الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA 1

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Mention Électronique
Spécialité Télécommunications & Réseaux

présenté par

CAMARA Mahamoud

&

KABA Alassane

Introduction Aux Codes Polaires

Proposé par : Mr BERSALI Mahdi

Année Universitaire 2016-2017

Remerciements

Dédicace

Nous dédions ce projet de fin d'étude à :

- Nos parents

Nos mères respectives, qui ont œuvré pour notre réussite, de par leur amour, leur soutien, tous les sacrifices consentis et leurs précieux conseils, pour toute leur assistance et leur présence dans notre vie, recevez à travers ce travail, l'expression de nos sentiments et de notre éternelle gratitude.

Nos pères respectifs, qui peuvent être fier et trouver ici le résultat de longues années de sacrifices et de privations pour nous aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; merci pour les valeurs nobles, l'éducation et le soutien permanent venu de vous.

Nos frères et sœurs, qui n'ont cessé d'être pour nous des exemples de persévérances, de courage et de générosité.

- Nos enseignants

Nos professeurs et enseignants de l'Université de Blida 1 - Saad Dahleb qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

Remerciements

Nous remercions Dieu, le tout puissant, de nous avoir donné la force et le courage dans les moments difficiles d'éditer ce mémoire.

Ce projet de fin d'étude est le résultat d'un travail de recherche de près de sept mois. En préambule, nous voudrions adresser tous nos remerciements aux personnes avec lesquelles nous avons pu échanger et qui nous ont aidés pour la rédaction de ce mémoire.

En commençant par remercier tout d'abord Mr. Bersali Mahdi, enseignant à l'USDB, directeur de recherche de ce mémoire, pour son aide précieuse et pour le temps qu'il nous a consacré, pour ces conseils, son encouragement qui nous ont beaucoup aider dans la réalisation de ce projet de fin d'étude.

Ce travail n'aurait jamais vu le jour sans l'implication de nos professeurs, enseignants dans nos études en Télécommunications et Réseaux. Nous tenons à les remercier tous de leur sens de partage, de leur assiduité, de leur application, de leur dévouement.

Nos très sincères remerciements à l'encontre des membres du jury qui nous ont fait l'honneur d'accepter de juger notre travail.

Nous remercions sincèrement l'Etat Algérien de nous avoir donné la chance d'étudier chez lui.

Enfin, nous adressons nos plus sincères remerciements à nos familles respectives : parents, frères, sœurs et tous nos proches et amis qui nous ont accompagnés, aidés, soutenus et encouragés tout au long de la réalisation de ce projet de fin d'étude.

ملخص: التطبيقات في مجال الاتصالات الرقمية أصبحت تزداد تعقيدا وتنوعا. تشهد الحاجة إلى تصحيح الأخطاء من الرسائل المرسلة. لمعالجة هذه المشكلة، وتستخدم رموز تصحيح الخطأ. وسيركز العمل على دراسة ومحاكاة واحدة من هذه التحيزات التقنيات، واكتشاف وتصحيح أخطاء الإرسال باستخدام رموز القطبية الذي فك التشفير والترميز على التوالي إلغاء SC القطبي.

كلمات المفاتيح: رموز القطبية. الاستقطاب للقناة، SC، SCL، SCAN

Résumé : les applications dans le domaine des communications numériques deviennent de plus en plus complexes et diversifiées. En témoigne la nécessité de corriger les erreurs des messages transmis. Pour répondre à cette problématique, des codes correcteurs d'erreurs sont utilisés. Le travail a porté sur l'étude et la simulation d'une de ces techniques de préventions, de détections et de correction d'erreurs de transmission utilisant les codes polaires dont le codage polaire et le décodage par annulation successive SC.

Mots clés : Codes polaires ; polarisation du canal ; SC ; SCL ; SCAN.

Abstract : applications in the field of digital communications are becoming increasingly complex and diverse. It shows the need to correct the errors of the messages transmitted. To correct this problem, corrective codes are used. The work focused on the study and simulation of one of this techniques of prevention, detection and correction of transmission errors using polar codes whose polar coding and the decoding by successive cancellation SC.

Keywords : Polar codes; channel polarization ; SC ; SCL ; SCAN.

Listes des acronymes et abréviations

#

5G : Fifth Generation

A

ARQ : Automatic Repeat reQuest

AWGN : Additive White Gaussien Noise

B

B-AWGN : Binary-Additive White Gaussien Noise

BCH : Bose Chaudhury Hockenghem

BEC : Binary Erasure Channel

BER : Binary Error Rate

BLER : Block Error Rate

BMS : Binary-Input Memoryless Symmetric Channel

BP : Belief Propagation

BPSK : Binary Phase Shift Keying

BSC : Binary Symmetric Channel

C

CCITT : Consultative Committe International Telephone and Telegraph

CD : Compact Disc

CP : Codes Polaires

CRC : Cyclic Redundancy Codes

D

DVB-S2 : Digital Video Broadcasting- Satellite 2 generation

F

FEC : Forward Error Correction

FPGA : Field Programmable Gate Array

G

GSM : Global System for Mobile communication

H

HARQ : Hybrid ARQ

HDCL : High Level Data Link Control

I

IMT : International Mobile Telecommunications for the year 2000

IP : Internet Protocol

I/Q : In phase/ Quadrature

IS-95: Interim Standard -95

L

LDPC : Low Density Parity Check

LR : Likelihood Ratio

LLR : Logarithmic LR

LRC : Longitudinal Redundancy Check

LTE : Long Term Evolution

M

MIMO : Multiple Input Multiple Output

O

OFDM : Orthogonal Frequency Division Multiplexing

Q

QPSK : Quadrature Phase Shift Keying

R

RAN : Radio Acces Network

RM : Reed Muller

RS : Reed Solomon

S

SC : Successive Cancellation

SCL : SC with Lists

SNR : Signal to Noise Ratio

V

VRC : Vertical Redundancy Checking

W

WGN : White Gaussien Noise

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Dans le cadre des travaux de recherches au sein du laboratoire DIC (Detection Information et Communication) de l'université de Blida 1, il nous a été demandé de faire une étude détaillée sur une technique de codage canal, assez récente (2009), qui n'est autre que celle utilisant les codes polaires et de développer ses algorithmes de codage et décodage afin de pouvoir faire des simulations, notamment dans le cas systèmes de transmission basés sur les techniques HARQ (ARQ avec codes polaires), en mode de transmission MIMO OFDM et, de faire une étude comparative du point de vue BER avec les autres techniques de codage (convolutifs et turbo codes).

Quel que soit la qualité des supports de communications et les performances des techniques de transmissions utilisées, des perturbations vont se produire entrainant des erreurs sur les données transmises. Dans ces conditions, la suite binaire reçue ne sera pas identique à la suite transmise.

Donc pour remédier à ce problème, il a été mis en place des techniques de protection contre les erreurs de transmission.

Les réseaux numériques ne présentent guère de problèmes de ce point de vue. Mais les réseaux analogiques nettement moins sûrs et les réseaux sans fils qui sont en train de se développer montrent la nécessité de ces techniques.

Le principe général pour la détection des erreurs de transmission peut se résumer en trois (03) étapes [3]:

- l'émetteur veut transmettre un message (suite binaire quelconque) à un récepteur.

- l'émetteur transforme le message initial à l'aide d'un procédé de calcul spécifique qui génère une certaine redondance des informations au sein du message codé.
- le récepteur vérifie à l'aide du même procédé de calcul que le message reçu est bien le message envoyé à ces redondances.

On peut citer comme exemple la technique de détection par répétition où le message codé est un double exemplaire du message initial, le récepteur sait qu'il a eu erreur si les deux exemplaires ne sont pas identiques. Notons bien que certaines erreurs sont indétectables telle une même erreur sur les deux exemplaires simultanément.

Pour ce qui est la correction des erreurs de transmission, après détection d'une erreur, la redondance est suffisante pour permettre de retrouver le message initial. On peut citer comme exemple de technique de correction par répétition.Le message codé est un triple exemplaire du message initial, le récepteur suppose que le message initial corresponde aux deux exemplaires qui sont identiques. Néanmoins certaines erreurs détectées ne sont pas corrigeables telle une erreur différente sur au moins deux exemplaires.Certaines erreurs sont détectées et mal corrigée telle une même erreur sur deux exemplaires simultanés.

Lors de la transmission d'information dans un réseau de télécommunication, les données peuvent être altérées. Les codes correcteurs d'erreurs (FEC) ont été développé pour pouvoir palier à ces inconvénients. Leur principe est le rajout aux messages envoyés de bits de contrôles pour détecter et corriger les erreurs de transmission sans demander une retransmission du message.

En modélisant mathématiquement l'information transmise sur un canal de transmission, une étape importante fût franchie par Claude Shannon (Shannon, 1948)[6]. Grâce à l'approche qu'il propose, il devient possible de répondre aux deux questions fondamentales posées par la théorie de l'information :

- Quelles sont les ressources nécessaires à la transmission de l'information ?

- Quelle est la quantité d'information que nous pouvons transmettre de façon fiable ?

En répondant à ces questions de base de la théorie de l'information, Shannon n'a cependant pas trouvé de code permettant d'atteindre la limite théorique qu'il venait de poser[6].

Depuis l'établissement de cette approche mathématique de l'information, c'est ce vide que tente de combler la théorie du codage en proposant des constructions de codes efficaces.

Historiquement, un des premiers codes correcteurs d'erreurs utilisé fut un code à répétition. Ce dernier consiste à transmettre plusieurs fois de suite la même valeur binaire. La décision est alors prise par un vote à majorité.

D'autres codes correcteurs d'erreurs, plus complexes et plus performants BCH, Reed Solomon, LDPC, Turbocodes, …etc sont apparus ces dernières décennies [12].

Le choix du type de code dépend du cahier des charges de l'application cible telque le besoin en autonomie, la forte intégration, la rapidité de traitement, la capacité de correction, …etc.

Les codes polairessont des codes en blocs linéaires et ont été proposés par Erdal Arikan en 2008 dans son document phare [1].C'est une famille de codes correcteurs d'erreurs ayant construction explicite et une complexité acceptable de codage et décodageen utilisant un algorithme particulier appelé ou Annulation Successive SC.

L'originalité des travaux d'Arikan est la technique de polarisation du canal qui consiste à transformer le canal initial en N canaux binaires différents, tel que quand N tend vers l'infini. Une partie de ces canaux binaires auront une capacité qui tend vers "1" et seront considérés comme les "bons canaux"alors que les autres canaux binaires auront une capacité qui tend vers "0" et seront considérés les "mauvais canaux"[17].

Les codes polaires fait l'objet d'une recherche active ces derniers temps, principalement en raison du fait qu'ils sont les premiers codes atteignant la capacité des canaux BMS [9].Il a été prouvé mathématiquement que ces derniers étaient capables de corriger toutes les erreurs de transmission sous certaines conditions [2].Cependant ils nécessitent l'utilisation d'un message de taille infinie[1].Pour des implémentations FPGA, taille du mot de code s'élève jusqu'à$N = 2^{21}$[19].

Commence alors une analyse des codes polaires afin de déterminer leur intérêt (complexité, performances) et leur praticité (implémentation) à taille finie.

Pour ce qui de leurs applications, Huawei a annoncé en octobre 2016 qu'elle avait réalisé 27Gbps dans des tests d'essais sur champs 5G en utilisant les codes polaires pour le codage des canaux [11].

Lors de la rencontre 87 RAN1 organisée le 17 novembre 2016 par le 3GPP, organisme international de développement des standards pour les télécommunications mobiles, les codes polairesont été choisi comme schéma de codage du canal de contrôle pour l'application de la 5G dans le scénario eMBB[11].

Après avoir introduit principe général de la détection et la correction des erreurs de transmission, les codes polaires ainsi que leurs applications nous présentons dans la figure qui suit le synoptique de notre chaîne de transmission:

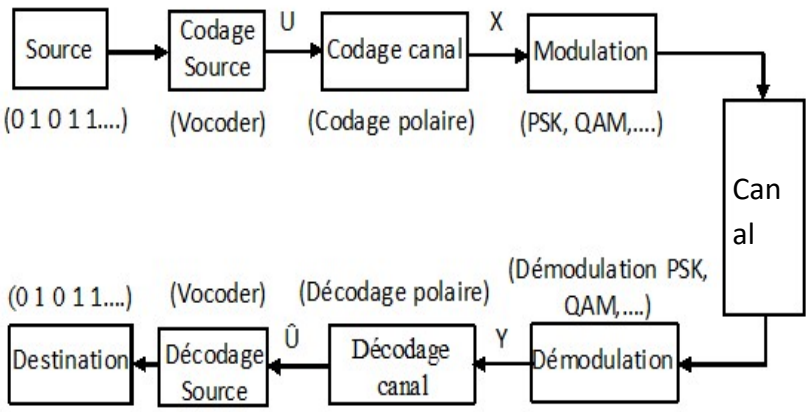


Figure 0.1. Schéma synoptique d'une chaine de transmission

- Source : K bits d'informations (informations utiles)
- Codage/décodage source : a pour but principal de réduire la quantité d'information à transmettre en éliminant de la redondance
- Codage/décodage canal : son but n'est pas de réduire la quantité d'information mais de rajouter de la redondance au message à transmettre afin de pouvoir détecter et corriger le plus d'erreurs possibles
- Modulation/démodulation : avant d'être transmit le message subit une modulation et avant d'êtrereçu il subit une démodulation
- Canal de transmission : le message est transmis via le canal de transmission(air, fibre optique, câble…..)
- Destination : K bits d'informations estimées reçues.

Afin de mener à bien notre travail, nous avons organisé notre mémoire de la façon suivante :

Dans le premier chapitre, nous rappellerons les différentes techniques de détection et de correction des erreurs de transmissions appelées FEC. Nous parlerons ainsi du codage, du décodage, ainsi que des performances (la capacité de correction) de ces techniques.

Dans le deuxième chapitre, nous entamerons le cœur de cette thèse qui est l'étude des codes polaire, où nous essayerons de détailler au mieux avec des exemples à l'appui les algorithmes de codage et décodage ainsi que leur capacité à corriger les erreurs de transmissions dans le cas de canaux BMS.

Dans le dernier troisième chapitre, nous présenterons une synthèse des différents résultats sélectionnés à partir des bases de données scientifiques en notre possession (articles, livres et tutoriaux) et, éventuellement nos propres simulations des codes polairesdans le cas du canal B-AWGN, voire Rayleigh - AWGN.

Enfin nous terminerons notre travail par une conclusion.

Chapitre 1 Les codes correcteurs d'erreurs

1.1 Introduction

Lors de la transmission d'informations dans un réseau, les données peuvent être altérées à cause des imperfections des canaux de transmission. Les codes correcteurs d'erreurs ont été développés pour pouvoir pallier à ces inconvénients : on rajoute aux messages envoyés des bits de contrôles pour détecter et corriger de telles erreurs.

En télécommunication, précisément enthéorie de l'information, les codes correcteurs d'erreurs ou codage canal sont des techniques utilisées pour la détection et correction d'erreurs en transmission des données, à travers des canaux de communications bruités ou non fiable. L'idée est que l'émetteur ajoute des informations (bits) redondantes au message utile et le récepteur exploite cette redondance pour décider quel message a été envoyé.

De nombreux codes correcteurs ont été inventés durant ces cinq dernières décennies.

Le mathématicien américain Richard Hamminga ouvert la voie dans ce domaine dans les années 40 et c'est en 1950 qu'il invente le premier code correcteur d'erreur appelé code de Hamming.[7]

On peut citer aussi les travaux de Peter Elias[5, 8] pour les codes produits et le codes convolutifs en 1954, de Irving Stoy Reed et David Eugène Müller[12] pour les codes RM en 1954, de Raj Chandra Bose, DwijendraKumarRay-Chaudhuri et Alexis Hocquenghem [12] pour les codes BCH en 1959, de Marcel Jules ÉdouardGolay en 1954 pour les codes de Golay ou I.S Reed et Gustave Solomon [12] pour les RS en 1960.

En 1993 Claude Berrou et Alain Glavieux[5, 8, 16] ont révolutionné le domaine des codes correcteurs d'erreurs en communication numérique en inventant le turbo codes qui se rapprochent des limites théoriques de Shannon. En 1995 David John Cameron Mackaya redécouvert les codes LDPC qui ont été inventé par RobertGray Gallager [12] en 1963 mais oublié puisqu'ily avait un manque de moyen et de calculateur rapide pour les tester à l'époque, alors qu'à actuellement ils sont considérés comme l'un des codes les plus performants. Les turbo-codes produits (Ramesh Pyndiah et al. , 1994), les turbo-codes en blocs (Erwan Piriou, 2007).

Les turbo-codes et les codes LDPC sont des codes pseudo aléatoires.

Les codes correcteurs sont classés en trois grandes familles qui sont :

- Les codes en bloc (linéaires, cycliques ou non) : le codage/décodage d'unbloc dépend uniquement des informations de ce bloc.
- Les codes en treillis (convolutifs, récursifs ou non) : le codage/décodage d'un bloc dépend des informations d'autres blocs (généralement de blocs précédemment transmis).
- Les codes approchant la limite de Shannon : les turbo-codes et les codes LDPC.

1.2 Les codes en blocs linéaire[5, 13]

1.2.1 Définitions

Les codes en blocs linéaires sont des codes linéaires notés $\mathrm{C}(\mathrm{N}, \mathrm{K})$.

La linéarité provient du fait que la somme de deux mots de codes est elle-même un mot de code. L'idée consiste à coder non pas les bits individuellement, mais à les coder par blocs d'une certaine longueur fixée.

Le codage en bloc consiste à associer à un bloc de donnéed de K symboles en provenance de la source d'information, un bloc$\mathrm{C}$, appelé mot de code de N symboles avec$\mathrm{N} \geq \mathrm{K}$. La quantité $(\mathrm{N} - \mathrm{K})$ est la redondance introduite par le code [5, 13]. L'ensemble des 2^{K} mots construits par le codeur est appelé le code et noté$\mathrm{C}(\mathrm{N}, \mathrm{K})$ avec :

- K le nombre de bits d'information
- N le nombre de bits de mot de code
- $N - K$ le nombre de bits de redondance.

Le taux de codage ou rendement du code se calcule par le rapport $R = \frac{K}{N}$.

La protection est dite forte si R tend vers0, et faible si R tend vers 1.

Les symboles messages de l'information d et du mot de code cprennent leurs valeurs dans un champ fini F_qà q éléments qu'on appelle champ de Galois.

Dans le reste du mémoire, l'ensemble des messages et des mots de codes prennent leur valeur dans le corps binaire $\{0,1\}$ notéF_2.

Un mot de code est obtenu par additions et multiplications. Dans le corpsF_2, ces opérations sont respectivement équivalentes à un OU exclusif (XOR) logique et à un ET logique. Les règles des opérations dans F_2 sont les suivantes :

a	b	$a \oplus b$	a.b
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tableau 1.1.Opérations élémentaires dans F_2

Le pouvoir de correction d'un code dépend de la distance de Hamming minimale de ce code (voir §1.2.3).

En pratique, on s'intéresse particulièrement aux codes linéaires. Ils sont tels que si $c_1$et c_2 sont deux codes respectivement des deux messages $r_1$etr_2, alors la suite K bits $r_1 \oplus r_2$est associé le mot de code$c_1 \oplus c_2$, avec l'opération d'addition bit à bit modulo 2.

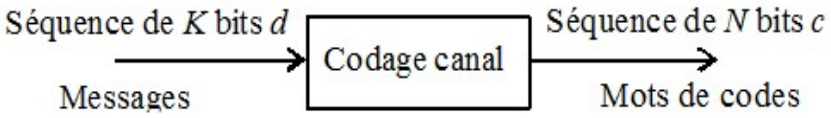


Figure 1.1. Schéma d'un codeur canal

La somme de deux (2) mots de codes forme un mot de code aussi c'est-à-dire que si $c_1 \in c$ et $c_2 \in c$ alors $c_1 \oplus c_2 \in c$.

Le mot nul est un mot de code de n'importe quel code en blocs.

Il existe plusieurs manière d'ajouter de la redondance mais toujours en s'assurant que le décodage peut être effectué sans problème.

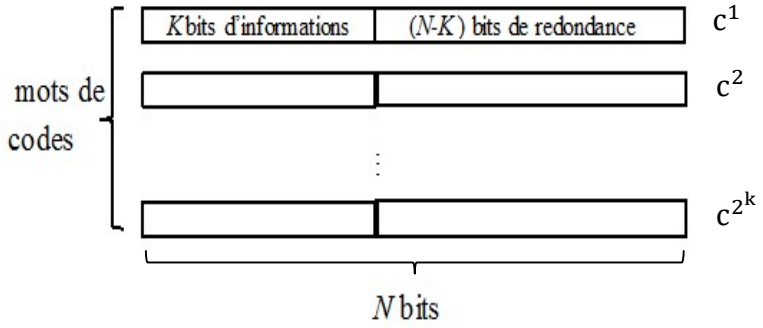


Figure 1.2. Répartition des éléments d'un mot de code

$(d_1, d_2, \ldots, d_k) \rightarrow (c_1, c_2, \ldots, c_n)$

1.2.2 Différentes représentations d'un code à bloc linéaire

Il est possible de représenter un code en bloc linéaire sous trois (2) principales formes : forme matricielle (forme systématique et non systématique) et forme d'une matrice de parité.

1- Représentation matricielle

L'opération du codage dans les codes en blocs linéaire peut être effectuée par une matrice génératrice du code G de dimension$(K \times N)$, K étant le nombre de lignes et N le nombre de colonnes telle que le mot de code c du message d est calculé par :

$$c = d \times G \quad (1.1)$$

$d = (d_0, d_1, ..., d_{k-1})$ vecteur ligne $(1 \times K)$est le vecteur de l'information.

$c = (c_0, c_1, ..., c_{n-1})$ vecteur ligne $(1 \times N)$ est le vecteur mot de code.

La matrice génératrice d'un code à bloc linéaire est donnée par :

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{K-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,(N-1)} \\ g_{10} & g_{11} & \cdots & g_{1,(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(K-1),0} & g_{(K-1),1} & \cdots & g_{(K-1),(N-1)} \end{bmatrix}_{K \times N} \quad (1.2)$$

L'ensemble des mots de codec, est un ensemble formé des combinaisons linéaires des K lignes de$G$. En fait les K lignes linéairement indépendantes de G définissent complétement le code. Donc le vecteur mot de code c se calcul de la manière suivante :

$$c = \sum_{j=0}^{k-1} d_i g_{ij} \; j = 0, 1, \ldots, N-1 \quad (1.3)$$

Lesmots de codes d'un code à bloc linéaire sont donné par :

$$c = [d_0, d_1, ..., d_{K-1}] \times \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,(N-1)} \\ g_{10} & g_{11} & \cdots & g_{1,(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(K-1),0} & g_{(K-1),1} & \cdots & g_{(K-1),(N-1)} \end{bmatrix}$$

$$= d_0 \square g_0 \oplus d_1 \square g_1 \oplus ... \oplus d_{K-1} \square g_{K-1}$$

Exemple 1.2.1

Pour un code de parité de taille $N = 5$ nous avons :

- le message de K = 4 bits est tel que $d = [d_0, d_1, d_2, d_3]$
- le mot de code de N = K + 1= 5 bits est tel que $c = [c_0, c_1, c_2, c_3, c_4]$
- la matrice de l'application linéaire est :

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

En fin :

$$c = d \times G = [d_0, d_1, d_2, d_3] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_0 + d_1 + d_2 + d_3 \end{bmatrix}^T$$

Figure 1.3. Codage de parité par produit matriciel

a Forme systématique

Un codage est dit systématique lorsque l'on retrouve dans le mot codé les K symboles d'information dans K positions déterminées.

Si un codage peut être mis sous forme systématique, la matrice génératrice G d'un code en bloc linéaire peut être mise sous forme systématique G_{Sys}par des opérations élémentaires sur les lignes et/ou des permutations sur les colonnes.

La matrice G_{Sys}est composée de deux sous matrices :

- La matrice identité $K \times K$(notééI_K) et
- La matrice $K \times (N-K)$de parité (notéeP) telles que :

$$G_{Sys} = \left[I_K \middle| P \right]$$

$$G_{Sys} = \left[I_K \middle| P_{k\times(N-K)} \right] = \left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & p_{00} & p_{01} & \cdots & p_{0,N-K-1} \\ 0 & 1 & \dots & 0 & p_{10} & p_{11} & \cdots & p_{1,N-K-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{K-1,0} & p_{K-1,1} & \cdots & p_{K-1,N-K-1} \end{array}\right] \quad (1.4)$$

P : est une matrice $K \times (N-K)$

Dans ce cas les bits de contrôle de parité peuvent être obtenus en fonction des bits de message en utilisant l'expression suivante :

$$c_j = d_j \qquad 0 \le j < K-1$$

$$c_{(K+i)} = d_0 \cdot p_{0,i} \oplus d_1 \cdot p_{1,i} \oplus \dots \oplus d_{K-1} \cdot p_{K-1,i} \quad i = 0,1,\dots,N-K-1$$

$$c = [d \;\; dP] \quad (1.5)$$

b Forme non systématique

Un code non systématique serait tel que : $G = [M \quad P]$ où la matrice M est une matrice qui mélange les bits du message pour le crypter. Nous pouvons bien sûr nous ramener à un code systématique par combinaison des lignes entre elles pour faire intervenir la matrice I_K. Tout code de bloc peut ainsi se ramener à l'étude d'un code systématique équivalent.

Exemple 1.2.2

1°) Soient le message $[d_0, d_1, d_2]$ et la matrice génératrice $G = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$

Déterminons les mots de codes C(6,3) :

$$c = d \times G = [d_0, d_1, d_2] \times \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$c_0 = d_0$

$c_1 = d_1$

$c_2 = d_2$

$c_4 = d_1 \oplus d_2$

$c_4 = d_1 \oplus d_2$

$c_5 = d_0 \oplus d_2$

Message			Mot de code					
d_0	d_1	d_2	c_0	c_1	c_2	c_3	c_4	c_5
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	1
0	1	0	0	1	0	1	1	0
0	1	1	0	1	1	1	0	1
1	0	0	1	0	0	1	0	1
1	0	1	1	0	1	1	1	0
1	1	0	1	1	0	0	1	1
1	1	1	1	1	1	0	0	0

Tableau 1.2.Les messages possibles et les mots de code correspondants

2°) Soient le message et le mot de code correspondant suivants :

Message			Mot de code					
d_0	d_1	d_2	c_0	c_1	c_2	c_3	c_4	c_5
0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1
0	1	0	0	1	0	1	0	1
1	1	0	1	1	0	1	1	0
0	0	1	0	0	1	1	1	0
1	0	1	1	0	1	1	0	1
0	1	1	0	1	1	0	1	1
1	1	1	1	1	1	0	0	0

Trouvons la matrice génératrice correspondant :

$$c = d \times G,\ G = \left[\begin{array}{ccc|ccc} g_{11} & g_{12} & g_{13} & g_{14} & g_{15} & g_{16} \\ g_{21} & g_{22} & g_{23} & g_{24} & g_{25} & g_{26} \\ \underbrace{g_{31} \quad g_{32} \quad g_{33}}_{I_K} & & & g_{34} & g_{35} & g_{36} \end{array}\right]$$

$$[1\ 0\ 0] \times G = c$$

$g_{14} \oplus 0 \oplus 0 = 0$ g_{14}=0 ,

$g_{15} \oplus 0 \oplus 0 = 1$ $g_{15} = 1$,

$g_{16} \oplus 0 \oplus 0 = 1$ $g_{16} = 1$,

$[0\ 1\ 0] \times G = c$

$0 \oplus g_{24} \oplus 0 = 4$ $g_{24} = 1$,

$0 \oplus g_{25} \oplus 0 = 0$ $g_{25} = 0$,

$0 \oplus g_{26} \oplus 0 = 1$ $g_{26} = 1$,

D'où $G = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \underbrace{0 \quad 0 \quad 1}_{I_K} & & & 1 & 0 & 1 \end{array}\right]$

$[0\ 0\ 1] \times G = c$

$0 \oplus 0 \oplus g_{34} = 1$ $g_{34} = 1$,

$0 \oplus 0 \oplus g_{35} = 0$ $g_{35} = 0$,

$0 \oplus 0 \oplus g_{36} = 1$ $g_{36} = 1$.

c Représentation par une matrice de parité

Pour un code $C(N, K)$il est possible de déterminer une matrice génératrice de contrôle de parité notée H et telle que :

$$c \cdot H^T = 0 \qquad (1.6)$$

On peut conclure aussi que pour un code $C(N,K)$de matrice génératrice$G$, la matrice contrôle de parité H vérifie l'équation suivante :

$$H.G^T = 0 \Leftrightarrow G.H^T = 0 \tag{1.7}$$

Si d est un bloc, de mot code$c = d \times G$, alors on doit avoir :

$$H.\ c^T = H.\ (d.\ G)^T = (H.G^T).\ d^T = 0 \tag{1.8}$$

Le bloc creçu est un mot code (donc jugé non erroné) si et seulement si $c.H^T = 0$. Pour les codes de formes systématique avec $G_{Sys} = \left[I_K \middle| P_{k\times(N-K)} \right]$, la matrice de contrôle de parité Hest donnée par :

$$H = \left[P^T_{(N-K)\times K} \middle| I_{(N-K)} \right] \tag{1.9}$$

Exemple 1.2.3

$$G = \left[I_3 \middle| P \right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

$$\Leftrightarrow$$

$$H = \left[P^T \middle| I_3 \right] = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Figure 1.4. Matrice de contrôle de parité

1.2.3 Distance minimale d'un code

- La distance de Hamming entre deux mots de codes C_I et $C_J$correspond au nombre d'éléments différents entre eux. Elle est notée $d_H(C_I, C_J)$. On l'obtient par le poids de Hamming de l'opération OU exclusif binaire (XOR) des deux mots.

- Le poids de Hamming d'un mot $c_I$noté w_H, est sa distance de Hamming au mot nul :

$$w_H(c_I) = d_{min}(c_I, 0). \tag{1.10}$$

Le poids de Hamming d'un mot correspond également au nombre de bits à1qu'il contient.

- La distance minimale correspond à la distance de Hamming entre les deux mots de codes les plus proches. Elle est déterminée par l'équation suivante :

$$d_{min} = \min d_H(C_I, C_J) \tag{1.11}$$

C_I et C_J étant deux mots de code différents.

En tenant compte du fait que la distance entre deux(2) mots de code est égale au poids de leur somme, la distance minimale d'un code de bloc est également égale au poids minimal de ses mots de codes non nuls.

$$d_{min} = \min w_H(c)\ c \neq 0 \tag{1.12}$$

Lorsque le nombre de mots de code est très élevé, la recherche de la distance minimale peut être laborieuse. Une première solution pour contourner cette difficulté consiste à déterminer la distance minimale à partir de la matrice de contrôle de parité H.

Nous avons vu que d_{min}est égale au poids de Hamming minimal des mots de code non nuls. Considérons un mot de code de poidsd_{min}. La propriété d'orthogonalité

$c.H^T = 0$implique que la somme des d_{min}colonnes de la matrice de contrôle de parité est nulle. Ainsi d_{min} correspond au nombre minimum de colonnes linéairement dépendant de la matrice de contrôle de parité.

Exemples 1.2.4

Soit deux (2) mots de codes $c_1 = (01100110)$ et $c_2 = (01011010)$.

- La distance de Hamming entre c_1 et c_2 est :

$$d_H(c_{1,}c_2) = w_H(c_1 \oplus c_2)$$

$$w_H(01100110 \oplus 01011010) = w_H(0011110) = 4.$$

- Les poids de Hamming de $c_1$et $c_2$sont : $w_H(c_1) = 4$et $w_H(c_2) = 4$.

- Distance minimale :

$$H = \begin{bmatrix} \overset{(1)}{1} & \overset{(2)}{0} & \overset{(3)}{1} & \overset{(4)}{1} & \overset{(5)}{1} & \overset{(6)}{0} & \overset{(7)}{0} \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

La distance minimale peut se calculer à partir de la matrice de contrôle de parité H**.** On remarque qu'on ne peut pas trouver moins detrois (3)colonnes à sommer pour avoir la colonne nulle. On peut prendre : col (1) $\oplus$ col(3) $\oplus$ col(7) = (0, 0, 0) ^{T}. Par conséquent la distance minimale $d_{min} = 3$.

Propriétés

- $d_H(c_{1,}c_2) \geq 0$,
- $d_H(c_I,c_J) = d_H(c_J,c_I)$.

Une autre façon de déterminer la distance minimale est d'utiliser les limites supérieures de la distance minimale. Une première borne peut être exprimée comme une fonction des paramètres du code K et$N$.

Pour un code de bloc de linéaire dont la matrice génératrice est écrite sous la forme systématique : $G = [I_K \setminus P]$ les $N - K$ colonnes de la matrices I_{N-K} de la matrice de contrôle de parité $H = [P^T \setminus I_{N-K}]$ étant linéairement indépendant, une colonne P^T peut être étant que tout au plus une combinaison de ces colonnes$N - K$. La distance minimum à une borne supérieure qu'on appelle borne de Singleton délimitée par :

$$d_{min} \leq N - K + 1 \tag{1.13}$$

Une autre borne supérieur de la distance minimale, appelé borne de Plotkinlimite est donné par :

$$d_{min} \leq \frac{N.2^k - 1}{2^k - 1} \tag{1.14}$$

1.2.4 Code Hamming [7]

Les codes de Hamming sont une généralisation du bit de parité (un bit de contrôle pour fixer la parité du nombre de 1 dans les mots). On insère dans le code des bits de contrôle de parité. Les bits de contrôle vérifient la parité des bits dont ils apparaissent dans leur décomposition en base 2.

De plus la distance minimale de ce code est3 car il n'existe aucun mot de code de poids 1 ou 2 puisque cela signifierait qu'il y a une colonne nulle ou deux colonnes égales dans$H$.

Un code de Hamming est donné sous la forme $C(2^m - 1, 2^m - 1 - m, 3)$ $m = N - K$.

Exemple 1.2.5

Prenons les codes de Hamming $C\ (7,4)$: 4 bits de donnée, 3 bits de contrôle. Envoyons le message abcd , on génère le premier bit de contrôle x qui est la somme des bits $0,1,3$ qui occuperont respectivement la place $3(2+1)$, $5(4+1)$, $7(2+4+1)$ dans le message final donc $\mathrm{x} = \mathrm{a} + \mathrm{b} + \mathrm{d}$, de mêmey $= \mathrm{a}(2+1) + \mathrm{c}(4+2) + \mathrm{d}(4+2+1)$ et $\mathrm{z} = \mathrm{b}(4+1) + \mathrm{c}(4+2) + \mathrm{d}(4+2+1)$ et le message envoyé est $\mathrm{xyazbcd}$.

Par exemple 1010 se codera : $x = 1+0+0 = 1$,$\mathrm{y} = 1+1+0 = 0$,$\mathrm{z} = 0+1+0 = 1$, le message envoyé est 1011010.

Le codage et le décodage correspondent à un simple produit de matrice. Différente implémentation de ces codes existent pour les rendre plus efficace dans certains cas (rafale d'erreurs,…) ou certaines circonstance.

1.2.5 Code Hadamard [7]

Les codes de Hamming sont très populaire de par leur simplicité cependant, ils exigent un grand nombre de bit de contrôle pour corriger des erreurs multiples. Les codes de Hadamard sont constitués par itération à partir de matrice telle que :

$$\mathrm{M}_{2\mathrm{N}} = \begin{bmatrix} \mathrm{M}_{\mathrm{N}} & \mathrm{M}_{\mathrm{N}} \\ \mathrm{M}_{\mathrm{N}} & \overline{\mathrm{M}_{\mathrm{N}}} \end{bmatrix}$$ Où $\overline{\mathrm{M_N}}$ désigne la matrice complémentaire $0 \rightarrow 1$et $1 \rightarrow 0$.

La matrice initiale est : $\mathrm{M}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \rightarrow \mathrm{M}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ (1.15)

Les matrices $\mathrm{M_N}$ et $\overline{\mathrm{M_N}}$ contiennent comme lignes ou colonnes les $2\mathrm{N}$ mots du code. La structure de la matrice permet donc de détecter des erreurs de transmission. Sans démonstration, on montre que $\mathrm{d}_{\min} = \frac{1}{2\mathrm{N}}$.

1.2.6 La détection des erreurs

1- Les codes cycliques (CRC)

Ces codes appelés CRC sont des codes en blocs particuliers. Ils sont très utilisés en pratique à cause de leur efficacité et de leur facilité de réalisation matérielle.

Principe

Soit G un polynôme générateur de degrés $r$connu par l'émetteur et le récepteur. Ce polynôme générateur permet de générer la séquence à ajouter au message (la redondance) de la manière suivante : on multiplie la trame à envoyer parX^r, on obtient une nouvelle trame qu'on appelle M(X)= T(X). X^r . Puis on procède à la divisioneuclidienne (XOR) $M(X)/G(X)$. Le reste de cette division R(X) est ajouté à la trame à transmettre.

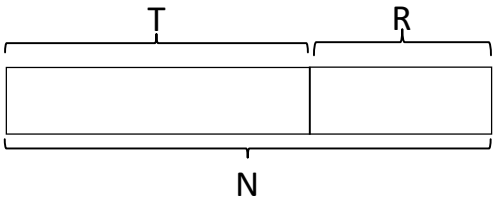


Figure 1.5.Trame à envoyer protégée par les bits CRC

Exemple 1.2.6

Soient $G = X^3 + X^2 + 1$ le polynôme générateur de degrés $r = 3$ et $T(X) = X^7 + X^4 + X^3 + X$ la trame à envoyer avec $T = 10011010$

$M(X) = T(X).X^3 = X^{10} + X^7 + X^6 + X^4$

$M(X) / G(X) = M(X) \oplus G(X)$ avec $M(X) = 10011010000$ et $G(X) = 1101$

$10011010000 \oplus 1101 = ... = 0101$

Le reste de la division est égale à 0101 d'où $R = 101$carr $= 3$.

La trame à transmettre sera$N = 10011010101$.

A la réception, le récepteur va prendre la trame reçue N et la diviser par G.

Si $R = \begin{cases} 0 & \text{pas d'erreurs} \\ 1 & \text{il y a au moins une erreurs} \end{cases}$

$N \oplus G = 10011010101 \oplus 1101 = \cdots .. = 0000$

$R = 0000$ donc il n'y a pas d'erreurs.

Remarque : le choix du polynôme générateur s'effectue en fonction du type d'erreur à détecter. Il est en particulier important de trouver des polynômes capables de détecter des erreurs groupées. Les trois polynômes générateurs suivants sont les plus utilisés :

$CRC - 12\, X^{12} + X^{11} + X^3 + X^2 + X + 1$,

$CRC - 16\, X^{16} + X^{15} + X^2 + 1$,

$CRC - CCITT\, X^{16} + X^{12} + X^5 + 1$,

$CRC - 12$ et $CRC - 16$ sont utilisés par des protocoles en mode de base avec des caractères de longueurs de 6 bits et 8 bits respectivement,

$CRC - CCITT$ est utilisé par le protocole de liaison HDCL. A titre d'efficacité de détection des polynômes générateurs $CRC - 16$ et $CRC - CCITT(r = 16)$ on a :

- 100% des erreurs simples ;
- 100% des erreurs doubles ;
- 100% des erreurs sur un nombre impair de bits.
- 100% des paquets d'erreurs de longueur ≤ 16 ;
- 99.998% des paquets d'erreurs de longueur $= 17$;
- 99.997% des paquets d'erreurs de longueur ≥ 18.

2- Notion du syndrome d'erreurs [5]

Considérons que le mot de code transmis par le codeur est c et soit r le mot de N symboles reçus et présentés à l'entrée du décodeur.

Le mot r peut s'écrire sous la forme suivante :

$r = c + e$où e est un mot dont les symboles non nuls représentent les erreurs. Un symbole non nul de e indique la présence d'une erreur dans la position correspondante de c.

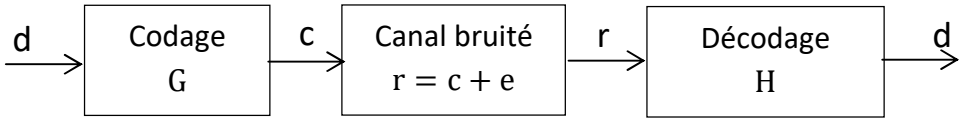


Figure 1.6.Opération de codage et de décodage sur un canal bruité

Les erreurs sont détectées en utilisant la propriété d'orthogonalité de la matrice de contrôle de parité H avec les mots de codes et ceci en calculant le syndrome d'erreur S.

$$S = r.H^T = (c + e)H^T = e.H^T \qquad (1.16)$$

En principe

- Si e=0 → S=0 car r=c → aucune erreur détectée,
- Si e≠0 → S≠0 → il y a erreur.

Le syndrome S est nul si, et seulement si, r est un mot de code. Un syndrome non nul implique la présence d'erreurs. Toutefois, il convient de noter qu'un syndrome nul ne signifie pas nécessairement l'absence d'erreurs puisque r peut appartenir à l'ensemble des mots de code même si elle est différente dec. Pour se faire, il suffit que le mot e soit comme un mot de code (e=c') et donc S=e. H^T=0.

En effet, pour un code de bloc linéaire, la somme de deux mots de codes est un autre mot de code.

Il peut y avoir ($2^{K}-1$) indétectable configurations d'erreurs non nuls. On peut dire aussi qu'un code en bloc linéairec(N,K)est capable de détecter ($2^{N}-2^{K}$) configurations d'erreurs et de corriger 2^{N-K} configurations d'erreurs.

Exemple 1.2.7

Soient $H=\left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$ la matrice de parité et $r=[011101]$ le mot de code reçu. Vérifions si r est erroné :

Calcul du syndrome : $S=r.H^{T}=[011101].\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}=[110]\neq[000]\rightarrow r$ est erroné

$S=[110]$ se trouve dans la 3ème colonne de H donc l'erreur se trouve sur la 3ème position de r c'est-à-dire : $R=[0\ 1\ 1\ 1\ 0\ 1]$ d'où après correction on aura :

$$r=[0\ 1\ 0\ 1\ 0\ 1]$$

3- Détection par vérification de parité

- Vérification de parité par caractère VRC

Principe

Le message est sectionné par caractère de i bits. On ajoute à chaque caractère un bit de parité tel que: $N = i + 1$, $N =$ caractère + le bit de parité (on compte le nombre de bit qui valent « 1 » soit paire ou impaire). Ce contrôle de parité est appelé VRC.

Exemple 1.2.8

On transmet quatre caractères de 5 bits chacun.

A	B	C	D
1	0	0	1
0	1	0	1
1	1	1	0
1	0	0	1
0↓	0↓	0↓	0↓
1	0	1	1

- Si le nombre de « 1 » est paire, le bit de parité serait égal à « 0 » ;
- Si le nombre de « 1 » est impair, le bit de parité serait égal à « 1 ».

Remarque : la distance de Hamming est égale à 2. Les erreurs simples sont détectées (ainsi que les erreurs d'ordre impair). Par contre les erreurs doubles ne sont pas détectées.

Si on envoie par exemple le caractère A de bit de parité égal à 1, le récepteur vérifie le bit parité :

$$\text{Si le bit de parité est égal à } \begin{cases} 1 & \text{ca veut dire qu'il y a pas d'erreur} \\ 0 & \text{il y a au moins une erreur} \end{cases}$$

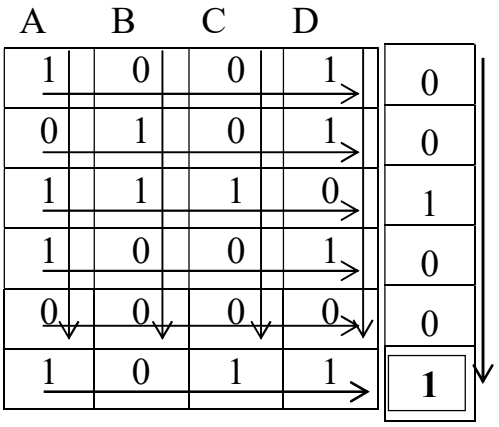
- Vérification de parité croisée (LRC/VRC)

Principe

On peut regrouper les caractères en bloc et ajouter à chaque fin de bloc un caractère supplémentaire LRC combiné avec la vérification de parité VRC. Chaque bit du caractère LRC est égal à la parité des bits du même rang de tous les caractères du bloc.

Exemple 1.2.9

Remarque : la distance de Hamming est égale à 4, le changement d'un bit de donnée entraine la modification d'un bit du VRC et un bit du LRC et de la parité croisée soient 4 bits en tous.



Si les parités VRC et LRC sont différentes alors il y a une erreur.

Un tel code détecte toutes les erreurs simples, doubles et triples et peut corriger toutes les erreurs simples.

1.2.7 Le nombre d'erreurs détectable et d'erreurs corrigeable [5]

Un code en blocs linéaire est souvent noté $c(N, K, d_{min})$. Un code en blocs linéaire est capable de détection toutes les configurations comportant ($d_{min} - 1$) erreurs dans un bloc de N éléments binaires et de corriger t erreurs avec :

$$t = \left[\frac{d_{min} - 1}{2} \right] \tag{1.17}$$

Où [.] signifie l'entier le plus proche.

Exemple 1.2.10

Soit d_{min}d'un mot de code :$d_{min} = 7$

- Nombre d'erreurs détectable = $d_{min} - 1 = 7 - 1 = 6$;
- Nombre d'erreurs corrigeable = $t = \left[\frac{d_{min} - 1}{2} \right] = \left[\frac{7-1}{2} \right] = 3$.

1.2.8 Autres codes en blocs [12]

1- Les codes BCH

Ce sont des codes qui ont la plus grande capacité de correction d'erreurs indépendantes pour une redondance et une longueur données. Leur rendement n'est pas très élevé. Ce sont une extension des codes cycliques, ils sont non pas construit sur un alphabet binaire mais un alphabet composé d'un grand ensemble de symboles.

2- Les codes RS

Le code Reed-Solomon est un dérivé des codes CRC. Il a été découvert par I.S. Reed et G. Solomon qui ont publié leurs recherches en 1960.

A l'époque, les besoin d'un code correcteur puissant n'étaient pas encore nécessaires. C'est seulement quelque année plus tard que ce code c'est révélé utile. Ce code est particulièrement efficace lorsqu'il s'agit de corriger des rafales d'erreurs importantes.

Egalement il permet de reconstituer des messages alors que des paquets entiers d'information sont perdus.

A la différence des autres codes, celui-ci code des paquets des bits : par exemple, prenons un code Reed-Solomon pour 3×8 bits de donnée, on transmet deux paquets supplémentaires contenant les bits de contrôle.

Ainsi pour le message$03\ 10\ 15$, on génère les mots de contrôle $03 + 10 + 15 = 28$ et $03 \times 1 + 10 \times 2 + 15 \times 3 = 68$ ainsi le message envoyé est$03\ 15\ 18\ 28\ 68$.

Pour le décodage, supposons que l'on reçoive le message erroné $03\ 12\ 15\ 28\ 68$ alors en calculant les bits de contrôle $03 + 12 + 15 = 30$, $03 \times 1 + 12 \times 2 + 15 \times 3 = 72$, on détecte une erreur de calcul qui est corrigeable en faisant la différence des premiers mots de contrôle $(30\ 28) = 2$ qui donne la valeur de l'erreur et la différence des deuxième mots pondérés par l'erreur donne le mot erroné $((72\ 28)/2) = 2$.

1.3 Les codes convolutifs (ou convolutionnel) [5, 8, 12]

1.3.8 Introduction et définition

Introduits par Peter Elias en 1954[12], ce sont probablement les codes les plus populaires. On trouve ceux-ci dans de nombreuses applications : communication sans fils (IMT-2000, GSM, IS-95), communication terrestre et satellitaire, communication spatiale. Leur méthode de décodage la plus populaire repose sur l'algorithme de Viterbi. Récemment, il a été montre que les codes convolutifs combinés avec l'entrelacement dans les schémas de concaténation pouvait s'approcher de la limite théorique de Shannon.

La sortie d'un codeur convolutif dépend d'un symbole courant à coder ainsi que du symbole précédent et du résultat de codage du symbole précédent.

Chacune des sorties du codeur est égale au produit de convolution (d'où l'expression « convolutif ») entre la suite binaire présente à l'entrée du codeur et la réponse de ce même codeur définie par ses séquences génératrice [8].

1.3.2 Présentation du codeur

Pour un code convolutif, à chaque instant k le codeur délivre un bloc de n symbole binaire $c_k = (c_{k,1}, c_{k,2}, ..., c_{k,n})$, en fonction du bloc de k symboles d'information $d_k = (d_{k,1}, d_{k,2}, ..., d_{k,k})$ présente à son entrée avec m précédent blocs. Ce codage introduit un effet mémoire de l'ordre m. La quantité v=m+1 est appelée la longueur de contrainte du code et le rapport $r = \frac{k}{n}$ est appelé le rendement du code.

On dit que le code est donc systématique, si k symbole d'information à l'entrée du codeur se trouve explicitement dans le bloc codé c_k tel que :

$$c_k = (d_{k,1}, d_{k,2}, ..., d_{k,k}, c_{k,1}, c_{k,2}, ..., c_{k,n}) \quad (1.18)$$

Dans le cas contraire il est connu comme non systématique. Le schéma général d'un codeur avec une sortie $\frac{k}{n}$ et une mémoire m est représenté sur la figure suivante :

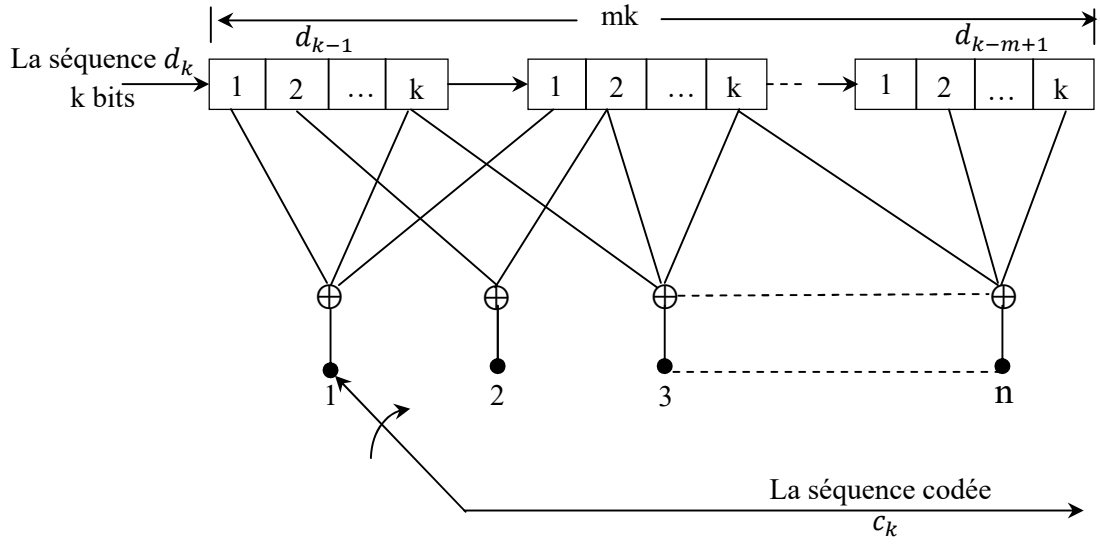


Figure 1.7.Schéma d'un codage convolutif

- n : le nombre de sorties,
- k : le nombre d'entrées,
- m : le nombre de cellules de tous les étages.

A chaque instantk, le codeur a m blocs de k symboles d'information dans la mémoire. Ces mk symboles binaires définissent l'état S_k du codeur :

$$S_k = (d_k , d_{k-1},\ldots,d_{k-m+1}) \tag{1.19}$$

Supposons que pour chaque séquence codée$i = 1, \ldots, n$, on associe une transformée en D définie par :

$$c_i(D) = \sum_k c_{k,i} D^k = c_{0,i} D^0 + c_{1,i} D^1 + \ldots + \text{avec } i = 1,\ldots,n \tag{1.20}$$

La variable D représente l'opérateur de retard unitaire et $c_i(D)$ est la transformée en D de $c_{k,i}$.

Chaque symbole codé $c_{k,i}$, peut être exprimé comme une combinaison linéaire des k symbole d'information présente à l'entrée du codeur et km symbole contenu dans sa mémoire :

$$c_{k,i} = \sum_{l=1}^{k} \sum_{j=0}^{m} g_{i,j}^{i} d_{k-j,l} \tag{1.21}$$

Où les coefficients $g_{l,j}^{i}$ prennent des valeurs de 0 ou 1 et les opérations d'addition sont réalisées en modulo 2. C'est une convolution entre $g_{l,j}^{i}$ etd_k. En prenant la transformée en D, on trouve :

$$c_i(D) = \sum_{l=1}^{k} G_l^i(D) d_l(D) \tag{1.22}$$

$$\text{avec } G_l^i(D) = \sum_{j=0}^{m} g_{l,j}^{i} D^j$$

$$d_l(D) = \sum\nolimits_p d_{p,l}(D^p)$$

Les quantités $G_l^i(D), 1 \le l \le k$, $1 \le i \le n$ sont appelés polynômes générateurs du code.

Introduisant les notations matricielles :

$$d(D) = (d_1(D), ..., d_k(D))$$

$$c(D) = (c_1(D), ..., c_n(D))$$

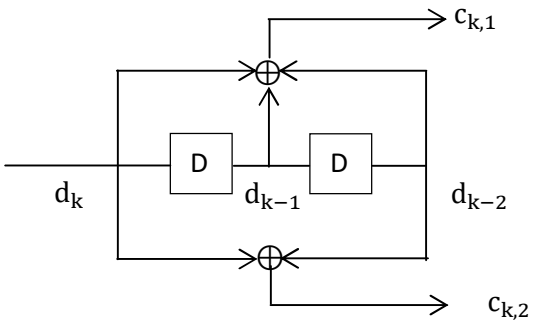
$$c(D) = d(D) \times G(D) \tag{1.23}$$

Avec G(D) la matrice génératrice du code :

$$G(D)=\begin{bmatrix} G_1^1(D) & \cdots & \cdots & G_1^i(D) & \cdots & \cdots & G_1^n(D) \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ G_1^1(D) & \cdots & \cdots & G_1^i(D) & \cdots & \cdots & G_1^n(D) \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ G_k^1(D) & \cdots & \cdots & G_k^i(D) & \cdots & \cdots & G_k^n(D) \end{bmatrix} \quad (1.24)$$

Exemple1.2.10

Considérons le code convolutif non systématique suivant, avec $m = 2$ nombre de mémoires, i.e. la longueur de la contrainte est $v = 3$ et le rendement est $r = 1/2(k = 1, n = 2)$.



$$G(D) = (G^1(D), G^2(D))$$

$c_{k,1} = d_k \oplus d_{k-1} \oplus d_{k-2} = g^1 \otimes d_k$ (produit de convolution)

$$c_{k,2} = d_k \oplus d_{k-2} = g^2 \otimes d_k$$

La matrice génératrice du code est de la forme suivante :

$$G(D) = [G^1(D),\ G^2(D)]$$

$$g_{1,0}^1 = g_{1,1}^1 = g_{1,2}^1 = 1$$

$$g_{1,0}^2 = g_{1,2}^2 = 1,\ g_{1,1}^2 = 0$$

$$G^1(D) = 1 + D + D^2 \quad \text{et} \quad G^2(D) = 1 + D^2$$

$$\rightarrow G(D) = [1 + D + D^2, \quad 1 + D^2]$$

On peut associer une représentation binaire des polynômes générateurs :

$$G^1(D) \rightarrow g^1 = (1, 1, 1) \rightarrow g^1 = 7_{octal}$$

$$G^2(D) \rightarrow g^2 = (1, 0, 1) \rightarrow g^2 = 5_{octal}$$

C'est le code convolutif C(5,7).

Supposons que le message à transmettre est$d = (10011)$, en utilisant la transformée en$D$, on trouve $d(D) = 1 + D^3 + D^4$. Les deux sorties correspondantes sont données en transformée en D par :

$C^1(D) = G^1(D)\, d(D) = (1+D+D^2)\,(1+D^3+D^4) = (1+D+D^2+D^3+D^6)$

$C^2(D) = G^2(D)\, d(D) = (1+D^2)\,(1+D^3+D^4) = (1+D^2+D^3+D^4+D^5+D^6)$

On déduit ainsi les sorties en binaire $C^1 = (1111001)$ et $C^2 = (1011111)$. On construit maintenant le mot de code par multiplexage des deux sorties :

$$c = (11, 10, 11, 11, 01, 01, 11)$$

On remarque que le message d de longueur 5 bits est codé en un mot de code c de 14 bits.

1.3.3 Représentations graphiques des codes convolutifs

1- Diagramme de transition d'états

Le fonctionnement d'un codeur convolutif peut être représenté par un graphique appelé diagramme de transition d'état.

Reprenons l'exemple précèdent, le code $C(5,7)$. L'état de l'encodeur à l'instant k noté $S_k = (d_k, d_{k-1})$ peut prendre quatre valeurs indépendamment de k. Le nombre $a = (00), b = (01), c = (10), d = (11)$.

En s'appuyant sur le tableau ci-dessous, on construit le diagramme de transition d'états.

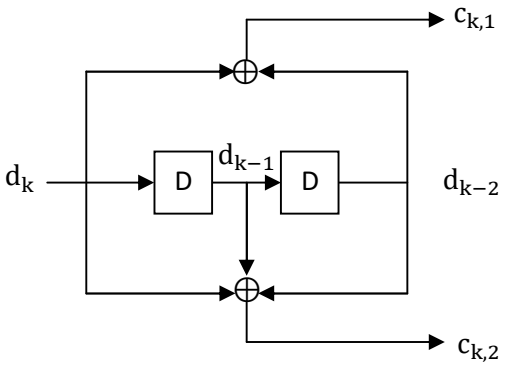


Figure 1.8. Diagramme de transition d'états du code convolutif C(5,7)

d_k	S_k(t)	S_k(t+1)	$c_{k,1}$	$c_{k,2}$
-	00	00	-	-
0	00	00	0	0
1	00	10	1	1
0	01	00	1	1
1	01	10	0	0
0	10	01	1	0
1	10	11	0	1
0	11	01	0	1
1	11	11	1	0

Tableau 1.3. Séquences de sorties et séquence d'entrée et les états du code C(5,7) de l'exemple précédent

Les étiquettes de chaque branche correspondent aux sorties du codeur ($c_{k,1}c_{k,2}$). Seules deux transitions sont possibles à partir de chacun des états.

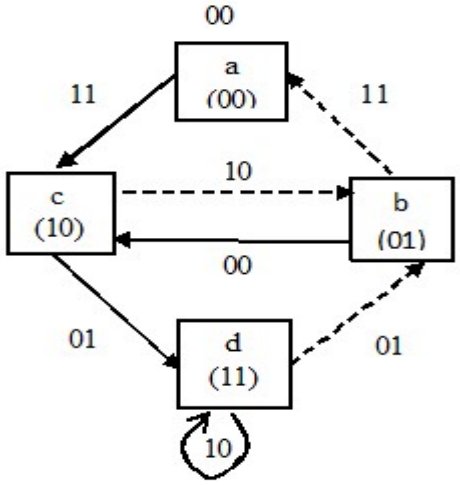


Figure 1.9.Diagramme de transition d'état de l'exemple précédent

2- Diagramme en treillis

Une autre représentation de l'opération de codage, indiquant l'évolution de son état S_k à travers le temps, est possible. C'est le diagramme en treillis. L'intérêt de cette représentation par rapport au diagramme de transition d'états est son utilité lors du décodage des codes convolutif.

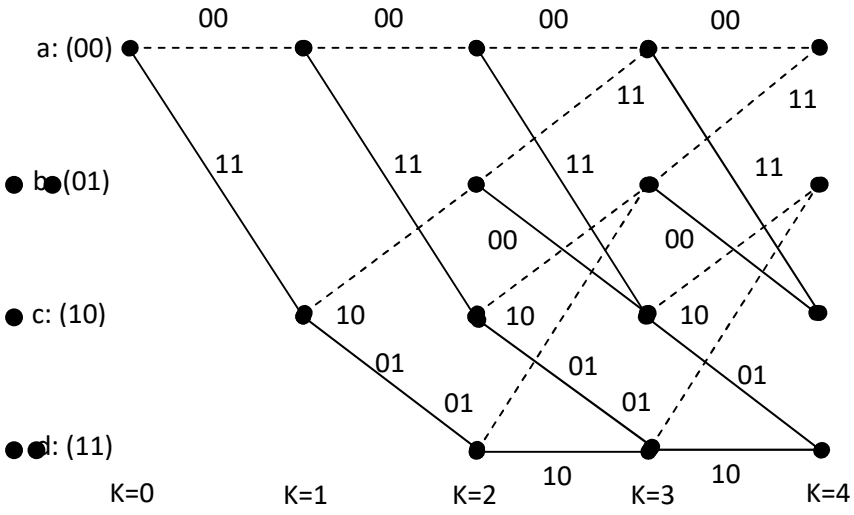


Figure 1.10.Diagramme en treillis d'un code convolutif (5,7)

- De chaque nœud partent 2^k branches.
- Les étiquettes de chaque branche correspond aux sorties du codeur $(c_{k,1}, c_{k,2})$.
- Les transitions sont les entrées possibles ; pour un bit d'entrée égale à 1, la branche est un trait noir, sinon c'est un trait en pointillés.
- Les mots de codes sont tous les chemins possibles dans le treillis donnés par la concaténation des étiquettes.

1.3.4 Décodage convolutif

Dans les canaux de communication sans mémoire, les systèmes utilisant le codage convolutif sont parmi les plus intéressants tant du point de vue de leurs performances (s'approchant le plus des performances ultimes prévues par la théorie de Shannon) que du point de vue de leur réalisation et implantation matérielle. Les deux principales techniques de décodage des codes convolutifs sont le décodage de Viterbi et le décodage séquentiel.

Chacune de ses techniques consiste à trouver un chemin particulier (le message transmis), dans un graphe orienté où on assigne aux branches des métriques ou valeurs de vraisemblance entre les données reçues et les données qui auraient pu être transmises.

L'objectif général du décodeur se résume donc à déterminer avec la plus grande fiabilité etle minimum d'efforts le chemin de métrique minimale. Ce chemin est la séquence décodée.

Les algorithmes de décodage sont basés selon certains critères ; on considère le décodage fermé « hard », lorsqu'il utilise des entrées binaires résultant du seuillage de la sortie du canal ou le décodage souple « soft », lorsqu'il utilise des entrées réelles.

Le décodage convolutifs, algorithme se viterbi, consiste à chercher dans le treillis le chemin qui correspond à la séquence la plus probable, c'est-à-dire celle qui est à la distance minimale de la séquence reçue : pour cela on introduit les deux catégories de décisions.

- Décision dure
- Décision souple

On utilise l'algorithme de viterbi.

1.4 Les codes approchant la limite de Shannon[5,16]

1.4.8 Les turbo-codes

Ils sont obtenus par la concaténation (en série ou en parallèle) de deux ou plusieurs codes de faible complexité (des codes convolutifs), séparés par une fonction d'entrelacement (voir ci-après) introduisant de la diversité ou l'aléatoire et c'est justement cette structure qui rend ces codes plus performant et avoisinant les limites de Shannon, notamment avec l'utilisation d'algorithme de décodage itératif.

L'entrelacement : consiste à permuter une séquence de bits de manière à ce que deux symboles proches à l'origine soient le plus éloignés possibles l'un de l'autre. Cela permet en particulier de transformer une erreur portant sur des bits regroupés en une erreur répartie sur l'ensemble de la séquence. Un exemple classique de l'utilité de l'entrelacement est celui des codes correcteurs d'erreurs protégeant les disques compacts : il s'agit d'un code en bloc de Reed-Salomon entrelacé qui permet de corriger plus de 4 000 erreurs consécutives dues à une poussière, une éraflure… On cherche également, en particulier pour les turbo-codes, à réaliser une permutation aussi « chaotique » que possible.

1.4.2 Les codes LDPC [16]

Les codes LDPC font partie de la classe des codes blocs linéaires et s'approchent d'avantage de la limite de Shannon (capacité d'un canal) [6].

Un code LDPC est un code dont la matrice de contrôle de parité H est de faible densité. La faible densité signifie qu'il y a plus de 0 que de 1 dans la matrice H. Un code LDPC

peut être représenté sous forme matricielle ou bien sous la forme d'un graphe bipartite (représentation de Tanner).

Conclusion : après avoir passé en revue les techniques de codage…… (un petit résumé comparatif sis possible des différentes techniques citées), nous allons introduire dans la chapitre qui suit un nouveau type de code correcteur d'erreur appelé dans la littérature codes polaire (…. explication brève de son principe).

1.5 Conclusion

Dans ce chapitre, nous avons fait une étude générale des codes correcteurs d'erreurs en commençant par détailler les codes en blocs à savoir les codes en blocs linéaires, les codes cycliques et ensuite nous avons détaillé brièvement des codes en blocs plus performants tel que les codes BCH et ceux de Reed Salomon.Nous avons, par la suite, abordé les codes convolutifs et terminé ce chapitre par un bref aperçu des codes approchant la limite de Shannon tel que les turbo codes et les codes LDPC.

Ce chapitre nous a permis d'avoir une idée générale sur les codes correcteurs d'erreurs les plus utilisés.

Des codes en bloc linéaire plus performants en raison de la prévention, de la détection et de la correction d'erreursferont l'objet duchapitre qui suit, ce sont les codes polaires.

Chapitre 2 Les codes polaires

2.1 Introduction

Les codes polaires sont des codes en blocs linéaires. Ils ont fait l'objet d'une recherche active ces derniers temps, principalement en raison du fait qu'ils sont les premiers codes atteignant la capacité des canaux BMS [9], avec une construction explicite et une très faible complexité de codage et de décodage en utilisant un algorithme particulier appelé algorithme par annulation successive SC.

Il a été prouvé mathématiquement, en 2008, par Erdal Arıkan [1]que les codes polaires étaient capables de corriger toutes les erreurs de transmission sous certaines conditions.

En effet, ayant été inventé par Arikan[1], les codes polaires utilisent un concept nouveau appelé polarisation de canal[17]. Peu de temps après, le concept de polarisation des canaux ainsi que les codes polaires ont été étendus à un certain nombre d'applications et de généralisations et des améliorations ont été introduites de sorte que les performances du canal ont maintenant presque fermé l'écart à la limite de Shannon qui, définit la barre pour le débit maximal pour une bande passante donnée et un niveau de bruit donné [11].

Le code devrait aider à corriger les erreurs de transmission ainsi qu'à augmenter les débits et l'efficacité spectrale des réseaux cellulaires.

Les codes polaires sont constitués de trois (3) blocs essentiels qui sont :

- la construction du code
- le codage polaire et
- le décodage polaire.

Sur le plan de leurs applications dans l'industrie des télécommunications, le groupe Huawei a annoncé, en octobre 2016, qu'elle avait réalisé un débit de 27Gbps dans des tests d'essais sur champs 5G en utilisant les codes polaires pour le codage des canaux.

Lors de la rencontre 87 RAN1organisée le 17 novembre 2016 par le 3GPP, organisme international de développement des standards pour les télécommunications mobiles, les codes polairesont été choisi comme schéma de codage du canal de contrôle pour l'application de la 5G dans le scénario eMBB (*Enhanced Mobile Broadband* ou amélioration du haut-débit mobile).

2.2 Construction du code

2.2.1 Canal de transmission[9]

1- Définition

Le canalde transmission, du point de vue théorie de l'information, est situé entre la sortie du codeur de canal et l'entrée du décodeur de canal, alors que du point de vue radio fréquence il serait situé entre les antennes d'émission et de réception.

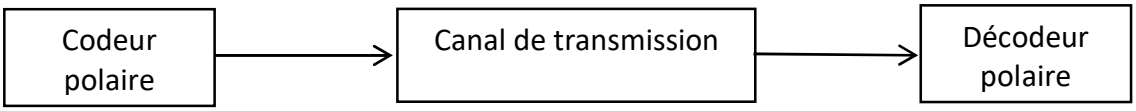


Figure 2.1. Canal de transmission du point de vue théorie de l'information

Un milieu de transmission est un support transportant de l'information comme par exemple l'air, une paire de fils torsadés, un câble coaxial, une fibre optique…etc.

Par extension,les supports physiques tels que les disques durs ou les CD, sont également des milieux de transmission.

De manière générale, un canal de transmission est défini mathématiquement par deux ensembles et une probabilité de transition d'un ensemble vers l'autre :

- un ensemble X, contenant toutes les entrées possibles
- un ensemble Y, contenant toutes les sorties possibles
- une probabilité de transition notée $\mathrm{P}(\mathrm{Y} \mid \mathrm{X})$.

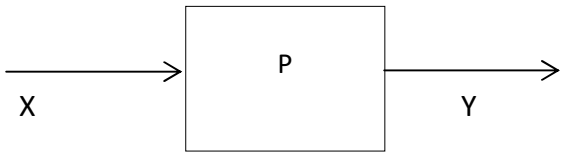


***Figure 2.2*.Schéma bloc d'un canal de transmission**

Un canal de transmission subit des perturbations de différents types selon la nature du canal. Ces perturbations peuvent être de type électromagnétique, de type thermique ou des interférences liées à la présence de plusieurs utilisateurs sur le canal.

Le canal prend en entrée des données binaires et produit en sortie une estimation des bits originaux. Cette estimation peut être dure(*hard*) ou souple (*soft*). Une sortie dure implique une prise de décision 0 ou 1 quant à la valeur des bits de sortie. Une sortie souple implique que la valeur du bit en sortie soit représentée par une probabilité de sa valeur.

Des valeurs souples permettent d'obtenir en général de meilleurs résultats lors du décodage du message mais nécessitent une complexité calculatoire supérieure à celle nécessaire pour des valeurs dures.

Il existe quatre propriétés principales des canaux :

- discrétisation : un canal discret est un canal de communication qui ne transmet que des nombres entiers ou, plus généralement, un nombre fini de symboles.

Le plus souvent, il s'agit d'un canal binaire qui ne transmet donc quedes 0 oudes 1 comme indiqué en figure 2.3 (les canaux BSCet BEC) seront examinés en détail par la suite.

Figure 2.3.a)canal binaire symétrique (BSC) avec $0 \leq P_e \leq 1/2$,b)canal binaire à effacement (BEC) avec εl'effacement et P_e la probabilité de l'effacement.

- continuité : un canal continu possède un alphabet d'entrée et de sortie qui est à valeur dans l'ensemble de réels.

- stationnarité : un canalstationnairepossède une probabilité de transition qui ne dépend pas du temps.

- effet mémoire : un canal sans effet mémoire implique qu'un symbole en sortie ne dépend que du symbole d'entrée.

2- Capacité d'un canal [6]

La capacité d'un canal est définie par l'information mutuelle maximale entre une variable aléatoire X à valeur sur l'alphabet d'entrée du canal et sa sortie correspondante Y par le canal.

$$\mathrm{C} \stackrel{\text{def}}{=} \sup \mathrm{I}(\mathrm{X};\mathrm{Y}) \qquad (2.1)$$

On remarquera que $\mathrm{I}(\mathrm{X};\mathrm{Y})$ peut s'écrire en fonction des seules lois de transition et d'émission :

$$I(X;Y) = \sum_{x,y} P(y|x)P(x)\log_2 \frac{P(y|x)}{P(y)} \qquad (2.2)$$

$$P(y) = \sum_{x} P(y|x)P(x) \qquad (2.3)$$

Où$P(y|x)$désigneprobabilité d'obtenir Y sachant X a été envoyée.

Rappelle : la limite (ou la capacité) de Shannon est de -1.6 dB

3- Les principaux canaux BMS

- **Canal binaire à effacement BEC**

Le canal à effacement BEC est un canal discret, stationnaire et sans effet mémoire. Sa sortie est constituée de l'ensemble {0, ε,1}.

Le symbole ε représente un effacement (ou bit effacé).L'effacement correspond à une perte totale d'information permettant de prendre une décision quant à la valeur reçue.C'est-à-dire que le bit est brouillé, de sorte que le récepteur n'a aucune idée de ce qu'était le bit à l'émission. Ce canal est souvent utilisé comme modèle pour les réseaux du type internet. L'effacement modélise alors la perte d'un paquet.

Par exemple, dans les communications internet, des paquets IP sont utilisés, et certains n'atteignent pas le destinataire car le routeur peut avoir fait une mauvaise redirectionou bien ces paquets sont perdus en raison de débordements des mémoires tampon ou à cause de retards excessifs. Les probabilités de transitions du canal sont représentées dans la figure2.3.b et peuvent être exprimées comme suit :

$$\Pr(\text{effacement}) = P_e \qquad (2.4)$$

$$P_e \in [0 \;\; 1]$$

$$\Pr(y_i = 0 \mid x_i = 0) = \Pr(y_i = 1 \mid x_i = 1) = 1 - P_e \qquad (2.5)$$

4- Canal binaire symétrique BSC

Le canal binaire symétrique BSC est un canal discret, stationnaire et sans effet mémoire.Ses ensembles d'entrée $X = \{0,1\}$ et de sortie $Y = \{0,1\}$ sontde dimensions finies.

Puisque ce canal est stationnaire et sans effet mémoire, les probabilités de transitions sont indépendantes du temps et l'élément y_k ne dépend que de l'élément x_k .

Les probabilités de transitions du canal (probabilité d'obtenir la sortie sachant l'entrée) sont représentées dans la figure2.3.aet peuvent être exprimées comme suit :

$$\Pr(y_i = 0 \mid x_i = 1) = \Pr(y_i = 1 \mid x_i = 0) = P_e \qquad (2.6)$$

$$\Pr(y_i = 0 \mid x_i = 0) = \Pr(y_i = 1 \mid x_i = 1) = 1 - P_e \qquad (2.7)$$

Avec $P_e$la probabilité d'erreur, souvent appelée probabilité croisée.$P_e \in [0 \;\; 0.5]$.

5- Canal gaussien à entrée binaire (B-AWGN)

Le canal B-AWGNreprésenté dans la figure2.4, est un canal continu, stationnaire et sans effet mémoire. Il est couramment utilisé en communications numériques car il est simple d'utilisation et permet de fournir une première approximation des phénomènes qui s'appliquent sur un canal de transmission réel.

Figure 2.4.Canal B-AWGN

Le canal AWGN possède les caractéristiques suivantes :

- il est à une entrée x_k discrète et à valeur dans {0,1} dans le cas d'un canal binaire ;
- il est à sortie y_k continue $y_k = x_k + n_k$, avec $n_k$la valeur discrète du bruit
- il est stationnaire (ne dépend pas du temps) ;
- il est sans effet mémoire (y_k ne dépend que de x_k)

Le bruit blanc gaussien WGNest un bruit dont la densité spectrale de puissance est la même pour toutes les fréquences (bruit blanc). Il est dit additif car il est simplement ajouté au signal entrant. Enfin, il est dit gaussien du fait de sa densité de probabilité de transmission est définie comme suit :

$$\Pr(x_i \mid y_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x_i - y_i)^2}{2\sigma^2}} \qquad (2.8)$$

La variance σ du signal est fonction du rapport signal à bruit (SNR) noté :

$$SNR = \frac{E_b}{N_0} = \frac{1}{2\sigma^2} \qquad (2.9)$$

$\sigma \in [0 \ \infty]$.

Avec E_bl'énergie moyenne utilisée pour transmettre un bit (ou énergie bit) utile et N_0 la densité spectrale de puissance du bruit.

$N_0 = KT, \ K = 1{,}38\ 10^{-2}$ J/°K et T: température en °K.

Exemple : cas d'un canal gaussien en visibilité directe avec un effet multi-trajet négligeable etoù la modulation utilisée est la BPSK[19]:

- Le modulateur BPSK mappe en premier lieu les bits 0 et 1 dans le plan I/Q

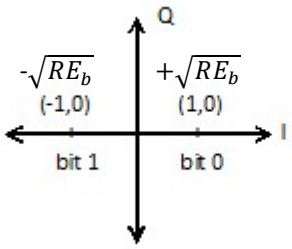


Figure 2.5. Schéma de constellation d'une modulation BPSK

Lacomposante Q est ignorée puisqu'elle n'affecte pas la sortie du démodulateur.

- Les points I/Q sont convertis en sinusoïdes :

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \quad \text{pour le bit 1} \quad (2.10)$$

$$s_0(t) = -\sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \quad \text{pour le bit 0} \quad (2.11)$$

AvecT_b la durée bit ou inverse du débit binaire.

- le récepteur corrige la forme d'onde (du« 0 » logique ou du « 1 » logique) pour compenser l'atténuation
- Le démodulateur mappe la forme d'onde dans l'espace I / Q et produit le résultat
- L'effet global est que $0 \to 1 + n$ et$1 \to -1 + n$, où n est une variable désignant un bruit aléatoire supposée être distribuée par la loi normale avec une moyenne nulle.

Pour toute valeur$\sigma > 0$, le canal gaussien avec une variance σ^2 à sont ensemble de sortie Y dans $\mathbb{R}$ (ensemble des réels) et des probabilités de transition :

$$P(y|0) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-1)^2}{2\sigma^2}} \quad (2.12)$$

Une loi de distribution normale avec une moyenne égale à 1 et déviation standardσ.

$$P(y|1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y+1)^2}{2\sigma^2}} \quad (2.13)$$

Une loi de distribution normale avec une moyenne égale à -1 et déviation standard σ.

Enfin, la probabilité d'erreur d'un canal AWGN avec une modulation BPSK est fonction du SNR. La probabilité d'erreur binaire d'un tel canal avec une telle configuration est donnée par la formule suivante (Proakis, 2000) [14] :

$$P_{eb} = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (2.14)$$

Où erfc désigne la fonction d'erreur complémentaire :

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2.15)$$

2.2.2 Polarisation du canal[17]

Les codes polaires se construisent par la méthode de polarisation du canal.

Cette méthodeconsiste à transformer le canal initial en N canaux binaires différents, tel que quand N tend vers l'infini, une partie de ces canaux binaires auront une capacité qui tend vers "1"et seront considérés comme les "bons canaux" alors que les autres canaux binairesauront une capacité qui tend vers "0" et serontconsidérés les "mauvais canaux"[17].

- Concept de polarisation :

La polarisation a été introduite par Erdal Arikan en 2008 dans son document phare [2].

Arikan a utilisé cette technique dans le contexte du codage canal et dans une classe spéciale de canaux, ceux du type BMS. Le concept de polarisation peut s'expliquer comme suit :

Soit P un canal BMS et $I(P)$ sa capacité. L'idée de polarisation est de prendre deux copies indépendantes de P pour créer, à partir decelles-ci, deux nouveaux canaux P^- et P^+ tel que :

- la somme des capacités de P^- et P^+est égale au double de la capacité deP, ainsi donc aucune information n'est perdue

- P^+ est « meilleur » que P et P^-est « pire » que P

Considérons le schéma de la figure (2.6) :

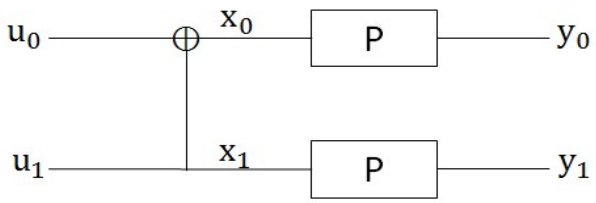


Figure 2.6.Transformation du canal P en deux copies (utilisations) indépendantes

$$x_0 = u_0 \oplus u_1 \quad (2.16)$$

$$x_1 = u_1 \quad .(2.17)$$

$\oplus$ désigne l'opérateur « OU exclusif ».

On peut dès lors affirmer que :

$$u_0 = x_0 \oplus x_1 \quad (2.18)$$

et que :

$$u_1 = x_1 \quad (2.19)$$

Puisque la matrice, notée G reliant $[u_0, \ u_1]$ à $[x_0, x_1]$ est son propre inverse :

$$X = U \times F \text{ avec } F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ la matrice génératrice} \quad (2.20)$$

Étant données les sortie (y_0, y_1), on suppose que l'on désire déduire les valeurs des entrées (u_0, u_1). Cette tâche peut être accomplie de manière successive :

- en première étape on déduit la valeur deu_0, u_1 est considérée comme un bruit
- en seconde étape on utilisera l'estimé deu_0 , noté $\hat{u}_0$, pour déduire u_1

De ce point de vue, on peut affirmer que :

- le canal P^-est le canal que $u_0$voit étant donné les observations (y_0, y_1),
- le canal P^+ est le canal que u_1 voit étant donnée les sorties (y_0, y_1) et la valeur actuelle de u_0 soit$\hat{u}_0$.

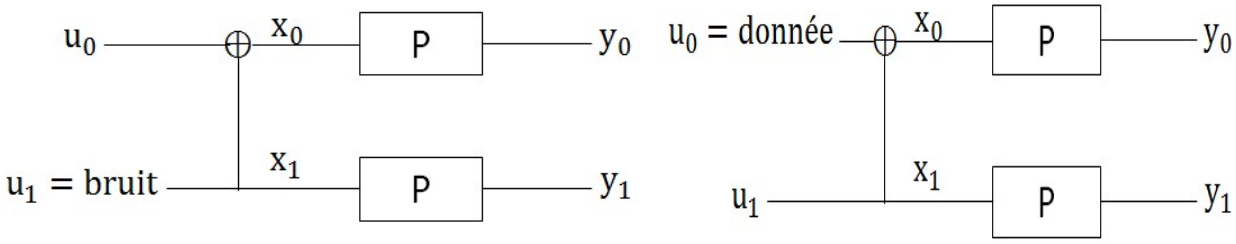


Figure2.7.à gauche : canalP^- (mauvais) vue par u_0 , à droite et canal P^+(bon) vu par u_1

Intuitivement, on peut affirmer que P^- est le mauvais canal (canal dégradé par rapport àP) et P^+le bon canal (canal optimisé par rapport àP).

Autrement dit, P^+ est la version à « moindre bruit » de P et P^- est la version « à plus de bruit » de$P$.

En conséquence, à partir de deux copies du canal deux variables indépendantes et identiquement distribuées (iid) du canal P, nous avons créé deux canaux P^- et P^+avec la propriété :

$$P^- \leq P \leq P^+ (2.21)$$

La relation (2.21) ci-dessus nous indique déjà une sorte de polarisation vers les deux canaux extrêmes: totalement sans bruit (canal parfait ou meilleur canal) ou complétement bruité (canal inutile ou mauvais canal).

L'idée de la polarisation est d'appliquer récursivement cette simple transformation aux canaux P^{-}et P^{+}pour créer d'autres canaux polarisées :

$P^{(+)+}$, $P^{(+)-}$

$P^{(-)+}$, $P^{(-)-}$

Ce processus continuera jusqu'à créer un certain nombre $N = 2^{n}$ canaux (nombre limite que nous verrons par la suite) :

$P^{++\cdots+}$, $P^{++\cdots-}$, $P^{--\cdots-}$…etc.

Un phénomène remarquable apparait est que : lorsque n augmente, les canaux créés deviennent de plus en plus polarisés, c'est-à-dire que presque tous les canaux sont très proches de l'un des deux états extrêmes suivants:

- totalement sans bruit (avec capacité 1)

Ou

- complètement bruités (avec capacité 0).

Il est clair que pour longueurs de blocs très grandes la fraction de canaux complètement sans bruit tend vers $I(P)$ et la fraction de les canaux complètement bruité tend vers$1 - I(P)$.

La complexité de codage et décodage des codes construits par cette méthode est, par rapport à la longueur du mot de code N, $O(N \log_2(N))$ avec O l'ordre de grandeurdans les processus de codage et de décodage [10]. Pour un décodeur à maximum de

vraisemblance ML, le nombre d'opération est $O(2^K)$avec $K = RN$ et où Rreprésente le taux de codage[19].

Exemple : Si $N = 64$ et $R = 1/2$alors$K = 32$. Dans ce cas $O(N\log_2(N)) = 384$et$O(2^K) = 4294967296$.

La polarisation du canal permet d'identifier les indices des K bits de message à envoyer notés I et les indices des bits faiblesnotés I_c.

On utilise lesbons canaux pourenvoyer les Kbits d'information et on envoie sur les mauvais canaux une séquence de bits connueau niveau du décodeur.Ces derniers sont appelés bits gelés (*frozen bits*).

La somme des capacités de ces bons canaux tend vers la capacité du canal initial ce qui permet de communiquer avec un débit proche de la capacité du canal.

Pour mieux comprendre laméthode de polarisation, considérons le canal initial

$$P: X \to Y$$

Ce canal est un canal symétrique ayant :

- un alphabet d'entrée binaire$X = \{0,1\}$
- un alphabet de sortie Yet
- une probabilité de transition $P(Y \mid X)$.

Rappels : $P(Y|X)$ est une probabilité conditionnelle ou probabilité de Y sachant X a été réalisé. X étant de probabilité non nulle.

En utilisant la formule suivante [15]

$$P^-(y_0, y_1|u_0) = \sum_{u_1 \in \{0,1\}} \frac{1}{2} p(y_0|u_0 \oplus u_1) p(y_1|u_1). \quad (2.22)$$

$$P^{+}(y_0, y_1, u_0|u_1) = \frac{1}{2}P(y_0|u_0 \oplus u_1)p(y_1|u_1) \quad (2.23)$$

On transformeP en deux canaux distincts :

$P^{-}: X \to Y^2$ et $P^{+}: X \to X \times Y^2$

Autrement dit :

$P = \{0,1\} \to$ Yalors$P^{-} = \{0,1\} \to Y^2$et$P^{+} = \{0,1\} \to \{0,1\} \times Y$

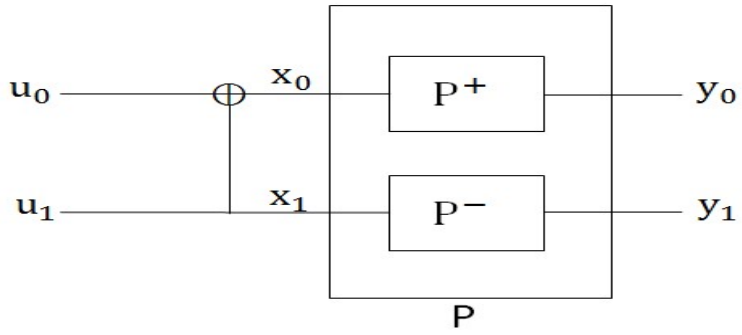


Figure 2.8. Codeur CP(2,2) (voir §2.3)

- $U = [u_1, u_2]$: vecteur d'information
- $\oplus$: XOR (ou exclusif)
- P_2 : canal initial polarisé en deux canaux binaires P^{-}et P^{+}
- y_1 et y_2 : vecteur en sortie du canal.

Puis, on applique cette même transformation sur les canaux P^{-}et P^{+}pour avoir quatre canaux distincts et ainsi de suite. En réalisant cette division nfois on obtient :

$$N = 2^n \text{canaux.} \quad (2.24)$$

Ces canaux sont équivalents aux Nbits du mot de code, et sont appelés canaux binaires.

Pour la conception d'un bon code polaire, il faut savoir choisir les meilleurs canaux binaires pour envoyer les bits d'information.

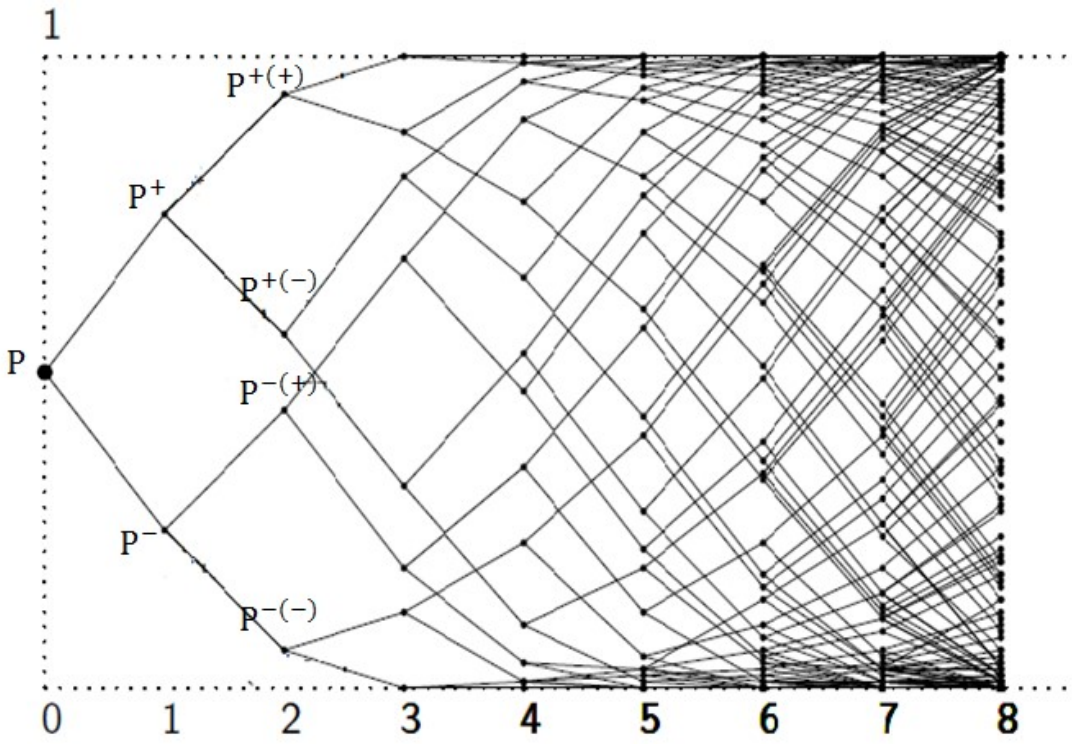


Figure 2.9. Processus de polarisation d'un canal BMS à l'aide des codes polaires.

2.2.3Conception des codes polaires [1]

Pour la conception d'un bon code polaire, il faut savoir choisir les meilleurs canaux binairespour envoyer les bits d'information$\mathrm{K}$. Pour se faire, il existe plusieurs méthodes. Certaines méthodes se basent sur le paramètre de Bhattacharyya[15]pour choisir ces canaux.Pour cela on calcule, pour chaque canal binaire, le paramètre de Bhattacharyya Z qui représente une borne supérieure sur la probabilité d'erreur du canal.Puis, on sélectionne les canaux binaires ayant les plus petites valeurs de Z (faibles probabilités d'erreur) pour envoyer les bits d'information et on fixe à " 0 " ou à " 1 " les entrées des canaux binaires restants(fortes probabilités d'erreur) pour envoyer les bits gelés (*frozen bits*).

Par la suite, on multiplie ce vecteur d'entrée par une matrice d'inversion des bits définie par Arikan [1]puis on multipliele résultat obtenu par la matrice génératrice du code polaire (voircodage polaire, § 2.3).

Le paramètre de Bhattacharyya est donné par la formule suivante :

$$Z = \sum_{y \in Y} \sqrt{p(y|0)p(y|1)} \qquad (2.25)$$

Où Y est l'alphabet de sortie, $p(y|0)$ et $p(y|1)$ représentent les probabilités de transition du canal lorsque les entrées sont 0 et 1 respectivement.

Mais le calcul de la valeur exacte de$Z$ est impossible à cause du nombre de sortie qui augmente rapidement avec chaque partition(branche)et aura une valeur énorme à la fin de la polarisation.Pour cela, il faut utiliser une méthode d'approximation du paramètre$Z$.

1- Méthode d'approximation du canal à un canal binaire à effacement [15]

L'idée est d'utiliser la méthode appliquée canal binaire à effacement quel que soit le canal étudié. Ce qui change seulement pour chaque canal est la méthode de calcul de la valeur initial du paramètre de Bhattacharyya Z_0pour le canal avant polarisation.

Pour le canal binaire à effacement(voir §2.2.1), le paramètre de Bhattacharyya a une forme récursive simple.

Soit Pun canal binaire à effacement.P^-et P^+sont les canaux obtenus par les formules de polarisations (2.22) et (2.23) à partir de P. On peut calculer le paramètre Zde ces canaux par les formules suivantes :

$$Z(P^-) = 2Z_0(P) - Z_0(P)^2 \quad (2.26)$$

$$Z(P^{+}) = Z_0(P)^2 \quad (2.27)$$

On applique ces formules Nfois pour obtenir les paramètres de Bhattacharyya des 2^ncanaux binaires.

Enfin, la polarisation de n'importe quel canal BMS se fait comme suit :

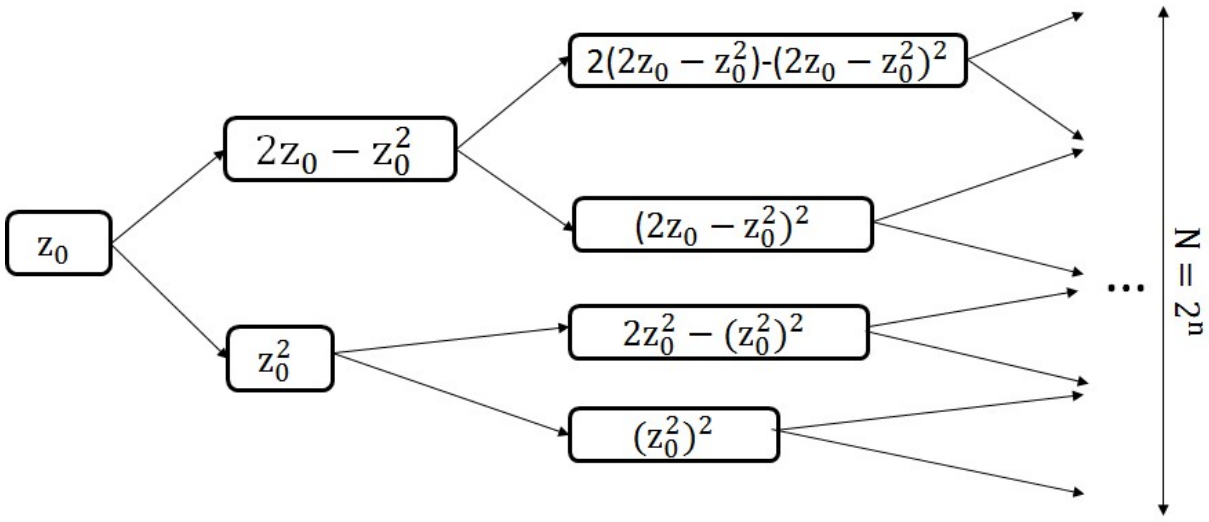


Figure 2.10. Polarisation du canal (n'importe quel type de canal) avec le paramètre de Bhattacharyya

S. Kamel a donné dans [15]la valeur initiale du paramètre de Bhattacharyya Z_0dequelques canaux :

- Canal binaire à effacement :

$$Z_0 = P_e \quad (2.28)$$

avec la probabilité d'effacement du canal BEC(voir §2.1.1)

- Canal binaire symétrique :

$$Z_0 = 2\sqrt{P_e(1-P_e)} \quad (2.29)$$

Avec la probabilité d'erreur du canal BSC (voir §2.1.1)

- Canal Gaussien à entrée binaire :

$$Z_0 = \exp\left(-R\frac{E_b}{N_0}\right) \quad (2.30)$$

R : représentant le taux de codage (ou rendement du code)

E_b : l'énergie nécessaire pour la transmission d'un bit

N_0 : densité de puissance de bruit

$\frac{E_b}{N_0}$: le rapport signal sur bruit (SNR) en transmission numérique

2- Sélection des indices de bits d'information

Il existe plusieurs algorithmes pour sélectionner les indices de K (bits d'information) à partir des indices de$N$. La plus simple est l'utilisation de la forme récursive du paramètre de Bhattacharyya : $z \to \{2z - z^2, z^2\}$ (d'après la figure2.10).

Pour se faire, reprenons le schéma de lafigure2.10:

- On arrête d la polarisation lorsque l'arbre atteint N feuilles indexées du haut de $0,...,N-1$

- On recherche les feuilles ayantles plus petites valeurs (faible probabilité d'erreur) et notons leurs indices I

- Les autres feuilles ayant les plus grandes valeurs (forte probabilité d'erreur)sont gelées et fixés à "0" ou à "1" et notons leurs indices I_c .

3- Non universalité des codes polaires

En théorie de codage, la plus part des codes sont « universels » dans le sens où leur définition est indépendante du SNR du canal [10].

Arikan a défini un ensemble I_c de codes polaires tel que le BLER de ces codes soit minimum en utilisant l'algorithme de décodage SC. Puisque le BLER est une fonction duSNR, ce n'est donc pas surprenant que les codes polaires changent avec la valeur fixée du SNR ou (*design-SNR*).

Cette caractéristique de non-universalité des codes polaires est très importante.

Une modification du SNR opérationnel est possible en pratique, mais un changement de code selon le SNR n'est pas souhaitable. Par conséquent, il est souhaitable deconstruire un code polaire dans un *design-SNR* et l'utiliser pour une gamme de SNR possibles [10].

Le choix du *design-SNR* est essentiel pour la performance à tous les SNR d'intérêt. Malheureusement, nous n'avons pas trouvé d'étude pour identifier le meilleur design-SNR pour toute construction de code polaire.

Il a eu quelques tentatives récentes de conception de codes polaires universels mais le coût fut beaucoup plus élevé du point de vue complexité au décodeur et / ou encodeur. Selon [10] il n'y a pas eu beaucoup d'études dans ce sens sur les codespolaires. Beaucoup de travaux de recherche considèrent souvent choix heuristiquede la conception du design-SNR.

La meilleure performance qu'on peut atteindre est approximativement la même pour tout algorithme de construction pour au moins $N \leq 64K$ [10].

2.3 Codage polaire

Un code polaire, notéCP(N, K), est un code en bloc linéaire (Arıkan, 2008)[1] de taille $N = 2^n$,n étant un entier naturel, contenant Kbits d'information.Sa matrice génératrice est une sous-matrice de la nième puissance de Kronecker[1] de $k = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ notée :

$$F = k^n = \begin{bmatrix} k^{n-1} & 0 \\ k^{n-1} & k^{n-1} \end{bmatrix} \tag{2.31}$$

Composée de K lignes.

Ces Klignes sont choisies en supposant un décodage par annulation successiveSCqui permet de polariser la probabilité d'erreur des bits du message.Voici quelques exemples :

- Pour $N = 2 \rightarrow n = 1 \rightarrow F = k^1 = \begin{bmatrix} k^0 & 0 \\ k^0 & k^0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

- Pour $N = 4 \rightarrow n = 2 \rightarrow F = k^2 = \begin{bmatrix} k^1 & 0 \\ k^1 & k^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

- Pour $N = 8 \rightarrow n = 3 \rightarrow F = k^3 = \begin{bmatrix} k^2 & 0 \\ k^2 & k^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$.

- …etc.

Le processus de codage consiste à construire un vecteur Ude tailleN, $U = [u_0, u_1, \dots, u_{N-1}]^T$contenant les K bits d'information et les N-K bits gelés fixé à 0 ou 1 (voir §2.2.3).

Ce vecteur Uest construit de telle manière que les bits d'information soient localisés sur les indices les plus fiables correspondant aux K lignesde k^nsélectionnées précédemment. Le mot de code correspondant$X = [X_0, X_1, \dots, X_{N-1}]^T$, peut ensuite être calculé simplement de la façon suivante:

$$X = U \times F = U \times k^n \qquad (2.32)$$

2.3.1 Représentations des codes polaires

La complexité de codage et décodage d'un code dépend du nombre d'opération utilisé par la méthode de construction de ce code. On cherche toujours à avoir une complexité acceptable et pour se faire il faut minimiser le plus possible le nombre d'opération d'où les différentes représentions des codes polaires.

Un code polaire peut être représenté sous forme matricielle à partir de la matrice F et sous forme graphique (*factor graph)*. Ce dernier pouvant être utilisé pour le codage et pour le décodage.

1- Exemple de représentation matricielle

Soit un code polaire noté $CP(8,4)$ où :

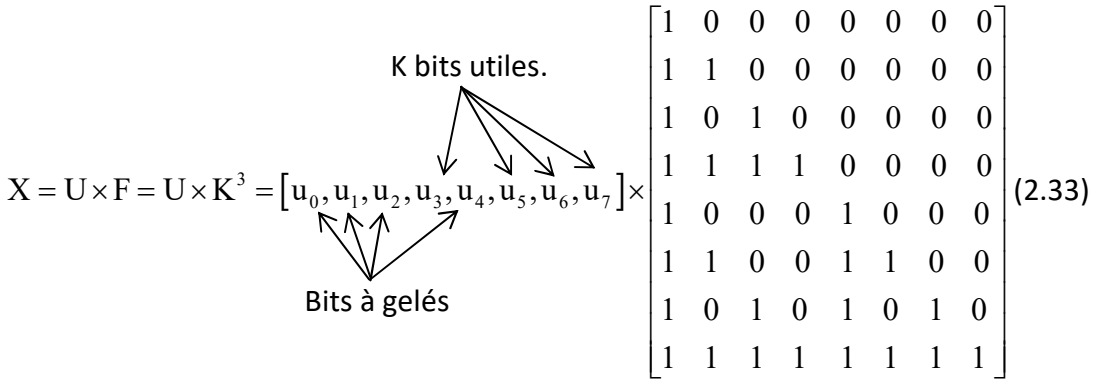
- la taille de la donnée est $\boldsymbol{K = 4}$
- la longueur du code est $\boldsymbol{N = 8}$
- le vecteur U, en entrée, est de taille N et noté $U = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7]$
- le rendement du code est $R = 0.5$.

Supposons qu'après la polarisation du canal, les indices I des K bits sont le 3, 5, 6 et 7 et celles des bits à gelés $\mathrm{I_c}$ sont 0, 1, 2, 4.

Notons que le nombre de bits gelés dépend du rendement $\boldsymbol{R} = \frac{\boldsymbol{K}}{\boldsymbol{N}}$du code.

Dans cet exemple le rendement est égale à 0.5 cela implique que la moitié du vecteur U serait gelé et l'autre moitié serait composé des K bits d'information.

Le mot de code sera donc :



$$\mathrm{X} = \mathrm{U} \times \mathrm{F} = \mathrm{U} \times \mathrm{K}^3 = \left[\mathrm{u}_0, \mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \mathrm{u}_4, \mathrm{u}_5, \mathrm{u}_6, \mathrm{u}_7\right] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.33)$$

Le résultat de cette multiplication donne :

$$\mathrm{X} = \begin{bmatrix} \mathrm{u}_0 + \mathrm{u}_1 + \mathrm{u}_2 + \mathrm{u}_3 + \mathrm{u}_4 + \mathrm{u}_5 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_1 + \mathrm{u}_3 + \mathrm{u}_5 + \mathrm{u}_7 \\ \mathrm{u}_2 + \mathrm{u}_3 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_3 + \mathrm{u}_7 \\ \mathrm{u}_4 + \mathrm{u}_5 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_5 + \mathrm{u}_7 \\ \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_7 \end{bmatrix}^{\mathrm{T}} \quad (2.34)$$

A l'entrée du canal, le mot de code sera :

$$X = \begin{bmatrix} 0+0+0+u_3+0+u_5+u_6+u_7 \\ 0+u_3+u_5+u_7 \\ 0+u_3+u_6+u_7 \\ u_3+u_7 \\ 0+u_5+u_6+u_7 \\ u_5+u_7 \\ u_6+u_7 \\ u_7 \end{bmatrix}^T \quad (2.35)$$

2- Exemple de représentation sous forme de *factor graph*

Dans cet exemple, deux matrices génératrices ainsi que leur représentation graphique sont présentées.

- $CP(2,2)$ et $U = [u_0, u_1] \rightarrow X = [u_0, u_1] \times \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^T \Rightarrow$

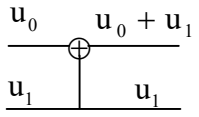


Figure 2.11.Matrice génératrice et graphe du codeur polaire CP(2,2).

- $CP(8,4)$ et $U = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7]$

$$\Rightarrow X = \begin{bmatrix} 0+0+0+u_3+0+u_5+u_6+u_7 \\ 0+u_3+u_5+u_7 \\ 0+u_3+u_6+u_7 \\ u_3+u_7 \\ 0+u_5+u_6+u_7 \\ u_5+u_7 \\ u_6+u_7 \\ u_7 \end{bmatrix}^T \quad (2.36)$$

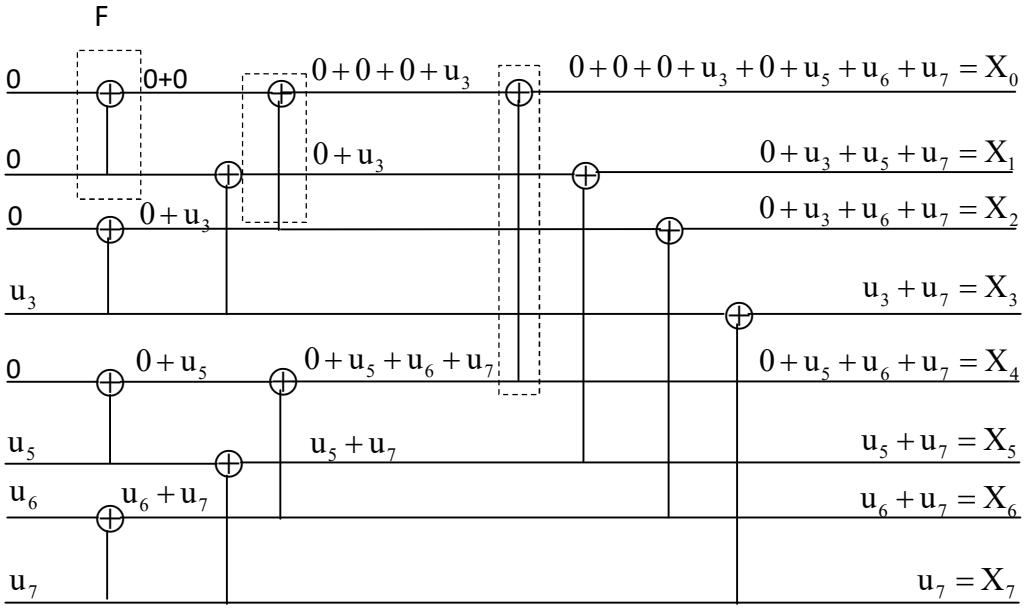


Figure 2.12.Matrice génératrice et graphe de codeur CP(8,4)

Les nœuds de parités ($\oplus$) se comportent comme des XOR et les nœuds de variables($\perp$) font simplement passer la valeur binaire à l'étage suivant.

Plus généralement, le *factor graph* d'un code polaire, présenté dans (Arikan 2009) [2], de taille $N = 2^n$ est composé de :

n étapes de $\frac{N}{2}$ nœuds de parités de degrés 3 et, $\frac{N}{2}$ nœuds de variables de degrés 3.

Le degré d'un nœud représente son nombre de connexion avec d'autre nœud. Le *factor graph* peut être utilisé pour le codage et le décodage. Pour le codage, le vecteurd'entréU, sur le côté gauche, est propagé dans le graphe dans le but de générer le mot de code$X$, sur la droite.

2.4 Décodage des codes polaires

Figure 2.13.Schéma synoptiquedu processus de décodage polaire

$u_i = [u_0, u_1, ..., u_{N-1}]$: bits d'information + bits gelés ;

$X = [X_0, X_1, ..., X_{N-1}]$: mot de code

$Y = [Y_0, Y_1, ..., Y_{N-1}]$: version bruité du mot de code reçu

$\hat{u}_i = [\hat{u}_0, \hat{u}_1, ..., \hat{u}_{N-1}]$: bits estimés

N : la taille du mot de code.

Une fois le message transmis au travers du canal de communication, la version bruitée, $Y = [Y_0, Y_1, ..., Y_{N-1}]$ du mot de code $X = [X_0, X_1, ..., X_{N-1}]$ est reçue. Le but du décodage est d'estimer le vecteur U à partir de la version bruitée du mot de code Y.

Plusieurs algorithmes de décodage ont été proposés dans Arikan [1] pour les codes polaires tel que:

- *Successive Cancellation* (SC)
- *Successive Cancellation with Lists* (SCL)
- *Belief Propagation* (BP).
- *Soft-Cancellation* (SCAN)
- …etc.

2.4.1 Décodage SC

La complexité du décodeur SC est de l'ordre de $N \log_2(N)$, N étant la longueur du code.

Ce décodeur se base sur les rapports de vraisemblance LR pour estimer les valeurs des bits d'informations. Il décode les bits u_i par ordre c'est-à-dire du bit d'indice 0 jusqu'au bit d'indice $N - 1$.

Pour pouvoir décoder le bit numéro i, le décodeur est sensé connaitre :

- les LR des sorties des N canaux binaires
- les estimations des i premiers bits

- les positions des bits d'information K et,
- les valeurs des bits gelés$\mathrm{N}-\mathrm{K}$.

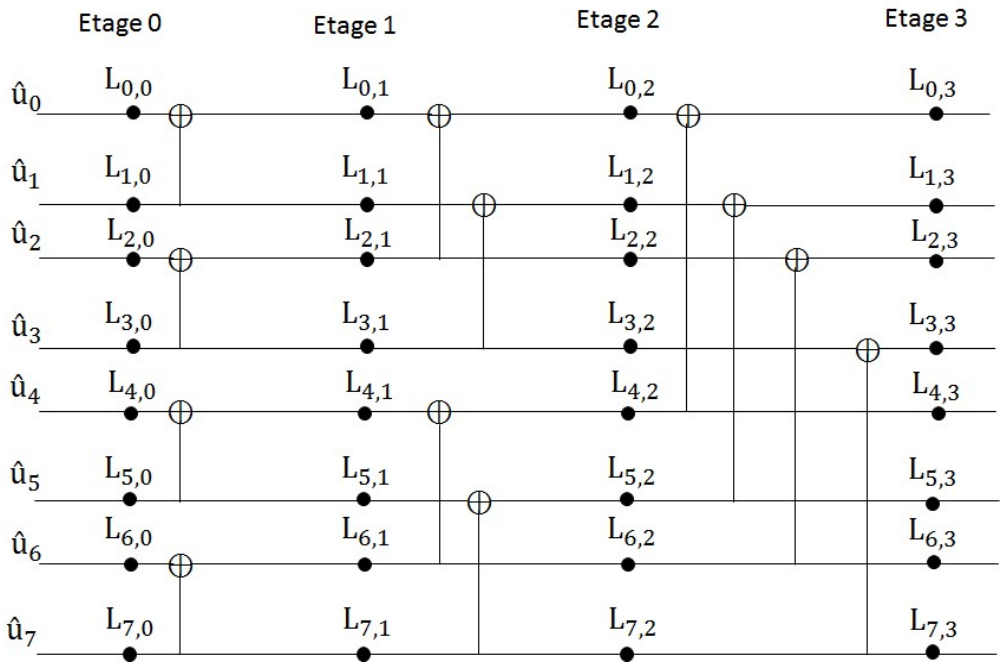


Figure 2.14.Schéma synoptiquedu processus du décodage SC avec

$\boldsymbol{L_{i,j}}$ la valeur de LR du $\boldsymbol{i^{ème}}$ bit au $\boldsymbol{j^{ème}}$étage

Si le bit iest un bit gelé, le décodeur lui donne directement sa valeur qu'il connait déjà. Sinon, il examine la valeur duLRcorrespondant à ce bit, si elle est supérieure à 1 ($\mathrm{L_{i,j}}$ >1) alors le bit est 0, sinon le bit est 1.

i : le nombre de sortie du décodeur avec $0 \leq \mathrm{i} \leq \mathrm{N}-1$

j : le nombre d'étage du décodeur avec $0 \leq \mathrm{j} \leq \mathrm{n}$

Pour estimer la valeur du bit$\mathrm{u_i}$, notée$\hat{\mathrm{u}}_\mathrm{i}$, le décodeur se base sur l'observation$\mathrm{Y} = [\mathrm{Y_0 Y_1 \dots Y_{N-1}}]$ et sur les valeurs des bits estimésprécédemment $\hat{\mathrm{u}}_0^{\mathrm{N}-1} = [\hat{\mathrm{u}}_0 \hat{\mathrm{u}}_1 \dots \hat{\mathrm{u}}_{\mathrm{i}-1}]$et ce en calculant les valeurs des LR(s) suivants :

$$L_{i,0} = \frac{\Pr(Y_0^{N-1}, \hat{u}_0^{N-1} | u_i = 0)}{\Pr(Y_0^{N-1}, \hat{u}_0^{N-1} | u_i = 1)} \quad (2.37)$$

Le *factor graph* peut être utilisé pour le codage et le décodage.

Afin de bien comprendre le décodage par annulation successives, prenons l'exemple du codeur de base où N=2 [19] :

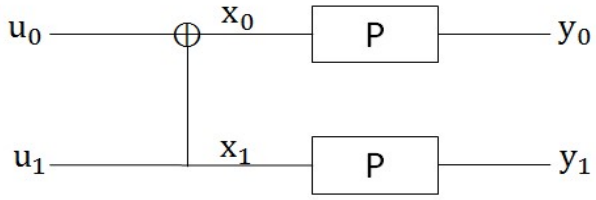


Figure 2.15.Encodage et transmission dans une chaine d'ordre 2

L'encodeur mappe les u_i (incluant les bits gelés) auxx_i . Ces derniers sont transmis à travers le canal P et, les y_i sont reçusen sortie.

En supposant avoir une connaissance complète deP, les sorties y_i peuvent être des bits (*hard*) ou des valeurs réelles (*soft*).

Comme indiqué au § 2.2.2 :$x_0 = u_0 \oplus u_1$, $x_1 = u_1$ et $u_0 = x_0 \oplus x_1$.

Pour chaque valeur de yreçue nous connaissons les probabilités de transition $P(y|x = 0)$ et $P(y|x = 1)$ (voir probabilités de transition du canal gaussien dans §2.2.1).

En posant :

$$a = P(y_0|x_0 = 0), b = P(y_0|x_0 = 1),\ c = P(y_1|x_1 = 0) \text{ et } d = P(y_1|x_1 = 1) \quad (2.38)$$

Sachant que les transitions $x_0 \to y_0$ et $x_1 \to y_1$ sont indépendantes et en factorisant les probabilités par paire,on obtient :

$$P(y_0, y_1|u_0 = 0) = \frac{ac+bd}{2} \quad \text{et } P(y_0, y_1|u_0 = 1) = \frac{bc+ad}{2} \qquad (2.39)$$

Le rapport de ceux probabilités est : $\frac{\frac{ac}{bd}+1}{\frac{c}{d}+\frac{a}{b}}$. C'est le rapport de vraisemblance de u_0 connaissant les sorties y_0 et y_1 qu'on note $LR(u_0)$.

En posant $p = \frac{a}{b}$ et $q = \frac{c}{d}$, on définit la fonction

$$LR(u_0) = f(p, q) = \frac{pq+1}{p+q} \qquad (2.40)$$

Si $f(p, q) \geq 1$ alors l'estimé de u_0 vaut 0 ($\hat{u}_0 = 0$)

Si $f(p, q) < 1$ alors $\hat{u}_0 = 1$

u_0 étant décodé, l'étape qui suit est celle du décodage de u_1 et qui consiste à calculer $LR(u_1)$.

En se basant sur la règle du décodage par annulations successives (SCD) alors on suppose que u_0 a été correctement décodé.

$$\text{Si } u_0 = 0 \text{ alors } LR(u_1) = \frac{P(y_0,y_1|u_0=0,u_1=0)}{P(y_0,y_1|u_0=0,u_1=1)} = \frac{P(y_0,y_1|x_0=0,x_1=0)}{P(y_0,y_1|x_0=1,x_1=1)} = \left(\frac{a}{b}\right)\left(\frac{c}{d}\right) \qquad (2.41)$$

$$\text{Si } u_0 = 1 \text{ alors } LR(u_1) = \frac{P(y_0,y_1|u_0=1,u_1=0)}{P(y_0,y_1|u_0=1,u_1=1)} = \frac{P(y_0,y_1|x_0=1,x_1=0)}{P(y_0,y_1|x_0=0,x_1=1)} = \left(\frac{a}{b}\right)^{-1}\left(\frac{c}{d}\right) \qquad (2.42)$$

Comme précédemment, on définit la fonction

$$LR(u_1) = g(p, q, u_0) = p^{1-2u_0}q \qquad (2.43)$$

$\hat{u}_1 = 1$ si $g(p, q, u_0) \geq 1$

$\hat{u}_1 = 0$ si $g(p, q, u_0) < 1$

Dans la partie le codage d'une structure de codage/décodage polaire d'ordre N, les nœuds de parités ($\oplus$) se comportent comme des XOR et les nœuds de variables($\perp$) font simplement passer la valeur binaire à l'étage suivant.

Dans la partie décodage, ces nœuds ne symbolisent plus un simple XOR ni une simple copie (voir figure2.15).

Le décodeur estime successivement la valeur des bits $\hat{u}_i$ est se basant sur les valeurs des LR ($L_{i,j}$) et les valeurs des sommes partielles $S_{i,j}$ qui n'est autre que le recodage des bits estimés.

Ces valeurs sont calculées par les formules suivantes [4]:

$$L_{i,j} = \begin{cases} F(L_{i,j}, L_{i+2^j,j+1}) & \text{Si } B_{i,j} = 0 \\ G(L_{i-2^j,j+1}, L_{i,j+1}, S_{i-2^j,j}) & \text{Si } B_{i,j} = 1 \end{cases} \quad (2.44)$$

$$S_{i,j} = \begin{cases} H_l(S_{i,j-1}, S_{i+2^{j-1},j-1}) & \text{Si } B_{i,j-1} = 0 \\ H_u(S_{i,j-1}) & \text{Si } B_{i,j-1} = 1 \end{cases} \quad (2.45)$$

Avec :

$$\begin{cases} F(a,b) = \dfrac{1+ab}{a+b} \\ G(a,b,S) = b \times a^{1-2S} \\ H_u(S) = S \\ H_l(S,S') = S \oplus S' \\ B_{i,j} = \dfrac{i}{2^j} \bmod 2 \end{cases} \quad \begin{array}{l} \\ \\ \\ \\ 0 \le i < N\text{-}1 \text{et} \\ 0 \le j < n \text{ (le numéro des étages)} \end{array} \quad (2.46)$$

En effet, les fonctions F et G peuvent être représentées telles que :

$$F(L_a, L_b) = \frac{1 + L_a L_b}{L_a + L_b} \quad (2.47)$$

$$G(L_a, L_b, S) = L_b \times L_a^{1-2S} \quad (2.48)$$

Les fonctions de décodage F et G sont complexe du côté implémentation. D'où le besoin d'un changement de domaine suivie par une simplification.

Pour se faire, comme explicité dans l'article de C. Leroux (2014)[4], il faut passer dans le domaine logarithmique. Les valeurs ainsi manipulées sont appelées *Log-Likelihood Ratio* (rapport de vraisemblance logarithmique LLR). Les fonctions F et G deviennent alors :

$$f(L_a, L_b) = 2\tanh^{-1}(\tanh(\frac{L_a}{2}).\tanh(\frac{L_b}{2})) \quad (2.49)$$

$$f(L_a, L_b) \approx \text{sign}(L_a).\text{sign}(L_b) \times \min(|L_a|, |L_b|) \quad (2.50)$$

$$g(S, L_a, L_b) = (-1)^S \times L_b + L_a \quad (2.51)$$

AvecS le recodage des bits estimé (voir figure2.15).

Dans ce cas, si le bit $\hat{u}_i$ à estimer n'est pas un bit gelé, la décision à prendre par le décodeur SC est la suivante :

- $\hat{u}_i = 0$ si LLR > 0,
- $\hat{u}_i = 1$ sinon.

Voici un exemple de décodeur SC CP(8,4) présenté dans Leroux [4] :

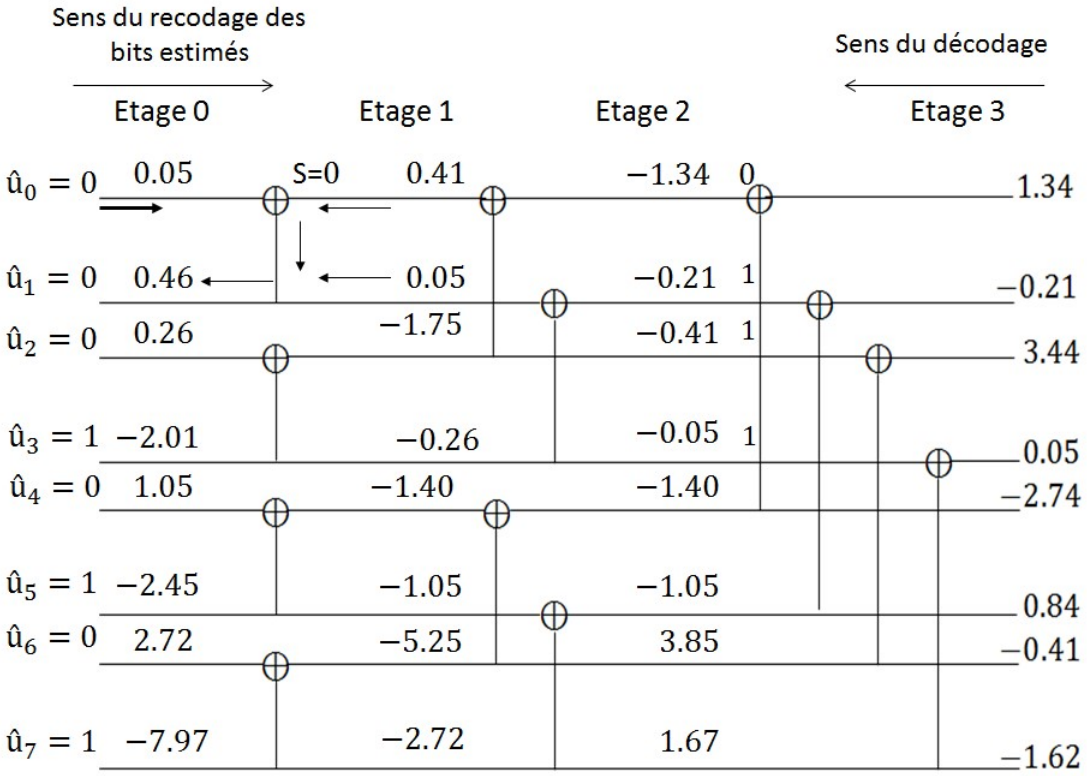


Figure 2.16.Décodeur SC du CP(8,4)

Remarque:

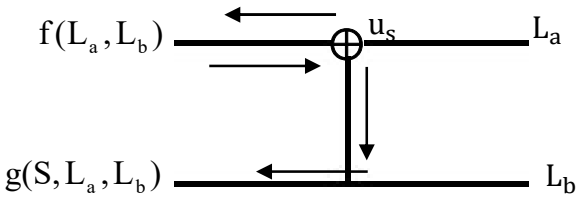


Figure 2.17.Définition des fonctions de décodage f et g

Les valeurs du LLR à l'étage 0 sont les informations du canal. Dans cet exemple, elles ont été données. [4]

Des études faites dans Camille Leroux [4] ont montré qu'il existe d'autres algorithmes de décodage, pour les codes polaires, plus performant que le décodage SC.

2.5 Performances des codes polaires

2.5.1 Introduction

Dans ce paragraphe, nous allons mettre en évidence les performances des codes polaires et ce en faisant des comparaisons du point de vu BER en fonction du SNR avec les LDPC et les turbo codes.

2.5.2 Estimation du BER dans un canal AWGN pour différentes longueur du code [19]

1- Cas d'une modulation BPSK

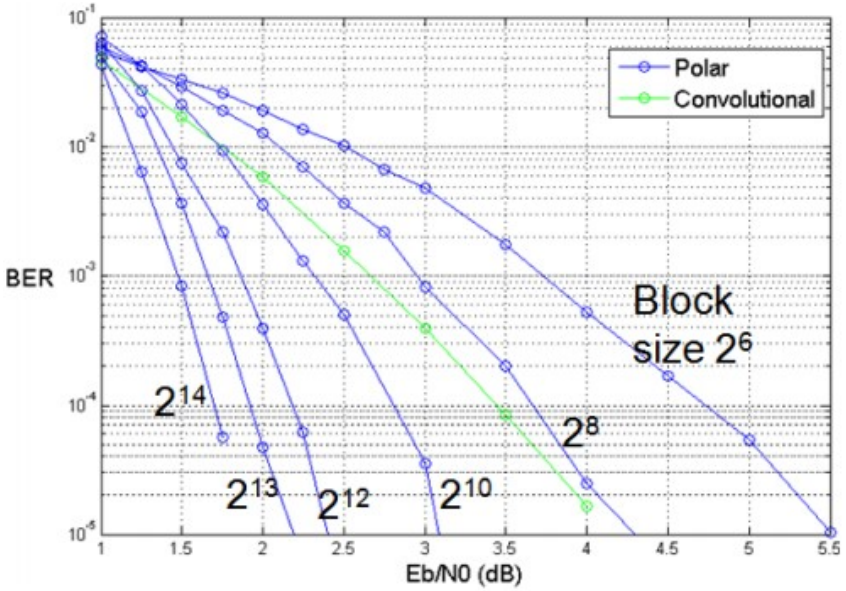


Figure 2.18. Estimation du BER pour code polaire de taux de codage de ½ en BPSK dans des canaux gaussiens [19]

Interprétation : une ligne bleue de la figure 3.1 qui atteint le bas du graphique indique qu'une estimation de BER inférieure à $\mathbf{10^{-5}}$ a été calculée.Cependant, dans les lignes bleues de la figure, chaque point sur la ligne représente un code construit pour une valeur particulière de$\frac{\mathbf{E_b}}{\mathbf{N_0}}$. C'est une pratique courante dans la littérature des codes polaires parce que ces derniers sont facilement optimisés pour différents niveaux de$\frac{\mathbf{E_b}}{\mathbf{N_0}}$

.

2- Cas d'une modulation QPSK

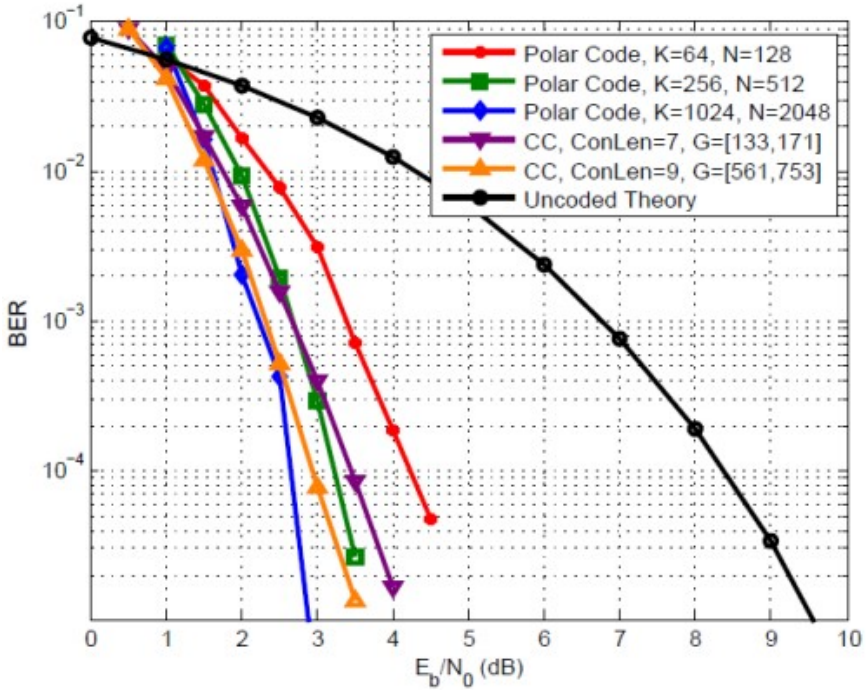


Figure 2.19. Estimation du BER pour code polaire de taux de codage de ½ en QPSK dans des canaux gaussiens [19]

Interprétation : sur la figure 2.2, la courbe en rouge représente un code polaire avec K=64 et N=128. Les courbes représentées en vert et en bleu représentent aussi un code polaire mais respectivement avec quatre fois la taille code en rouge et quatre fois la taille de celui en bleu.

La courbe violète représente un code convolutif avec ConLen=7 (6 registres +1) et $\mathrm{G} = [133,171]$ d'où un taux de codage égal à $\frac{3}{4}$, et la courbe orange un code convolutif avec ConLen=9 (8 registre +1) et $\mathrm{G} = [561,753]$ d'où un taux de codage égal à $\frac{1}{2}$.

Nous remarquons que les codes polaires atteignent une meilleur performance avec une grande valeur de N et les codes convolutifs ont une meilleur performance avec un faible taux de codage.

2.5.3 Performance de l'algorithme SC [9]

Les codes polaires atteignent théoriquement la capacité d'un canal pour une taille de code infinie. C'est le seul code correcteur d'erreurs, à ce jour, pour lequel il est possible de démontrer mathématiquement cette propriété [2]. Il est donc naturel de se demander s'ils peuvent remplacer les codes de la littérature. Pour cela, il est nécessaire de comparer les performances de différents codes, ainsi que leur complexité calculatoire, pour des contraintes équivalentes comme la taille du code ou le rendement.

1- Comparaison avec le turbo code

Dans les courbes qui suivent (figure 2.20 et 2.21) sont regroupées les comparaisons des performances sur un canal à AWGN d'un codeur avec turbo code et de codes polaires.

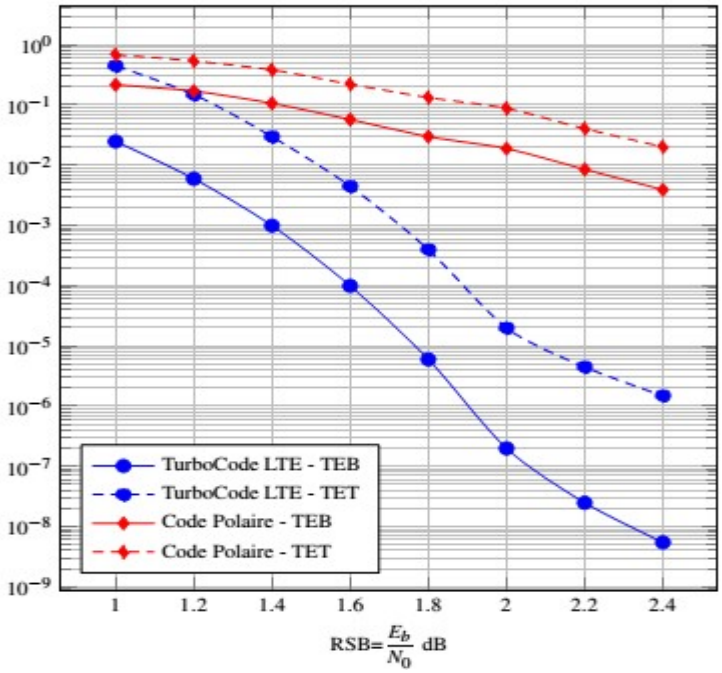


Figure 2.20. Turbo code (1024; 512), m = 3, 6 itérations issu du mémoire de thèse de Sanchez G. (2013) - Code Polaire non systématique, CP (1024; 512), décodé avec un algorithme SC [9].

Interprétation : Au niveau des performances pures, les codes polaires sont en deçà des Turbocodes avec un décodage SC original (non systématique). Une perte d'environ $\mathbf{0.8\ dB}$pour un$\mathbf{TEB} = \mathbf{2.10^{-2}}$. En revanche, la complexité de décodage est bien plus

faible pour un code polaire, en utilisant un algorithme SC, systématique ou non puisque le décodage n'est pas itératif et que l'algorithme SC est très simple à mettre en œuvre car il est récursif et régulier.

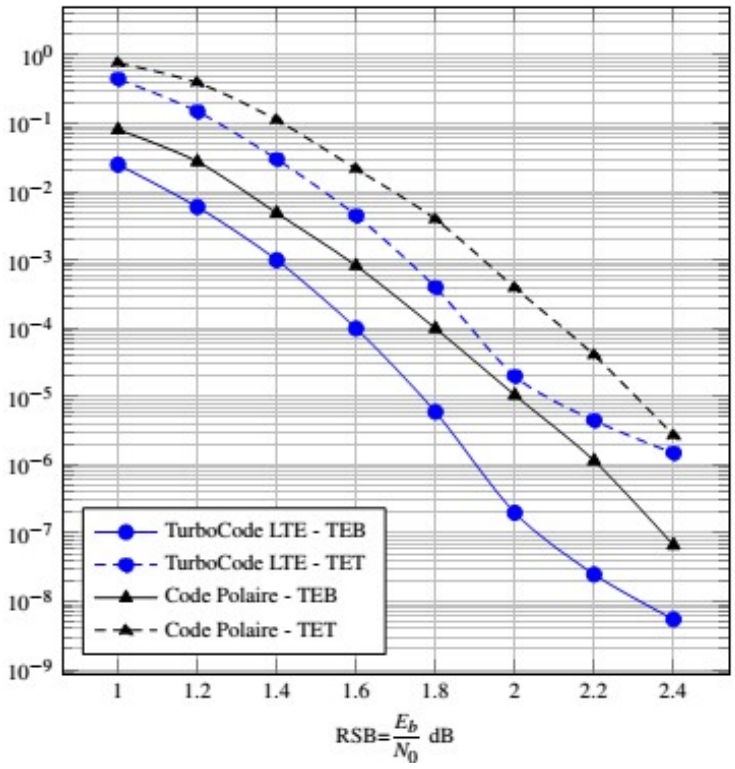


Figure 2.21.Turbocode (1024; 512), m = 3, 6 itérations issu du mémoire de thèse de Sanchez G. (2013) – Code Polaire CP (16384; 8192) systématique, décodé avec un algorithme SC [9].

Interprétation : les performances d'un code polaire systématique ayant une longueur 16 fois plus importante, CP(16384; 8192), sont alors comparées aux performances du turbo code précédent de la figure2.21. L'encodage et le décodage systématiquene modifient pas la complexité de l'algorithme mais permettent d'améliorer les performances de décodage (§ 2.4). C'est la raison pour laquelle un code polaire systématique est retenu. Il apparaît, que dans cette configuration de complexité de décodage équivalente, les performances du code polaire ne sont plus qu'à environ 0:2 dB des performances du turbo code pour un$\mathrm{TEB} = 10^{-4}$.

2- **Comparaison avec les codes LDPC**

Une autre comparaison a été menée cette fois entre un code LDPC et un Code Polaire. Ces comparaisons sont données dans les courbes (figure 2.22 et 2.23).

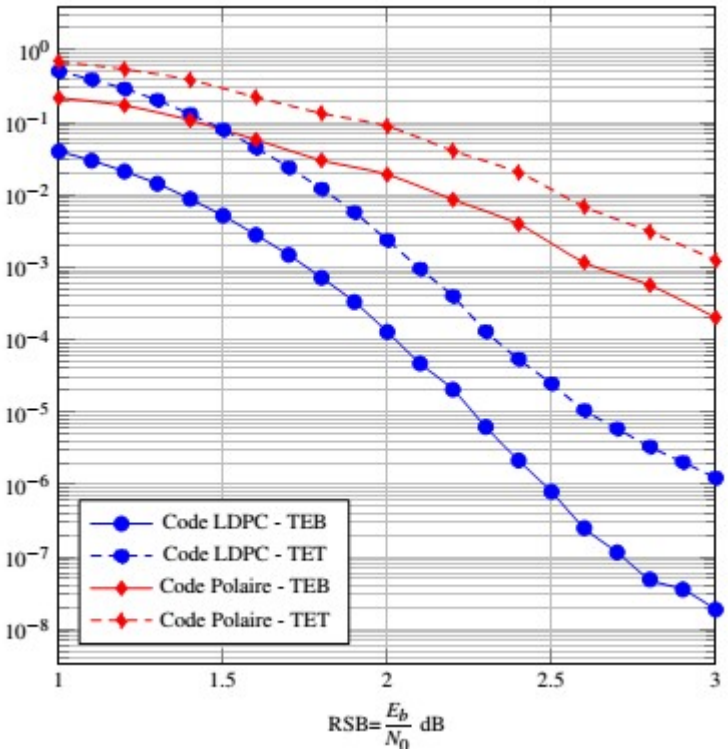


Figure 2.22. Code LDPC (1056; 528) (50 itérations) – code polaire non systématique CP(1024; 512) [9] .

Au niveau des performances pures, les codes polaires sont en deçà des codes LDPC pour undécodage SC originalles Codes Polaires sont en deçà des codes LDPC pour undécodage SC original. Une perte d'environ $\mathbf{0.9\ dB}$ a été observée pour un$\mathbf{TEB} = \mathbf{10^{-3}}$.

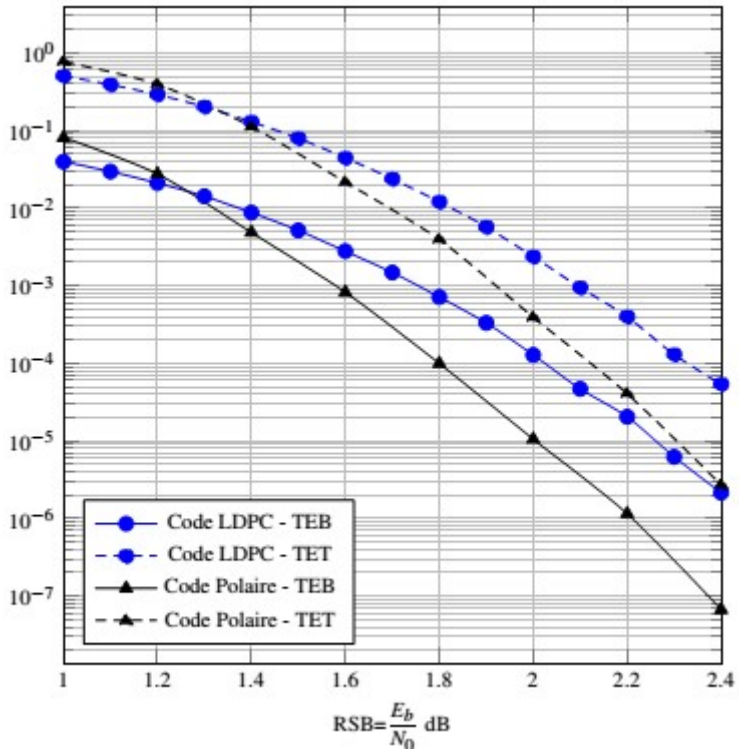


Figure 2.23.Code LDPC (1056; 528) (50 itérations-code polaire systématique CP(16384; 8192) [9].

Les performances d'un code polaire systématique ayant une taille 16 fois plus large, CP (16384,8192), sont alors comparées à celles du code LDPC précédent Il apparaît, que dans cette configuration de complexité calculatoire équivalente, les performances du code polaire sont supérieures d'environ $\mathbf{0.2\ dB}$ pour un $\mathbf{TEB = 10^{-4}}$par rapport au code LDPC.

Cette famille de code correcteur d'erreurs est donc une alternative aux codes LDPC.

2.6 Conclusion

Dans ce chapitre nous avons passé en revue une nouvelle famille de codes correcteurs d'erreurs : les codes polaires. Ils utilisent un nouveau concept appelé la polarisation du canal qui est utilisé pour leur construction.

La construction, le codage et le décodage des codes polaires ont été présentés en détail. L'algorithme de décodage utilisé, appelé SC, a été détaillé car beaucoup de travaux en découlent.

Les codes polaires sont les premiers codes atteignant la capacité des canaux BMS lorsque la taille du mot de code augmente et, avec un choix judicieux du design-SNR dans le cas particulier du canal B-AWGN.

Nous avons également fait une comparaison de performance entre les codes polaires et le turbo code, et entre les codes polaires et les codes LPDC.

Pour une très grande valeur de$N$, les codes polaires sont plus performant plus que les turbo code et les codes LDPC pour un décodage SC original.

Dans le chapitre qui suit, nous allons effectuer des simulations sous le logiciel matlab en estimant le taux d'erreurs binaire (BER) en fonction du SNR afin de trouver le meilleur design-SNR (cf chapitre 3) pour lequel les codes polaires atteignent leur meilleure performance.

Chapitre 2 Les codes polaires

2.1 Introduction

Les codes polaires sont des codes en blocs linéaires. Ils ont fait l'objet d'une recherche active ces derniers temps, principalement en raison du fait qu'ils sont les premiers codes atteignant la capacité des canaux BMS [9], avec une construction explicite et une très faible complexité de codage et de décodage en utilisant un algorithme particulier appelé algorithme par annulation successive SC.

Il a été prouvé mathématiquement, en 2008, par Erdal Arıkan [1]que les codes polaires étaient capables de corriger toutes les erreurs de transmission sous certaines conditions.

En effet, ayant été inventé par Arikan[1], les codes polaires utilisent un concept nouveau appelé polarisation de canal[17]. Peu de temps après, le concept de polarisation des canaux ainsi que les codes polaires ont été étendus à un certain nombre d'applications et de généralisations et des améliorations ont été introduites de sorte que les performances du canal ont maintenant presque fermé l'écart à la limite de Shannon qui, définit la barre pour le débit maximal pour une bande passante donnée et un niveau de bruit donné [11].

Le code devrait aider à corriger les erreurs de transmission ainsi qu'à augmenter les débits et l'efficacité spectrale des réseaux cellulaires.

Les codes polaires sont constitués de trois (3) blocs essentiels qui sont :

- la construction du code
- le codage polaire et
- le décodage polaire.

Sur le plan de leurs applications dans l'industrie des télécommunications, le groupe Huawei a annoncé, en octobre 2016, qu'elle avait réalisé un débit de 27Gbps dans des tests d'essais sur champs 5G en utilisant les codes polaires pour le codage des canaux.

Lors de la rencontre 87 RAN1organisée le 17 novembre 2016 par le 3GPP, organisme international de développement des standards pour les télécommunications mobiles, les codes polairesont été choisi comme schéma de codage du canal de contrôle pour l'application de la 5G dans le scénario eMBB (*Enhanced Mobile Broadband* ou amélioration du haut-débit mobile).

2.2 Construction du code

2.2.1 Canal de transmission[9]

1- Définition

Le canalde transmission, du point de vue théorie de l'information, est situé entre la sortie du codeur de canal et l'entrée du décodeur de canal, alors que du point de vue radio fréquence il serait situé entre les antennes d'émission et de réception.

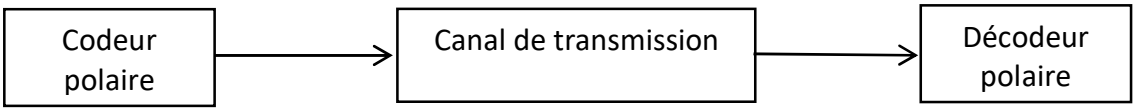


Figure 2.1. Canal de transmission du point de vue théorie de l'information

Un milieu de transmission est un support transportant de l'information comme par exemple l'air, une paire de fils torsadés, un câble coaxial, une fibre optique…etc.

Par extension,les supports physiques tels que les disques durs ou les CD, sont également des milieux de transmission.

De manière générale, un canal de transmission est défini mathématiquement par deux ensembles et une probabilité de transition d'un ensemble vers l'autre :

- un ensemble X, contenant toutes les entrées possibles
- un ensemble Y, contenant toutes les sorties possibles
- une probabilité de transition notée $\mathrm{P}(\mathrm{Y} \mid \mathrm{X})$.

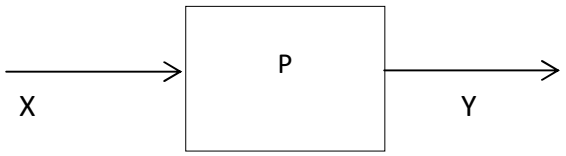


***Figure 2.2*.Schéma bloc d'un canal de transmission**

Un canal de transmission subit des perturbations de différents types selon la nature du canal. Ces perturbations peuvent être de type électromagnétique, de type thermique ou des interférences liées à la présence de plusieurs utilisateurs sur le canal.

Le canal prend en entrée des données binaires et produit en sortie une estimation des bits originaux. Cette estimation peut être dure(*hard*) ou souple (*soft*). Une sortie dure implique une prise de décision 0 ou 1 quant à la valeur des bits de sortie. Une sortie souple implique que la valeur du bit en sortie soit représentée par une probabilité de sa valeur.

Des valeurs souples permettent d'obtenir en général de meilleurs résultats lors du décodage du message mais nécessitent une complexité calculatoire supérieure à celle nécessaire pour des valeurs dures.

Il existe quatre propriétés principales des canaux :

- discrétisation : un canal discret est un canal de communication qui ne transmet que des nombres entiers ou, plus généralement, un nombre fini de symboles.

Le plus souvent, il s'agit d'un canal binaire qui ne transmet donc quedes 0 oudes 1 comme indiqué en figure 2.3 (les canaux BSCet BEC) seront examinés en détail par la suite.

Figure 2.3.a)canal binaire symétrique (BSC) avec $0 \leq P_e \leq 1/2$,b)canal binaire à effacement (BEC) avec εl'effacement et P_e la probabilité de l'effacement.

- continuité : un canal continu possède un alphabet d'entrée et de sortie qui est à valeur dans l'ensemble de réels.

- stationnarité : un canalstationnairepossède une probabilité de transition qui ne dépend pas du temps.

- effet mémoire : un canal sans effet mémoire implique qu'un symbole en sortie ne dépend que du symbole d'entrée.

2- Capacité d'un canal [6]

La capacité d'un canal est définie par l'information mutuelle maximale entre une variable aléatoire X à valeur sur l'alphabet d'entrée du canal et sa sortie correspondante Y par le canal.

$$\mathrm{C} \stackrel{\text{def}}{=} \sup \mathrm{I}(\mathrm{X};\mathrm{Y}) \qquad (2.1)$$

On remarquera que $\mathrm{I}(\mathrm{X};\mathrm{Y})$ peut s'écrire en fonction des seules lois de transition et d'émission :

$$I(X;Y) = \sum_{x,y} P(y|x)P(x)\log_2 \frac{P(y|x)}{P(y)} \qquad (2.2)$$

$$P(y) = \sum_{x} P(y|x)P(x) \qquad (2.3)$$

Où$P(y|x)$désigneprobabilité d'obtenir Y sachant X a été envoyée.

Rappelle : la limite (ou la capacité) de Shannon est de -1.6 dB

3- Les principaux canaux BMS

- **Canal binaire à effacement BEC**

Le canal à effacement BEC est un canal discret, stationnaire et sans effet mémoire. Sa sortie est constituée de l'ensemble {0, ε,1}.

Le symbole ε représente un effacement (ou bit effacé).L'effacement correspond à une perte totale d'information permettant de prendre une décision quant à la valeur reçue.C'est-à-dire que le bit est brouillé, de sorte que le récepteur n'a aucune idée de ce qu'était le bit à l'émission. Ce canal est souvent utilisé comme modèle pour les réseaux du type internet. L'effacement modélise alors la perte d'un paquet.

Par exemple, dans les communications internet, des paquets IP sont utilisés, et certains n'atteignent pas le destinataire car le routeur peut avoir fait une mauvaise redirectionou bien ces paquets sont perdus en raison de débordements des mémoires tampon ou à cause de retards excessifs. Les probabilités de transitions du canal sont représentées dans la figure2.3.b et peuvent être exprimées comme suit :

$$\Pr(\text{effacement}) = P_e \qquad (2.4)$$

$$P_e \in [0 \;\; 1]$$

$$\Pr(y_i = 0 \mid x_i = 0) = \Pr(y_i = 1 \mid x_i = 1) = 1 - P_e \qquad (2.5)$$

4- Canal binaire symétrique BSC

Le canal binaire symétrique BSC est un canal discret, stationnaire et sans effet mémoire.Ses ensembles d'entrée $X = \{0,1\}$ et de sortie $Y = \{0,1\}$ sontde dimensions finies.

Puisque ce canal est stationnaire et sans effet mémoire, les probabilités de transitions sont indépendantes du temps et l'élément y_k ne dépend que de l'élément x_k .

Les probabilités de transitions du canal (probabilité d'obtenir la sortie sachant l'entrée) sont représentées dans la figure2.3.aet peuvent être exprimées comme suit :

$$\Pr(y_i = 0 \mid x_i = 1) = \Pr(y_i = 1 \mid x_i = 0) = P_e \qquad (2.6)$$

$$\Pr(y_i = 0 \mid x_i = 0) = \Pr(y_i = 1 \mid x_i = 1) = 1 - P_e \qquad (2.7)$$

Avec $P_e$la probabilité d'erreur, souvent appelée probabilité croisée.$P_e \in [0 \;\; 0.5]$.

5- Canal gaussien à entrée binaire (B-AWGN)

Le canal B-AWGNreprésenté dans la figure2.4, est un canal continu, stationnaire et sans effet mémoire. Il est couramment utilisé en communications numériques car il est simple d'utilisation et permet de fournir une première approximation des phénomènes qui s'appliquent sur un canal de transmission réel.

Figure 2.4.Canal B-AWGN

Le canal AWGN possède les caractéristiques suivantes :

- il est à une entrée x_k discrète et à valeur dans {0,1} dans le cas d'un canal binaire ;
- il est à sortie y_k continue $y_k = x_k + n_k$, avec $n_k$la valeur discrète du bruit
- il est stationnaire (ne dépend pas du temps) ;
- il est sans effet mémoire (y_k ne dépend que de x_k)

Le bruit blanc gaussien WGNest un bruit dont la densité spectrale de puissance est la même pour toutes les fréquences (bruit blanc). Il est dit additif car il est simplement ajouté au signal entrant. Enfin, il est dit gaussien du fait de sa densité de probabilité de transmission est définie comme suit :

$$\Pr(x_i \mid y_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x_i - y_i)^2}{2\sigma^2}} \qquad (2.8)$$

La variance σ du signal est fonction du rapport signal à bruit (SNR) noté :

$$SNR = \frac{E_b}{N_0} = \frac{1}{2\sigma^2} \qquad (2.9)$$

$\sigma \in [0 \ \infty]$.

Avec E_bl'énergie moyenne utilisée pour transmettre un bit (ou énergie bit) utile et N_0 la densité spectrale de puissance du bruit.

$N_0 = KT, \ K = 1{,}38\ 10^{-2}$ J/°K et T: température en °K.

Exemple : cas d'un canal gaussien en visibilité directe avec un effet multi-trajet négligeable etoù la modulation utilisée est la BPSK[19]:

- Le modulateur BPSK mappe en premier lieu les bits 0 et 1 dans le plan I/Q

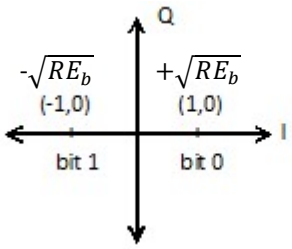


Figure 2.5. Schéma de constellation d'une modulation BPSK

Lacomposante Q est ignorée puisqu'elle n'affecte pas la sortie du démodulateur.

- Les points I/Q sont convertis en sinusoïdes :

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \quad \text{pour le bit 1} \quad (2.10)$$

$$s_0(t) = -\sqrt{\frac{2E_b}{T_b}} \sin(2\pi f_c t) \quad \text{pour le bit 0} \quad (2.11)$$

AvecT_b la durée bit ou inverse du débit binaire.

- le récepteur corrige la forme d'onde (du« 0 » logique ou du « 1 » logique) pour compenser l'atténuation
- Le démodulateur mappe la forme d'onde dans l'espace I / Q et produit le résultat
- L'effet global est que $0 \to 1 + n$ et$1 \to -1 + n$, où n est une variable désignant un bruit aléatoire supposée être distribuée par la loi normale avec une moyenne nulle.

Pour toute valeur$\sigma > 0$, le canal gaussien avec une variance σ^2 à sont ensemble de sortie Y dans $\mathbb{R}$ (ensemble des réels) et des probabilités de transition :

$$P(y|0) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-1)^2}{2\sigma^2}} \quad (2.12)$$

Une loi de distribution normale avec une moyenne égale à 1 et déviation standardσ.

$$P(y|1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y+1)^2}{2\sigma^2}} \quad (2.13)$$

Une loi de distribution normale avec une moyenne égale à -1 et déviation standard σ.

Enfin, la probabilité d'erreur d'un canal AWGN avec une modulation BPSK est fonction du SNR. La probabilité d'erreur binaire d'un tel canal avec une telle configuration est donnée par la formule suivante (Proakis, 2000) [14] :

$$P_{eb} = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (2.14)$$

Où erfc désigne la fonction d'erreur complémentaire :

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2.15)$$

2.2.2 Polarisation du canal[17]

Les codes polaires se construisent par la méthode de polarisation du canal.

Cette méthodeconsiste à transformer le canal initial en N canaux binaires différents, tel que quand N tend vers l'infini, une partie de ces canaux binaires auront une capacité qui tend vers "1"et seront considérés comme les "bons canaux" alors que les autres canaux binairesauront une capacité qui tend vers "0" et serontconsidérés les "mauvais canaux"[17].

- Concept de polarisation :

La polarisation a été introduite par Erdal Arikan en 2008 dans son document phare [2].

Arikan a utilisé cette technique dans le contexte du codage canal et dans une classe spéciale de canaux, ceux du type BMS. Le concept de polarisation peut s'expliquer comme suit :

Soit P un canal BMS et $I(P)$ sa capacité. L'idée de polarisation est de prendre deux copies indépendantes de P pour créer, à partir decelles-ci, deux nouveaux canaux P^- et P^+ tel que :

- la somme des capacités de P^- et P^+est égale au double de la capacité deP, ainsi donc aucune information n'est perdue

- P^+ est « meilleur » que P et P^-est « pire » que P

Considérons le schéma de la figure (2.6) :

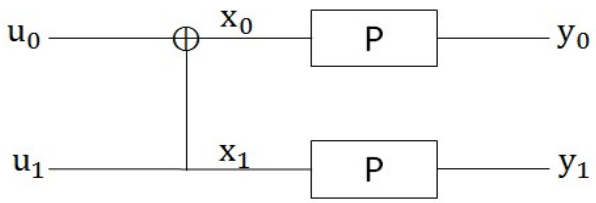


Figure 2.6.Transformation du canal P en deux copies (utilisations) indépendantes

$$x_0 = u_0 \oplus u_1 \quad (2.16)$$

$$x_1 = u_1 \quad .(2.17)$$

$\oplus$ désigne l'opérateur « OU exclusif ».

On peut dès lors affirmer que :

$$u_0 = x_0 \oplus x_1 \quad (2.18)$$

et que :

$$u_1 = x_1 \quad (2.19)$$

Puisque la matrice, notée G reliant $[u_0, \ u_1]$ à $[x_0, x_1]$ est son propre inverse :

$$X = U \times F \text{ avec } F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ la matrice génératrice} \quad (2.20)$$

Étant données les sortie (y_0, y_1), on suppose que l'on désire déduire les valeurs des entrées (u_0, u_1). Cette tâche peut être accomplie de manière successive :

- en première étape on déduit la valeur deu_0, u_1 est considérée comme un bruit
- en seconde étape on utilisera l'estimé deu_0 , noté $\hat{u}_0$, pour déduire u_1

De ce point de vue, on peut affirmer que :

- le canal P^-est le canal que $u_0$voit étant donné les observations (y_0, y_1),
- le canal P^+ est le canal que u_1 voit étant donnée les sorties (y_0, y_1) et la valeur actuelle de u_0 soit$\hat{u}_0$.

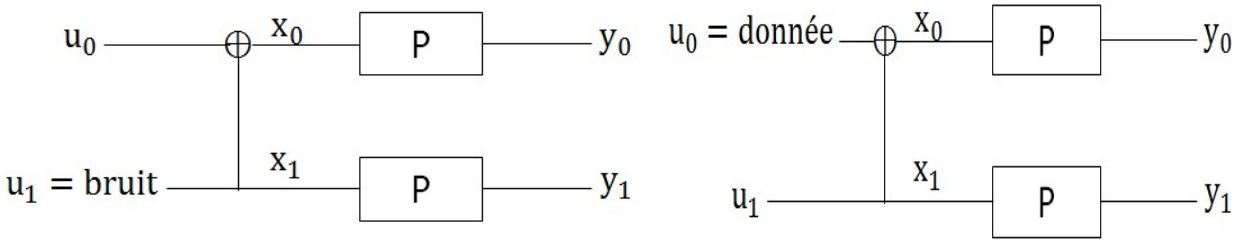


Figure2.7.à gauche : canalP^- (mauvais) vue par u_0 , à droite et canal P^+(bon) vu par u_1

Intuitivement, on peut affirmer que P^- est le mauvais canal (canal dégradé par rapport àP) et P^+le bon canal (canal optimisé par rapport àP).

Autrement dit, P^+ est la version à « moindre bruit » de P et P^- est la version « à plus de bruit » de$P$.

En conséquence, à partir de deux copies du canal deux variables indépendantes et identiquement distribuées (iid) du canal P, nous avons créé deux canaux P^- et P^+avec la propriété :

$$P^- \leq P \leq P^+ (2.21)$$

La relation (2.21) ci-dessus nous indique déjà une sorte de polarisation vers les deux canaux extrêmes: totalement sans bruit (canal parfait ou meilleur canal) ou complétement bruité (canal inutile ou mauvais canal).

L'idée de la polarisation est d'appliquer récursivement cette simple transformation aux canaux P^{-}et P^{+}pour créer d'autres canaux polarisées :

$P^{(+)+}$, $P^{(+)-}$

$P^{(-)+}$, $P^{(-)-}$

Ce processus continuera jusqu'à créer un certain nombre $N = 2^{n}$ canaux (nombre limite que nous verrons par la suite) :

$P^{++\cdots+}$, $P^{++\cdots-}$, $P^{--\cdots-}$…etc.

Un phénomène remarquable apparait est que : lorsque n augmente, les canaux créés deviennent de plus en plus polarisés, c'est-à-dire que presque tous les canaux sont très proches de l'un des deux états extrêmes suivants:

- totalement sans bruit (avec capacité 1)

Ou

- complètement bruités (avec capacité 0).

Il est clair que pour longueurs de blocs très grandes la fraction de canaux complètement sans bruit tend vers $I(P)$ et la fraction de les canaux complètement bruité tend vers$1 - I(P)$.

La complexité de codage et décodage des codes construits par cette méthode est, par rapport à la longueur du mot de code N, $O(N \log_2(N))$ avec O l'ordre de grandeurdans les processus de codage et de décodage [10]. Pour un décodeur à maximum de

vraisemblance ML, le nombre d'opération est $O(2^K)$avec $K = RN$ et où Rreprésente le taux de codage[19].

Exemple : Si $N = 64$ et $R = 1/2$alors$K = 32$. Dans ce cas $O(N\log_2(N)) = 384$et$O(2^K) = 4294967296$.

La polarisation du canal permet d'identifier les indices des K bits de message à envoyer notés I et les indices des bits faiblesnotés I_c.

On utilise lesbons canaux pourenvoyer les Kbits d'information et on envoie sur les mauvais canaux une séquence de bits connueau niveau du décodeur.Ces derniers sont appelés bits gelés (*frozen bits*).

La somme des capacités de ces bons canaux tend vers la capacité du canal initial ce qui permet de communiquer avec un débit proche de la capacité du canal.

Pour mieux comprendre laméthode de polarisation, considérons le canal initial

$$P: X \to Y$$

Ce canal est un canal symétrique ayant :

- un alphabet d'entrée binaire$X = \{0,1\}$
- un alphabet de sortie Yet
- une probabilité de transition $P(Y \mid X)$.

Rappels : $P(Y|X)$ est une probabilité conditionnelle ou probabilité de Y sachant X a été réalisé. X étant de probabilité non nulle.

En utilisant la formule suivante [15]

$$P^-(y_0, y_1|u_0) = \sum_{u_1 \in \{0,1\}} \frac{1}{2} p(y_0|u_0 \oplus u_1) p(y_1|u_1). \quad (2.22)$$

$$P^{+}(y_0, y_1, u_0|u_1) = \frac{1}{2}P(y_0|u_0 \oplus u_1)p(y_1|u_1) \quad (2.23)$$

On transformeP en deux canaux distincts :

$P^{-}: X \to Y^2$ et $P^{+}: X \to X \times Y^2$

Autrement dit :

$P = \{0,1\} \to$ Yalors$P^{-} = \{0,1\} \to Y^2$et$P^{+} = \{0,1\} \to \{0,1\} \times Y$

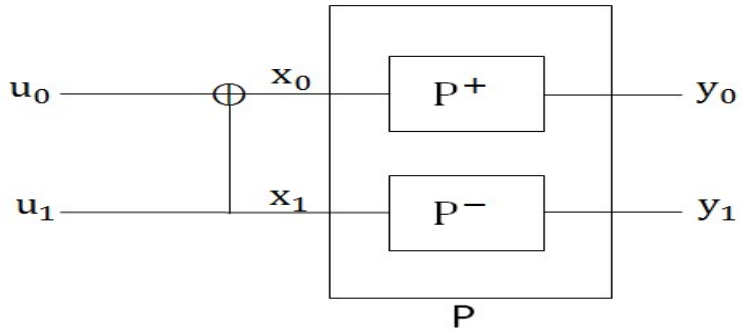


Figure 2.8. Codeur CP(2,2) (voir §2.3)

- $U = [u_1, u_2]$: vecteur d'information
- $\oplus$: XOR (ou exclusif)
- P_2 : canal initial polarisé en deux canaux binaires P^{-}et P^{+}
- y_1 et y_2 : vecteur en sortie du canal.

Puis, on applique cette même transformation sur les canaux P^{-}et P^{+}pour avoir quatre canaux distincts et ainsi de suite. En réalisant cette division nfois on obtient :

$$N = 2^n \text{canaux.} \quad (2.24)$$

Ces canaux sont équivalents aux Nbits du mot de code, et sont appelés canaux binaires.

Pour la conception d'un bon code polaire, il faut savoir choisir les meilleurs canaux binaires pour envoyer les bits d'information.

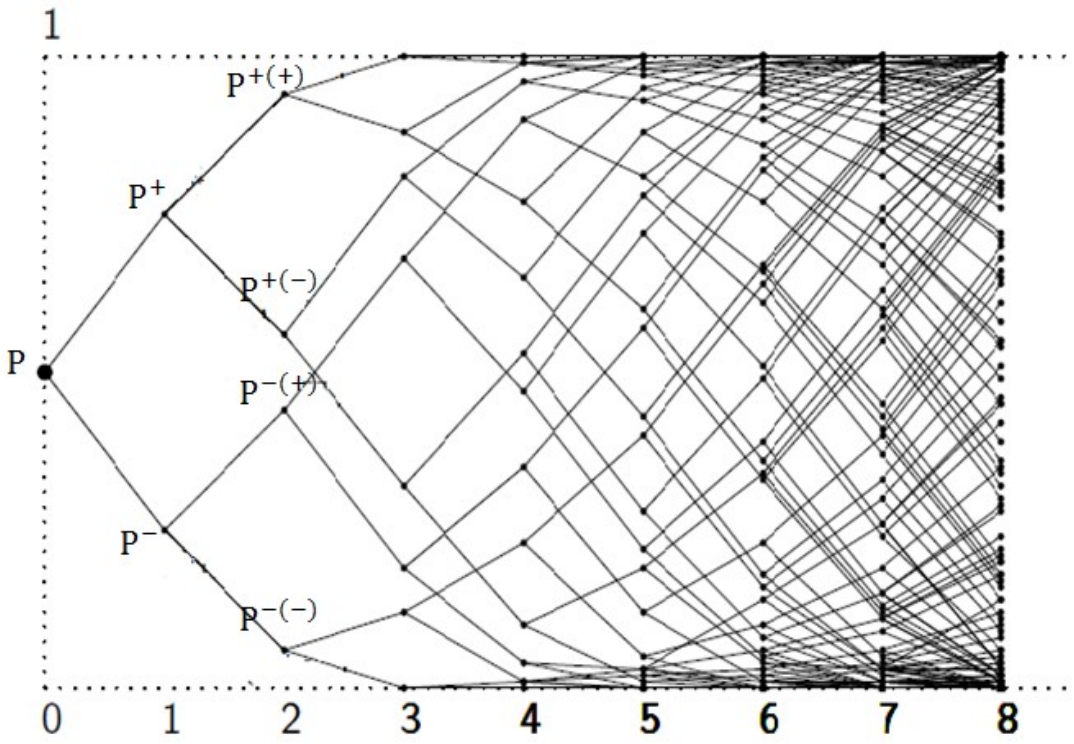


Figure 2.9. Processus de polarisation d'un canal BMS à l'aide des codes polaires.

2.2.3Conception des codes polaires [1]

Pour la conception d'un bon code polaire, il faut savoir choisir les meilleurs canaux binairespour envoyer les bits d'information$\mathrm{K}$. Pour se faire, il existe plusieurs méthodes. Certaines méthodes se basent sur le paramètre de Bhattacharyya[15]pour choisir ces canaux.Pour cela on calcule, pour chaque canal binaire, le paramètre de Bhattacharyya Z qui représente une borne supérieure sur la probabilité d'erreur du canal.Puis, on sélectionne les canaux binaires ayant les plus petites valeurs de Z (faibles probabilités d'erreur) pour envoyer les bits d'information et on fixe à " 0 " ou à " 1 " les entrées des canaux binaires restants(fortes probabilités d'erreur) pour envoyer les bits gelés (*frozen bits*).

Par la suite, on multiplie ce vecteur d'entrée par une matrice d'inversion des bits définie par Arikan [1]puis on multipliele résultat obtenu par la matrice génératrice du code polaire (voircodage polaire, § 2.3).

Le paramètre de Bhattacharyya est donné par la formule suivante :

$$Z = \sum_{y \in Y} \sqrt{p(y|0)p(y|1)} \qquad (2.25)$$

Où Y est l'alphabet de sortie, $p(y|0)$ et $p(y|1)$ représentent les probabilités de transition du canal lorsque les entrées sont 0 et 1 respectivement.

Mais le calcul de la valeur exacte de$Z$ est impossible à cause du nombre de sortie qui augmente rapidement avec chaque partition(branche)et aura une valeur énorme à la fin de la polarisation.Pour cela, il faut utiliser une méthode d'approximation du paramètre$Z$.

1- Méthode d'approximation du canal à un canal binaire à effacement [15]

L'idée est d'utiliser la méthode appliquée canal binaire à effacement quel que soit le canal étudié. Ce qui change seulement pour chaque canal est la méthode de calcul de la valeur initial du paramètre de Bhattacharyya Z_0pour le canal avant polarisation.

Pour le canal binaire à effacement(voir §2.2.1), le paramètre de Bhattacharyya a une forme récursive simple.

Soit Pun canal binaire à effacement.P^-et P^+sont les canaux obtenus par les formules de polarisations (2.22) et (2.23) à partir de P. On peut calculer le paramètre Zde ces canaux par les formules suivantes :

$$Z(P^-) = 2Z_0(P) - Z_0(P)^2 \quad (2.26)$$

$$Z(P^{+}) = Z_0(P)^2 \quad (2.27)$$

On applique ces formules Nfois pour obtenir les paramètres de Bhattacharyya des 2^ncanaux binaires.

Enfin, la polarisation de n'importe quel canal BMS se fait comme suit :

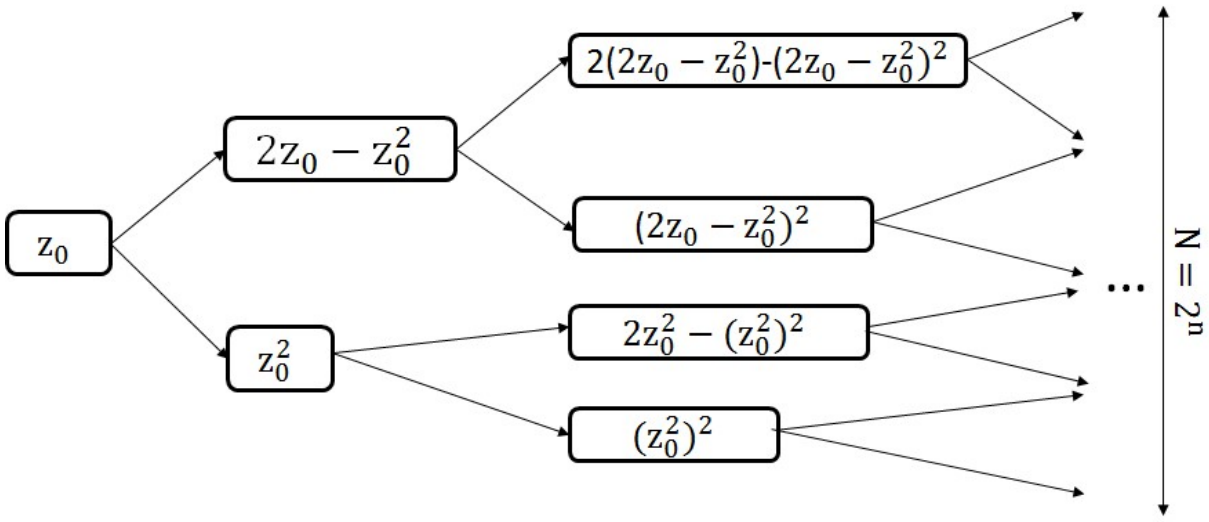


Figure 2.10. Polarisation du canal (n'importe quel type de canal) avec le paramètre de Bhattacharyya

S. Kamel a donné dans [15]la valeur initiale du paramètre de Bhattacharyya Z_0dequelques canaux :

- Canal binaire à effacement :

$$Z_0 = P_e \quad (2.28)$$

avec la probabilité d'effacement du canal BEC(voir §2.1.1)

- Canal binaire symétrique :

$$Z_0 = 2\sqrt{P_e(1-P_e)} \quad (2.29)$$

Avec la probabilité d'erreur du canal BSC (voir §2.1.1)

- Canal Gaussien à entrée binaire :

$$Z_0 = \exp\left(-R\frac{E_b}{N_0}\right) \quad (2.30)$$

R : représentant le taux de codage (ou rendement du code)

E_b : l'énergie nécessaire pour la transmission d'un bit

N_0 : densité de puissance de bruit

$\frac{E_b}{N_0}$: le rapport signal sur bruit (SNR) en transmission numérique

2- Sélection des indices de bits d'information

Il existe plusieurs algorithmes pour sélectionner les indices de K (bits d'information) à partir des indices de$N$. La plus simple est l'utilisation de la forme récursive du paramètre de Bhattacharyya : $z \to \{2z - z^2, z^2\}$ (d'après la figure2.10).

Pour se faire, reprenons le schéma de lafigure2.10:

- On arrête d la polarisation lorsque l'arbre atteint N feuilles indexées du haut de $0,...,N-1$

- On recherche les feuilles ayantles plus petites valeurs (faible probabilité d'erreur) et notons leurs indices I

- Les autres feuilles ayant les plus grandes valeurs (forte probabilité d'erreur)sont gelées et fixés à "0" ou à "1" et notons leurs indices I_c .

3- Non universalité des codes polaires

En théorie de codage, la plus part des codes sont « universels » dans le sens où leur définition est indépendante du SNR du canal [10].

Arikan a défini un ensemble I_c de codes polaires tel que le BLER de ces codes soit minimum en utilisant l'algorithme de décodage SC. Puisque le BLER est une fonction duSNR, ce n'est donc pas surprenant que les codes polaires changent avec la valeur fixée du SNR ou (*design-SNR*).

Cette caractéristique de non-universalité des codes polaires est très importante.

Une modification du SNR opérationnel est possible en pratique, mais un changement de code selon le SNR n'est pas souhaitable. Par conséquent, il est souhaitable deconstruire un code polaire dans un *design-SNR* et l'utiliser pour une gamme de SNR possibles [10].

Le choix du *design-SNR* est essentiel pour la performance à tous les SNR d'intérêt. Malheureusement, nous n'avons pas trouvé d'étude pour identifier le meilleur design-SNR pour toute construction de code polaire.

Il a eu quelques tentatives récentes de conception de codes polaires universels mais le coût fut beaucoup plus élevé du point de vue complexité au décodeur et / ou encodeur. Selon [10] il n'y a pas eu beaucoup d'études dans ce sens sur les codespolaires. Beaucoup de travaux de recherche considèrent souvent choix heuristiquede la conception du design-SNR.

La meilleure performance qu'on peut atteindre est approximativement la même pour tout algorithme de construction pour au moins $N \leq 64K$ [10].

2.3 Codage polaire

Un code polaire, notéCP(N, K), est un code en bloc linéaire (Arıkan, 2008)[1] de taille $N = 2^n$,n étant un entier naturel, contenant Kbits d'information.Sa matrice génératrice est une sous-matrice de la nième puissance de Kronecker[1] de $k = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ notée :

$$F = k^n = \begin{bmatrix} k^{n-1} & 0 \\ k^{n-1} & k^{n-1} \end{bmatrix} \tag{2.31}$$

Composée de K lignes.

Ces Klignes sont choisies en supposant un décodage par annulation successiveSCqui permet de polariser la probabilité d'erreur des bits du message.Voici quelques exemples :

- Pour $N = 2 \rightarrow n = 1 \rightarrow F = k^1 = \begin{bmatrix} k^0 & 0 \\ k^0 & k^0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

- Pour $N = 4 \rightarrow n = 2 \rightarrow F = k^2 = \begin{bmatrix} k^1 & 0 \\ k^1 & k^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

- Pour $N = 8 \rightarrow n = 3 \rightarrow F = k^3 = \begin{bmatrix} k^2 & 0 \\ k^2 & k^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$.

- …etc.

Le processus de codage consiste à construire un vecteur Ude tailleN, $U = [u_0, u_1, \dots, u_{N-1}]^T$contenant les K bits d'information et les N-K bits gelés fixé à 0 ou 1 (voir §2.2.3).

Ce vecteur Uest construit de telle manière que les bits d'information soient localisés sur les indices les plus fiables correspondant aux K lignesde k^nsélectionnées précédemment. Le mot de code correspondant$X = [X_0, X_1, \dots, X_{N-1}]^T$, peut ensuite être calculé simplement de la façon suivante:

$$X = U \times F = U \times k^n \qquad (2.32)$$

2.3.1 Représentations des codes polaires

La complexité de codage et décodage d'un code dépend du nombre d'opération utilisé par la méthode de construction de ce code. On cherche toujours à avoir une complexité acceptable et pour se faire il faut minimiser le plus possible le nombre d'opération d'où les différentes représentions des codes polaires.

Un code polaire peut être représenté sous forme matricielle à partir de la matrice F et sous forme graphique (*factor graph)*. Ce dernier pouvant être utilisé pour le codage et pour le décodage.

1- Exemple de représentation matricielle

Soit un code polaire noté $CP(8,4)$ où :

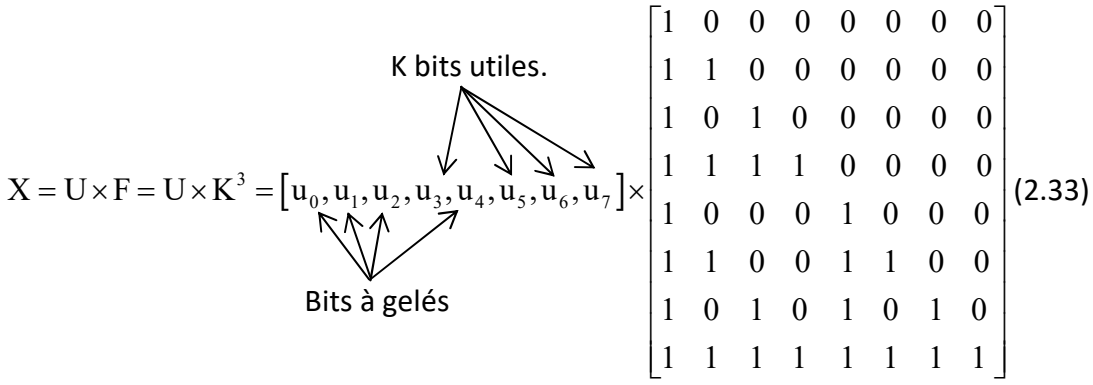
- la taille de la donnée est $\boldsymbol{K = 4}$
- la longueur du code est $\boldsymbol{N = 8}$
- le vecteur U, en entrée, est de taille N et noté $U = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7]$
- le rendement du code est $R = 0.5$.

Supposons qu'après la polarisation du canal, les indices I des K bits sont le 3, 5, 6 et 7 et celles des bits à gelés $\mathrm{I_c}$ sont 0, 1, 2, 4.

Notons que le nombre de bits gelés dépend du rendement $\boldsymbol{R} = \frac{\boldsymbol{K}}{\boldsymbol{N}}$du code.

Dans cet exemple le rendement est égale à 0.5 cela implique que la moitié du vecteur U serait gelé et l'autre moitié serait composé des K bits d'information.

Le mot de code sera donc :



$$\mathrm{X} = \mathrm{U} \times \mathrm{F} = \mathrm{U} \times \mathrm{K}^3 = \left[\mathrm{u}_0, \mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \mathrm{u}_4, \mathrm{u}_5, \mathrm{u}_6, \mathrm{u}_7\right] \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.33)$$

Le résultat de cette multiplication donne :

$$\mathrm{X} = \begin{bmatrix} \mathrm{u}_0 + \mathrm{u}_1 + \mathrm{u}_2 + \mathrm{u}_3 + \mathrm{u}_4 + \mathrm{u}_5 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_1 + \mathrm{u}_3 + \mathrm{u}_5 + \mathrm{u}_7 \\ \mathrm{u}_2 + \mathrm{u}_3 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_3 + \mathrm{u}_7 \\ \mathrm{u}_4 + \mathrm{u}_5 + \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_5 + \mathrm{u}_7 \\ \mathrm{u}_6 + \mathrm{u}_7 \\ \mathrm{u}_7 \end{bmatrix}^{\mathrm{T}} \quad (2.34)$$

A l'entrée du canal, le mot de code sera :

$$X = \begin{bmatrix} 0+0+0+u_3+0+u_5+u_6+u_7 \\ 0+u_3+u_5+u_7 \\ 0+u_3+u_6+u_7 \\ u_3+u_7 \\ 0+u_5+u_6+u_7 \\ u_5+u_7 \\ u_6+u_7 \\ u_7 \end{bmatrix}^T \quad (2.35)$$

2- Exemple de représentation sous forme de *factor graph*

Dans cet exemple, deux matrices génératrices ainsi que leur représentation graphique sont présentées.

- $CP(2,2)$ et $U = [u_0, u_1] \rightarrow X = [u_0, u_1] \times \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^T \Rightarrow$

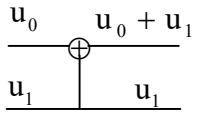


Figure 2.11.Matrice génératrice et graphe du codeur polaire CP(2,2).

- $CP(8,4)$ et $U = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7]$

$$\Rightarrow X = \begin{bmatrix} 0+0+0+u_3+0+u_5+u_6+u_7 \\ 0+u_3+u_5+u_7 \\ 0+u_3+u_6+u_7 \\ u_3+u_7 \\ 0+u_5+u_6+u_7 \\ u_5+u_7 \\ u_6+u_7 \\ u_7 \end{bmatrix}^T \quad (2.36)$$

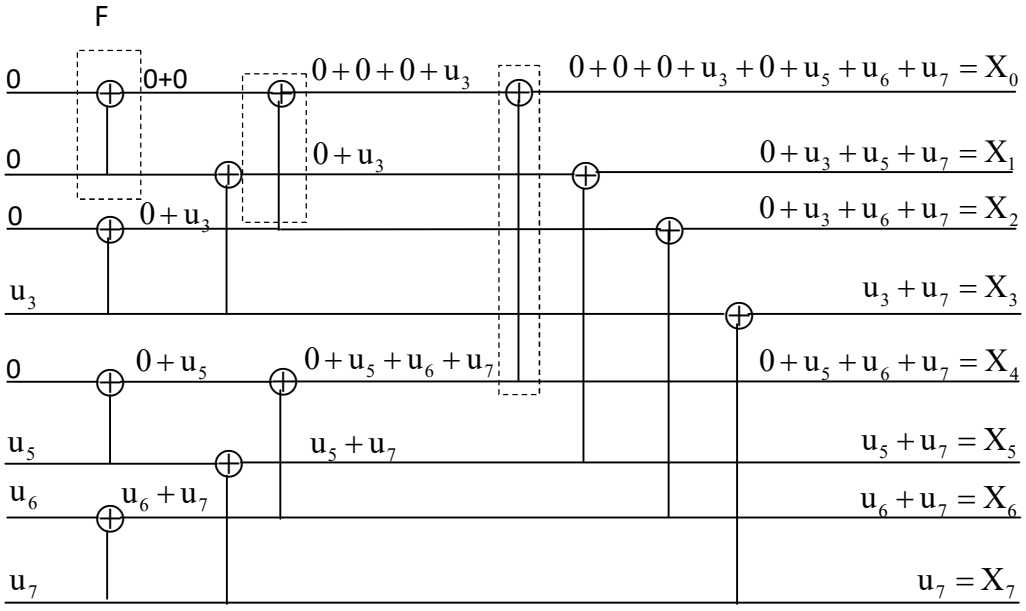


Figure 2.12.Matrice génératrice et graphe de codeur CP(8,4)

Les nœuds de parités ($\oplus$) se comportent comme des XOR et les nœuds de variables($\perp$) font simplement passer la valeur binaire à l'étage suivant.

Plus généralement, le *factor graph* d'un code polaire, présenté dans (Arikan 2009) [2], de taille $N = 2^n$ est composé de :

n étapes de $\frac{N}{2}$ nœuds de parités de degrés 3 et, $\frac{N}{2}$ nœuds de variables de degrés 3.

Le degré d'un nœud représente son nombre de connexion avec d'autre nœud. Le *factor graph* peut être utilisé pour le codage et le décodage. Pour le codage, le vecteurd'entréU, sur le côté gauche, est propagé dans le graphe dans le but de générer le mot de code$X$, sur la droite.

2.4 Décodage des codes polaires

Figure 2.13.Schéma synoptiquedu processus de décodage polaire

$u_i = [u_0, u_1, ..., u_{N-1}]$: bits d'information + bits gelés ;

$X = [X_0, X_1, ..., X_{N-1}]$: mot de code

$Y = [Y_0, Y_1, ..., Y_{N-1}]$: version bruité du mot de code reçu

$\hat{u}_i = [\hat{u}_0, \hat{u}_1, ..., \hat{u}_{N-1}]$: bits estimés

N : la taille du mot de code.

Une fois le message transmis au travers du canal de communication, la version bruitée, $Y = [Y_0, Y_1, ..., Y_{N-1}]$ du mot de code $X = [X_0, X_1, ..., X_{N-1}]$ est reçue. Le but du décodage est d'estimer le vecteur U à partir de la version bruitée du mot de code Y.

Plusieurs algorithmes de décodage ont été proposés dans Arikan [1] pour les codes polaires tel que:

- *Successive Cancellation* (SC)
- *Successive Cancellation with Lists* (SCL)
- *Belief Propagation* (BP).
- *Soft-Cancellation* (SCAN)
- …etc.

2.4.1 Décodage SC

La complexité du décodeur SC est de l'ordre de $N \log_2(N)$, N étant la longueur du code.

Ce décodeur se base sur les rapports de vraisemblance LR pour estimer les valeurs des bits d'informations. Il décode les bits u_i par ordre c'est-à-dire du bit d'indice 0 jusqu'au bit d'indice $N - 1$.

Pour pouvoir décoder le bit numéro i, le décodeur est sensé connaitre :

- les LR des sorties des N canaux binaires
- les estimations des i premiers bits

- les positions des bits d'information K et,
- les valeurs des bits gelés$\mathrm{N}-\mathrm{K}$.

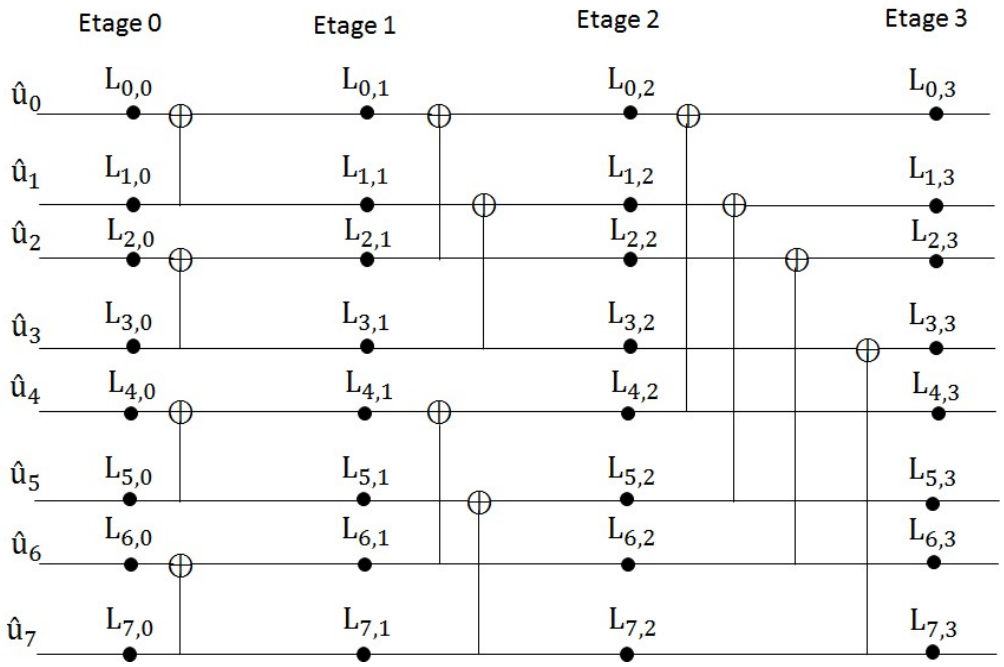


Figure 2.14.Schéma synoptiquedu processus du décodage SC avec

$\boldsymbol{L_{i,j}}$ la valeur de LR du $\boldsymbol{i^{ème}}$ bit au $\boldsymbol{j^{ème}}$étage

Si le bit iest un bit gelé, le décodeur lui donne directement sa valeur qu'il connait déjà. Sinon, il examine la valeur duLRcorrespondant à ce bit, si elle est supérieure à 1 ($\mathrm{L_{i,j}}$ >1) alors le bit est 0, sinon le bit est 1.

i : le nombre de sortie du décodeur avec $0 \leq \mathrm{i} \leq \mathrm{N}-1$

j : le nombre d'étage du décodeur avec $0 \leq \mathrm{j} \leq \mathrm{n}$

Pour estimer la valeur du bit$\mathrm{u_i}$, notée$\hat{\mathrm{u}}_\mathrm{i}$, le décodeur se base sur l'observation$\mathrm{Y} = [\mathrm{Y_0 Y_1 \dots Y_{N-1}}]$ et sur les valeurs des bits estimésprécédemment $\hat{\mathrm{u}}_0^{\mathrm{N}-1} = [\hat{\mathrm{u}}_0 \hat{\mathrm{u}}_1 \dots \hat{\mathrm{u}}_{\mathrm{i}-1}]$et ce en calculant les valeurs des LR(s) suivants :

$$L_{i,0} = \frac{\Pr(Y_0^{N-1}, \hat{u}_0^{N-1} | u_i = 0)}{\Pr(Y_0^{N-1}, \hat{u}_0^{N-1} | u_i = 1)} \quad (2.37)$$

Le *factor graph* peut être utilisé pour le codage et le décodage.

Afin de bien comprendre le décodage par annulation successives, prenons l'exemple du codeur de base où N=2 [19] :

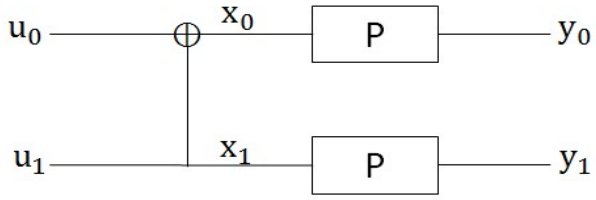


Figure 2.15.Encodage et transmission dans une chaine d'ordre 2

L'encodeur mappe les u_i (incluant les bits gelés) auxx_i . Ces derniers sont transmis à travers le canal P et, les y_i sont reçusen sortie.

En supposant avoir une connaissance complète deP, les sorties y_i peuvent être des bits (*hard*) ou des valeurs réelles (*soft*).

Comme indiqué au § 2.2.2 :$x_0 = u_0 \oplus u_1$, $x_1 = u_1$ et $u_0 = x_0 \oplus x_1$.

Pour chaque valeur de yreçue nous connaissons les probabilités de transition $P(y|x = 0)$ et $P(y|x = 1)$ (voir probabilités de transition du canal gaussien dans §2.2.1).

En posant :

$$a = P(y_0|x_0 = 0), b = P(y_0|x_0 = 1),\ c = P(y_1|x_1 = 0) \text{ et } d = P(y_1|x_1 = 1) \quad (2.38)$$

Sachant que les transitions $x_0 \to y_0$ et $x_1 \to y_1$ sont indépendantes et en factorisant les probabilités par paire,on obtient :

$$P(y_0, y_1|u_0 = 0) = \frac{ac+bd}{2} \quad \text{et } P(y_0, y_1|u_0 = 1) = \frac{bc+ad}{2} \qquad (2.39)$$

Le rapport de ceux probabilités est : $\frac{\frac{ac}{bd}+1}{\frac{c}{d}+\frac{a}{b}}$. C'est le rapport de vraisemblance de u_0 connaissant les sorties y_0 et y_1 qu'on note $LR(u_0)$.

En posant $p = \frac{a}{b}$ et $q = \frac{c}{d}$, on définit la fonction

$$LR(u_0) = f(p, q) = \frac{pq+}{p+q} \qquad (2.40)$$

Si $f(p, q) \geq 1$ alors l'estimé de u_0 vaut 0 ($\hat{u}_0 = 0$)

Si $f(p, q) < 1$ alors $\hat{u}_0 = 1$

u_0 étant décodé, l'étape qui suit est celle du décodage de u_1 et qui consiste à calculer $LR(u_1)$.

En se basant sur la règle du décodage par annulations successives (SCD) alors on suppose que u_0 a été correctement décodé.

$$\text{Si } u_0 = 0 \text{ alors } LR(u_1) = \frac{P(y_0,y_1|u_0=0,u_1=0)}{P(y_0,y_1|u_0=0,u_1=1)} = \frac{P(y_0,y_1|x_0=0,x_1=0)}{P(y_0,y_1|x_0=1,x_1=1)} = \left(\frac{a}{b}\right)\left(\frac{c}{d}\right) \qquad (2.41)$$

$$\text{Si } u_0 = 1 \text{ alors } LR(u_1) = \frac{P(y_0,y_1|u_0=1,u_1=0)}{P(y_0,y_1|u_0=1,u_1=1)} = \frac{P(y_0,y_1|x_0=1,x_1=0)}{P(y_0,y_1|x_0=0,x_1=1)} = \left(\frac{a}{b}\right)^{-1}\left(\frac{c}{d}\right) \qquad (2.42)$$

Comme précédemment, on définit la fonction

$$LR(u_1) = g(p, q, u_0) = p^{1-2u_0}q \qquad (2.43)$$

$\hat{u}_1 = 1$ si $g(p, q, u_0) \geq 1$

$\hat{u}_1 = 0$ si $g(p, q, u_0) < 1$

Dans la partie le codage d'une structure de codage/décodage polaire d'ordre N, les nœuds de parités ($\oplus$) se comportent comme des XOR et les nœuds de variables($\perp$) font simplement passer la valeur binaire à l'étage suivant.

Dans la partie décodage, ces nœuds ne symbolisent plus un simple XOR ni une simple copie (voir figure2.15).

Le décodeur estime successivement la valeur des bits $\hat{u}_i$ est se basant sur les valeurs des LR ($L_{i,j}$) et les valeurs des sommes partielles $S_{i,j}$ qui n'est autre que le recodage des bits estimés.

Ces valeurs sont calculées par les formules suivantes [4]:

$$L_{i,j} = \begin{cases} F(L_{i,j}, L_{i+2^j,j+1}) & \text{Si } B_{i,j} = 0 \\ G(L_{i-2^j,j+1}, L_{i,j+1}, S_{i-2^j,j}) & \text{Si } B_{i,j} = 1 \end{cases} \quad (2.44)$$

$$S_{i,j} = \begin{cases} H_l(S_{i,j-1}, S_{i+2^{j-1},j-1}) & \text{Si } B_{i,j-1} = 0 \\ H_u(S_{i,j-1}) & \text{Si } B_{i,j-1} = 1 \end{cases} \quad (2.45)$$

Avec :

$$\begin{cases} F(a,b) = \dfrac{1+ab}{a+b} \\ G(a,b,S) = b \times a^{1-2S} \\ H_u(S) = S \\ H_l(S,S') = S \oplus S' \\ B_{i,j} = \dfrac{i}{2^j} \bmod 2 \end{cases} \quad \begin{array}{l} \\ \\ \\ \\ 0 \le i < N\text{-}1 \text{et} \\ 0 \le j < n \text{ (le numéro des étages)} \end{array} \quad (2.46)$$

En effet, les fonctions F et G peuvent être représentées telles que :

$$F(L_a, L_b) = \frac{1 + L_a L_b}{L_a + L_b} \quad (2.47)$$

$$G(L_a, L_b, S) = L_b \times L_a^{1-2S} \quad (2.48)$$

Les fonctions de décodage F et G sont complexe du côté implémentation. D'où le besoin d'un changement de domaine suivie par une simplification.

Pour se faire, comme explicité dans l'article de C. Leroux (2014)[4], il faut passer dans le domaine logarithmique. Les valeurs ainsi manipulées sont appelées *Log-Likelihood Ratio* (rapport de vraisemblance logarithmique LLR). Les fonctions F et G deviennent alors :

$$f(L_a, L_b) = 2\tanh^{-1}(\tanh(\frac{L_a}{2}).\tanh(\frac{L_b}{2})) \quad (2.49)$$

$$f(L_a, L_b) \approx \text{sign}(L_a).\text{sign}(L_b) \times \min(|L_a|, |L_b|) \quad (2.50)$$

$$g(S, L_a, L_b) = (-1)^S \times L_b + L_a \quad (2.51)$$

AvecS le recodage des bits estimé (voir figure2.15).

Dans ce cas, si le bit $\hat{u}_i$ à estimer n'est pas un bit gelé, la décision à prendre par le décodeur SC est la suivante :

- $\hat{u}_i = 0$ si LLR > 0,
- $\hat{u}_i = 1$ sinon.

Voici un exemple de décodeur SC CP(8,4) présenté dans Leroux [4] :

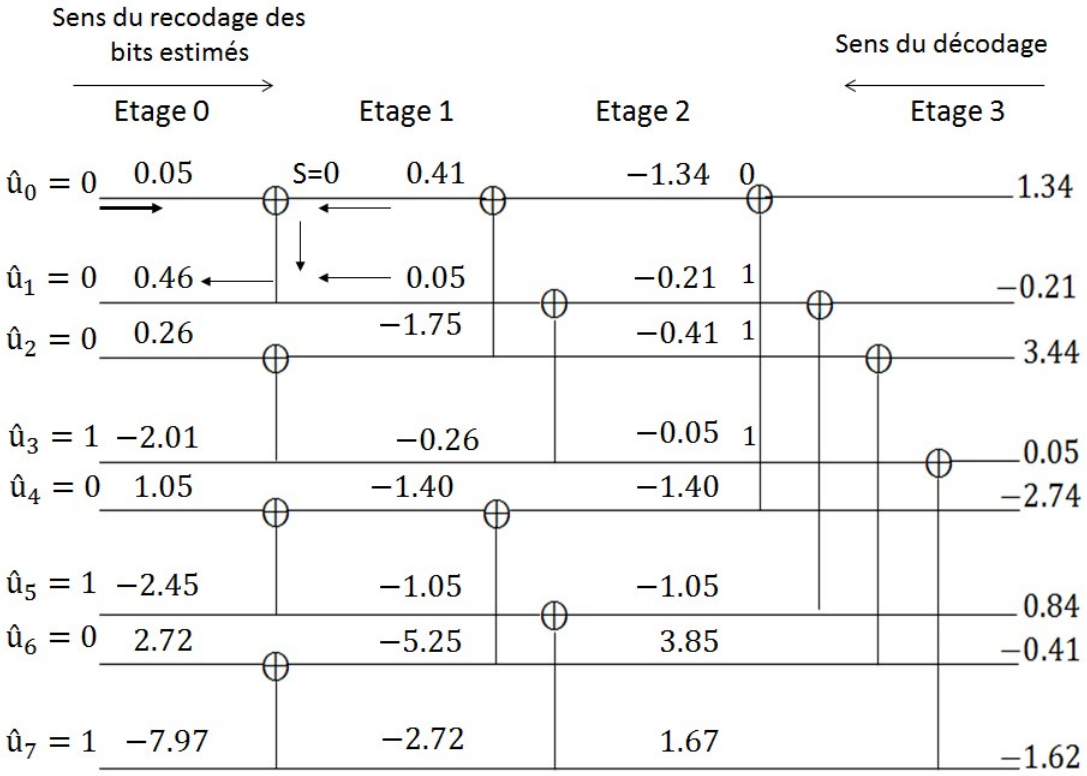


Figure 2.16.Décodeur SC du CP(8,4)

Remarque:

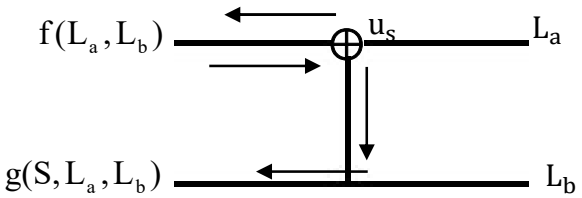


Figure 2.17.Définition des fonctions de décodage f et g

Les valeurs du LLR à l'étage 0 sont les informations du canal. Dans cet exemple, elles ont été données. [4]

Des études faites dans Camille Leroux [4] ont montré qu'il existe d'autres algorithmes de décodage, pour les codes polaires, plus performant que le décodage SC.

2.5 Performances des codes polaires

2.5.1 Introduction

Dans ce paragraphe, nous allons mettre en évidence les performances des codes polaires et ce en faisant des comparaisons du point de vu BER en fonction du SNR avec les LDPC et les turbo codes.

2.5.2 Estimation du BER dans un canal AWGN pour différentes longueur du code [19]

1- Cas d'une modulation BPSK

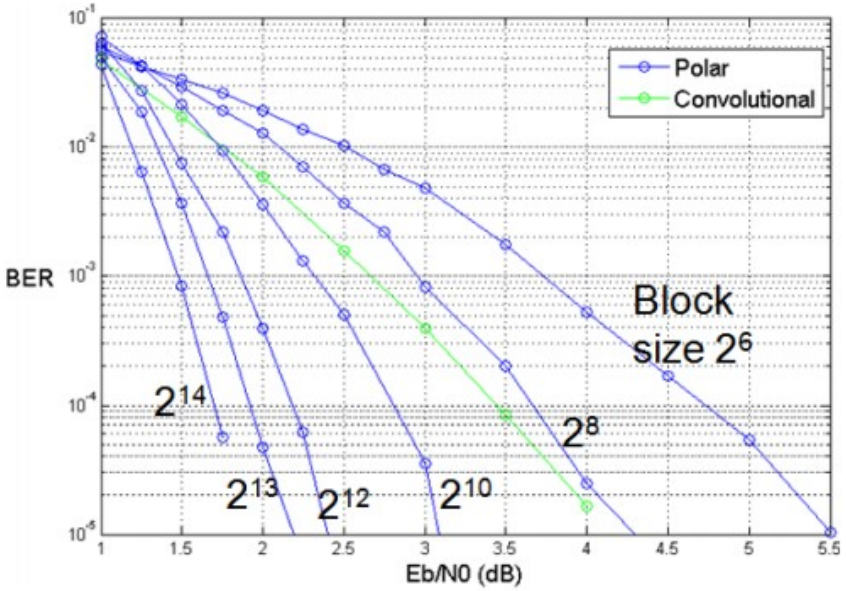


Figure 2.18. Estimation du BER pour code polaire de taux de codage de ½ en BPSK dans des canaux gaussiens [19]

Interprétation : une ligne bleue de la figure 3.1 qui atteint le bas du graphique indique qu'une estimation de BER inférieure à $\mathbf{10^{-5}}$ a été calculée.Cependant, dans les lignes bleues de la figure, chaque point sur la ligne représente un code construit pour une valeur particulière de$\frac{\mathbf{E_b}}{\mathbf{N_0}}$. C'est une pratique courante dans la littérature des codes polaires parce que ces derniers sont facilement optimisés pour différents niveaux de$\frac{\mathbf{E_b}}{\mathbf{N_0}}$

.

2- Cas d'une modulation QPSK

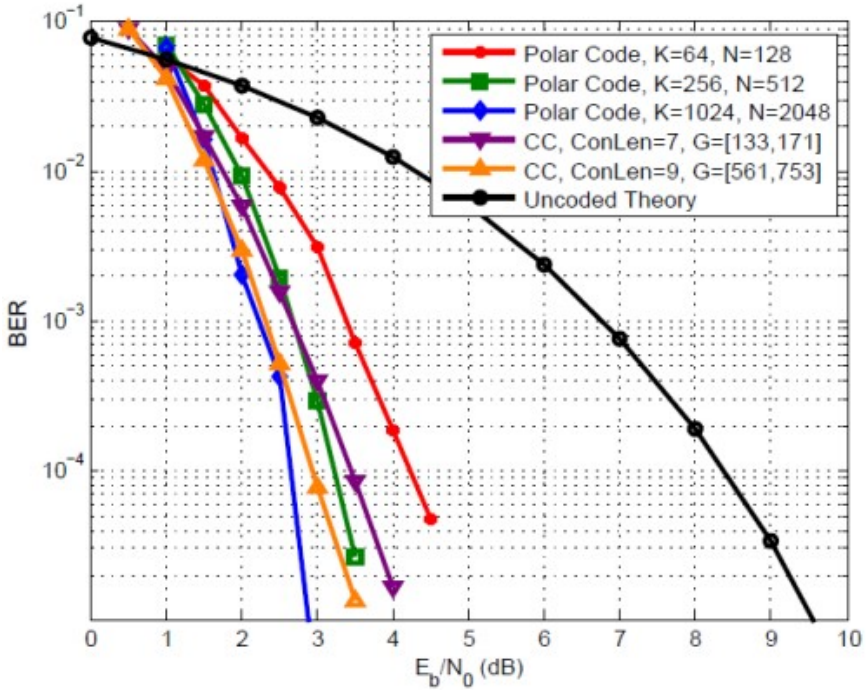


Figure 2.19. Estimation du BER pour code polaire de taux de codage de ½ en QPSK dans des canaux gaussiens [19]

Interprétation : sur la figure 2.2, la courbe en rouge représente un code polaire avec K=64 et N=128. Les courbes représentées en vert et en bleu représentent aussi un code polaire mais respectivement avec quatre fois la taille code en rouge et quatre fois la taille de celui en bleu.

La courbe violète représente un code convolutif avec ConLen=7 (6 registres +1) et $\mathrm{G} = [133,171]$ d'où un taux de codage égal à $\frac{3}{4}$, et la courbe orange un code convolutif avec ConLen=9 (8 registre +1) et $\mathrm{G} = [561,753]$ d'où un taux de codage égal à $\frac{1}{2}$.

Nous remarquons que les codes polaires atteignent une meilleur performance avec une grande valeur de N et les codes convolutifs ont une meilleur performance avec un faible taux de codage.

2.5.3 Performance de l'algorithme SC [9]

Les codes polaires atteignent théoriquement la capacité d'un canal pour une taille de code infinie. C'est le seul code correcteur d'erreurs, à ce jour, pour lequel il est possible de démontrer mathématiquement cette propriété [2]. Il est donc naturel de se demander s'ils peuvent remplacer les codes de la littérature. Pour cela, il est nécessaire de comparer les performances de différents codes, ainsi que leur complexité calculatoire, pour des contraintes équivalentes comme la taille du code ou le rendement.

1- Comparaison avec le turbo code

Dans les courbes qui suivent (figure 2.20 et 2.21) sont regroupées les comparaisons des performances sur un canal à AWGN d'un codeur avec turbo code et de codes polaires.

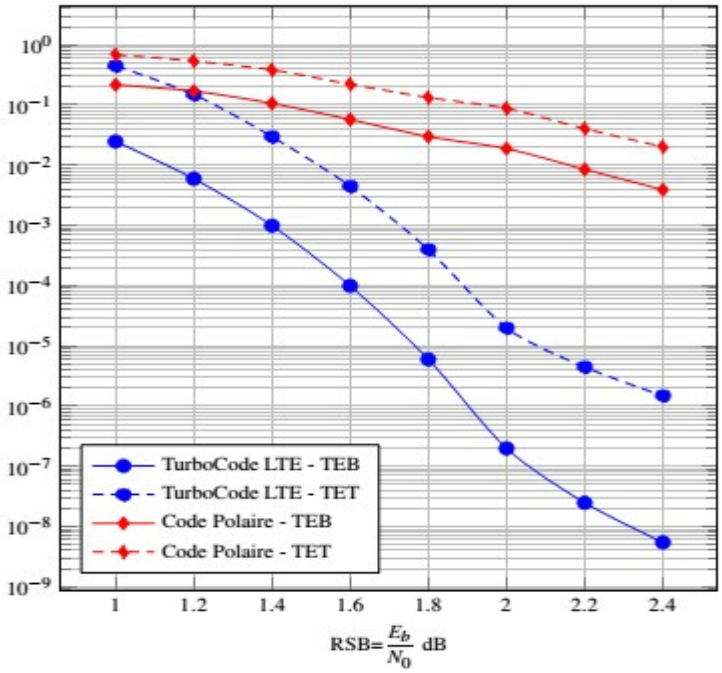


Figure 2.20. Turbo code (1024; 512), m = 3, 6 itérations issu du mémoire de thèse de Sanchez G. (2013) - Code Polaire non systématique, CP (1024; 512), décodé avec un algorithme SC [9].

Interprétation : Au niveau des performances pures, les codes polaires sont en deçà des Turbocodes avec un décodage SC original (non systématique). Une perte d'environ $\mathbf{0.8\ dB}$pour un$\mathbf{TEB} = \mathbf{2.10^{-2}}$. En revanche, la complexité de décodage est bien plus

faible pour un code polaire, en utilisant un algorithme SC, systématique ou non puisque le décodage n'est pas itératif et que l'algorithme SC est très simple à mettre en œuvre car il est récursif et régulier.

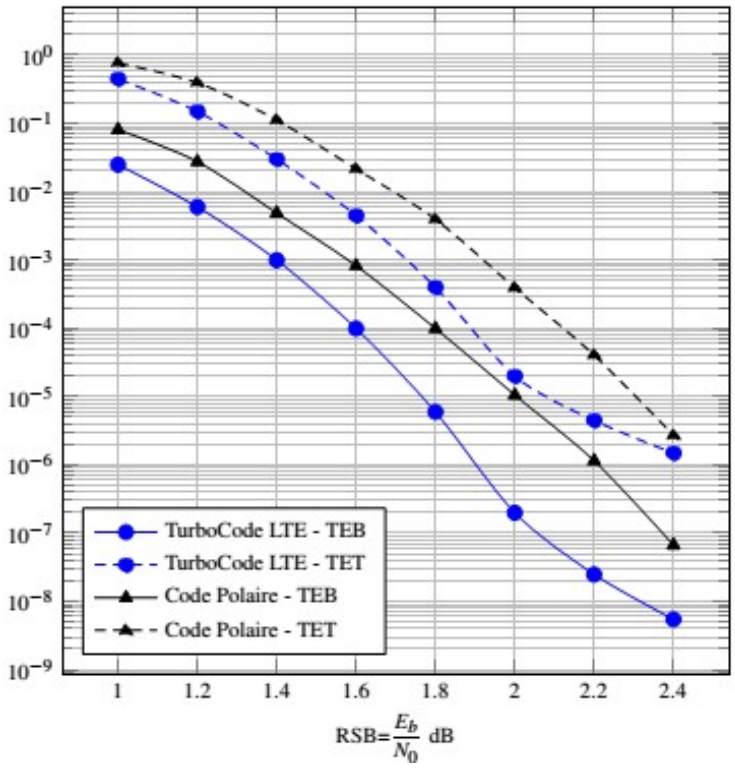


Figure 2.21.Turbocode (1024; 512), m = 3, 6 itérations issu du mémoire de thèse de Sanchez G. (2013) – Code Polaire CP (16384; 8192) systématique, décodé avec un algorithme SC [9].

Interprétation : les performances d'un code polaire systématique ayant une longueur 16 fois plus importante, CP(16384; 8192), sont alors comparées aux performances du turbo code précédent de la figure2.21. L'encodage et le décodage systématiquene modifient pas la complexité de l'algorithme mais permettent d'améliorer les performances de décodage (§ 2.4). C'est la raison pour laquelle un code polaire systématique est retenu. Il apparaît, que dans cette configuration de complexité de décodage équivalente, les performances du code polaire ne sont plus qu'à environ 0:2 dB des performances du turbo code pour un$\mathrm{TEB} = 10^{-4}$.

2- **Comparaison avec les codes LDPC**

Une autre comparaison a été menée cette fois entre un code LDPC et un Code Polaire. Ces comparaisons sont données dans les courbes (figure 2.22 et 2.23).

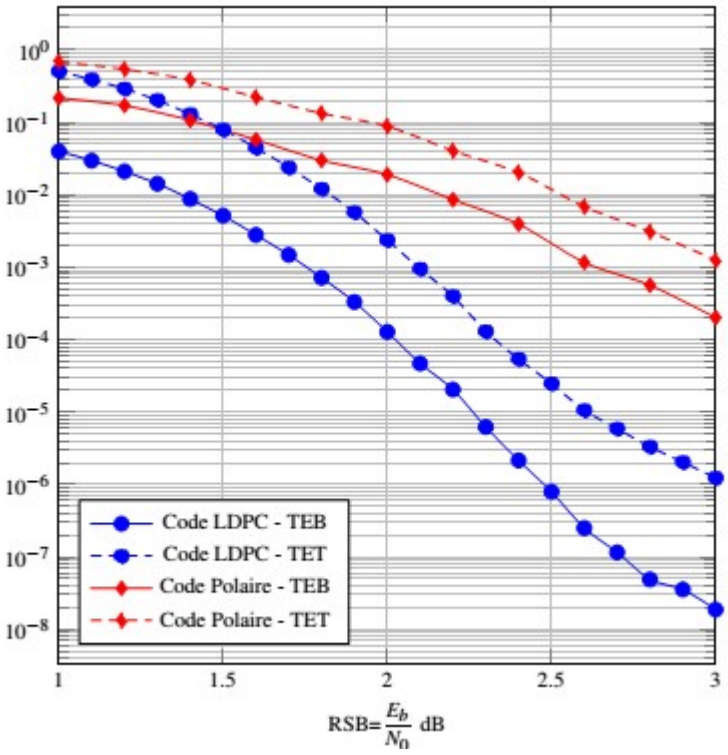


Figure 2.22. Code LDPC (1056; 528) (50 itérations) – code polaire non systématique CP(1024; 512) [9] .

Au niveau des performances pures, les codes polaires sont en deçà des codes LDPC pour undécodage SC originalles Codes Polaires sont en deçà des codes LDPC pour undécodage SC original. Une perte d'environ $\mathbf{0.9\ dB}$ a été observée pour un$\mathbf{TEB} = \mathbf{10^{-3}}$.

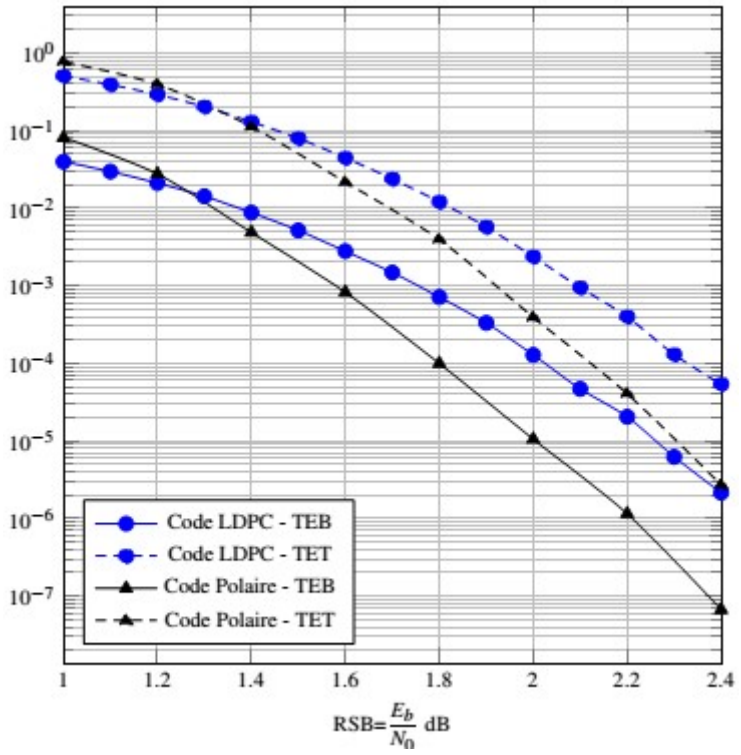


Figure 2.23.Code LDPC (1056; 528) (50 itérations-code polaire systématique CP(16384; 8192) [9].

Les performances d'un code polaire systématique ayant une taille 16 fois plus large, CP (16384,8192), sont alors comparées à celles du code LDPC précédent Il apparaît, que dans cette configuration de complexité calculatoire équivalente, les performances du code polaire sont supérieures d'environ $\mathbf{0.2\ dB}$ pour un $\mathbf{TEB = 10^{-4}}$par rapport au code LDPC.

Cette famille de code correcteur d'erreurs est donc une alternative aux codes LDPC.

2.6 Conclusion

Dans ce chapitre nous avons passé en revue une nouvelle famille de codes correcteurs d'erreurs : les codes polaires. Ils utilisent un nouveau concept appelé la polarisation du canal qui est utilisé pour leur construction.

La construction, le codage et le décodage des codes polaires ont été présentés en détail. L'algorithme de décodage utilisé, appelé SC, a été détaillé car beaucoup de travaux en découlent.

Les codes polaires sont les premiers codes atteignant la capacité des canaux BMS lorsque la taille du mot de code augmente et, avec un choix judicieux du design-SNR dans le cas particulier du canal B-AWGN.

Nous avons également fait une comparaison de performance entre les codes polaires et le turbo code, et entre les codes polaires et les codes LPDC.

Pour une très grande valeur de$N$, les codes polaires sont plus performant plus que les turbo code et les codes LDPC pour un décodage SC original.

Dans le chapitre qui suit, nous allons effectuer des simulations sous le logiciel matlab en estimant le taux d'erreurs binaire (BER) en fonction du SNR afin de trouver le meilleur design-SNR (cf chapitre 3) pour lequel les codes polaires atteignent leur meilleure performance.

Conclusion générale

Il existe de nombreux codes correcteurs d'erreurs qui sont utilisés dans les standards des télécommunications numériques comme les codes LDPC dans le standard DVB-S2ou les Turbocodes dans le standard LTE. Les codes polaires sont également des codes correcteurs d'erreurs. Arikan a prouvé en 2008 [1] qu'ils atteignent la limite de Shannon, mais pour un code de taille infini. En revanche, pour un code de taille faible, si on utilise l'algorithme de décodage SC, ces codes possèdent des performances de décodage inférieures à celle des codes comme les codes LDPC et les Turbo codes pour une même taille de code.Ce qui a poussé les chercheurs à améliorer les performances du décodeur SC pour les codes polaires de faible taille en utilisant un algorithme appelé décodage SC avec List que nous proposons comme porte ouverte à ce travail.

En d'autres termes, les codes polaires nécessitent des codes de grande taille $N \sim 2^{17}$afin d'atteindre des performances de décodage proche d'autres codes correcteurs de taille$N \sim 2^{13}$.

Notons que des études concernant l'utilisation des codes polaires dans les futurs standards de télécommunication sont en cours. Par exemple, la question de l'utilisation des codes polaires dans le standard 5G agite la communauté scientifique.

Enfin, notons que l'un des acteurs majeurs dans les télécommunications a ouvert un livre blanc sur le standard 5G, en présentant les codes polairescomme le code correcteur d'erreurs à employer pour le futur standard 5G [11].

Ce travail de mémoire nous a permis de se familiariser avec la notion de codage canal et ce en étudiants les principaux types de codage utilisés dans les transmissions numérique tel que lescodes en blocs linéaires, les codes convolutifs, les turbo-codes,

les codes LDPC… et en particulier les codes polaires, candidats pour les futurs standards de téléphonie mobile (5G).

La compréhension du principe de ces codes n'était pas aisée vu que ces derniers utilisent la notion de polarisation du canal qui est en elle-même une originalité.

Les simulations qui ont été faites nous ont permis de comprendre la place du design-SNR dans le codage polaire. Nous aurions aimé comparer, du point de vu performance du BER en fonction du SNR, nos résultats avec celles des turbo codes et celles des codes LDPC.

Comme suite à ce travail, nous proposons l'étude de ces codes dans le cas de canaux multi-trajets en utilisant plusieurs schémas de constellation et ce dans le cas de canaux de propagation fixes ou mobiles.

Bibliographie

[1]Arikan, E (2008) : Channel polarization : A method for construction capacity achieving codes, in IEE International Symposium on Information Theory, pages 1173-1177, 2008.

[2]Arikan, E (2009) :Channel polarization : A method for construction capacity achieving codes for symmetric binary-input memoryless channels, IEE transactions on information theory, 55(7) : pages 3051-3073.

[3] B. Cousin et C. Viho : Protection contre les erreurs de transmission, internet de base, 07 janvier 2007, 15 pages.

[4]Camille Leroux :Architecture of polar decoders, Laboratoire de l'intégration du matériau au système (IMS) Bordeaux-INP, 2014.

[5] C. Berrou : Codes and turbo codes, springer, 2010

[6] Cours 6 : Capacité d'un canal-second théorème de Shannon, 34 pages

[7] Damien Regnault : Les codes FEC (Forward Error Correction) et leur utilisation en transmission, 24 mai 2004, 5 pages.

[8] Glavieux Alain : Channel coding in communication networks, Lavoisier, 2005

[9]Guillaume BERHAULT : Exploration architecturale pour le décodage des codes polaires, Electronique, Université de Bordeaux, pages 8-11,2015.

[10]Harish VANGALA, Emmanuel VITERBO and Yi HONG : A comparative study of polar code constructions for the AWGN channel, pages 1-9,2015.

[11]Huawei : achieves 27Gbps 5G speeds with Polar Code, dans des tests d'essais sur champs ,2016.

[12] K. Lahèche : Les codes en informatique : codes détecteurs et correcteurs, Hermès, 1995.

[13 M. Charbit : Système de communication et théorie de l'information, Lavoisier,2003

[14]Proakis, J (2000) : Digital communication, McGraw-Hill Science/Engineering/Math, Bostom, 4 edition, 2000.

[15]Sarah KAMEL, Sécurité à la couche physique, Laboratoire ComElec de télécom Paris Tec, pages 18-24 ,2013.

[16] S. Crozier, P. Guinand and A. Hunt : Computing minimum distance of turbo codes using iterative decoding technics, communication research centre.

[17] S. Hamed Hassani, Polarization and Spatial Coupling: Two Techniques to Boost Performance. Thèse de doctorat. Ecole Polytechnique Fédérale de Lausanne, mars 2013

[18]TAL I. et VARDY A. : List decoding of polar codes. In Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on, 2011, pages 1 –5.

[19] Wasserman David : « Polar Codes », Technical Report 2054.Space Systems Branch/ISR Division. Space and Naval Warfare Systems Center Pacific. San Diego.CA.2014