République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Amar Thelidji- Laghouat

FACULTE: DE TECHNOLOGIE

DEPARTEMENT : D'ELECTRONIQUE

MEMOIRE DE MASTER

Réalisé par : Guesmia Rabeh

DOMAINE :Science et Technologie

FILIERE : Télécommunications

OPTION : Systèmes de Télécommunications

Thème

Study of Polar Codes

Jury de soutenance :

Nom et Prénom	Grade	Qualité
CHELLALI Safouane	MCB	Encadrant
Benkouider Fatiha	MCB	Président
Chouireb Fatima	Pr	Examinateur

Promotion : 2022/2023

بسم الله الرحمن الرحيم

dedication

I dedicate this humble work to the soul of my dear father, may God have mercy on him, and to the soul of my dead friend, Silaa Ahmed chaib, may God have mercy on him, and to all members of my family, friends, and honorable professors in the electronics department who supported me in everything, especially during my university career, and to everyone who has credit for me.

GUESMIA Rabeh

Thanks and gratitude

يقول النبي □ من لا يشكر الناس لا يشكر الله

فالحمد والشكر لله أولا واخيرا

On this happy occasion, I am pleased to extend my sincere thanks to all of my honorable family for all the sacrifices they made for the success of my academic career, especially to my dear mother. I also extend my sincere thanks and gratitude to all the professors of the Electronic Department and my dear friends. I also extend special thanks and unparalleled gratitude to Professor Supervisor **CHELLALI Safouane** and Professors **MERAH Lahcene** and **SAADI Ramdani**, who had a great crediton me that I will not forget as long as I live

GUESMIA Rabeh

Summary

List of figures

List of tables

General Introduction

General Introduction

The purpose of communication systems is to transmit data reliably over a noisy channel. The general schematic diagram of the communication system is shown in Figure 1.1. The source encoder removes redundant information from the source data. The channel encoder adds redundancy to the data so that reliable communication can be achieved over a noisy channel. The job of the channel decoder is to reproduce the data transmitted over the channel from the channel output. Finally, the source decoder reproduces the source data from the output of the channel decoder. Our main goal in this note will be to learn about the polar code.

Figure 01: Block diagram of communication systems

Chapitre I

General Description of a Digital Transmission Chain

I.1. Introduction

Man has always needed to communicate. In 1794 CLAUDE CHAPPIE built a telegraph between Paris and Lille. It was then possible to transmit a message over a long distance. Then, thanks to electricity, the telegraph was perfected: communication was established in the form of coded messages. The years 1832s and 1876s saw important inventions such as the Morse code thanks to SAMUEL MORSE and the telephone thanks to GRAHAM BELL where the voice could be transmitted.

Long-distance communication is now possible with a simple pair of copper wires and a power source. Today, thousands of simultaneous conversations can be transmitted at any distance.

Digital transmission systems carry information between a source and a recipient using a physical medium such as cable, fiber optics, propagation over a radio channel, etc. The signals carried may be either directly digital or analog (speech, image, etc.) but converted into digital form. The task of the transmission system is to carry the signal from the source to the recipient as reliably as possible.

The synoptic diagram of a digital transmission system is given in **figure I.1** where we limit ourselves to the basic functions. The source transmits a digital message in the form of binary elements. The encoder generally encompasses two fundamentally different functions: the first, called source coding, associates a suitable physical medium with the abstract elements emitted by the source, and the second, called channel coding, consists in introducing redundancy into the emitted signal to protect it against noise and interferers present on the transmission channel. The transmission channel includes the transmitter, the physical transmission medium on which the signal will be transmitted, and the receiver. Finally, on the receiver side, the source decoding and channel decoding functions are the respective inverses of the source coding and channel coding functions on the transmitter side.

I.2. The digital transmission chain

The block diagram of a digital transmission system is given in **figure I.1**we limit ourselves to the basic functions:

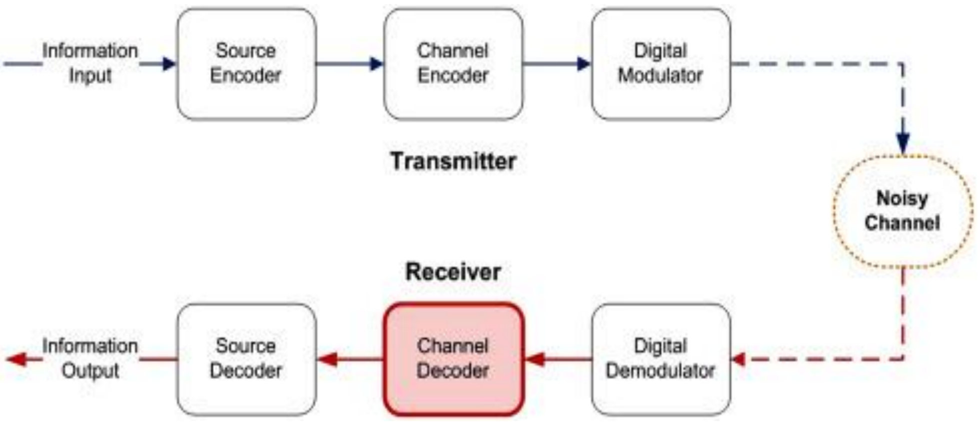

Figure I.1: General diagram of a digital transmission chain.

When detailing the sending part, the following blocks can be distinguished:

- **Information Input:**

The source can be an analog signal. Example: A Sound signal

- **Source Encoder**

The source encoder compresses the data into a minimum number of bits. This process helps in the effective utilization of the bandwidth. It removes the redundant bits

Example: unnecessaryexcessbits, i.e.

- **Channel Encoder**

The channel encoder does the coding for error correction. During the transmission of the signal, due to the noise in the channel, the signal may get altered, and hence to avoid this, the channel encoder adds some redundant bits to the transmitted data. These are the error-correcting bits.

- **Digital Modulator**

The signal to be transmitted is modulated here by a carrier. The signal is also converted to analog from the digital sequence, to make it travel through the channel or medium.

- **Channel**

The channel or a medium, allows the analog signal to transmit from the transmitter end to the receiver end.

- **Digital Demodulator**

This is the first step at the receiver end. The received signal is demodulated as well as converted again from analog to digital. The signal gets reconstructed here.

- **Channel Decoder**

The channel decoder, after detecting the sequence, does some error corrections. The distortions which might occur during the transmission, are corrected by adding some redundant bits. This addition of bits helps in the complete recovery of the original signal.

- **Source Decoder**

The resultant signal is once again digitized by sampling and quantizing so that the pure digital output is obtained without the loss of information. The source decoder recreates the source output.

- **Output Signal**

This is the output that is produced after the whole process. Example − the sound signal received.

I.3. Coding

I.3.1. Source coding

Source coding aims to represent the information to be transmitted in the most compact digital form possible see **Figure I.2**, by eliminating the redundancy contained in the source messages. On the one hand to digitize the data if they are analog (sound, image, video), on the other hand, to compress the digital data, to reduce the transmission rate or the storage volume.

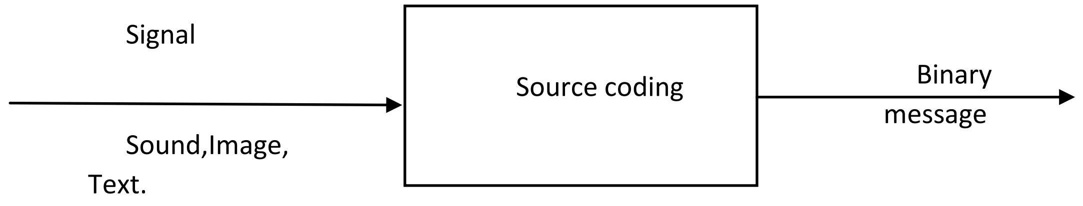

Figure I.2: Source coding

We will distinguish two types of compression:

- Lossless, or conservative, compression allows the original data to be recovered exactly after decompression. Ex: zip compression.

- Lossy or non-conservative, compression leads to a loss of information. It is used for objects intended to be perceived by a human, ensuring that the loss of information is not perceptible. This is the principle on which the perceptual encoders MPEG (audio and video) and JPEG (image) is based.

After digitization and coding, the digital message source is characterized by its bit rate D. defined as the number of bits transmitted per unit time.

The bit rate is equal to $D=\frac{1}{T}\,(bit/s)$

Where T is the duration of one bit.

The digitization of a signal is broken down into two successive operations, sampling, and quantization.

- Sampling: an operation performed on the signal to be transmitted to carry out the "analog/digital" conversion. It consists of substituting, for the original signal, a series of instantaneous values taken from the signal and regularly spaced out in time at precise, regularly spaced instants. On reception, a digital RII filter is used to recover the original signal. Figure (1.3)

- Quantization: to reconstitute the signal on reception, it is not necessary to transmit these pulses directly, it is sufficient to transmit information characterizing the amplitude of each of them. This operation consists of matching each sample amplitude with the closest amplitude of a discrete sequence of "standards" called "levels".

Each level of the quantization scale is characterized by a binary number.

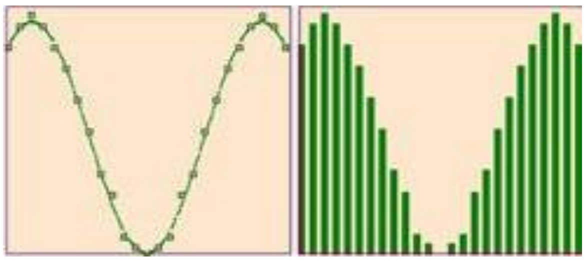

Figure I.3: Signal sampling and quantification.

I.3.2. Channel coding

Channel coding, also known as error-correcting coding, consists of protecting the binary messages provided by the source coding by introducing information redundancy. Bits are added to the original bits which depend on them. This redundancy can enable error detection and possibly error correction [4].

A. Causes of transmission errors

The causes of errors are numerous and depend mainly on:

- Transmission lines used.

- Type of modulation and coding used.

- Thermal noise due to electronic components can also cause errors if its level becomes quantifiable.

- Pulse noise, is an important source of error since a pulse lasting about ten milliseconds can induce several bits in error and the number of bits in error increases with the transmission speed.

B. Method of correcting transmission errors

These noises produce a large number of clustered errors and therefore error detection and correction systems have been developed to protect the integrity of the transmitted binary information. These systems are based on additional coding of the information at transmission and analysis of the message at reception.

Two strategies can be distinguished in case of error detection by the receiver:

- Either a request to retransmit the erroneous bits: this is the ARQ strategy.

- Or a correction by channel decoding, known as FEC.

In our project, we are interested in the first correction strategy (ARQ) See figure (1.4).

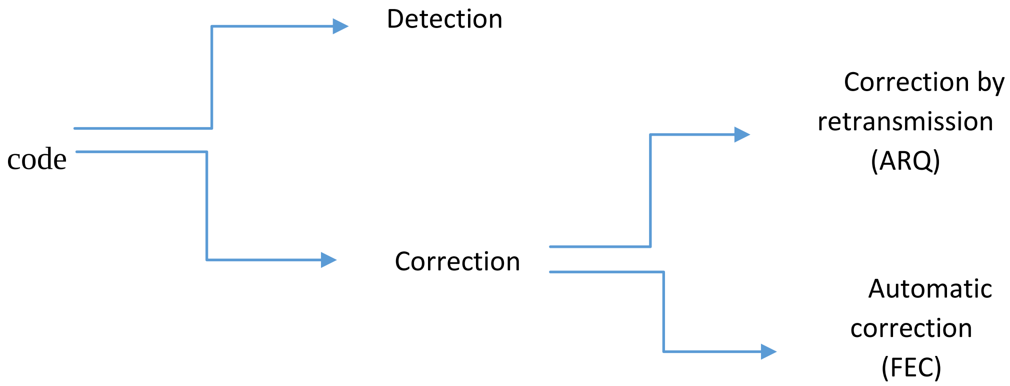

Figure I.4: Error correction method

Traditionally, two types of codes are distinguished, block and convolutional codes. Our study will focus on the Hamming block code.

C.1.Introduction to block codes HAMMING

The information from the source is put into fixed-length frames which will be transmitted: this is the message. The channel coding takes this message and makes it into a codeword.

The message consists of k characters, i.e. $2k$ possible messages. The codeword

The code word used will also have a fixed length of n characters, i.e. $2n$ possible code-words. With $n>k$ there will therefore be n - k characters in the code word which are redundant and will be used to handle any errors.

Moreover, $2n> 2k$ so a certain number of codewords do not correspond to a message but only to transmission errors. [5] We thus speak of a block code C (n, k) and the coding ratio or code efficiency R defined by: $R={}^{k}/{}_{n}$ **(I.2)**

C.2. Code weight

The weight of a codeword is by definition the number of non-zero characters contained in that word.

Example:

Table I.1. Example of different code weights

1011001011	Weight =6
0011010100	Weight =4
0000000000	Weight =null

C.3. Hamming distance

The Hamming distance is defined as the number of different bits between two words. It is denoted $dh\ (C, R)$ with C, R two different code words.

To maximize the detection (correction) capability of the code, the Hamming distance between the code words should be as large as possible. The objective is therefore to maximize the minimum distance between codewords, using the linearity property, which implies that the difference between two codewords is one codeword[6].

If a code is of minimum distance:

- It will be possible to detect all errors of weight less than or equal to $dmin$ - 1.

- We can correct t errors if we can associate (without risk of error) the codeword to erroneous receive the word; this can be done if t is strictly less than $dmin/2$.

In summary, if s denotes the number of detectable errors, and t the number of correctable errors, the link to the minimum distance $dmin$ is as follows:

$$d_{min} = s + 1 \quad \textbf{(I.3)}$$

$$dmin = 2t + 1 \quad \textbf{(I.4)}$$

C.4. generating a systematic linear block code

We will generate a systematic linear block codeword Cm from a message Xm such that:

$$\underbrace{X_m=[m_0 m_1 m_2 \ldots\ldots m_{k-1}]}_{k\text{Caractèresdumessage}}$$

$$C_m=[R_0 R_1 \ldots\ldots R_{n-k-1} m_0 m_1 \ldots\ldots m_{k-1}]$$

Either C_m= [$R_m X_m$] or R_m contains the control characters also called redundancy characters

C.5. Codeword generation

C.5.1. Codeword generation matrix

A matrix which generates the code word Cm from the message Xm is called a generation matrix, denoted G, such that:

$$C_m = G * X_m \qquad \textbf{(I.5)}$$

Turning to systematic codes, the matrix G is of the following form:

$$G[P_{(k,n-k)}, I_{(k,k)}] \qquad \textbf{(I.6)}$$

C.5.2 Unsystematic code

A non-systematic code noted:G = [P, M] where the matrix M is a matrix that shuffles the bits of the message to encrypt it. We can of course reduce to a systematic code by combining the rows to gather to involve the matrix Ik. Any block code can thus be reduced to the study of a systematic code. The essential part of the G matrix is therefore the P matrix called the parity matrix [7].

C.5.3 Encoder design principle

The principle of the technical implementation is therefore simple, it is sufficient to have two shift registers and an inverter (electronic switch), as shown in figure (I.5).

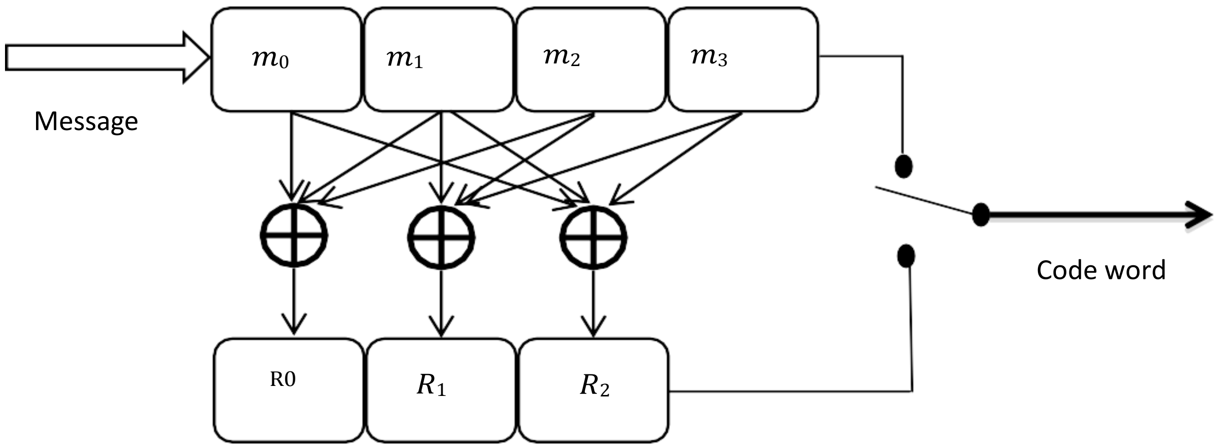

Figure I.5: Principle of the encoder.

- each parity bit is therefore expressed as$R_i = \sum_{j=1}^{k} m_j p_{ji}$
- In binary the Pj,i∈ {0; 1}.
- The Pi are obtained by simple binary addition. See table (1.2)

Table 1.2. Binary addition

$\oplus$	0	1
0	0	1
1	1	0

- We take as an example a code $C(7,4)$ $telque$:

- G= $\begin{bmatrix} 1\,0\,1\,1\,0\,0\,0 \\ 1\,1\,1\,0\,1\,0\,0 \\ 1\,1\,0\,0\,0\,1\,0 \end{bmatrix} \Rightarrow \begin{cases} R0 = m0 + m1 + m2 \\ R1 = m1 + m2 + m3 \\ R2 = m0 + m1 + m3 \end{cases}$

C.5.4 Parity control matrix

Let a systematic code of generating matrix G [P (k, n-k), Ik], the control matrix H will have the following form:

$$H=[I_{n-k}P^T] \qquad \textbf{(I.7)}$$

The relation (1.8) makes it possible to determine if a word of n bits noted r belongs or not to the code word, the idea is to define the set of words n bits orthogonal to the code word and to check if r is orthogonal to this set [4].

$$G*H^T=0 \qquad \textbf{(I.8)}$$

The relationship that controls whether a word belongs to the code or not is given as follows:

$$c*H^T=0 \qquad \textbf{(I.9)}$$

With c code word.

Exemple :

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

C.5.5 Syndrome

Consider the case where the sender transmits the code word Xm and the receiver receives the word Cm. We call syndrome, the vector :

$$S=C_m.H^T \qquad \textbf{(I.10)}$$

We Can also write :

S=X$_m.G.H^T$

For $Cm$to to be a codeword, the syndrome must be zero.

If the received code mode Cm is tainted with an error, the received Code will be the sum of the transmitted code and the error (equation (I.11).)

$$Y_m=C_m+E_m \qquad \textbf{(I.11)}$$

Where Ym is the error-ridden code word. The syndrome will be deduced by relation (I.12):

$$S=Y_m.\mathrm{H}^\mathrm{T} \qquad \textbf{(I.12)}$$

Let us take the case of a single error on thei^{th} bit of the error word. The syndrome is then identical to the i^{th} row of HT (or the ie column of H). It is possible to correct the error if all columns of H are distinct. The decoding strategy is then very simple: we compare the syndrome with the columns of H. If it is identical to one of them, we correct the corresponding bit of the received word [8].

I.4. Modulation

I.4.1. Introduction

A transmission with prior modification of the spectrum of the signal to be transmitted is called transposed band transmission or modulation. It generally uses two signals:

- The analog or digital message, called the modulating signal or message (LF)
- A carrier or sampling signal (HF) Modulation can be:
- Either a more or less direct transposition of the message spectrum to HF (amplitude modulation, frequency modulation).

- Or a radical modification of the signal itself using digital means, notably sampling (pulse modulation).
- Or a combination of the two previous techniques (Wide Band Code Division Multiple Access - W-CDMA).

Example:

- Shift keying modulation.
- Pulse modulation and PCM coding.
- CPM continuous phase modulation.

In our project, we will study CPM

I.4.2. General study of CPM

Continuous phase modulations (CPM) are a family of constant envelope phase modulations. They were introduced in the early 1980s by John B. Anderson and Carl- Erik Sandberg [6]. In addition to a constant envelope, these modulations have other properties that differentiate them from classical linear modulations. These properties include non-linearity with respect to the transmitted sequence and the lattice modeling of CPM signals. Modulations in which the phase of the complex envelope $z\ (t)$ evolves continuously with time but is not constant are called continuous phase modulations, CPM. Ensuring the continuity of the phase makes it possible to limit the spectral occupation of the modulated signal.

The general expression for the phase $\varphi\ (t)$ of signal (t), in the interval, is $[nT,\ (n + 1)\ T]$, is as follows:

$$\varphi\ (t)=2\pi\sum_{k=-\infty}^{n} a_k b_k q(t-kt) \quad \textbf{(I.13)}$$

Or the sequence (ak) the sequence of M-ary amplitudes associated with successive symbols. These symbols are uniformly distributed. The sequence (bk) is the sequence of modulation indices. Generally, it is constant and the modulation index b is fixed. It can vacyclicallyaly, and is referred to as multi-index modulation. The elementary pulse $q\ (t)$ is a normalized continuous waveform, which is often the integral of a function (t).

$$q(t)=\int_{-\infty}^{t} g(r)dr \quad \textbf{(I.14)}$$

The function $2\pi bq\ (t)$ is called the phase elementary pulse, and it is noted hereafter $\varphi\ (t)$. The function $g\ (t)$ is called the frequency elementary pulse.

The most commonly used CPM modulations are described in the following sections. They are the CPFSK continuous phase frequency modulations, with in particular the MSK modulations, and the binary GMSK modulations. These modulations have a constant envelope [2].

- CPFSK is a continuous phase frequency hopping modulation, which associates each symbol with a frequency. The frequency change is made while maintaining phase continuity.
- MSK is a CPFSK modulation with a modulation index (5 = 0.5), its name comes from the use of the minimum deviation to obtain orthogonal frequencies
- GMSK modulation is an MSK modulation to which a Gaussian low-pass filter has been added to reduce the spectral occupancy of the modulated signal. It is therefore a continuous phase frequency modulation of the index (5 = 1/2).

I.4.3. Transmission channel

I.4.3.1. Definition

Before any transmission chain is designed, and in particular the selection of the waveform, the nature, and properties of the channel used must be studied. Noise strength, channel type, and stationarity are parameters that must be known a priori to make an effective choice of the waveform. This knowledge then allows the capacity of the channel to be evaluated given the waveform adopted. From an operator's point of view, information on propagation conditions is essential for an initial assessment of the system's capacity and the quality and nature of the service it can offer.

The transmission channel represents the link between the transmitter and the receiver and can be of different types depending on the type of data it carries. The transmission channel is characterized by its capacity and bandwidth.

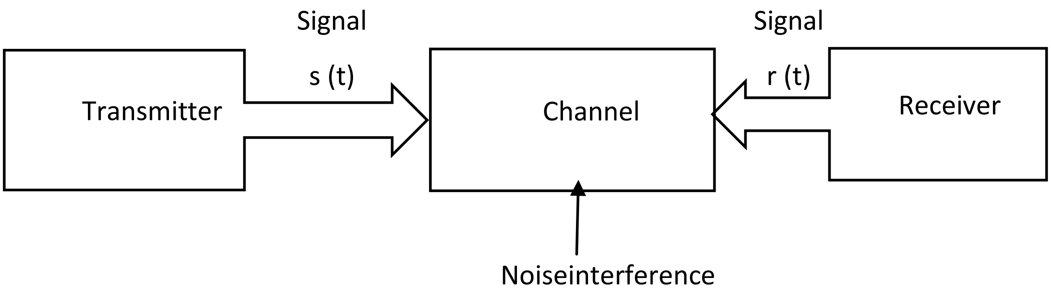

Figure I.6. Transmission channel.

The sources of interference are diverse and depend mainly on the environment in which the transmission channel is located (figure (I.6)). The main types of noise are galactic noise between 20 MHz and 200 MHz due to radiation from various energy sources in space; atmospheric noise up to 20 MHz induced by thunderbolts, industrial noise, urban noise, micro outages corresponding to short interruptions of the signal, phase jumps and flickers related to sudden variations in phase or slow variations caused by electrical power supplies; crosstalk when several links are routed through the same cable [9].

I.4.3.2. Capacity of a noisy digital channel

A formula specifies the capacity of the transmission channel for a digital signal passing through a real, and therefore noisy, line

$$D = B \log_2 1 + {}^{S}/_{N} \qquad \textbf{(I.15)}$$

Or

- : Bit rate ($bits/S$).
- : Bandwidth (HZ).
- $S/$: Signal to noise ratio (W/W).

There are several theoretical models of the transmission channel according to the most frequent types of errors:

I.4.3.3. Symmetric binary channel

The part of the link from the input of the transmitter to the output of the receiver can be considered as a binary channel, which is characterized by input and output of finite and equal bitstreams (0, 1). The physical phenomena at work in the real channel are therefore left aside, to connect simply to the binary transformation between the input and the output.

$$P\left(\frac{R_0}{S_1}\right) = P\left(\frac{R_1}{S_0}\right) \qquad \textbf{(I.16)}$$

Where Ri and Si represent the transmit and receive event of the binary element respectively

At the output of the transmission channel, the noisy signal is demodulated to obtain a sequence of binary elements. It is possible to represent all the parts modulation, transmission channel, and demodulation by a binary channel. The principle of a binary channel, represented in figure (I.7), is to associate to each input bit a certain probability that the received bit is erroneous. This depends on the one hand on the errors generated by the propagation channel and on the other hand on the errors due to the demodulation of the signal.

A Description of a binary channel

The simplest model is the binary symmetric channel (BSC). A BSC is defined by its probability of error, denoted P. The value of this probability, which depends on the channel and the modulation, corresponds to the (BER) obtained at the output of the demodulator. If we denote c and y as the input and output elements of the BSC, then the probability that the received symbol is erroneous is equal to the P equation (I.17) and conversely, the probability that the received symbol is correct is 1-P equation (I.18):

$$P_r(y=0,c=1)=P_r\,(y=1,c=0)=P \qquad \textbf{(I.17)}$$

$$P_r(y=0,c=0)=P_r\,(y=1,c=1)=1-P \qquad \textbf{(I.18)}$$

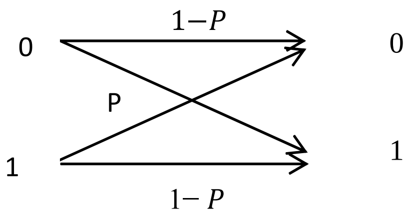

Figure I.7: Symmetric binary channel.

I.4.3.4. AWGN channel

An Additive White Gaussian noise (AWGN) transmission channel is shown in Figure (I.8). It consists of the addition of awhite Gaussian noise, of bilateral power spectral density DSP given by equation (I.19) [10].

$$S_b(\text{f})=\frac{N^0}{2} \qquad \textbf{(I.19)}$$

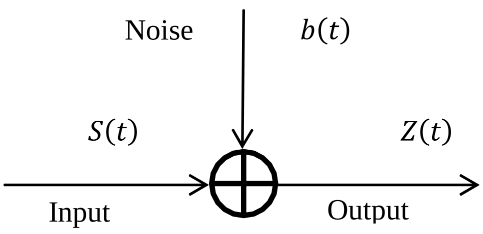

Figure I.8. The additive white Gaussian noise channel AWGN

I.5. Conclusion

Information transmitted between a source and a recipient is always threatened by interference, disturbances, and transmission errors located along with the transmission medium.

Therefore, a theory of detection and correction of these transmission errors is produced by researchers to address these problems.

Chapitre II

Encoding and Decoding in Polar Codes

II.1. Introduction

Polar codes were introduced by Erdal Arıkan in 2009 and they provide the first deterministic construction of capacity-achieving codes for binary memoryless symmetric (BMS) channels . They are the culmination of Arıkan's research into the computational cutoff rate of sequential decoding. [1]

Polar codes are a class of capacity-achieving codes introduced [2]. The main motivation for the introduction of polar codes was theoretical, namely, to show the existence of a family of codes that are provably capacity-achieving and have low complexity encoding and decoding algorithms.

Polar coding owes its analytical tractability to its recursive structure. This recursive structure also leads to low-complexity encoding and decoding algorithms for polar coding. It is shown in [3], [2] that asymptotically, as a function of code block length N, polar codes can be encoded in complexity O(N log N), decoded using a successive-cancellation decoder in complexity O(N log N), while achieving an overall block-decoding error probability that is bounded as O(2−Nβ) for any fixed $\beta < 1/2$ and fixed code rate below the channel capacity

II.2 The Philosophy of Polar Codes

Consider the discrete memoryless channel shown in Figure. II.1 where it is seen that the symbol u is passed through the discrete memoryless channel indicated by the symbol W. The symbol at the input of the discrete memoryless channel can be considered as generated by a discrete random variable $\tilde{U}_1$, similarly the symbol at the output of the channel can be considered as generated by another discrete random variable $\tilde{Y}$. [5]

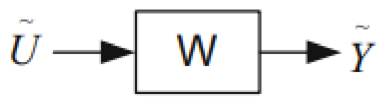

Figure. II.1 Discrete memoryless channel with random variables

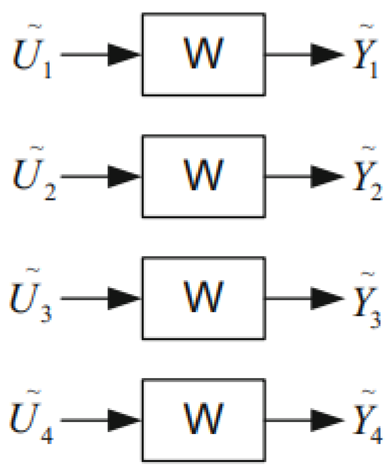

Figure. II.2 Parallel discrete memoryless channels

Consider the transmission of 4 different symbols $[u_1u_2u_3u_4]$ through the channel in a serial manner. In this case, we can assume that these 4 symbols are generated by IID random variables, and we can think the transmission of each symbol through the channel separately as in Figure. II.2.

Let $\bar{U} = [\tilde{U}_1\bar{U}_2\bar{U}_3\bar{U}_4]$ and $\bar{Y} = [\tilde{Y}_1\tilde{Y}_2\bar{Y}_3\tilde{Y}_4]$, then the mutual information between $\bar{U}$ and $\bar{Y}$ can be written as

$$I(\bar{U};\bar{Y}) = I(\tilde{U}_1;\bar{Y}) + I(\tilde{U}_2;\bar{Y} \mid \tilde{U}_1) + I(\tilde{U}_3;\bar{Y} \mid \tilde{U}_1,\tilde{U}_2) + I(\tilde{U}_4;\bar{Y} \mid \bar{U}_1,\tilde{U}_2,\tilde{U}_3)$$

If $\tilde{U}_1$ is independent of $\tilde{Y}_1,\bar{Y}_2,\bar{Y}_3,\bar{Y}_4$, then we can write that

$$I(\bar{U}_1;\bar{Y}) = I(\bar{U}_1;\bar{Y}_1) \qquad \text{(II.1)}$$

II.3 Encoding

II.3.1 Polar Encoding Operation

Let N be the length of information sequence. Consider the transmission of two information bits u1 and u2. The simplest polar encoder structure for N = 2, i.e., for ū = [u1u2], is depicted in Figure. II.3.

From Figure. II.3, we can write that $x_1 = u_1 \oplus u_2$ (II.2)

$x_2 = u_2.$ (II.3)

For the information word $\bar{u} = [u_1u_2]$, after polar encoding operation, the obtained code-word is $\bar{x} = [x_1x_2]$ where $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$. The relation between $\bar{x}$ and $\bar{u}$ for $N = 2$ can mathematically be expressed as

$\bar{x} = \bar{u}G$ (II.4)

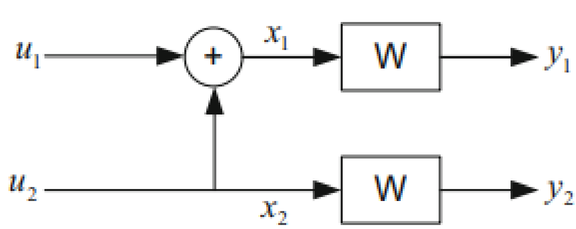

Figure. II.3 Polar encoder for N= 2 with discrete memoryless channel

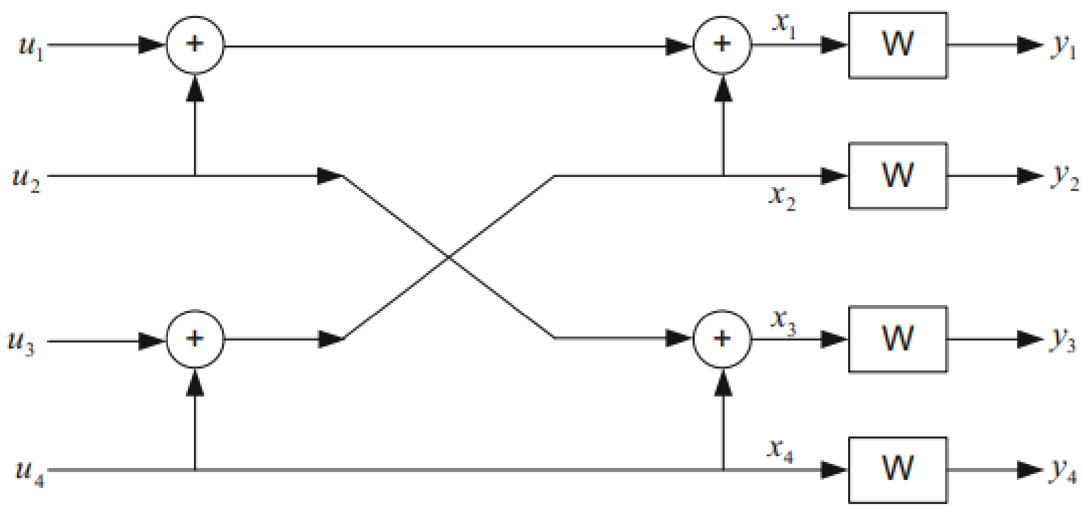

Figure. II.4 Polar encoder for N 4 with discrete memoryless channels

where G is the generator matrix for $N = 2$ and it is equal to

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

For $N = 4$, polar encoder structure is depicted in Fig. 2.4. From Figure. II.4, we get

$$\begin{aligned} x_1 &= u_1 \oplus u_2 \oplus u_3 \oplus u_4 \\ x_2 &= u_3 \oplus u_4 \\ x_3 &= u_2 \oplus u_4 \\ x_4 &= u_4 \end{aligned}$$

which can be written as $\bar{x} = \bar{u} \times G$ (II.4)

Where $\bar{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]\ \bar{u} = [u_1 u_2 u_3 u_4]$

and the generator matrix G equals to $G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$.

The encoder structure in Figure. II.4 can be redrawn using straight lines as in Figure. II.5

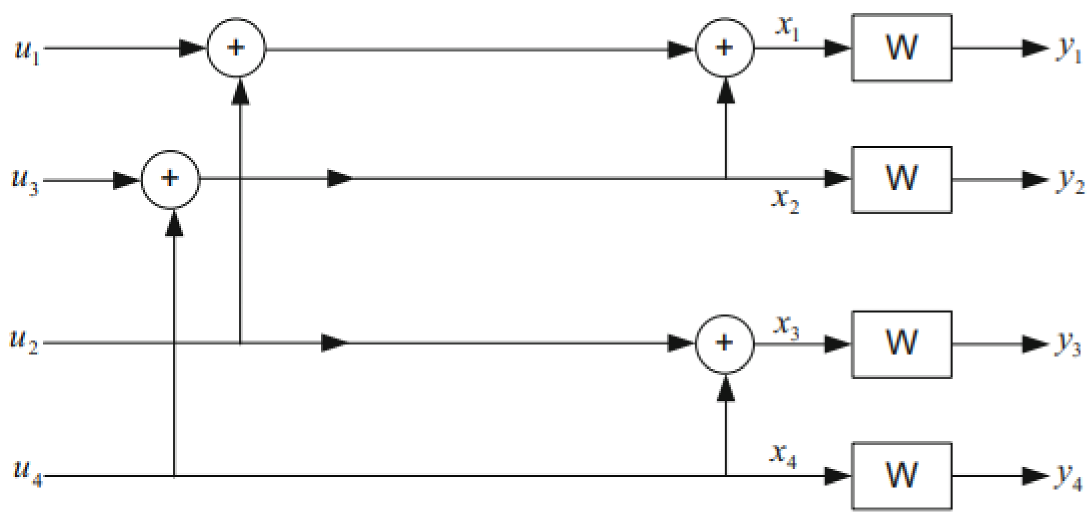

Figure. II.5 Redrawn polar encoder for N=4 with discrete memoryless channels

.

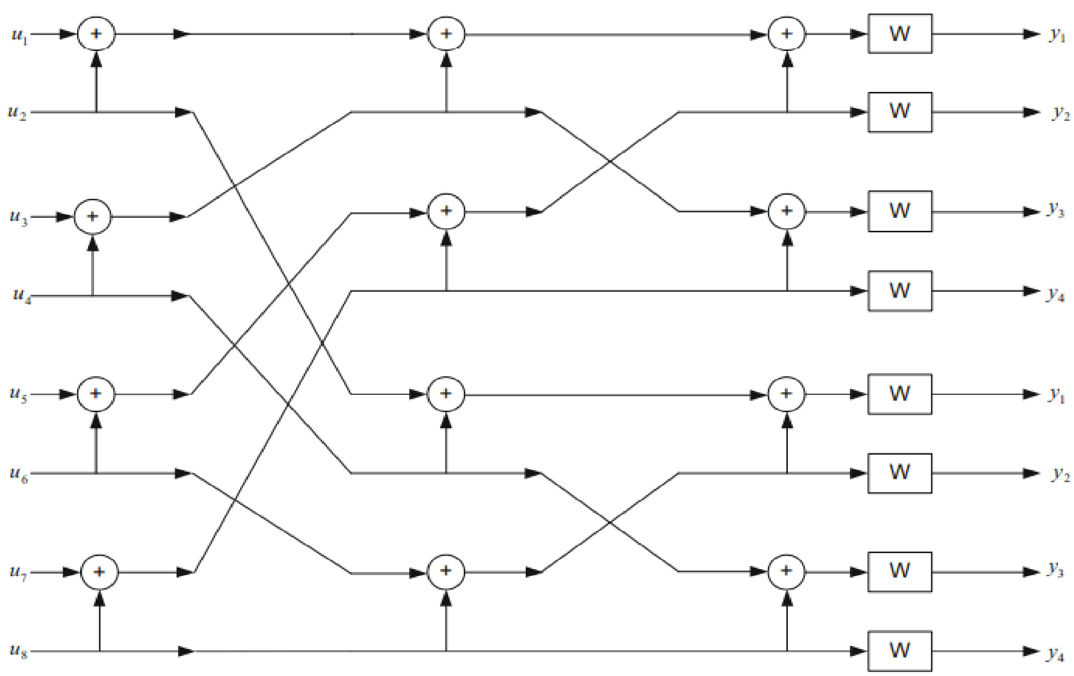

Figure. II.6 Polar encoder for N =8 with discrete memoryless channels

Although the encoder structure of Figure. II.5 seems to be different from the one in Figure. II.4, they are the same of each other and the relation between $\bar{u}$ and $\bar{x}$ stays the same in both structures.

Note that, in Fig. 2.4 the input sequence is $\bar{u} = [u_1 u_2 u_3 u_4]$ whereas in Figure. II.5 the input sequence

$$\bar{u} = [u_1 u_3 u_2 u_4]$$

For N=8 polar encoder structure is depicted in Figure. II.7.

From Fig. 2.6, we get

$$x_1 = u_1 \oplus u_2 \oplus u_3 \oplus u_4 \oplus u_5 \oplus u_6 \oplus u_7 \oplus u_8$$

$$\begin{aligned} x_2 &= u_5 \oplus u_6 \oplus u_7 \oplus u_8 \\ x_3 &= u_3 \oplus u_4 \oplus u_7 \oplus u_8 \\ x_4 &= u_7 \oplus u_8 \\ x_5 &= u_2 \oplus u_4 \oplus u_6 \oplus u_8 \\ x_6 &= u_6 \oplus u_8 \\ x_7 &= u_4 \oplus u_8 \\ x_8 &= u_8 \end{aligned}$$

which can be written as

$$\bar{x} = \bar{u} \times G$$

where

$$\bar{x} = [x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8] \; \bar{u} = [u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8]$$

and the generator matrix G equals to

$$G_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Note that, in Figure. II.6 the input sequence is $\bar{u} = [u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8]$ whereas in Figure. II.5 the input sequence is $\bar{u} =$ [u1 u5 u3 u7 u2 u6 u4 u8]

II.4 Decoding

II.4.1 Kernel Encoder and Decoder Units of the Polar Codes

In this section, we will inspect the kernel encoding and decoding units of the polar codes and derive the fundamental formulas necessary for the decoding operation. The kernel unit, which can also be considered as the polar encoder unit with the smallest length code-words at its output, is depicted in Figure. II.7. and it is repeatedly used in polar encoder structures. From

Figure. II.7 we can write that In Figure. II.7. [a b] is the data-word encoded, and [c d] is the code-word obtained from encoding of [a b]. The code bits are transmitted through the identical channels indicated by W. For the moment, let's assume that the bits c and d are transmitted through a discrete memoryless channel, and [y1 y2] are the signals at the receiver side At the decoder side, the flow of the signals change direction as shown in Figure. II.8. where cˆ and dˆ are the estimated code bits. Recovering data bits from the code-bits at the receiver side is called decoding operation.

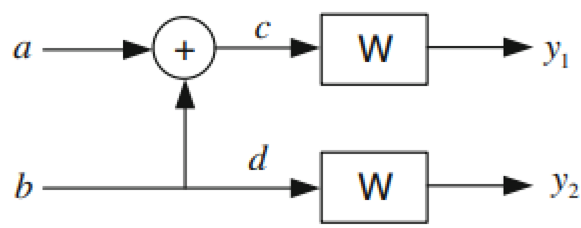

Figure. II. 7 Kernel encoder unit with two DMCs

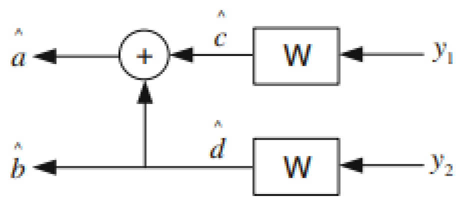

Figure. II.8. Kernel decoder unit

II.4.2 Decoding Tree for the Successive Cancelation Decoding of Polar Codes

In this section, we will explain the successive cancelation decoding of polar codes using a decoding tree [6].

The signal flow for the decoding of information bits is depicted in Figure. II.9. The decoding of the information bit u1 is depicted Figure. II.10 with bold lines. The decoding path shown in Figure. II.10 can be expressed using a tree as in Figure. II.11. In Figure. II.20, the likelihood ratios for gi,i 1,..., 4 can be calculated as

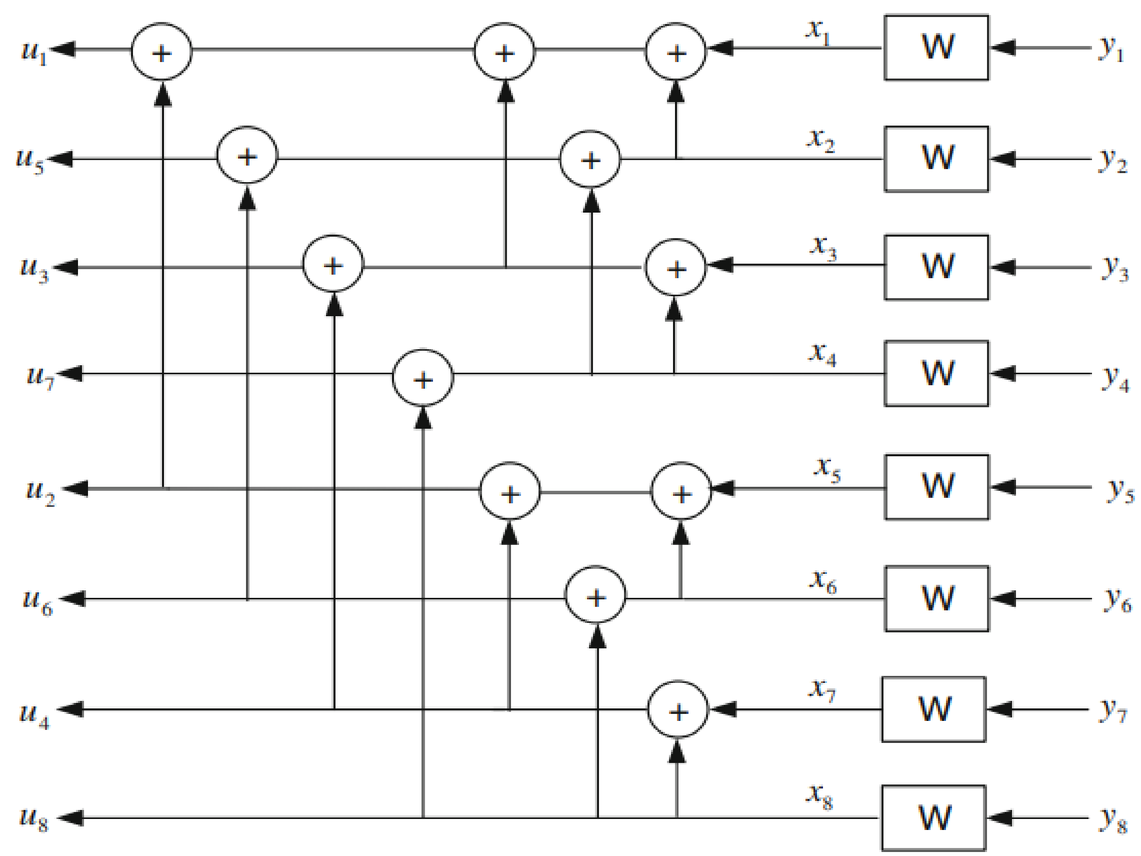

Figure. II.9 Polar decoder for N= 8

$$LR(g_1) = \frac{1 + LR(x_1)LR(x_2)}{LR(x_1) + LR(x_2)} \; LR(g_2) = \frac{1 + LR(x_3)LR(x_4)}{LR(x_3) + LR(x_4)}$$
$$LR(g_3) = \frac{1 + LR(x_5)LR(x_6)}{LR(x_5) + LR(x_6)} \; LR(g_4) = \frac{1 + LR(x_7)LR(x_8)}{LR(x_7) + LR(x_8)}.$$

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = \frac{1 + LR(g_1)LR(g_2)}{LR(g_1) + LR(g_2)} \; LR(h_2) = \frac{1 + LR(g_3)LR(g_4)}{LR(g_3) + LR(g_4)}.$$

The likelihood ratio for information bit u_1 is calculated as

$$LR(u_1) = \frac{1+LR(h_1)LR(h_2)}{LR(h_1)+LR(h_2)}. \qquad \text{(II.5)}$$

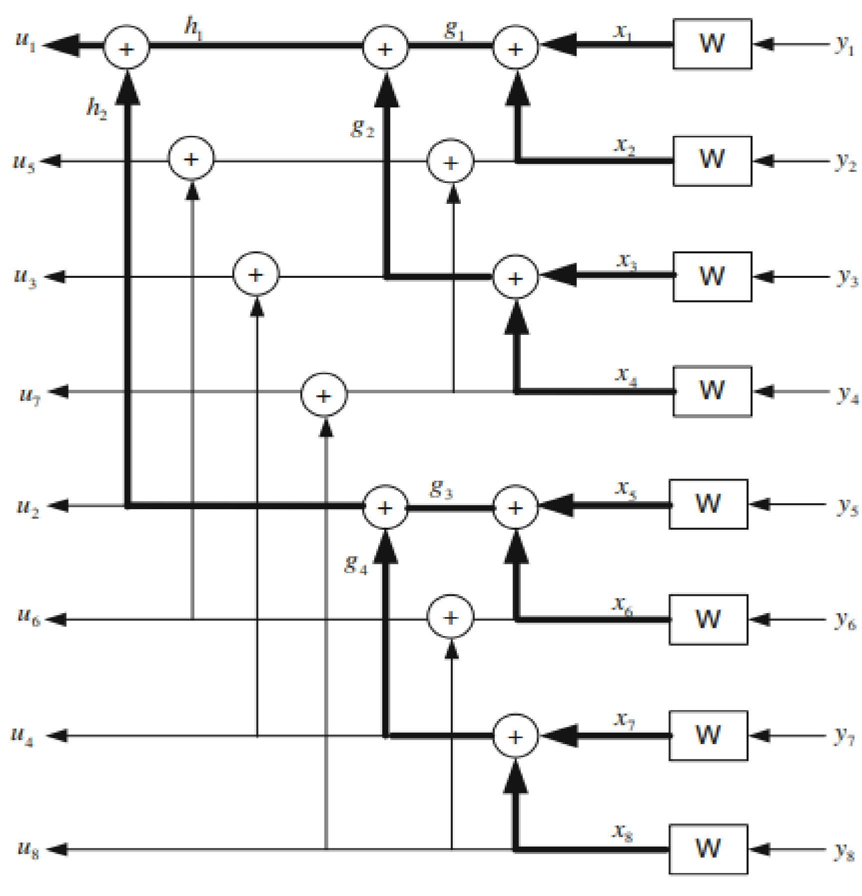

Figure. II.10 Decoding path of u1 when N= 8

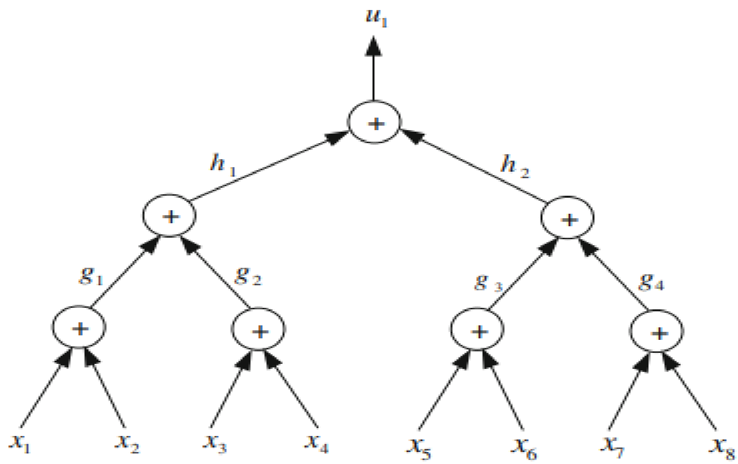

Figure. II.11 Decoding tree of u1 when N= 8

Once u1 is decoded, we proceed with the decoding of u2. The decoding path for u2 is shown with the bold lines in Figure. II.12. The decoding path shown in Figure. II.12 can be expressed using a tree as in Figure. II.13. In Figure. II.13, the likelihood ratios for gi,i 1, ⋯ , 4 can be calculated as

$$LR(g_1) = \frac{1 + LR(x_1)LR(x_2)}{LR(x_1) + LR(x_2)} \; LR(g_2) = \frac{1 + LR(x_3)LR(x_4)}{LR(x_3) + LR(x_4)}$$
$$LR(g_3) = \frac{1 + LR(x_5)LR(x_6)}{LR(x_5) + LR(x_6)} \; LR(g_4) = \frac{1 + LR(x_7)LR(x_8)}{LR(x_7) + LR(x_8)}.$$

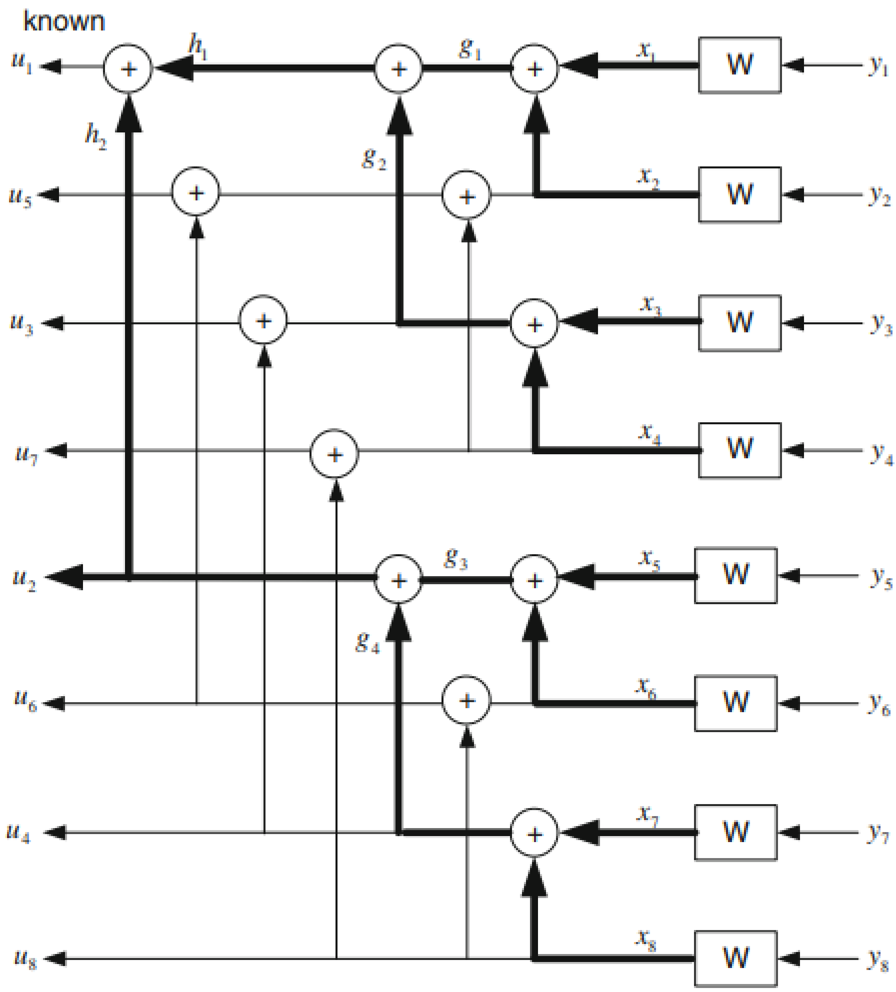

Figure. II.12 Decoding path of u2 when N =8

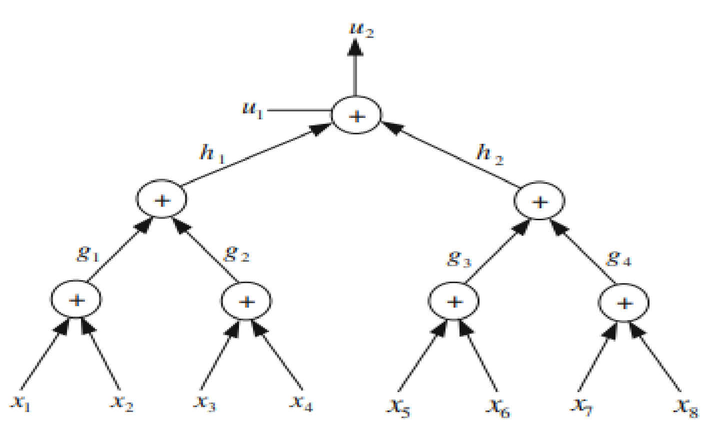

Figure. II.13 Decoding tree of u2 when N =8

The likelihood ratios for h1 and h2 can be calculated as

$$LR(h_1) = \frac{1 + LR(g_1)LR(g_2)}{LR(g_1) + LR(g_2)} \quad LR(h_2) = \frac{1 + LR(g_3)LR(g_4)}{LR(g_3) + LR(g_4)}$$

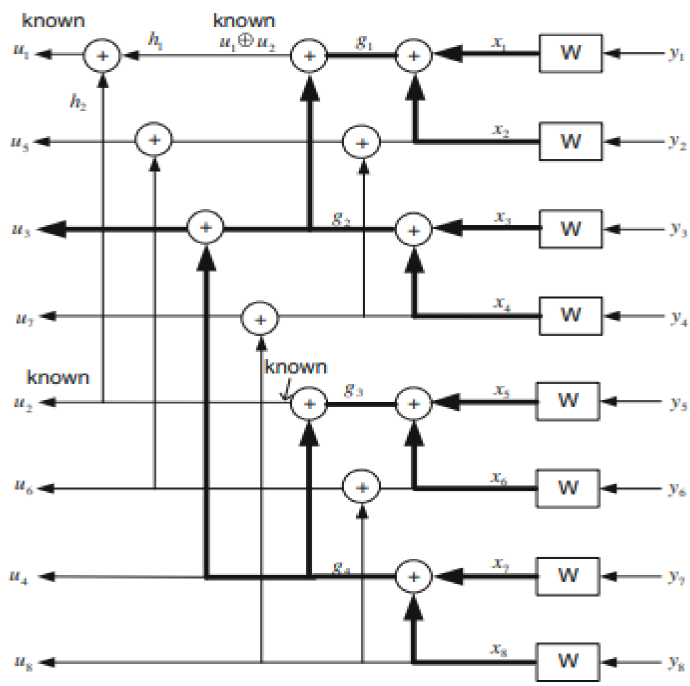

Figure. II.14 Decoding path of u3 when N =8

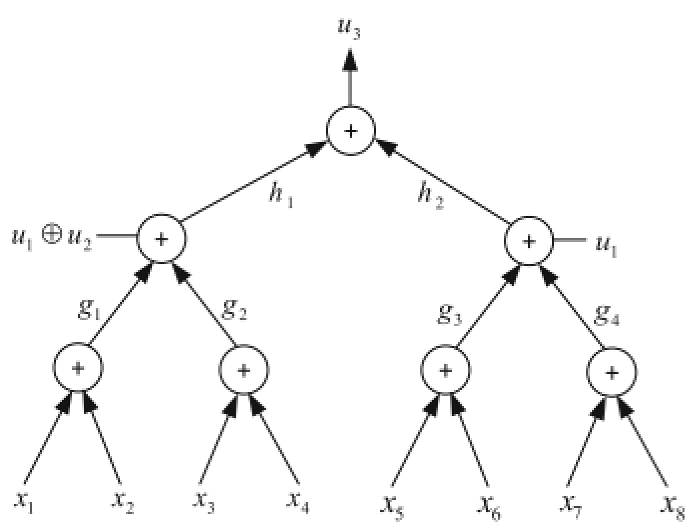

Figure. II.15 Decoding tree of u3 when N =8

Using the previously solved bit u1, the likelihood ratio for information bit u2 is calculated as

$$LR(u_1) = [LR(h_1)]^{1-2u_1} LR(h_2). \qquad \text{(II.6)}$$

After determining u_2, we can decode the information bit u_3 using the path shown in bold lines in Figure. II.14.

The decoding path shown in Fig. 2.14 can be expressed using a tree as in Figure. II.15. In Figure. II.15 , the likelihood ratios for $g_i, i = 1, \dots ,4$ can be calculated as

$$LR(g_1) = \frac{1 + LR(x_1)LR(x_2)}{LR(x_1) + LR(x_2)} \; LR(g_2) = \frac{1 + LR(x_3)LR(x_4)}{LR(x_3) + LR(x_4)}$$
$$LR(g_3) = \frac{1 + LR(x_5)LR(x_6)}{LR(x_5) + LR(x_6)} \; LR(g_4) = \frac{1 + LR(x_7)LR(x_8)}{LR(x_7) + LR(x_8)}.$$

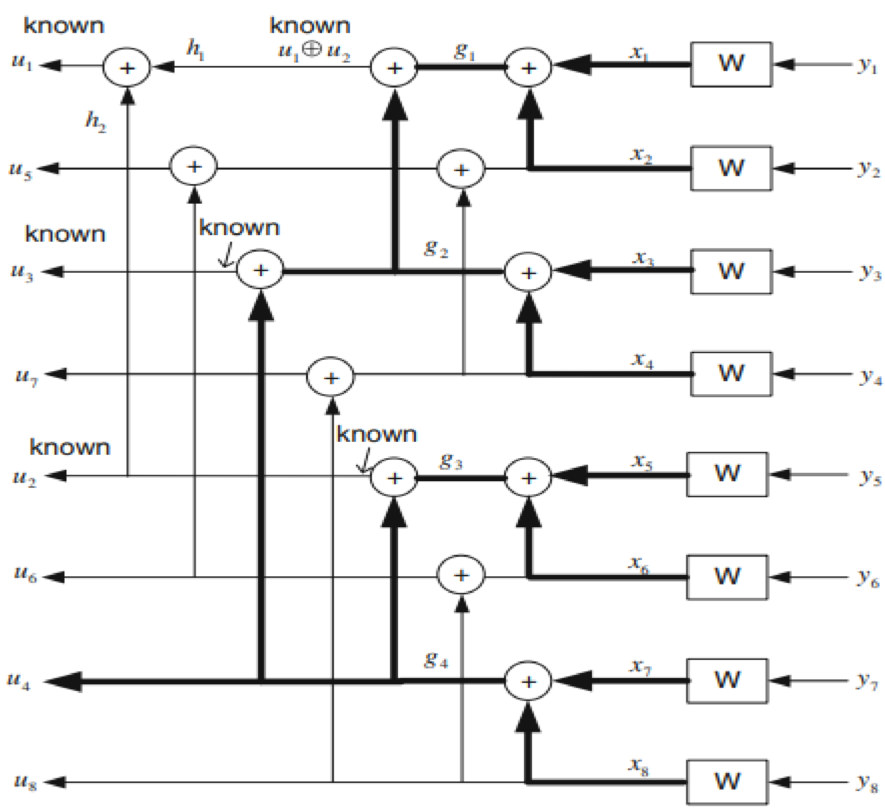

Figure. II.16 Decoding path of u4 when N =8

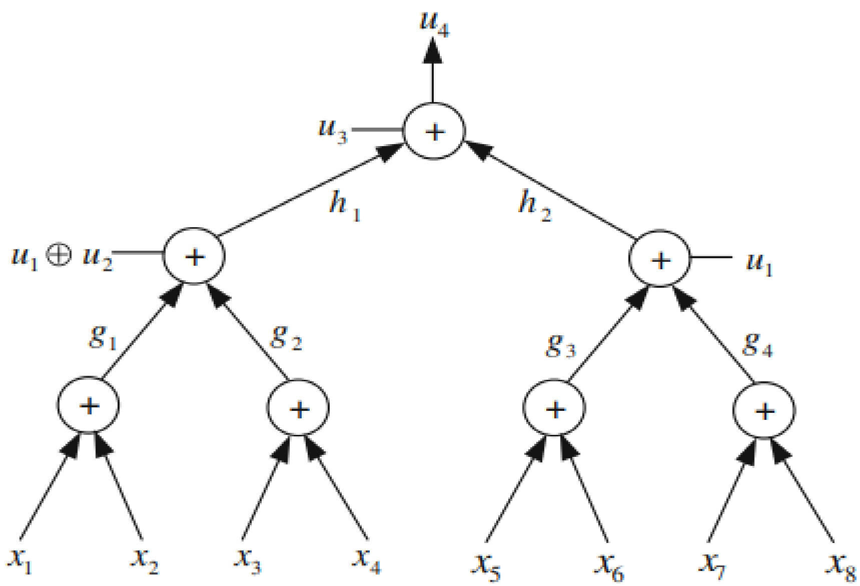

Figure. II.17 Decoding tree of u4 when N= 8

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = [LR(g_1)]^{1-2(u_1 \oplus u_2)} LR(g_2) \; LR(h_2) = [LR(g_3)]^{1-2u_1} LR(g_4)$$

The likelihood ratio for information bit u_3 is calculated as

$$LR(u_3) = \frac{1+LR(h_1)LR(h_2)}{LR(h_1)+LR(h_2)}. \qquad \text{(II.7)}$$

After deciding on the value of u_3, we proceed with the decoding of u_4. The decoding path of u_4 is shown with bold lines in Figure. II.16

The decoding path shown in Figure. II.16 can be expressed using a tree as in Figure. II.17

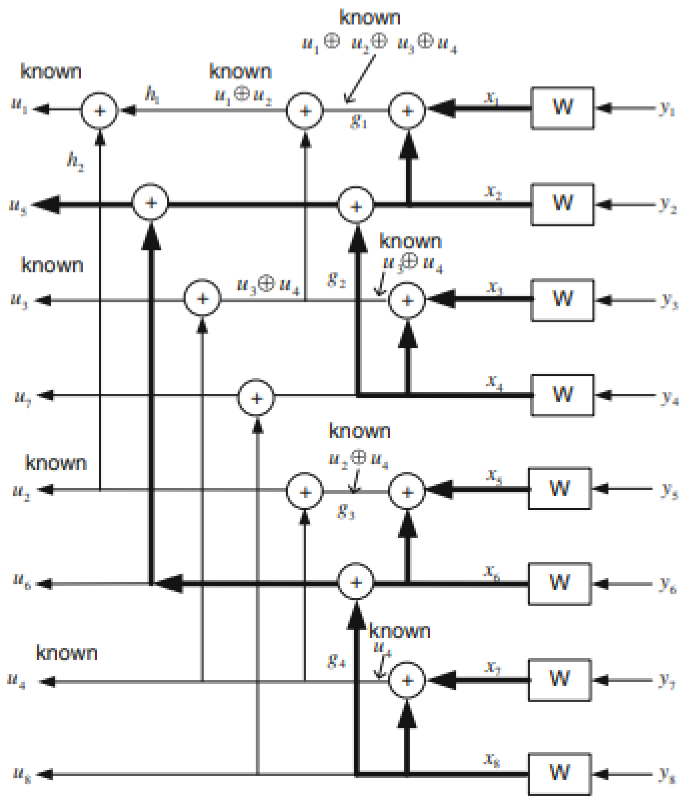

Figure. II.18 Decoding path of u5 when N =8

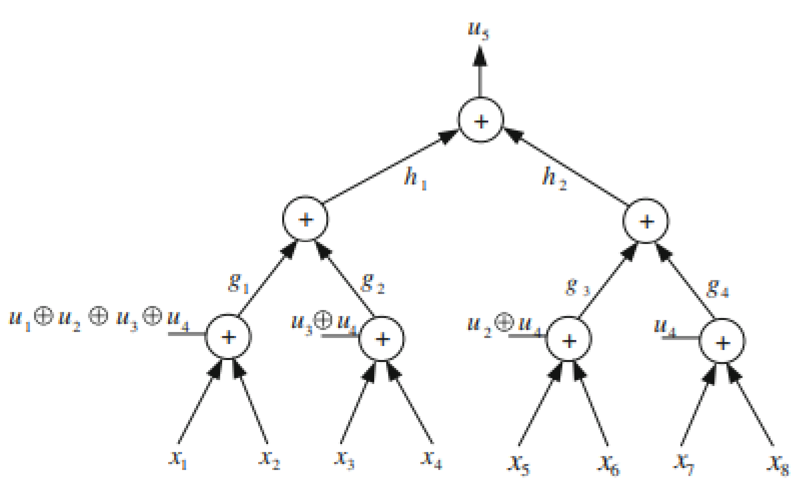

Figure. II.19 Decoding tree of u5 when N =8

In Figure. II.17, the likelihood ratios for $g_i, i = 1, \dots, 4$ can be calculated as

$$LR(g_1) = \frac{1 + LR(x_1)LR(x_2)}{LR(x_1) + LR(x_2)} \quad LR(g_2) = \frac{1 + LR(x_3)LR(x_4)}{LR(x_3) + LR(x_4)}$$
$$LR(g_3) = \frac{1 + LR(x_5)LR(x_6)}{LR(x_5) + LR(x_6)} \quad LR(g_4) = \frac{1 + LR(x_7)LR(x_8)}{LR(x_7) + LR(x_8)}$$

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = [LR(g_1)]^{1-2(u_1 \oplus u_2)} LR(g_2) \quad LR(h_2) = [LR(g_3)]^{1-2u_1} LR(g_4)$$

The likelihood ratio for information bit u_4 is calculated as

$$LR(u_4) = [LR(h_1)]^{1-2u_3} LR(h_2) \qquad \text{(II.8)}$$

After deciding on the value of u_4, we proceed with the decoding of the information bit u_5. The decoding path of u_5 is shown with bold lines in Figure. II.18.

The decoding path shown in Fig. 2.18 can be expressed using a tree as in Figure. II.19. In Figure. II.19, the likelihood ratios for $g_i, i = 1, \dots, 4$ can be calculated as

$$LR(g_1) = [LR(x_1)]^{1-2(u_1 \oplus u_2 \oplus u_3 \oplus u_4)} LR(x_2) \quad LR(g_2) = [LR(x_3)]^{1-2(u_3 \oplus u_4)} LR(x_4)$$
$$LR(g_3) = [LR(x_5)]^{1-2(u_2 \oplus u_4)} LR(x_6) \quad LR(g_4) = [LR(x_7)]^{1-2u_4} LR(x_8).$$

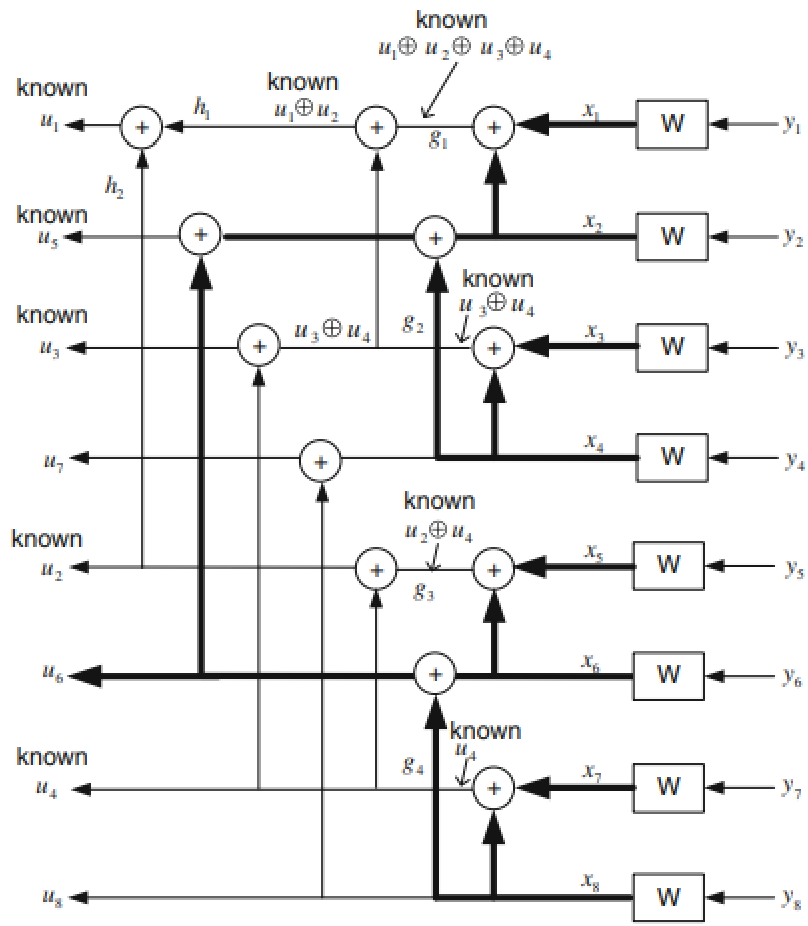

Figure. II.20 Decoding path of u6 when N =8

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = \frac{1 + LR(g_1)LR(g_2)}{LR(g_1) + LR(g_2)} \; LR(h_2) = \frac{1 + LR(g_3)LR(g_4)}{LR(g_3) + LR(g_4)}.$$

The likelihood ratio for information bit u_5 is calculated as

$$LR(u_5) = \frac{1+LR(h_1)LR(h_2)}{LR(h_1)+LR(h_2)} \qquad \text{(II.9)}$$

After deciding on the value of u_5, we proceed with the decoding of the information bit u_6. The decoding path of u_6 is shown with bold lines in Figure. II.20.

The decoding path shown in Fig. 2.20 can be expressed using a tree as in Figure. II.21.

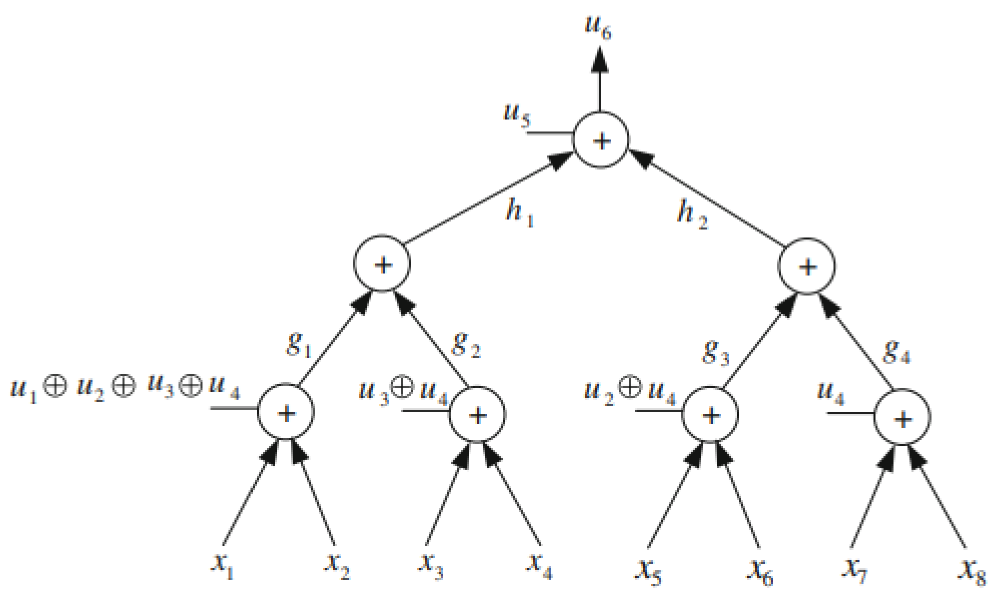

Figure. II.21 Decoding path of u6 when N =8

In Figure. II.21, the likelihood ratios for $g_i, i = 1, \dots, 4$ can be calculated as

$$LR(g_1) = [LR(x_1)]^{1-2(u_1 \oplus u_2 \oplus u_3 \oplus u_4)} LR(x_2)\ LR(g_2) = [LR(x_3)]^{1-2(u_3 \oplus u_4)} LR(x_4)$$
$$LR(g_3) = [LR(x_5)]^{1-2(u_2 \oplus u_4)} LR(x_6)\ LR(g_4) = [LR(x_7)]^{1-2u_4} LR(x_8).$$

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = \frac{1 + LR(g_1)LR(g_2)}{LR(g_1) + LR(g_2)}\ LR(h_2) = \frac{1 + LR(g_3)LR(g_4)}{LR(g_3) + LR(g_4)}.$$

The likelihood ratio for information bit u_6 is calculated as

$$LR(u_6) = [LR(h_1)]^{1-2u_5} LR(h_2) \qquad \text{(II.10)}$$

After deciding on the value of u6, we proceed with the decoding of the information bit u7. The decoding path of u7 is shown with bold lines in Figure. II.22. The decoding path shown in Figure. II.22 can be expressed using a tree as in Figure. II.23. In Figure. II.23, the likelihood ratios for gi,i 1,..., 4 can be calculated as

$$LR(g_1) = [LR(x_1)]^{1-2(u_1 \oplus u_2 \oplus u_3 \oplus u_4)} LR(x_2)\ LR(g_2) = [LR(x_3)]^{1-2(u_3 \oplus u_4)} LR(x_4)$$
$$LR(g_3) = [LR(x_5)]^{1-2(u_2 \oplus u_4)} LR(x_6)\ LR(g_4) = [LR(x_7)]^{1-2u_4} LR(x_8).$$

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = [LR(g_1)]^{1-2(u_5 \oplus u_6)} LR(g_2)\ LR(h_2) = [LR(g_3)]^{1-2u_6} LR(g_4)$$

The likelihood ratio for information bit u_7 is calculated as

$$LR(u_7) = \frac{1+LR(h_1)LR(h_2)}{LR(h_1)+LR(h_2)}. \qquad \text{(II.11)}$$

After deciding on the value of u_7, we proceed with the decoding of the information bit u_8. The decoding path of u_8 is shown with bold lines in Figure. II.24.

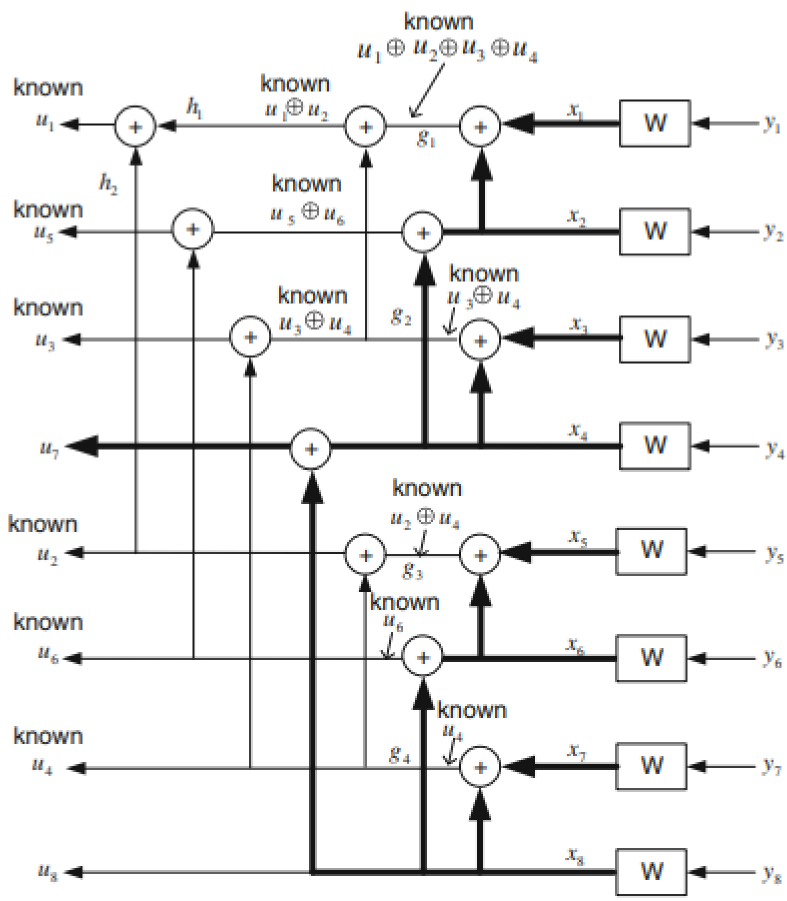

Figure. II.22 Decoding path of u7 when N =8

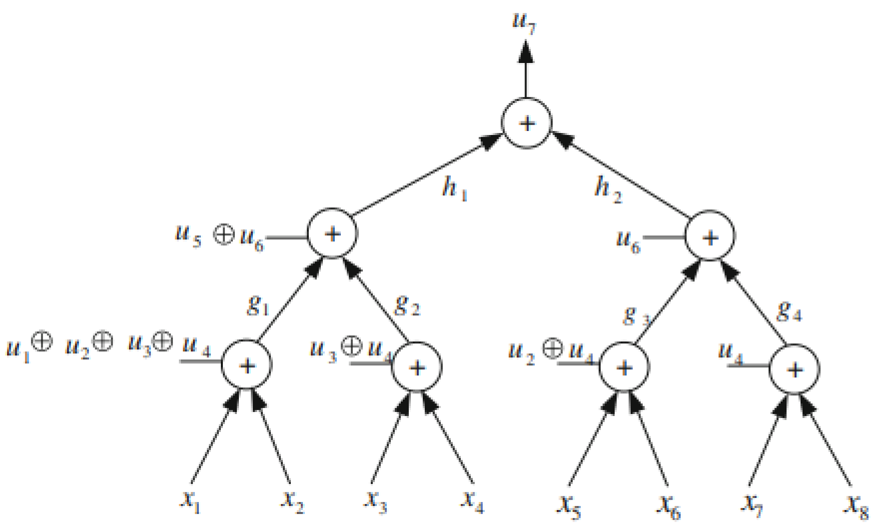

Figure. II.23 Decoding tree for u7 when N =8

The decoding path shown in Fig. 2.24 can be expressed using a tree as in Figure. II.25 In Figure. II.25, the likelihood ratios for $g_i, i = 1, \dots, 4$ can be calculated as

$$LR(g_1) = [LR(x_1)]^{1-2(u_1 \oplus u_2 \oplus u_3 \oplus u_4)} LR(x_2) \; LR(g_2) = [LR(x_3)]^{1-2(u_3 \oplus u_4)} LR(x_4)$$
$$LR(g_3) = [LR(x_5)]^{1-2(u_2 \oplus u_4)} LR(x_6) \; LR(g_4) = [LR(x_7)]^{1-2u_4} LR(x_8).$$

The likelihood ratios for h_1 and h_2 can be calculated as

$$LR(h_1) = [LR(g_1)]^{1-2(u_5 \oplus u_6)} LR(g_2) \; LR(h_2) = [LR(g_3)]^{1-2u_6} LR(g_4)$$

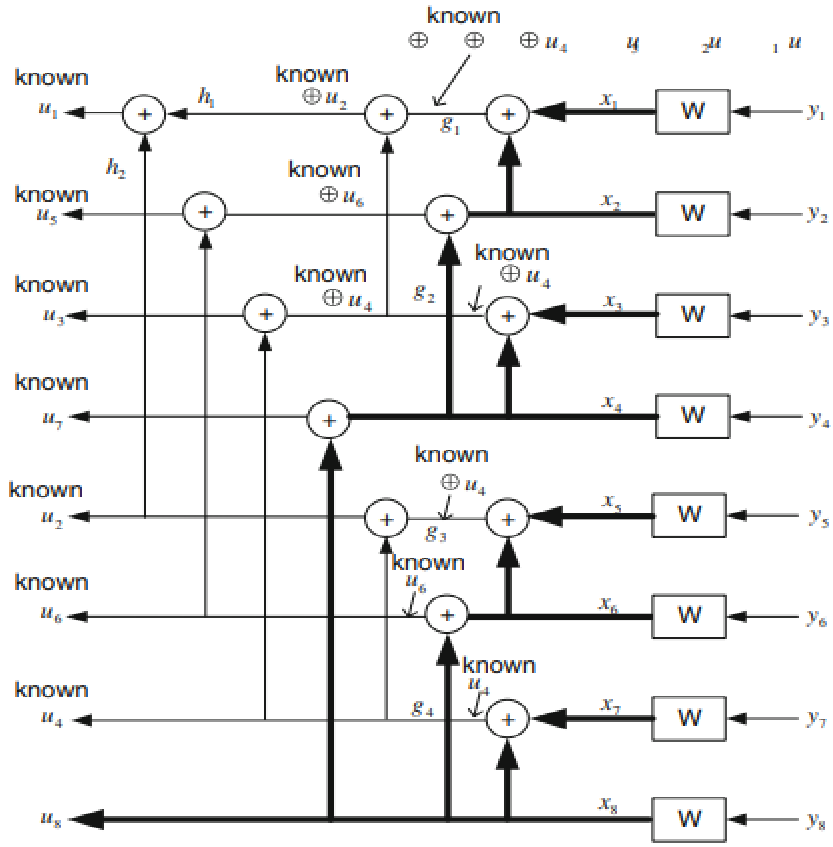

Figure. II.24 Decoding path of u8 when N= 8

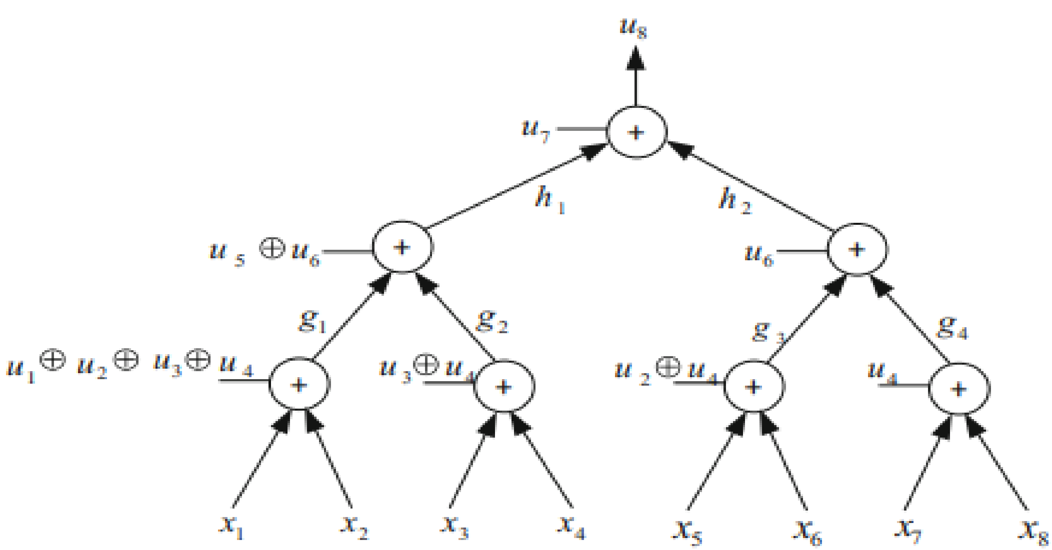

Figure. II.25 Decoding tree of u8 when N =8

The likelihood ratio for information bit u_8 is calculated as

$$LR(u_8) = [LR(h_1)]^{1-2u_7} LR(h_2) \qquad \text{(II.12)}$$

When the tree structure in Figure. II.25 is inspected in details, we see that the nodebits can be formed by considering columns of the matrices G_4, G_2, G_1. The matrices are given below for reminder

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} G_1 = [1]$$

It can be noticed that in the decoding operation of any data bit, some of the nodes at certain levels own some node bits in the tree structure. We will discuss in details the formation of node bits using generator matrices in the subsequent sections.

II.4.3 Decoding Algorithm for Polar Codes

The decoding approach detailed in the previous section can be expresses as a decoding algorithm which consists of two stages, and these stages are:

(a) Distribution of the previously decoded bits to the nodes.

(b) Decoding of the current bit.

The decoding tree consists of the repeated assembly of the tree-unit where the top node is called head node and left and right lower nodes are called left and right child nodes respectively

The distribution of the previously decoded bits to the nodes can be achieved using the method explained in Algorithm 2.1.

II.4.3.1 Algorithm Distribution of the decoded bits to the nodes:

Assume that i information bits are decoded, and we want to decode the $(i+1)$ th information bit. The codeword vector has a length of N, i.e. the total number of bits to be decoded is N. We need to distribute the i decoded bits to the nodes of the decoding tree. This can be achieved as follows:

(1) Let $u_1^i = [u_1 u_2 \dots u_i]$

(2) If i is an odd number, then assign u_i as the node bit of the head-node. And calculate the left child-node and right node-child bits as

(a) even bits $\rightarrow \bar{x}_e = (u_2 u_4 \dots u_{i-1})$

(b) odd bits $\rightarrow \bar{x}_o = (u_1 u_3 \dots u_{i-2})$

(c) left child - node bits $\rightarrow \bar{x}_l = \bar{x}_e \oplus \bar{x}_o$ where $\oplus$ denotes XOR operation

(d) right child - node bits $\rightarrow \bar{x}_r = \bar{x}_e$

(3) $i = \lfloor i/2 \rfloor$, form the tree units for the left and right child node bits, and for the left child node

$$u_1^i = \bar{x}_l \quad \text{(II.13)}$$

and for the right child node

$$u_1^i = \bar{x}_r \quad \text{(II.14)}$$

and continue with step (2) for the left and right child nodes separately if $i \neq 1$, terminate otherwise.

The bit distribution stage outline in the presented algorithm can also be achieved using the columns of the generator matrices at certain levels. If a generator matrix contains M columns, then it can be used at the level which includes M nodes.

II.4.3.2 Distribution of the decoded bits to the nodes using the generator matrices:

The decoded bits can be distributed to the nodes using the generator matrices. Assume that the code-word length is equal to N, and we decoded the first $M < N$ bits. The decoded bit vector can be written as

$$u_1^M = [u_1 u_2 \dots u_M] \quad \text{(II.15)}$$

For the decoding of $(M + 1)$ th bit, we first distribute the previously decoded bits to the tree nodes. For this purpose, we first write M as sum of powers of 2 , i.e., M is written as

$$M = \sum_i 2^i \quad \text{(II.16)}$$

where i indicates the indices of the levels where nodes are assigned bits. From (II.16) M can be written as

$$M = \underbrace{2^{i_1}}_{M_1} + \underbrace{2^{i_2}}_{M_2} + \cdots + \underbrace{2^{i_k}}_{M_k} \rightarrow M = M_1 + M_2 + \cdots + M_k.$$

In the second step, we divide the decoded bit vector u_1^M into sub-vectors $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_k$ containing $M_1, M_2, \ldots, M_k$ bits as

$$\underbrace{[u_1 u_2 \ldots \cdots\cdots \underbrace{\ldots\ldots}_{\bar{v}k-1} \underbrace{\ldots u_{M_k}}_{\bar{v}_k}]}_{\bar{v}_1}.$$

Once, we get the sub-vectors $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_k$, we can calculate the node bits at levels $i_1, i_2, \ldots, i_k$ as in (2.51)

$$\bar{b}_1 = \bar{v}_1 \times G_{M_1}\ \bar{b}_2 = \bar{v}_2 \times G_{M_2} \ldots \bar{b}_k = \bar{v}_k \times G_{M_k}$$

After obtaining the node bits for the levels $i_1, i_2, \ldots, i_k$, we can assign the bits to the corresponding nodes and start decoding of the current bit under concern.

II.5. Conclusion

In this chapter, we got to know more about the structure of the encoder in polar symbols and how it works, then we went to the study of the decompiler, where we noticed that this code is simple to install and work compared to other codes

Chapitre III

software Implementation (simulation)

III.1 Algorithm polar encoding in Matlab

```
N = 512;

K 300;

n = log2 (N);

Q1 - Q(Q<-N); %reliability sequence for N

%Frozen positions: Q1 (1:N-K)

%Message positions: Q1 (N-K+1:end)

Msg = randi ([0 11,1,K); %generate random K-bit message

u = zeros (1, N);

u (Q1 (N-K+1:end)) = msg; %assign message bits

m = 1; %number of bits combined

for d = n-1:-1:0

end

for I = 1:2*m:N

a = u(i:i+m-1); %first part

b = u(i+m:i+2*m-1); %second part

u = (i:i+2*m-1) - [mod (a+b, 2) b]; %combining

end

m=m*2;

end
```

III.2 Algorithm 1 Recursive Implementation of Polar Transform in Matlab

```
function x = polar_transform (u)
% Recurse down to length 1
if (length(u)==1)
x = u;
else
% Compute odd/even outputs of (I_{N/2} \otimes G_2) transform
u1u2 = mod (u(1:2: end) +u(2:2: end), 2);
u2 = u(2:2: end);
% RN maps odd/even indices (i.e., u1u2/u2) to first/second half X =
[polar_transform(u1u2) polar_transform(u2)];
end
```

III.3 Algorithm 2 Recursive Implementation of P1-Domain Polar Decoder in Matlab

```
function [u,x] = polar_decode(y,f)
% y = bit APP from channel in output order % f = input a priori probs in input order
% x = output hard decision in output order % u = input hard decisions in input order
% Recurse down to length 1
N = length(y);
if (N==1)
if (f==1/2)
% If info bit, make hard decision based on observation
x = round(y); u = x;
else
% Use frozen bit for output and hard decision for input (for monte carlo design)
x = f; u = round(y);
end
else
% Compute soft mapping back one stage
Ulest = cnop (y (1:2: end), y(2:2: end));
% R_NT maps ulest to top polar code
[uhat1,u1hardprev] = polar_decode(u1est, f(1: (N/2)));
% Using ulest and x1hard, we can estimate u2
u2est = vnop (cnop (u1hardprev, y(1:2: end)), y(2:2:end));
% R_NT maps ulest to top polar code [uhat1,u1hardprev] = polar_decode (u1est, f (1: (N/2)));
% Using ulest and x1hard, we can estimate u2 u2est = vnop (cnop (u1hardprev, y(1:2: end)), y(2:2: end));
% R_N^T maps u2est to bottom polar code
[uhat2,u2hardprev] = polar_decode (u2est, f ((N/2+1): end));
% Tunnel u decisions back up. Compute and interleave x1,x2 hard decisions
u = [uhat1 uhat2];
x = reshape([cnop (u1hardprev, u2hardprev); u2hardprev],1, []);
end
return
% Check-node operation in P1 domain
function z=cnop (w1,w2)
z = w1.*(1-w2) + w2.*(1-w1);
return
% Bit-node operation in P1 domain
function z=vnop (w1, w2)
z = w1.*w2./ (w1.*w2 + (1-w1).*(1-w2)); return
```

Algorithm 4Density Evolution Analysis of Polar Code over BEC

```
function E = polar_bec(n,e)
% Compute effective-channel erasure rates for polar code of length N=2^n on BEC(e)

E = e;

for i=1:n

% Interleave updates to keep in polar decoding order
E = reshape([1-(1-E).*(1-E); E. *E], 1, []);
end
```

III.5 Algorithm 4 Monte Carlo Estimate of Polar Code over the BSC

```
function [biterrd] =polar_bsc (n,p,M)
% Send M blocks for Monte Carlo estimate of length N=2^n polar code on BSC (p)
% Setup parameters
N = 2^n;
f = zeros(1,N);
biterrd = zeros(1,N);
% Monte Carlo evaluation of error probability
for i=1:M
% Transmit all-zero codeword through BSC (p)
y = zeros(1,N)+p;
y (rand(1, N) <p)=1-p;
% Decode received vector using all-zero frozen vector
[uhat,xhat] = polar_decode(y,f);
biterrd = biterrd+uhat;
end
biterrd = biterrd/M;
```

III.6Algorithm 5 Design a polar code based on effective-channel error rates

```
function f = polar_design (biterrd,d)
% Design a polar code based on effective-channel error rates

% Sort into increasing order and compute cumulative sum
[SE, order] = sort(biterrd);
CSE = cumsum(SE);
% Find best frozen bits
k = sum (double (CSE<d));
f = zeros(1,length(biterrd));
f(order (1:k)) = 1/2;
```

This program, which achieves polar code representation on MATLAB, has been taken from the following source:

A Brief Introduction to Polar Codes Notes for Introduction to Error-Correcting Codes Henry D. Pfister October 8th, 2017

General Conclusion

General Conclusion

At the end of this work, we saw that the polar code invented by Professor Erkan in 2008 is distinguished, as it was one of the first works concerned with the capacity of the transmission channel, as he tried through his work to increase the capacity of data that can be transmitted through the channel in a reliable, safe and conservative manner.

One of the advantages of Polar Code is its simplicity and uncomplicated structure, and this is what made them rely on it in 5G Standard in order to transfer the control bits

References

References

Chapter I

[1] C. E. SHANNON, «A mathematical theory of communication, » Bell SystemTechnical Journal, vol. 27, pp. 379-423 et 623-656, Juillet et Octobre 1948.

[2]‘YvonMori’,’Filtragenumérique’,’HermesScience’,2007.

[3]http://creativecommons.org/licenses/by-nc-sa/2.0/ , ‘Systèmes de communicationsnumériques’.

[4]GENEVIEVEBAUDOIN:’radiocommunicationsnumériques’,DUNOD,2002.

[5]‘YvonMori’,’théoriedel’informationetducodage,’HermesScience’,2007.

[6]‘JohnB.Anderson’,‘TorAulin,andCarl-ErikSundberg’.*DigitalPhaseModulation*.Plenum Press,New York,1986.[7]

[7]‘R. W Hamming ’,‘Error detecting and error correcting codes’ The Bell SystemTechnicalJournal ,April1950.

[8]’Gaël Mahé’, ‘Systèmes de communications numériques’, ‘UFR de MathématiquesetInformatiqueUniversitéParisDescartes’,’2010’.

[9]’A FISCHER’, ‘COURS DE TELECOMMUNICATION Commutations et systèmes detransmission’,’ IUTGTR -Université deParisXIII’.

[10] 'http://mediatools.iict.ch/document?url=Cours_de_Telecommunications Modulations/mod2/Mic.pdf&dpId=15', 'chapitre 8 modulationpar impulsion etcodage(MIC,PCM)’,

Chapter II

[1] A Brief Introduction to Polar Codes Notes for Introduction to Error-Correcting Codes Henry D. Pfister October 8th, 2017

[2] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” 2008. [Online]. Available: http://arxiv.org/abs/0807.3917. (To appear in IEEE Trans. Inform. Theory.)

[3] E. Arıkan and E. Telatar, “On the rate of channel polarization,” 2008.

References

[4] Polar Codes A Non-Trivial Approach to Channel Coding

[6] A. Andi, O. Gazi, Fast decoding of polar codes, IET Communications, submitted, (2017)

Chapter III

A Brief Introduction to Polar Codes Notes for Introduction to Error-Correcting Codes Henry D. Pfister October 8th, 2017

Abstract

The problem of information interference or damage during its transmission from the sender to the receiver is a problem with different stations, as the devices that deal with the information and the operations that occur to the information are different according to its position in the communication chain from the source to the transmission channel down to the receiver, during which the information is subject to encoding and decoding processes Encryption in order to protect it from damage and theft, and in this regard, the world Erdal Arıkan invented the Polar Code in 2008, as it provides us with two characteristics at the same time that works to protect the information and increase the capacity of the channel in order to send the largest possible number of bits At the same time

ملخص:

مشكلة تداخل أو تلف المعلومات أثناء نقلها من المرسل إلى المستقبل هي مشكلة مع محطات مختلفة ، حيث تختلف الأجهزة التي تتعامل مع المعلومات والعمليات التي تحدث للمعلومات حسب موقعها في سلسلة الاتصال عن المصدر إلى قناة الإرسال نزولاً إلى جهاز الاستقبال ، حيث تخضع المعلومات خلالها لعمليات التشفير وفك التشفير لحمايتها من التلف والسرقة ، وفي هذا الصدد ، اخترع العالم "إردال أريكان" Polar Code في عام 2008 ، حيث يوفر لنا خاصيتين في نفس الوقت تعمل على حماية المعلومات وزيادة سعة القناة من أجل إرسال أكبر عدد ممكن من البتات في نفس الوقت.