114
155	80	270	109	225	167	196	271	276	292	60	170	115	278	157	88	198	117	171	62	282
279	178	294	389	92	199	173	121	202	337	63	283	144	104	179	295	94	203	324	393	298
108	181	152	210	285	95	205	299	401	353	326	156	211	306	301	110	185	116	168	226	327
307	158	330	111	118	213	172	331	227	388	309	217	417	272	280	159	338	119	333	390	174
122	200	180	229	339	313	391	175	394	284	123	449	354	204	64	341	395	182	296	286	233
125	206	183	287	300	355	212	402	186	397	345	241	207	96	328	403	357	308	302	418	214
187	405	228	419	303	361	112	332	215	310	189	450	218	409	230	160	421	311	334	120	340
219	369	231	392	314	451	335	234	176	124	342	221	315	425	396	356	288	184	235	126	343
317	242	346	453	398	404	208	433	358	188	237	127	243	399	347	457	359	406	304	245	190
362	216	349	420	407	465	363	410	220	312	422	411	232	249	370	191	365	336	481	316	222
371	423	426	452	236	413	344	373	318	223	427	454	238	434	348	244	455	319	377	429	239
360	458	400	435	350	246	459	364	128	192	408	437	466	247	351	461	250	412	366	441	375
424	467	251	372	482	414	367	469	430	253	374	483	428	415	224	473	456	378	436	320	485
431	489	240	379	460	438	381	462	497	352	468	439	252	463	443	442	470	248	368	254	376
445	471	484	416	486	474	475	255	380	432	490	487	477	440	491	464	382	498	493	444	383
499	446	472	501	447	476	488	505	256	478	492	479	384	494	500	503	495	502	448	506	507
480	509	496	504	508	510	511	512													

Figure 3.27 : Reliability sequence for N=512.

Integers with values of less than 1024 are ranked by reliability, starting from the least reliable channel all the way up to the most reliable one. In the process of creating a mother of specific length directories, less than N is efficiently removed.

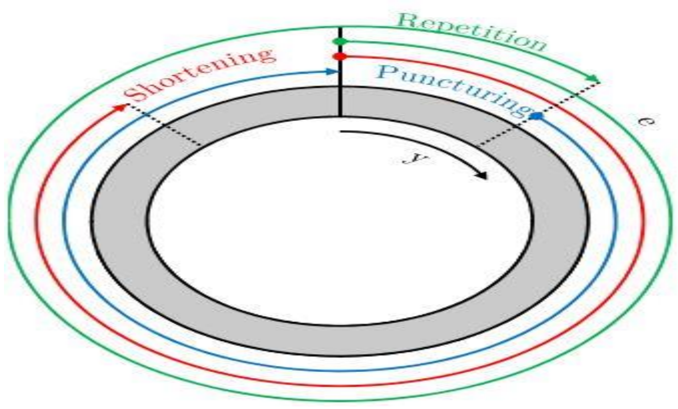


Figure 3.28 : Buffer scheme used for rate-matching [33].

So when the full-length mother polar is not used or there is a usage rate match, the scenarios of puncturing, repetition, and shortening occur. For example, when the mother polar is bigger than the used length code word E and the used rate is less than or equals $\frac{7}{16}$ then puncturing happens, so here the leading bits aren't transmitted.

4. RESULTS AND OUTCOMES

This chapter will discuss the results obtained from both the LDPC and polar codes. All the simulations were performed with MATLAB R2020b, the laptop used for the entire The procedure has the following specifications listed in table 4.1.

Table 4.1 : Laptop specifications

Name	Specs
Processor	Intel(R) Core(TM) i7-5500U CPU
Speed	2.40GHz
RAM	8.0 GB

4.1 BER, $EB/N0$ And Modulation

Before going into results, the scheme of EB/NO, BER, and modulation needs to be explained. SNR, which is signal power divided by noise power, When the SNR value is high, that means the signal level is higher than the noise level. At a low SNR, for example, when the SNR is 1, (0 dB), the signal power equals the noise power, so here it's most likely for an error to occur. For a continuing time system, SNR can be shown as:

$$SNR = \frac{P}{NoW} \tag{4.1}$$

As P is the signal power, with noise power spectral density of NO/2 and bandwidth of 2W, this makes noise power equal NoW. For a discrete time, it's about energy, not power, so here we have:

$$ES = \frac{P}{2W} \tag{4.2}$$

$$noise\ \ energy = \sigma^2 = \frac{NO}{2} \tag{4.3}$$

$$SNR = \frac{ES}{\sigma^2} = \frac{P}{2W} \times \frac{2}{NO} = \frac{P}{2W} \tag{4.4}$$

So, as seen from equation 4.4 equals 4.1, all we need to worry about is the discrete time component as energy of symbol or noise variance.

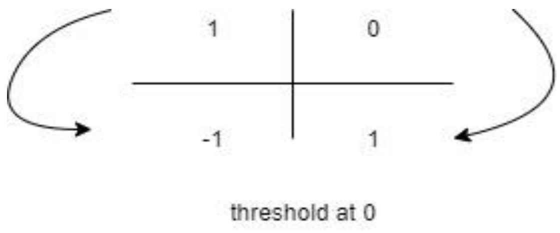


Figure 4.1: BPSK scheme.

For the BPSK moving to the BER section, considering figure 4.1, if the transmitted bit was +1 but the noise was so high that the received was -1, this is called BER and for BPSK it can be calculated with the Q function.

$$SNR = \frac{1}{\sigma^2} \tag{4.5}$$

For the BPSK moving to BER section considering figure 4.1 if the transmitted bit was +1 but the noise was so high that the received was -1 this is called BER and for BPSK it can be calculated with Q function

$$BER = Q\left(\sqrt{SNR}\right) \tag{4.6}$$

$$Q(Z) = \frac{1}{2}\ erfc\left(\frac{Z}{\sqrt{2}}\right) \tag{4.7}$$

Tentatively, these are the parameters for the curve. We do coding instead of sending the modulated signal directly because coding enables us to achieve the same BER levels at a lower SNR, and when we think about this from the concept of a tablet or smart phone, the energy used for transmitting the signal has a cost, which is the drain on the battery. Therefore, with coding, less power is consumed, which means saved battery for other applications.

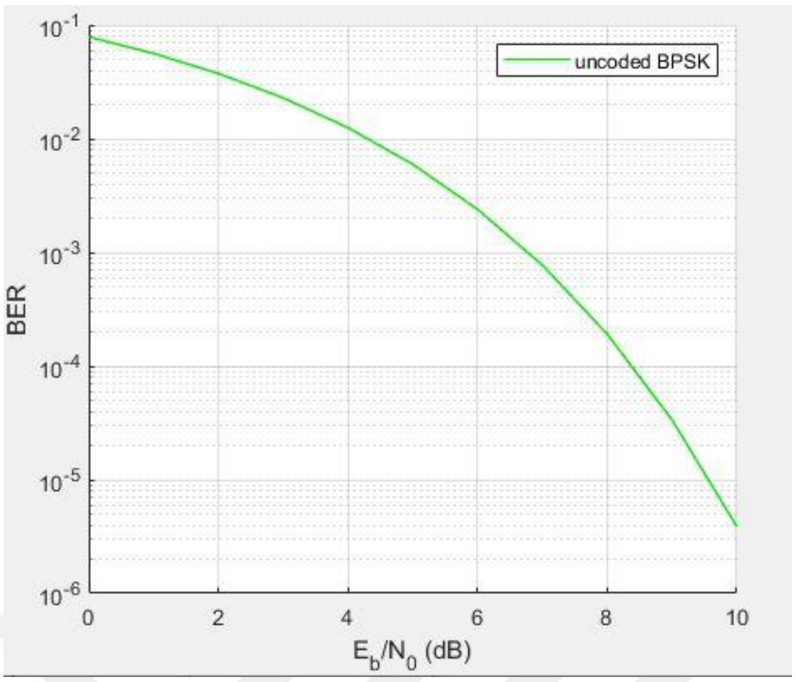


Figure 4.2: Uncoded BSPK.

We have been discussing this with BER and SNR, but the reason for the curve being between BER and EB/NO is that for the uncoded version it has a rate of one. Every symbol carries 1 bit of information after coding, less than 1 bit is carried, so the comparison is not reasonable. That is why most BER comparisons are done with EB/NO.

$$Eb = \frac{ES}{R} \tag{4.8}$$

As Eb is energy per bit and R is the rate used in the operation, this leads us to these two below equations.

$$\frac{Eb}{NO} = \frac{SNR}{2R} \tag{4.9}$$

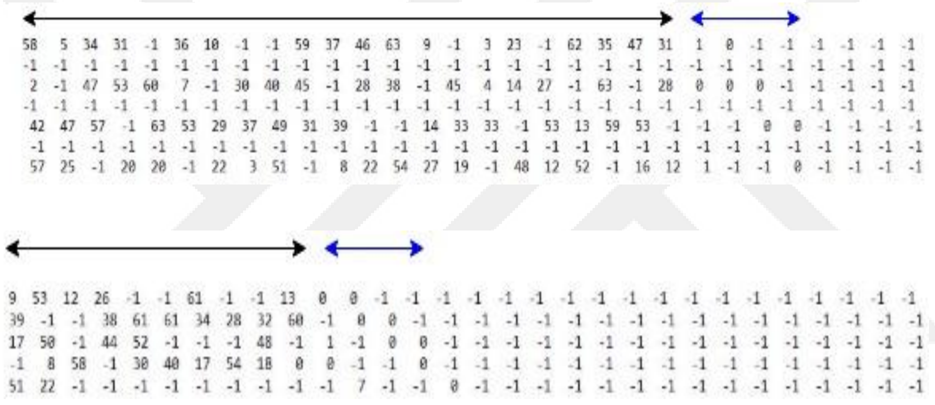
$$BER = \frac{1}{2}\, erfc\left(\sqrt{\frac{Eb}{NO}}\,\right) \tag{4.10}$$

4.2 LDPC Codes Results

This subsection will discuss the results from LDPC codes and their BER performance, and before applying any steps, the first thing to do is base graph selection.

There are 2 base graphs for LDPC codes. BG1 46X68 is primarily used for large transport blocks, whereas BG2 41X52 is used for smaller block sizes [21 The base graph is chosen based on both rate and K size; when K is less than 292 BG2 is chosen; if rate is less than 0.25, BG2 is also chosen. if K is less than or equal to 3824 and the rate is less than or equal to 0.67, BG2 is selected. In every other case, BG1 is chosen. Kb, or what is known as the information column, is set to 22 for BG1 and 10 for BG2.

As both BG1 and BG2 share similarities after the Kb value is reached, the double diagonal structure makes the encoding easier as all we need is from Parity 1 up to fourth parity, as noticed from figure 4.3, from fifth parity in each base graph, all values are zeros.



58	5	34	31	-1	36	10	-1	-1	59	37	46	63	9	-1	3	23	-1	62	35	47	31	1	0	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	47	53	60	7	-1	30	40	45	-1	28	38	-1	45	4	14	27	-1	63	-1	28	0	0	0	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
42	47	57	-1	63	53	29	37	49	31	39	-1	-1	14	33	33	-1	53	13	59	53	-1	-1	-1	0	0	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
57	25	-1	20	20	-1	22	3	51	-1	8	22	54	27	19	-1	48	12	52	-1	16	12	1	-1	-1	0	-1	-1	-1	-1

9	53	12	26	-1	-1	61	-1	-1	13	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
39	-1	-1	38	61	61	34	28	32	60	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
17	50	-1	44	52	-1	-1	-1	48	-1	1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	8	58	-1	30	40	17	54	10	0	0	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
51	22	-1	-1	-1	-1	-1	-1	-1	-1	-1	7	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 4.3 : Double diagonal structure for BG1, BG2.

From the double diagonal structure for encoding, the entire message from the first to the fourth row is XORed together. Because of the structure, parities from second to fourth get disregarded, but only the first parity remains. The first and fourth row first parity components are canceled, leaving us with only second or third parity, and the shift depends on which of them is not-1. step is to use the multiplying factor matrix for parity 1. The inverse of that is the needed parity 1.

The second to fourth parity is then found sequentially from parity 1. Starting from fifth parity, operations are not complex as all values in different base matrices from fifth parity to the end will be 0, or in other words, direct identities.

For decoding soft input decoder used operations inside that contain both row and column operations. In Chapter 3, the general way was explained in practical terms. For the layered decoding, we don’t wait for the entire row to finish to update columns, as columns get updated more violently without the need for the entire row to finish.

The expansion of the LDPC base matrix supports layering as each expanded row acts as a layer. Rows of parity are grouped into layers, and after that layer ends, column operations are applied.

We start with the received vector. We go down the parity row by row. As each row goes by, the received vector keeps updating.

The first thing is initialization as the incoming received vector in the matrix with the note that the place where zeros are placed remains untouchable.

The min-sum operations are applied to each row in the layer after the step vector is updated. This update is the result of addition between layer values after min-sum with the received vector; for the next layer, the updated vector is used as the received vector when all layers are finished, which is considered the 1st iteration.

From the second iteration to the end of the iterations, the arriving beliefs get subtracted from the layer. After applying this step, we have the correct input for the layer. The reason for doing this is that after the first iteration is finished, you will always have an effect from the previous iteration, so that effect gets removed before the start of the new iteration. The same operations as min-sum continue. The vectors are updated.

The same operations of min-sum and updating the vector continue until the desired number of iterations is reached. A decision is made with the final vector after the iteration with threshold values less than zero set to one and values greater than zero set to zero.

4.2.1 LDPC base graph 1, 2 at rate 1/2

As known, base graph 1 matrix size is 46x68 with a 22z message and the default code word length is 68z-2z (always punctured), which is 66z. This makes the default rate for base graph 1 22z/66z or 1/3. For base graph 2, the matrix size is 42x52 with a 10z message and the default code word length is 50z, which makes the default rate for base graph 2 10z/50z or 1/5.

So rate matching is needed in order to transmit half the rate in this case or other rates beside the default rates in other scenarios. For base graph 1 to transmit at half the rate, the first 44z needs to be transmitted. The last 22z are punctured or not used. So for base graph 1 rate, instead of the full matrix 46x68, only the 24x46 top left part of the matrix is used.

For base graph 2, only 20z needs to be transmitted, the other 32z are punctured, so instead of a 42x52 matrix, only the 12x22 top left side of the base matrix is used.

Table 4.2 shows the simulation parameters for Figure 4.4, which shows the results for Base Graph 1, 2 for rate half.

Table 4.2 : Simulation parameters for BG1, BG2.

Base Graph	Rate	Iterations	Modulation	Channel	CRC
1	1/2	20	BPSK	AWGN	NONE
2	1/2	20	BPSK	AWGN	NONE

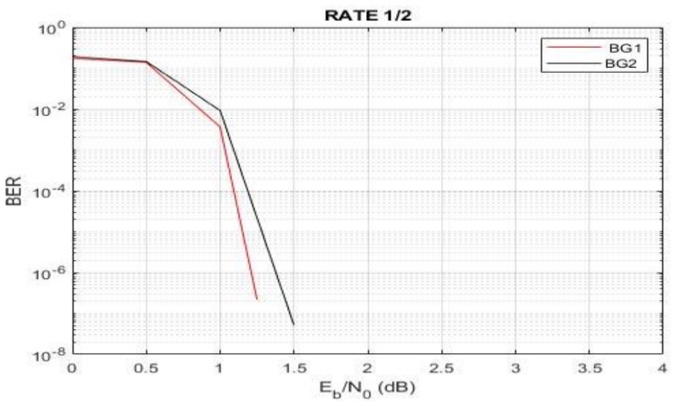


Figure 4.4 : BER of LDPC Codes for Base graph 1 and 2 at rate $\frac{1}{2}$.

Table 4.3 : Performance details for BG1 and BG2 at rate half.

EB/NO	Base Graph	Blocks Sent	BER	Blocks In Error
0	BG1	100	0.17315	100
0	BG2	100	0.18999	100
0.5	BG1	100	0.13905	100
0.5	BG2	100	0.14554	100
1	BG1	1000	0.0036615	139
1	BG2	1000	0.0092125	208
1.25	BG1	20,000	2.1899×10^{-7}	2
1.5	BG2	20,000	5.2083×10^{-8}	2

Both base graphs were tested for BER levels after 0.5 dB. We noticed the fall of both base graphs' BER levels at only 1 dB. Both base graphs had a BER level of10^{-3}.

Both had a great performance with a BER level ending before 2 dB, with base graph 1 slightly outperforming the other as noticed from table 4.3.

For this modulation type, the received vector is a real value, which means that with noise we may end up receiving numbers with high precision. Matlab has a high precision up to 64 bits, but in real life, in order for a decoder to be practical on hardware systems, the range of allowed bits per received value is only 5 bits, so quantization must be implemented for both the LDPC and polar code scenarios.

4.2.2 LDPC codes with different iterations 5, 10,20,30

Different iterations have been tested for rate 2/3 and rate 5/6 on both high and small block sizes. As for LDPC codes, the more iterations the better the results, but after a certain number of iterations, we can see almost no improvement in BER performance. Table 4.4 shows Simulation parameters for rate 5/6, 2/3.

More iterations may provide better BER results, but at the cost of computational power as more row and column operations are used.

Table 4.4 : Simulation parameters for rate 5/6, 2/3.

Block Size	Rate	Iterations	Modulation	Channel	CRC
352	5/6	(5,10,20,30)	BPSK	AWGN	NONE
7040	5/6	(5,10,20,30)	BPSK	AWGN	NONE
600	2/3	(5,10,20,30)	BPSK	AWGN	NONE
5280	2/3	(5,10,20,30)	BPSK	AWGN	NONE

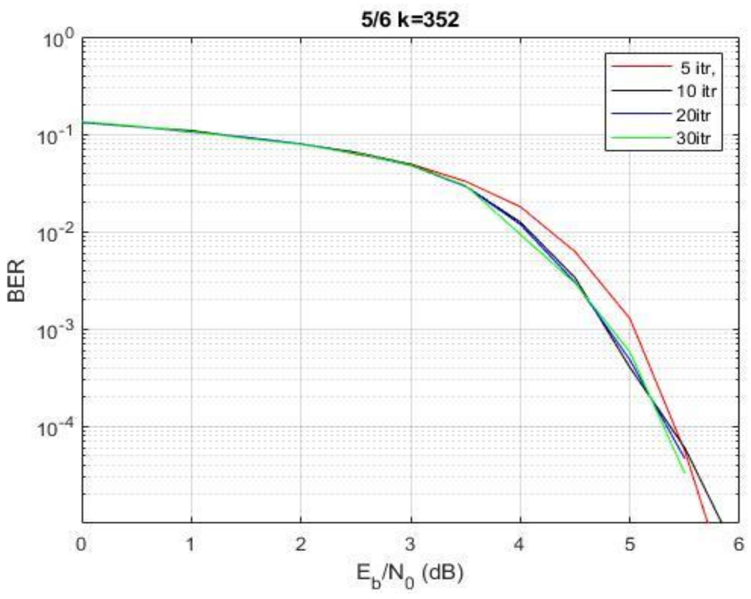


Figure 4.5 : BER for LDPC code at rate 5/6 with different iterations for a small block size

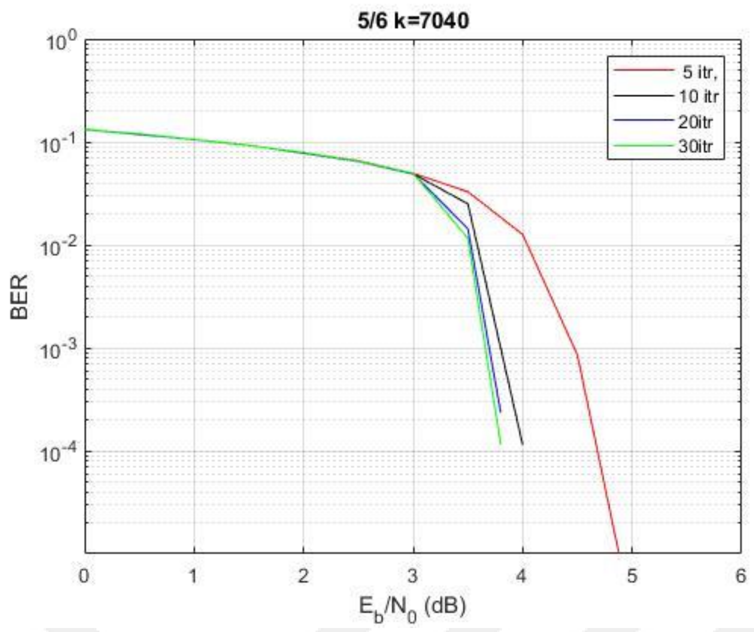


Figure 4.6 : BER for LDPC code at rate 5/6 with different iterations for a big block size.

As for rate, 5/6, both of small and big block size, has bad BER performance compared to other rates for LDPC codes in this thesis.

For the small block size at figure 4.5, 10, 20, and 30 iterations were close in performance, with not that big a difference seen in the BER routine, with 5 iteration being the lowest performer among the iterations; for 5 iteration, the BER was 0.0012699 at 5 dB, while all other iterations had a 10^{-4} BER level. Both of the 20, 30 iterations had a 0.5dB gain compared to the other iterations, ending at 5.5 dB.

Figure 4.6 clearly shows that for large block size, 5 iterations were by far the worst performer this time, with a 1 dB gain behind the 10 iterations.It takes until 4.5 dB so that the BER level for 5 iterations reaches 10^{-4} BER level ,while for 10 iterations it takes until 4 dB to reach its level, and less than 4 dB for both of the 20, 30 iterations to reach that level. Details of performance can be clearly seen from table 4.5 and table 4.6.

Table 4.5 : Performance details rate 5/6 various iteration, small block size.

EB/N0	Iterations	Blocks sent	BER	Blocks In Error
0	5	100	0.13187	100
0	10	100	0.13176	100
0	20	100	0.13241	100
0.5	5	100	0.11952	100
0.5	10	100	0.11912	100
0.5	20	100	0.11972	100
0.5	30	100	0.12085	100
1	5	100	0.10827	100
1	10	100	0.10892	100
1	20	100	0.10497	100
1	30	100	0.10619	100
1.5	5	100	0.091591	100
1.5	10	100	0.091278	100
1.5	20	100	0.092898	100
1.5	30	100	0.090426	100
2	5	100	0.079205	100
2	10	100	0.079006	100
2	20	1000	0.079401	1000
2	30	100	0.078295	100
2.5	5	100	0.064773	100
2.5	10	100	0.065057	100
2.5	20	1000	0.06367	990
2.5	30	2000	0.06403	1974
3	5	100	0.04929	100
3	10	1000	0.049045	946
3	20	2000	0.047692	1819
3	30	2000	0.048136	1833
3.5	5	1000	0.03273	932
3.5	10	1000	0.029287	730
3.5	20	2000	0.029097	1382
3.5	30	2000	0.029729	1417
4	5	1000	0.017955	745
4	10	1000	0.012455	391
4	20	2000	0.011818	688
4	30	2000	0.012706	738
4.5	5	1000	0.006179	390
4.5	10	2000	0.003331	240
4.5	20	2000	0.0029957	209
4.5	30	2000	0.0029304	195
5	5	1000	0.0012699	105
5	10	2000	0.00039915	33
5	20	4000	0.00047656	77

Table 4.6 : Performance details rate 5/6 various iteration, big block size.

EB/NO	iterations	Blocks sent	BER	Blocks in error
0	5	100	0.13226	100
0	10	100	0.13257	100
0	20	100	0.13152	100
0	30	100	0.13179	100
0.5	5	100	0.11858	100
0.5	10	100	0.11811	100
0.5	20	100	0.11867	100
0.5	30	100	0.11992	100
1	5	100	0.10568	100
1	10	100	0.10541	100
1	20	100	0.10512	100
1	30	100	0.10544	100
1.5	5	100	0.092476	100
1.5	10	100	0.092541	100
1.5	20	100	0.092232	100
1.5	30	100	0.092207	100
2	5	100	0.078452	100
2	10	100	0.077845	100
2	20	100	0.078558	100
2	30	100	0.078906	100
2.5	5	100	0.065547	100
2.5	10	100	0.064457	100
2.5	20	100	0.06447	100
2.5	30	1000	0.064811	1000
3	5	100	0.0495	100
3	10	100	0.048918	100
3	20	100	0.049128	100
3	30	1000	0.049687	1000
3.5	5	100	0.03275	100
3.5	10	100	0.02503	100
3.5	20	2000	0.014372	1172
3.5	30	2000	0.011538	770
4	5	100	0.012608	100
4	10	1000	0.00011435	73
3.8	20	4000	0.00023448	66
3.8	30	4000	0.00011555	20
4.5	5	1000	0.00085057	890
5.0	5	1000	2.4148×10^{-6}	16

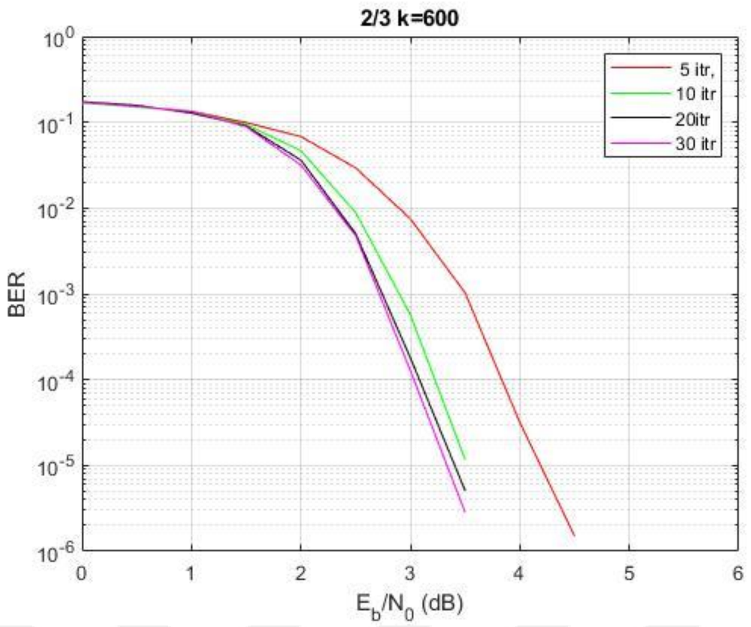


Figure 4.7 : BER for LDPC code at rate 2/3 with different iterations for a small block size

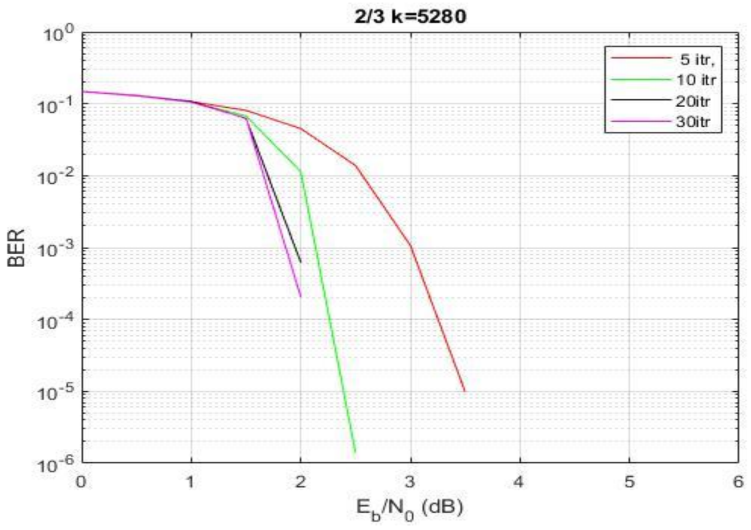


Figure 4.8 : BER for LDPC code at rate 2/3 with different iterations for a big block size.

For figure 4.7 block size, 5 iterations had the lowest performance compared to other iterations. 10,20,30 had a close performance as the 20,30 iteration did slightly better than the 10 iterations, achieving 10^{-6} BER level at the same EB/NO value.
Rate 2/3 performed better than rate 5/6; for the large block size iterations 20,30 were close in performance, with 10 iterations 0.5 dB lower in gain compared to them, and 5 iterations performed the worst by far. As seen from figure 4.8.

Table 4.7 : Comparison between block sizes last BER level at rate 2/3.

Block Size	Iteration	BER	EB/NO	Blocks Sent	Blocks In Error
600	5	1.5×10^{-6}	4.5	10000	5
600	10	1.1667×10^{-5}	3.5	10000	4
600	20	5.0833×10^{-6}	3.5	20000	1
600	30	2.8333×10^{-6}	3.5	20000	1
5280	5	9.8295×10^{-6}	3.5	10000	1
5280	10	1.3826×10^{-6}	2.5	10000	28
5280	20	0.00062231	2	10000	378
5280	30	0.00020744	2.5	10000	74

As seen from figures 4.5 to 4.8 and in detail from tables 4.5 to 4.7, The two of 20 iterations and 30 iterations had an overall better performance compared to the other iterations. As the performance of them was close enough with not that big an improvement in BER level at 30 iterations, 20 iterations were selected as the iteration level for the next section and for rate half.

4.2.3 LDPC codes with different rates

As 20 iterations were selected, the famous block sizes of 256, 512, 1024, 2048, and 4096 BER performance under rates of 2/3 and 5/6 were tested.

Table 4.8 : Simulation parameters for block sizes 256 to 4096.

Block Size	Rate	Iterations	Modulation	Channel	CRC
(256,512,1024,2048,4096)	5/6	20	BPSK	AWGN	NONE
(256,512,1024,2048,4096)	2/3	20	BPSK	AWGN	NONE

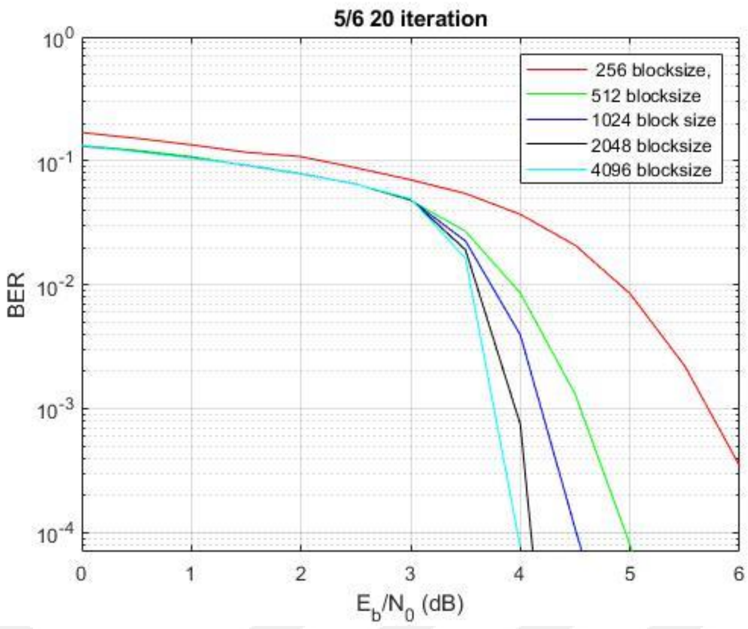


Figure 4.9 : BER for LDPC code at rate 5/6 for 256, 512, 1024, 2048, 4096 block sizes.

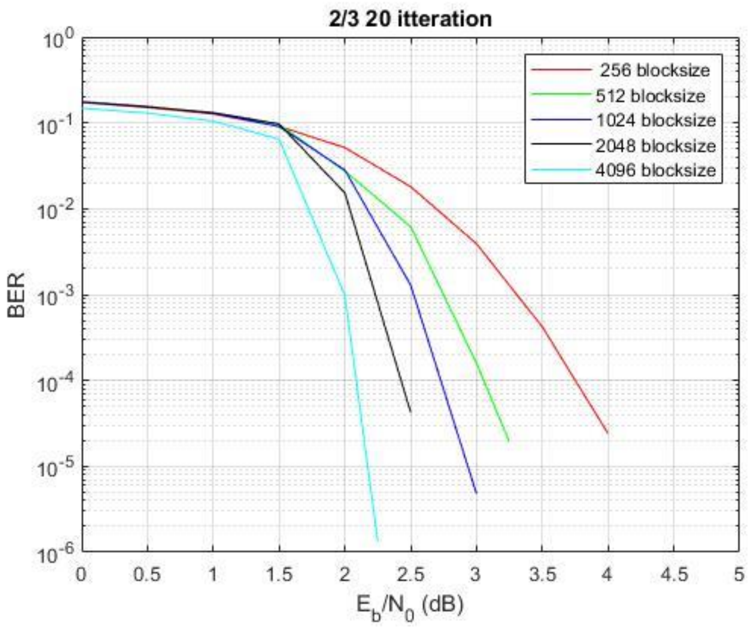


Figure 4.10 : BER for LDPC code at rate 2/3 for 256, 512, 1024, 2048, 4096 block sizes.

As for figure 4.9, the 256 block size had the worst performance by far compared to other block sizes, requiring more than 6 dB for BER levels to end, with more than 4 dB needed so that BER levels start to reduce. The 4096 block size had the best overall BER performance, ending with a BER level of 7.55×10^{-5} at 4 dB That makes it have almost a 0.5 dB gain to the closest performance block size of 2048.

Rate 2/3 had an overall better performance than the 5/6 rate; the 256 block size had almost the same performance as the 4096 at rate 5/6, with a BER level of 2.4×10^{-5} at 4 dB. Here also, the biggest block size of 4096 had the best BER performance, with a BER level of 1.32×10^{-6} at 2.25 dB That makes it have an almost 2 dB gain compared to rate 5/6. Details of performance can be seen from table 4.9.

Table 4.9 : Comparison between block sizes last BER level at rate 5/6, 2/3.

Block Size	Rate	BER	EB/NO	Blocks Sent	Blocks In Error
256	5/6	1.2821×10^{-6}	7	30000	4
256	2/3	2.4038×10^{-5}	4	4000	2
512	5/6	3.4091×10^{-6}	5.5	10000	2
512	2/3	1.9231×10^{-5}	3.25	4000	1
1024	5/6	3.3144×10^{-6}	5	10000	1
1024	2/3	4.8077×10^{-6}	3	2000	1
2048	5/6	1.8939×10^{-7}	4.4	5000	1
2048	2/3	4.2548×10^{-5}	2.5	4000	5
4096	5/6	7.5521×10^{-5}	4	2000	7
4096	2/3	1.3258×10^{-6}	2.25	10000	2

4.3 Polar Codes Results

Moving to polar codes, the main component in the operations of polar codes is known as the generator matrix or transform kernel.

$$G2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.11}$$

G4 is obtained from G2 by using what is known as a kronecker product, so we have this matrix.

$$G4 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{4.12}$$

$$G4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.13}$$

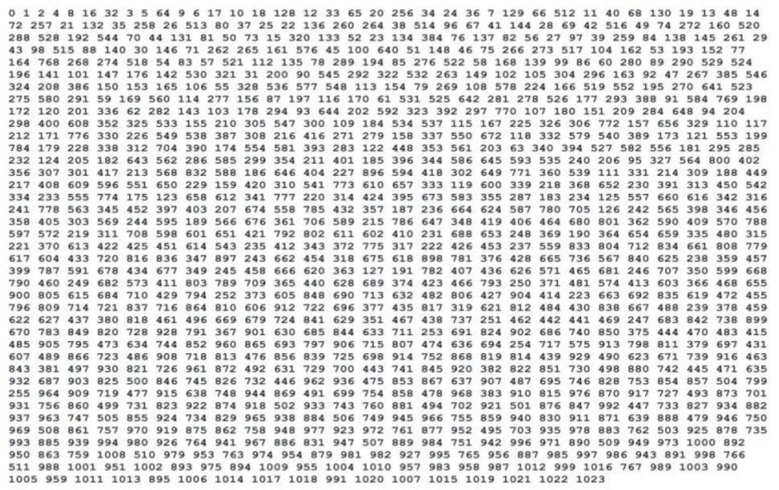
So the Kronecker product works as every value in the second matrix is replaced by the previous matrix if it’s 1, and replaced by all 0 values with the size of the previous matrix if the value is 0. So we can generally use this equation.

$$GN = (G2)^{\otimes n} \tag{4.14}$$

In 5G, the maximum N value is 1024, so the maximum n value here is 10.

$$N = 2^{n} \tag{4.15}$$

Before we get into coding, we need to define a reliability sequence. This is where the frozen bits or the worst channels are discarded before encoding message bits.



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

Figure 4.11 : Polar for N=1024.

So assuming (N, K) polar codes for encoding a message of length k, as the maximum N is 1024, the message should be 1023 or less.

Depending on the chosen N and message size K, N-K values are frozen and they are considered the least reliable channels.

$$codeword = u.GN \qquad (4.16)$$

The operations for encoding used in this thesis are shown in the figure below.

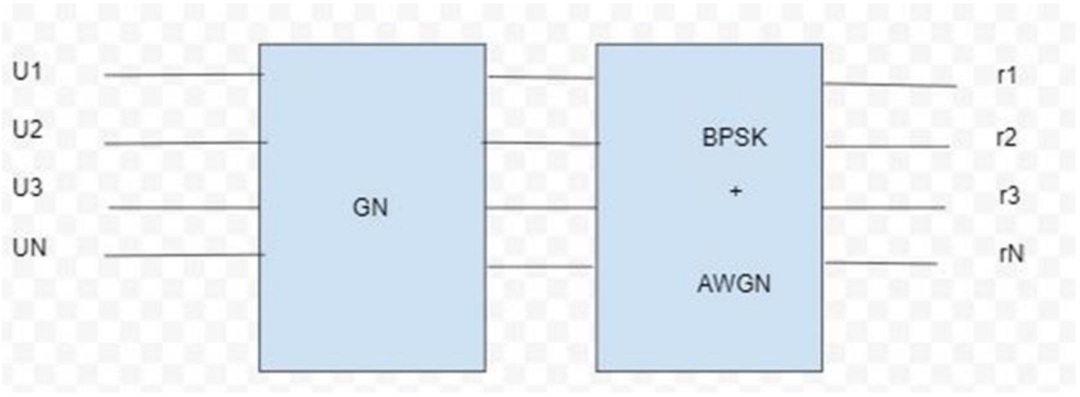


Figure 4.12 : Encoding operations for Polar codes.

BPSK is the same as in the LDPC procedure; 0 goes 1 and 1 goes -1. AWGN is an additive white Gaussian noise with a mean zero.

The figure below shows a small example of an (8,4) polar code and how the encoding procedure works. This is for illustration purposes only, as actual polar codes are always bigger than that.

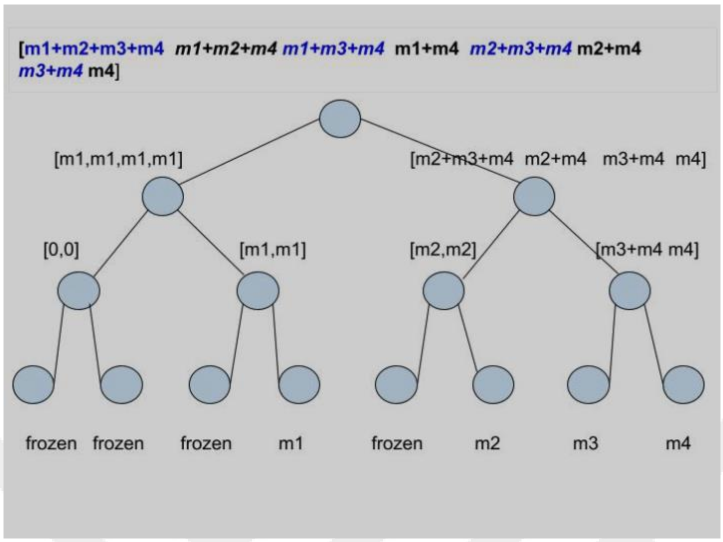

[m1+m2+m3+m4 m1+m2+m4 m1+m3+m4 m1+m4 m2+m3+m4 m2+m4 m3+m4 m4]
[m1,m1,m1,m1]
[m2+m3+m4 m2+m4 m3+m4 m4]
[0,0]
[m1,m1]
[m2,m2]
[m3+m4 m4]
frozen
frozen
frozen
m1
frozen
m2
m3
m4

Figure 4.13 : Encoding for (8,4) Polar code.

For the 5G standard, the reliability sequence of N = 8 is (1 2 3 5 4 6 7 8) so for k = 4, then the first four bits are frozen. The same concept goes for all the way up to the maximum N value. For decoding, there are two different operations starting with SC decoding. To illustrate the idea of a decoder, let us show a small decoder.

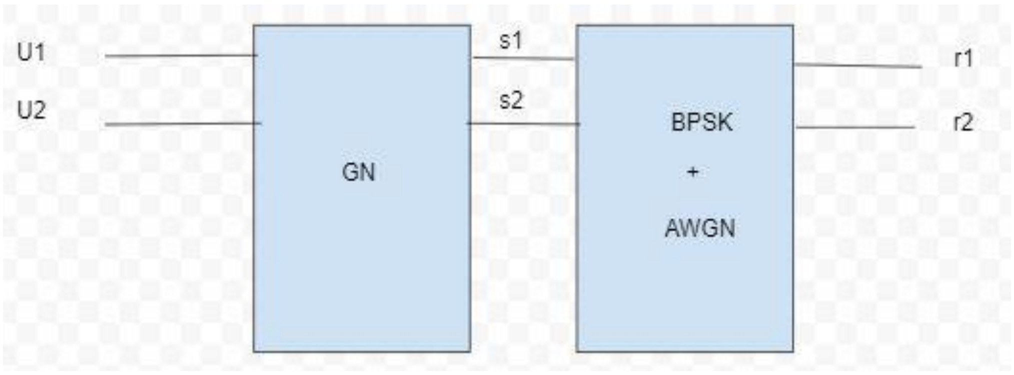


Figure 4.14 : Small decoder for polar of N=2.

The received values r1 and r2 are what we need to recover the sent bits U1 and U2. So the decoder job is obtaining U1 U2 to get the soft information of U1 U2.

The kernel can be inverted to other sides by means of U1 = S1+S2, U2 = S2

We obtain r1 when s1 transmitted and r2 received when s2 transmitted; value of U1 is needed and U1=s1+s2.

There is a belief for s1 and a belief for s2, and what is needed is the belief for s1+s2. This is the same as used in LDPC codes. The SPC problem is the extrinsic information min-sum operations used.

$$\widehat{u1} = sgn\,(r1).sgn\,(r2).min\,(\,|r1\,|,|r2\,|) \tag{4.17}$$

To obtain the value of the U1 threshold, 0 is used. To determine the value of a bit, make a direct hard decision.

$$\widehat{u1}\ \geq\ 0, \widehat{u1}\ =\ 0 \tag{4.18}$$

$$\widehat{u1} <\ 0, \widehat{u1}\ =\ 1 \tag{4.19}$$

If $\widehat{u1}$ is zero, this means s = [u2 u2] In other words, repetition code, both of r1 and r2 are beliefs for U2, because the assumption based on the value of $\widehat{u1}$ is 0. You add the beliefs r1 and r2.

If $\widehat{u1}$ is 1, here r1 is belief for the compliment of u2 in BPSK, as 0 goes for 1 and 1 goes for-1 for a particular bit, the compliment is the belief in negative. So for $\widehat{u1}$, the transmission was the compliment of u2 and u2. Here, r1 is a belief for u2complement, so –r1 is an additional belief for us. So the total belief is r2-r1.

Here u1 and u2 are separately decoded this is how the Polar decoder works it's different than the overall decoding methods. The guess for u1 is directly used to obtain u2 this is how the successive cancelation works.

The scenario above is the lowest N value, which is 2. This is only for illustration of how the mathematic part is done. This is how the SC decoder works.

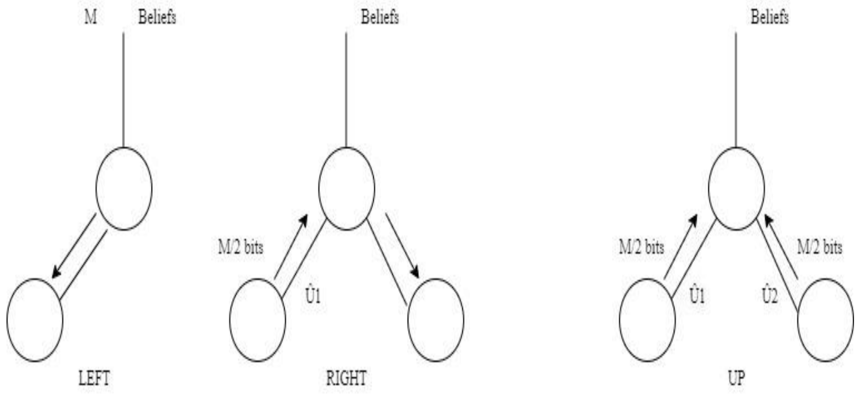


Figure 4.15 : Steps of SC Polar decoding.

The figure shows the operations happen at all nodes. Leaf nodes are not included. So it can be summarized in three steps: going left, then right, and finally up.

The left step is the first step where the min-sum operation occurs. What occurs is that the incoming belief vector is divided, so we get M/2 beliefs sent back up.

After the bits are sent back up again, the right-side operations occur, which is basically the repetition code as the belief vector splits into two halves and with Û1, repetition operations occur and M/2 values are sent up

Finally, the going up step, which is basically: [Û1+Û2   Û2]. When the node is a leaf, however, there are two options. it's frozen, then the value is zero. The other options, if it's not frozen, directly threshold applied to decide whether the value is 0 or 1.

4.3.1 Polar codes for 256 512 blocks with SC

The BER performance of both the 256 and 512 block sizes was tested using the conditions listed in table 4.10. Details are seen from tables 4.11 and 4.12.

Table 4.10 : Simulation parameters for 256 512 block sizes polar codes.

Block Size	N	Iterations	Modulation	Channel	CRC
256	1024	NONE	BPSK	AWGN	NONE
512	1024	NONE	BPSK	AWGN	NONE

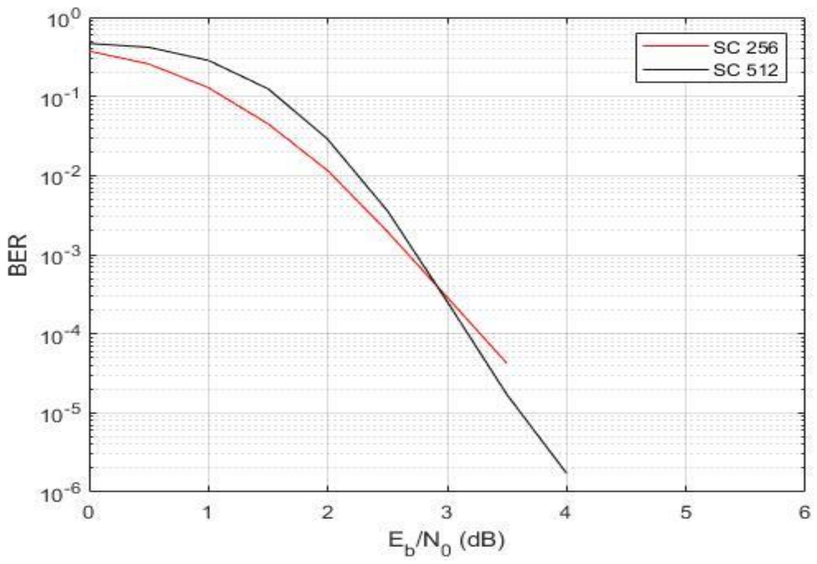


Figure 4.16 : BER for polar codes 256,512 block sizes with SC.

The 256 block size had a better BER performance compared to 512 as the difference of BER performance shows from beginning. A BER of 4×10^{-5} is obtained at 3.5 dB out of the 100,000 blocks sent only32 blocks were in error after that no BER is detected on the other hand an extra 0.5 dB needed for 512 block size .

Table 4.11 : Performance details for 256 block size polar code.

EB/NO	Blocks sent	BER	Blocks in error
0	1000	0.37278	918
0.5	2000	0.25721	1457
1	4000	0.12883	1632
1.5	10,000	0.044875	1627
2	20,000	0.011508	950
2.5	50,000	0.0019473	494
3	50,000	0.00028844	87
3.5	100,000	4.227×10^{-5}	32

Table 4.12 : Performance details for 512 block size polar code.

EB/NO	Blocks sent	BER	Blocks in error
0	1000	0.46618	997
0.5	2000	0.41721	1942
1	4000	0.28477	3088
1.5	10,000	0.12455	4022
2	20,000	0.028463	2175
2.5	50,000	0.0035666	878
3	50,000	0.00025449	100
3.5	100,000	1.7031×10^{-5}	17
4	100,000	1.7188×10^{-6}	3

4.3.2 Polar codes for different rates with SC

Different rates for polar codes SC have been applied with the parameters in table 4.13.

Table 4.13 : simulation parameters for polar codes at different rates with SC.

Block Size	Rate	Iterations	Modulation	Channel	CRC
205	1/5	NONE	BPSK	AWGN	NONE
683	2/3	NONE	BPSK	AWGN	NONE
853	5/6	NONE	BPSK	AWGN	NONE

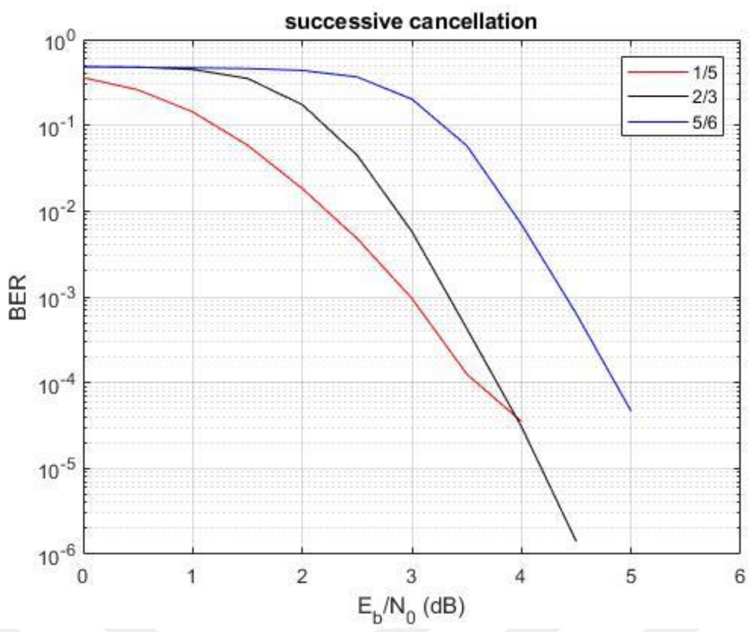


Figure4.17 : BER for polar codes at different rates with SC.

As it is seen from fig. 4.17, 1/5 has a better overall BER compared to others. On the other hand, 5/6 is the poorest compared to them. We can notice the difference at 1dB; the BER curve for 1/5 is 0.30365, or in other words, 30.36% better than 2/3. Also, at 0.32276, or 32.27% better BER than 5/6, at 2dB, the difference can be clearly seen. Especially between 1/5 and 5/6, as 1/5 has 0.41745, or 41.74% better BER performance. As compared with 2/3, it is 0.155283, or a 15.82% better BER performance.

Finally, at EB/N0, no more BER can be noticed after 4dB for 1/5. That is 0.5 dB enhanced more than the 2/3 rate and better than 5/6 by an entire 1dB.

Since the full N value of polar, which is equal to 1024, is used, that means the polar codes always give better performance the smaller the block size gets. Table 4.14 shows the SC performance details for different rates.

Table 4.14 : Performance details for Polar codes at different rates with SC.

EB/N0	Rate	Blocks sent	BER	Blocks in error
0	1/5	1000	0.3592	900
0	2/3	1000	0.48181	1000
0	5/6	1000	0.47764	1000
0.5	1/5	2000	0.25848	1458
0.5	2/3	2000	0.47605	2000
0.5	5/6	2000	0.47246	2000
1	1/5	4000	0.14314	1788
1	2/3	4000	0.44705	3970
1	5/6	4000	0.46616	4000
1.5	1/5	10,000	0.058473	2046
1.5	2/3	10,000	0.35079	8967
1.5	5/6	4000	0.45805	4000
2	1/5	20,000	0.018237	1411
2	2/3	20,000	0.17352	10806
2	5/6	4000	0.43569	3988
2.5	1/5	50,000	0.0047798	979
2.5	2/3	20,000	0.044655	3400
2.5	5/6	10,000	0.36514	9361
3	1/5	50,000	0.00095649	212
3	2/3	20,000	0.0057658	553
3	5/6	10,000	0.20151	6233
3.5	1/5	50,000	0.00012605	43
3.5	2/3	50,000	0.00042878	166
3.5	5/6	10,000	0.057725	2184
4	1/5	100,000	3.5268×10^{-5}	20
4	2/3	100,000	3.063×10^{-5}	35
4	5/6	20,000	0.006987	704
4.5	1/5			
4.5	2/3	100,000	1.4056×10^{-6}	2
4.5	5/6	40,000	0.00063532	170
5	1/5			
5	2/3			
5	5/6	100,000	4.6471×10^{-5}	45

4.3.3 Polar codes for 256,512 block sizes with SCL+CRC

Moving to SCL for these codes, the procedure of encoding is the same as SC codes, but the difference is in decoding. As the decoder in SC can be considered as a list of size 1, it only takes one path for decoding; on the other hand, SCL splits into lists like 2 or 4 or 8. With the help of CRC, the original bits get found in case the CRC series is not found in the list or the path with the lowest path metric is chosen here.

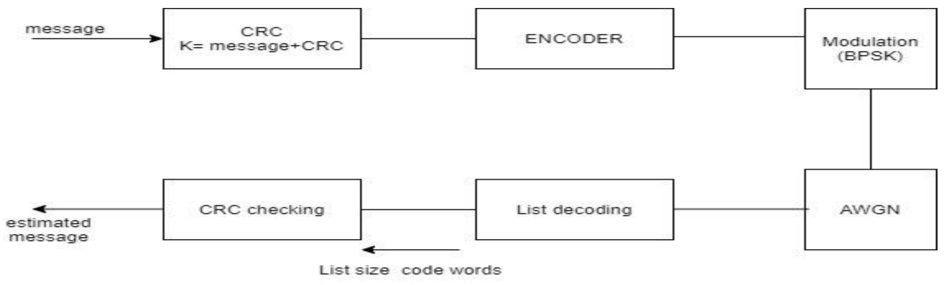


Figure 4.18 : SCL operations.

Table 4.15 shows the simulation parameters for Figure 4.19 and Figure 4.20, which show the results for 256 and 512 block sizes of BER performance with CRC-aided successive cancelation list.

Table 4.15: simulation parameters for 256,512 block sizes for Polar codes with SCL.

Block Size	List Size	Iterations	Modulation	Channel	CRC
256	(2,4,8)	NONE	BPSK	AWGN	11
512	(2,4,8)	NONE	BPSK	AWGN	11

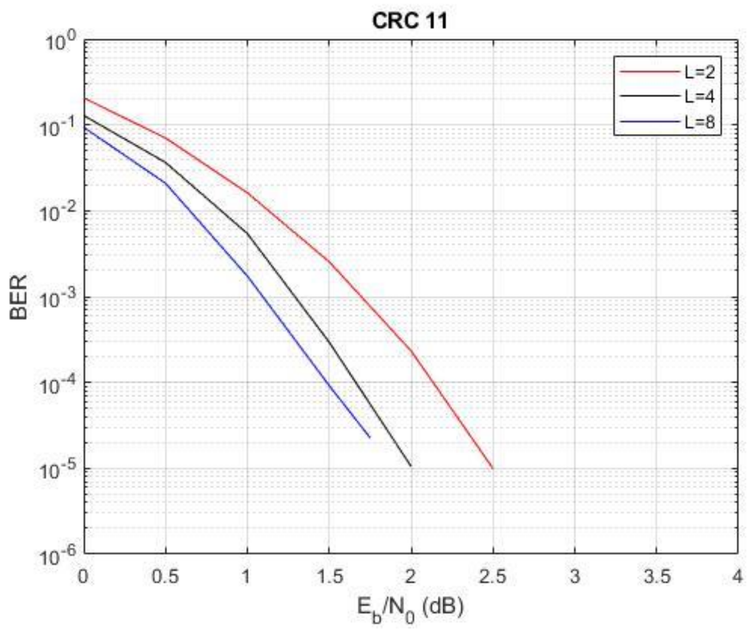


Figure 4.19 : BER for polar code 256 block size with SCL.

For 256 block size, the overall results had a 1 dB gain compared to SC only when using list 2. List 4 had a 1.5 gain compared with SC, and finally, list 8 had more than 1.5 gain compared to the SC method. Also, the BER level, especially when lists 4 and 8 are applied, shows significant improvement, as low BER levels can be noticed after only 0.5 dB.

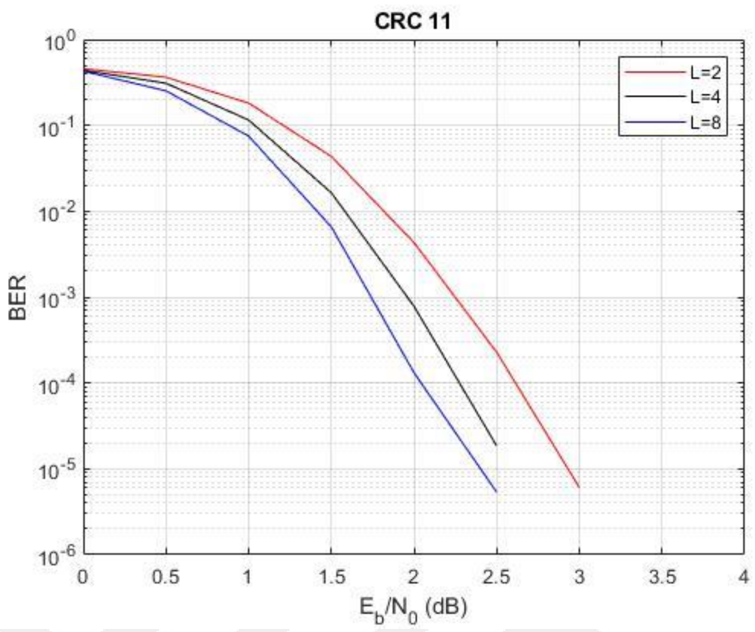


Figure 4.20 : BER for polar code 512 block size with SCL.

As for 512 block size, the overall result had a 1 dB improvement only when a list of 2 is applied. List sizes 4 and 8 provide close results, with list size 8 attaining slightly better results. Both of them have a 1.5 dB gain over the SC method. It takes until 2 dB before we notice a relatively low BER at SC, but in CRC SCL, especially for list sizes 4 and 8, we can clearly see a low BER obtained directly after 1 dB.

For figures of SCL with CRC implemented, performance clearly improved for both 256 and 512. Also, in SCL, again, lower sized blocks perform better.

Table 4.16 : Comparison between 256, 512 block sizes last BER level at L=2, 4, 8.

EB/NO	List size	BER	Block size	Blocks sent	Blocks in error
2.5	2	9.9609×10^{-6}	256	20,000	2
2	4	1.0547×10^{-5}	256	10,000	1
2	8	2.2656×10^{-5}	256	10,000	1
3	2	6.0547×10^{-6}	512	100,000	10
2.5	4	1.8516×10^{-5}	512	100,000	18
2.5	8	5.3516×10^{-6}	512	100,000	2

4.3.4 Polar codes for different rates with SCL+CRC

Various rates have been tested with the SCL method with list size 4. Even though list 8 provided slightly better results in the previous subsection, with the increasing number of lists at the PC specifications that were mentioned in Table 4.1, the operations consume a very long time. That's why the list with size 4 was picked for this subsection.

Table 4.17: Simulation parameters for polar codes different rates with SCL.

Block Size	Rate	Iterations	Modulation	Channel	CRC
205	1/5	NONE	BPSK	AWGN	11
683	2/3	NONE	BPSK	AWGN	11
853	5/6	NONE	BPSK	AWGN	11

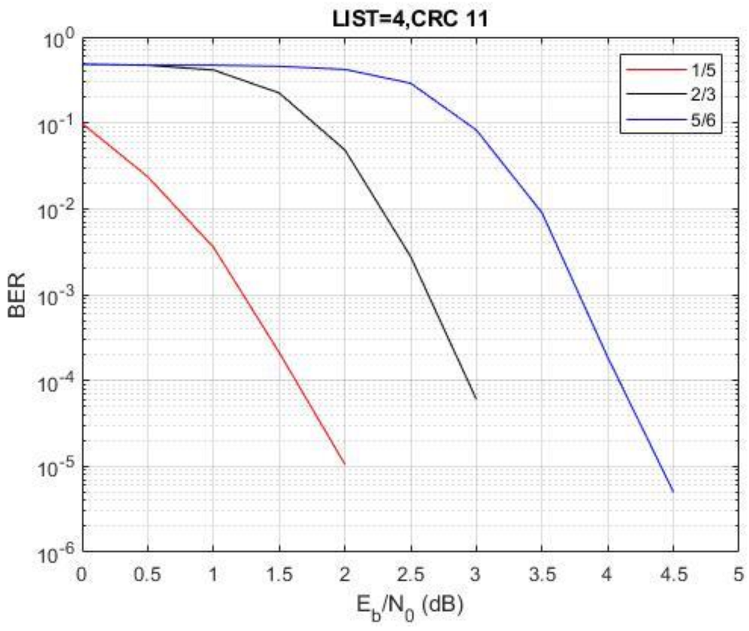


Figure 4.21: BER for polar codes at different rates with SCL.

The same rates for SC have been implemented again, but this time using SCL with CRC. At the rate of 5/6, there is a 0.5 dB gain compared to the SC method. It took until 4 for SC to reach 0.006987 BER, while 0.0089244 BER was reached at 3.5 for SCL. It's also noticed that the performance of this rate is the worst among them for both cases. Even when SCL is implemented at 1.5 dB, all the sent blocks are in error.

For 2/3, the achieved 1.5 gain compared to the SC method while the BER level was better at SCL. At 3dB the SC method had a 0.0057 BER level while at the same value, the BER was 6.0417×10^{-5} for SCL.

Finally, the 1/5 had the overall best performance in both SC and SCL. The SCL had a 2dB gain compared to the SC method for SC at 2.5 dB, achieving 0.0047798 BER, while it only took 1 dB to obtain 0.0035585 BER for SCL.

Table 4.18 : Performance details for polar codes at different rates with SCL.

EB/N0	Rate	Blocks sent	BER	Blocks in error
0	1/5	1000	0.098566	329
0	2/3	1000	0.48373	1000
0	5/6	1000	0.47868	1000
0.5	1/5	1000	0.023512	89
0.5	2/3	2000	0.46804	1998
0.5	5/6	1000	0.47021	1000
1	1/5	2000	0.023512	37
1	2/3	2000	0.46804	1918
1	5/6	1000	0.47021	0.46846
1.5	1/5	10000	0.00021171	13
1.5	2/3	4000	0.22266	2496
1.5	5/6	2000	0.45544	2000
2	1/5	10000	1.0537×10^{-5}	4
2	2/3	4000	0.048256	645
2	5/6	2000	0.41941	1962
2.5	1/5			
2.5	2/3	4000	0.0027701	52
2.5	5/6	2000	0.2891	1568
3	1/5			
3	2/3	100,000	6.0417×10^{-5}	37
3	5/6	4000	0.08236	1120
3.5	1/5			
3.5	2/3			
3.5	5/6	4000	0.0089244	148
4	1/5			
4	2/3			
4	5/6	10,000	0.00018511	14
4.5	1/5			
4.5	2/3			
4.5	5/6	100,000	5.0059×10^{-6}	6

4.4 Comparing Results

In this thesis, the BER relation for LDPC codes and polar codes was applied to different plots by using MATLAB 2020 software. As for LDPC codes, both base graphs were used. For rate half, with midsize block BG1, had a better performance; for rates 2/3 and 5/6, rate 2/3 had an overall better performance compared to rate 5/6, with 4096 block size providing the best results in both rates. For polar codes, SCL gave better results compared to only using SC, with good results for lists 4 and 8. In general, as noticed from these chapter BER figures, Polar codes had a better BER performance at short block sizes, while LDPC codes gave a better BER performance at larger block sizes.

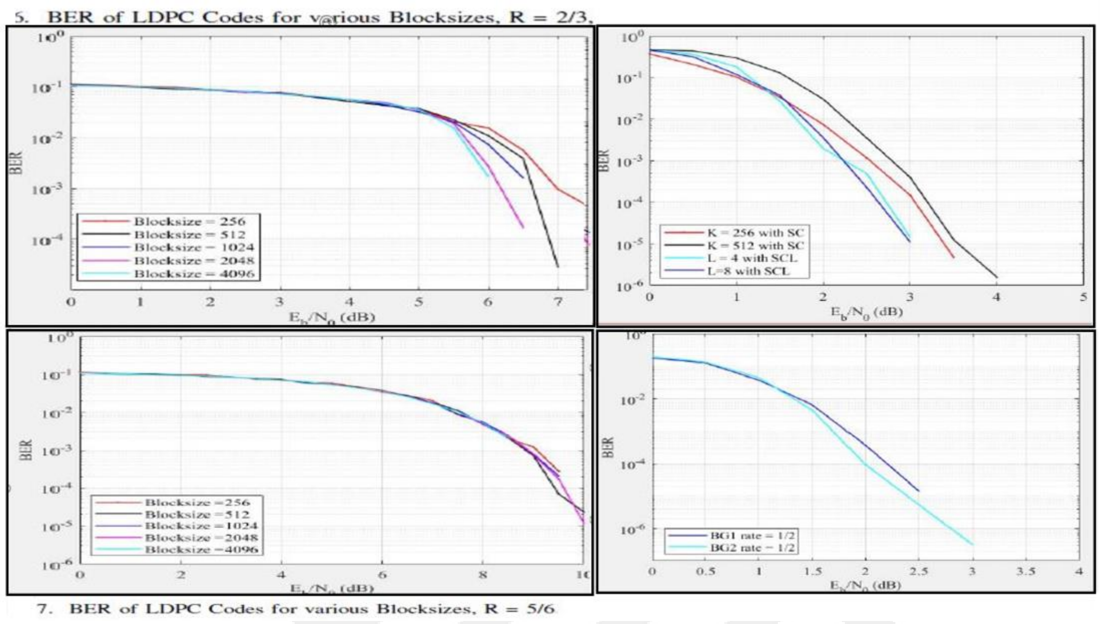


Figure 4.22 : Various BER for LDPC and Polar codes[16].

From figure 4.22, which shows the results of BER curves using QAM modulation, we can see that the results obtained in this thesis provide the same behaviour for LDPC codes. The 4096 block size gave the best results; the 2/3 had a better overall performance than the 5/6; polar codes with SCL had a better performance than the SC. Also, the LDPC codes were better at high block sizes and polar codes were better at lower block sizes, so we can clearly see the same code behaviour for both results, though the results obtained in this thesis were better in EB/NO gain due to the usage of BPSK modulation.

5. CONCLUSION

In this thesis, a direct low-coast approach has been used to check the BER performance for 5G codes. That model can be extended to other codes. The suggested approach uses BPSK modulation and AWGN for the channel. This combination provides the ideal scenario for the code to be simulated under.

This approach can be used by any researcher around the world, whether for the same topic (5G) as higher lists for polar codes, other rates and blocks for LDPC codes if the hardware capabilities for the simulation device allow these kinds of operations, or for any future codes that will be suggested for upcoming techniques.

This approach can tell you how the code performs under the best circumstances for a communication system, and you can see if at a specific rate or at a block size range if the code performance is bad from early stages, which will give you the chance to work on them and fix the issues before the need to apply complicated high QAM modulations.

The movement of the thesis provides an explanation. This work represents more than a year of working in the research field, and it could be used as a reference for other publications.

REFERENCES

[1] **Sae-Young Chung, Forney, G., Richardson, T., & Urbanke, R.** (2001). On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Communications Letters, 5*(2), 58–60.

[2] **Vigneswari, P., & Sivakumari, S.** (2021). Improving the Performance of Layered Iterative minsum Approximation Message Passing Decoding Algorithm for 5G NR LDPC Codes. *Journal of Physics: Conference Series, 1917*(1), 012017.

[3] **Patil, M. V., Pawar, S., & Saquib, Z.** (2020). Coding Techniques for 5G Networks: A Review. *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA).*

[4] **Dhuheir, M., & ÖZTÜRK, S.** (2019). Implementation of Polar Codes in 5G Systems with Different Waveform Modulations by Using USRP. *Sakarya University Journal of Science*, 1144–1153.

[5] **Shannon, C. E.** (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(4), 623–656.

[6] **Nyquist, H.** (2002). Certain topics in telegraph transmission theory. *Proceedings of the IEEE*, *90*(2), 280–305.

[7] **Verdu, S. (1998).** Fifty years of Shannon theory. *IEEE Transactions on Information Theory*, *44*(6), 2057–2078.
[8] **Hartley, R.V.L.** (1927). Transmission of Information. International Congress of Telegraphy and Telephony, 535-563.

[9] **Url-1**<https://www.linkedin.com/pulse/rate-matching-5gnr-ldpc-codes-naveen-chelikanitrk=portfolio_article-card_title.com>(accessed 15.10.2021).

[10] **Lal, N., Tiwari, S. M., Khare, D., & Saxena, M.** (2021). Prospects for Handling 5G Network Security: Challenges, Recommendations and Future Directions. *Journal of Physics: Conference Series*, *1714*(1), 012052.

[11] **Weisner., J. B.** (2005, August 16). *A device for quantizing, grouping, and coding amplitude-modulated pulses*. DSpace@MIT. http://dspace.mit.edu/handle/1721.1/12390

[12] **Tjalkens, T., & Willems, F.** (1987). Variable to fixed-length codes for Markov sources. *IEEE Transactions on Information Theory*, *33*(2), 246–257. https://doi.org/10.1109/tit.1987.1057285

[13] **Wolfowitz, J.** (1957). The coding of messages subject to chance errors. *Illinois Journal of Mathematics, 1*(4).

[14] **Costello, D. J., & Forney, G. D.** (2007). Channel coding: The road to channel capacity. *Proceedings of the IEEE*, *95*(6), 1150–1177.

[15] **Čarapić, D.** (2021). 4G vs 5G Channel Coding Techniques. *20th International Symposium INFOTEH-JAHORINA*, 255–259.

[16] **Patil, G. S.** (2017). Advanced Coding Techniques in 3G and 4G. *Journal of Electronics and Communication Systems*, 1–6.

[17] **Krouk, E., & Semenov, S. (Eds.).** (2011). Modulation and Coding Techniques in Wireless Communications. *Book.*

[18] **Reed-Solomon Codes and the Compact Disc.** (2009). *Reed-Solomon Codes and Their Applications*.

[19] **Golam Sadeque, M.** (2015). Bit Error Rate (BER) Comparison of AWGN Channels for Different Type's Digital Modulation Using MATLAB Simulink. *Am. Sci. Res. J. Eng. Technol. Sci.*, *13*, 61–71.

[20] **Pawula, R.** (1984). On M-ary DPSK Transmission Over Terrestrial and Satellite Channels. *IEEE Transactions on Communications*, *32*(7), 752–761.

[21] **Khan, M. A., Pal, S., & Jose, A.** (2015). BER Performance of Digital Modulation Schemes With and Without OFDM Model for AWGN, Rayleigh and Rician Channels. *International Journal of Science and Research (IJSR), 4*(11), 330–335.

[22] **YANG, R. (2010).** LDPC-coded Modulation for Transmission over AWGN and Flat Rayleigh Fading Channels [Master's thesis, University of Laval Quebec]. University of Laval Quebec Digital Assets. Available [Online].

[23] **Kolaylı, M.** (2006). Comparison of decoding algorithms for low-density parity-check codes [M.S. - Master of Science]. Middle East Technical University.

[24] **Tu, Z., & Zhang, S.** (2007). Overview of LDPC Codes. *7th IEEE International Conference on Computer and Information Technology (CIT 2007).*

[25] **Nguyen, T. T. B., Nguyen Tan, T., & Lee, H.** (2019). Efficient QC-LDPC Encoder for 5G New Radio. *Electronics*, *8*(6), 668.

[26] **Li, H., Bai, B., Mu, X., Zhang, J., & Xu, H.** (2018). Algebra-Assisted Construction of Quasi-Cyclic LDPC Codes for 5G New Radio. *IEEE Access*, *6*, 50229–50244.

[27] **Arora, K., Singh, J., & Randhawa, Y. S.** (2019). A survey on channel coding techniques for 5G wireless networks. *Telecommunication Systems*, *73*(4), 637–663.

[28] **Petrovic, V. L., Markovic, M. M., Mezeni, D. M. E., Saranovac, L. V., & Radosevic, A**. (2020). Flexible High Throughput QC-LDPC Decoder With Perfect Pipeline Conflicts Resolution and Efficient Hardware Utilization. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *67*(12), 5454–5467.

[29] **Ali, Z., Fredrik, A. and Jonas ,M.** (2018). 5G Physical Layer (1st ed). ELSEVIER.

[30] **Technical Specification** (2018). Multiplexing and Channel Coding (3GPP TS 38.212 Version 15.2. 0 Release 13). *ETSI TS*, *136*(212), v13.

[31] **Url-2**<https://www.linkedin.com/pulse/5g-nr-dl-sch-ldpc-channel-coding- base-graph-selection-chelikani.com> (accessed 13.10.2021).

[32] **Bae, J. H., Abotabl, A., Lin, H. P., Song, K. B., & Lee, J.** (2019). An overview of channel coding for 5G NR cellular communications. *APSIPA Transactions on Signal and Information Processing*, *8*(1).

[33] **Zhang, H., Li, R., Wang, J., Dai, S., Zhang, G., Chen, Y., Luo, H., & Wang, J.** (2018). Parity-Check Polar Coding for 5G and Beyond. *2018 IEEE International Conference on Communications (ICC)*.

[34] **Bioglio, V., Condo, C., & Land, I.** (2021). Design of Polar Codes in 5G New Radio. *IEEE Communications Surveys & Tutorials*, *23*(1), 29–40.

[35] **Rao, K. D., & Babu, T.** (2019). Performance Analysis of QC-LDPC and Polar Codes for eMBB in 5G Systems. *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*.

[36] **Goel, C., Sarkar, A., Kumaravelu, V. B., & Gudla, V. V.** (2020). Polar Codes for 5G New Radio. *2020 International Conference on Communication and Signal Processing (ICCSP)*.

[37] **Garro, E., Fuentes, M., Gomez-Barquero, D., & Carcel, J. L.** (2019). 5G Mixed Mode: An Innovative Point-to-Multipoint Solution for New Radio. *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*.

[38] **Hui, D., Sandberg, S., Blankenship, Y., Andersson, M., & Grosjean, L.** (2018). Channel Coding in 5G New Radio: A Tutorial Overview and Performance Comparison with 4G LTE. *IEEE Vehicular Technology Magazine*, *13*(4), 60–69.

CURRICULUM VITAE

Name: MOHAMED A YOUSSEF SANFAZ

Education

2017	B.SC. Electrical and electronics engineering
2019-now	M.Sc. Communication Systems