People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Amar Telidji University - Laghouat

FACULTY OF TECHNOLOGY

DEPARTMENT: ELECTRONICS

Master Dissertation

Prepared by: Nouacer Mohammed Abdelhafid and Othmane Mahmoud

DOMAIN: Science and Technology

FACULTY: Telecommunications

OPTION: Telecommunication Systems

Theme

Permutation Polynomial Interleavers for Turbocodes

Dissertation Jury:

Name	Grade	Quality
Chellali Safouane	MCB	Supervisor
Reguigue Mourad	MCB	President
Reggab Mourad	MAA	Examiner

Promotion : 2020/2021

Abstract

In this dissertation, a performance evaluation of different types of interleavers is compared in between each other, these components are ones of the most important elements in turbo codes, there are many different types of interleavers.

The simulation results showed that the Quadratic Permutation Polynomial (QPP) gives promising results when used in a large interleaver sizes, however, it did not show great results in comparison with other interleavers like the Dithered Relative Prime (DRP) and the Dithered Golden Interleaver (DGI) in smaller interleaver lengths.

Several factors, in addition to the kind of interleaver, influence the performance of the turbo code, including the generating polynomial, the number of decoding rounds, and the block size. Increasing these parameters improves speed, but increases simulation time.

Keywords: QPP, Interleavers, Turbo Coding, DRP, DGI, ARP, UNIF, RI, Parallel Concatenation, Permutation, Convolutional codes, Recursive Systematic Codes, Iterative Decoder, MAP Algorithm, Bit Error Rate, Signal to Noise Ratio.

Résume

Dans cette thèse, une évaluation des performances de différents types d'entrelaceurs est comparée entre eux, ces composants sont l'un des éléments les plus importants des turbocodes, il existe de nombreux types différents d’entrelaceurs.

Les résultats de la simulation ont montré que le polynôme de permutation quadratique (PPQ) donne des résultats prometteurs lorsqu'il est utilisé dans un entrelaceur de grande taille, cependant, il n'a pas montré d'excellents résultats en comparaison avec d'autres entrelaceurs comme le Dithered Relative Prime (DRP) et le Dithered Golden Interleaver (DGI) dans des longueurs d'entrelacement plus petites.

Plusieurs facteurs, en plus du type d'entrelaceur, influencent les performances du turbocode, notamment le polynôme générateur, le nombre de tours de décodage et la taille du bloc. L'augmentation de ces paramètres améliore la vitesse, mais augmente le temps de simulation.

Mots clés : L’entrelacement, turbocode, concaténation parallèle, permutation, codes convolutifs, code systématique récursif, décodeur itératif, algorithme MAP, taux d’erreurs binaire, rapport signal sur bruit, DRP, DGI, ARP, QPP, UNIF, RI

ملخص

في هذه الأطروحة ، تتم مقارنة تقييم الأداء لأنواع مختلفة من التشذير فيما بينها ، وهذه المكونات هي من أهم العناصر في أكواد التربو ، وهناك العديد من الأنواع المختلفة من المشذرات.

أظهرت نتائج المحاكاة أن (QPP) يعطي نتائج واعدة عند استخدامه بأحجام تشذير كبيرة ، ومع ذلك ، فإنه لم يُظهر نتائج جيدة مقارنة بالمشذرات الأخرى مثل (DRP) و (DGI) بأطوال تشذير أصغر.

تؤثر عدة عوامل ، بالإضافة إلى نوع التشذير ، على أداء شفرة التوربو ، بما في ذلك كثير حدود التوليد ، وعدد جولات فك التشفير ، وحجم الكتلة. تؤدي زيادة هذه المعلمات إلى تحسين السرعة ، ولكنها تزيد من وقت المحاكاة.

الكلمات الرئيسية: التشذير، الشفرة التوربينية، التسلسل الموازي، التبادل، الشفرات التلافيفية، الشفرة المنهجية العودية، وحدة فك الترميز التكرارية، خوارزمية MAP، معدل الخطأ الثنائي، نسبة الإشارة إلى الضوضاء، QPP،ARP ، DRP DGI, UNIF, RI

Dedication

This dissertation is dedicated to Nour Elislam Dharaoui, who has been no longer of this world, but his soul will stay forever with me

To my mother, father

My sisters and my brother

For all the love and the undying support that led me to this moment

To my friends

Adel, Chaouki, Sif Eddine, Said, Alaa Eddine, Mahmoud, Hocine, Salah Eddine,

Zaki, Hadjaissa ….

I will always appreciate all what you have done for me

having you by my side is a true blessing

Also, a special thanks to Reaman for everything

Abdelhafid

Dedication

I would like to dedicate this humble work to my dear father "Oussama",

For his patience and all the sacrifices he has made for me to reach this point.

To my mother "Djamilla",

That no dedication can ever express what I owe her.

To my dear sisters, Syrine, Sally and Selma for their continuous support for me through this journey.

To my dear cousin Billal who has stood with me throughout my entire life.

To my friends Abdelhafid, Hocine, Salah, Yassin, Abdesslam, Mounir, Lamine, Bilal and everyone whom I have met through my university years.

To my online and art friends who has always helped me improve as a person and find my hobby and passion, Majed, Miyuki, Remy, Rony, Hayato, Hana, Kaorin, Husain, Saso, Heba, Sula and everyone who have always been such a huge source of inspiration to me and always pushing me to get better.

Thank you to all my teachers,

To whom I love and all those who love me, I dedicate this work

Mahmoud

Acknowledgment

First and foremost, we thank Almighty God Allah for letting us live to see this dissertation through.

To everyone who has been a part of this project and everyone who helped us complete this work.

We are deeply grateful to Dr. S. Chellali for standing by us and guiding us to until the end, for his amazing skills for supervising us to try and perfect our work.

We thank the members of the jury for agreeing to examine this project.

Special thanks to all the teachers and doctors who have thought us throughout the years to reach the knowledge we currently have.

Finally, we thank all of our families and friends for their endless support and for all the hard work they have done to make sure we achieve this advanced stage of success, we will forever be grateful to all they have done for us.

Table of content

List of content

List of Figures

List of tables

Abbreviations:

ARP: Almost Regular Permutation

AWGN: Additive White Gaussian. Noise

BCJR: Based on the researchers Bahl, Cocke, Jelinek et Raviv

BER: Bit Error Rate.

BPSK: Binary Phase Shift Keying

DGI: Dithered Golden Interleaver

DRP: Dithered Prime relative interleaver

DVB-S: Digital Video Broadcasting

FEC: Forward error correcting

LDPC: Low Density Parity Check

LLR: Log-Likelihood Ratios

TC: Turbo Code

VLSI: very large scale integrated

VA : Viterbi Algorithm

BCH: Bose-Chaudhuri-Hocquenghem

RS: Reed-Solomon

ETDI: European Telecommunications Standardization Institute

AWGN: Additive White Gaussian Noise

IR: Information Retrieval

POE: probability of error,

QAM: Quadratic Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

BPSK: Binary phase-shift keying

ECC: Error-correction coding

CC: Convolutional Codes

RSC: Recursive Systematic Convolutional

PCE: Parallel Concatenated Encoding

SOVA: Soft-Output Viterbi Method

MAP: Maximum A-Posteriori

CPM: continuous phase modulation

PP: Permutation Polynomial

S-Random: Semi-Random

LTE: Long Term Evolution

General Introduction

Data processing and storage equipment has inevitably evolved towards using digital approaches since the introduction of high-speed logic circuits and very large-scale integration (VLSI). Data is represented in digital systems as strings of zeros and ones, which correspond to the on and off states of semiconductor switches. This has resulted in significant changes in the way information is handled. While most real-world data are in some sort of "analog form," the revolution in digital processing requires that this analog data be encoded into a digital representation, such as a string of ones and zeros.

The methods of converting from analog to digital and back have become commonplace. Digital voice, image, and video encoding and rendering are examples, as are the many other ways of recording and expressing data that we encounter in our current internet-based lifestyles.

In communications, digital data is processed differently than analog data. A communications receiver no longer needs to search for an analog signal and make a "best" estimate; instead, it just needs to choose between a finite number of discrete signals, such as a one or a zero in the most basic example. In a noisy communications environment, digital signals are more dependable; they can typically be identified properly as long as the noise levels are below a particular threshold. This allows us to recover digital data and fix faults in transmission using error-correcting algorithms.

Error control coding was originally used in deep-space communications, when low-power communications channels with almost infinite bandwidth are encountered. Convolutional codes are utilized on these data lines, along with sequential and Viterbi decoding (Both convolutional codes and Viterbi decoding algorithms are discussed ahead in the dissertation), and turbo coding will be utilized in the future. The compact disk player, which uses strong Reed-Solomon codes to handle the raw error probability from the optical reading device, which is too great for high-fidelity sound reproduction without error correction, was the next successful use of error control coding. Another obstacle overcome was the effective application of error control to bandwidth-limited telephone channels, where trellis-coded modulation was employed to achieve substantial gains and push transmission rates towards the channel's theoretical limit. Satellite communications, teletext transmission, computer storage devices, logic circuits, semiconductor memory systems, magnetic recording systems, audio-video systems, and Wi-Fi networks all use coding nowadays.

General Introduction

Turbo coding are a class of high-performance FEC (Forward Error Correction) codes that were the first practical methods to approach the Shannon limit (Maximum Channel Capacity).

The history of channel coding, also known as Forward Error Correction (FEC) coding, may be traced back to Shannon's pioneering work in 1948, which predicted that by adding redundant information to transmitted messages, arbitrarily reliable communications could be achieved. Shannon, on the other hand, did not provide specific channel coding techniques for practical implementations. Furthermore, he did not identify the greatest delay that might be allowed in order to communicate near the Shannonian limit, despite the fact that the amount of redundancy added grows as the related information delay grows. In recent years, academics have been working to lower the amount of delay that must be accepted, for example, by the interleaver of a turbo codec, in order to achieve a particular goal performance. The single error correcting Hamming code, a block code introduced in 1950, was historically one of the first viable FEC codes. Convolutional FEC codes were found by Elias in 1955, and Wozencraft and Reiffen, as well as Fano and Massey, suggested several decoding techniques. Viterbi's creation of a maximum likelihood sequence estimation method in 1967 was a watershed moment in the development of convolutional error correcting coding. Forney's frequently cited study provides a classic understanding of the Viterbi Algorithm (VA). Heller and Jacobs suggested one of the earliest practical implementations of convolutional codes in the 1970s.

FEC codes were first used in different deep-space and satellite communications systems in the early 1970s, and by the 1980s, they were widely used in practically all cellular mobile radio systems. FEC codes and modulation, on the other hand, have been addressed as separate issues in communication systems for a long time.

Turbo codes were considered to be selected during the study phase of LTE mobile communications to further enhance the performance of the new generation standard that had packet data support with rates up to a 100 Mbps on the downlink and a revolutionary 50 Mbps on the uplink, a very low latency of 10 milliseconds layer-2 round trip delay, flexible bandwidths that could go all the way up to 20 MHz, it had an efficient VoIP support and improved system capacity and coverage.

We will be looking at older versions of interleavers like DRP and ARP interleavers and then study the QPP interleavers leading to improved distance spectrum.

Chapter I:
Turbo Coding

Chapter I: Turbo Coding

I – 1- Introduction

Glavieux, Berrou and Thitimajshima first introduced Turbo codes in 1993; it shown to have a near Shannon limit error correction capability, which was revolutionary in communication at the time. Since then, turbo codes have been interesting to research and study among the coding community. Now after decades of its introduction, they have become a mature technology that quickly adopted for applications in many transmission systems.

Turbo coding are a class of high-performance FEC (Forward Error Correction) codes that were the first practical methods to approach the Shannon limit (Maximum Channel Capacity).

Turbo codes were considered to be selected during the study phase of LTE mobile communications to further enhance the performance of the new generation standard that had packet data support with rates up to a 100 Mbps on the downlink and a revolutionary 50 Mbps on the uplink, a very low latency of 10 milliseconds layer-2 round trip delay, flexible bandwidths that could go all the way up to 20 MHz, it had an efficient VoIP support and improved system capacity and coverage.

I-1-1- Overview on Information Theory

In 1940, Claude Shannon presented the main theorems and concepts of what is noted as information theory [1].

Information theory is the study of how information is transmitted, processed, extracted, and used. Information may be conceived of as the resolution of uncertainty in a broad sense. In the case of information communication over a noisy channel, Claude Shannon formalized this abstract concept in his paper A Mathematical Theory of Communication in 1948, in which information is thought of as a set of possible messages, and the goal is to send these messages over a noisy channel and have the receiver reconstruct the message with a low probability of error, in s Shannon's primary conclusion, the noisy-channel coding theorem, demonstrated that, in the limit of many channel uses, the asymptotically feasible rate of information is equal to the channel capacity, a number that is only reliant on the channel statistics over which the messages are transmitted [2].

Information theory attempts to analyze communication between a transmitter and a receiver over an unreliable channel, and this approach analyzes information sources, particularly the amount of information produced by a given source, on the one hand, and states the conditions for performing reliable transmission over an unreliable channel on the other.

In this theory, there are three basic concepts:

1. The definition of a quantity that may be used to measure information and is compatible with a physical knowledge of its qualities is the first.

2. The link between the information and the source that creates it is the subject of the second notion. The term "source information" will be used to describe this idea. This notion is connected to well-known information theory methods like compression and encryption.

3. The interaction between the information and the unstable route via which it will be delivered is the subject of the third notion. As a result of this approach, a critical metric known as channel capacity is defined. This notion is strongly connected to the well-known information theory approach of error-correction coding.

A practice termed coding, which is meant to maximize transmission and make optimum use of the capacity of a particular channel, is one of the most widely used methods in information theory [1].

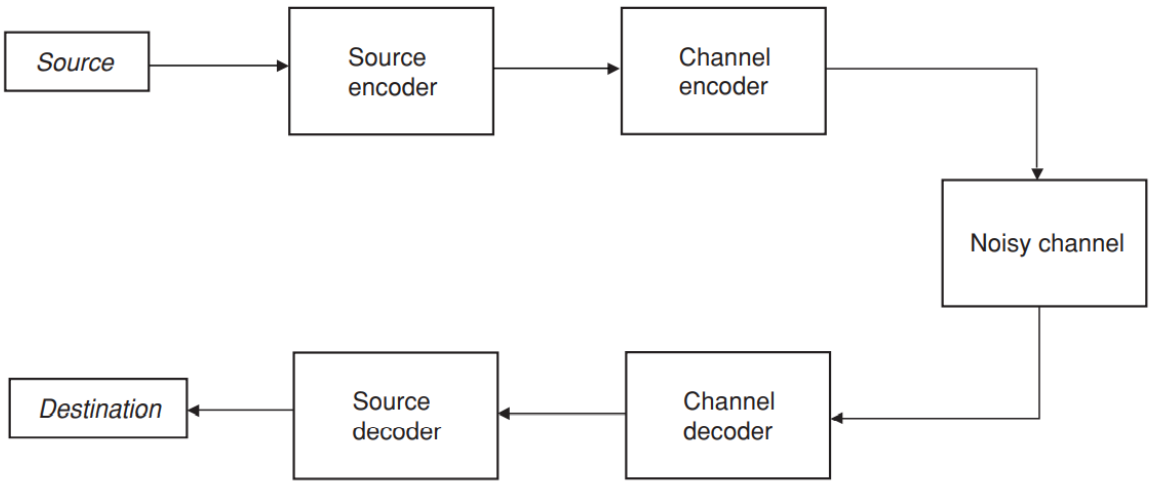


Figure 1: Basic Communication System

I – 2 The Shannon Limits on Performance

Shannon, Hamming, and Golay laid the groundwork for today's approach to error control in digital communications. While Shannon proposed a theory that outlined the basic constraints of communication system efficiency, Hamming and Golay were the first to create practical error control strategies. A new paradigm emerged, one in which faults are not synonymous with data that is irreversibly lost; rather, faults may be repaired or avoided entirely by creative design. This new way of thinking was groundbreaking. Despite the fact that Shannon's theory predicted significant increases in communication system performance, real advances had to be unearthed over the course of a half-century of rigorous study. One explanation for this is because Shannon's theory has a fundamental flaw. While it explicitly identifies theoretical communication efficiency limitations, its technique offers no guidance on how to actually accomplish these limitations since it is predicated on clever averaging arguments that ignore all system structure. Coding theory, on the other hand, has grown into a thriving field of practical mathematics as a result of Hamming and Golay's work [3].

To first understand why we had to invent something like turbo coding; it would be easier to first know what the motive behind it is. Shannon posed a question about how much coding gain can we have for a given communication system when used over an AWGN (Additive White Gaussian Noise) channel and answered it in the famous "A Mathematical Theory of Communication" by defining the *Capacity* of a channel

$$\mathrm{C} = \mathrm{W}\, log_2\left(1 + \frac{Es}{N0}\right) \text{bits per seconds}$$

Equation 1

C is the channel capacity, or the maximum number of bits that may be communicated across this channel per unit time (second), W is the channel bandwidth, and S/N is the receiver's signal-to-noise power ratio. Shannon's primary theorem, which goes along with (1.1), states that as long as the transmission rate R over the channel (in bits/second) is less than the channel capacity C, error probabilities as minimal as desired may be obtained. This may be accomplished by using the proper encoding and decoding techniques. Shannon's theory, on the other hand, is mute on the construction of these encoders and decoders [3].

The transfer of information from one source to another over a medium known as the communication channel is how a source and a destination communicate. The conditional

probability matrix formed between the input and the output is used to appropriately describe communication channels, allowing us to assess the dependability of the information arriving at the receiver. The Shannon capacity theorem proves that with a sophisticated coding scheme, it is feasible to have error-free (reliable) communication via a noisy (unreliable) channel, as long as the transmission rate is limited to a value less than or equal to the channel capacity. This theorem imposes a limit on the transmission rate of the communication, but not on the communication's dependability.

The Noisy-Channel Coding Theorem states that it is possible to communicate over a noisy channel with very small chances of error as long as the rate of communication is kept below the channel capacity explained in the equation above.

One of the Shannon theorems, the channel coding theorem, connects the source information measure, the channel capacity measure, and coding: If the information rate of a given source does not exceed the capacity of a given channel, then there exists a coding technique that allows transmission through this unreliable channel with an arbitrarily low error rate r [1].

I-2-1- Hamming Distance

The Hamming distance is a measurement of how much two-bit strings of the same dimension vary from one another. The Hamming distance was first used in error-correcting coding theory where it quantified the error caused by noise across a channel while a message, usually a sequence of bits, was transmitted between its source and destination. In the Information Retrieval (IR) context, bit sequences may also be found.

The only thing that matters in a conventional Hamming distance application is that the corresponding bits in two strings match. More nuanced differences are needed in a full text database, though. Bitmaps, for example, may be used to indicate the units (sentences, paragraphs, etc.) in which a term appears within a single document; in this context, it is troubling that two terms that tend to occur "closely" to one another, even if not in exactly the same units, are assessed by the Hamming distance in the same way as terms that are completely unrelated to one another. To put it another way, the original Hamming distance is blind to the concept of adjacent units.

I-2-2- Bit Error Rate (BER)

The bit error rate (BER) is the rate at which digital data is sent with mistakes. BER is used to measure a channel transporting data by measuring the rate of mistakes in a data string and is affected by a number of FACTORS. Telecommunications, networks, and radio systems all utilize it.

Radio data connections, fiber optic communication systems, Ethernet, and any system that sends data across a network in any manner where noise, interference, and phase jitter may degrade the digital signal are all examples of systems where bit error rate is relevant.

While there are significant variations in how these systems operate and how bit error rate is impacted, the fundamentals of bit error rate remain the same.

There is a chance that mistakes will be introduced into the system when data is sent via a data connection. If mistakes are introduced into the data, the system's integrity may be jeopardized. As a consequence, it's important to evaluate the system's performance, and the bit error rate, or BER, is an excellent method to do so.

Unlike many other types of evaluation, bit error rate (BER) evaluates a system's whole end-to-end performance, including the transmitter, receiver, and the medium that connects them. In this manner, the BER allows the real performance of a system in operation to be evaluated, rather than just evaluating the component components and assuming they would work well once installed.

A bit error rate is defined as the rate at which errors occur in a transmission system, as the name suggests. This may be immediately converted into the number of mistakes in a string of a certain length. The definition of bit error rate may be expressed in the following way:

$$BER = \frac{Number\ of\ Errors}{Total\ number\ of\ bits\ sent}$$

Equation 2

If the medium between the transmitter and receiver is excellent and the signal to noise ratio is strong, the bit error rate will be extremely low - maybe negligible and have no impact on the entire system. However, if noise is found, the bit error rate may need to be addressed.

The primary causes of a data channel's deterioration and the resulting bit error rate, BER, are noise and changes in the propagation route. Both effects are random, with the noise using a Gaussian probability function and the propagation model using a Rayleigh model. This implies that statistical analytic methods are often used to analyze channel properties.

Bit errors in fiber optic networks are mostly caused by flaws in the components that make up the connection. Optical drivers, receivers, connections, and the fiber itself are among them. Bit mistakes may also be created due to optical dispersion and attenuation. Noise may also be injected into the optical receiver. These are often photodiodes and amplifiers that must react to extremely tiny changes and, as a consequence, may have high noise levels.

Any phase jitter in the system, which may change the sampling of the data, is another contributing cause for bit mistakes [4].

I-2-3- BER and Eb/No

Radio connections and radio communications systems are more closely linked with signal to noise ratios and Eb/No numbers. The bit error rate, or BER, may alternatively be expressed in terms of probability of error, or POE. Three additional factors are utilized to determine this. They are the error function, erf, the energy in a single bit, Eb, and the noise power spectral density, No.

It should be noticed that the error function has a distinct value for each kind of modulation. Because each kind of modulation behaves differently in the presence of noise, this is the case. Greater order modulation systems, which may transport higher data rates (e.g. 64QAM, etc.), are less resilient in the presence of noise. Lower-order modulation schemes (such as BPSK, QPSK, and others) have lower data speeds but are more reliable.

The energy per bit, Eb, is a unit of energy that may be calculated by dividing the carrier power by the bit rate. It is measured in Joules. No is a power per Hertz, thus this has power (joules per second) divided by seconds as its dimensions. When you look at the dimensions of the ratio Eb/No, you'll see that they all cancel out, resulting in a dimensionless ratio. It's essential to remember that POE is a signal-to-noise ratio that is proportional to Eb/No.

I-2-4- Factors affecting the Bit Error Rate

Using Eb/No, it is clear that the bit error rate, or BER, may be influenced by a variety of variables. It is feasible to optimize a system to achieve the necessary performance levels by

changing the factors that can be controlled. This is usually done during the design phases of a data transmission system so that performance parameters may be adjusted during the early stages of the design idea.

- *Interference:* The levels of interference in a system are usually determined by external sources and cannot be altered by system design. The system's bandwidth, on the other hand, may be adjusted. The amount of interference may be decreased by decreasing the bandwidth. Reduced bandwidth, on the other hand, restricts the amount of data that can be sent.
- *Raise transmitter power:* It is also feasible to increase the system's power level to increase the power per bit. This must be weighed against other considerations such as the effect of increasing the power output on the size of the power amplifier, total power consumption, and battery life, among others.
- *Modulation at a lower level:* Lower order modulation methods may be employed, but data throughput suffers as a result.
- *Reduce bandwidth:* Reducing bandwidth is another way to lower the bit error rate. As a result, there will be less noise, and the signal-to-noise ratio will improve. Again, this reduces the amount of data that can be processed.

I-3- Error Detection and Correction (Channel Coding)

Almost every kind of information transfer, storage, and processing system currently employs error-control codes. Rapid advancements in electrical and optical devices and systems have made it possible to design very strong codes with near-optimal error-control performance [1].

When digital data is transferred across a noisy channel, there is always the possibility of mistakes in the received data. The user usually sets an error rate over which the data received is no longer useable. Error-correction coding may frequently be used to decrease mistakes to a level where they may be tolerated if the incoming data does not match the error rate criterion. The use of error-correction coding to solve this sort of issue has grown more common in recent years. Shannon's work revealed the usefulness of coding. " In 1948, he demonstrated that with proper encoding and decoding, communication over a noisy channel with an error probability as small as desired is possible if the data source rate is less than a quantity called the channel capacity. Shannon's work essentially asserts that signal strength, channel noise, and available bandwidth restrict communication pace rather than accuracy. The true limit on communication

rate was soon discovered to be the cost of coding scheme implementation, not channel capacity. Due to financial restrictions, communication is limited to a fraction of its potential. In recent years, a lot of effort has gone into developing efficient and practical coding systems for a variety of noisy channels. The majority of progress toward practical schemes has occurred in the last fifteen years, and it is now clear that coding can significantly improve performance in a variety of applications. There have been a number of successful instances where coding equipment has been constructed and utilized. New developments in the field of error-correcting codes, as well as dramatic reductions in the cost and size of solid-state electronic devices, have increased the practicality of coding [5].

Error-correction coding (ECC) is a signal processing method that is used to increase the dependability of digital communication. Individual coding systems may take many different shapes and have origins in a variety of mathematical areas, but they always have two things in common. The utilization of redundancy is one example. Extra or superfluous symbols are always included in coded digital transmissions. These symbols are used to emphasize each message's individuality. They're always selected to make it very improbable that a channel disruption would corrupt enough of a message's symbols to render it unusable. Noise averaging is the second component. The redundant symbols are made to rely on a span of many information symbols to achieve this averaging effect. Examining each of these parts independently may provide useful insight into the coding process [5].

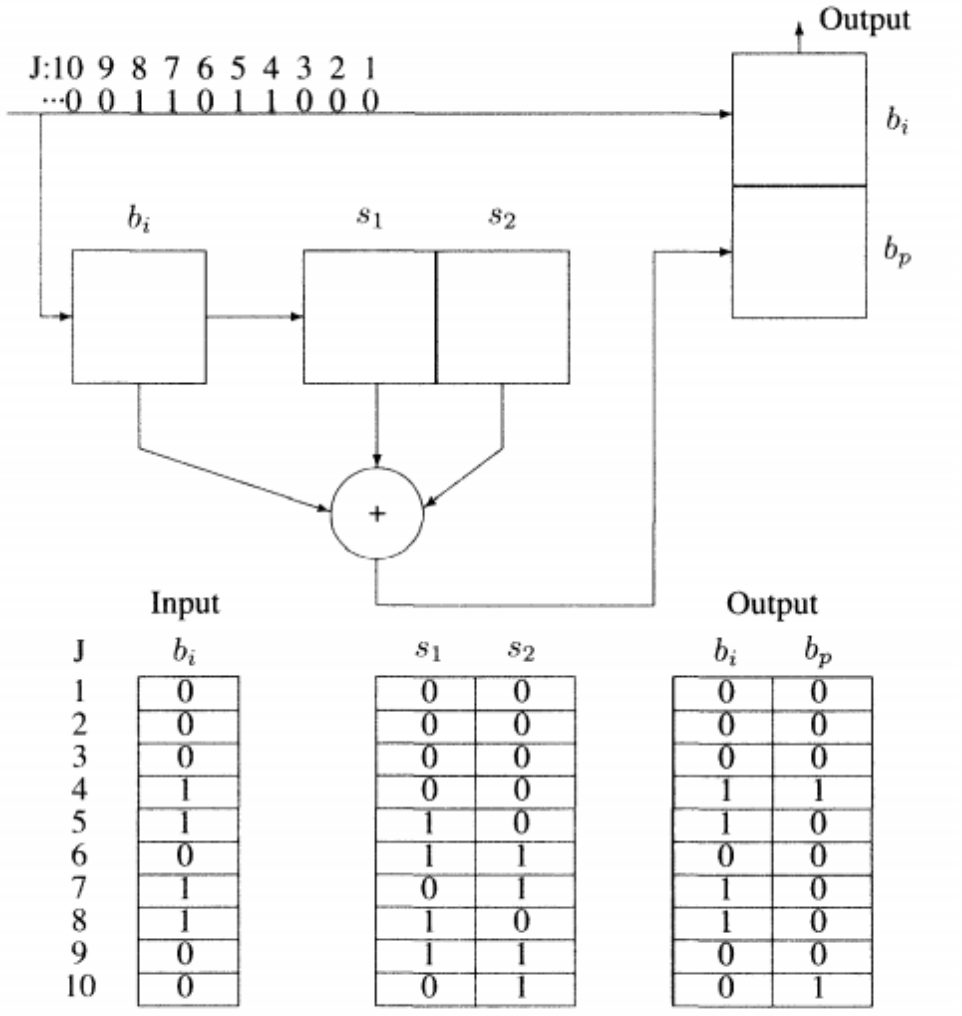
I-4- Convolutional Encoding

Both block codes and Convolutional Codes (CCs) may be classed as systematic or nonsystematic codes, with systematic codes implying that the original information bits or symbols are part of the encoded codeword and can therefore be identified clearly at the encoder's output. Their encoders are commonly built using linear shift-register circuitries, as shown in Figure 2. After presenting some of the fundamental convolutional coding parameters, the figure will be examined in further detail. In general, a k-bit information symbol is input into the encoder, which is made up of K shift-register stages. The equivalent two shift-register stages in our example of Figure 2 are s1 and s2. The constraint length of the code is often defined as the number of shift-register stages K. This code may also be referred to as a memory three code, suggesting that the CC's memory is determined by K + 1. The next state of this state machine is determined by the present shift-register state s1 s2 and the incoming bit bi. R = k/n denotes the amount of output bits, whereas n is the coding rate, suggesting that R 1. To properly explain

the code, we must additionally give the generator polynomial, which represents the architecture of the modulo-2 gates that generate the convolutional encoder's output bits. n generator polynomials are required to generate n bits. A CC is designated as a CC (n, k, K) scheme in general, and the code is completely described given the n generator polynomials. Once a bit enters the encoder's shift register in Figure 2, it must traverse the register, therefore the sequence of state changes in the register is not random. In addition, the modulo-2 gates provide additional limits on the output bit-stream. Because of these limits, legal transmitted sequences are limited to certain bit patterns, and if transmission faults occur, the decoder concludes that the encoded sequence could not have been created by the encoder and must be due to channel problems. In this situation, the decoder will try to find the most closely related lawfully encoded sequence and output the matching bitstream as the decoded text.

We'll show the encoding and decoding principles in the context of the systematic code given as (k = 1), half-rate (R — k/n = 1/2), CC (2,1,2), with three binary phases of memory (K — 2).

Before the information bits are entered to the shift register, it is normally cleared by setting it to an all-zero state at the start of the encoding [6].



J	b_i	s_1	s_2	b_i	b_p
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	1	0	0	1	1
5	1	1	0	1	0
6	0	1	1	0	0
7	1	0	1	1	0
8	1	1	0	1	0
9	0	1	1	0	0
10	0	0	1	0	1

Figure 2 Systematic half-rate, constraint-length two convolutional encoder (7(7(2.1,2).

We shall explain the encoding and decoding principles in the context of the systematic code given as (k = 1), half-rate (R — k/n = 1/2), CC (2,1,2), with three binary phases of memory (K — 2).

Before the information bits are entered to the shift register, it is normally cleared by setting it to an all-zero state at the start of the encoding [6].

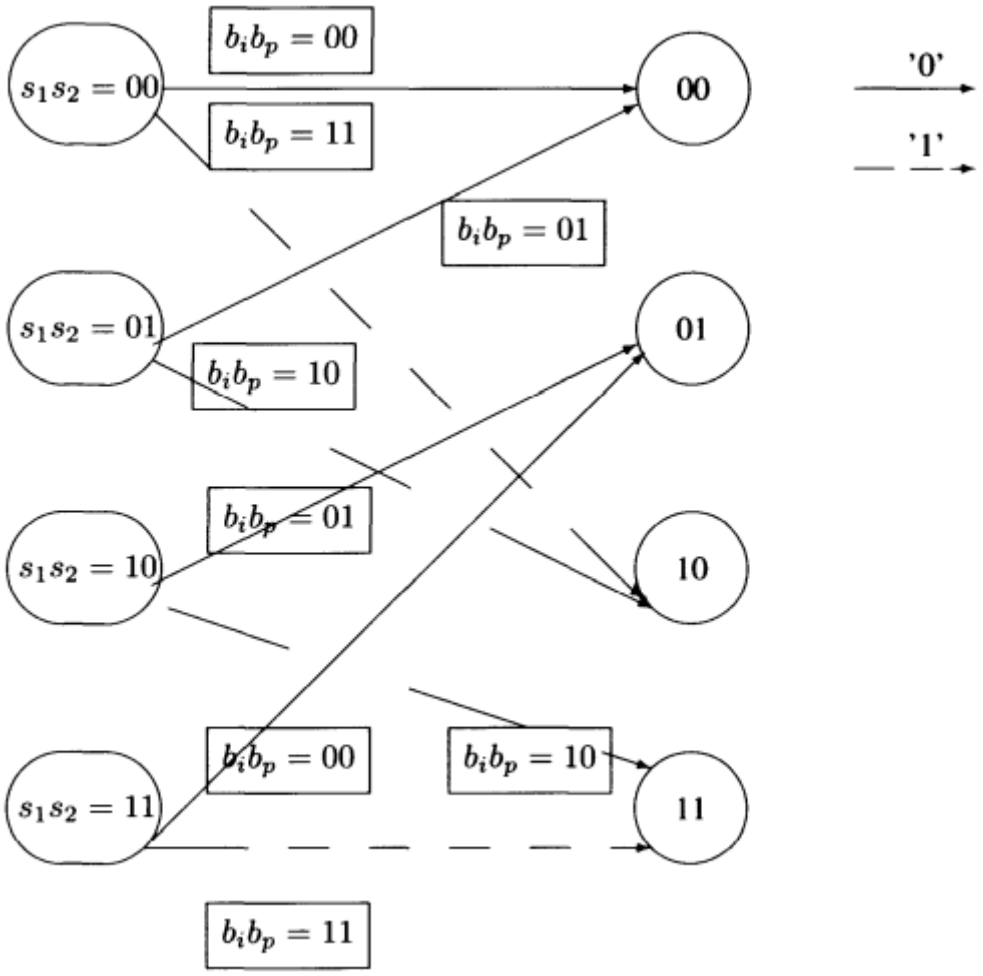


Figure 3: State transition diagram of the CC(2,1,2) systematic code, where broken lines indicate transitions due to an input one, while continuous lines correspond to input zeros

I-5- State and Trellis Transitions

The state transition diagram in Figure 4 is a commonly used methodology for describing the operations of a state machine, such as our convolutional encoder. There are four potential states in the state machine when there are two bits in the shift register at any one time, and the state transitions are determined by the incoming bit bi. In the diagram, a state transition caused by a logical zero is represented by a continuous line, whereas a state transition caused by a logical one is shown by a broken line. The encoder's intrinsic restrictions are shown here in the fact that, depending on the binary input bit, there are only two acceptable state transitions from any state. Similarly, there are two merging pathways in each state. A logical one input results in a transition to (1,1) from state (s1, s2) = (U) in the encoder circuit of Figure 4, while an input zero results in state (s1, s2) = (U) (0,1). The reader may also easily verify the remaining transitions. In addition, the related encoded output bits are represented in the boxes associated

with each of the transitions in this diagram. As a result, this figure accurately depicts the encoder's activities [6].

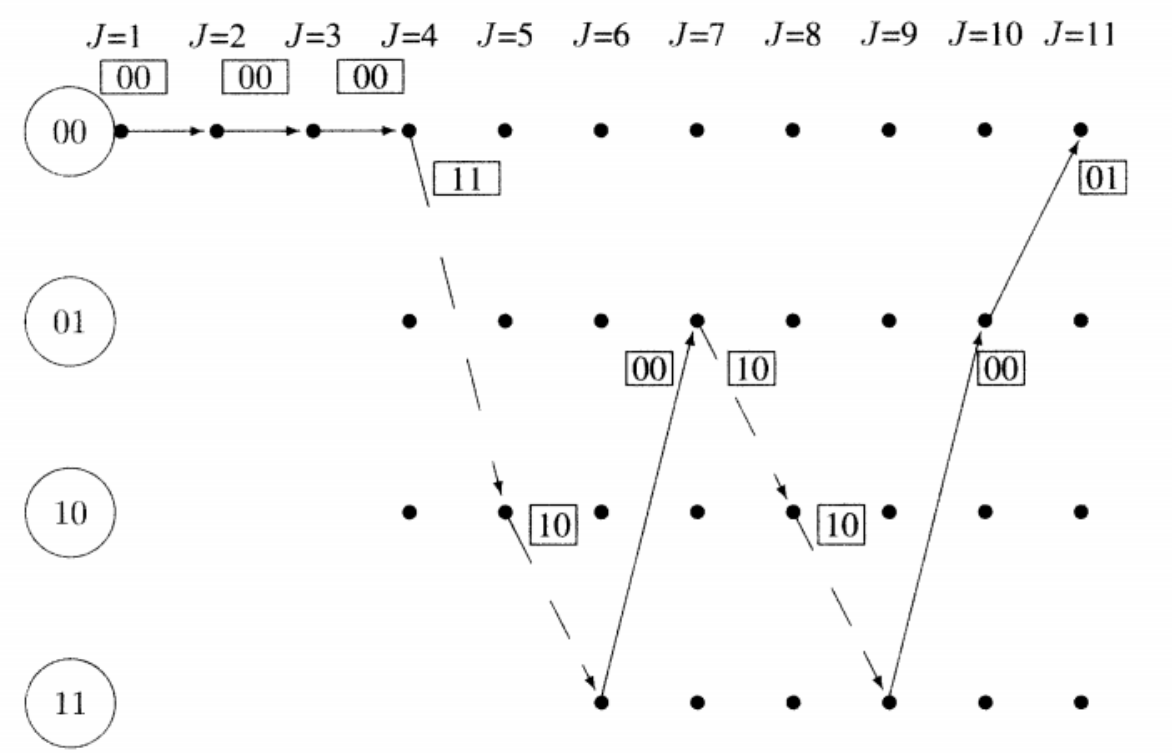


Figure 4: Trellis-diagram of the CC (2,1, 2) systematic code, where broken lines indicate transitions due to a binary one, while continuous lines correspond to input zeros.

Since QPP is linked and used a lot in LTE and mobile communications then it would make sense to take a look at the Trellis diagram of a 3gpp Turbo Code.

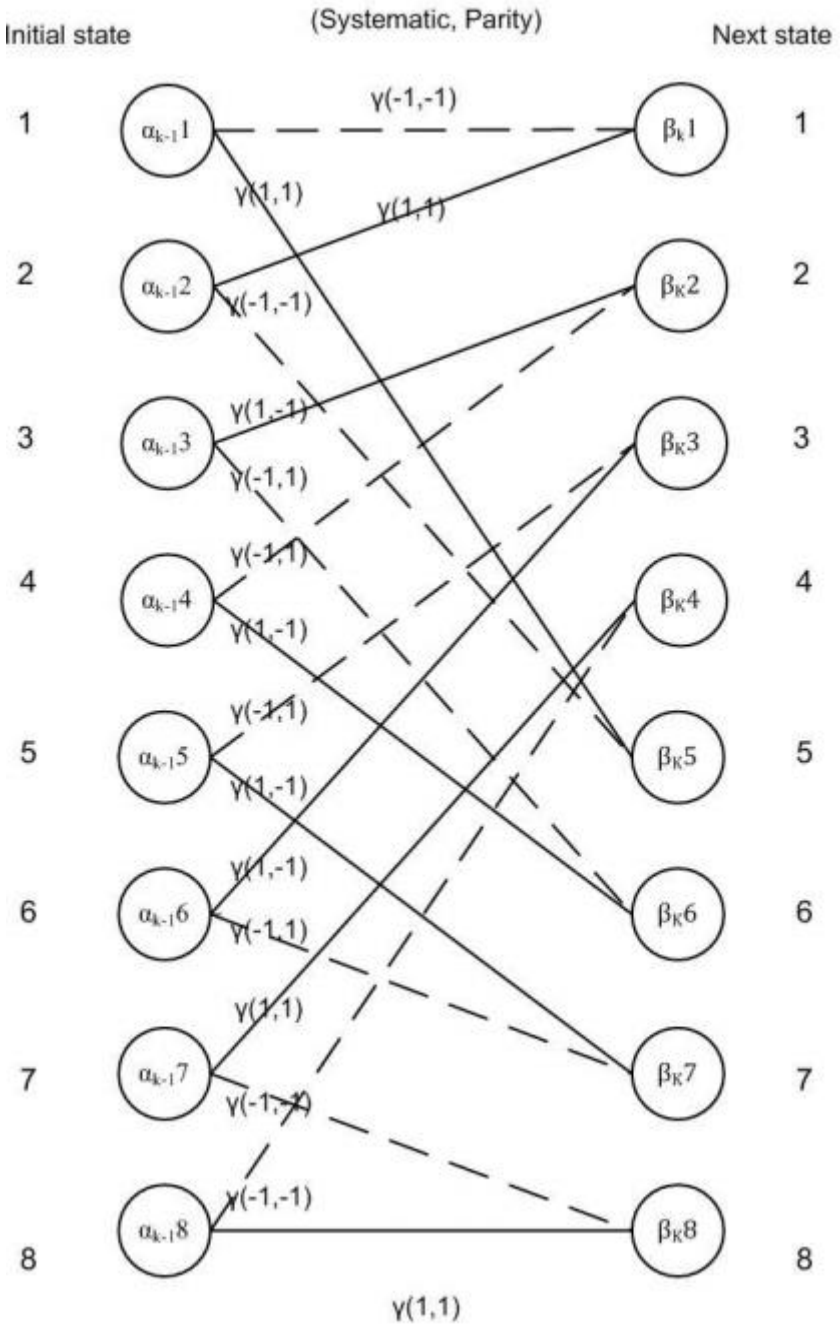


Figure 5: Trellis of 3GPP turbo code.

I-6- The Viterbi Algorithm

I-6-1- Error-free Hard-decision Viterbi Decoding

The decoder must make the best approximation of the original encoded information sequence based on the incoming bit-stream. As a result, the previously indicated encoder limits on lawful bit sequences must be used in order to exclude illegitimate sequences and hence eliminate transmission mistakes. For the purpose of computational simplicity, suppose that the all-zero

bit-stream was sent and that the demodulator recognized the received sequence of Figure 6, which was then given to the FEC decoder. This procedure is known as hard-decision demodulation if the demodulator makes a binary judgment about the received bit. Soft-decision demodulation, on the other hand, occurs when the demodulator does not make a binary choice and instead produces a finely graded multilevel confidence measure about the likelihood of a binary one and a binary zero.

For the time being, only hard-decision demodulation will be considered. The decoder must now compare the received bits to all of the valid sequences in Figure 6's trellis diagram and quantify the likelihood of each of the associated routes, which lends a probability-related quantity to particular decoded sequences, as shown below.

Referring to Figure 6, we calculate the Hamming distance of the obtained two-bit symbol with regard to both of the valid encoded patterns of the trellis diagram for the first trellis section (i.e., for the first trellis transition), mentioning that the Hamming distance is given by the range of different bit positions between two binary sequences for example, the related Hamming distances for the initial two-bit received symbol 10 are 1 for both the 00 and 11 encoded sequences. As a result, the decoder is unable to say whether 00 or 11 was the most probable conveyed sign at this point. These Hamming distances are also represented at the top of the nodes corresponding to the new encoder states we arrived at as a result of the state transitions due to a logical one and zero, respectively, in the trellis diagram of Figure 6. In Viterbi decoding, these Hamming distances are referred to as the branch metric. The Viterbi decoding algorithm's strength comes from the fact that it uses maximum probability sequence estimate rather than symbol-by-symbol judgments, allowing it to take advantage of the encoder's limits on acceptable encoded sequences. As a result, before a judgment on the most probable encoder route and information sequence can be made, the branch metrics will be gathered across a number of successive trellis stages [6].

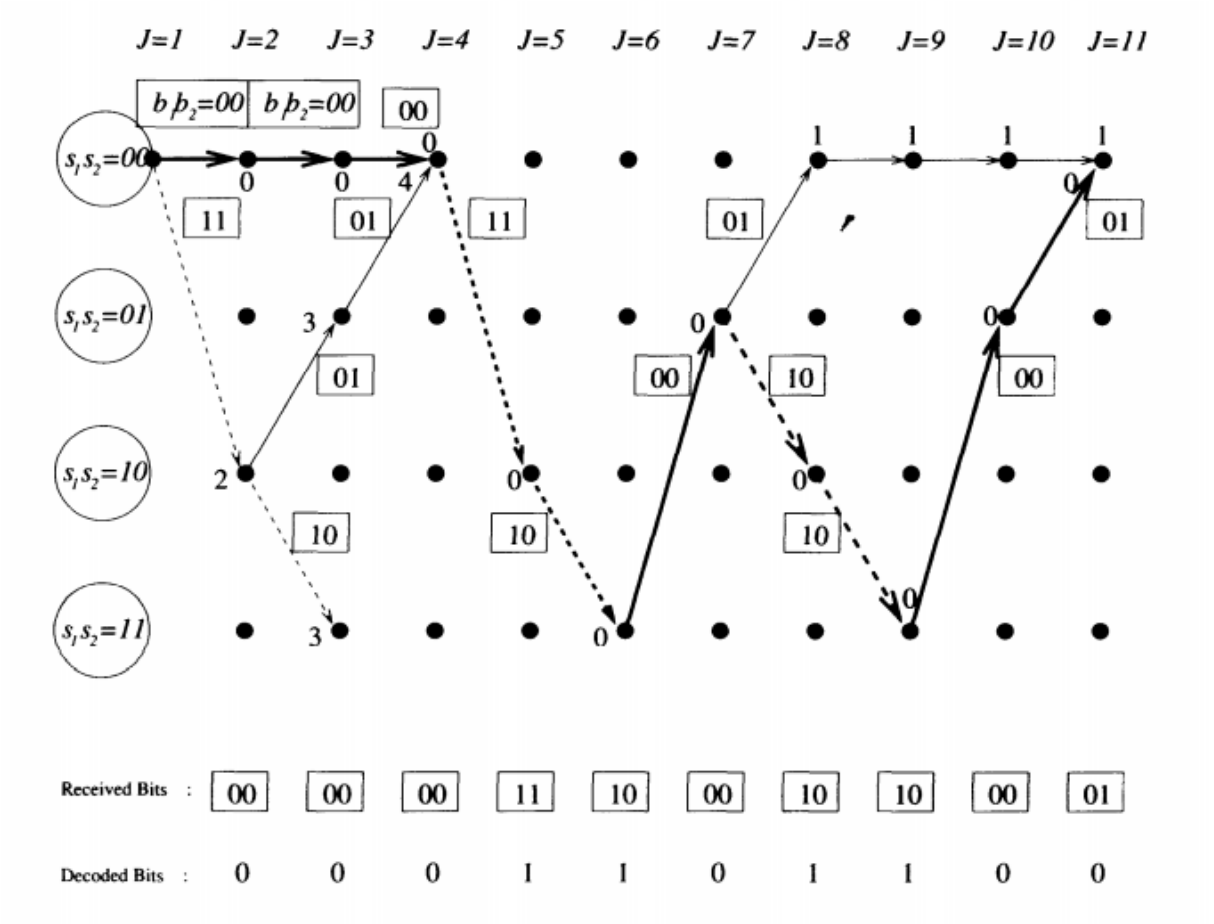


Figure 6: Trellis-diagram-based Viterbi decoding of the (7(7(2,1,2) systematic code, where broken lines indicate transitions due to an input one, while continuous lines correspond to input zeros — erroneous hard-decision decoding of burst errors.

I-6-2- Error-Free Soft-decision Viterbi Decoding:

Soft-decision demodulation is defined as a demodulator that does not make a binary choice and instead generates a finely graded soft-decision confidence measure relating to the likelihood of a binary one and a binary zero, respectively. We may use an eight-level soft-decision demodulator output as an example. This gives a better indication of whether the demodulator's judgment on a particular demodulated bit is high-reliability or low-reliability. This certainly gives the Viterbi decoder a lot more information than the binary zero/one choices before. As a result, as shown in the accompanying diagram, a greater error correcting capacity will be realized [6].

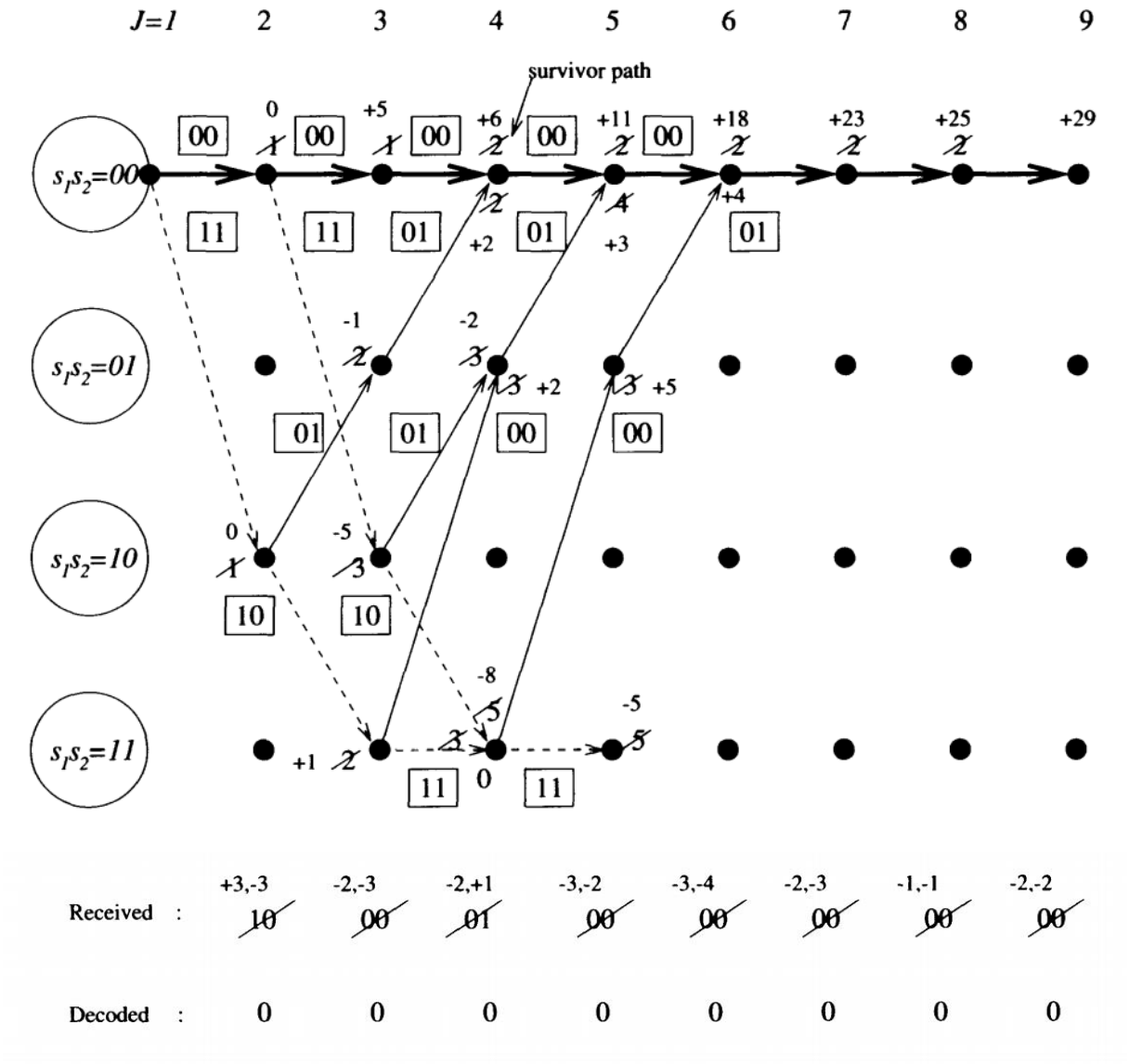


Figure 7: Trellis-diagram-based Viterbi decoding of the CC (2,1,2) systematic code, where broken lines indicate transitions due to an input one, while continuous lines correspond to input zeros — error-free soft-decision decoding of two isolated bit errors

The state transition diagram and trellis diagram were used to characterize the convolutional encoder. The traditional Viterbi algorithm was then taught in the context of a simple decoding example, with both hard- and soft-decision-based situations considered. In the next chapter, we'll look at the block coding family, which has a wide range of applications in both conventional and custom wireless communication systems. Bose-Chaudhuri-Hocquenghem codes have been used in a number of video systems described in this work.

I-7- Systematic Recursive Convolutional Encoders

Costello and Forney Jr. invented these kinds of encoders, which include feedback branches.

By feeding back one of the nonrecursive nonsystematic (conventional) convolutional encoder's encoded outputs to its input, the recursive systematic convolutional (RSC) encoder is produced. A traditional convolutional encoder is shown in Figure 8.

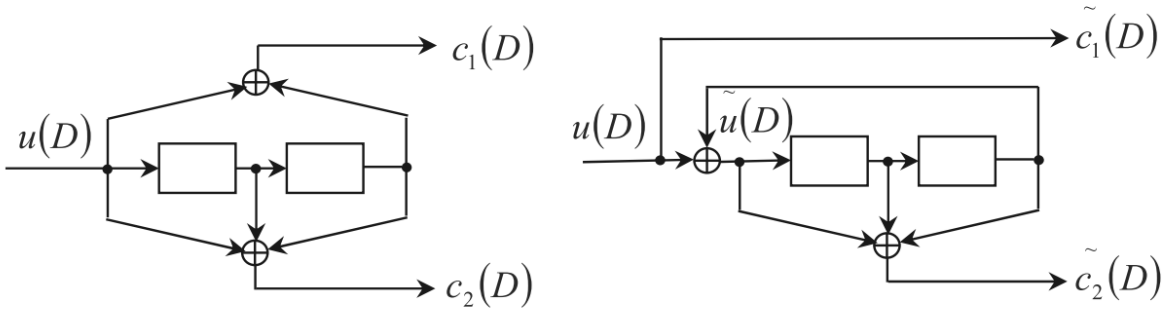


Figure 8: Convolutional encoder: a non-recursive non-systematic; b recursive systematic

The generator sequences g1 = [111] and g2 = [101] represent the standard convolutional encoder, which may be written in a more compact manner as G= [g1, g2]. G= [1, g2 / g1], where the first output (represented by g1) is fed back to the input, is the RSC encoder of this standard convolutional encoder. In the diagram above, 1 represents the systematic output, g2 represents the feedforward output, and g1 represents the feedback to the RSC encoder's input. The RSC encoder as a consequence is shown in Figure 9.

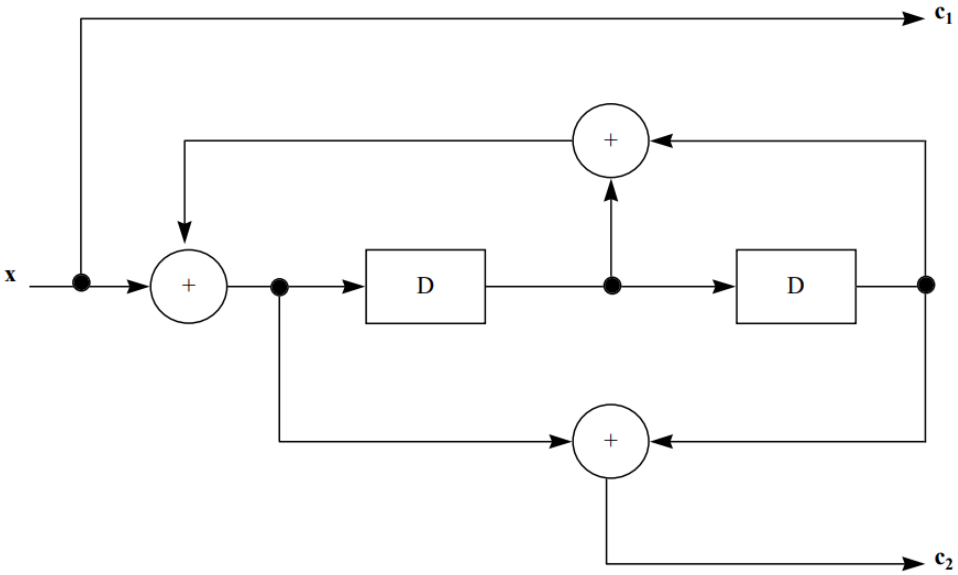


Figure 9: The RSC encoder with r=1/2 and K=3.

It was proposed that excellent codes might be produced by setting the RSC encoder's feedback to a basic polynomial, since the primitive polynomial produces maximum-length sequences, which provide unpredictability to the turbo code.

I-8- Trellis Termination

The trellis is ended in a typical convolutional encoder by adding m = K-1 extra zero bits after the input sequence. These additional bits drive the conventional convolutional encoder to the all-zero state (trellis termination). However, owing to feedback, this approach is not feasible for the RSC encoder. The extra termination bits for the RSC encoder are very unpredictable and dependent on the encoder's state. Furthermore, owing to the existence of the interleaver between the component encoders, even if the termination bits for one of the component encoders are discovered, the other component encoder may not be driven to the all-zero state with the same m termination bits. Figure 10 depicts a basic approach proposed in for overcoming this issue.

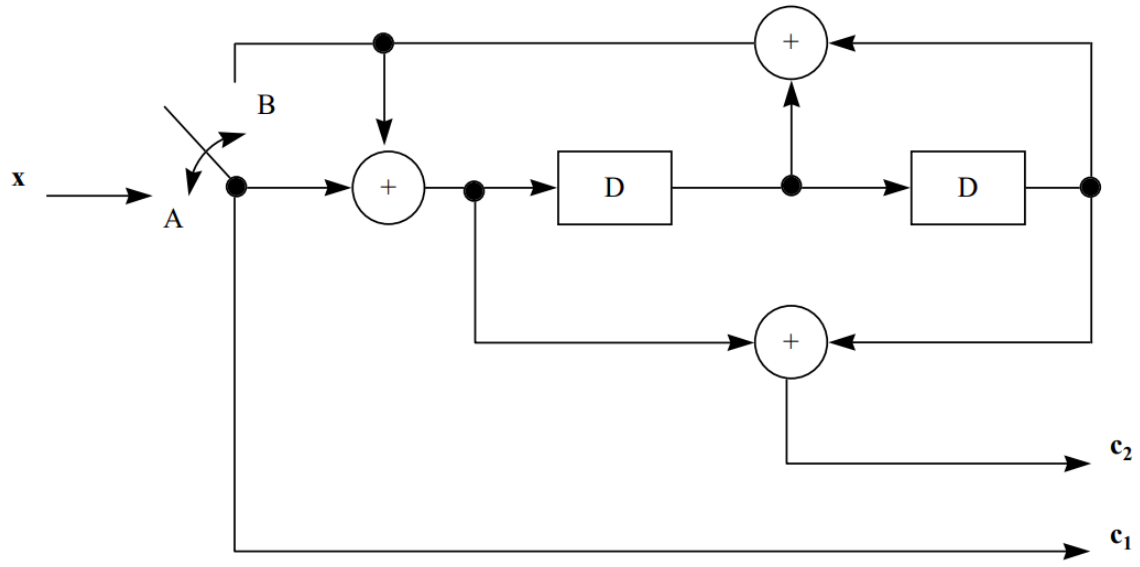


Figure 10: Trellis termination strategy for RSC encoder.

The switch is set to position A for encoding the input sequence, and it is set to position B for terminating the trellis.

I-9- Recursive and Nonrecursive Convolutional Encoders

An example is the easiest way to understand the differences between recursive and nonrecursive convolutional encoders. A basic nonrecursive convolutional encoder with generator sequences g1= [11] and g2= [10] is shown in Figure 11.

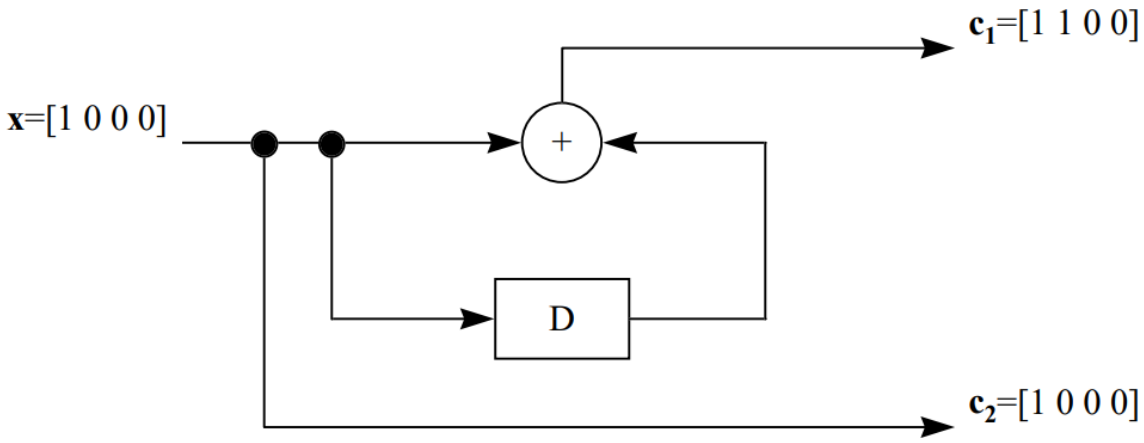


Figure 11: Nonrecursive r=1/2 and K=2 convolutional encoder with input and output sequences.

Figure 12 illustrates the G= [1, g2 / g1] comparable recursive convolutional encoder of Figure 11.

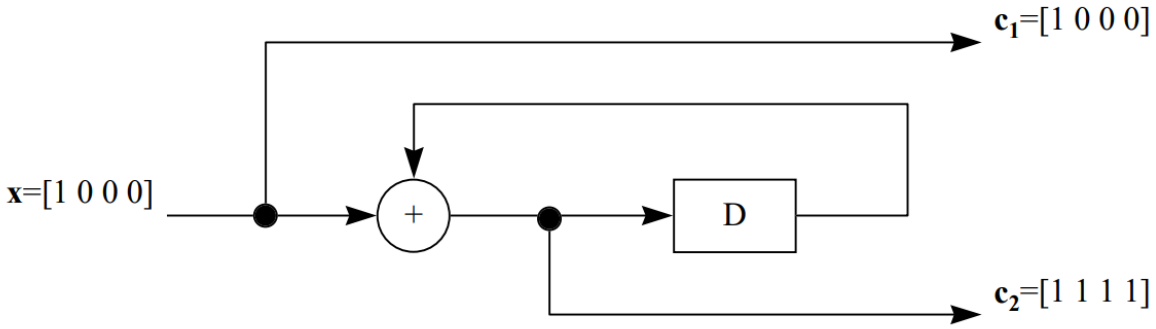


Figure 12: Recursive r=1/2 and K=2 convolutional encoder with input and output sequences.

Given the identical input sequence as in Figures 11 and 12, the nonrecursive encoder generates an output codeword with a weight of 3 while the recursive encoder generates an output codeword with a weight of 5. As a result, compared to a nonrecursive encoder, a recursive convolutional encoder produces codewords with higher weight. As a consequence, there are fewer codewords with lower weights, resulting in improved error performance. The primary reason for using RSC encoders as component encoders in turbo codes is to take use of their recursive character rather than their systematic nature.

The state diagram of the nonrecursive encoder is shown in Figure 13.

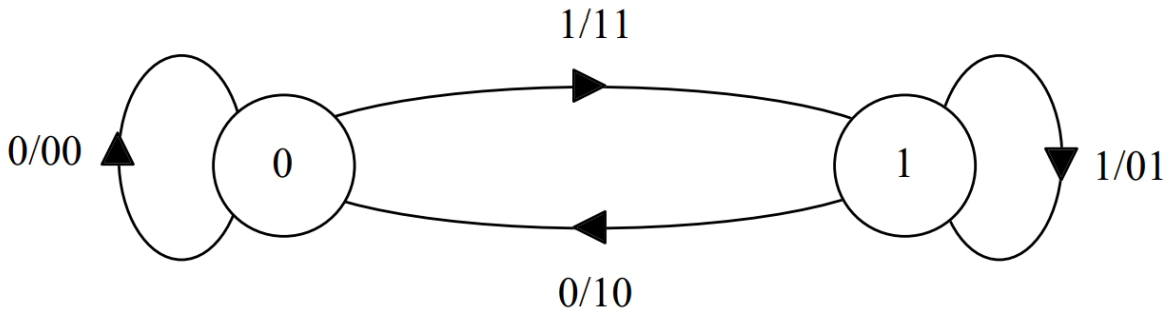


Figure 13: State diagram of the nonrecursive encoder.

The state diagrams of the recursive encoder are shown in Figure 14.

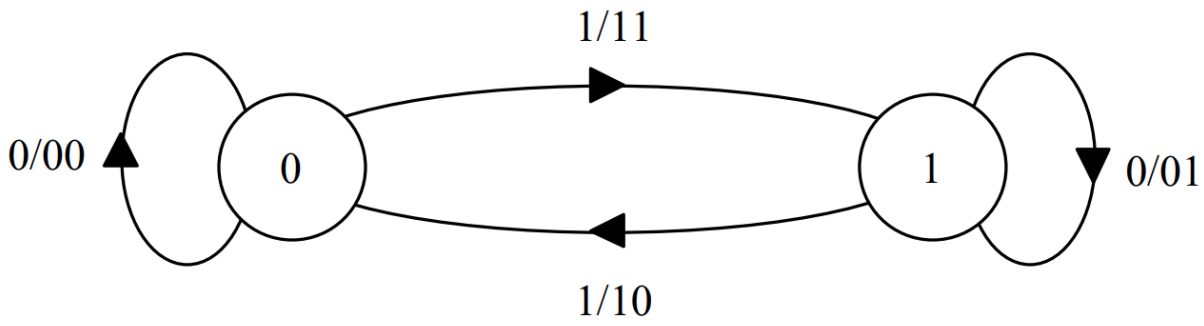


Figure 14 State diagram of recursive encoder in Figure 13.

Clearly, the state diagrams of the encoders are very similar. The transfer function for both encoders is identical and is found to be $T(D) = \frac{D^3}{1-D}$ where N and J are neglected. The two codes also have the same minimum free distance and the same trellis structure. As a result, the first event error probability is the same for both codes. These two codes, however, have distinct bit error rates (BERs). This is because BER is dependent on the encoders' input-output correspondence. At low signal-to-noise ratios Eb/No, the BER of a recursive convolutional code is lower than that of the equivalent non-recursive convolutional code.

I-10- Concatenation of Codes

A concatenated code is made up of two distinct codes that are joined together to create a bigger code. Serial and parallel concatenation are the two forms of concatenation. The serial concatenation method for transmission is shown in Figure 15.

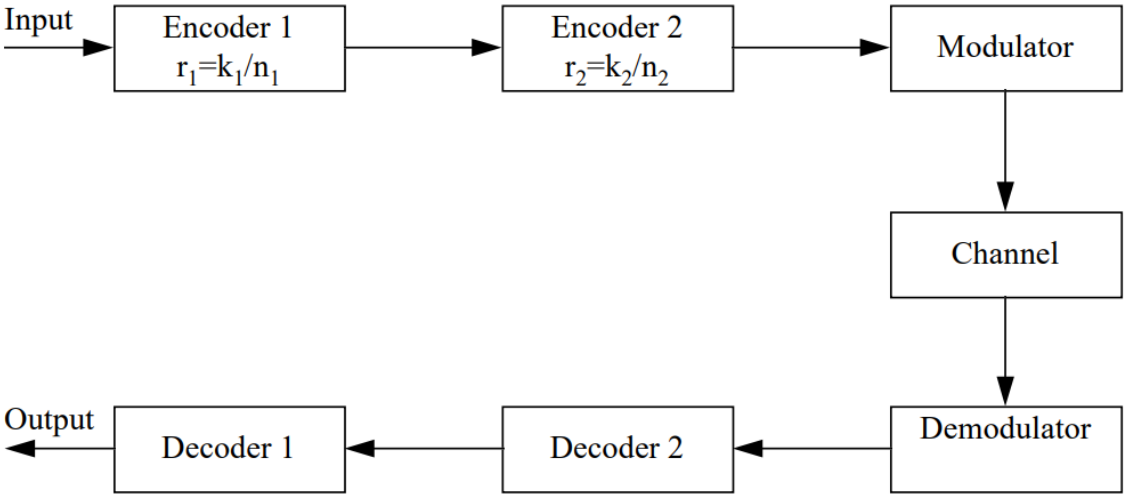


Figure 15: Serial concatenated code.

The total code rate for serial concatenation is

$$r_{tot} = \frac{k_1 k_2}{n_1 n_2}$$

Equation 3

which is equal to the product of the two code rates. Figure 16 shows the parallel concatenation scheme for transmission.

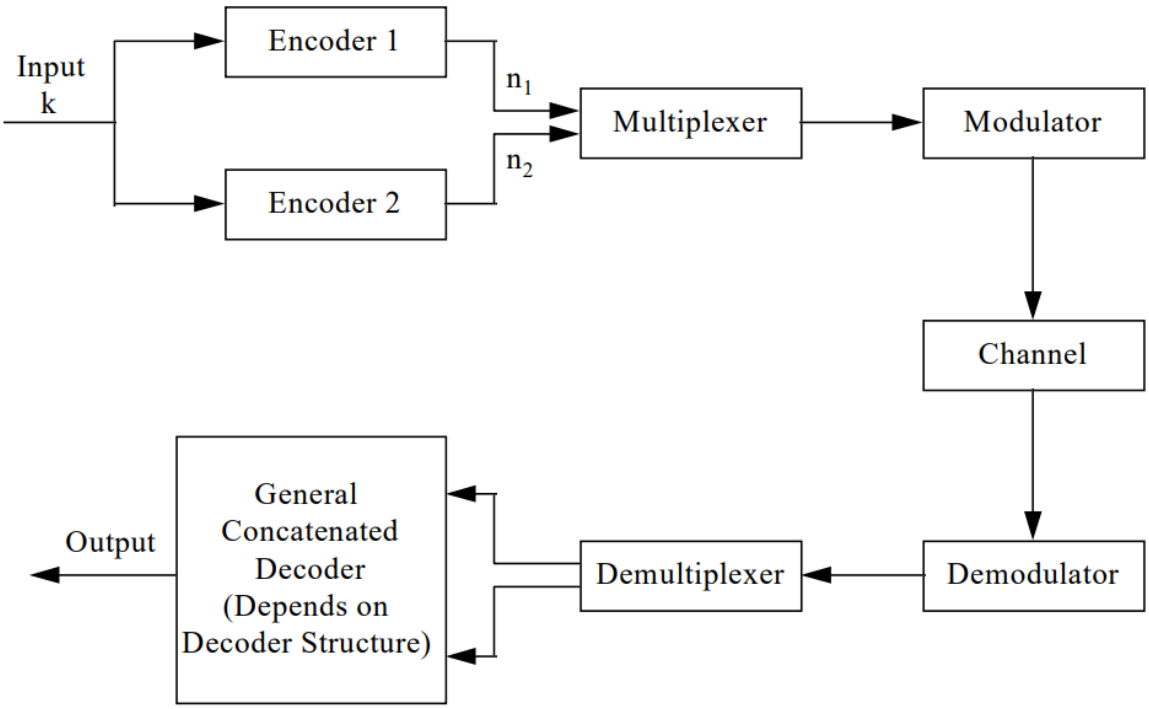


Figure 16: Parallel concatenated code.

The total code rate for parallel concatenation is

$$r_{tot} = \frac{k}{n_1 + n_2}$$

Equation 4

An interleaver is often employed between the encoders in both serial and parallel concatenation systems to enhance burst error correcting capacity or increase the unpredictability of the code. The parallel concatenated encoding method is used by Turbo codes. The turbo code decoder, on the other hand, uses a serial concatenated decoding method. The serial concatenated decoders are employed because they perform better than the parallel concatenated decoding scheme because the serial concatenation method allows the concatenated decoders to exchange information, while the parallel concatenation system requires the decoders to decode separately.

I-11- Turbo Coding

Berrou, Glavieux, and Thitimajshima developed the notion of turbo coding in a major contribution in 1993, reporting outstanding coding gain values that approached Shannonian expectations. The information sequence is encoded twice, with an interleaver between the two encoders providing to approach statistical independence between the two encoded data

sequences. Half-rate Recursive Systematic Convolutional (RSC) encoders are often employed, with each RSC encoder giving a systematic output that is equal to the original information sequence as well as a stream of parity data. After that, the two parity sequences may be punctured before being sent to the decoder with the original information sequence. Because the parity information is punctured, a broad variety of coding speeds may be achieved, and half of the parity information from each encoder is often delivered. This leads in an overall coding rate of 1/2 when combined with the original data sequence.

The reason turbo coding is very necessary in modern digital communication is because it provides error control performance within a few $1/10^{th}$ of a dB of the Shannon limit equation we talked about earlier, using decoders that are simpler than those used in the previous NASA/ESA standard.
The increase in data rate and range resulting from turbo coding had a significant impact on the telecommunication industry, it had a particular use and effect in wireless personal communicating systems, those that have increased demands for bandwidths as the days go on to keep up with the increased data services.

Turbo coding key design consists of two key innovations, one being the Iterative Decoding and the second being the Parallel Concatenated Encoding, or for short, PCE's, let us start with those. PCE's contain two or more component encodes for block or convolutional codes, in its least complex form, the parallel concatenated encoders work this following way, let us suppose that there are two component encoders, a message block $\boldsymbol{m}$ is going to be encoded using our first component, resulting in a generated codeword (also called code-sequence) $\boldsymbol{c_1}$. After that, the original message $\boldsymbol{m}$ is going to get interleaved and the result is to be used as a message for the second encoder, generating a second codeword $\boldsymbol{c_2}$. After that, $\boldsymbol{m}$, $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are multiplexed and sent over the channel to the receiver.

Two RSC decoders are utilized at the decoder. Decoding algorithms that take soft inputs and provide soft outputs for the decoded sequence must be employed. These soft inputs and outputs offer not only a 0 or 1 signal, but also a likelihood ratio, which indicates the likelihood that a given bit was properly decoded. The turbo decoder works in an iterative fashion. The first RSC decoder gives a soft output in the first iteration, estimating the original data sequence only based on the soft channel inputs. There is also an extrinsic output. The information for surrounding bits and the limitations imposed by the code are utilized to determine the extrinsic output for a specific bit, rather than the channel input for that bit. The second RSC decoder uses this

extrinsic output from the first decoder as a-priori information, and this information, together with the channel inputs, is utilized by the second RSC decoder to produce its soft output and extrinsic information. The extrinsic information from the second decoder in the first iteration is utilized as a-priori information for the first decoder in the second iteration, and the decoder can ideally decode more bits properly using this a-priori information than it could in the first iteration. This cycle repeats itself, with both RSC decoders providing a soft output and extrinsic information depending on the channel inputs and a-priori information derived from the preceding decoder's extrinsic information at each iteration. The Bit Error Rate (BER) in the decoded sequence decreases with each iteration, but the benefits made with each iteration decrease as the number of iterations grows, therefore only 4 to 12 iterations are commonly employed for complexity reasons [7].

The concurrent concatenation of two or more, generally identical, rate R = 1/2 convolutional encoders, accomplished in recursive systematic or systematic feedback form,1 plus a pseudorandom interleaver, constitutes an encoder for a classical turbo code. In contrast to standard serial concatenation, where the second encoder acts on the output bits of the first encoder, this encoder structure is termed a parallel concatenation since the two encoders act on the same block of input bits. Consider turbo encoders with two identical component convolutional encoders, while this encoding approach may be readily expanded to various encoders or three or more encoders. However, just two component encoders are required to get the primary findings and performance outcomes. The interleaver is used to permute the input bits such that the two encoders work on the same block of bits but in separate orders. The first encoder gets the input bit U_r directly and generates the output pair (U_r, v (1) r), while the second encoder gets the input bit u r and creates the output pair (U_r, v (2) r), with the u r being deleted. The input bits are arranged into finite length blocks with a length of N equal to the interleaver's size. Because both encoders are systematic and work with the same set of input bits, the input bits only need to be sent once, and the entire code has a rate of R = 1/3. The two parity sequences, v(1)(D) and v(2)(D), may be punctured to boost the rate of the code to R = 1/2, often by deleting v (1) r or v (2) r alternatively. An (h0, h1, N) turbo code is one in which the component encoders contain parity-check polynomials h0(D) and h1(D) and the interleaver is of length N, where h0 and h1 are the octal representations of the connector polynomials. It's worth noting that N is sometimes referred to as the code's block length in the literature, despite the fact that this nomenclature contradicts established definitions of block length. In reality,

n = N/R is the codelength [20].

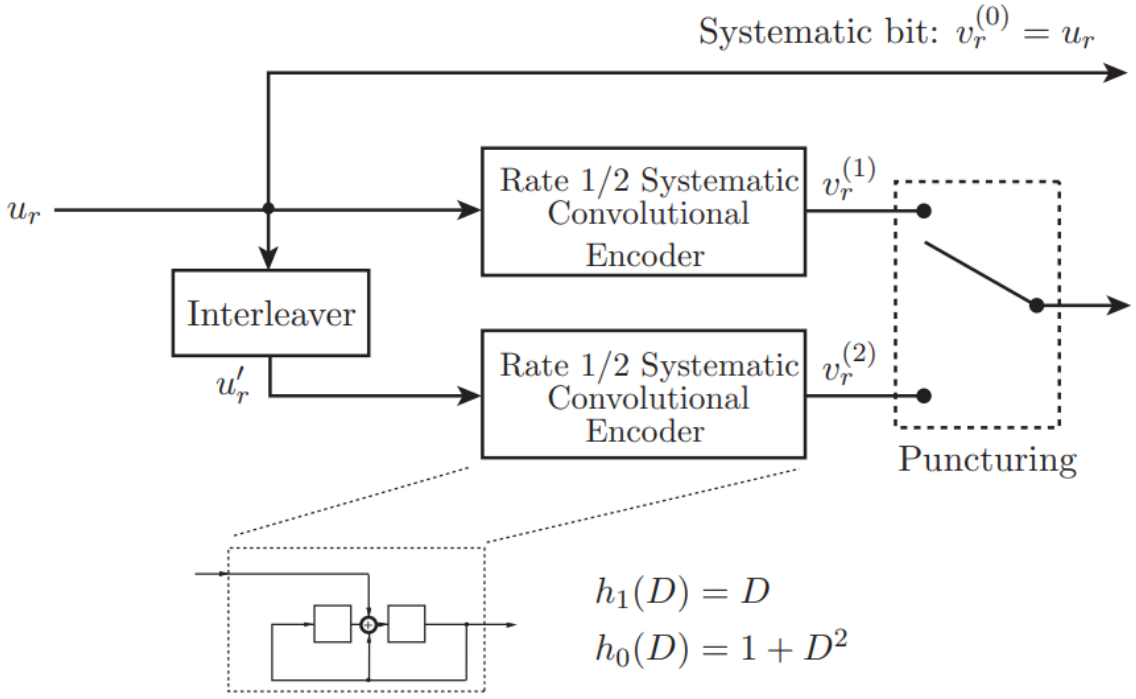


Figure 17: Block diagram of a turbo encoder with two component encoders and an optional puncturing device to increase the code rate above the base rate of R = 1/3

I-11-1- Turbo Encoder

Figure 18 depicts the overall construction of turbo encoders. The identical input bits are encoded using two component codes, but an interleaver is positioned between the encoders. RSC codes are often used as component codes, but alternative component codes, such as block codes, may be employed to obtain high performance using a structure like that shown in Figure 18, as proposed by Hagenauer and Offer and Papke and Pyndiah. Additionally, it is possible to use more than two component codes. This chapter, on the other hand, focuses only on the typical turbo encoder construction, which employs two RSC codes. Chapter 7 discusses turbo codes that use block codes as component codes [8].

The two component codes' outputs are then punctured and multiplexed. Both component RSC codes are usually half rate, resulting in one parity bit and one systematic bit produced for each input bit. The output bits from the two encoders must then be pierced in half to produce an overall coding rate of one-half. The most common configuration, which we have utilized in our

research, is to send all of the systematic bits from the first RSC encoder and half of the parity bits from each encoder. It's worth noting that puncturing the systematic bits lowers the code's performance more drastically than puncturing the parity bits [9].

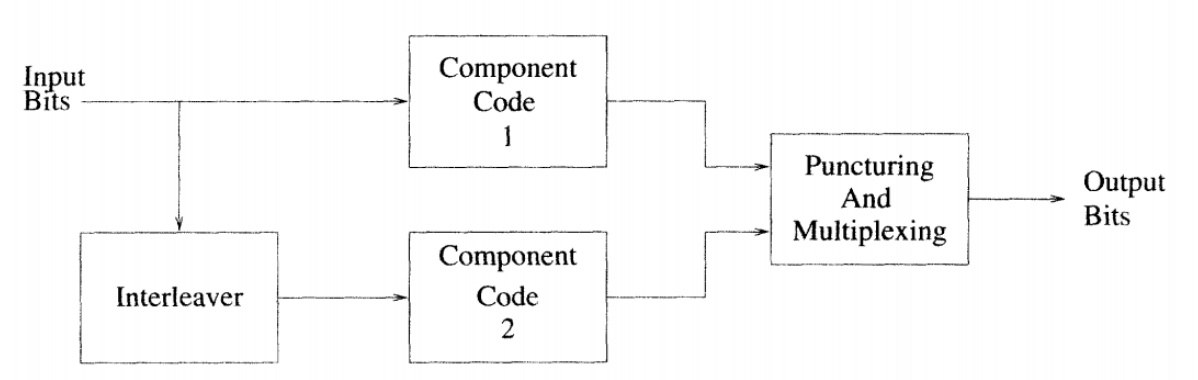


Figure 18: Turbo encoder schematic Berrou et al. ©IEEE, 1993

Although parallel concatenated codes had been studied extensively before to Berrou *et al*'s 1993 breakthrough study, turbo codes provided a huge gain in performance due to the interleaver employed between the encoders and the usage of recursive codes as component codes.

It seems that turbo codes provide a performance advantage proportionate to the length of the interleaver employed. The decoding difficulty per bit, on the other hand, is unaffected by the interleaver length. As a result, by utilizing very lengthy interleavers, exceptionally excellent performance may be accomplished with appropriate complexity. Extremely lengthy frame lengths, on the other hand, are not viable for many key applications, such as voice transmission, due to the delays they entail [6].

I-11-2- Turbo Decoder

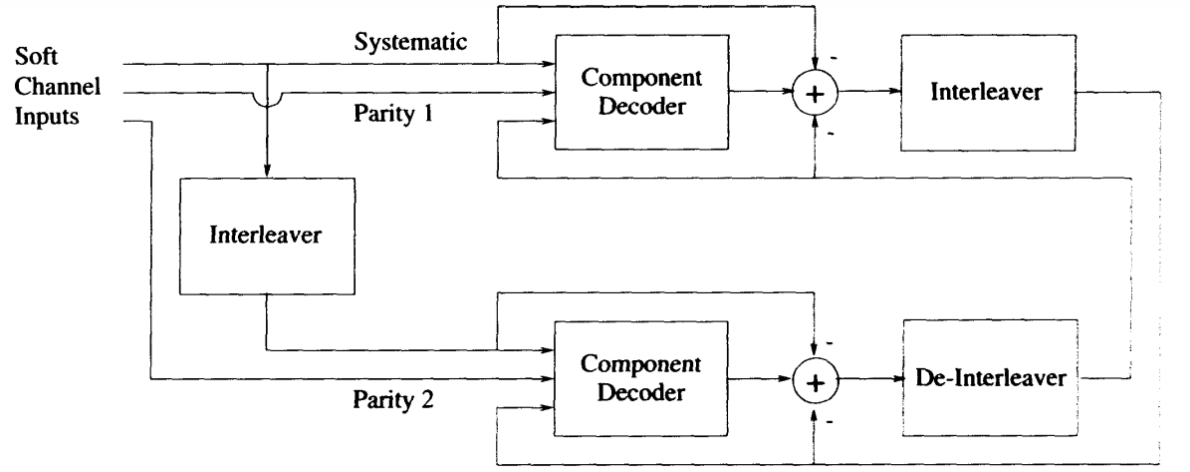


Figure 19: Turbo decoder schematic Berrou et al. ©IEEE, 1993

Figure 19 depicts the overall structure of an iterative turbo decoder. Interleavers connect two component decoders in a form similar to that of the encoder. Each decoder receives three inputs, as shown in the diagram: consistently encoded channel output bits, parity bits communicated from the corresponding component encoder, and information from the other component decoder regarding the probable values of the bits in question. A-priori information refers to the information obtained from the other decoder. Both the channel's inputs and this a-priori information must be used by component decoders. They must also offer so-called soft outputs for the decoded bits. This implies that the component decoders must not only provide the decoded output bit sequence, but also the corresponding probability for each bit that has been properly decoded. The so-called Log Likelihood Ratios are often used to describe soft outputs (LLRs). The LLR's polarity determines the bit's sign, while its amplitude indicates the likelihood of making a right judgment. The Soft-Output Viterbi Method (SOVA) suggested by Hagenauer and Hoher and Bahl's Maximum A-Posteriori (MAP) algorithm are two acceptable decoders [6].

Figure 19 shows an iterative decoder, with the first component decoder using just channel output values and producing a soft output as an approximation of the data bits in the first iteration. The soft output from the first encoder is then utilized as extra information by the second decoder, which calculates its estimate of the data bits using this information as well as the channel outputs. The second iteration may now begin, and the first decoder decodes the channel outputs once again, but this time with the value of the input bits supplied by the second

decoder's output in the first iteration. This extra information enables the first decoder to get a more precise set of soft outputs, which are subsequently utilized as a-priori information by the second decoder. This cycle is repeated, and the BER of the decoded bits tends to decrease with each repetition. The performance advantage gained with increasing numbers of iterations, on the other hand, declines as the number of iterations grows. As a result, only roughly eight iterations are normally employed due to complexity.

Because of the encoder's interleaving, it's important to appropriately interleave and de-interleave the LLRs that represent the soft values of the bits, as shown in Figure 19. Furthermore, due to the repeated nature of the decoding, care must be used to avoid repeating the same information at each decoding phase. Berrou et al. developed the idea of so-called extrinsic and intrinsic information to describe iterative decoding of turbo codes in their landmark study.

Other non-iterative decoders that provide optimum decoding of turbo codes have been presented. However, the gain in performance over iterative decoders was only approximately 0.35 dB, despite the fact that they are quite complicated. As a result, the iterative approach shown in Figure 19 is often employed [6].

I-12- The Maximum A-Posteriori Algorithm

Bahl, Cocke, Jelinek, and Raviv devised the Maximum A-Posteriori (MAP) method in 1974 for calculating the a-posteriori probabilities of the states and transitions of an observable Markov source when exposed to memoryless noise. The BCJR algorithm, named after its creators, is another name for this method. They demonstrated how to utilize the method to decode both block and convolutional codes. The method is optimum for decoding convolutional codes in terms of minimizing the decoded, unlike the Viterbi method, which minimizes the likelihood of the decoder selecting an erroneous route through the trellis. As a result, the Viterbi algorithm may be viewed of as reducing the number of groups of bits connected with these trellis routes, rather than the number of bits that are erroneously decoded. Nonetheless, as mentioned by Bahl et al., the performance of the two methods will be almost comparable in most cases. However, since the MAP algorithm evaluates every potential route through the convolutional decoder trellis, it first seemed to be too complicated for most systems to implement. As a result, until the discovery of turbo codes, it was not frequently employed.

The MAP method, on the other hand, not only offers the estimated bit sequence, but also the probability for each bit that has been accurately decoded. This is required for Berrou et al's decoding of turbo codes, hence MAP decoding was utilized in this important research. Since then, a lot of effort has gone into reducing the MAP algorithm's complexity to a manageable level [6].

I-13- The Soft-output Viterbi Algorithm

The Viterbi algorithm (VA) is tweaked to provide not only the most probable route sequence in a finite-state chain, but also the a posteriori probability for each bit or a reliability rating. The updated VA generates soft choices to be employed in the decoding of outer codes using this reliability indicator. Similar to an FM demodulator, the inner Soft-Output Viterbi Algorithm (SOVA) receives and produces soft sample values and may be seen as a device for enhancing SNR. Concatenated convolutional codes, concatenation of convolutional and basic block codes, concatenation of Trellis-Coded Modulation with convolutional FEC codes, and coded Viterbi equalization are among the applications researched to demonstrate the advantage over the standard hard-deciding VA. In comparison to traditional hard-deciding algorithms, we discovered extra improvements of 1 to 4 dB for certain applications. We also looked at the more difficult symbol-by-symbol MAP technique, which can translate optimum a posteriori probability into soft outputs.

The Viterbi algorithm [I], which performs operations including demodulation, decoding, and equalization, has become a common instrument in communication receivers. Concatenation of two VA is becoming more common in applications. Without bandwidth extension, coded modulation techniques such as coded quadrature amplitude modulation (QAM) [Z] and continuous phase modulation (CPM) are examples. Viterbi receivers are used instead of traditional modulation techniques in these systems. Convolutional codes with Viterbi decoding might be used in an extra outer coding scheme to accomplish forward error-correction (FEC) decoding.

There are usually two disadvantages to such a solution: first, the inner VA for demodulation causes bursts of mistakes, which the outside VA must counteract.

Second, the inner VA makes harsh judgments, which prevents the outer VA from exploiting its capacity to accept soft choices. The first flaw may be overcome by including some interleaving between the inner and outer VAs. The inner VA must produce soft-decisions, i.e., dependability

information, to remove the second flaw. This should significantly increase the outer VA's performance [10].

Chapter II: Interleavers

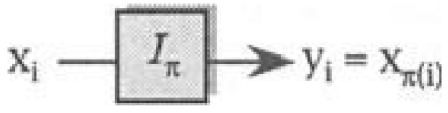
Chapter II: Interleavers

II-1- Introduction

Interleaving is a signal processing method that is often employed in a number of communication systems. An interleaver is a device that accepts symbols from a set alphabet and outputs identical symbols in a different temporal sequence at the output. The traditional use of interleaving is to "randomize" the locations of faults created during transmission, enabling the receiver to employ random error correction codes. Burst error channels (e.g., wireless communications channels) and concatenated coding are examples of situations where the initial step of decoding causes burst errors (e.g., a Viterbi decoder). Berrou, Glavieux, and Thitimajshima devised parallel concatenated encoders, which are a more contemporary use of interleaving. "Block" and "convolutional" interleaving are the two most popular forms of interleaving. The input data is written along the rows of a set of memory components organized as a matrix in a traditional block interleaver, and then read out along the columns. The pseudorandom block interleaver is a variant of the traditional block interleaver in which data is stored to memory sequentially and read out in a pseudorandom sequence. The dominant trend in interleaver design for turbo coding applications has been this kind of interleaving [11].

The data is multiplexed into and out of a fixed number of shift registers in a traditional convolutional interleaver. The difference in the length of the i_{th} and $(i+1)^{st}$ register is a fixed increment. Such a classical interleaver will be called a multiplexed interleaver. As the applications of turbo coding advance, convolutional interleaving is likely to become more important.

An interleaver is a single-input, single-output device that receives symbol sequences in a fixed alphabet A and creates an output sequence that is identical to the input sequence save for order, across the same alphabet.



A de-interleaver for a given I_π is an interleaver I_μ that operates on the output of the interleaver and restores the symbols' original order (possibly with a delay). For example, the inverse interleaver $I_\pi = I_\mu^{-1}$ where the permutation $\mu = \pi^{-1}$, is a zero-delay de-interleaver [12].

$$\begin{pmatrix} 0 & 1 & \cdots & T-1 \\ \pi_0 & \pi_1 & \cdots & \pi_{T-1} \end{pmatrix}$$

The fundamental permutation of the interleaver is the description supplied by the preceding equation. Combining the basic permutation with the periodicity restriction provides the remaining values of the interleaver function, which cover all integer values.

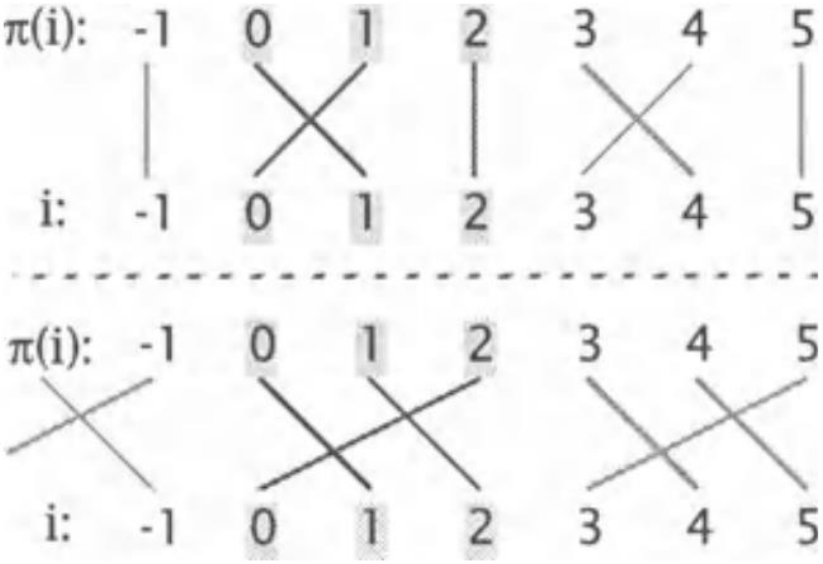


Figure 20: Two Period-3 Interleaver

II-2- Interleaver and Spreading

The primary goal of interleaver design for turbo codes is to reduce the error floor, which is the flattening of the error-rate curve that turbo codes show for moderate and high SNR values. A well-designed interleaver may also contribute to faster decoder convergence.

The turbo code asymptotic performance approaches the free-distance asymptote, according to Perez et al. Turbo codes have a low error floor owing to their short free distance and, as a result, a generally flat free-distance asymptote. To reduce the error floor, the turbo code's free-distance must be maximized for a given interleaver length. Based on this finding, several techniques for maximizing the turbo code output distance have been suggested, including tailoring the interleaver to particular component codes and attempting to break turbo code fault occurrences with low output weight. Interleaver design may also be proposed using cost function reduction in certain approaches [13].

Interleavers can be defined and implemented in a number of different ways. Figure 2 shows the definition used here. The interleaver reads from a vector of input symbols or samples, v_{in} and writes to a vector of interleaved or permuted output samples, v_{out}. The output samples are written using the write indexes $i = 0 \ldots K - 1$, where K is the interleaver length. Vector I defines the order that the samples are read from the input vector. That is, the i-th output, written to location i in the output vector, is read from location $I(i)$ in the input vector. The interleaver is completely defined by read vector I.

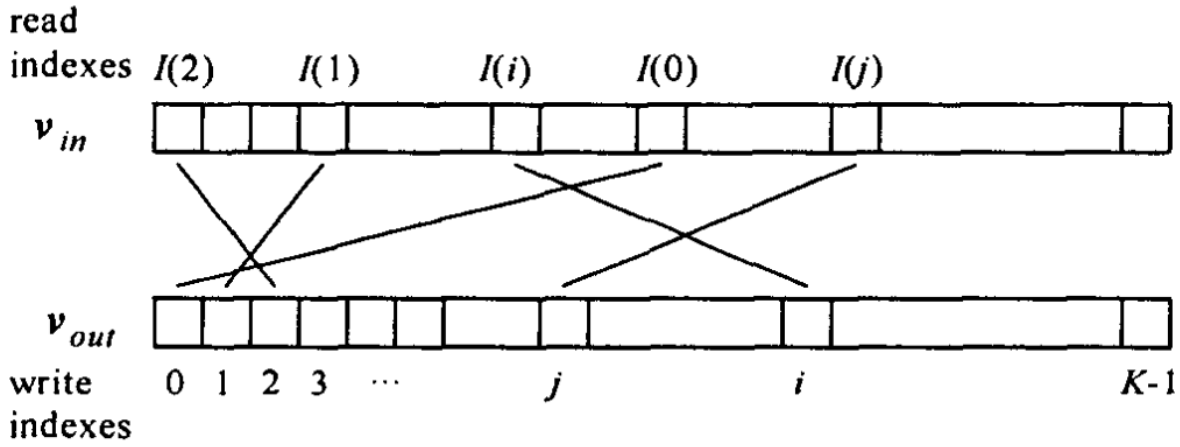


Figure 21: Interleaving Example

II-2-1- The Effect of Interleaver Size on Code Performance

The turbo code error performance is determined by the code distance spectrum. The interleaver in a turbo encoder can reduce the error coefficients of low weight codewords through a process called "spectral thinning". This distance spectrum results in a reduced bit error probability by a factor $1/N$, which is called the interleaving gain. The turbo code error performance at low SNR's is dominated by the interleaver size.

To illustrate the effect of the interleaver size on the code error performance, we consider a memory order 2 turbo code with generator matrix $\left(1, \frac{5}{7}\right)_{(oct)}$. The bit error probability of a turbo code over an additive white Gaussian noise channel is upper-bounded by

$$P_b(e) \leq \sum_{d=d_{min}} B_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right)$$

Equation 5

where B_d is the error coefficient and the set of all pairs of (d, B_d) represents the code distance spectrum [14].

The distance spectra of the turbo code with various interleaver sizes are calculated and substituted in the previous equation to calculate the bit error probability. The results are shown in Figs. 22 and Fig. 23 for the interleave sizes of 128, 256 and 512.

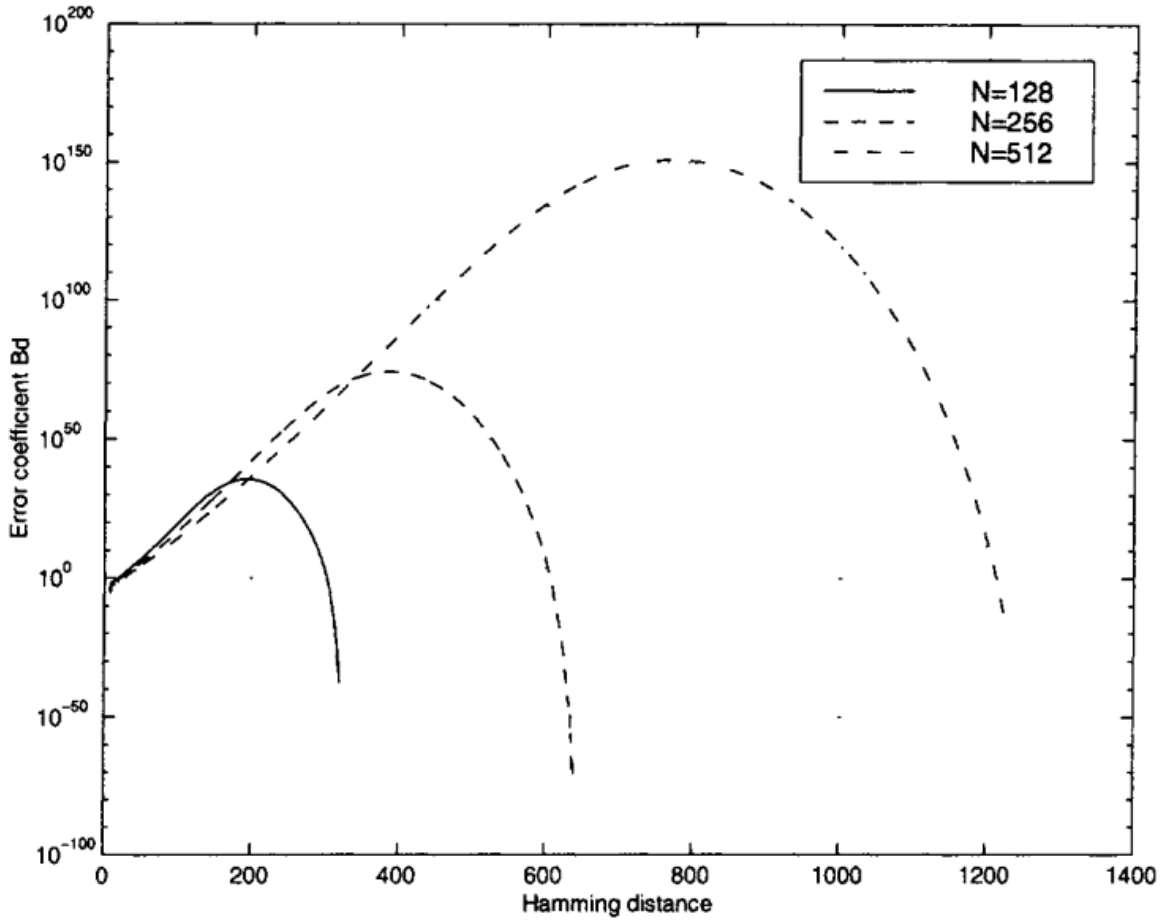


Figure 22: Distance spectra for a turbo code with various interleaver sizes

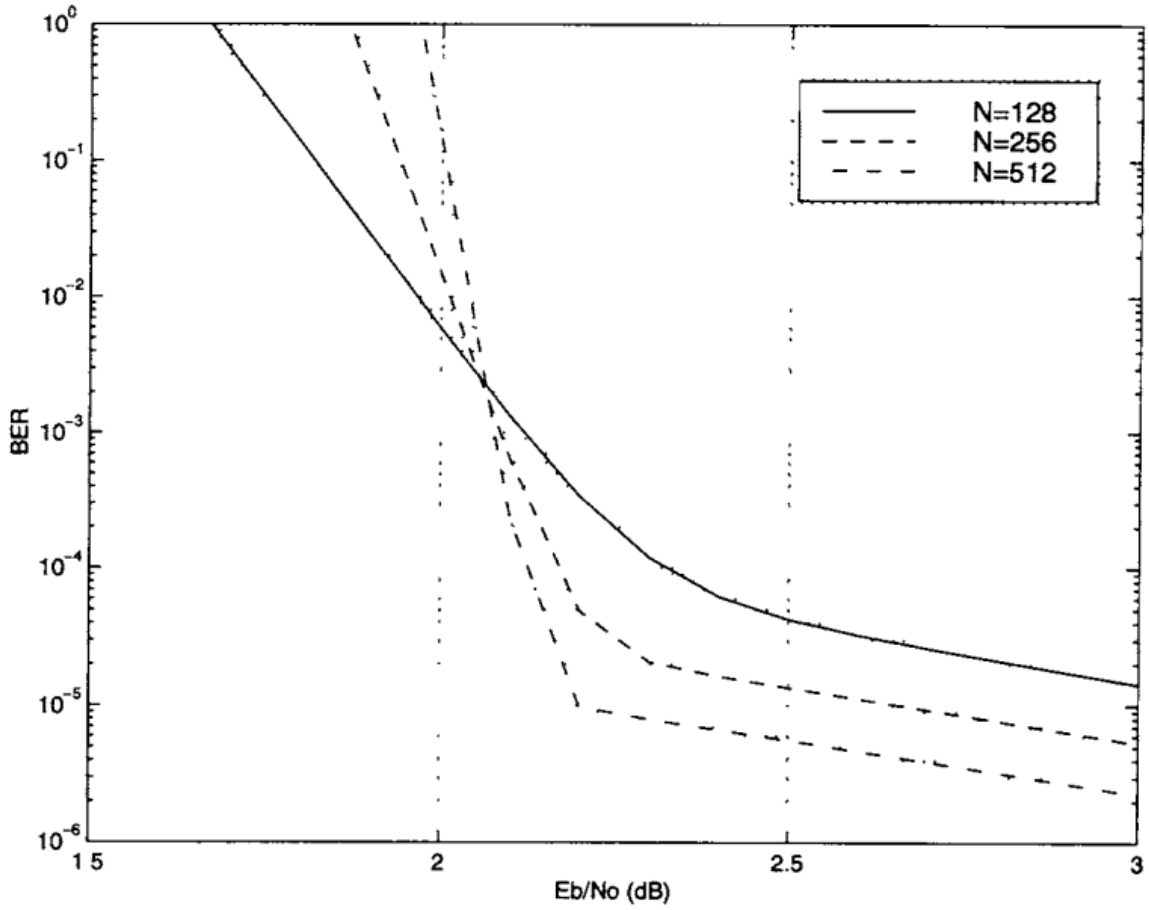


Figure 23: Bit error probability upper bounds for a turbo code with various interleaver sizes

II-2-2- Good Interleaver Measures

Because there are $N!$ unique permutations of length N, it is critical to establish effective interleaver measures for turbo codes; this lowers the number of interleavers that must still be filtered by expensive analysis and computer simulations for a complicated turbo codec system. The theory's major flaw was that, although it offered some guidelines for selecting appropriate PPs, the processes were too complicated when dealing with input weights greater than two. The spread factor and "randomness" are two common metrics for interleavers in turbo coding. "Randomness" is replaced with a more logical idea of an interleaver's degree of nonlinearity. An interleaver is represented by an interleaver-code, which is a geometric representation of an interleaver made up of pairs of coordinates $(x, f(x))$ that create points in Z_N^2. The number of discontinuous orbits (a collection of points) of the action of an isometry group of the interleaver code is measured by the degree of nonlinearity. Because of the algebraic-geometric character of PPs, they may be selected quickly to maximize the new metric [15].

DRP interleavers are among the most well-known interleavers for turbo codes, offering a combination of good error rate performance (beating –random interleavers) and simplicity, although they are not completely algebraic. Simulation curves of various turbo codes employing PP interleavers with remarkable frame error rate performance, comparable to those with DRP interleavers, demonstrate the effectiveness of the new measure. With the findings given in this dissertation, however, selecting excellent PP interleavers (especially QPP interleavers) is considerably easier [15].

II-3- Different Types of Interleavers

II-3-1- Block Interleavers

In communication systems, the block interleaver is the most frequently used interleaver. It writes from top to bottom and left to right in columns and reads from left to right and top to bottom in rows.

One basic type of interleaver is described by a finite permutation on the numbers $Z_T = \{0, 1, \ldots, T - l\}$

$$\begin{pmatrix} 0 & 1 & \cdots & T-1 \\ \pi_0 & \pi_1 & \cdots & \pi_{T-1} \end{pmatrix}$$

that correspond to the image of the T integers $Z = \{0, 1, \ldots, T-1\}$ under the operation of the permutation.

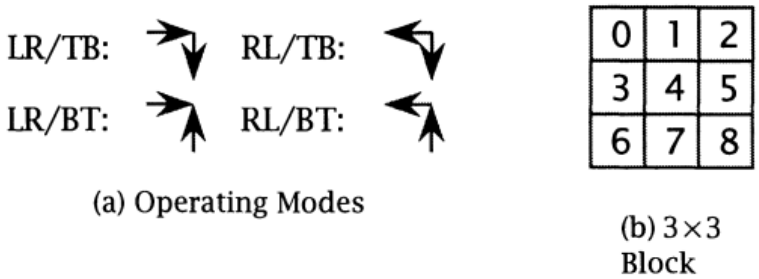


Figure 24: Classical Block Interleaver Scheme

A "block" permutation interleaver is one such interleaver. A permutation interleaver is a kind of block interleaver in which the basic permutation corresponds to a Z_T permutation.

II-3-2- Multiplex Interleavers

A multiplexed interleaver (or shift register interleaver) of period T is constructed by multiplexing the input sequence into T subsequences, introducing a delay for each subsequence, and de-multiplexing the T results. Such an interleaver can be implemented with a set of T shift registers where the length of the i_{th} register, m_i, determines the delay [11].

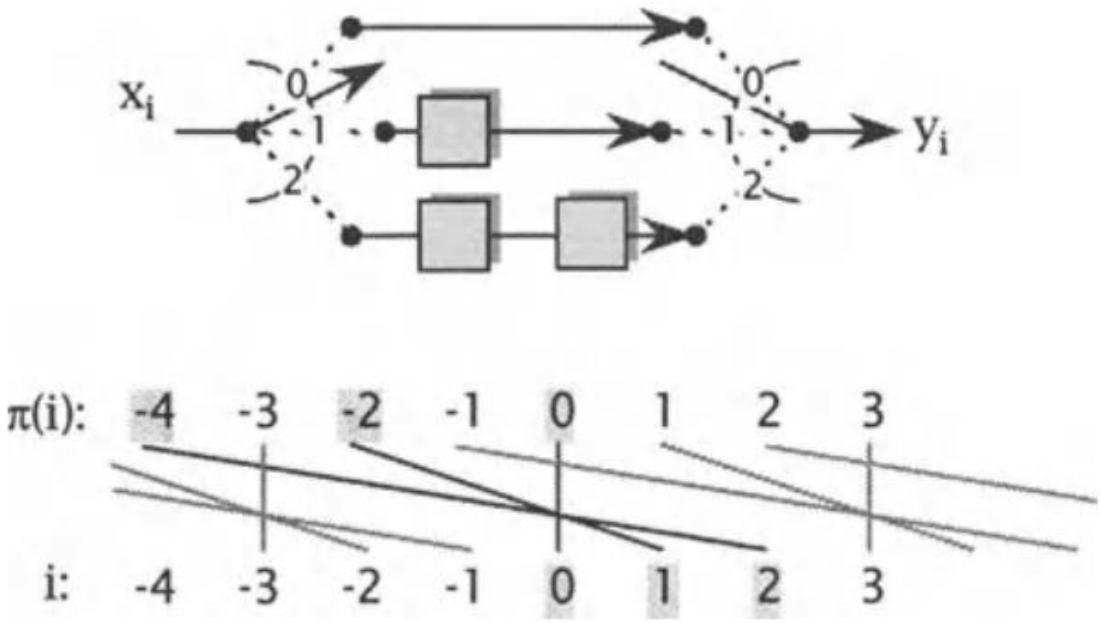


Figure 25: Another Period 3 Interleaver

I-3-3- Convolutional Interleavers

Convolutional interleavers have been proposed by Ramsey and Forney. The structure proposed by Forney is illustrated in Fig. 26.

Both the interleaver and de-interleaver consist of an input and output commutator and a bank of L shift registers. The information sequence to be interleaved is arranged in blocks of L bits. The input commutator cyclically inserts each block of L bits into the bank of L registers. The i^{th} bit in each block is delayed by the ith shift register and the delay of the shift register is $(i - l)B$. The output commutator cyclically samples the bank of L registers in the same order as the input one.

The de-interleaver performs the inverse operation. That is, the i^{th} bit in each block is delayed by the i^{th} shift register, where the delay of the shift register is $(L - i)B$ [15].

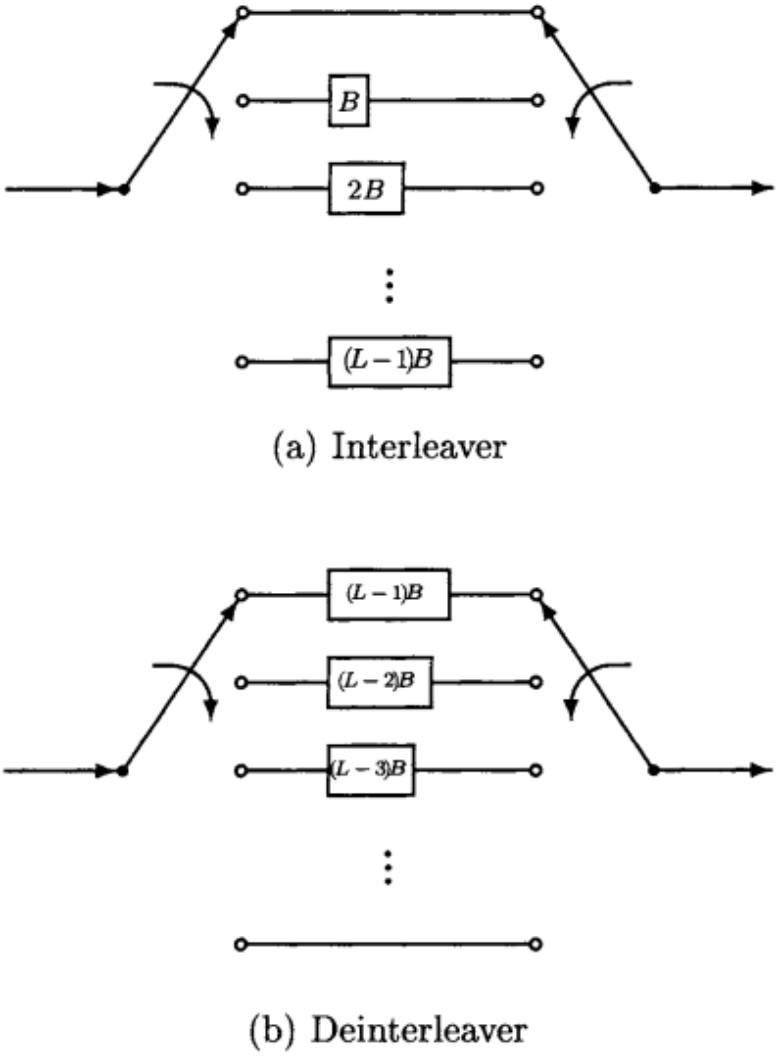


Figure 26 Interleaver and De-interleaver

II-3-4- Shuffle Interleaver

A shuffle interleaver is a general-purpose causal interleaving structure. A basic kind of card shuffling is referred to by the word. Consider a deck of $M + 1$ cards that must be shuffled or rearranged in a "random" order. The following procedure is followed at each time i:

1 – The deck's bottom card, x_i is chosen (consider this the "input" to the shuffler; the M remaining cards are considered the "state").

2 - A shuffle selector value, $0 \leq u_i \leq M$, is used to identify where the card x_i should be put in the deck. A value $u_i = M$ indicates the top of the deck, a value $u_i = 0$ indicates the bottom, a value $u_i = 1$ signifies one from the bottom and so forth.

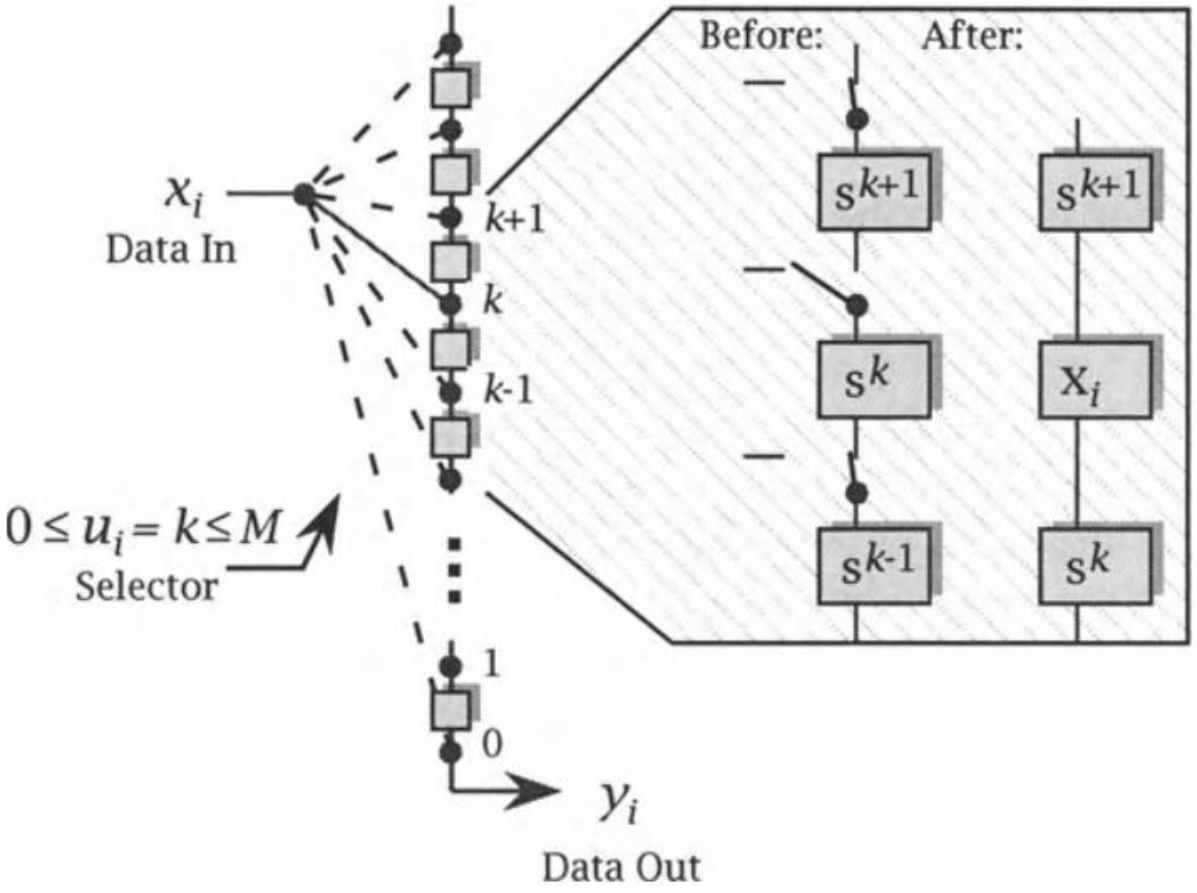


Figure 27 A Shuffle Interleaver

The specification of the shuffling sequence $\{u_i\}$ guides the functioning of the shuffling method.

The basic card shuffling process yields a shuffling interleaver by simply taking the bottom card as the output y_i and introducing a new card x_i as the input, as shown in Figure 27. Furthermore, the selection sequence must be considered to be periodic with a period of T. The interleaver is totally determined after the first T terms of the selection sequence $\{u_0, u_1, \ldots, u_{T-1}\}$ are known, The period of the interleaver, for example, is equal to the period of the selection sequence T and the memory required for the interleaver $M = max_{0 \leq i \leq T}\, u_i$, (In a minimum delay implementation of a shuffling interleaver, $min_{0 \leq i \leq T}\, u_i = 0$).

II-3-5- Random Interleaver

In the random interleaver a block of N input bits are read into the interleaver and read out randomly. The interleaver vector $\pi(i), i \in 1,2,\ldots,N$, can be generated according to the following algorithm, which requires N steps:

Step 1.

Choose randomly an integer i_1 from the set $A = \{1,2,\cdots,N\}$ according to a uniform distribution between 1 and N, with the probability of $p(i_1) = \frac{1}{N}$. The chosen integer i_1 is set to be $\pi(1)$

Step k.

$(k > 1)$ Choose randomly an integer i_k from the set $A_k = \{i \in A, i \neq i_1, i_2, \cdots, i_{k-1}\}$, according to a uniform distribution, with the probability of $p(i_k) = \frac{1}{N-k+1}$. The chosen integer i_k is set to be $\pi(k)$.

When $k = N$, the last integer i_N is set to be $\pi(N)$

When the interleaver size is $N = 2^m - 1$, a pseudo-random interleaver can be generated by an m-stage shift registers with linear feedback as illustrated in Fig. 28. The feedback polynomial represented by

$$1 + a_1 D + a_2 D^2 + \cdots + a_m D^m$$

must be primitive with degree m. If the initial state of the shift register is not the all-zero state, the shift register will go through all $2^m - 1$ states cyclically. Therefore, the state of the m-stage shift register can represent the interleaving function.

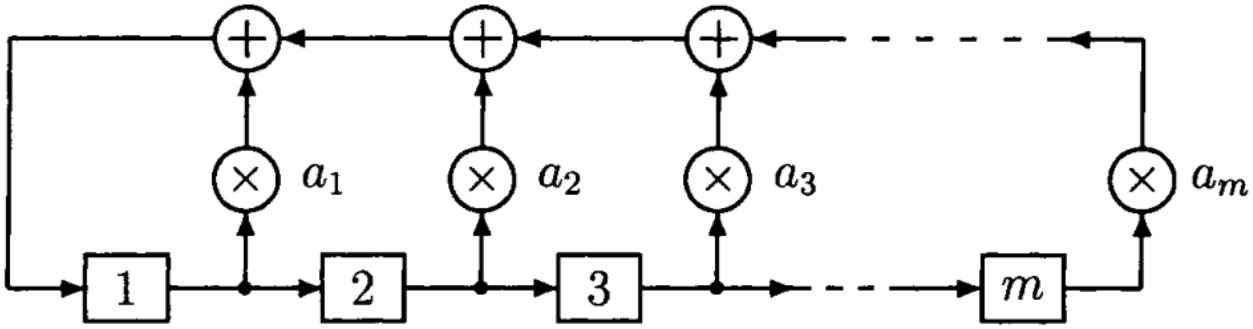


Figure 28: A general m-stage shift register with linear feedback

The random interleaver uses a fixed random permutation and maps the input sequence according to the permutation order. The length of the input sequence is assumed to be L. Figure 29 shows a random interleaver with L=8.

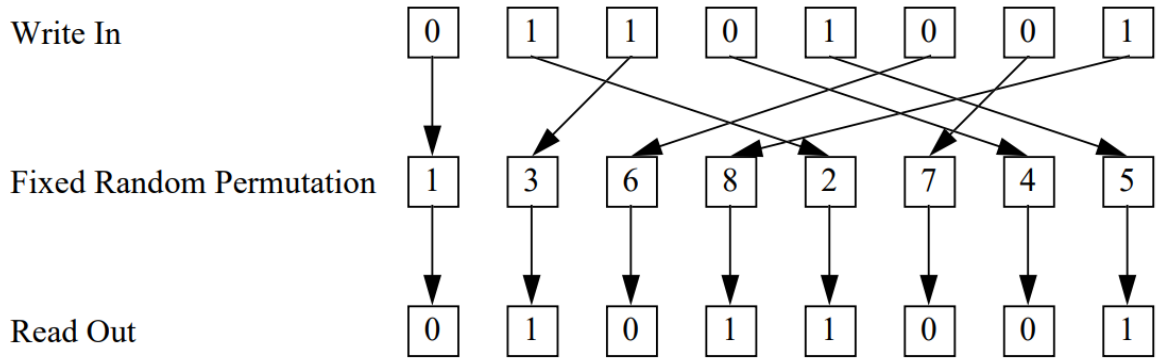


Figure 29:A random (pseudo-random) interleaver with L=8.

From Figure 29, the interleaver writes in [0 1 1 0 1 0 0 1] and reads out [0 1 0 1 1 0 0 1] [15].

II-3-6- S-Random Interleaver

S-random interleavers proposed by Divsalar and Pollara are pseudo-random interleavers. They are based on the random generation of N integers from 1 to N with an S-constraint, where S is defined as the minimum interleaving distance. S-random interleavers are also called spread interleavers [16].

An S-random interleaver is defined as follows. Each randomly selected integer is compared to the S_1 previously selected integers. If the absolute value of the difference between the current selected integer and any of the S_1 previous selected integers is smaller than S_2, then the current

selected integer is rejected. This process is repeated until all N integers are selected. In general, an S-random interleaver can be described as

$$|i_1 - i_2| \geq S_2$$

Equation 6

whenever

$$|\pi(i_1) - \pi(i_2)| \leq S_1, \qquad i_1, i_2 \in A$$

Equation 7

where S_1 and S_2 are two integers smaller than N. In a turbo encoder, these two parameters should, in general, be chosen to correspond to the maximum input pattern lengths to be broken by the interleaver. Thus, they should be chosen as large as possible. The length of an input pattern is defined as the length of the input sequence, starting from the first binary "1" to the last binary "1".

When two identical component encoders are employed in a turbo encoder, it is appropriate to set $S_1 = S_2$. In the following analysis, we assume the two component codes are identical and $S_1 = S_2 = S$. Note that, for $S = 1$, the S-random interleaver becomes a random interleaver.

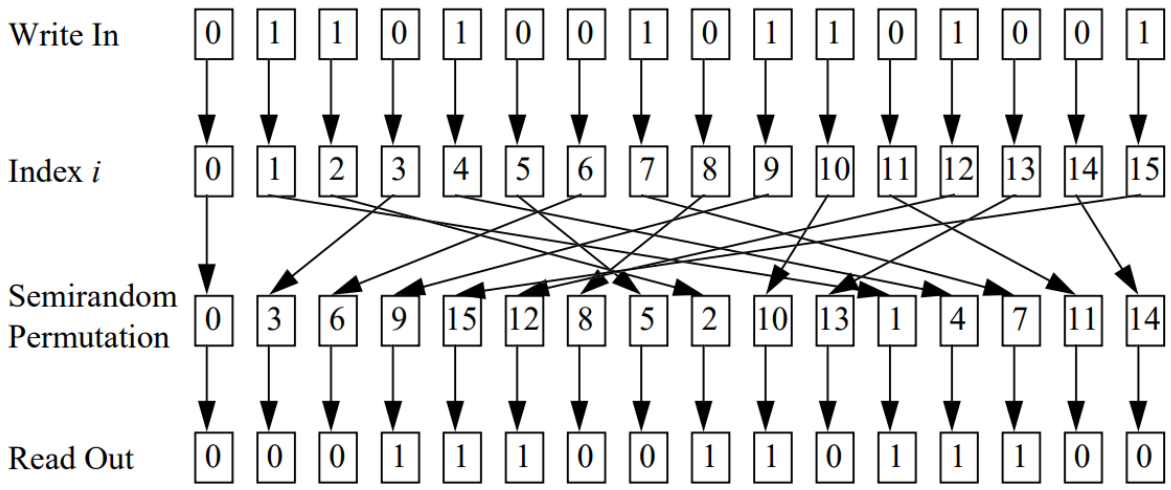


Figure 30: A semirandom interleaver with L=16 and S=2

From Figure 30, the interleaver writes in [0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1] and reads out [0 0 0 1 1 1 0 0 1 1 0 1 1 1 0 0]. The semirandom interleaver tries to introduce some randomness to

overcome the permutation regularity; however, the algorithm does not guarantee to finish successfully.

II-3-7- Golden Interleaver

A golden section ($g = (\sqrt{5} - 1)/2 = 0.618$). is used to create golden interleavers. They don't use integer modulo arithmetic, instead opting for sorting real numbers generated from the gold section. The method is based on calculating the true value of increment c, which is equal to $c = N(g^m + j)/r$. The golden vector v must now be created with real values. The following is how v elements are calculated [17]:

$$v(n) = s + nc \quad \text{mod} N, n = 0, \dots \dots, N - 1$$

Equation 8

Although the default value for s is 0, different real values for s may be used. In general, the recommended values for m are 1 or 2. j is set to 0 and r is set to 1 to stretch it out to the maximum number of neighboring elements.

The following stage consists of sorting the gold vector v and to find the vector of index z which defines this sorting, to find vector z of sorting such as a $(n) = v(z(n)), n = 0 \dots N - 1$, where $a = \text{sorts sort (v)}$. The indexes of the golden interleaver are then given by $\pi(z(n)) = N, n = 0 \dots N - 1$.

II-3-8- Dithered Golden Interleaver (DGI)

We get a decent spreading characteristic of the golden interleaver by adding a random component to it. D is the normalized width, and this dither (random) component d(n) is a uniform distribution between 0 and ND. The experimental value of D for turbo codes is set to 0.01. As a result, the indexing function will be as follows [17]:

$$v(n) = s + nc + d(n) mod N, n = 0, \dots \dots, N - 1$$

Equation 9

II-3-9- Algebraic Interleaver Models for Turbo Codes

Turbo Codes relies heavily on the interleaver (TCs). It serves a dual purpose. For starters, it has a significant influence on the Turbo codes' minimum Hamming distance. Second, it affects the correlation of transferred extrinsic information throughout the iterative decoding process owing

to its dispersion features. Algebraic permutations are favored over random-based permutations in real turbo coded systems. Permuted addresses may be calculated in this scenario by using a mathematical expression instead of using storage elements or a look-up table. As a result, they are simpler to define and implement. Dithered Relative Prime (DRP) interleavers, Quadratic Permutation Polynomial (QPP) interleavers used in LTE, and Almost Regular Permutation (ARP) interleavers used in DVB-RCS/RCS2, and WiMAX standards are three of the most common interleavers having the aforementioned features. There hasn't been a generalized permutation model for TCs till now. We demonstrate that these three interleaver families have a relationship in this study. Indeed, every DRP or QPP interleaver may be described as an ARP interleaver, as we show. As a result, ARP interleavers may attain, at the very least, the same minimal Hamming distances as DRP and QPP interleavers. Furthermore, since Turbo codes have been adopted by various standards, each with a distinct interleaving structure, standardizing the interleaving structure might be advantageous for implementation. In reality, an ARP interleaver may be designed to accommodate any specified interleaver architectures. When numerous interleaving structures must be implemented on the same chip, the total complexity may be lowered [18].

II-3-10- The ARP Interleaver

The ARP interleaver was suggested by Berrou et al. It's based on a shift vector S and a normal permutation with a period of P. The interleaving function's definition is as follows [19]:

$$\pi(i) = \big(iP_0 + A + d(i)\big) mod K \ ,$$

Equation 10

where $0 \leq i \leq K-1$ is the sequential index of the bit positions after interleaving, $\pi(i)$ is the bit index before interleaving corresponding to position i, K is the information block size in bits, $P0$ is an integer relatively prime to K, A is a constant offset, and $d(i)$ is a "dither" vector of length C where C is a small number (e.g., 4, 8) called the cycle length. For all block sizes, the dither $d(i)$ assumes the form

$$d(i) = \alpha(i mod C) + P_0 \beta(i mod C)$$

Equation 11

where $\alpha(\cdot)$ and $\beta(\cdot)$ are vectors each of length C, periodically applied for $0 \leq i \leq K-1$. Both the dithered vectors $\alpha(\cdot)$ and $\beta\ (\cdot)$ are composed of multiples of the cycle length C.

Because the block size K must be a multiple of C in an ARP interleaver, some block sizes (for example, primes) are not appropriate for ARP interleaver design. In reality, however, this restriction is reasonable since block sizes (K) that are multiples of tiny integers (C) offer more parallelism factor M options.

The ARP design has the advantage of being vectorizable for a variety of parallelism factors. For every parallelism M where K/C is a multiple of M, the ARP is CF for any block size K and cycle length C. For example, if C=8 is used in the construction of a 2048-bit interleaver, any integer from the range 1, 2, 4, 8, 16, 32, 64, 128, 256 is a viable option for the parallelism factor M.

(4) and its inverse have the same structure because (4) is equal to a linear permutation with a constant that is a function of index i. This allows for the use of a single interleaver and de-interleaver circuit.

II-3-11- The DRP Interleaver

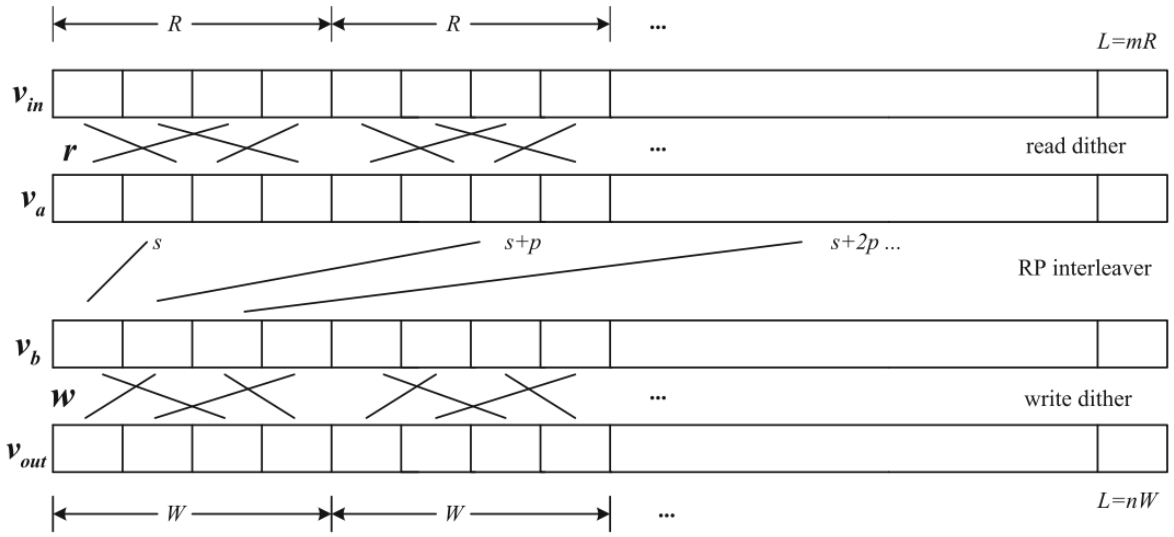


Figure 31: Design approach of a DRP interleaver

As introduced by Crozier and Guinand, the DRP interleaver is composed of three interleaving stages:

$$\Pi_a(i) = R\lfloor i/R \rfloor + r_{(imodR)}$$
$$\Pi_b(i) = (s + P \cdot i) mod K$$
$$\Pi_c(i) = W\lfloor i/W \rfloor + w_{(imodW)}$$

Equation 12

where $[x]$ denotes the closest lower integer value with respect to x; r and w are the read and write dither vectors with lengths R and W, respectively. The interleaver length K must be a multiple of the length of both dither vectors. P denotes the regular interleaver period, relatively prime to K, and s represents a constant shift. Then, the complete interleaver function is defined as:

$$\Pi_{\mathrm{DRP}}(i) = \Pi_a\big(\Pi_b(\Pi_c(i))\big)$$

Equation 13

where $i = 0, \dots, K-1$ is the address of the data symbol after interleaving and $\Pi_{\mathrm{DRP}}(i)$ represents its corresponding address before interleaving [19].

II-3-12- The QPP Interleaver

The quadratic permutation polynomials (QPP) were chosen as turbo code interleavers, proving to be the most promising answer to the LTE requirements. There are 188 possible lengths of the QPP interleavers for the LTE standard. The goal of this research is to identify quadratic permutation polynomials (QPP) interleavers that will increase the distance spectrum. Special performance, comprehensive algebraic structure, and economical implementation are all advantages of polynomial interleavers (high speed and low memory requirements).

The QPP interleaver has been adopted for the 3GPP LTE standard. Unlike the earlier 3G interleavers, the QPP interleaver is based on algebraic constructions. The QPP interleaver can be expressed by a simple mathematical formula. The relationship between the output index x and the input index $f(x)$ can be derived from the formula given below

QPP interleavers, proposed by Sun and Takeshita, are based on permutation polynomials over the integer ring $\mathbb{Z}_k$ where K corresponds to the interleaver length. Such an interleaver is completely defined by the algebraic expression:

$$\Pi_{\mathrm{QPP}}(i) = (f_1 i + f_2 i^2) mod K$$

Equation 14

where $i = 0, \dots, K-1$ denotes the address of the data symbol after interleaving and $\Pi_{\mathrm{QRP}}(i)$ represents its corresponding address before interleaving. For even data sequence

lengths, the necessary and sufficient condition for the polynomial in to define a valid permutation (i.e., one to one mapping) can have two different expressions:

1) 2^n divides K for $n > 1$: then, f_1 is relatively prime to K and all prime factors of K are also factors of f_2.
2) 2^n divides K for $n = 1$, but not for $n > 1$: then, $f_1 + f_2$ is odd, f_1 is relatively prime to $\frac{K}{2}$ and all prime factors of K, excluding 2, are also factors of f_2.

According to these conditions, K and f_2 can be factorized in their prime factors as:

$$K = 2^{\alpha_{K,1}} \prod_{i=2}^{\omega(K)} p_i^{\alpha_{K,i}}$$

$$f_2 = 2^{\alpha_{f,1}} \prod_{i=2}^{\omega(K)} p_i^{\alpha_{f,i}} \prod_{i=\omega(K)+1}^{\omega(f)} p_i^{\alpha_{f,i}}$$

Equation 15

where $\omega(K)$ and $\omega(f)$ represent the number of different prime factors of K and f_2, respectively. $\alpha_{f,1} = 0$ for $\alpha_{K,1} = 1$ and $\alpha_{f,1} > 1$ for $\alpha_{K,1} > 1$

Chapter III: Simulation

Chapter III: Simulation

III – 1 Introduction

In this chapter, we will be simulating different types of Interleavers for Turbo Codes.

As we said in the previous chapters, we need interleavers in order to shuffle the positions of the bits before we encode them with the recursive systematic encoders, we then send them through a modulator and a multiplexer.

We will be using an Additive White Gaussian Noise channel to send our simulated interleaved results in order to get the results of a real channel.

The simulation will be realized in MATLAB, by changing the random interleaver in the default simulation into different types of interleavers, these interleavers being:

Uniform Distribution Random Interleaver, *Regular Interleaver* (RI), *Dithered Golden Interleaver* (DGI), *Dithered Relative Prime* (DRP), *Almost Regular Permutation* (ARP) and finally a *Quadratic Permutation Polynomial* (QPP) which is the theme of this dissertation.

We simulated this interleaver using two lengths, 400 bits and 1024 bits.

The modulated and encoded information is going to be sent over an AWGN (Additive White Gaussian Noise) channel.

In order to see the clear difference between the two, we simulated using 5 different iterations at the decoding side, this way, we shall be able to have a more precise result than single decoding.

We chose to set a limit for how long the simulator can keep going, so we chose that it should stop once it has hit 60 frames or 300 bits lost. This way we are able to optimize the time needed to simulate the random interleaver and get some decent results.

The simulation was done on a Signal to noise ratio based on [0.0dB, 0.5dB, 1.0dB, 1.5dB, 2.0dB] and in the K=400 case, we will also be simulating in SNR = 2.5dB

these results are based on the original work of YuFei for their amazing work on coding the turbo codes into MATLAB.

simulation here was applied on MATLAB 2020a

III – 2 Simulation Parameters

We shall be using random values for the size of data, K=400 bits and K=1024 bits and for the polynomial generator will be using $[15,13]_8$

We are going to use the following parameters:

- Additive white Gaussian noise (**AWGN**)
- Decoding algorithm Log-MAP.
- Code Generator G= $(15, 13)_{8.}$
- The code rate R=1/2.
- Number of Iteration for each block transmitted: 5.
- The number of incorrect frames to terminate the simulation: 60.
- The number of incorrect bits to terminate the simulation: 300.
- The number of iterations is going to be unchanged through the study of both interleaver lengths, K= 400 bits and K= 1024 bits.

III – 2 – 1 Results and Comments

III – 2 – 1 – 1 K=400 bits

Let us start with K = 400 bits, we will be discussing the different results of each interleaver's Bit Error Rate (BER) and their respective Frame Error Rate (FER), the results for every interleaver is going to be shown in the following table, the BER and FER results will be shown in Figure 32 and 33.

These results were made possible by the help and hard word of our past colleges in [21].

Table 1: Results of the simulation for K=400bits

E_b/N_0		0dB	0.5dB	1dB	1.5dB	2dB	2.5dB
Aléatoire	**Ber**	1.3140e-01	5.4576e-02	1.4589e-02	5.7603e-03	1.5120e-04	1.8238e-05
	Fer	1.0000e+00	8.0000e-01	2.5000e-01	1.0853e-01	7.8383e-03	1.4726e-03
DRP (P=27, M=4)	**Ber**	1.2175e-01	8.0605e-02	1.7572e-02	3.5055e-03	1.6383e-04	4.3234e-06
	Fer	1.0000e+00	1.0000e+00	3.5714e-01	7.5000e-02	5.9328e-03	2.3936e-04
DGI (D=0.01)	**Ber**	1.1125e-01	5.2729e-02	1.3780e-02	1.4694e-03	1.4228e-04	4.5421e-06
	Fer	1.0000e+00	7.3333e-01	3.6111e-01	4.3651e-02	3.5303e-03	3.0819e-04
ARP (α=0.β=4, C=4, P=27)	**Ber**	1.0327e-01	5.1427e-02	1.8892e-02	1.8420e-03	3.4023e-04	6.6097e-06
	Fer	1.0000e+00	6.6667e-01	2.8571e-01	4.9096e-02	8.2418e-03	3.2462e-04
RI P=27	**Ber**	1.0579e-01	6.7590e-02	2.6952e-02	2.2416e-03	1.5461e-04	1.0248e-05
	Fer	1.0000e+00	7.5000e-01	5.0000e-01	5.8104e-02	4.9283e-03	2.8865e-04
QPP (f1=151 f2=40)	**Ber**	1.3140e-01	5.4576e-02	1.4589e-02	5.7603e-03	1.5120e-04	1.1491e-05
	Fer	1.0000e+00	8.8889e-01	4.6914e-01	7.7869e-02	6.1779e-03	4.5773e-04

- Bit Error Rate

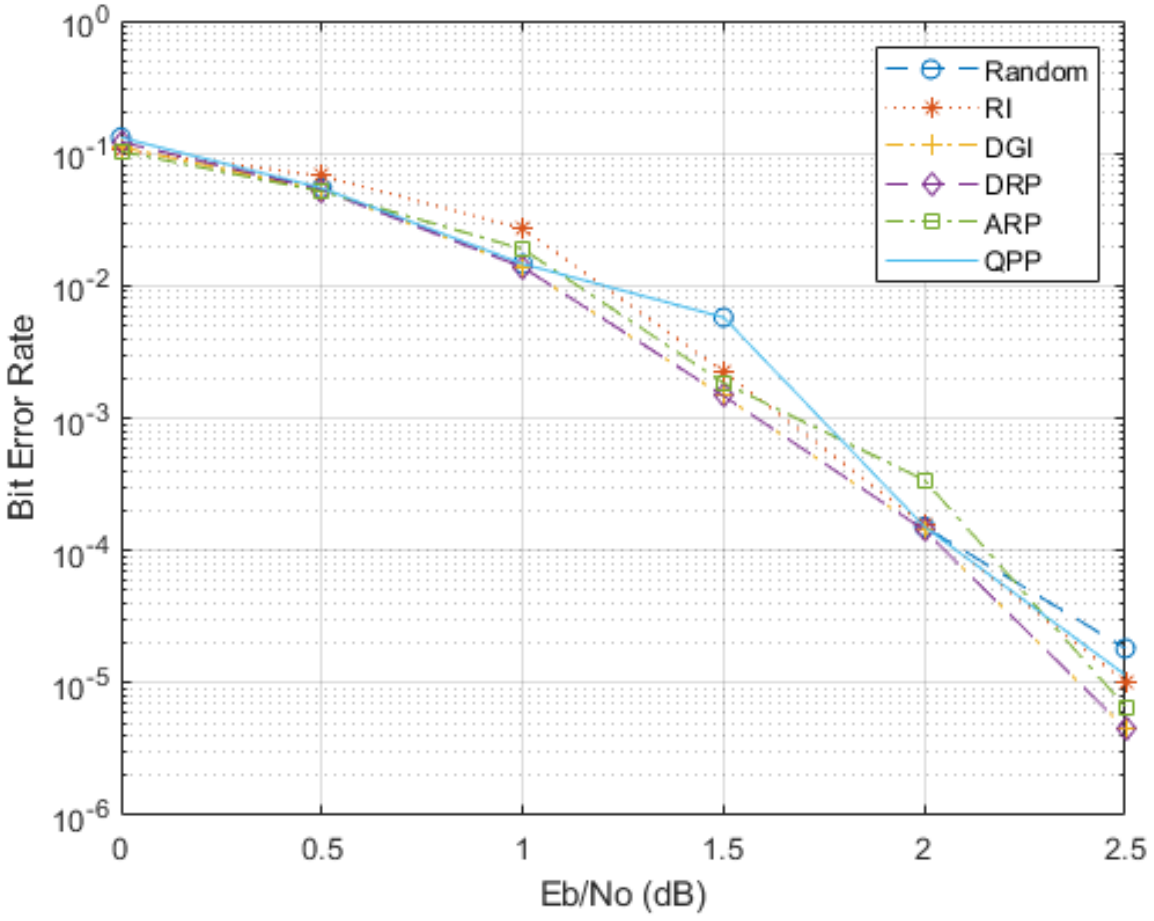


Figure 32: The Bit Error Rate / Eb/N0 of every interleaver of a length equal to 400 bits.

Figure 30 shows the difference in the Bit Error Rate using different interleavers in size 400 in different Signal-to-Noise Ratio (0dB, 0.5dB, 1dB, 1.5dB, 2dB, 2.5dB).

From the first glance, we can already notice that the random interleaver which is the oldest one, also has the lowest performance out of all, ending at 2.5dB with 1.82×10^{-5}

Quickly followed to the random is the RI, surprisingly, the Quadratic Permutation Polynomial is at the same place, we can see that the QPP interleaver doesn't deliver the best performance this kind of interleaver parameter. Even so, RI performs a little better than the QPP interleaver in this case, the RI's BER rests at 1.024×10^{-5} while the QPP performs at 1.15×10^{-5}

Moving forward, we can see that the Dithered Golden Interleaver and the Dithered Relative Prime interleaver perform better than the rest. Both these interleavers has Bit Error Rates equal to 4.54×10^{-6} for the DGI, and 4.32×10^{-6} for the DRP, it's safe to say that the previous two take the lead in this kind of parameter.

In order to fully understand the improvement difference between each interleaver, we will be studying that by finding the gain in dB in between each curve for each interleaver

We will be comparing the gain in between the UNIF Interleaver with every other interleaver because it is the lowest and it makes it easier for us to find the difference in gain between each and every interleaver.

Staring with the gain in the interleaver length of 400 bits, the difference between UNIF and QPP is 0.092dB, the coding between it and RI is 0.113dB, between ARP and UNIF is closer to the RI with 0.131dB, and finally the gain with DGI and DRP is equal at 0.202dB

From these results, we can see that DGI and DRP are the best in comparison with other interleavers.

- Frame Error Rate

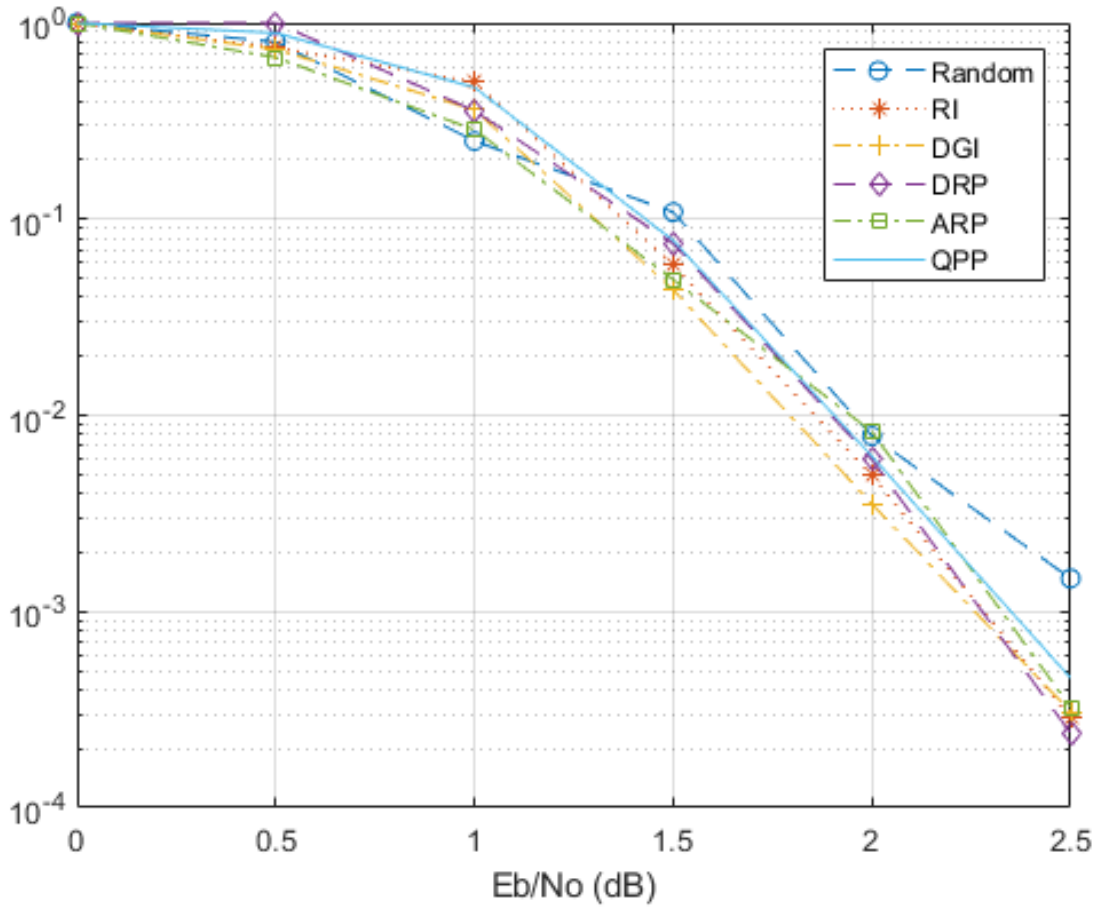


Figure 33: The Frame Error Rate / Eb/N0 of every interleaver of a length equal to 400 bits.

As we saw in the Bit Error Rate results, almost everything is still in its positions as before, this time, however, we see an increase of performance from the ARP Interleaver and the RI interleaver, although they are closer to the DGI and DRP but they still take the lead overall.

As usual, we will be getting the FER gain by comparing between UNIF and every other interleaver.

The gain is 0.221dB in between (UNIF and QPP), 0.233dB between (UNIF and ARP), 0.285dB between (UNIF and DRP and RI), and finally 0.323dB between (UNIF and DRP).

Once again, DGI and DRP are the ones in lead in these parameters, based off these results it's not recommended to use QPP interleavers in smaller interleaver lengths.

III – 2 – 1 – 2 K=1024 bits

Next, we will be simulating everything again using K = 1024bits, we will be discussing the different results of each interleaver's Bit Error Rate (BER) and their respective Frame Error Rate (FER), the results for every interleaver is going to be shown in the following table, the BER and FER results will be shown in Figure 34 and 35:

Table 2: Results of the simulation for K=1024 bits

E_b/N_0		0dB	0.5dB	1dB	1.5dB	2dB
Aléatoire	**BER**	1.1296e-01	8.0313e-02	1.0744e-02	2.4851e-04	1.9923e-05
	FER	1.0000e+00	1.0000e+00	3.6364e-01	2.1891e-02	1.8893e-03
DRP (P=45,M=8)	**BER**	1.2863e-01	8.1619e-02	9.5005e-03	1.2243e-04	1.6838e-06
	FER	1.0000e+00	1.0000e+00	4.3333e-01	1.2821e-02	3.5205e-04
DGI (D=0.01)	**BER**	1.1231e-01	6.3010e-02	1.4365e-02	1.3852e-04	2.8694e-06
	FER	1.0000e+00	1.0000e+00	2.8571e-01	2.1739e-02	4.0308e-04
ARP (α=0,β=8, C=8, P=45)	**BER**	1.1818e-01	6.4643e-02	6.4542e-03	2.3409e-04	7.9414e-06
	FER	1.0000e+00	1.0000e+00	2.8205e-01	1.8385e-02	1.2244e-03
RI P=45	**BER**	1.3190e-01	1.0121e-01	5.4631e-03	3.5216e-04	1.4531e-05
	FER	1.0000e+00	1.0000e+00	2.0000e-01	2.9963e-02	2.2479e-03
QPP (f1=31,f2=64)	**BER**	1.2259e-01	6.4185e-01	1.6895e-02	1.4144e-04	1.0320e-06
	FER	1.0000e+00	8.6667e+01	3.7500e-01	7.9745e-03	7.5714e-05

- Bit Error Rate

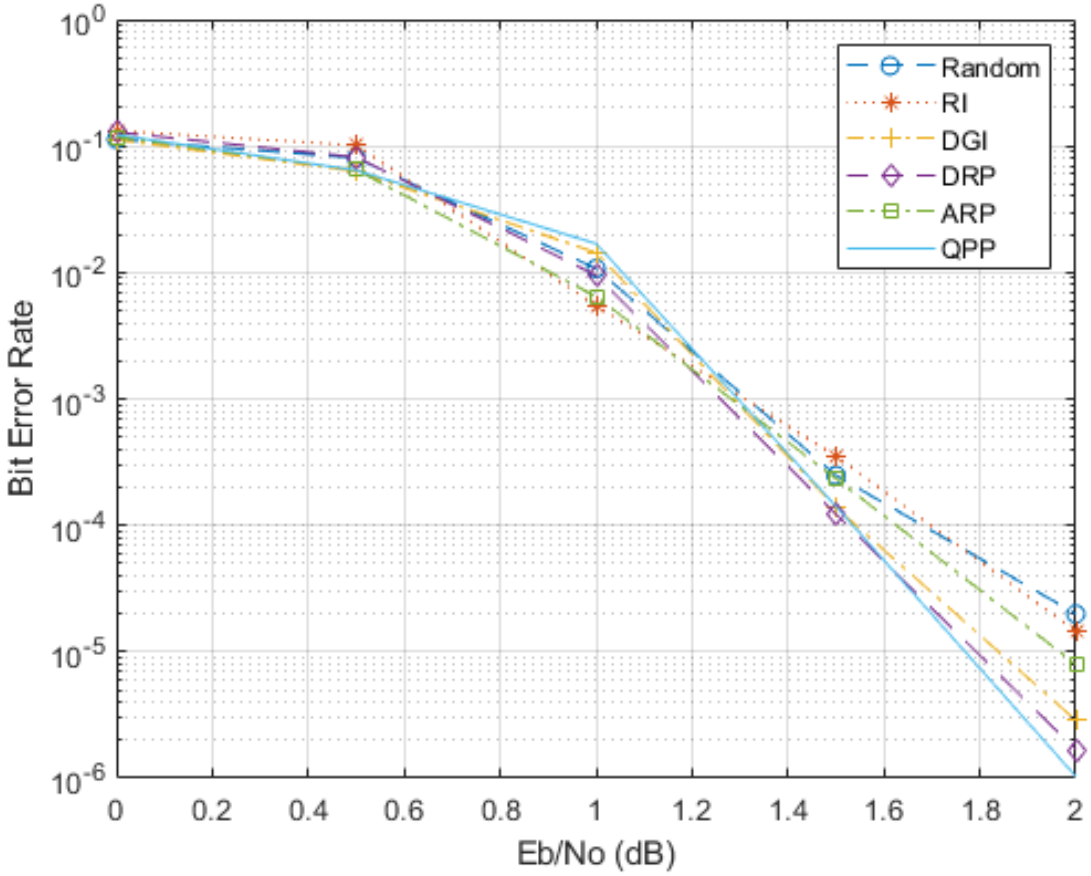


Figure 34 : The Bit Error Rate / Eb/N0 of every interleaver of a length equal to 1024 bits.

Figure 32 shows the difference in the Bit Error Rate using different interleavers in size 1024 in different Signal-to-Noise Ratio (0dB, 0.5dB, 1dB, 1.5dB, 2dB).

Again, in this case, we will be comparing the gain between each interleaver with the UNIF.

The difference between UNIF and RI is 0.05dB, between it and ARP is 0.132dB, DGI is 0.251dB, DRP is 0.29dB and finally the QPP has a gain of 0.3dB between it and the UNIF.

We can see that in this case, QPP is the better choice and has a much better gain comparing with others, even though the DRP is very close, but we can conclude that the QPP works best in these parameters.

- Frame Error Rate

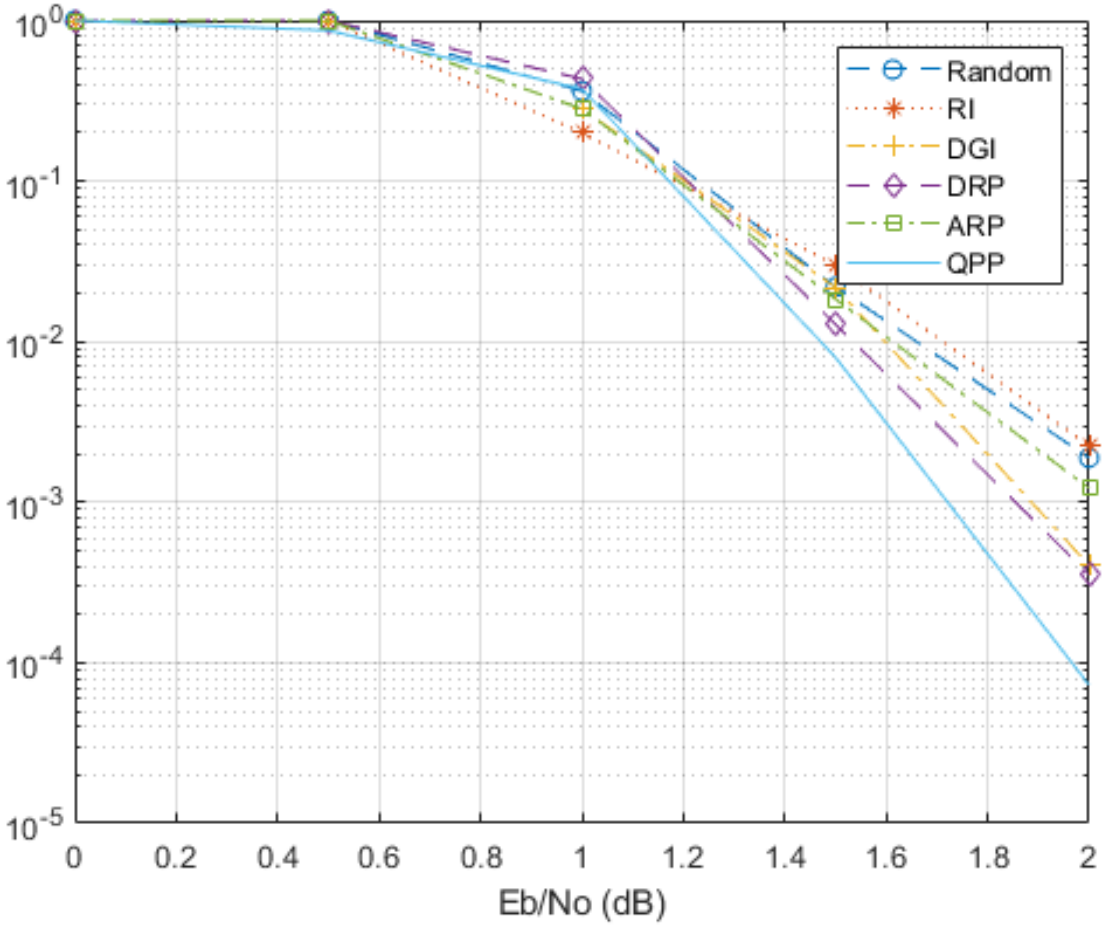


Figure 35 : The Frame Error Rate / Eb/N0 of every interleaver of a length equal to 1024 bits.

This time, we will be comparing the gain using the RI since it is the lowest one in comparison with the other.

The gain is 0.04dB between (RI and Random), 0.11dB between (RI and ARP), 0.21dB in between (RI and DGI), 0.26dB between (RI and DRP) and finally 0.36dB between (RI and QPP)

Just like we've seen with the Bit Error Rate, QPP takes the lead in comparison with every other Interleaver.

III-3 Conclusion

After the simulation, we can conclude that the QPP interleaver is a very well optimized interleaver, by looking at the received results we can see that the QPP interleaver works well with large interleaver lengths and does not perform too well in comparison with others in small interleaver lengths, it is easy to implement and has many lengths with optimized values.

General conclusion

Turbo Codes are at the heart of current telecommunications standards thanks to

their excellent error correction skills. This dissertation is made of two parts:

a theoretical part and a simulation. The theoretical part is intended to identify the

theoretical basis of these channel coders, studies on generalities on digital communications.

We presented and talked about the Shannon limits of performance, the Bit Error Rate and the hamming distance.

We specialized the first chapter to talk about the principles of turbo codes and the different parts that make them up, as well as speaking about some of the decoding algorithms.

Since this was specialized on talking about turbo codes, we had to talk about convolutional encoders and decoders, the Convolutional Encoders and the Recursive Systematic Convolutional Encoders.

In the second chapter, we decided to discuss all about interleavers and their parameters, their types, their measures and what differentiates between a good and bad interleaver, and different types of interleavers, we ended the chapter by speaking about some of most important ones of all, the ARP, DRP and finally the QPP, their equations and how to calculate their constants and variables

Finally, in the third and most important chapter, we got to talk about the simulation part and show all the results that we ended up with, we used MATLAB 2020a to simulate our results by changing the random interleaver in YuFei's code into the other different interleavers and simulated everything based on the parameters we wanted to end up with.

In our two attempts of simulation, we used two different interleaver lengths, 400bits and 1024bits, by using five iterations and the decoding part, we ended up with more precise results, we also used different Signal-to-Noise ratios to have better understandings of how each interleaver behaves in different mediums.

In the end, we found that the QPP interleaver which is the study of this dissertation, is that based on the results we have received, the QPP interleaver does not work well in smaller lengths like the 400, and is often overshadowed when compared with other interleavers like the DGI and the DRP.

However, we also found that the Quadratic Permutation Polynomial is very good to use in higher interleaver lengths like 1024 bits.

So, in conclusion, based on this dissertation's results, the QPP is better off used in higher interleaver sizes and that is why it was chosen to be used in the LTE communication systems, if the system used is requiring only a smaller interleaver size, then it is better to stick with another interleaver like the Dithered Golden Interleaver or the Dithered Relative Prime Interleaver.

We advise our next colleagues to try much more combinations and compare using different parameters and try to know for sure which is the best interleaver for these kinds of communication systems.

References

[1]. Farrell, P. G., & Jorge Castiñeira Moreira. (2006). *Essentials of Error-Control Coding 1st Edition.*

[2]. RED RIEKE, D. W. (1997). *Spikes: Exploring the Neural Code.*

[3]. Schlegel, C. B. (s.d.). *Trellis and Turbo Coding* .

[4]. Abraham Bookstein, V. A. (s.d.). *Generalized Hamming Distance.*

[5]. Clark Jr., G. C. (s.d.). *Error-Correction Coding for Digital Communications.*

[6]. Lajos Hanzo, T. H. (2002). *Turbo Coding, Turbo Equalisation and Space-Time Coding.*

[7]. Vucetic, B. J. (s.d.). *Turbo Codes Principles and Applications.*

[8]. Tujkovic, D. (s.d.). *Recursive space-time trellis codes for turbo coded modulation.* IEEE.

[9]. Hagenauer, J. (s.d.). *A Viterbi algorithm with soft-decision outputs and its applications.* IEEE.

[10]. Wicker, C. H. (s.d.). *Interleaving. In Turbo Coding* .

[11]. Gupta, V. C. (s.d.). *Performance Analysis for Different Interleavers in Various Modulation Schemes with OFDM over an AWGN Channel.*

[12]. Fragouli, C., & Wesel, R. (s.d.). *Semi-random interleaver design criteria.* IEEE.

[13]. Perez, L., Seghers, J., & Costello, D. (s.d.). *A distance spectrum interpretation of turbo codes.* IEEE.

[14]. Takeshita, O. Y. (s.d.). *Permutation Polynomial Interleavers: An Algebraic-Geometric Perspective.* IEEE.

[15]. Divsalar, D., & Pollara, E. (s.d.). *Low-rate turbo codes for deep-space communications.* IEEE.

[16]. Abderrahmane, L. H., Bacha, M., & Mebrek, A. (n.d.). *A new optimised interleaver structure for turbo coding.* IEEE.

[17]. Abraham Bookstein, V. A. (n.d.). *Generalized Hamming Distance.*

[18]. Berrou, C., Amat, A. G., Ould-Cheikh-Mouhamedou, Y., & Saouter, Y. (n.d.). *Improving the distance properties of turbo codes using a third component code: 3D turbo codes - [transactions letters]*. IEEE.

[19]. Bohórquez, R. G., Nour, C. A., & Douillard, C. (n.d.). *On the Equivalence of Interleavers for Turbo Codes.* IEEE.

[20]. Christian B. Schlegel, L. C. (2015). *Turbo Coding: Basic Principles.*

[21]. Rezzoug Hadjer., Tidjani Affaf (2020). Comparaison entre différentes techniques de permutation sur les turbo-codes. University Amar Telidji